

# DESIGN AVEC FONTFORGE

UN LIVRE SUR LA CRÉATION DE POLICES  
À L'AIDE DE FONTFORGE

Copyright © 2012 - 2017, Les auteurs de Design With FontForge.

'Design With FontForge' par la communauté FontForge est sous license CC BY-SA 3.0.

Publié avec



**GitBook**

---

# Table des matières

Introduction	1.1
Qu'est-ce qu'un police de caractères?	1.2
Faites confiance à vos yeux	1.3
Planification de votre projet	1.4
Le cadratin (EM)	1.5
Installation de Fontforge	1.6
Configuration de Fontforge	1.7
Aperçu de l'interface utilisateur	1.8
Utilisation des outils de dessin de FontForge	1.9
Dessin avec Spiro	1.10
Création de "o" et "n"	1.11
Infos fonte et métadonnées	1.12
Inter-mot	1.13
Création de l'ADN de votre police	1.14
Lettres capitales	1.15
Interligne	1.16
Ponctuation et symboles	1.17
Finalisation des minuscules	1.18
Diacritiques et accents	1.19
Chiffres	1.20
Gras	1.21
Italique	1.22
Espacement, métriques et crénage	1.23
S'assurer que votre police fonctionne	1.24
Le produit final, génération des polices	1.25
Quand les choses tournent mal avec FontForge	1.26
Conception de polices Devanagari	1.27
Importation de glyphes d'autres programmes	1.28
Ajout de glyphes à une police arabe	1.29
Lectures complémentaires	1.30
Glossaire	1.31

# Introduction

Ce livre a été produit pour aider à rendre le processus de conception de police de caractères accessible à tous. La conception de caractères est visuellement complexe ainsi que hautement technique – cependant il est plus facile que jamais de commencer à créer des polices, largement grâce à la disponibilité d'outils gratuits tel que [FontForge](#). Tout en étant un outil pratique avec lequel commencer, FontForge n'est pas seulement pour les débutants. Il dispose d'un ensemble d'outils avancés et s'améliore rapidement au moment de l'écriture de ce livre.

Ce livre a pour but d'offrir une assistance technique et un aperçu général de la planification d'un projet de conception de police et offre également des conseils sur la façon de rendre votre flux de travail plus efficace.

Si vous souhaitez nous aider, vous pouvez contribuer à l'amélioration de FontForge en donnant des commentaires ou même en contribuant au contenu et à des corrections [sur GitHub](#). Si vous faites face à un bogue dans FontForge, recherchez dans le [suivi de problème](#) sur GitHub pour voir si vous avez rencontré un bogue connu et quel est l'état de sa résolution. Sinon, consultez [ici](#) pour les directives de création de rapport de bogues.

Nous espérons que vous apprécierez lire ce tutoriel autant que nous avons aimé l'écrire.

— *Les contributeurs du projet FontForge*

## Qu'est-ce qu'un police de caractères?

— Qu'est-ce qui rend les polices de caractères différentes de l'écriture manuscrite, de la calligraphie, du lettrage et des logos?

Ce qui est différent dans la conception de polices est la nécessité pour chaque glyphe dans la police de caractères de fonctionner avec tous les autres glyphes. Cela signifie souvent que la conception et l'espacement de chaque partie de la police de caractères finit par être une série de compromis méticuleux. Ces compromis signifient que nous devons voir la conception de caractères comme la création d'une merveilleuse collection de lettres, et non comme une collection de lettres merveilleuses. En d'autres termes, nous devons penser au groupe et à la façon dont il se comportera ensemble et prioriser cela sur tout ce qui est magnifique dans une seule lettre.

Ce besoin de prioriser avec le système plutôt qu'avec n'importe quelle partie unique mène également à un besoin d'analyser notre processus de conception au niveau du système. Les caractéristiques qui s'étendent sur plusieurs lettres deviennent les éléments sur lesquels nous voulons nous concentrer, en particulier au début du processus de conception.

Une autre singularité dans la conception de police est que dans une très grande mesure, les formes que nous concevons sont déjà considérablement établies. Notre tâche en tant que concepteurs de polices n'est pas tant de créer une forme totalement nouvelle mais plutôt de créer une nouvelle version d'une forme existante. Cela peut être déconcertant pour les nouveaux concepteurs de police. Trouver le juste montant à changer pour enthousiasmer mais pas pour s'aliéner un lecteur est une chose délicate. Souvent, les concepteurs restent coincés à penser à des lettres spécifiques. Cette erreur peut être facilement évitée si vous réalisez dès le début que ce qui est le plus significatif dans une police de caractères sont les parties de celle-ci qui se répètent le plus. La conception de police n'est pas seulement de concevoir les caractéristiques des formes communes que nous reconnaissons tous, mais aussi celles des formes qui se produisent le plus souvent.

Il est également utile de reconnaître que ces caractéristiques non seulement aident à créer le ton ou l'atmosphère d'une police, mais aussi à déterminer à quoi la police sera utile ou non, et elles aident parfois à déterminer les contextes technologiques pour lesquels une police est appropriée.

Il peut sembler intimidant ou excessivement abstrait de penser à la conception d'une police de cette façon. Cependant, s'habituer à ces idées est la clé d'un processus de conception de police plus rapide, plus efficace et plus satisfaisant.

Commençons par identifier les principales caractéristiques systémiques dans la conception de police.

### Construction



La construction désigne la structure des traits sous-jacents qui forment un glyphe particulier, c'est-à-dire le squelette du glyphe. Le type de construction à utiliser est sans doute l'une des questions les plus importantes à penser, parce que la construction affecte tant de choix, en particulier si votre conception va sembler quelque peu familière aux lecteurs. Dans l'exemple ci-dessus, la ligne blanche à l'intérieur des lettres indique la construction approximative suggérée par la forme des lettres elles-mêmes.

Cependant, la façon dont les traits se terminent (les 'terminaisons' et les 'empattement' (voir ci-dessous) ne font pas partie généralement de ce que nous entendons par 'construction'. La construction est le squelette du glyphe, tandis que le reste – la largeur, la graisse, les terminaisons – font toutes parties de la chair.

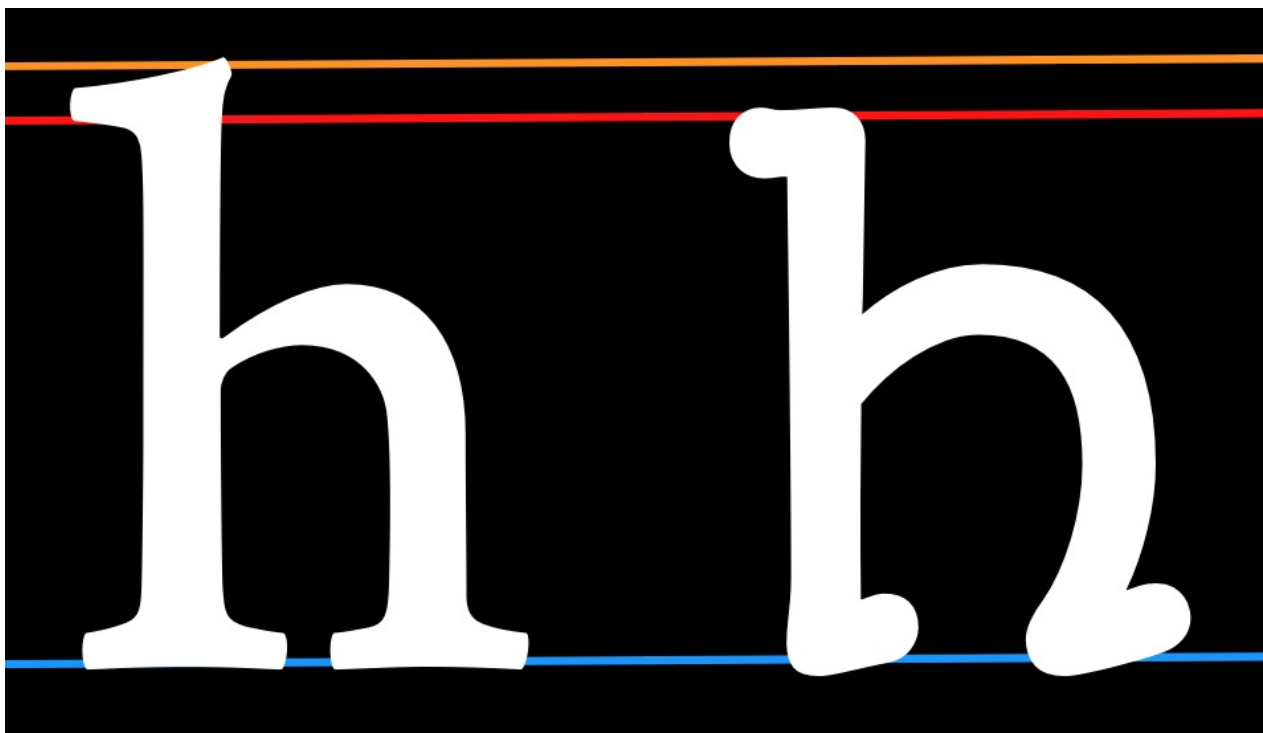


**Proportion de la hauteur d'x à la hauteur de capitale**



Les lettres sur la gauche provenant de [Playfair Display](#) ont une grande hauteur d'x par rapport à leur hauteur de capitale. Les lettres sur la droite sont de [EB Garamond](#) et ont une plus petite hauteur d'x. Dans l'exemple ci-dessus, la taille du H a été ajustée pour qu'ils correspondent.

### Hauteur de l'ascendante



Dans l'exemple ci-dessus, les hauteurs d'x ont été adaptées afin d'illustrer la différence relative des hauteurs de l'ascendante.

Les ascendantes dépassent habituellement la hauteur de capitale de quelque peu, en particulier dans les designs de texte. Dans certains cas, cependant, elles peuvent égaler ou même être inférieures à la hauteur de capitale. Des ascendantes plus longues peuvent ajouter de l'élégance à l'apparence d'une police de caractères. Elles vont souvent de paire avec de plus petite hauteur d'x.

### Profondeur de la descendante



Comme les ascendantes, les descendantes qui sont longues peuvent sembler élégantes.

Prises ensembles, de longues ascendantes et descendantes peuvent devenir difficiles à gérer. Si la police de caractères est utilisée avec de petites hauteurs de ligne, l'allongement moyen des lettres peut causer une collision entre les lignes de texte.

## Largeur



La largeur de design d'une police va modifier non seulement à quoi elle ressemble, mais aussi à quoi elle est utile. L'exemple sur la droite est celui d'une police de texte. L'exemple sur la gauche est celui d'un design d'affichage destiné à être accrocheur. Des lettres qui sont plus étroites que l'exemple de police de texte sont également possibles et peuvent être utilisées pour économiser de l'espace ou pour permettre plus de texte dans un espace plus petit.

### **Largeur régulière versus variable**

Les lettres dans la ligne supérieure de cet exemple montrent une plus grande variété de largeur que les lettres dans la ligne inférieure.



### **Graisse**



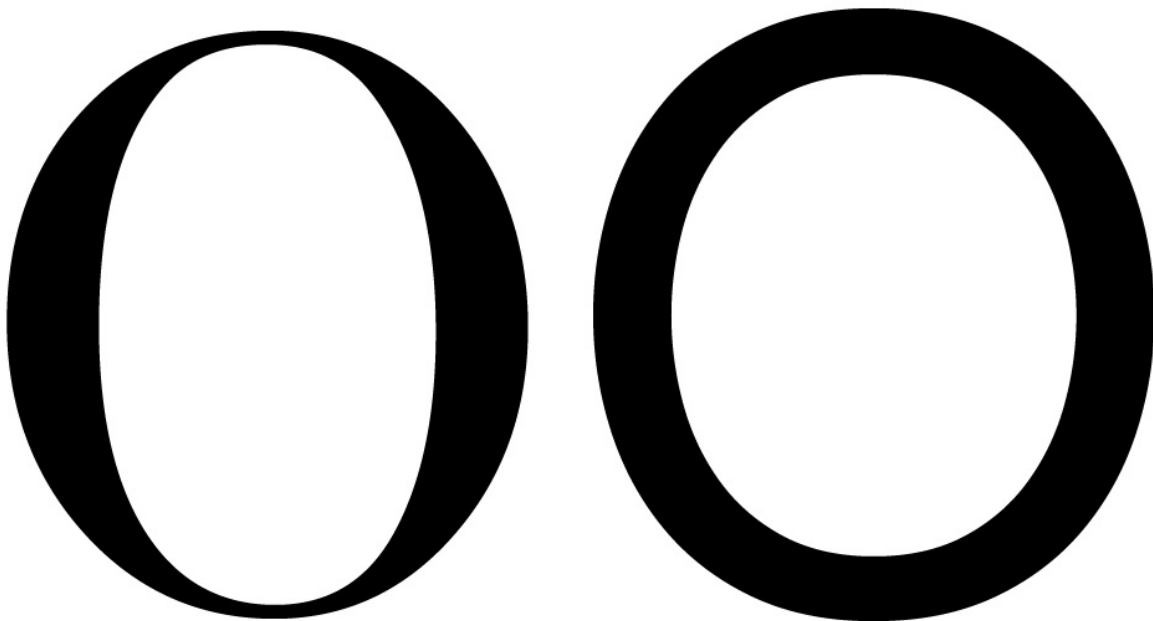
### **Inclination**





## Contraste

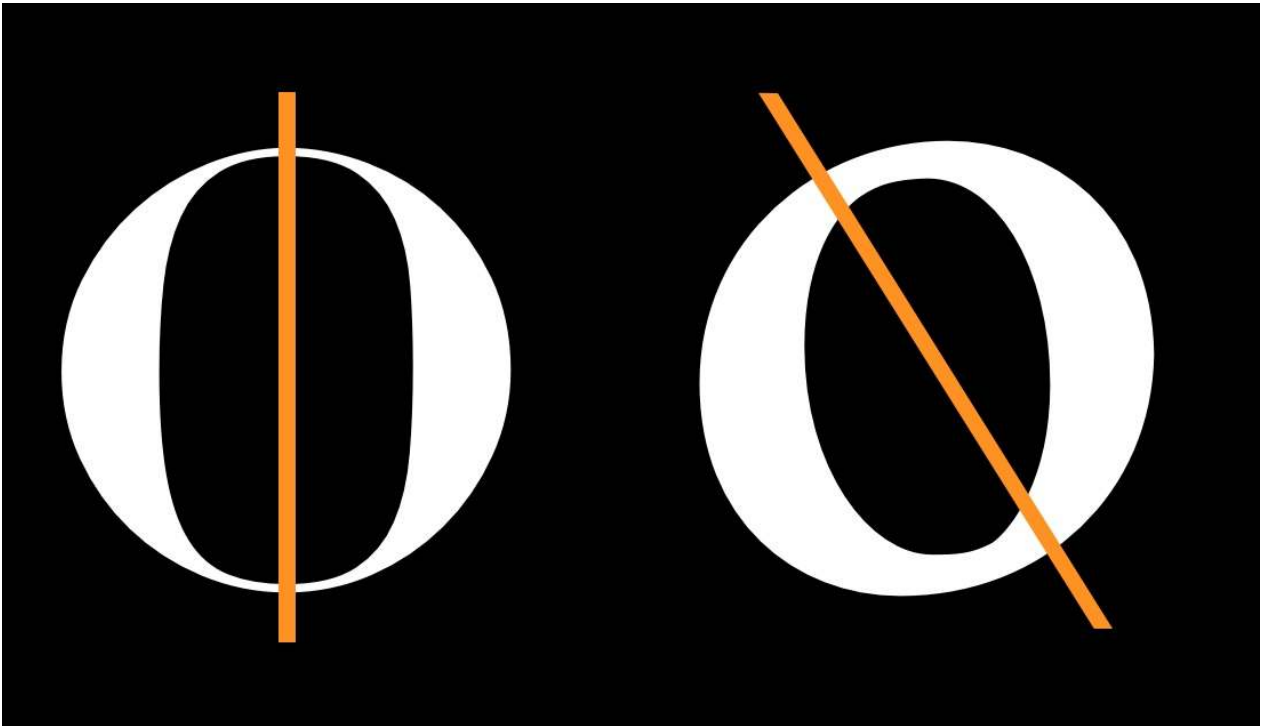
Le contraste désigne la variation de la largeur du trait dans un glyphe. Remarquez dans les deux glyphes 'O' ci-dessous que celui de gauche a une plus grande variabilité dans l'épaisseur de ligne entre le haut et les côtés du glyphe. Les deux glyphes ont un certain contraste, mais celui de gauche en a beaucoup plus que celui de droite.



Une police avec graisse constante (largeur du trait) dans ses formes de lettre ou sans contraste visible se distingue grandement d'une police contrastée. Comme le choix entre serifs ou sans-serifs, le contraste est un choix primordial de la conception de police. Il est intéressant de noter que les designs avec empattement rectangulaire utilisent généralement une largeur de trait constante dans leurs lettres, et que le design des empattements rectangulaires n'est pas seulement à propos des empattements, comme on peut l'imaginer! Il faut se rappeler que les règles de la perception s'appliquent (voir ["Faites confiance à vos yeux"](#)) – le contraste a plus rapport à la façon dont la graisse apparaît que comment elle est mesurée.

## Angle de contraste

Dans l'image ci-dessous, nous voyons que les parties minces des formes de la lettre minuscule 'o' sont différentes. Dans le glyphe à gauche, les points minces reposent sur un axe parfaitement vertical. Dans le glyphe à droite, l'axe est diagonal.



### **Distribution de la graisse**

Si votre police utilise très peu ou pas de contraste, vous n'avez pas vraiment besoin d'y penser. La plupart des fontes, cependant, ont au moins un certain degré de contraste. Dans ce cas, vous avez une grande variété d'options à choisir quand il s'agit de distribuer la graisse dans votre fonte.

#### **Verticale**



Une distribution verticale de la graisse est très commune. Le 9 et le 8 ci-dessus sont des exemples particulièrement frappant.

### Horizontale



Une distribution de graisse horizontale est beaucoup moins commune, mais est néanmoins visible dans de nombreuses polices.

### Forte à la base



**Forte au sommet**



**Irrégulière**



## Fûts



Il est facile de supposer que vos fûts seront tout simplement droits et que ce n'est pas une préoccupation réelle, mais à la fois la graisse et la forme de vos fûts sont des choix délibérés que vous pouvez et devez faire.

## Jonctions



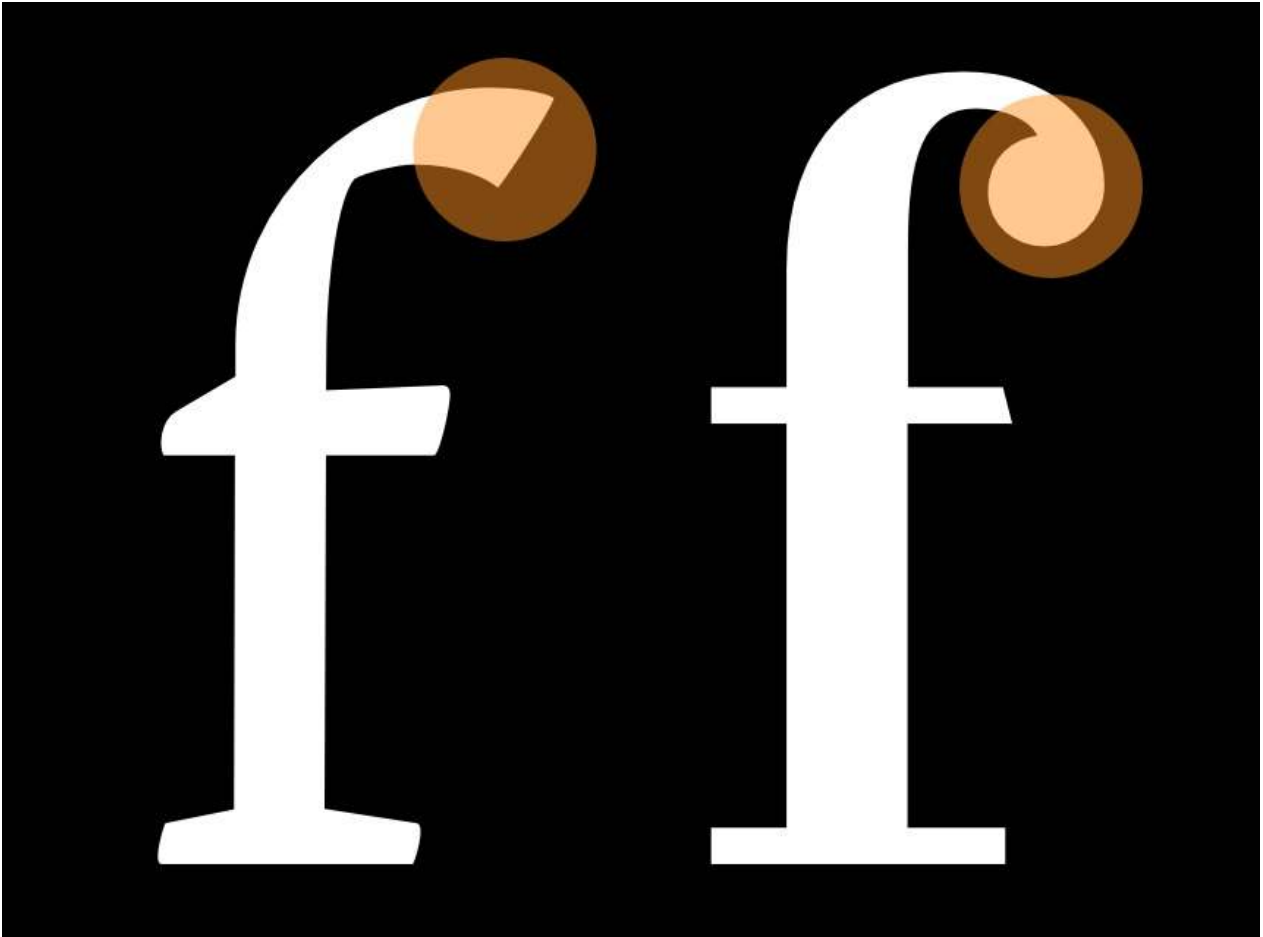
### Panses

Notez que les panses sont les traits courbes dans les illustrations ci-dessous et non les formes noires intérieures. Les traits internes sont appelés "contrepointons". Lors de la conception d'une police, vous devrez souvent altérer votre travail non pas en raison de la forme ou la largeur du trait, mais en raison de la forme et de la taille du contrepointon.



### Terminaisons

Les terminaisons sont les formes aux extrémités des traits. Ce ne sont pas des empattements. Elles sont souvent perpendiculaires à l'angle du trait à son extrémité, ou ciselés horizontalement ou verticalement. Elles peuvent également refléter la forme de la pointe ou des autres outils de traçage que les lettres évoquent.



### Vitesse



Le 'n' de gauche semble être écrit beaucoup plus vite ce celui de droite. La vitesse est discutée plus en détail dans le [chapitre sur les italiques](#)

### Régularité



Les caractéristiques suivantes ne sont pas présentes dans tous les designs de police, mais ce sont des variables qui peuvent faire partie de votre conception. Si tel est le cas, il convient de considérer le degré auquel elles joueront un rôle.

## **Ornements**





Notez que dans la police du dessus, les ornements sont plus présents dans la lettre majuscule et que dans celle du bas, les ornements le sont plus dans la minuscule.

## Empattement — Être ou ne pas être

Les empattements sont l'un des aspects les plus distincts d'une police de caractères, et souvent la première classification de police est de distinguer les polices avec empattements (serif) et sans empattements (sans-serif).

Ce choix affecte à quoi les terminaisons vont ressembler. Les empattements peuvent être des deux côtés ou d'un seul côté. Ils peuvent être perpendiculaires au trait ou avoir leur propre direction (comme être toujours horizontal ou vertical). Les empattements peuvent être avec ou sans congé. Tout design d'empattement est un mélange de tout ce qui précède appliqué uniformément au design de la police avec quelques déviations pour des lettres particulières, en particulier les 'S', 'C' et 'Z' (c.-à-d. une police avec empattement horizontale pour toutes les lettres aura souvent s, c et z avec un empattement vertical).

Il existe une légende urbaine affirmant que les polices avec empattement sont plus faciles à lire que sans empattement – cela est [seulement un mythe](#), jusqu'à nouvel ordre.

La forme des empattements est liée aux formes des terminaisons.

## Congés

Les parties en coin d'un empattement où elles se connectent au trait principal sont appelées 'congé'. Un design particulier peut donner une apparence douce à l'empattement ([Times New Roman](#) est un exemple) ou peut ne pas inclure aucun congé. Certains designs utilisent également des empattements seulement d'un côté ou avec des proportions différentes sur les deux côtés.

Il s'agit d'un paramètre relativement fort qui donne une touche spéciale à la police – élégance (congé doux ou grands de [Times New Roman](#)) ou lourdeur (congé absents de [Arvo](#)).

## Empattements rectangulaires

Également appelé de type mécanique ou égyptien, les empattements rectangulaires sont épais et en bloc. Les empattements rectangulaires n'utilisent pas de congé. D'une manière générale, le design de police avec de tels empattements est considéré comme ayant moins de contraste dans leurs glyphes – [Rockwell](#), [Courier](#) ou [American Typewriter](#) reflètent cela.

On peut supposer que les empattements rectangulaires ont été utilisés pour ajouter un peu d'ornement ou de rythme à un design de police autrement sans contraste. Mais ce n'est pas une règle absolue.

## Terminaisons d'empattement

Tout comme les terminaisons de lettres, la forme finale des empattements eux-mêmes contribue à l'aspect de la police – soit doux ou lourd. Les terminaisons d'empattement peuvent être douces et arrondies ([Courier](#)) ou émoussées et angulaires ([Rockwell](#)).

## Décoration



## Dimension



## Faites confiance à vos yeux

Le design de polices est le processus de tester itérativement des choix individuels qui, collectivement, composent le design complet. Vous allez tester votre police pour voir si la combinaison de décisions que vous avez prises:

- Permet de lire la police
- Confère à la police l'aspect que vous désirez
- Rend la police utile pour les travaux que vous voulez que la police soit capable de faire.

Lorsque vous testez le design, vous devez faire confiance à vos perceptions et à votre design de façon pratique. Une grande partie du design de police nécessite que vous fassiez des lettres similaires et que vous répétiez des formes.

Il est tentant de supposer que si vous mesurez les parties et les espaces entre les glyphes, vous obtiendrez des résultats fiables. Bien que très utile, cette approche présente des limitations réelles. Vous devriez vous attendre à faire des ajustements si quelque chose vous semble incorrect. De plus, vous devriez vous sentir confiant que de faire des changements jusqu'à ce que cela "semble bon" soit la bonne chose à faire.

La raison pour laquelle cela est vrai, c'est qu'il y a un certain nombre d'illusions d'optique naturelle auquel tous les lecteurs sont soumis. Ces illusions doivent être prises en compte en modifiant les formes des lettres jusqu'à ce qu'elles vous semblent bien.

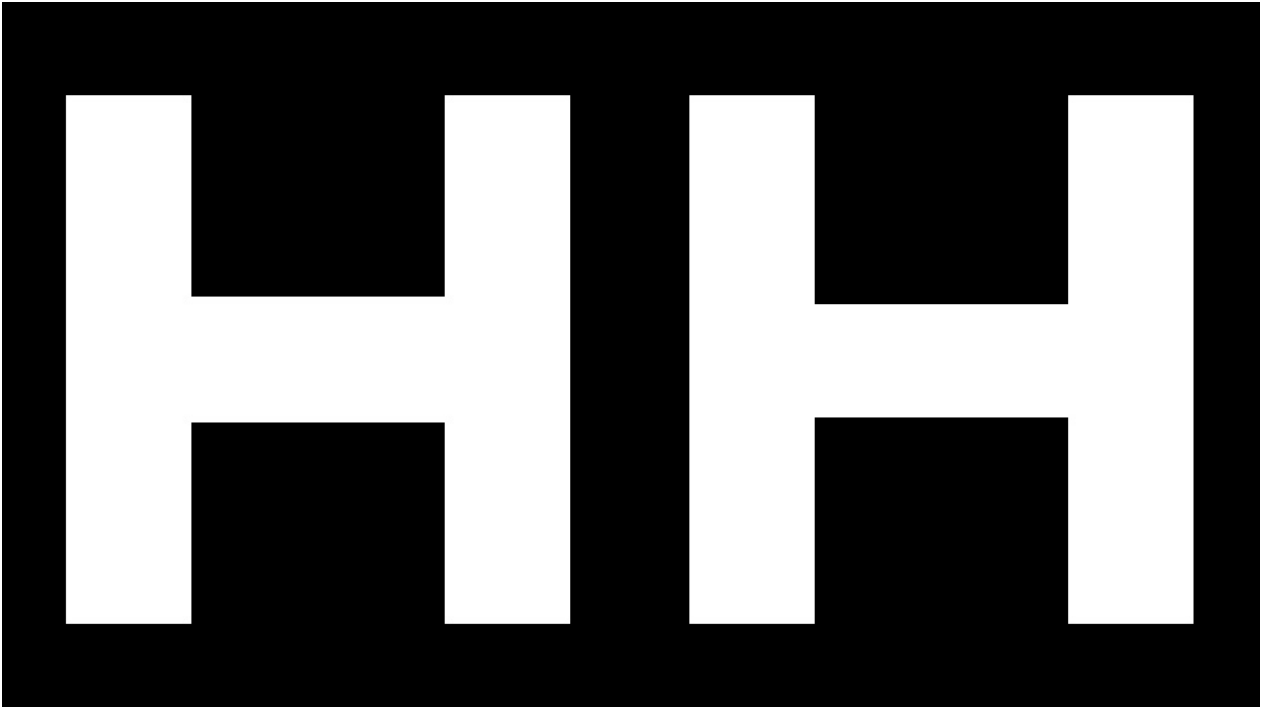
Pour avoir une idée de par où commencer et quels éléments ajuster, consultez ces [vidéos de révision de police](#) (lien externe)

## Exemples d'illusions

Certaines illusions impliquent la graisse perçue des lignes, certaines impliquent la longueur perçue des lignes, et d'autres impliquent la perception de l'œil des formes.

### Graisse horizontale vs. verticale

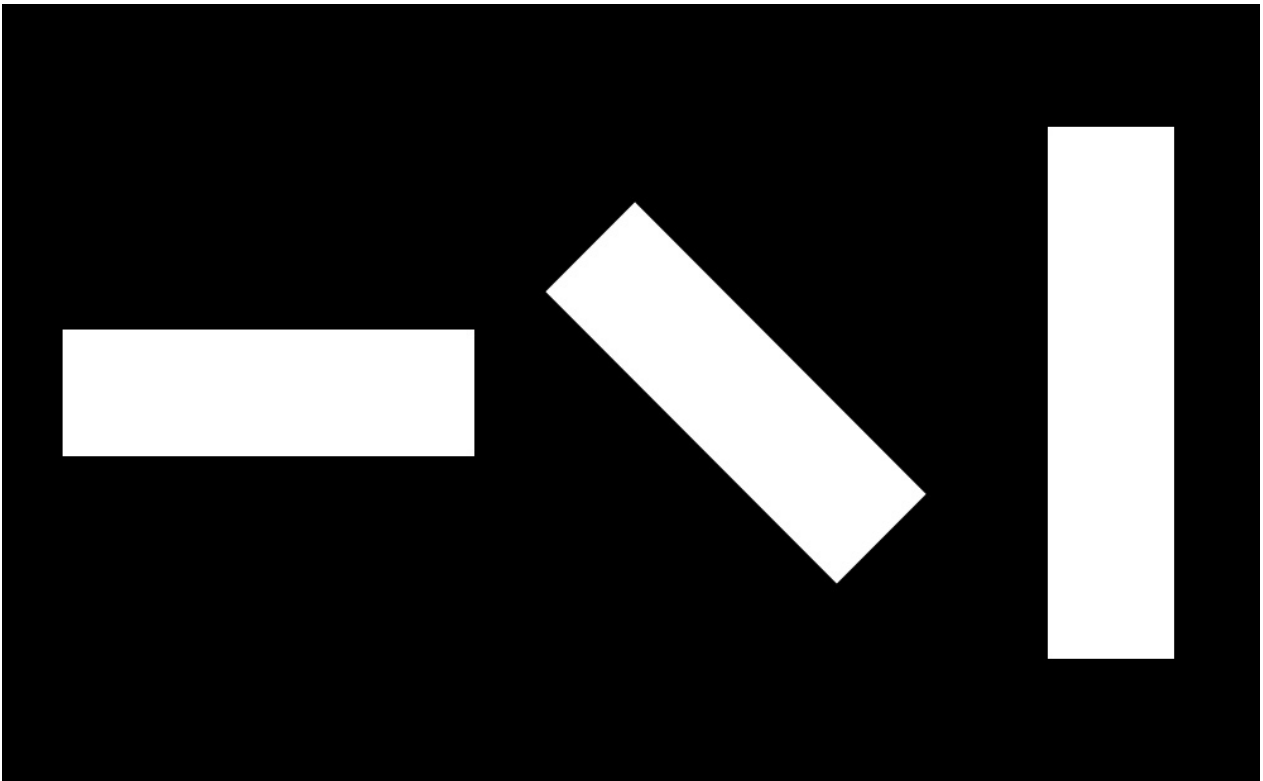
L'exemple de gauche montre un 'H' dont les barres sont exactement égales en épaisseur. Cela semble incorrect. Pouvez-vous le voir? L'autre sur la droite a une barre horizontale qui a été amincie pour apparaître égale en épaisseur.



Les glyphes dans lesquels des ajustements optiques ont dû être faits sont nombreux et incluent A, E, F, L, H, f, t et z.

### Épaisseur de diagonale

De même, si vous avez des barres de la même largeur et que l'une d'elles est placée en diagonal, la barre diagonale semblera légèrement plus lourde que la barre verticale et légèrement plus mince que l'horizontale. Si vous voulez que cela paraisse bien, vous devrez l'ajuster pour être plus légère que horizontal, mais juste un peu moins.

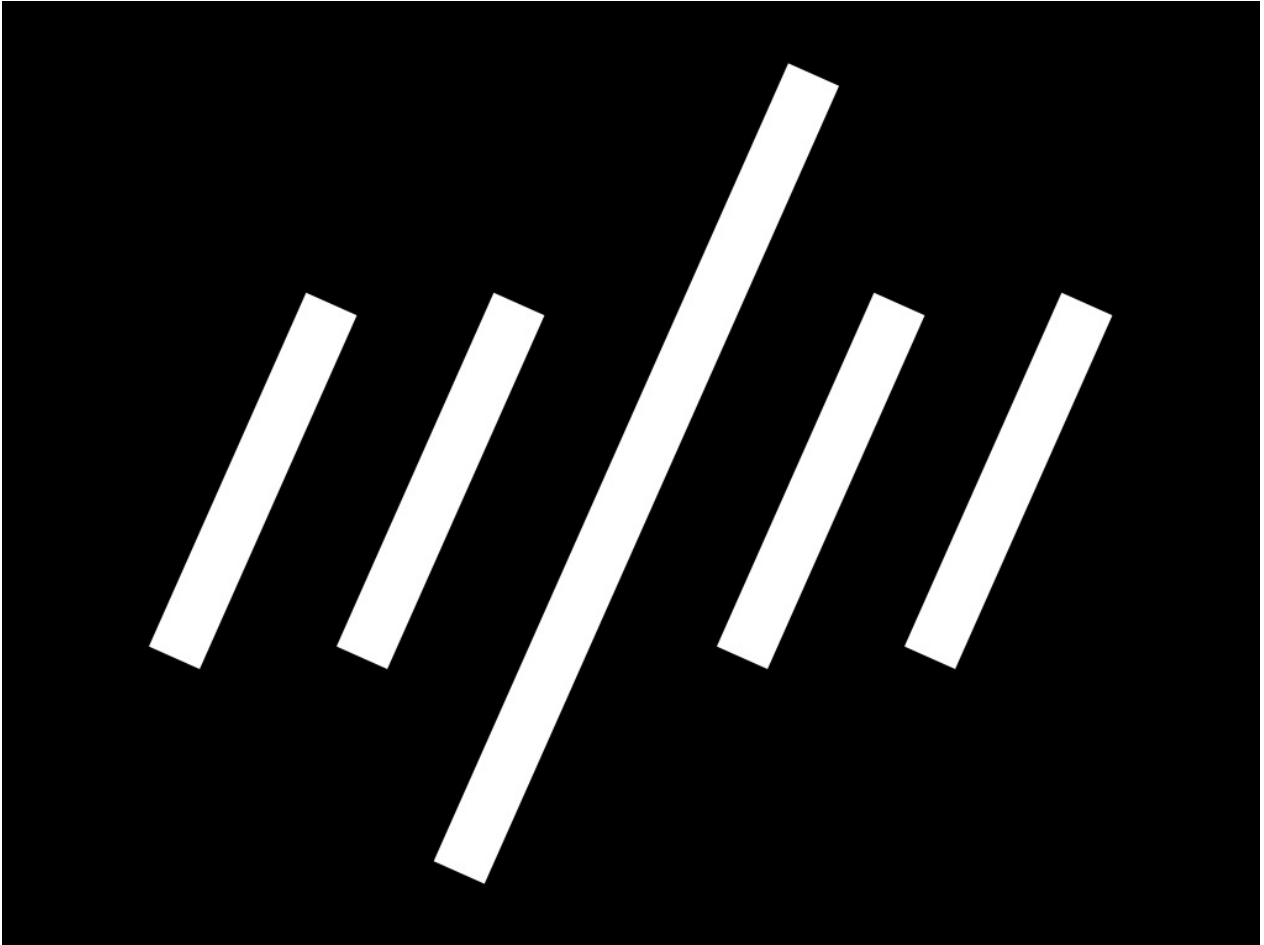


Les glyphes dans lesquels cette perception humaine peut être importante sont assez nombreux et incluent k, K, N, Q, R, v, V, w, W, x, X, y, Y, 7, 2, &, †, ‡, ø, Ø, √, /, ‹, ›, «, », ½, ⅓, ¼, ≤, ≥, et ×.

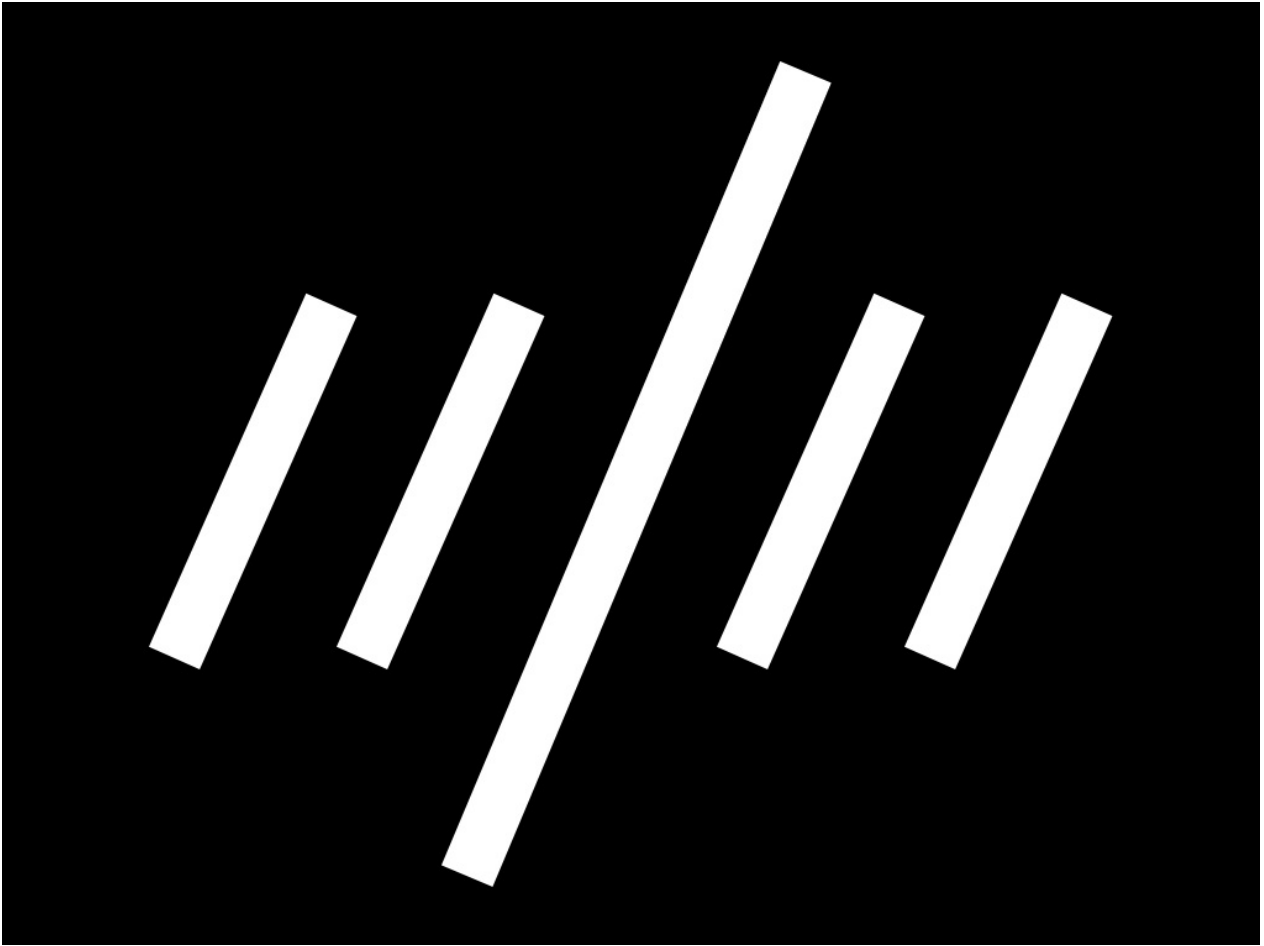
### Longueur et angle de diagonale perçu

Les formes plus longues ont besoin de moins d'inclinaison que les plus courtes pour donner l'apparence d'une même inclinaison.

L'image ci-dessous a des lignes diagonales qui ont toutes le même angle. Les plus longues semblent être à un angle différent.



Dans l'image suivante, l'inclinaison de la ligne la plus longue a été ajustée:

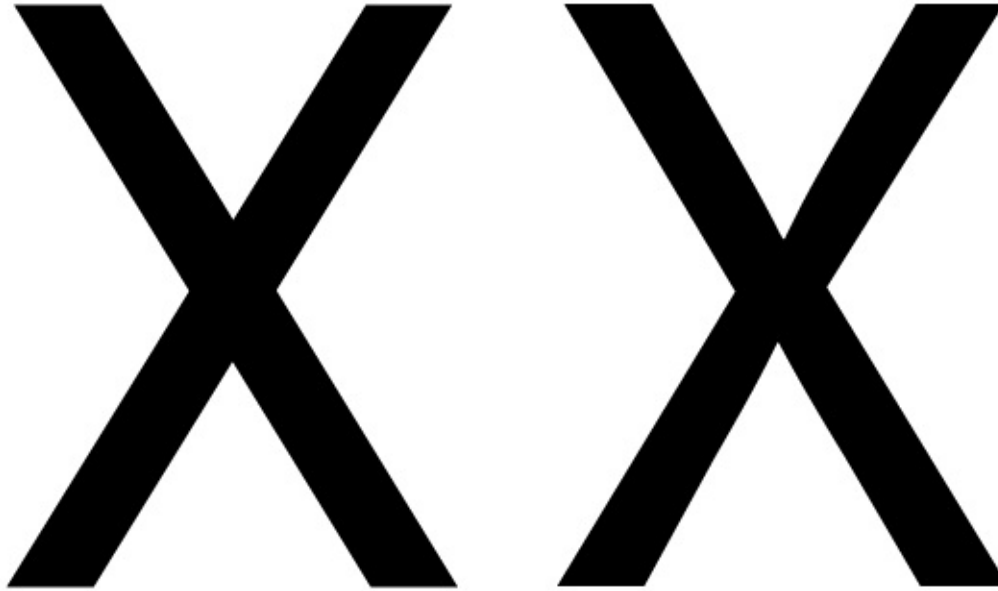


Maintenant, regardons une vraie italique, en appliquant ces corrections aux glyphes:



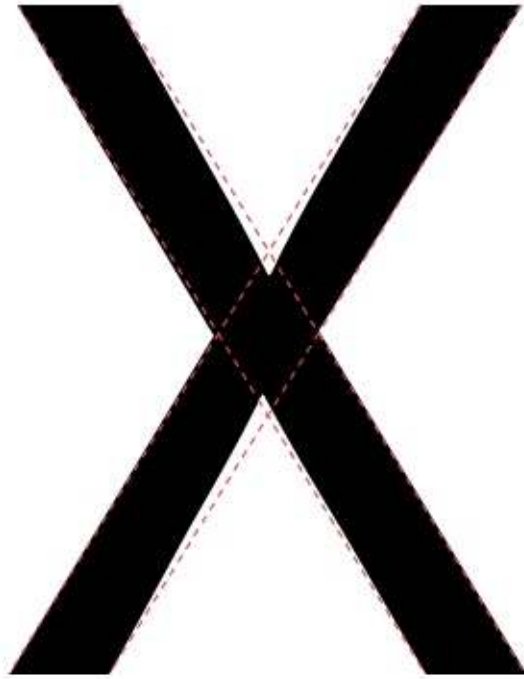
**Croisement des diagonales**

Lorsqu'une barre franchit une autre diagonale ou une ligne droite, elle aura besoin d'être ajustée pour ne pas apparaître comme étant mal alignée.



Dans l'exemple ci-dessus, le X à gauche comporte deux barres non ajustées qui se croisent. L'exemple de droite a été ajusté de façon à ce qu'elles semblent alignées.



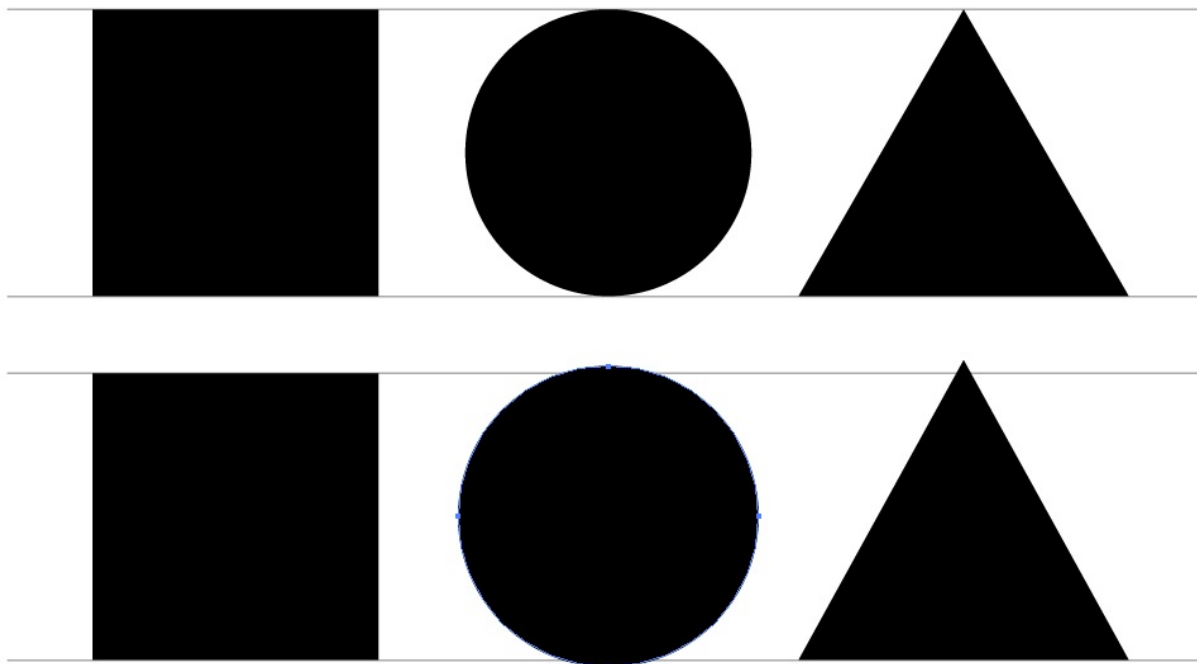


Comme vous pouvez le voir dans ce X avec une ligne pointillée au-dessus, le X qui apparaît visuellement aligné implique un décalage.

Les glyphes dans lesquels cette illusion est applicable incluent x, X, k, K, ×, # et la lettre islandaise 'eth' (ð).

### **Hauteur perçue**

La forme d'un glyphe contribuera à la hauteur qu'il doit avoir afin de paraître de même hauteur que les autres glyphes. Les glyphes ronds doivent dépasser un peu la hauteur des glyphes plats. Les glyphes qui ont des formes pointues devront les surpasser davantage. Plus la forme est nette, plus elle aura besoin de dépasser pour avoir l'air correcte.



Dans l'image ci-dessus, les trois formes supérieures sont non corrigées — c'est-à-dire qu'elles ont des hauteurs identiques. Les trois formes inférieures ont été ajustées afin qu'elles apparaissent les mêmes en hauteur.

Cette illusion est applicable à tout glyphe qui a des parties qui sont soit rondes ou pointues. Ceux-ci comprennent O, Q, C, S, A, V, W et ainsi de suite.

## **Vous êtes pleinement qualifié pour corriger ces illusions**

Parce que vous pouvez voir à la fois l'illusion et l'effet de la correction sur l'illusion, vous serez en mesure de faire ces corrections par vous-même. Il suffit de faire confiance à vos impressions.

## **Testez l'aptitude à l'emploi**

Comme vous êtes capable de voir des illusions d'optique et de les corriger, vous avez également la capacité de dire si une police fonctionne pour l'utilisation spécifique (ou les utilisations) que vous avez à l'esprit. C'est là que vous devriez également faire confiance à votre jugement.

Tout à fait séparément, il est intéressant de noter qu'aucune police ne peut être évaluée en dehors de la façon dont elle est utilisée et ce à quoi elle servira. C'est pourquoi il est essentiel de commencer à tester dès le début du processus de conception, et de continuer à tester jusqu'à ce que vous sentiez que le projet est terminé.

À quoi ressembleront ces tests? Les tests seront simples au début, vous permettant de tester les premiers choix de design. Comme votre design devient plus complet, vos tests devront suivre le rythme et vous permettront d'évaluer le succès relatif ou l'échec des nouveaux choix que vous avez fait — ou, mieux encore, de comparer deux (ou trois, ou plus...) options que vous envisagez.

Parfois, vous trouverez que vous devez enlever et modifier un choix de design que vous pensiez fonctionnel. C'est normal. Pour créer une police, il faut équilibrer de nombreuses variables et des surprises se produisent souvent. Plus vous concevez des polices, plus vous aurez d'expérience dans ces choix arbitraires.

En approchant de la fin du processus, si la police doit être utilisée de façon simple, les tests devraient également rester simples. Cependant, si une police doit être utilisée de nombreuses façons ou dans un large éventail d'environnements d'impression ou d'écran, elle doit être testée dans toutes ces situations, ce qui inclut l'impression de divers échantillons de la police.

Avoir une idée bien définie de l'utilisation finale peut vous faire épargner du temps de design. Cependant, cela n'est pas toujours possible et vos idées peuvent évoluer. La clé est de penser et de définir les cas d'utilisation aussi complètement que vous pouvez, puis de veiller à ce que vos tests gardent le rythme avec les questions que vous vous posez tout en concevant la police.

## Lecture complémentaire (en anglais)

- <http://typographica.org/on-typography/making-geometric-type-work/>
- <http://typedrawers.com/discussion/1085/the-letter-s>

## Planification de votre projet

Maintenant que vous avez une idée de la façon dont un design de police peut varier, vous pouvez décider si votre projet aura une seule fonte, si ce sera une collection de plusieurs fontes inter-connexes, si ce sera une famille de trois ou quatre styles (maintenant classique) ou si ce sera quelque chose de plus grand encore.

Les styles communs de familles de polices comprennent:

- Une graisse régulière et grasse
- Régulier, gras, italique – éventuellement avec un italique gras
- Mince, léger, book, régulier, semi-gras, gras, extra-gras, noir
- Régulier, condensé, gras et gras condensé
- Étroit, condensé, large et extra large
- Régulier, semi-orné, orné, très orné, extrêmement orné.

Bien qu'il existe des raisons pour lesquelles des modèles typiques dans les familles existent, vous pouvez décider que vous voulez un genre très différent de regroupement.

La portée du projet peut être déterminée exclusivement par votre ambition et votre quantité de temps libre. Mais les objectifs de projet sont souvent déterminés par l'utilisation que vous avez pour la collection ou la famille de polices, ou, encore plus, par les besoins de votre client. Certes, pour les concepteurs de type professionnel, les deux dernières questions sont généralement les facteurs déterminants.

## Sentiment

L'aspect le plus important d'un design de police est le sentiment qu'il évoque. C'est notoirement difficile à verbaliser, mais c'est ce qui rend une police de caractères significativement différente de toute autre.

Un concepteur de police du Portugal, Natanael Gama, a conçu la [famille Exo](#) avec FontForge. Sur sa page d'accueil, il décrit un autre projet pour le sculpteur [John Williams](#) et inclut un graphique montrant son exposé dans une matrice de continuums de sentiments:

- Figuratif à abstrait: 50%
- Gracieux à robuste: 30%
- Calme à énergique: 0%
- Étonnant à simple: 15%
- Expérimental à standard: 15%
- Prestigieux à ordinaire: 15%
- Autres idées: Beau, Espaces extérieurs, Condition humaine

## Couverture des glyphes

Une police est toujours une police, même si elle ne contient qu'un seul glyphe. Mais une police peut également avoir quelques centaines voire des milliers de glyphes. Si votre projet est auto-initié, alors ce choix est finalement arbitraire. Vous pouvez décider que vous voulez seulement des capitales, ou que vous voulez inclure les glyphes trouvés dans les autres polices que vous utilisez. Si vous travaillez pour un client, vous voudrez peut-être préciser la langue ou les langues que la police est destinée à soutenir. Votre objectif pourrait également être d'étendre une police existante, en ajoutant quelques glyphes pour l'utiliser dans une ou plusieurs langues supplémentaires.

C'est certainement une bonne idée de faire ce choix délibérément, et de rester du côté sûr en incluant moins plutôt que trop. Souvent au fur et à mesure que la police de caractères est réalisée, il peut être tentant d'inclure de plus en plus de glyphes – mais il est souvent plus bénéfique de continuer à améliorer l'ensemble de glyphes que d'en ajouter de nouveaux.

## Flux de travail pour famille multi-styles

Si vous savez dès le début que vous aurez plus d'une fonte, vous économiserez du temps si vous planifiez et construisez la famille de fontes systématiquement, et travaillez sur les styles en parallèle, plutôt que de compléter un style à la fois.

Il est bien sûr impossible de créer *chaque* style d'une manière complètement parallèle, mais il est possible de terminer une étape de conception donnée pour chaque style afin de vérifier et d'être sûr des relations entre les styles, au début du processus. Vous trouverez peut-être qu'il est utile de terminer un ensemble complet de lettres de test (comme "adhésion") pour une version régulière, puis de faire des "adhésion" pour les autres styles suivants. Cependant, vous pouvez également rendre le processus encore plus granulaire et prendre des décisions sur des parties spécifiques des lettres de base (comme le 'n' et 'o') pour tous les styles ensembles.

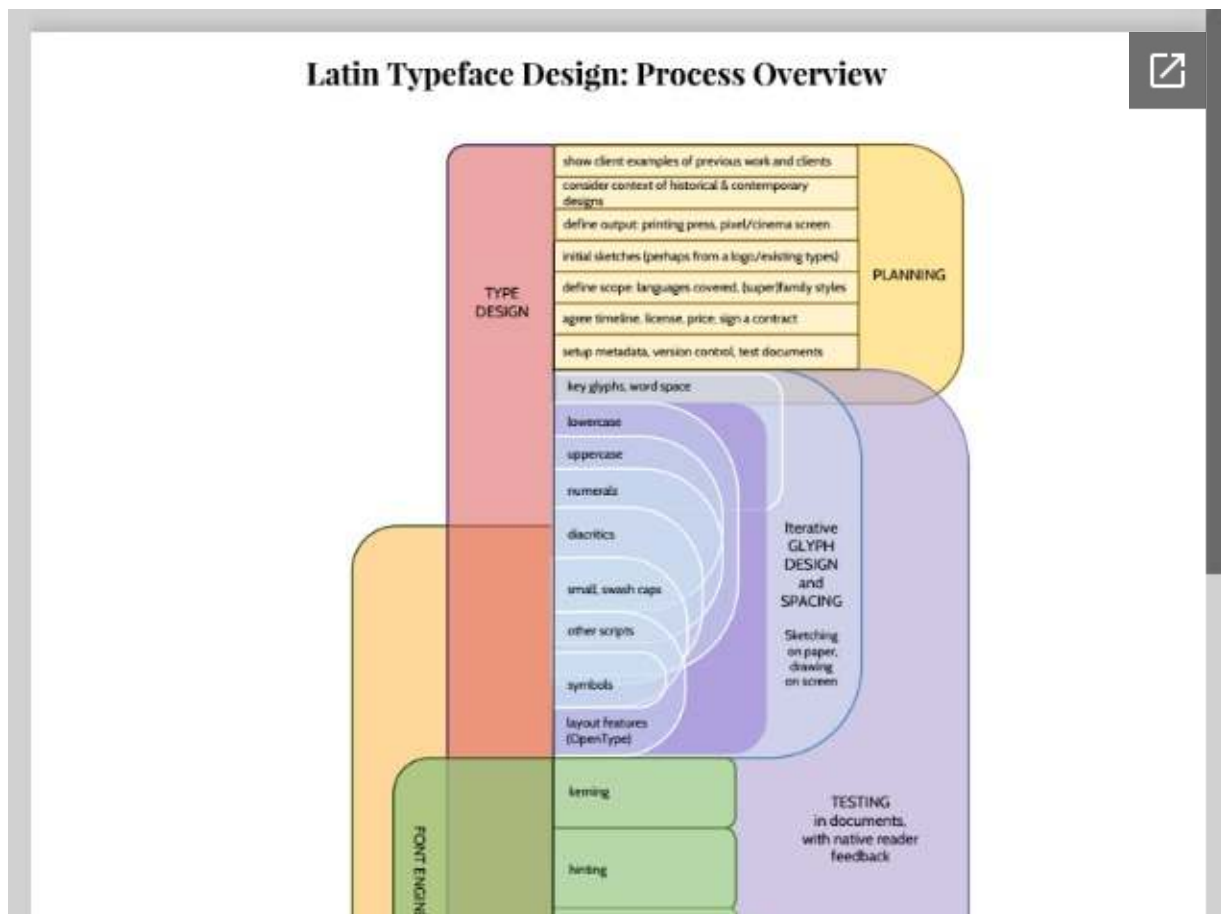
En fonction de la taille et de la composition de la famille que vous planifiez, vous constaterez que de créer des exemples qui peuvent être interpolés de glyphes pourrait vous faire gagner du temps, non seulement pour interpoler des styles intermédiaires, mais pour aider à faire des choix de conception au sujet de ces variables typographiques qui se répètent à travers les membres d'une famille. Pour une description des variables que vous devriez envisager, consultez le chapitre "Qu'est-ce qu'une police de caractères?".

## Technique: Gestion des versions

Vous devriez apprendre à utiliser Git et Github pour stocker vos fichiers et utiliser le format "SFDiR" pour vos sources.

- <https://help.github.com/articles/what-are-other-good-resources-for-learning-git-and-github>
- <http://justinhileman.info/article/git-pretty/>

## Processus général



## Environnements de test

Lors de la planification de votre projet, vous devez considérer le médium typographique auquel la police de caractères sera destinée principalement et secondairement: Mobile et web, ou projecteurs numériques, ou imprimantes à jet d'encre et laser de bureau à bas prix, ou imprimantes laser pour impression haut de gamme, ou impression offset de magazine, ou impression de journaux à grande volume... et ainsi de suite. Vous devriez alors essayer d'acquérir ou avoir accès à ces technologies de composition, afin que vous puissiez voir les résultats réels de votre travail.

Tout au long du processus de conception de police, vous trouverez très utile de prévisualiser un jeu de texte avec votre police (prototype) à une résolution supérieure à celle de votre ordinateur portable ou de votre poste de travail. Cela signifie typiquement une imprimante laser avec «vraie» DPI de 1200 et Adobe PostScript 3. Pour les particuliers, il est possible d'en acheter une pour environ 500 \$. En 2013, ces imprimantes étaient recommandées:

- HP P2055d
- Xerox Phaser 4510
- Xerox Phaser 5550
- Nashua / Ricoh P7026N

En mai 2013, le studio [Production Type](#) offrait une Xerox 7525 avec un contrôleur "fiery", qui coûtait environ 12 000 € à l'achat mais qui était de 300 € par mois en location avec toner, pièces et maintenance . À la fin de 2015, Octavio Pardo a loué un [Xerox Phaser 7100](#) de manière similaire pour 30 € par mois.

## Fonctionnalités OpenType

Vous pouvez planifier les fonctionnalités OpenType de votre projet avant de commencer à dessiner. Les caractéristiques communes incluent:

- `liga` Ligatures
- `onum` , `lnum` Numéros

Pour certaines langues `locl` fonctionne mais pour d'autres il ne marche pas, il est donc préférable d'exposer des formes spécifiques à la langue via `locl` et `ssNN` OU `cvNN` .

La spécification OpenType permet certaines caractéristiques qui ne sont pas recommandées:

- Fonction `hist` . Lire la suite dans cette [discussion sur TypeDrawers](#).

## Lecture complémentaire (en anglais)

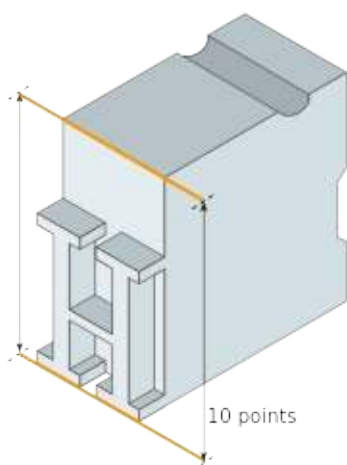
- Aoife Mooney's presentation on the type design process at TypeCon 2014: <https://vimeo.com/107421895>
- TypeDrawers discussion of [Printer recommendations for proofing](#)

## Le cadratin (EM)

— Aussi appelée 'em size' ou 'UPM' en anglais. Dans une police, chaque caractère est placé dans son propre espace défini. Dans la police de métal traditionnelle cette espace était le bloc de métal réel de chaque caractère. La hauteur de chaque morceau de caractère était uniforme, permettant aux caractères d'être rangés soigneusement en rangées et en blocs (voir ci-dessous).



La hauteur d'une police est connue sous le nom de 'em' ou cadratin, et elle provient de la largeur du caractère majuscule 'M'; il a été fait en sorte que les proportions de cette lettre soit carrée (d'où la dénomination 'cadratin'). La taille em est la taille de point sur laquelle une police de métal est calculée. Ainsi, une police de 10 points a une taille em de 10 points (voir ci-dessous).



Dans une police numérique, l'em est une quantité d'espace définie numériquement. Dans une police OpenType, l'UPM – ou la taille em est généralement fixé à 1000 unités. Dans les polices TrueType, l'UPM est par convention une puissance de deux, généralement définie à 1024 ou 2048.

Lorsque la police est utilisée pour définir le caractère, l'em est mis à l'échelle à la taille de point souhaité. Cela signifie que, pour une police de 10 pt, les 1000 unités par exemple sont réduites à 10 pt.

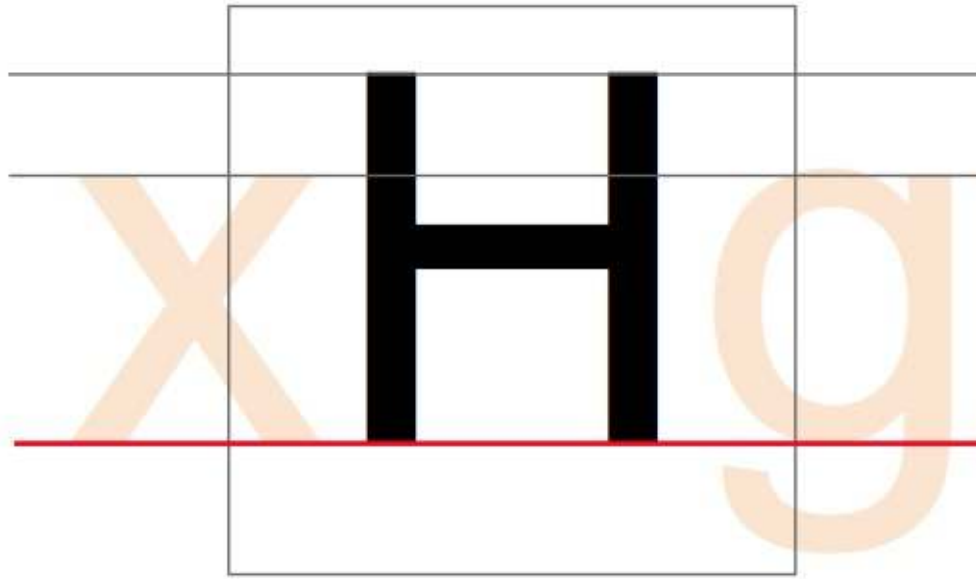
Donc, si votre 'H' majuscule est de 700 unités de haut, il sera de 7 pt de haut sur une police de 10 pt.

## Mise en place dans la fenêtre de glyphes

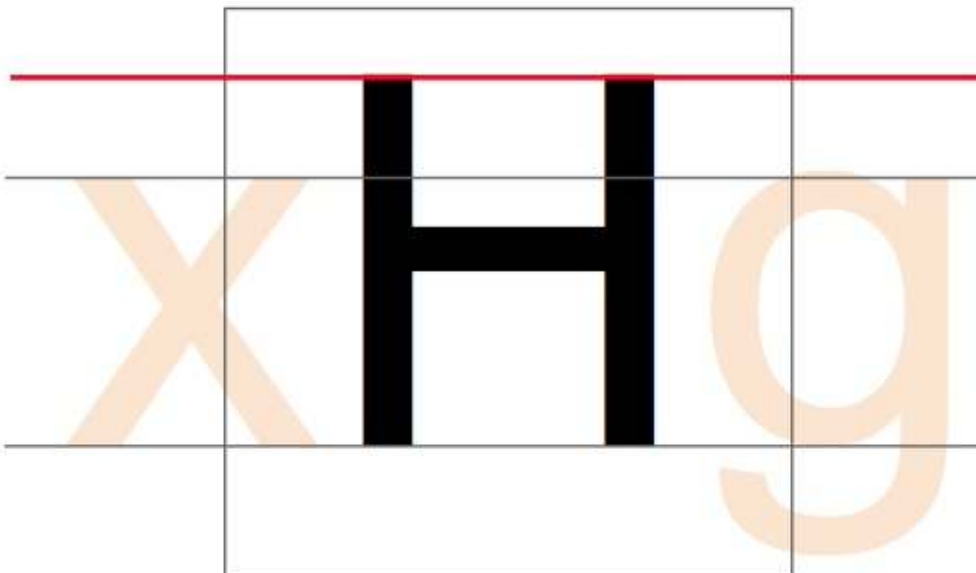
En sachant que votre police utilise un UPM de 1000, 1024 ou 2048, vous devez configurer le dessin de vos glyphes pour vous assurer que tous les aspects de votre police de caractères correspondent bien à ce carré UPM.

La taille du carré em peut être définie à partir de *Élément > Infos fonte...* puis cliquez sur l'onglet Général et vous verrez le paramètre *EM*, lequel sera réparti entre les hauteurs *Ascendante* et *Descendante*, les hauteurs respectives au-dessus et au-dessous de la ligne de base.

La ligne de base:

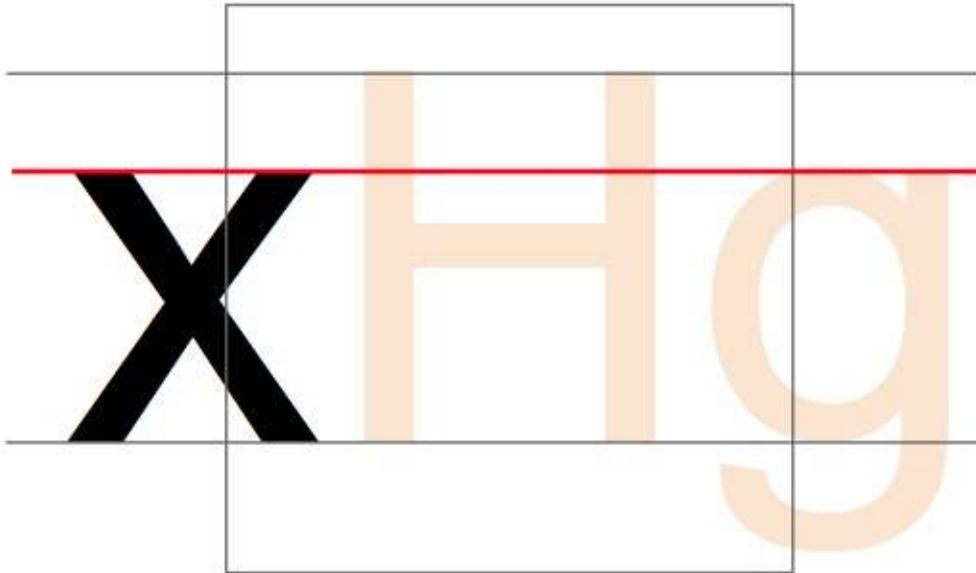


La hauteur de capitale:



La hauteur d'x:





Plus tard, lors de la conception de votre police, vous devrez définir les *Blue values* qui servent pour les contours PostScript et également pour l'autohinter de FontForge – indépendamment des contours sur lesquels vous travaillez. Vous trouverez le paramètre dans *Élément > Infos fonte...*, sur l'onglet *Dictionnaire PS privé*. FontForge peut d'abord deviner les valeurs en fonction de vos contours, mais vous devrez les modifier vous-même pour régler les valeurs trop basses/trop hautes — mais nous sommes quelques chapitres en avance sur ce concept (voir "[Création de 'o' et de 'n'](#)"); apprenons d'abord FontForge et ses fonctionnalités de dessin.

# Installation de Fontforge

FontForge est un logiciel libre, vous pouvez donc télécharger, partager et installer des copies sans aucune restriction sur l'utilisation — l'utilisation commerciale ou personnelle est encouragée. Il s'agit d'une application gérée par la communauté et n'importe qui peut contribuer au code source.

FontForge est disponible en paquets faciles à installer pour les systèmes d'exploitation Windows, Mac OS X et GNU+Linux.

## Installation de FontForge sur Windows

Un [guide d'installation \(en anglais\)](#) est disponible pour les versions officielles de Windows, préparées par Jeremy Tan.

## Installation de FontForge sur Mac OS X

Un [guide d'installation \(en anglais\)](#) est disponible pour les versions officielles Mac, préparé par le Dr Ben Martin.

## Installation sur GNU/Linux

La méthode la plus simple pour obtenir FontForge sur votre machine Linux est d'utiliser le dépôt de paquets de votre distribution Linux.

### Debian ou Ubuntu

Le paquet FontForge inclus dans Ubuntu 14.04 par défaut date de 2012, il est donc préférable d'installer le paquet plus à jour à partir de la [PPA \(Personal Package Archive\)](#) de FontForge.

Vérifiez que le script d'aide `add-apt-repository` est installé:

```
sudo apt-get install software-properties-common;
```

Ajoutez la PPA FontForge (qui ajoutera également la clé d'authentification):

```
sudo add-apt-repository ppa:fontforge/fontforge;
```

Mettez à jour la liste des logiciels pour inclure les paquets de la PPA:

```
sudo apt-get update;
```

Installez FontForge:

```
sudo apt-get install fontforge;
```

### Fedora

Pour installer FontForge sur votre machine Fedora Linux, exécutez la commande yum suivante en tant qu'utilisateur root. Environn 10 Mo devront être téléchargé pour terminer.

```
yum install fontforge;
```

Si vous n'avez pas compilé de logiciel sur votre machine Fedora, après avoir installé gcc, automake, autoconf et autres, vous pourriez obtenir une erreur lors de l'exécution de autogen.sh avec libtoolize. Si c'est le cas, vous devrez peut-être installer le paquet libtool-ltdl-devel sur Fedora ou un package de développement similaire sur une autre distribution GNU+Linux.

Après l'installation de yum, vous devriez être en mesure d'exécuter FontForge à partir de votre menu ou directement à partir de la console ou gnome-terminal en émettant la commande `fontforge`.

## Compiliez votre propre version à partir de Github

GitHub est un service d'hébergement de code source où tout le monde peut contribuer au développement d'un logiciel. Il stocke l'état de développement actuel de l'application. Si vous souhaitez accéder à une fonctionnalité qui n'est pas encore disponible dans les paquets officiels, vous pouvez créer votre propre version à partir de Github.

Des instructions complètes (en anglais) sont disponibles à l'adresse <https://github.com/fontforge/fontforge/blob/master/INSTALL-git.md>

## Débogage de FontForge

Voir la section [Débogage](#) pour plus d'informations.

# Configuration de Fontforge

FontForge peut être peaufiné de différentes façons. Voici quelques trucs et astuces pour le faire. Vous avez beaucoup d'options pour optimiser FontForge pour votre plateforme et flux de travail.

S'il vous plaît, [dites-nous](#) si vous avez des conseils que vous souhaitez partager.

## Commençons par le commencement

Lors de toute modification de configuration, assurez-vous de suivre ceci:

1. Quittez FontForge (et X11)
2. Effectuez les modifications
3. Démarrez FontForge et testez vos modifications

## Windows

Actuellement, nous n'avons rien de spécifique à la distribution Windows. Si vous pensez à quelque chose, [dites-le nous](#).

## GNU+Linux

Actuellement, nous n'avons rien de spécifique à aucune distribution GNU+Linux. Si vous pensez à quelque chose, [dites-le nous](#).

## Mac OS X

Pour ouvrir l'emplacement d'un long fichier ou dossier:

1. Copiez le chemin
2. `⌘ Tab` pour passer au Finder
3. `⌘⇧G` pour ouvrir le menu Aller → Aller au dossier...
4. `⌘V` pour coller dans le chemin
5. `⌘⇧A` pour ouvrir une nouvelle fenêtre Finder à cet emplacement

## Raccourcis clavier

De nombreux dialogues et éléments de menu ont une lettre qui est soulignée. Vous pouvez y accéder immédiatement en appuyant sur `Ctrl-Alt` et sur cette touche. Par exemple, si une boîte de dialogue vous demande si c'est OK, appuyez sur `Ctrl + Alt + o`.

Si vous n'utilisez pas un clavier anglais américain, vous pouvez trouver que certains des raccourcis clavier sont idiots. Ou, vous voudrez peut-être simplement les personnaliser pour être ce que vous vous en attendez. Pour modifier ces touches, ouvrez et modifiez le fichier texte `default`, situé ici:

```
/Applications/FontForge.app/Contents/Resources/opt/local/share/fontforge/hotkeys/default
```

Lorsque vous installerez la prochaine version, tous les fichiers à l'intérieur de `/Applications/FontForge.app` seront écrasés, donc sauvegardez une copie de votre fichier `default` ailleurs.

## Taille de l'interface utilisateur

Si l'interface utilisateur semble trop grande ou trop petite, elle peut être mise à l'échelle pour s'adapter à votre ordinateur. Ouvrez et modifiez le fichier texte `resources`, situé ici:

```
/Applications/FontForge.app/Contents/Resources/opt/local/share/fontforge/pixmaps/resources
```

Ajoutez la ligne `Gdraw.ScreenWidthCentimeters: 34` si la largeur de votre écran est de 34 cm. Essayez différentes valeurs jusqu'à ce que vous soyez satisfait.

## Signets

Dans la boîte de dialogue Fichier, il y a une option `Ajouter un signet à cet emplacement`, mais l'option `Supprimer le signet...` ne fonctionne pas # 2054. Vous pouvez modifier la liste manuellement dans la section `FCBookmarks` du fichier `prefs` situé à

```
~/.config/fontforge/prefs
```

Réinitialisez vos signets en ouvrant le terminal et en collant le texte suivant:

```
sed -i bak -e 's/^FCBookmarks.*FCBookmarks:  ~\/Library\/Fonts\/;\/Library\/Fonts\/;\/System\/Library\/Fonts\/g'
~/.config/fontforge/prefs;
```

Appuyez ensuite sur Entrée pour exécuter cette commande. Si vous ne voyez pas d'erreurs, elle devrait avoir fonctionné correctement.

## Souris à 3 boutons

FontForge peut utiliser les clics d'une souris à trois boutons pour certaines fonctions supplémentaires. Si vous n'avez pas de souris à trois boutons, vous pouvez l'émuler en l'activant dans les préférences X11/Xquartz, dans l'option `Input` de la section `Emulate three button mouse`

## Changez l'icône de X11/XQuartz pour l'icône de FF

Si vous utilisez principalement X11 pour FontForge, vous pouvez modifier son icône. Copiez et collez le texte suivant dans le terminal et suivez les instructions

```
sudo cp -f /Applications/FontForge.app/Contents/Resources/FontForge.icns /Applications/Utilities/XQuartz.app/Contents
/Resources/X11.icns | sudo touch /Applications/Utilities/XQuartz.app
```

## Gestion de fenêtre

FontForge n'est pas une application Mac native, donc le traitement des fenêtres peut être légèrement décalé, en particulier sur les systèmes à double moniteur. Pour reprendre le contrôle du positionnement des fenêtres, utilisez l'utilitaire libre [ShiftIt](#) pour attribuer des raccourcis clavier définissant les positions de fenêtres.

# Aperçu de l'interface utilisateur

FontForge a la même interface sur les systèmes d'exploitation Windows, Mac OS et GNU/Linux.

Voici une brève introduction aux caractéristiques essentielles donné par Dave Crossland à l'atelier Crafting Type, un atelier de conception de police sans but lucratif qui soutient le projet FontForge:



Il y a 4 fenêtres principales:

## 1. La fenêtre de fonte

Cette fenêtre apparaît lorsque vous exécutez FontForge. Elle montre un tableau des glyphes dans la police.

## 2. La fenêtre de glyphe

Double-cliquez sur un glyphe dans la fenêtre de fonte pour l'ouvrir dans la fenêtre de glyphe. C'est là que vous pouvez dessiner et modifier des glyphes, soit un à la fois avec une interface graphique, soit côte à côte avec un fichier texte de mots à parcourir préparé à l'avance.

## 3. La fenêtre de métriques

Allez à Métriques ou Fenêtres et sélectionnez Ouvrir fenêtre de métriques. Ou sélectionnez certains glyphes dans la fenêtre de glyphe et appuyez sur «Ctrl-K». Vous pouvez également glisser et déposer des glyphes depuis la fenêtre de glyphe vers la fenêtre de métriques.

C'est là que vous pouvez ajuster l'espacement et le crénage de votre police. Vous pouvez également parcourir un fichier texte contenant une liste de mots préparés.

## 4. La fenêtre des infos sur la fonte

Allez à Éléments, Infos fonte... Ici vous trouverez toutes les *métadonnées* au sujet de la police.

## 5. L'environnement de composition, pour les tests

En plus d'un éditeur de polices, vous aurez besoin d'un système de composition pour tester vos polices en cours d'utilisation. C'est l'un des **grands secrets** des designers de polices professionnels : la majeure partie du travail d'un projet de conception de police est faite en dehors de l'éditeur de polices, dans des documents de test créés avec un système de composition. Les principaux systèmes sont:

- Web (Firefox, Chromium, etc)
- Outils PAO (Scribus, Inkscape, etc.)
- Traitements de texte (Libre Office, AbiWord, Calligra Words, etc)
- Systèmes de composition de documents (LaTeX, ConTeXt, SILE, ShoeBot, Even, etc)

## Utilisation des outils de dessin de FontForge

Concevoir une police dans FontForge impliquera l'utilisation d'un certain nombre d'outils et d'utilitaires, en commençant par un ensemble d'outils de dessin qui peuvent être familiers aux utilisateurs ayant de l'expérience avec des graphiques vectoriels – bien qu'il y ait des différences notables. Nous chercherons d'abord à comprendre comment les courbes de Bézier fonctionnent, avant de regarder les outils de dessin FontForge eux-mêmes.

### Comprendre les courbes de Bézier

Les "courbes de Bézier" sont une représentation mathématique particulière utilisée pour produire des courbes lisses numériquement. Généralement, les ordres *cubiques*- et *quadratiques*- de ces courbes sont employés — bien que FontForge supporte également les courbes de *Spiro* qui sont une représentation alternative pour le concepteur.

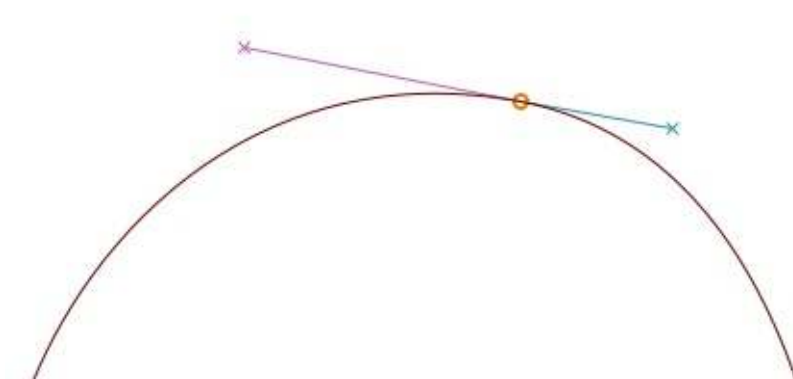
Dans ce chapitre, nous ne discuterons que des chemins *cubiques*, car c'est ce qui est généralement utilisé lors du dessin des glyphes. Les chemins *Spiro* seront discutés dans le chapitre suivant et les courbes *quadratiques* ne se retrouvent que dans les polices TrueType et sont rarement utilisées dans le dessin – elles sont plutôt générées au moment de la construction.

Un chemin typique de Bézier est composée d'une ancre, avec deux poignées qui marquent la direction globale — la longueur de chaque poignée détermine la longueur de la courbe de chaque côté –voir ci-dessous.

#### Différents types de points

##### Points curvilignes (Représentés sous forme de points ronds)

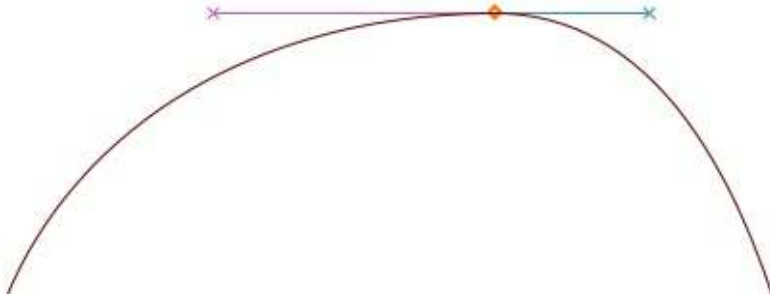
Les *points curvilignes* ont deux poignées, chacune étant liée à l'autre de sorte que la ligne entre elles reste toujours droite, afin de produire une courbe lisse de chaque côté.



##### Points curvilignes H/V (Représentés sous la forme de points en forme de losange)

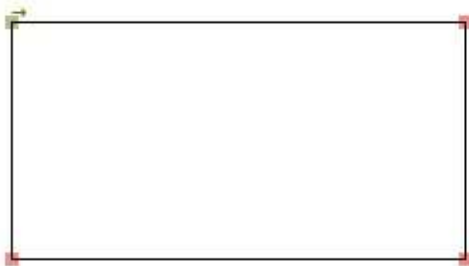
Les *points curvilignes H/V* ('horizontal/vertical') sont une variante de points curvilignes qui s'accrochent à l'axe horizontal ou vertical – un outil essentiel pour obtenir des formes de Bézier correctes (plus de détails dans la section suivante).





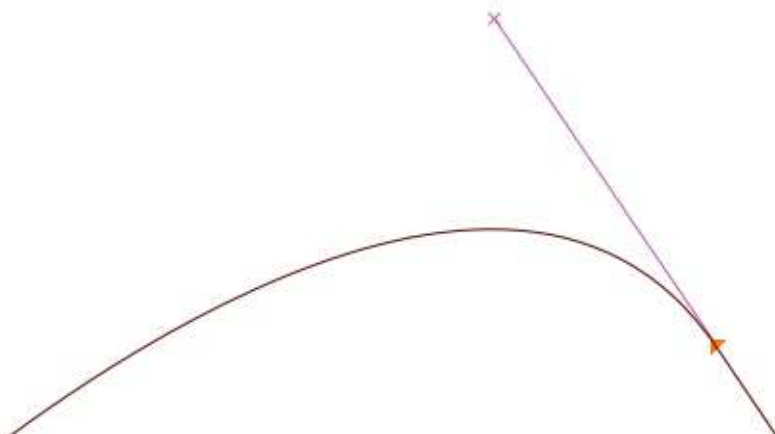
### Coins ou points angulaires (représentés par des points carrés)

Les *coins* peuvent avoir 0, 1 ou 2 poignées de Bézier. La position de chaque poignée est indépendante des autres, ce qui la rend apte à des discontinuités dans le contour. Sans poignées, ces points produiront des lignes droites.



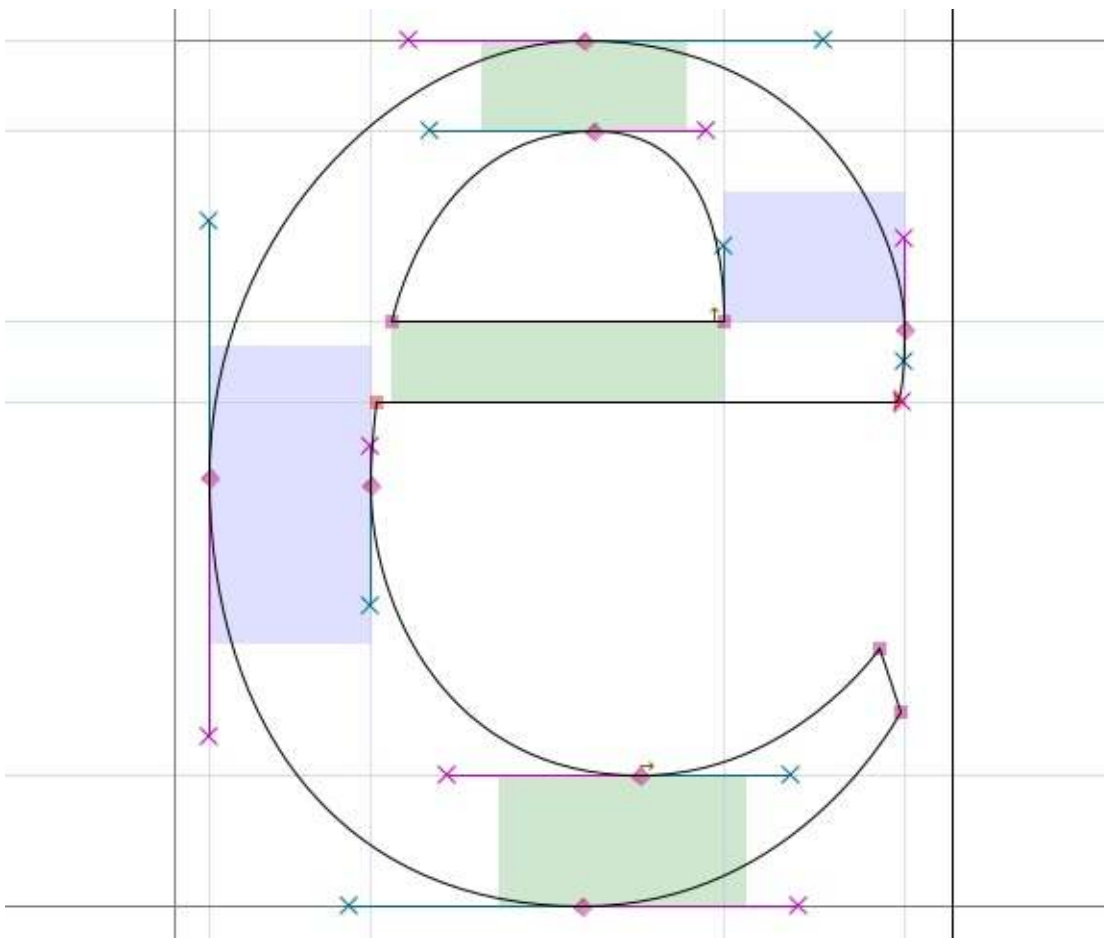
### Points tangents (représentés par des points triangulaires ou des 'pointes de flèches')

Si vous voulez commencer à partir d'une ligne droite, puis commencer à arrondir en douceur, vous voudrez utiliser des *points tangents*. Une *tangente* laisse une ligne droite d'un côté, tandis que la poignée de Bézier de l'autre côté est sa direction – ceci assure une transition continue entre la ligne et la courbe.

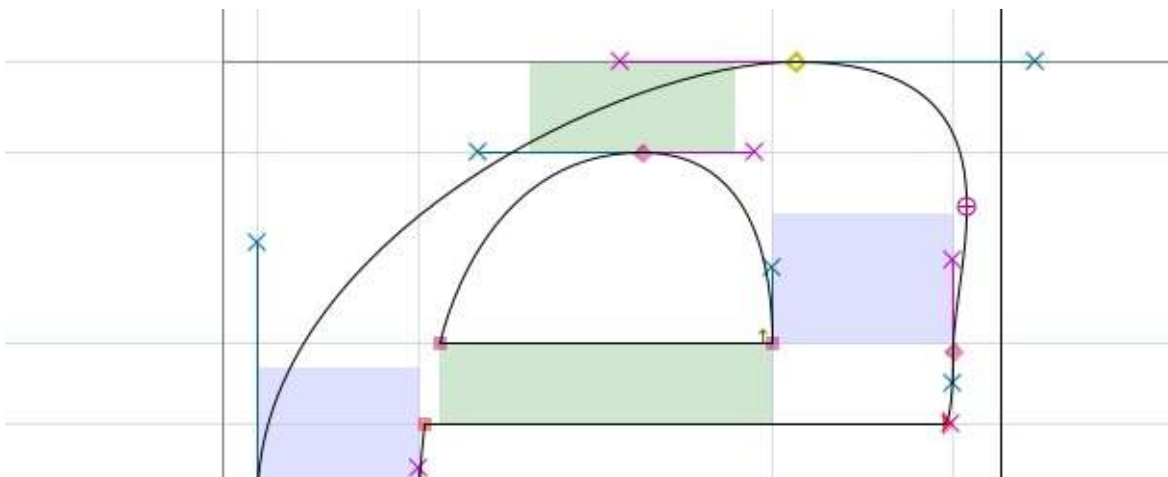


## Faire les bons choix

Afin de produire des courbes appropriées – avec points de contrôle minimaux et une rasterisation facilitée, les ancrés doivent toujours être placés aux **extrema de la courbe**, et sauf dans les endroits où vous avez des angles dans la forme de vos lettres, la ligne qui détermine le chemin doit être **horizontale ou verticale**.



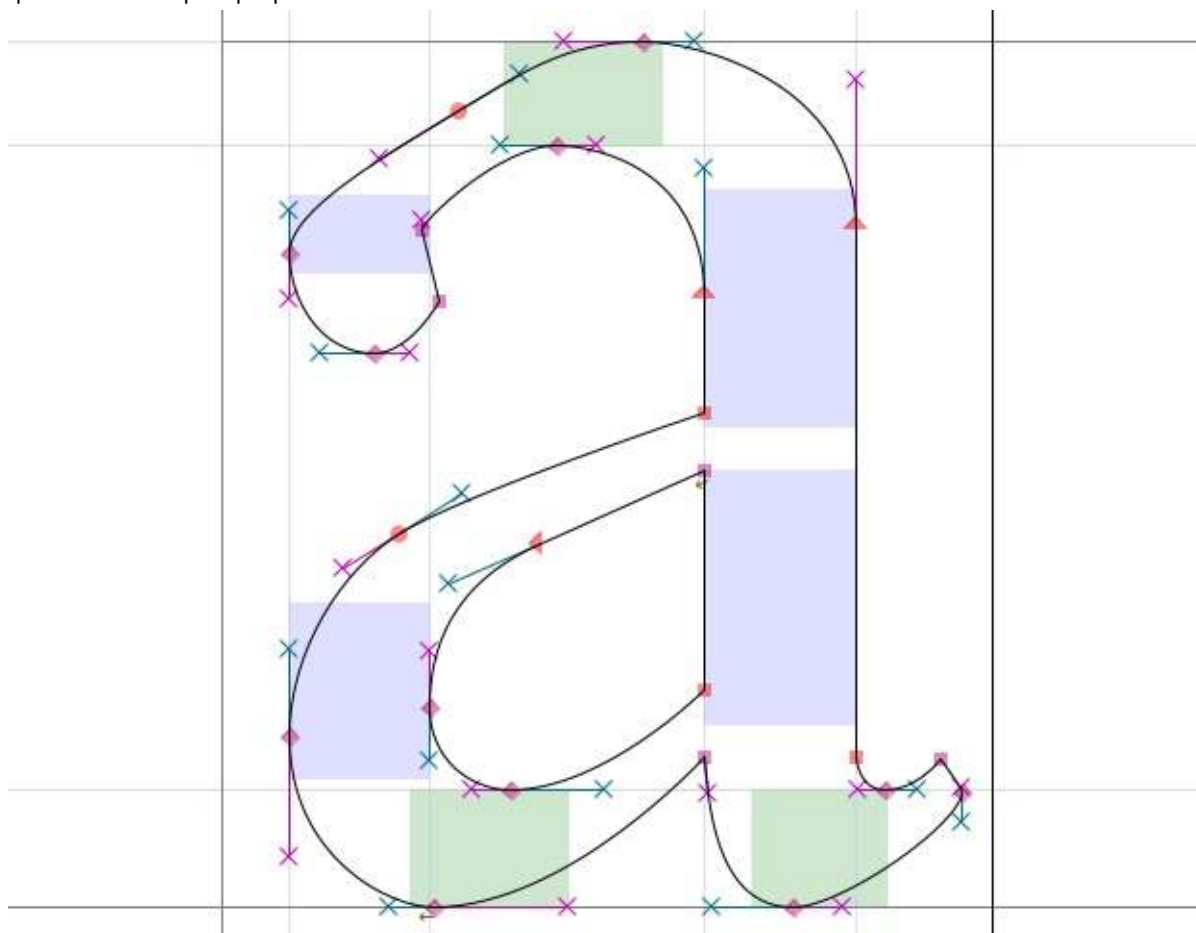
**Remarque:** Si vos points de contrôle ne sont pas placés aux extrema, FontForge signalera les extrema réels avec une icône de mire:



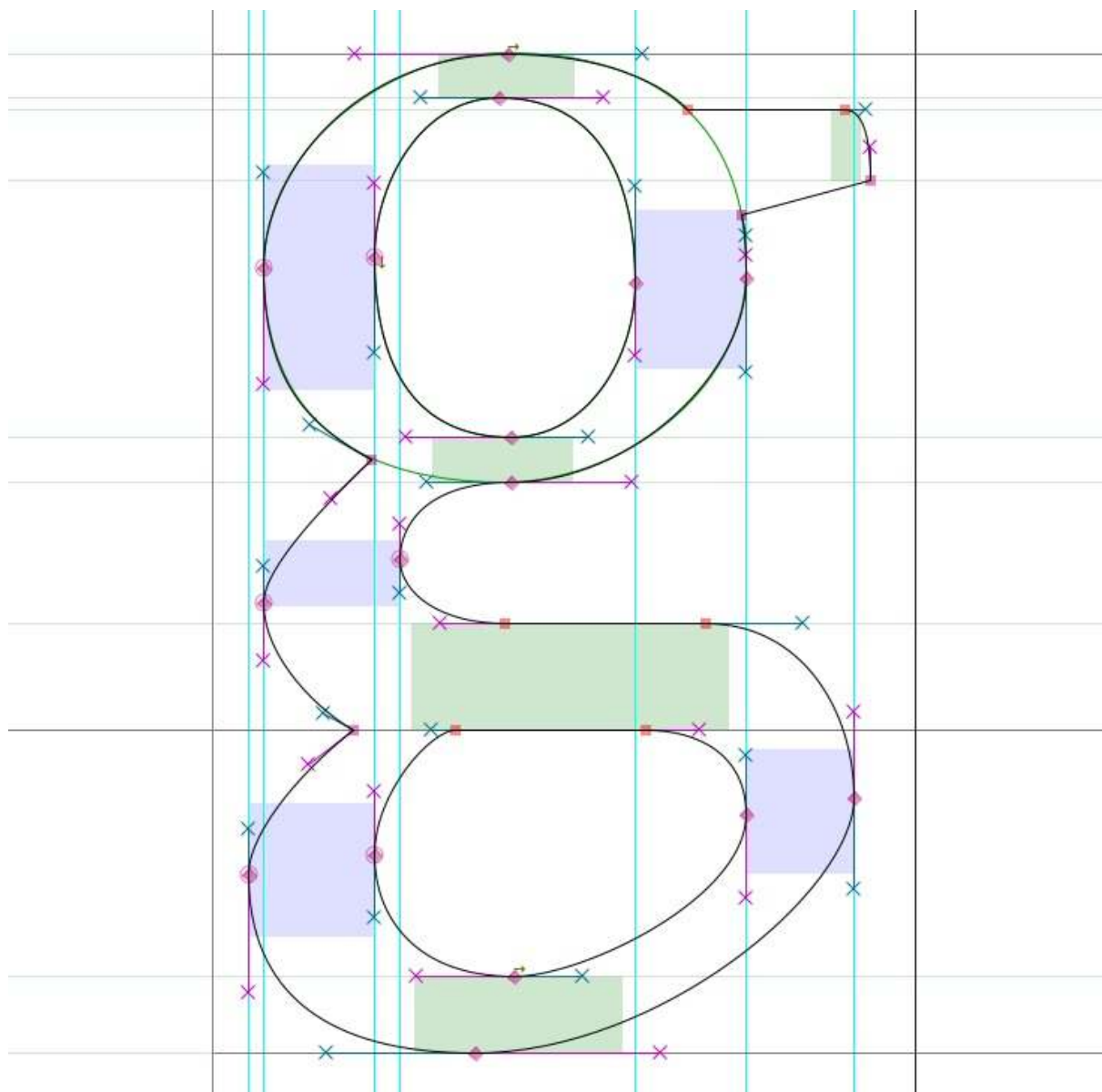
Vous pouvez alors corriger ce problème en copiant votre contour actuel vers un autre calque, puis déplacer les points de contrôle autour de sorte que le contour soit bien tracé – sinon l'outil de validation de FontForge ajoutera automatiquement le point aux extrema. Vous pouvez fusionner l'ancre mal placée de ce point avec *bouton droit* > *Fusionner* . Plus d'informations à ce sujet seront données dans le chapitre sur la validation .

Pour élaborer, il y a deux cas où vous devrez renoncer à des chemins de Bézier horizontaux et verticaux:

- Si vous voulez changer la pente globale de votre courbe, comme avec la partie supérieure gauche du 'a' ci-dessous qui est maintenu presque plat:



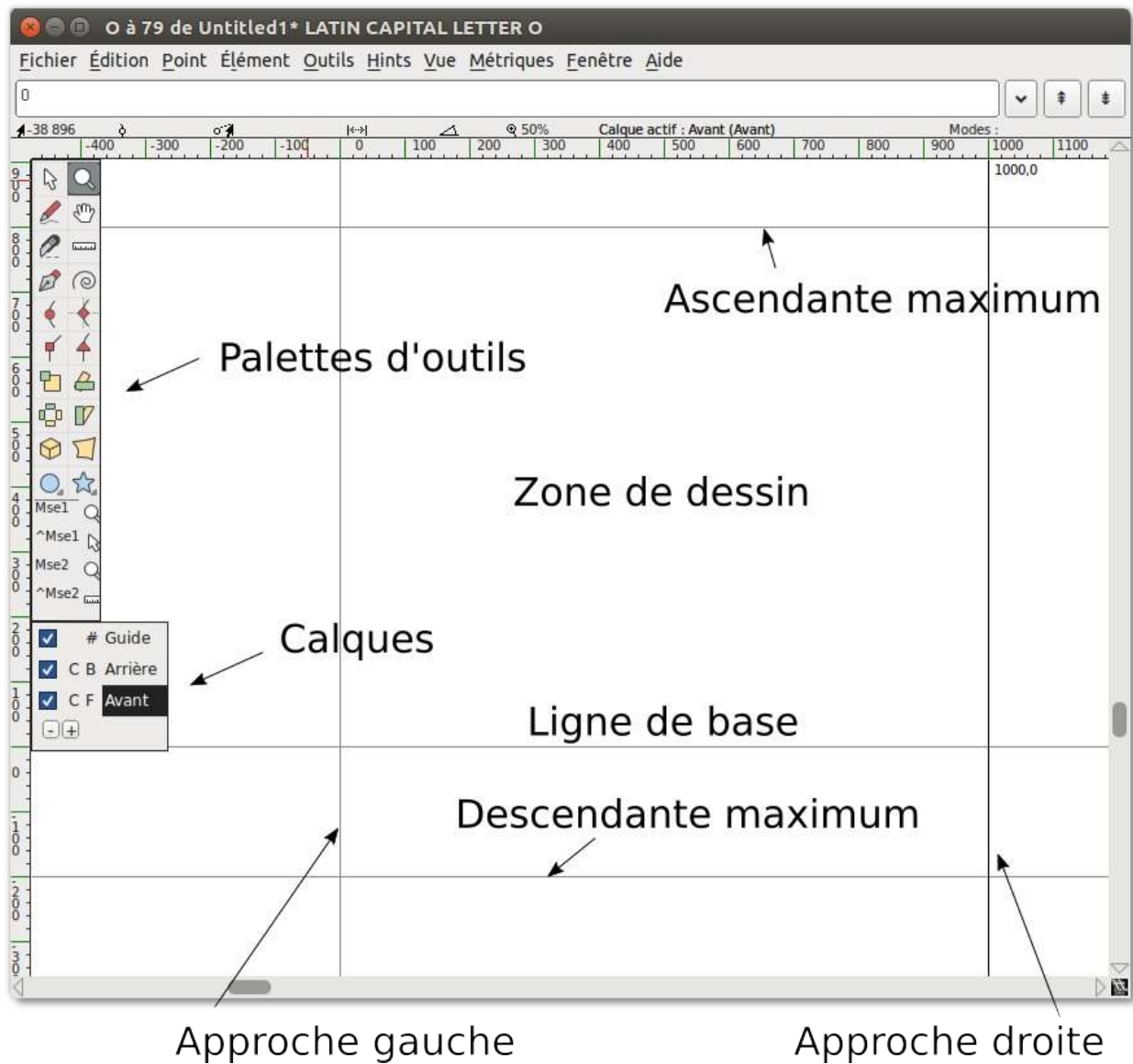
- Si vous voulez placer des angles dans vos lettres, comme dans la partie inférieure gauche du 'g' ci-dessous – c'est généralement où vous voulez utiliser un *coin* (à part pour dessiner des lignes):



**Remarque:** Comme vous pouvez le constater, lors de l'ajout d'angle avec un *coin*, la direction de chaque poignée doit être tangente à la courbe où elle arrive.

## Maîtriser les outils de dessin de FontForge

Dans la fenêtre principale, double-cliquez sur l'une des cases de glyphe pour lancer la fenêtre de glyphe.

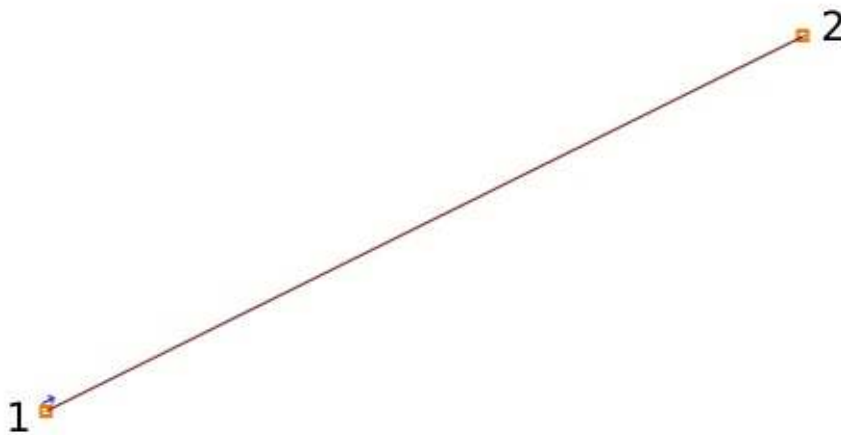


**Remarque:** Les nombres sur le haut où se croisent les axes x et y, de gauche à droite indiquent:

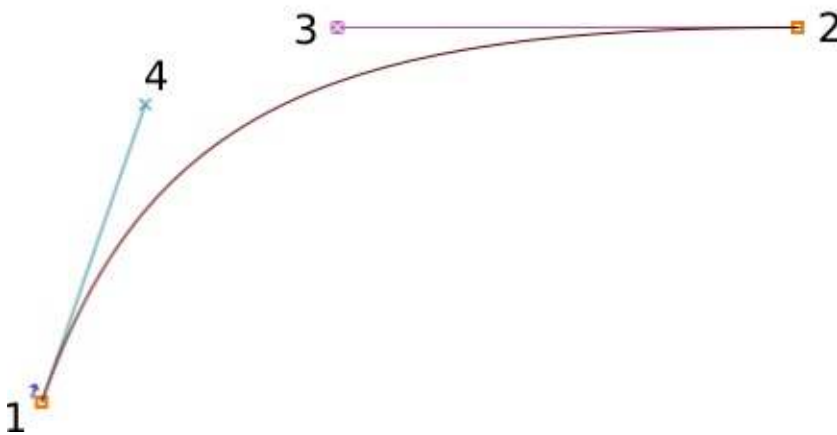
- L'emplacement actuel (x, y) de votre curseur sur le canevas
- L'emplacement du dernier point sélectionné
- La position relative du curseur sur le point sélectionné
- La distance entre votre curseur et le point sélectionné
- L'angle du point sélectionné au curseur (par rapport à la ligne de base)
- Le niveau de grossissement actuel, suivi du nom du calque actif.

**Mise en garde:** Parfois, il semble que FontForge ne répond pas lorsque vous êtes à l'intérieur de la fenêtre de glyphe. Il se peut qu'il y ait une boîte de dialogue ouverte cachée derrière – il suffit de la déplacer et de fermer la boîte de dialogue.

Une *ligne* se compose de 2 points.



Une courbe ou *spline* se compose de 4 points: 2 points à l'extrémité de la spline et 2 'poignées', qui décrivent la pente de la spline à ces points d'extrémité.



## Copier, coller, couper et supprimer des points, des splines et des lignes

Comme pour la plupart des logiciels de dessin, FontForge vous permet de copier, couper, coller ou supprimer n'importe quel point, ligne ou spline. Ces commandes sont disponibles dans le menu Édition, ou en utilisant les touches typiques de votre système d'exploitation (également affichées à côté de chaque commande dans le menu).

## Familiarisez-vous avec les outils de dessin

Maintenant que vous connaissez votre chemin autour de la zone de travail, il est temps de se familiariser avec les outils.

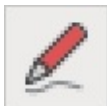
### Pointage et Zoom



Le pointage et le zoom se comportent de la même manière que les outils équivalents dans d'autres applications. Le pointeur est un outil de sélection, utilisé pour sélectionner des points, des chemins et d'autres objets sur le canevas. L'outil Zoom vous permet de zoomer (Z) facilement; pour effectuer un zoom arrière: allez dans le menu Vue et sélectionnez *Réduire* (X) ou *Adapter*.

Notez que vous pouvez également passer momentanément à l'outil de pointeur tout en utilisant un autre en maintenant enfoncée la touche Control (Ctrl).

## Dessin à main-levée



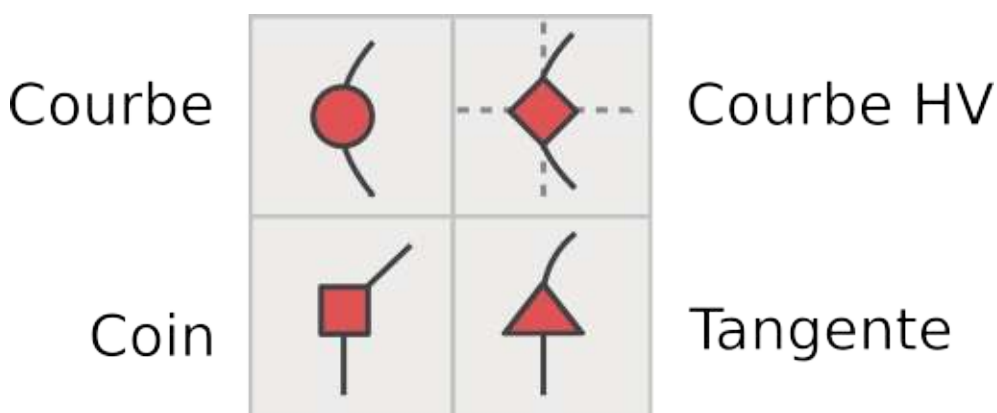
L'outil de dessin à main-levée vous permet d'esquisser des tracés irréguliers.

Sur la zone de dessin, cliquez et maintenez, puis déplacez-vous pour dessiner. Retournez à l'outil pointeur, et vous pouvez sélectionner des points sur le chemin que vous avez dessiné.

Lorsque vous sélectionnez un des points sur le chemin, il se transforme en un cercle jaune. Si le point sélectionné est sur une courbe, il affichera ses points de contrôle avec une poignée magenta et une poignée cyan. Vous pouvez saisir n'importe quelle poignée et la faire glisser pour modifier la forme de la courbe.

## Les outils point

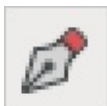
Expliquons maintenant l'utilisation des outils point.



Pour ajouter un point à un chemin, sélectionnez tout d'abord un de ces outils, puis cliquez sur le chemin et donnez-lui une petite poussée. Vous obtiendrez un nouveau point sur la ligne.

L'outil point courbe permet d'ajouter un point dans un segment incurvé. L'outil point courbe HV limite les nouveaux points afin qu'ils aient des points de contrôle horizontaux ou verticaux – Ceci est important pour la mise en place de points aux extrema. L'outil point de coin vous permet de faire un virage serré dans le chemin. L'outil point de tangente vous permet de passer d'un segment droit à un segment courbe le long du chemin.

## La plume (courbes interactives)



L'outil plume vous permet d'ajouter un point sur la courbe et de faire glisser ses points de contrôle.

## Spiro



La sélection de l'outil Spiro vous place en mode de dessin Spiro. Ce mode vous permet de dessiner des courbes qui refondent lorsque vous repositionnez les nœuds. Certaines personnes préfèrent cette méthode à l'approche standard (connue sous le nom d'édition Bézier), mais si vous êtes habitué à l'édition Bézier vous trouverez peut-être que cela mène à des choses inattendues.

## Le couteau



L'outil couteau vous permet de couper les splines en deux. Cela est pratique si vous avez dessiné une forme et que vous avez seulement besoin d'une partie de celle-ci.

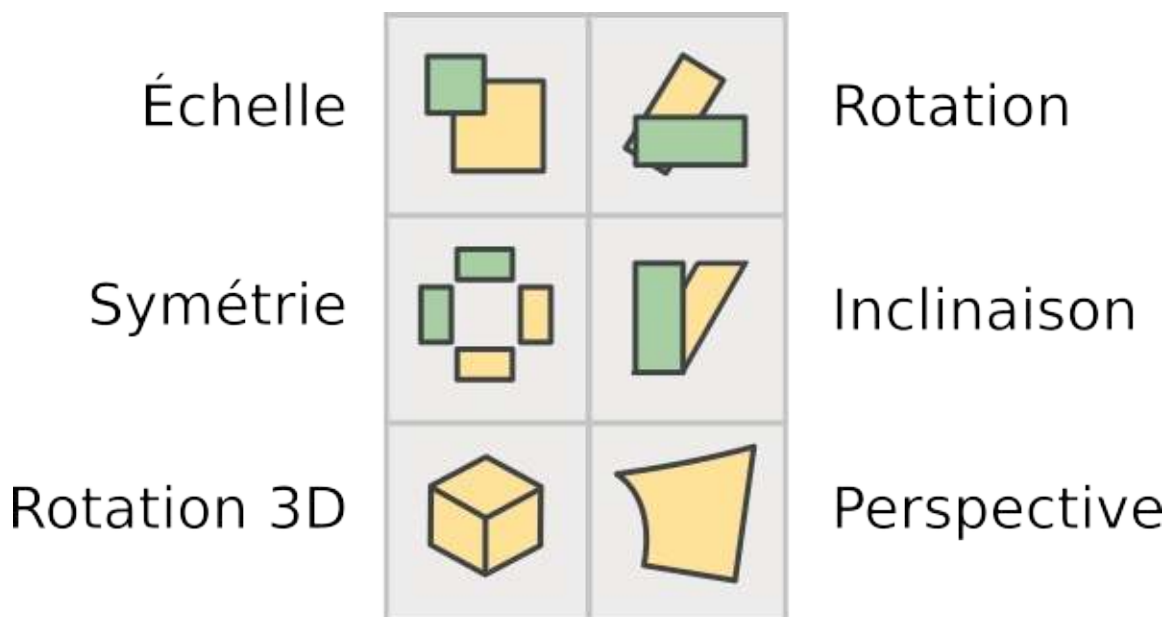
## La règle



L'outil règle vous donne des informations de mesure et de coordonnées. Lorsque vous l'utilisez, il affiche une info-bulle flottante à côté du curseur. Si vous placez votre curseur sur un point, l'info-bulle vous donne des informations de mesure et de coordonnées encore plus détaillées. Si vous la placez à côté d'une spline, il vous donne des informations sur la courbure et le rayon. Le plus utile, si vous cliquez et faites glisser l'outil règle, vous verrez la distance de déplacement du curseur, plus chaque intersection que vous avez tendu à travers.

## Les outils de transformation

Il y a six outils de transformations:



**Remarque:** Pour plusieurs outils de transformation, si vous double-cliquez sur l'outil, vous pouvez saisir des valeurs numériques.

L'outil échelle vous permet de redimensionner un objet à la main. Maintenir la touche Maj enfoncée vous permet d'augmenter l'échelle d'un objet tout en le conservant ses proportions.

L'outil rotation vous permet de tourner librement un objet. Il tourne l'objet sélectionné autour de la position où vous avez cliqué.

L'outil symétrie vous permet de retourner une sélection à l'horizontale ou à la verticale. Le point auquel vous cliquez sur le canevas est le point d'origine de la transformation.

**Remarque:** Après avoir inversé un point, vous voudrez probablement appliquer Élément > *Corriger direction*.



L'outil inclinaison vous permet d'incliner horizontalement la sélection dans le sens des aiguilles d'une montre ou dans le sens inverse des aiguilles d'une montre.

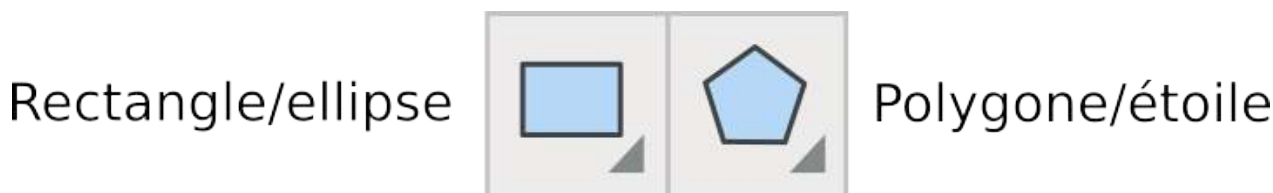
L'outil de rotation 3D vous permet de faire pivoter un objet dans la troisième dimension et de projeter le résultat sur le plan x-y.

L'outil Perspective vous donne une autre façon de déformer une forme de façon non linéaire.

**Remarque:** Il n'y a pas d'option numérique pour la transformation en perspective.

## Les outils rectangle/ellipse et polygone/étoile

Ces outils vous permettent de dessiner des formes géométriques de base, ce qui est plus rapide que de construire ces formes à partir de segments de ligne distincts.



En cliquant sur la zone de chevron sur ces outils vous aurez la possibilité de passer à un autre outil. Si vous double-cliquez sur l'un des outils, vous affichez les options relatives à la forme.

Options de rectangle: style de coin et boîte englobante (coin ou centre).

Options d'ellipse: boîte englobante ou autour du centre.

Options de polygone: nombre de sommets.

Options d'étoiles: nombre de pointes d'étoiles et profondeur des pointes en pourcentage. Plus le pourcentage est élevé, plus les branches de l'étoile sont longues.

## Mse1 et Mse2



Sous la palette d'outils, vous pouvez afficher l'outil actuel et les opérations disponibles pour les deux boutons de la souris:

- Bouton gauche (Mse1)
- Bouton gauche + Ctrl (^ Mse1)
- Bouton de la molette de la souris (Mse2)
- Bouton de la molette de la souris + Ctrl (^ Mse2)

De cette façon, vous pouvez utiliser quelques outils différents sans avoir à cliquer à plusieurs reprises sur la palette d'outils.

**Attention:** Il semble que la fonctionnalité Mse ne fonctionne pas correctement.

## Calques

Le canevas de FontForge a trois calques par défaut: le calque guide, le calque d'arrière-plan et le calque d'avant-plan. Les calques guide sont utilisées pour insérer des guides (tels que des guides de hauteur d'x ou de hauteur de capitale). Les calques d'avant-plan et les calques d'arrière-plan sont tous les deux utilisés pour le dessin, mais seule le calque de premier plan le plus élevé sera rendu dans votre police finale.



Une case à cocher indique si un calque est visible, et vous pouvez cliquer dessus pour changer la visibilité. Le C (ou Q) indique si vous utilisez des courbes cubiques ou quadratiques.

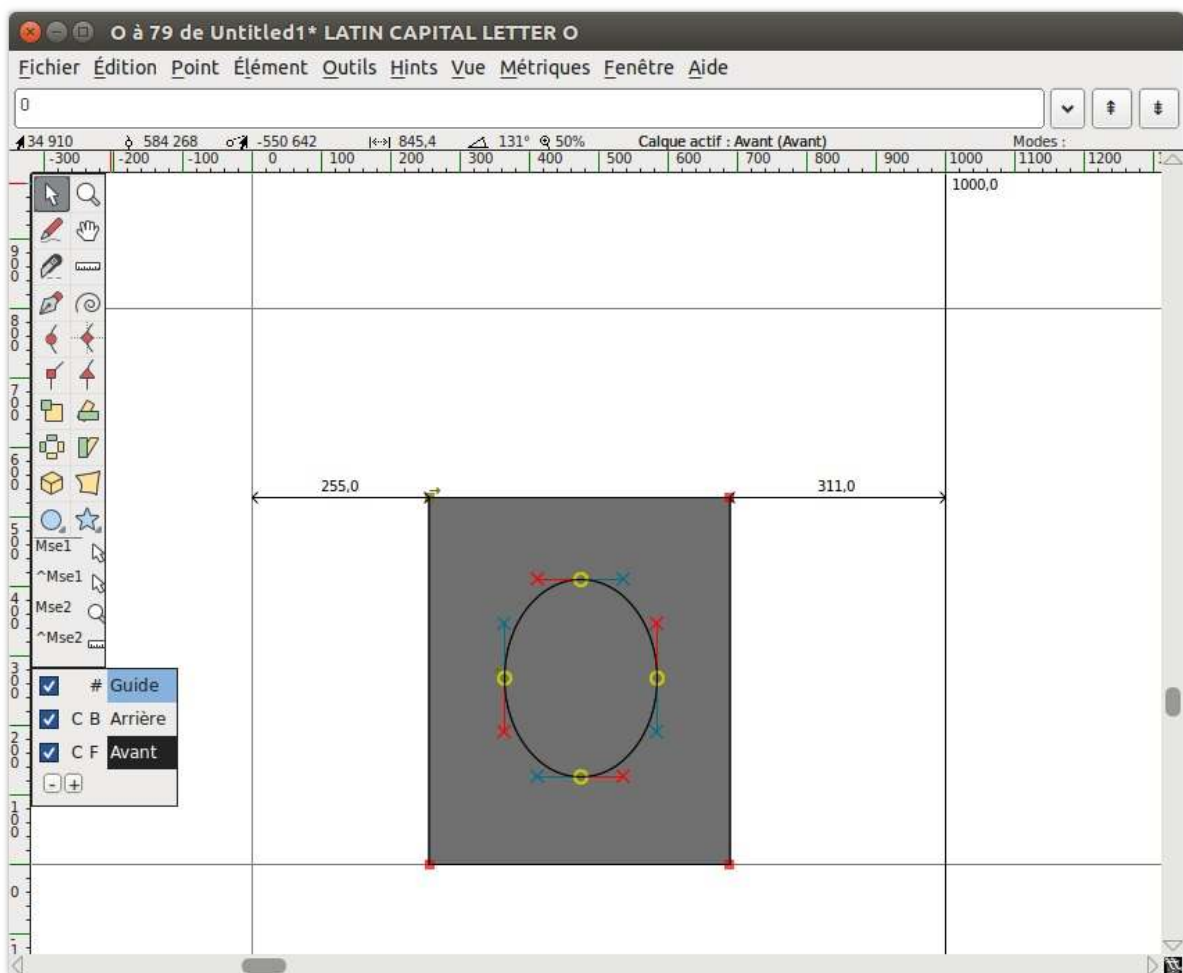
Le #, B ou F indique si le type de calque est guide, d'arrière-plan ou d'avant-plan, ce qui est important si vous ajoutez des calques supplémentaires. Vous pouvez créer et supprimer des calques supplémentaires à l'aide des boutons plus (+) ou moins (-) dans cette section de la palette d'outils. Le type de calque et le type de courbe peuvent également être contrôlés en cliquant avec le bouton droit (une fois que vous avez ajouté des calques supplémentaires).

## Dessin de base

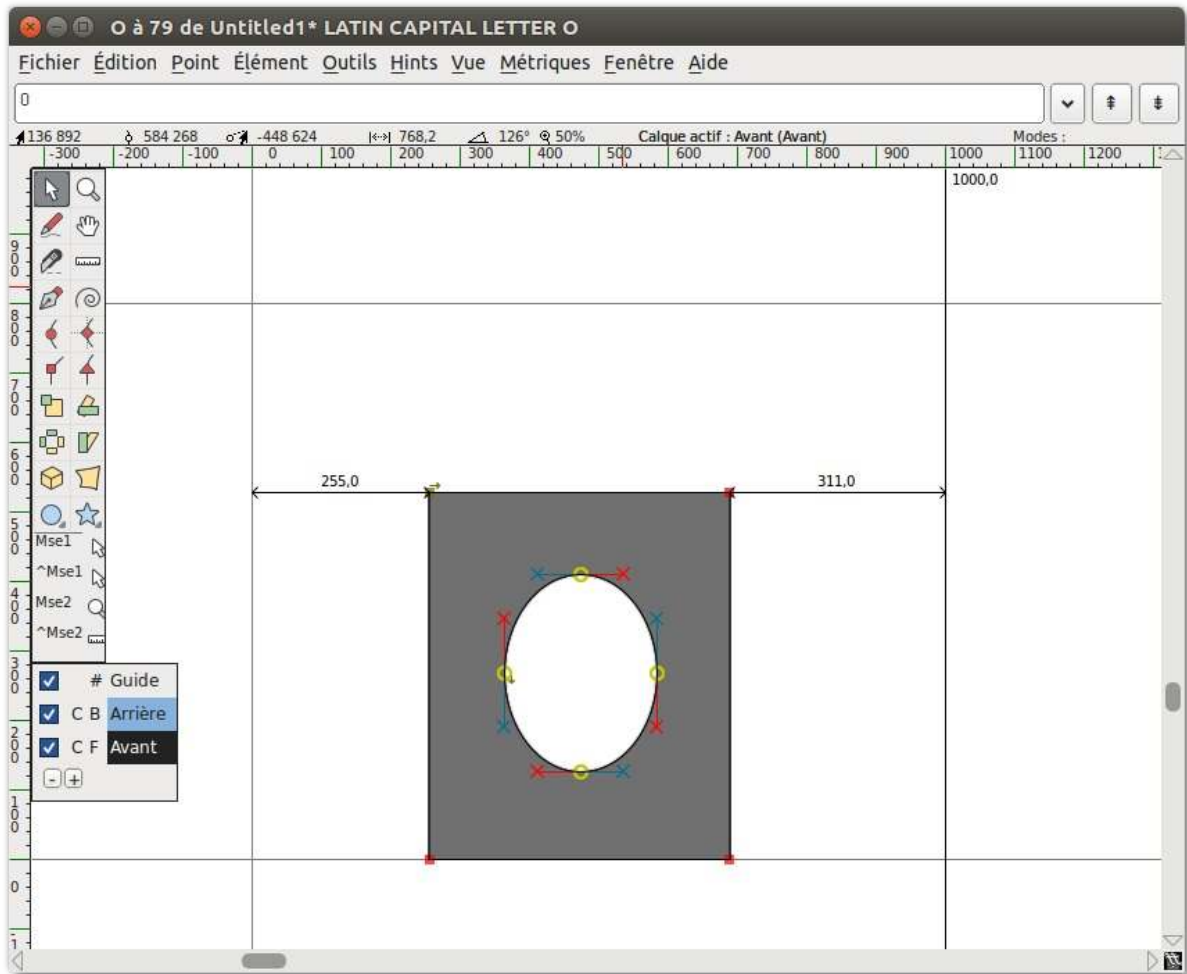
Maintenant, examinons quelques workflows de dessin de base, dont vous aurez souvent besoin.

### Couper une forme dans une autre

1. Commencez par utiliser l'outil rectangle pour dessiner un rectangle dans la zone de dessin de la fenêtre de glyphe.
2. Ensuite, utilisez l'outil ellipse pour dessiner une ellipse dans le rectangle que vous venez de dessiner.

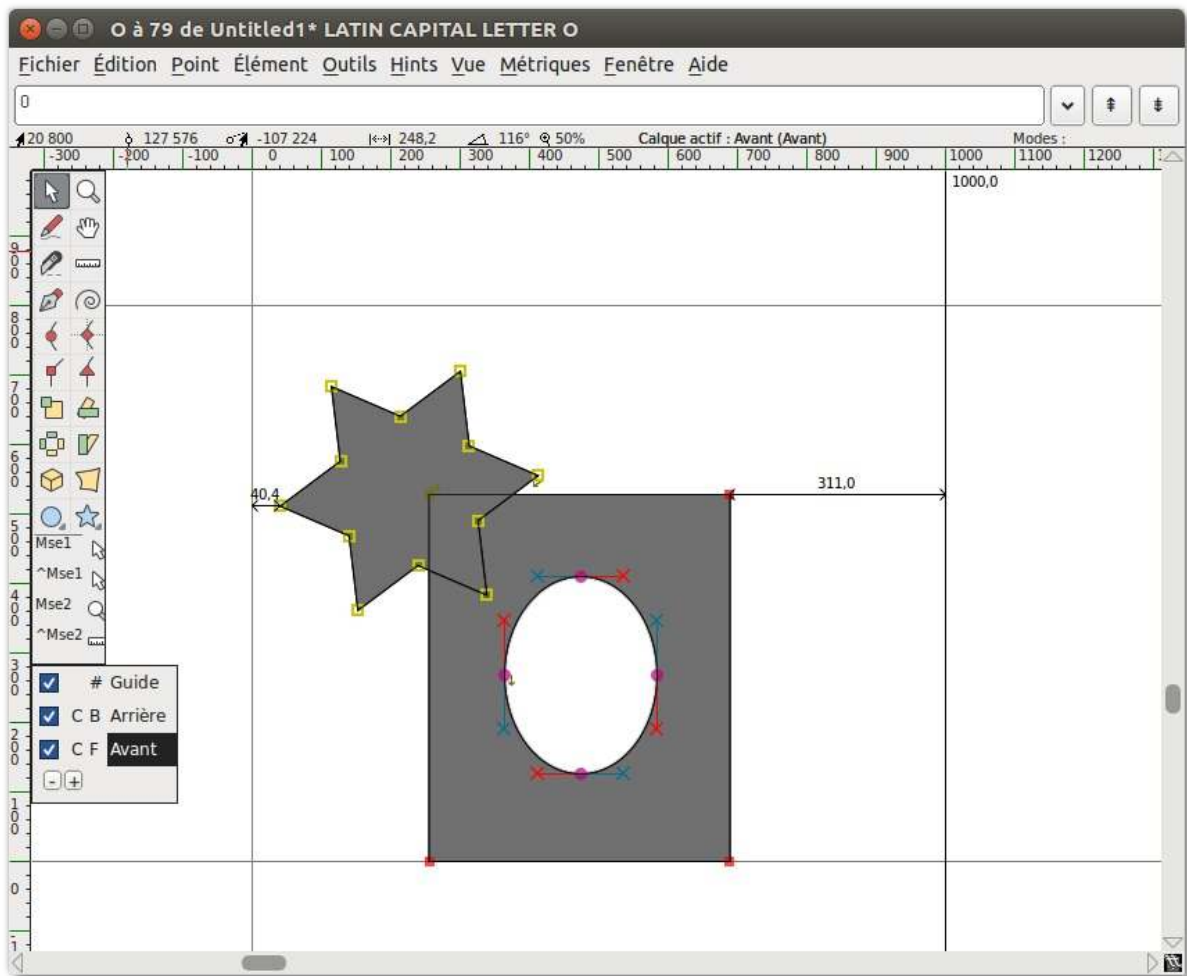


3. Allez dans le menu Élément et choisissez *Corriger direction*. Vous verrez que les deux formes se sont fusionnées, et que vous avez essentiellement percé un trou dans le centre du rectangle.

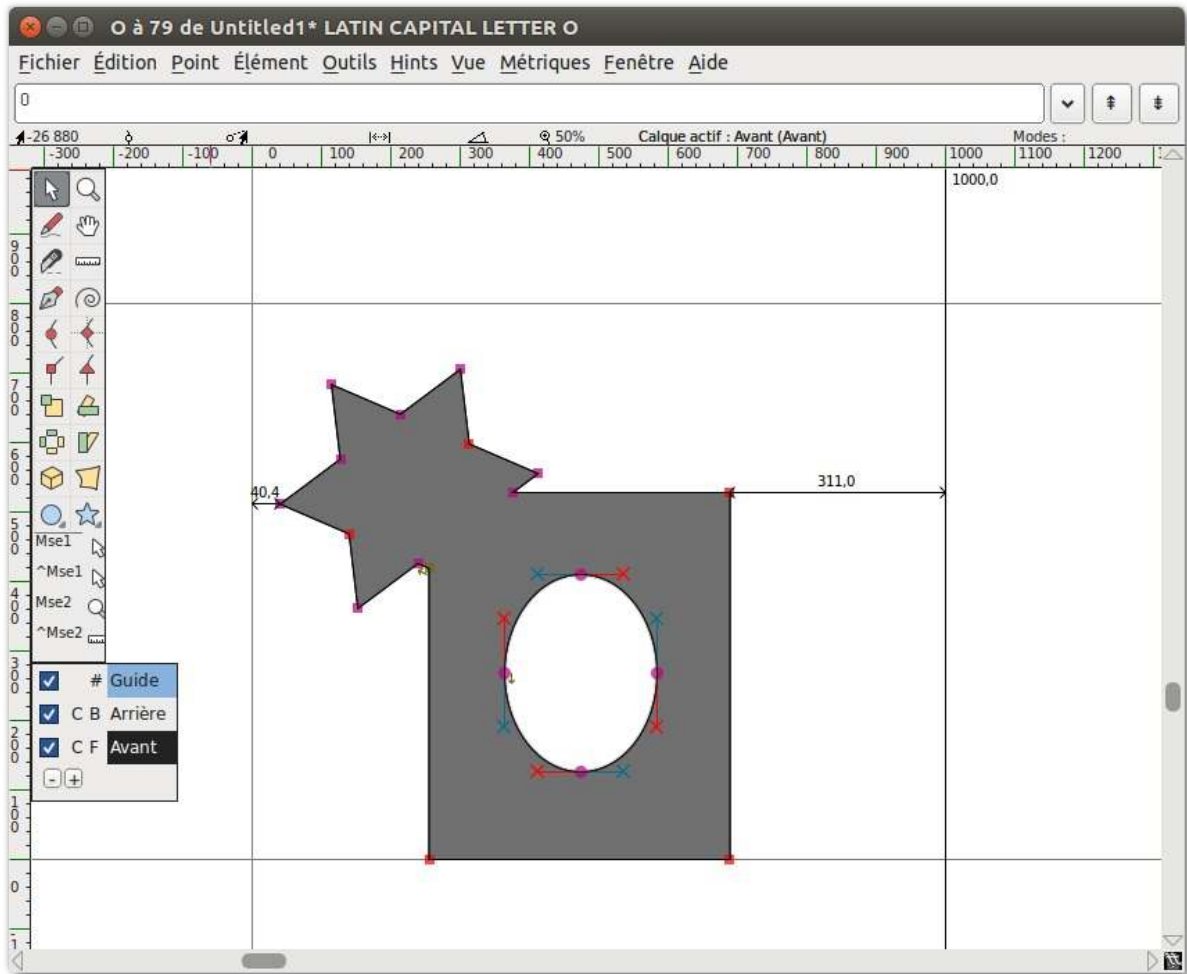


## Supprimer le chevauchement

1. Ajoutez une étoile qui chevauche le coin du rectangle.

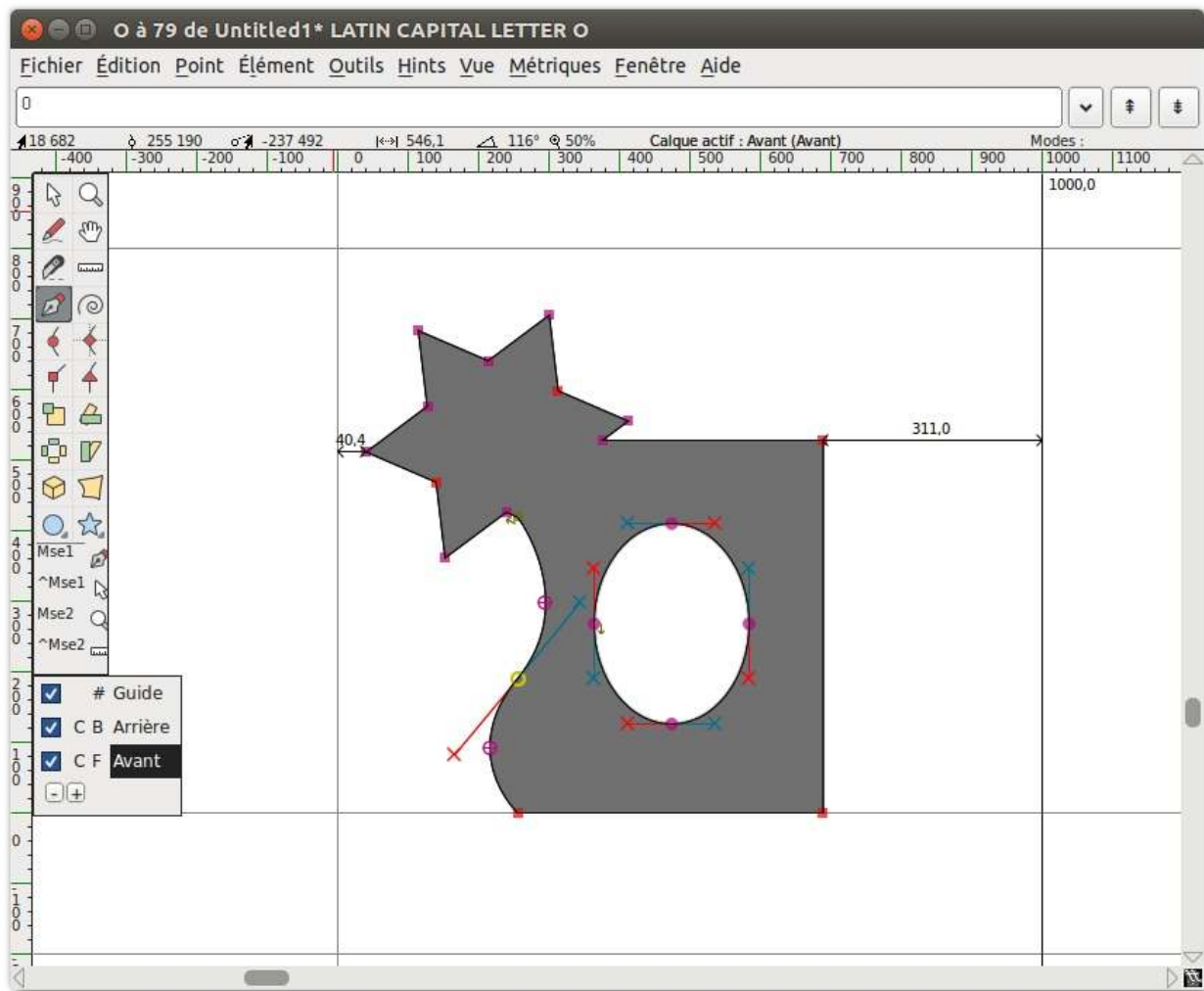


2. Sélectionnez l'étoile et la forme précédente. Il vous suffit de sélectionner un point de chaque forme chevauchante, mais il est acceptable de sélectionner des points supplémentaires.
3. Allez à Élément > Recouvrement > *Enlever les recouvrements*. Vous verrez que vos deux formes sont devenues une seule.



## Ajouter un point

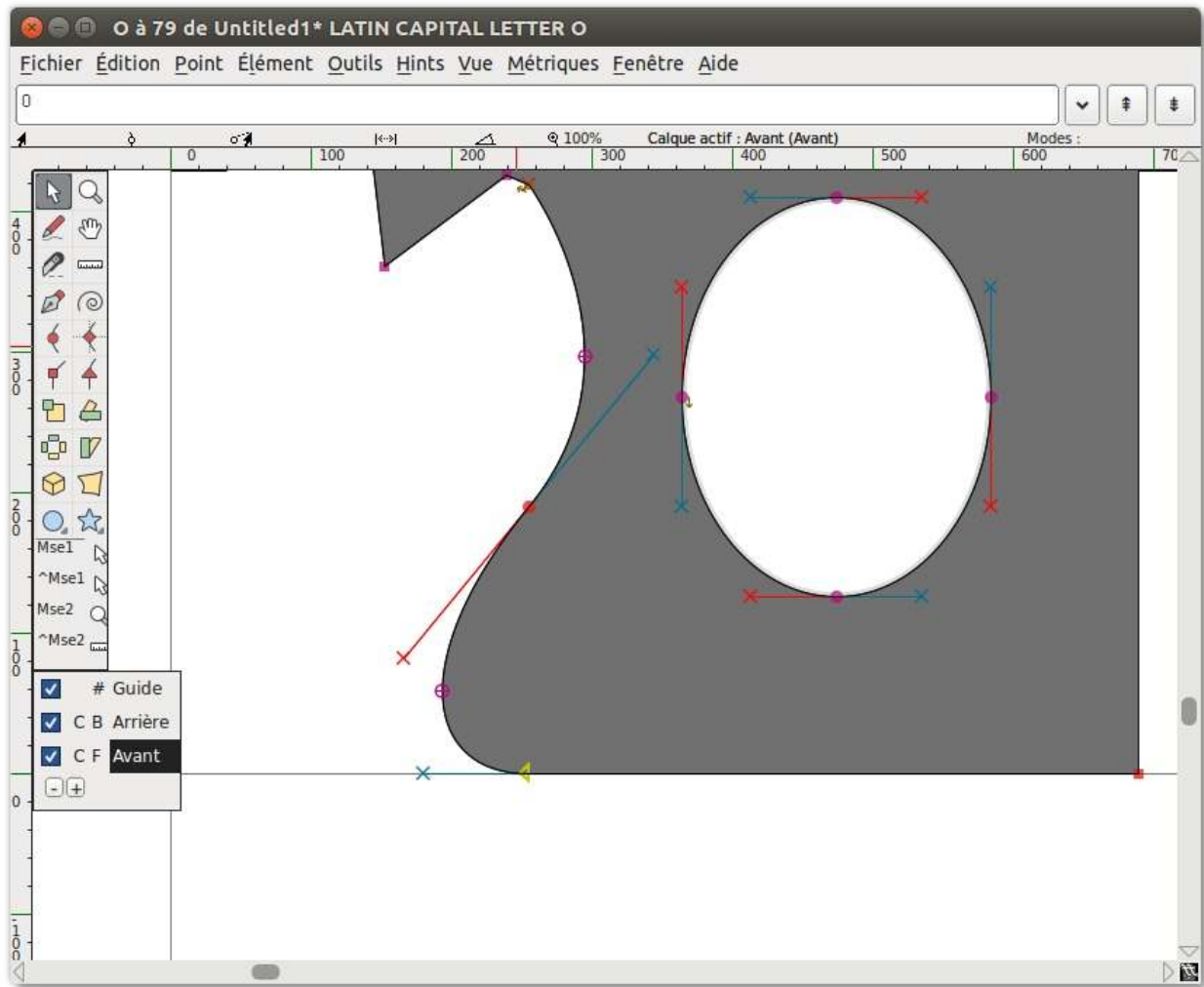
À l'aide de l'outil plume, cliquez et maintenez au milieu d'un segment de ligne, puis faites glisser la souris pour changer la forme.



## Points tangents

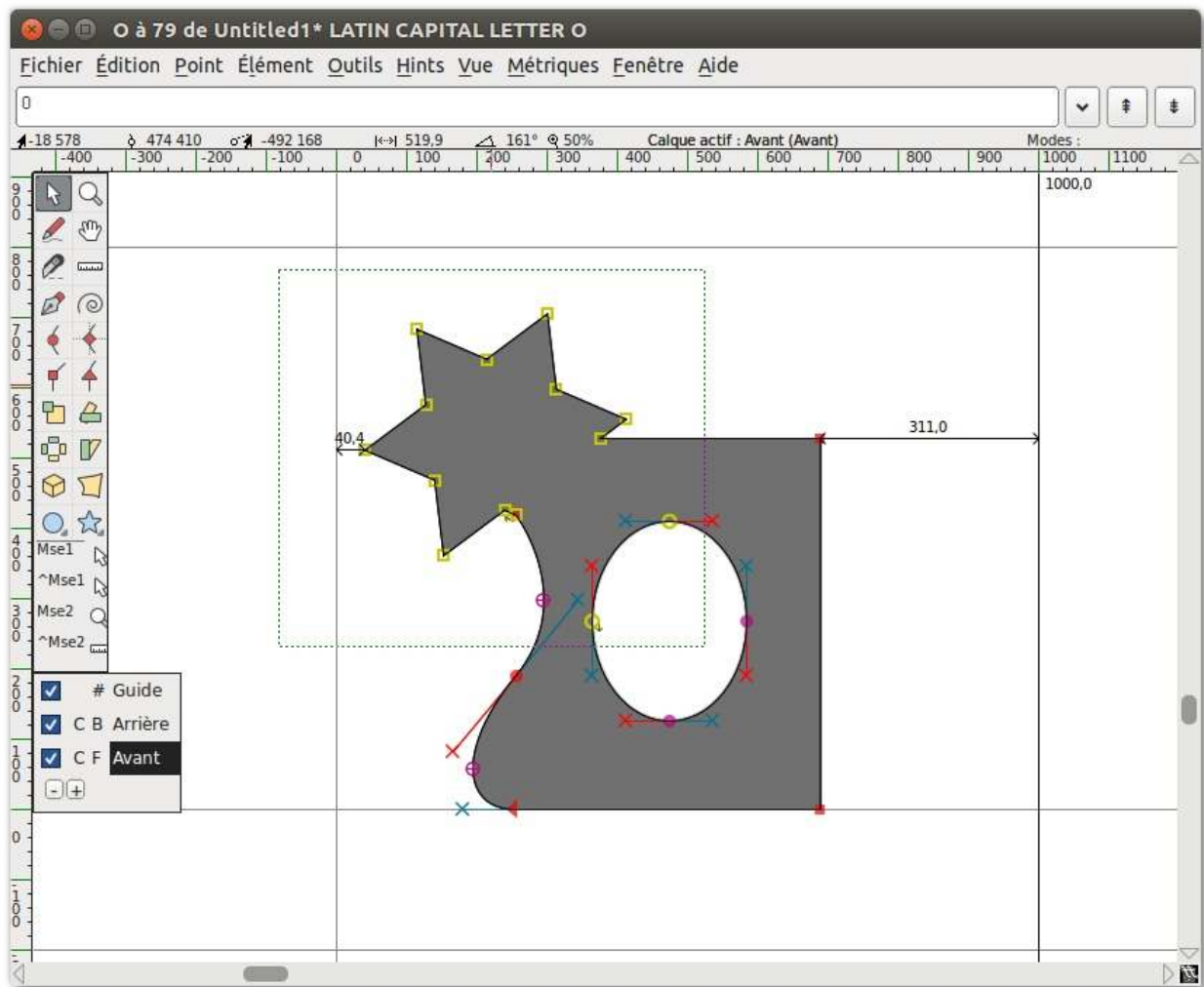
Sélectionnez le coin inférieur gauche de votre nouvelle forme (l'intersection de la courbe et de la droite). Dans le menu Point, vous verrez que *point angulaire* est coché. Sélectionnez *Convertir en point tangent*. Cela change le nœud carré en triangle, mais c'est tout ce que ça fait jusqu'à ce que vous fassiez l'étape suivante: Extension des points de contrôle.

Pour ce faire, choisissez Élément > Infos..., qui ouvre la fenêtre Infos points. Dans l'onglet Emplacement de cette fenêtre, accédez au champ PdeC suivant et définissez une grande distance, par exemple 75. Cliquez sur OK. Vous verrez que la courbe se poursuit maintenant doucement dans la droite.



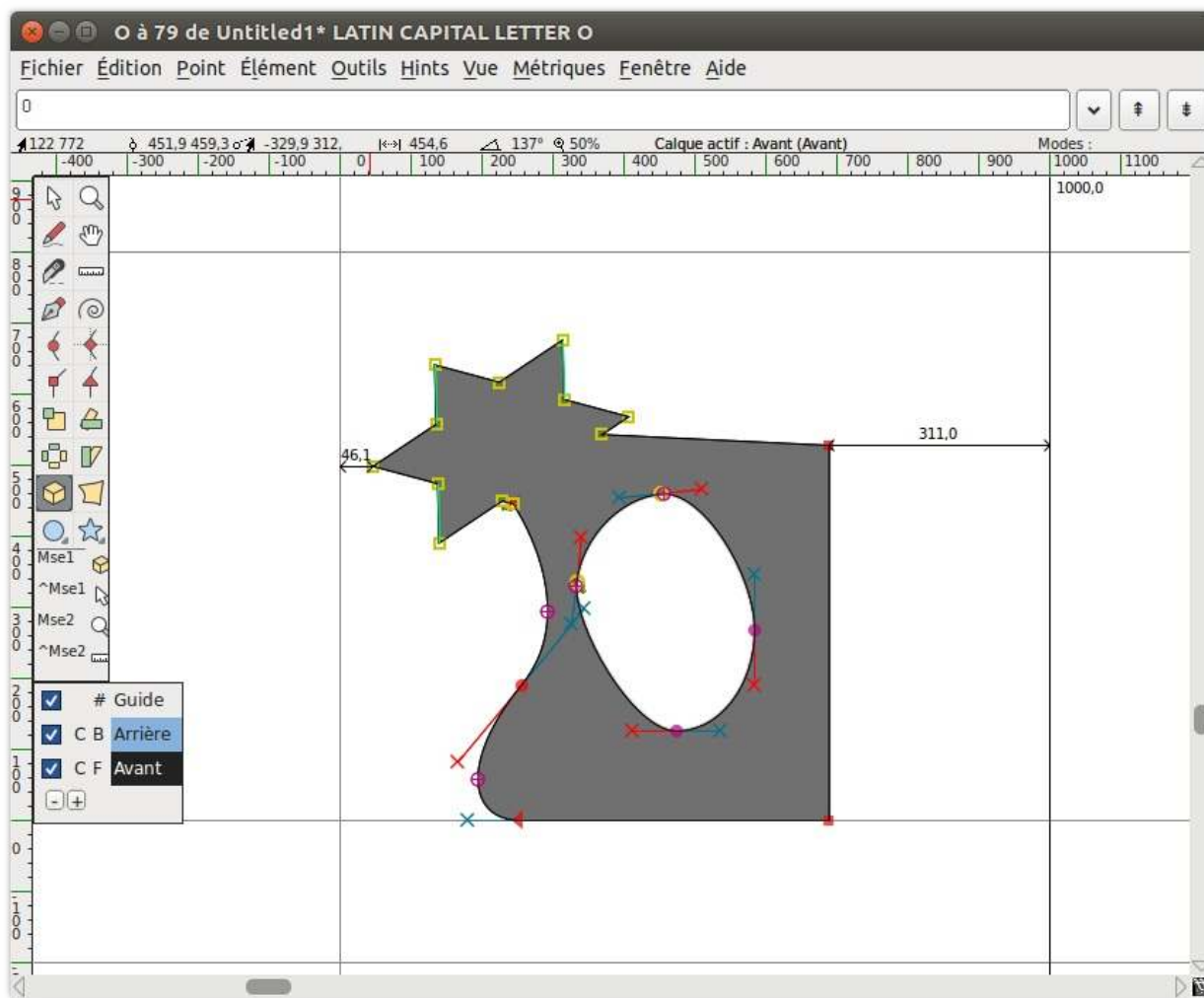
## Transformation

Maintenant, sélectionnez environ un quart de la forme — l'étoile et une partie de l'ellipse au milieu.



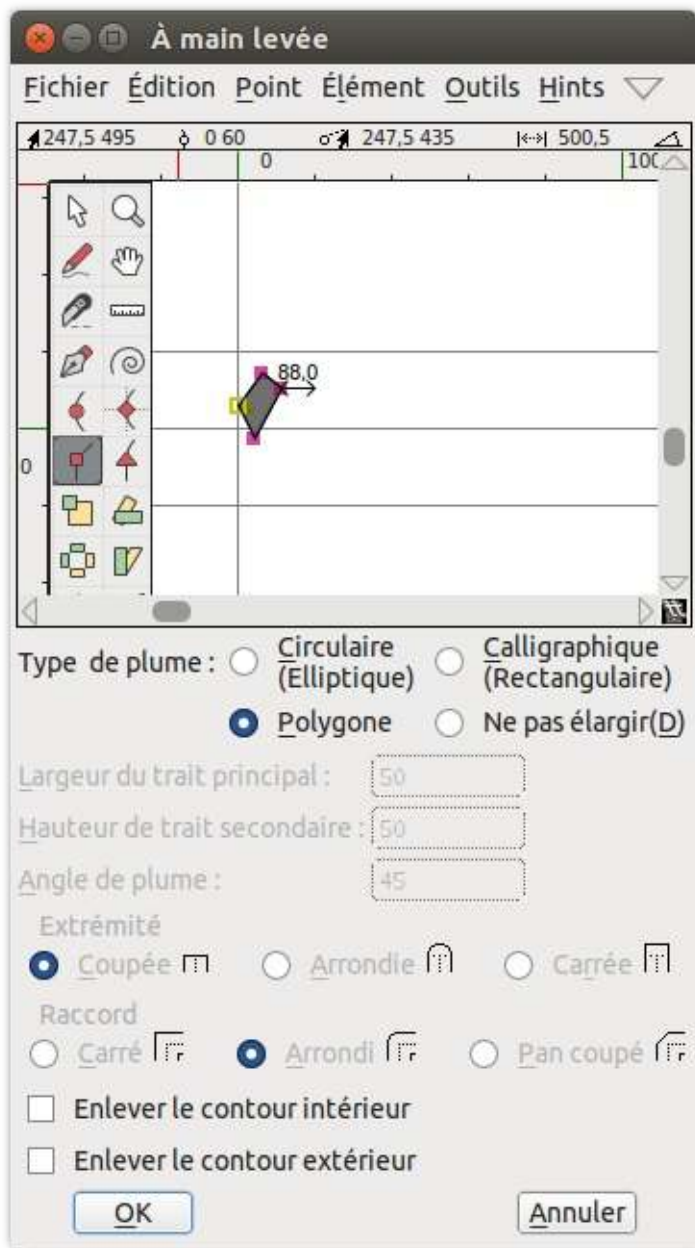
Choisissez l'outil Rotation 3D, déplacez-vous au milieu de la zone sélectionnée, puis cliquez lentement et faites glisser jusqu'à ce que vous voyiez quelque chose que vous aimez, puis relâchez. Voici un exemple de rotation 3D utilisé sur l'image de pratique:





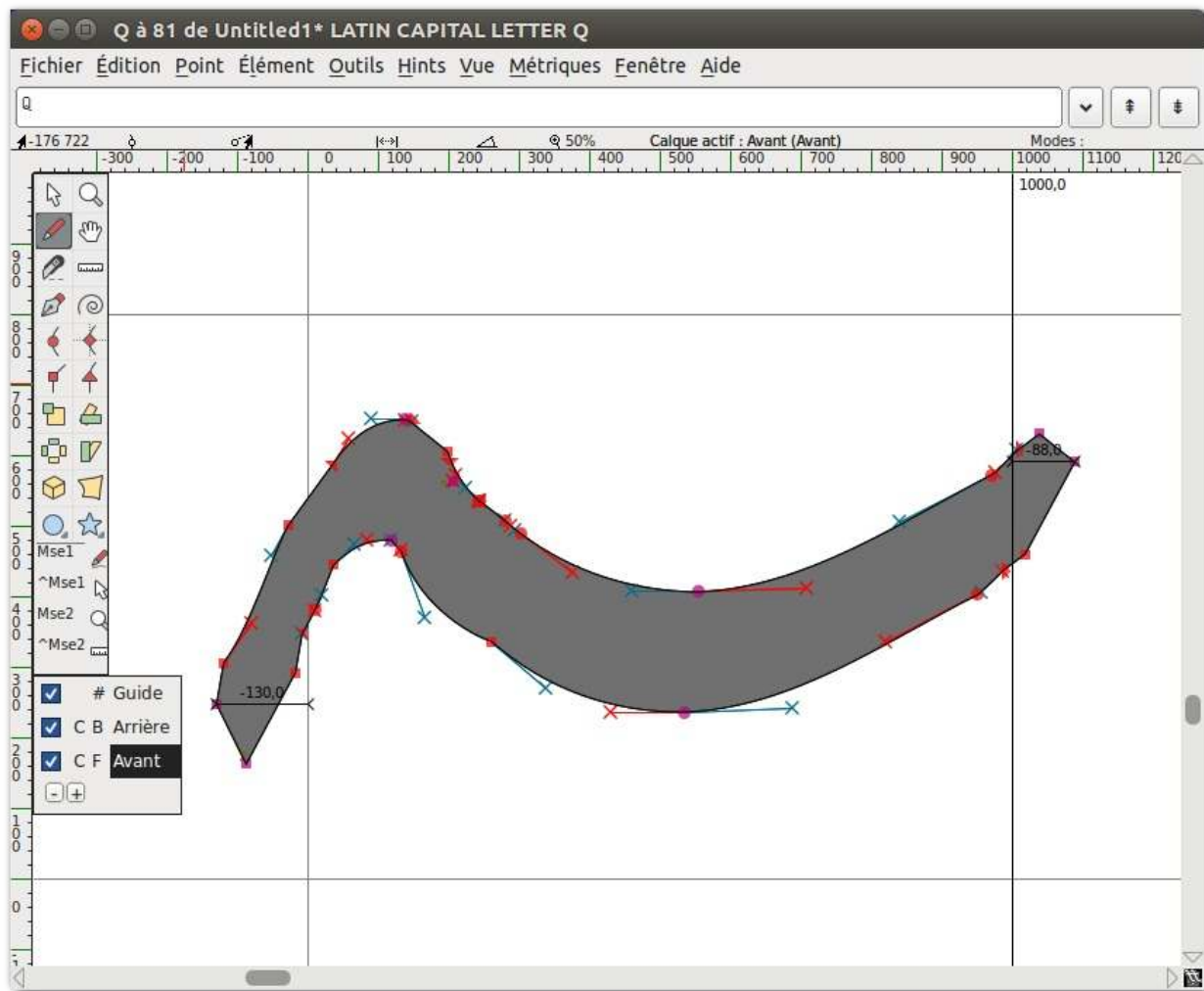
## Définir la forme et la largeur du trait

Jusqu'à présent, vous avez utilisé l'outil de dessin à main levée pour tracer une ligne. Si vous double-cliquez sur l'outil de dessin à main levée, vous obtenez la boîte de dialogue à main levée illustrée ici, qui contient une fenêtre de dessin. C'est là que vous sélectionnez la forme et la taille de la plume. Cette boîte de dialogue apparaît également lorsque vous choisissez l'option *Étendre ligne* dans le menu *Élément*.



À l'aide de l'outil point de coin, dessinez un polygone et cliquez sur OK.

Maintenant, tracez une ligne avec l'outil de dessin à main levée. Lorsque vous relâchez le bouton de la souris, le nouveau chemin est automatiquement tracé avec la forme que vous avez choisie dans la boîte de dialogue à main levée, comme montré ici.



## Continuez à dessiner!

Vous devriez continuer à expérimenter avec les outils de dessin jusqu'à ce que vous vous sentiez à l'aise pour dessiner et transformer toutes les formes dont vous avez besoin. À ce stade, vous êtes équipé pour commencer à construire les composants des glyphes, mais vous devriez également prendre le temps de regarder les autres ensembles d'outils de FontForge. Le chapitre suivant, "[Dessin avec Spiro](#)", décrit le mode de dessin Spiro. Le dessin Spiro est assez différent de l'édition de courbes de Bézier qu'il nécessite ses propres explications.

## Lectures complémentaires (en anglais)

A [TypeDrawers Forum Discussion on Beziers](#) incluant ces liens partagés par Nina Stössinger [sur twitter](#):

- [Bezier Curves and Type Design: A Tutorial](#) par Fábio Duarte Martins
- [So What's the Big Deal with Horizontal & Vertical Bezier Handles Anyway?](#)
- [Hand Lettering: How to Vector Your Letterforms](#) par Scott Biersack
- [Type Basics](#) par Underware
- [The Bézier Game](#) par Marc MacKay

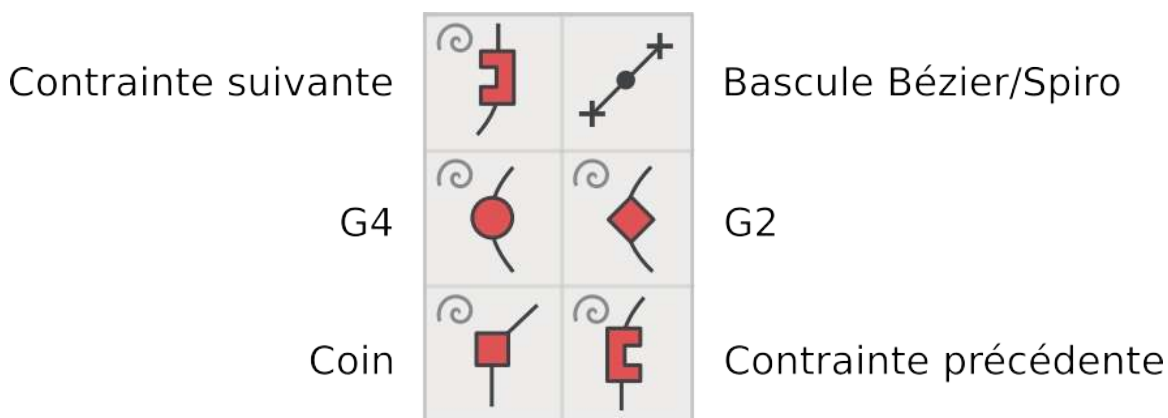
## Dessin avec Spiro

Spiro est une boîte à outils pour concevoir des courbes avec une méthode alternative aux courbes de Bézier plus traditionnelles. Bien que cela soit facultatif, FontForge peut être installé pour inclure un mode Spiro qui vous offre des outils pour créer ce type spécifique de courbes. Voir "[Installation de FontForge](#)" pour plus de détails sur la façon d'inclure la bibliothèque Spiro dans votre programme.

Le dessin avec Spiro a une approche différente, qui peut vous aider à obtenir des courbes faites d'une autre manière et à résoudre vos problèmes de conception. Veuillez expérimenter!

## Le jeu d'outils Spiro

Plusieurs des outils décrits dans le chapitre "[Utilisation des outils de dessin de FontForge](#)" sont aussi disponibles en mode Spiro, mais certains d'entre eux fonctionnent très différemment.



Il y a cinq types différents de points Spiro:

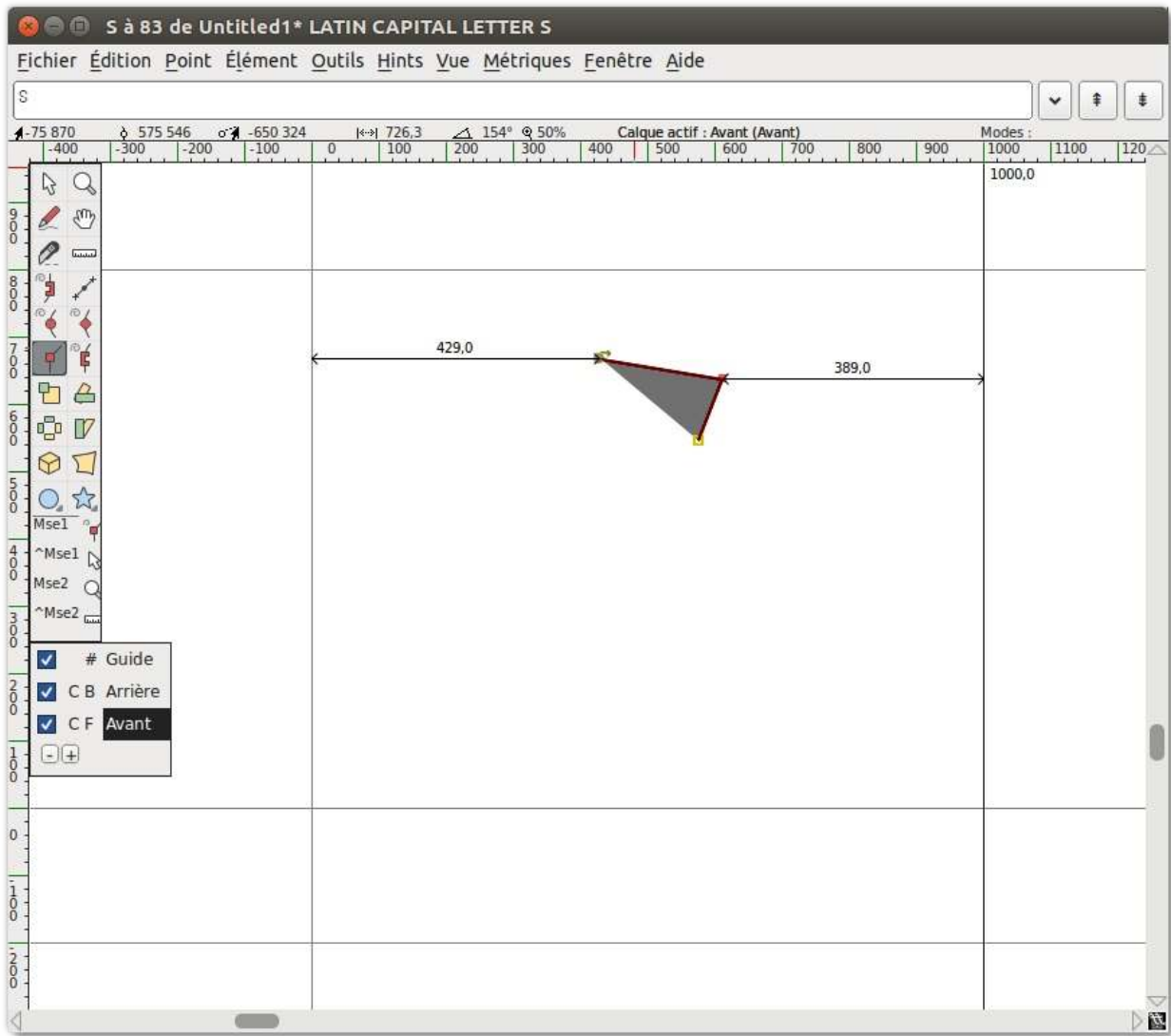
1. Points G4, utilisés pour une courbe plus douce
2. Points G2, utilisés pour une courbe plus prononcée
3. Points de coin pour angles abrupts
4. Points de contrainte précédente, utilisés lorsque le contour du chemin passe d'une courbe à une droite
5. Points de contrainte suivante, utilisés lorsque le chemin passe d'une ligne droite à une courbe

## Dessin d'un 'S' avec Spiro

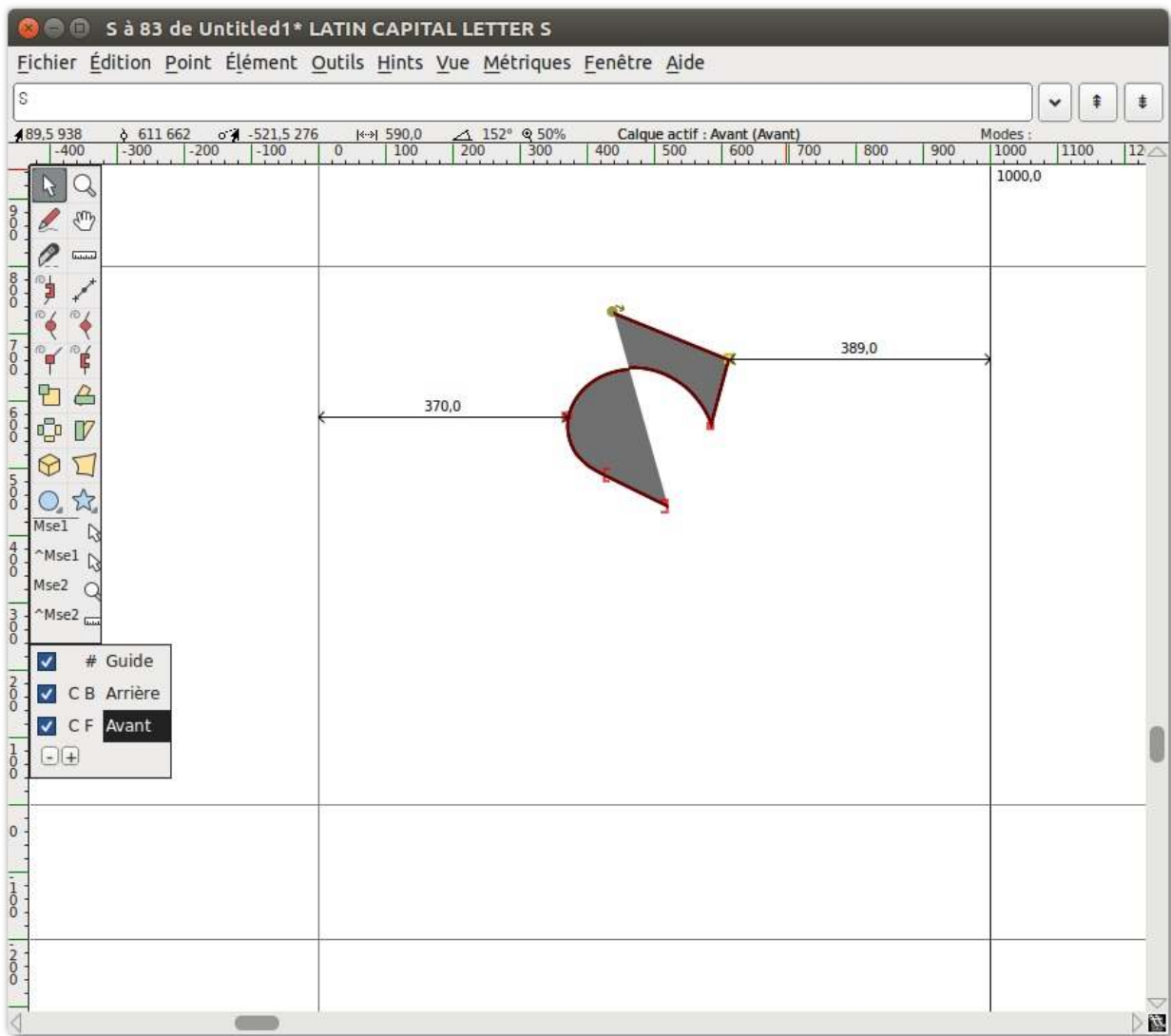
Cet exercice de dessin d'un 'S' avec Spiro vous rendra à l'aise avec ce mode.

**Conseil:** Lors du dessin en mode Spiro, commencez toujours par un point G4 ou G2. Commencer par les autres types de point ne fonctionne pas vraiment dans FontForge.

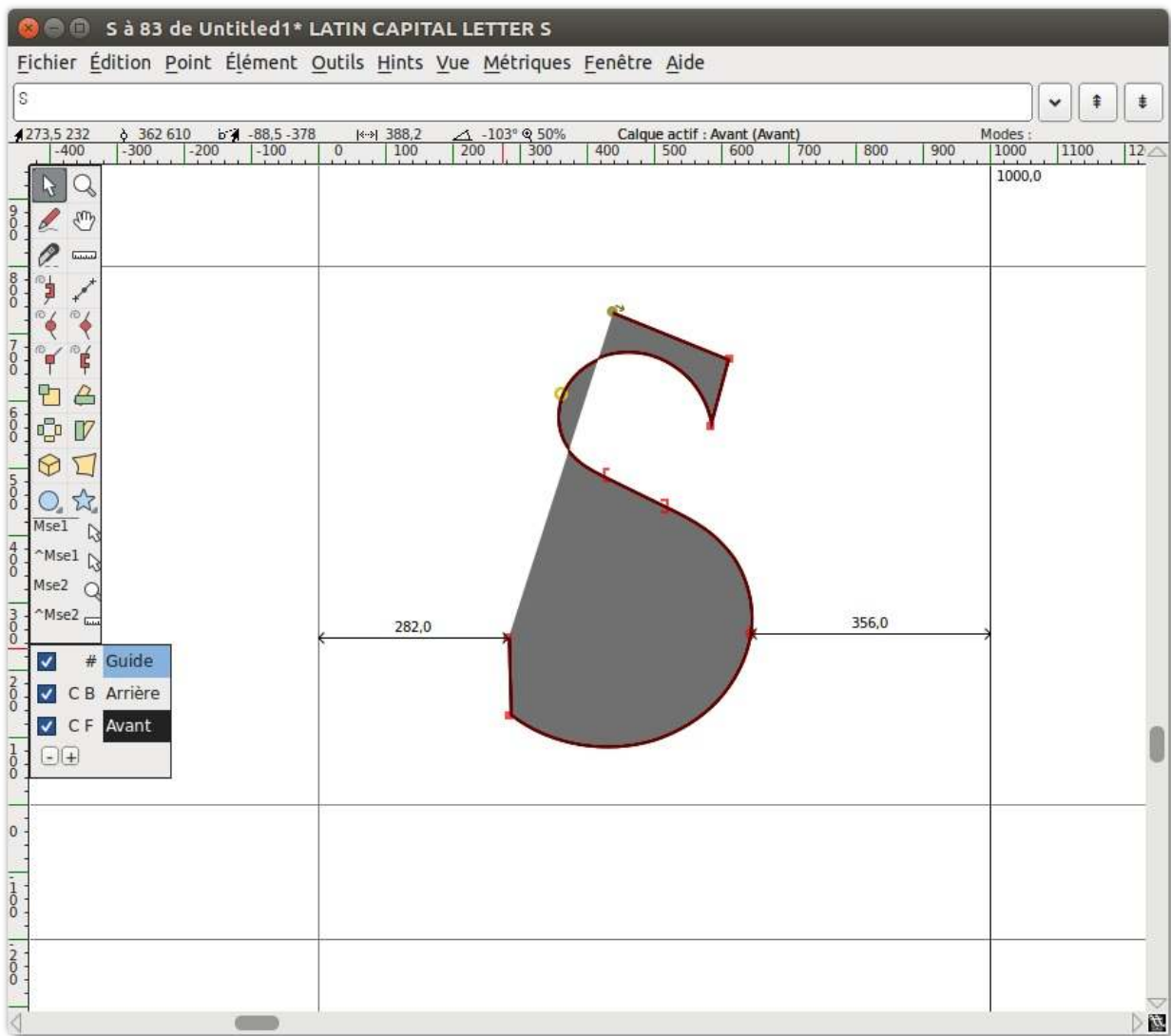
Commencez par un point G4 au point le plus élevé de votre 'S,' suivi d'un coin, puis d'un autre coin. Travaillez dans le sens horaire autour de la forme de la lettre.



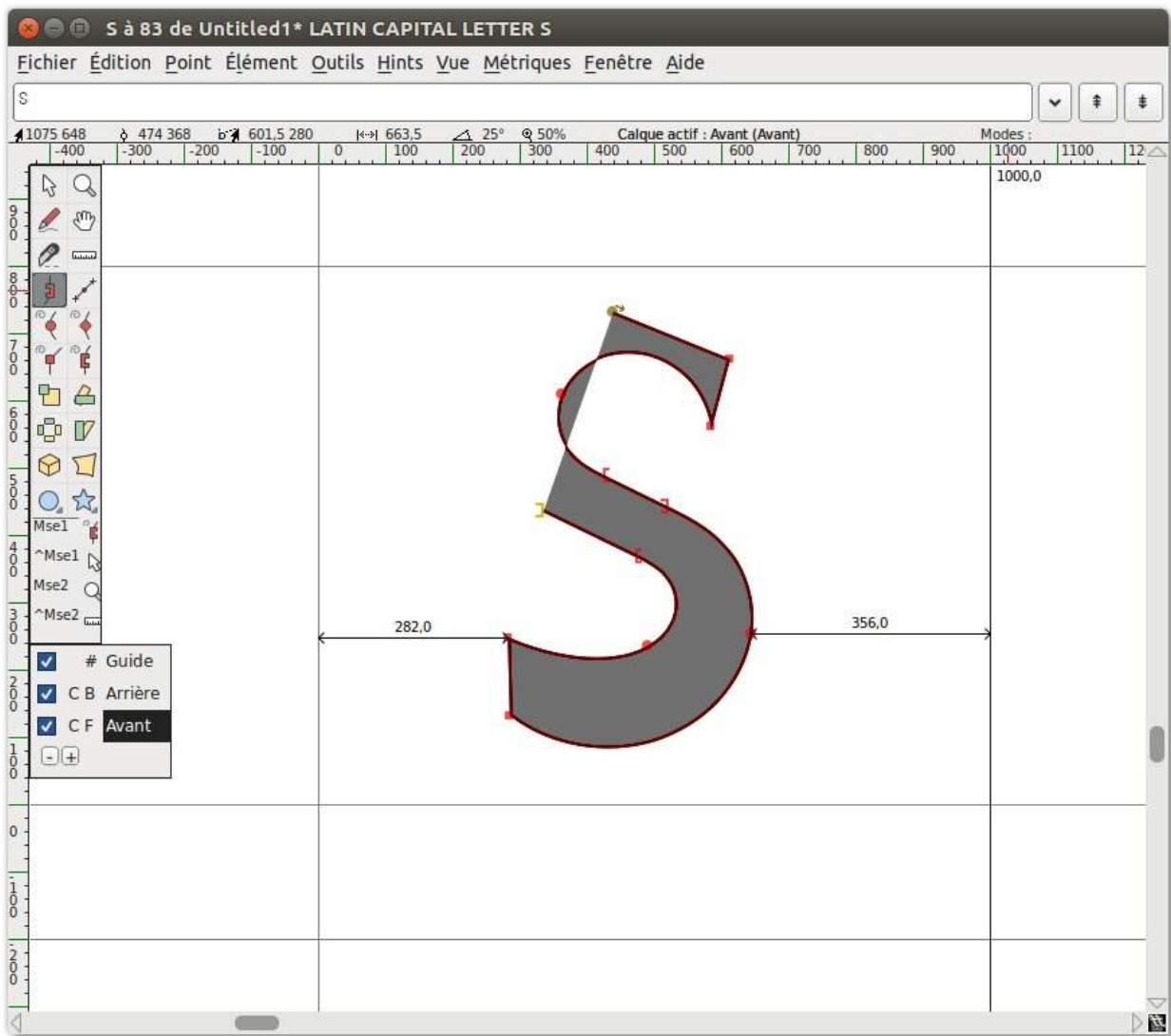
Continuez ceci avec un G4, un point de contrainte précédent et un point de contrainte suivant.



Ensuite, ajoutez un autre point G4, suivi de deux autres points de coin.

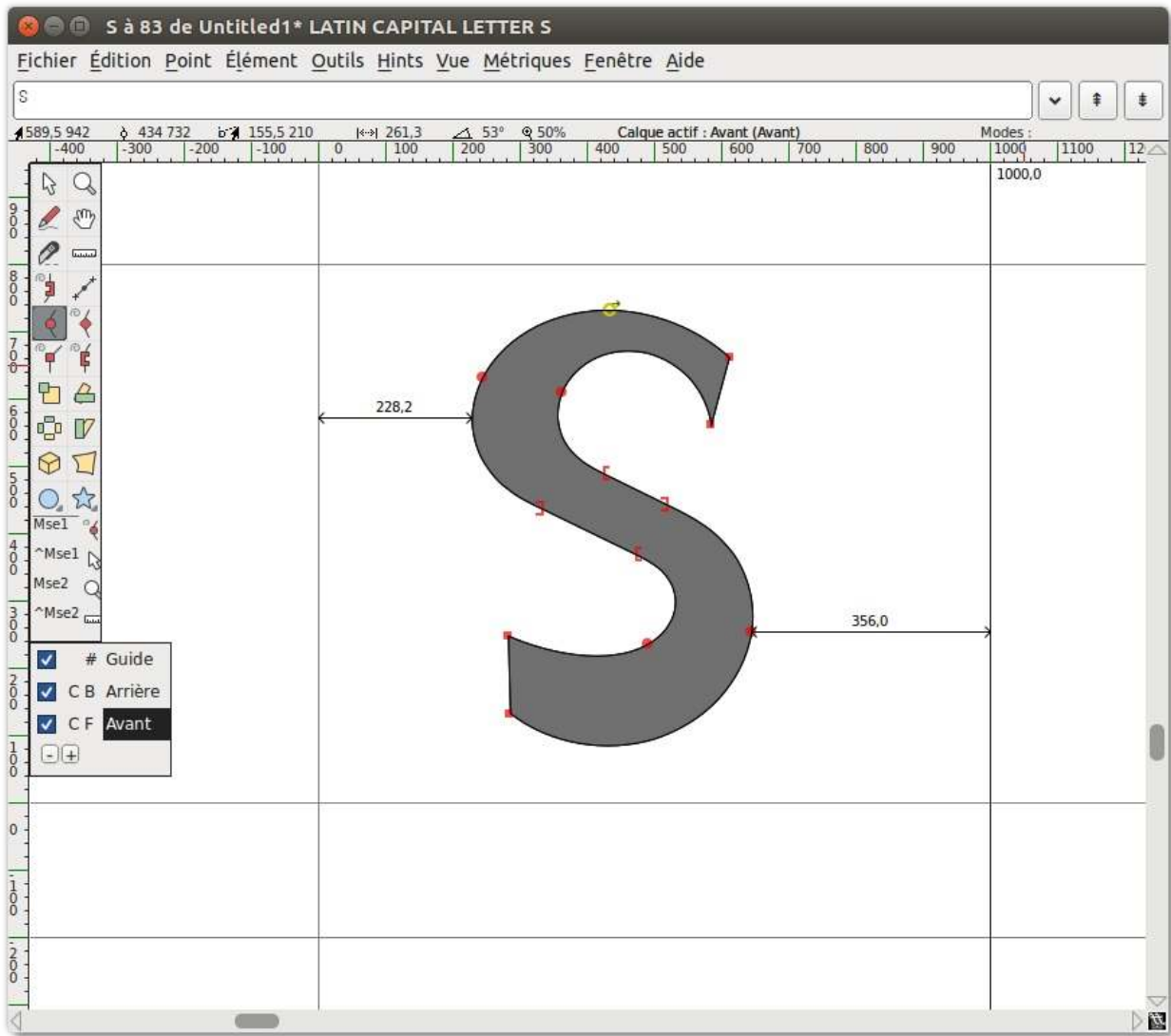


Ensuite, un G4, suivi d'une contrainte précédente, suivi d'une contrainte suivante.



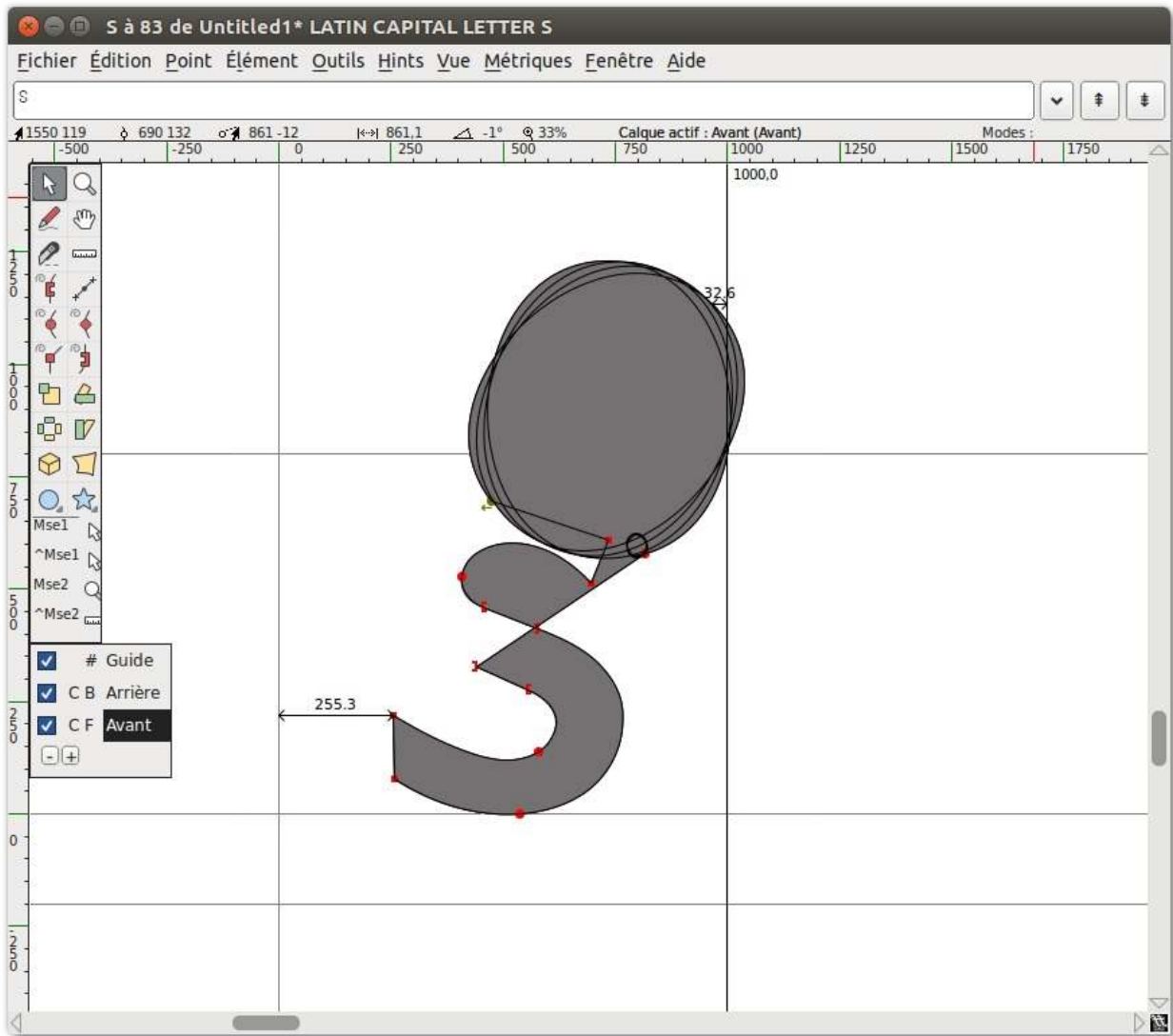
Ensuite, ajoutez un point G4 de plus, et enfin, fermez la forme au point de départ en cliquant dessus avec l'outil de point G4.





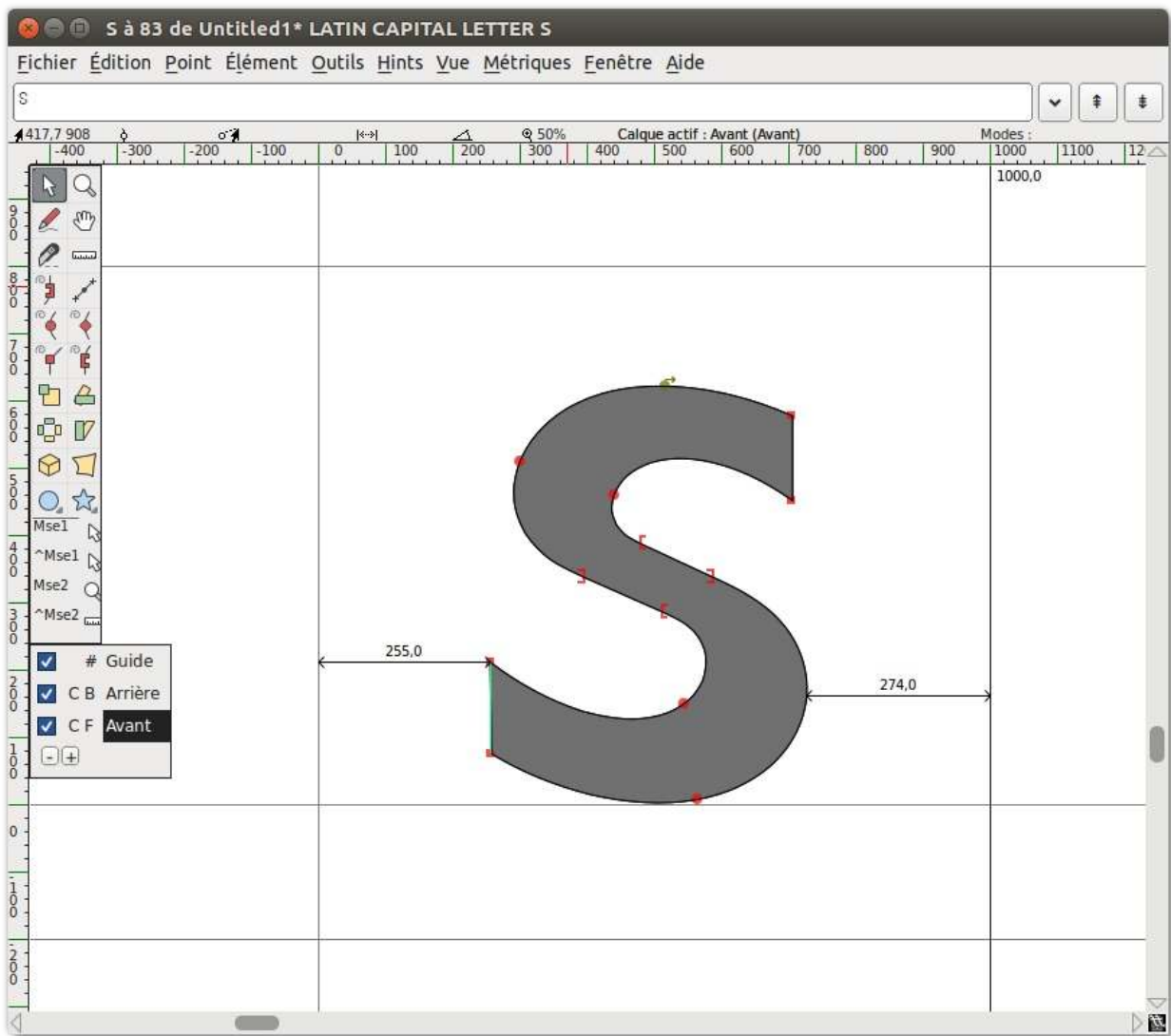
Maintenant vous avez presque un 'S'! Commencez à pousser les points autour pour obtenir le S désiré.

Oups, qu'est-ce qui s'est passé?

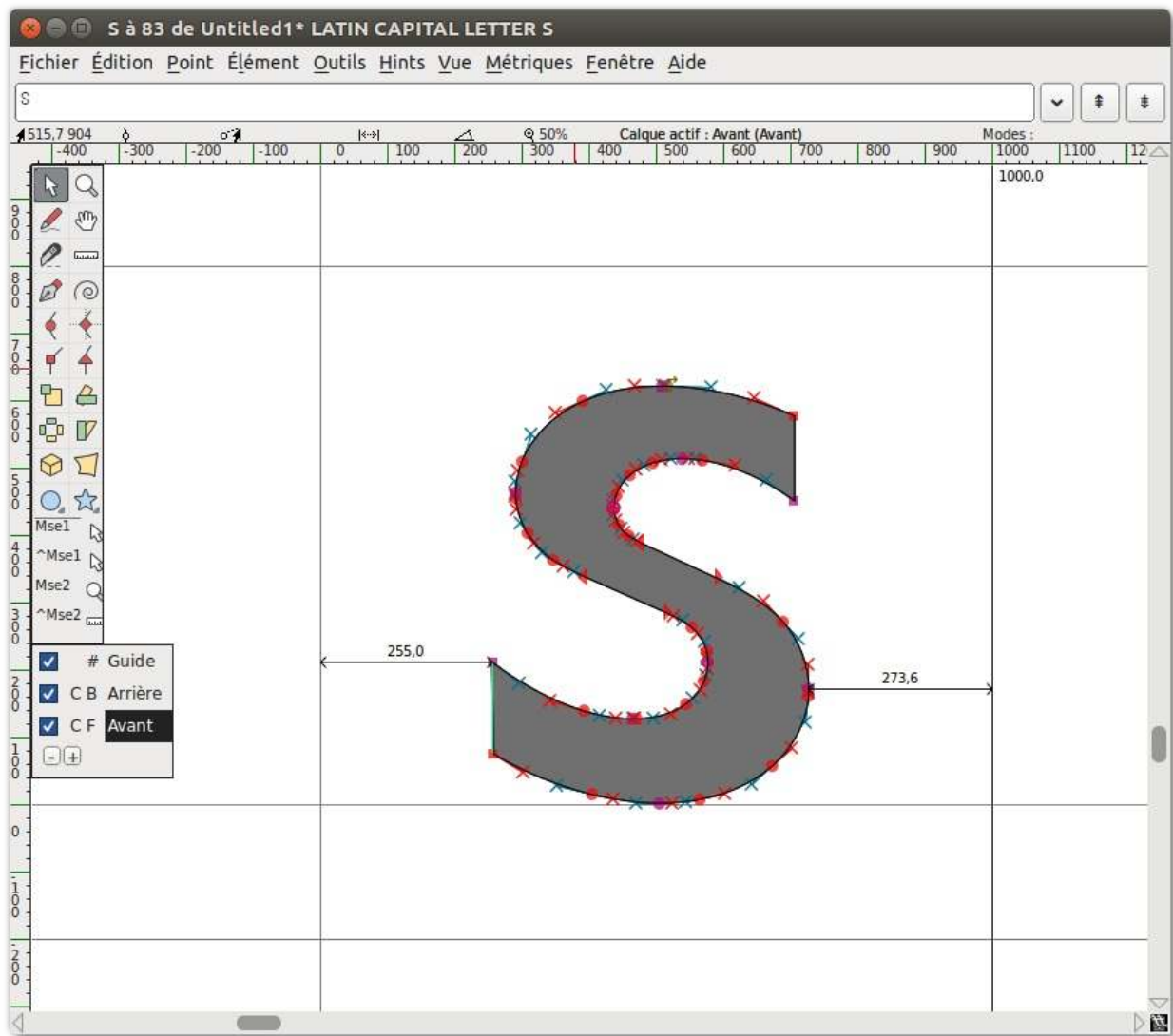


Ne vous inquiétez pas – Spiro fait parfois de drôles de choses. Il suffit de cliquer sur *Annuler*, ou de continuer à pousser les points pour remettre les choses sur la bonne voie.

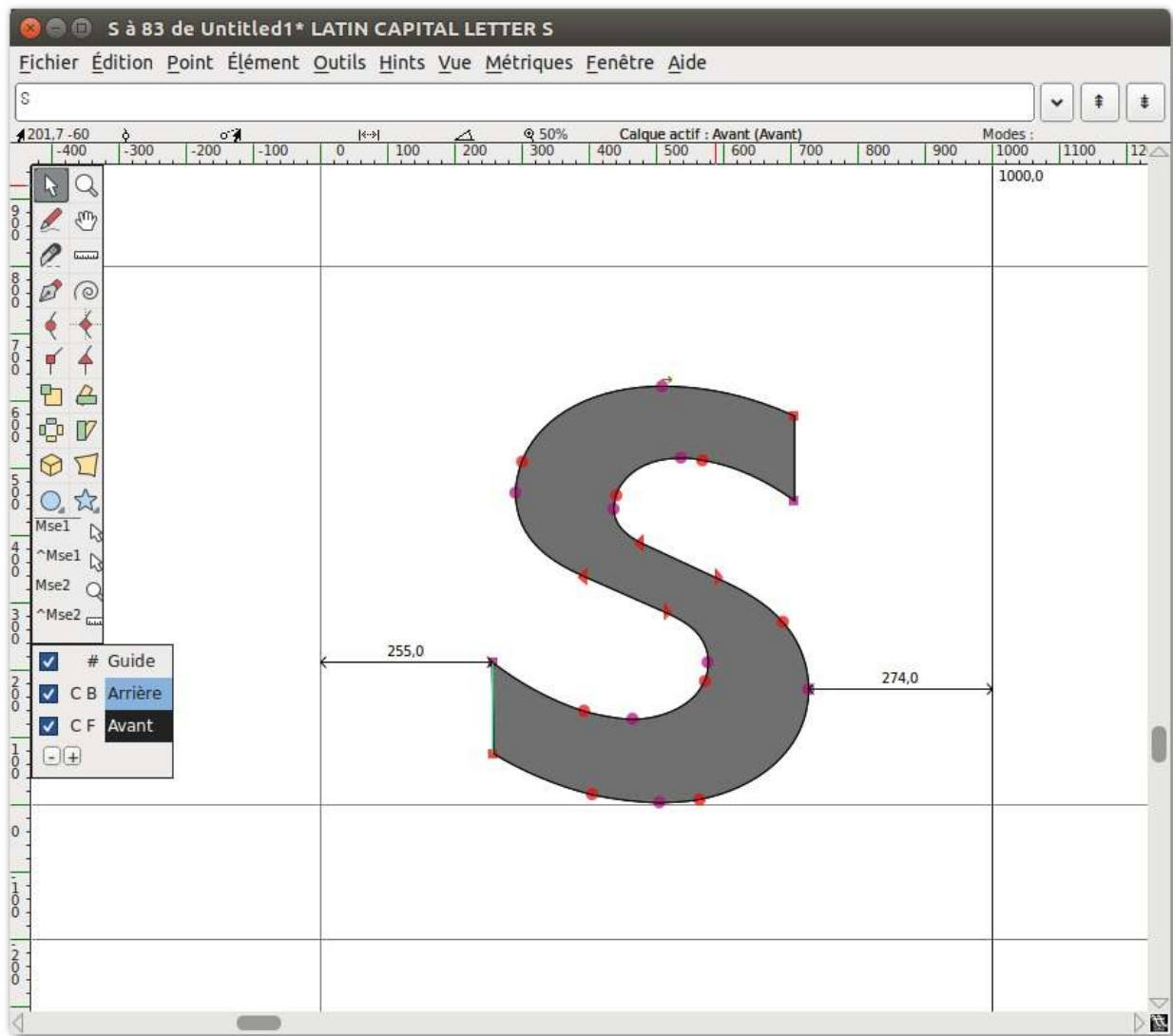
Maintenant, vous devriez voir quelque chose comme ceci:



Basculer hors du mode Spiro en mode Bézier. Vous remarquerez qu'il ya beaucoup de points sur la courbe résultante – Vous voudrez peut-être nettoyer certains d'entre eux.



Pour nettoyer ces points supplémentaires, allez dans le menu **Élément** et sélectionnez *Simplifier > Simplifier* . Ensuite, allez dans *Élément > Ajouter des extrema* . Enfin, allez dans *Élément > Arrondir > Aux entiers* . Après ces opérations de nettoyage, vous verrez quelque chose comme ceci:



Vous pouvez continuer à expérimenter avec le mode Spiro pour obtenir une idée de la façon dont il diffère du dessin avec Bézier. La terminologie est différente, mais comme avec les autres outils de dessin et de réglage de FontForge, avec la pratique vous obtiendrez ce que vous voudrez.

## Création de “o” et “n”

Il existe de nombreuses approches pour concevoir une police. Il peut être utile de déconstruire les processus plus importants afin de commencer rapidement, et de fournir une base solide pour la totalité de la police de caractères. Une approche populaire et bénéfique est de concevoir les lettres ‘o’ et ‘n’ d’abord, établir les éléments essentiels de leur forme, de l’espace et de leur équilibre, avant de les réunir pour former d’autres lettres. La création des lettres minuscules ‘o’ and ‘n’ peut nous fournir certaines des formes et des structures fondamentales qui sous-tendent tous les autres lettres nécessaires.

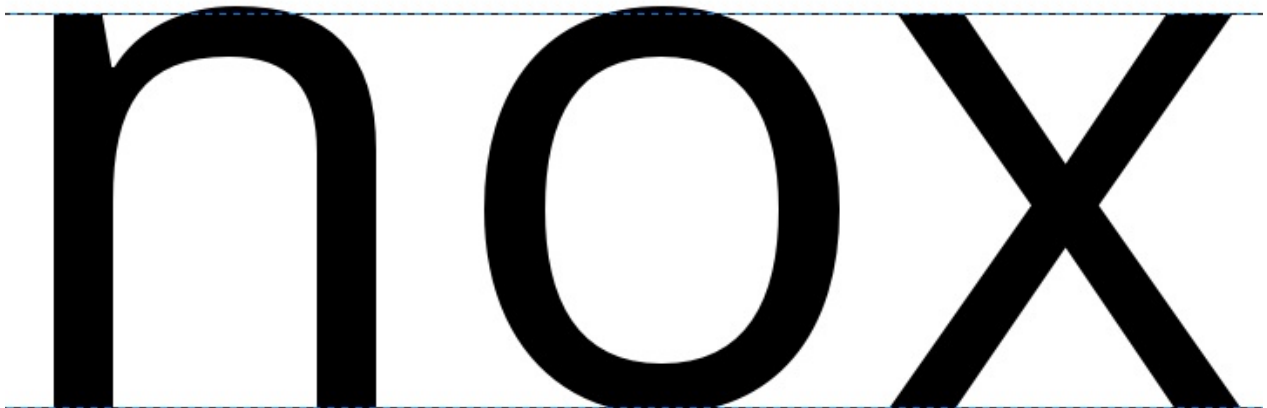
Bien que le design du ‘o’ puisse sembler assez simple, toutes les caractéristiques mentionnées dans le chapitre [“Qu’est-ce qu’une police de caractères?”](#) entrent en jeu. Le choix que vous faites sur chaque caractéristique doit être un choix délibéré.

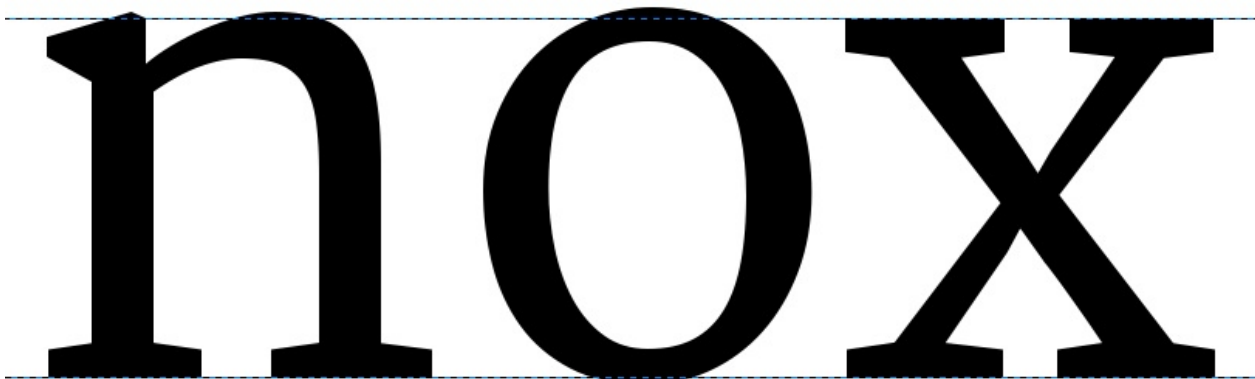
## Dépassement des rondes

Comment les courbes et les bords droits apparaissent à l’œil est un effet d’optique qui affecte la conception de polices. Par exemple, pour qu’une courbe et un bord droit semblent alignés correctement sur la ligne de base, la courbe doit en fait se situer un peu au-dessous de la ligne, produisant un *undershoot*. La partie du caractère qui descend juste en dessous de la ligne de base afin d’apparaître assis sur la ligne de base s’appelle le *underhang* – démontré ci-dessous. Sans *underhang*, les caractères avec des courbes autour de la ligne de base apparaîtront désalignés dans une ligne de texte.



De façon similaire au *undershoot*, un dépassement (*overshoot*) est nécessaire pour fournir l’illusion d’alignement à la hauteur d’x et à la hauteur de capitale (voir ci-dessous).





## Concevoir le ‘o’ minuscule

La conception du ‘o’ n’est pas seulement à propos de la partie noire de la lettre. Alors que le ‘o’ fournit le poids et la forme de base de la panse, le blanc – ou contrepoint – fournit la taille et la forme utilisée par le reste de la police. D’une manière générale, on peut également observer que la forme ronde du ‘o’ se répétera dans les autres caractères. Ceux-ci incluent les b, c, d, e, p et q, et la forme impliquera également le façonnement des formes des courbes dans tout autre caractère de la police, comme les O, C, D et Q.

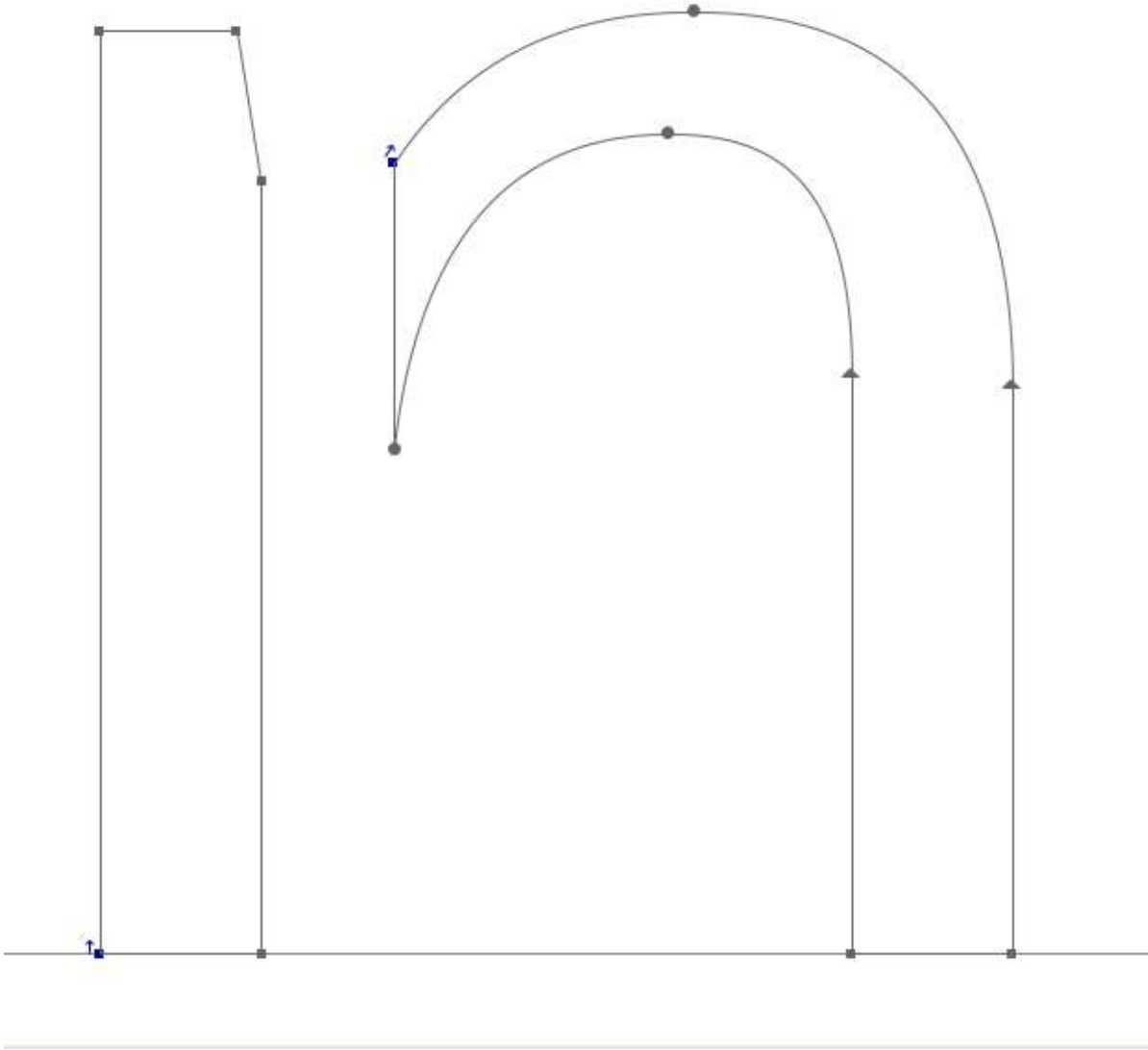
En outre, le blanc à l’intérieur du ‘o’ doit être utilisé lors de la conception de l’espacement de notre police; le ‘o’ établit le rythme de référence des espaces utilisés entre tous les autres glyphes de la police. Ces deux valeurs sont très liées, donc essentiellement vous devrez concevoir la quantité d’espace blanc qui sont les approches latérales de votre ‘o’ aussi. Comme principe général, à l’exception des polices inclinées ou italiques, le ‘o’ devrait avoir la même quantité d’espace sur les côtés gauche et droit, et l’espace blanc entre une chaîne de caractères de ‘o’ devrait équilibrer l’espace blanc à l’intérieur des ‘o’s.

Ici, nous empiétons un peu sur le territoire de l’espacement et des métriques, donc même à ce stade précoce, vous voudrez peut-être jeter un oeil au chapitre [“Espacement, métriques, et crénage”](#), qui couvre les implications de base de l’espacement dans une police. Cela devrait vous permettre de faire un ‘o’ bien espacé, ce qui vous aidera à concevoir le ‘n’.

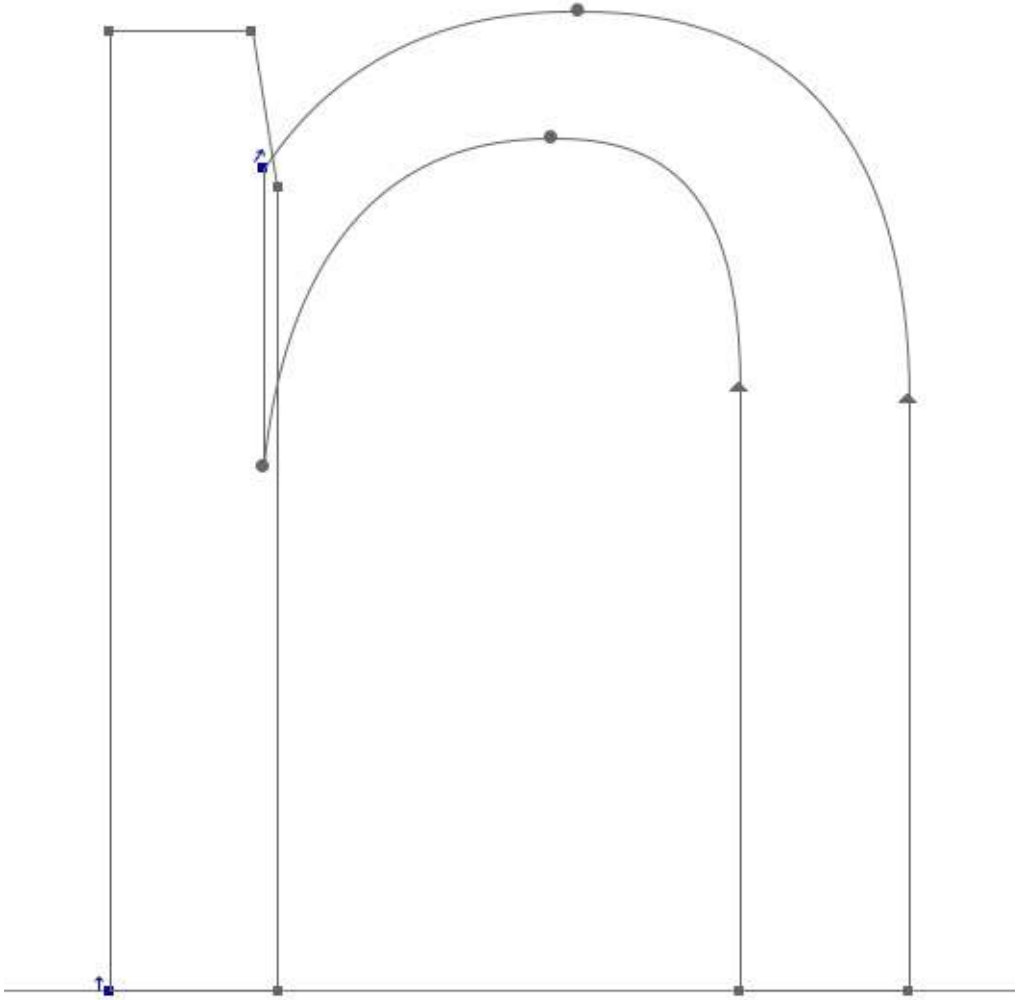
## Concevoir le 'n' minuscule

Une fois que vous êtes satisfait de la forme et de l’espacement de votre caractère ‘o’ minuscule tel qu’observé dans un exemple de chaîne, la prochaine étape de cette approche est de créer un ‘n’ minuscule de forme appropriée, équilibrée, et bien espacées, que vous injectez dans votre chaîne de ‘o’s.

Si l’on considère l’anatomie d’un ‘n’, on peut le diviser en deux ou trois composants constitués d’un *fût* et d’une *courbe*. Cette approche peut nous fournir un raccourci pour maintenir l’équilibre et l’harmonie au sein de nos caractères à mesure qu’ils sont formés, et que notre ensemble de caractères grandit. En regardant l’échantillon ‘n’ ci-dessous; Il est divisé en deux composants. Ces composants séparés se combinent pour former un ‘n’, mais les mêmes composants seront réutilisés plus tard lors de la formation d’autres caractères; par exemple, le fût à gauche du ‘n’ peut être utilisée pour former le fût gauche de tous les autres caractères minuscules.

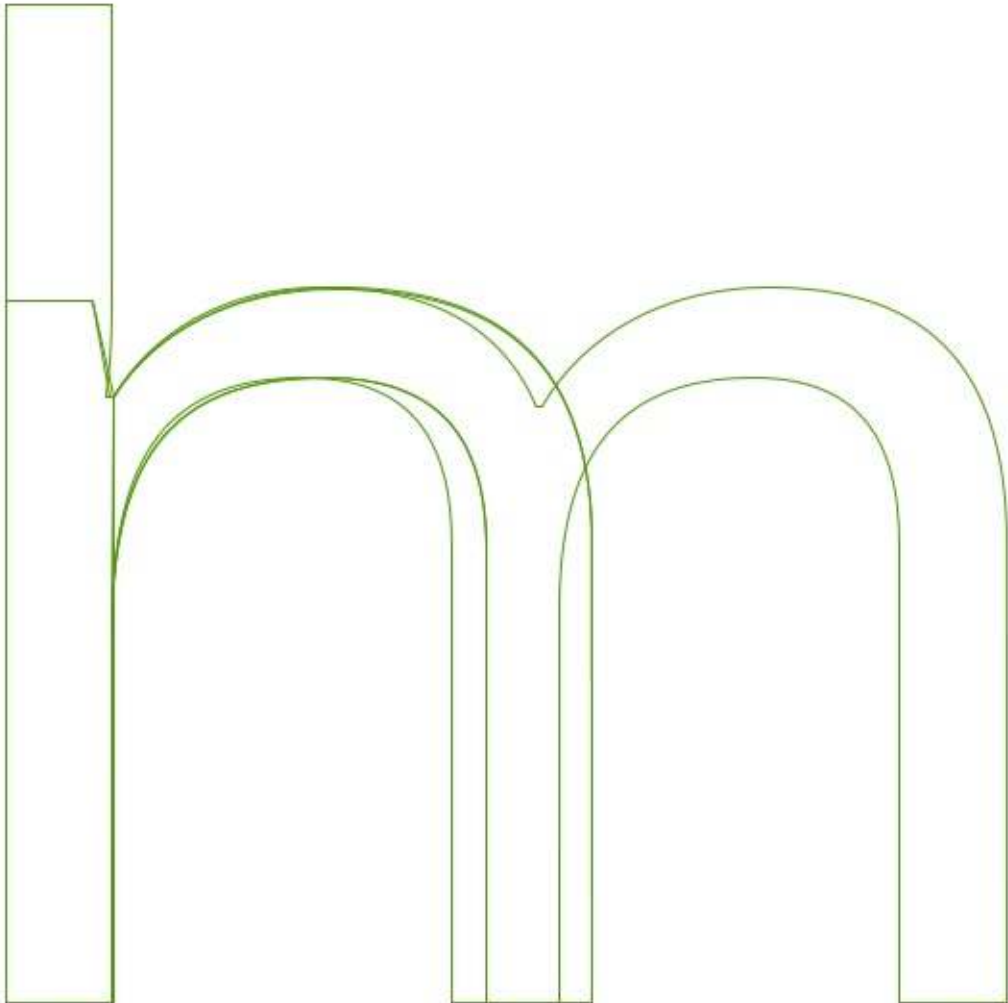


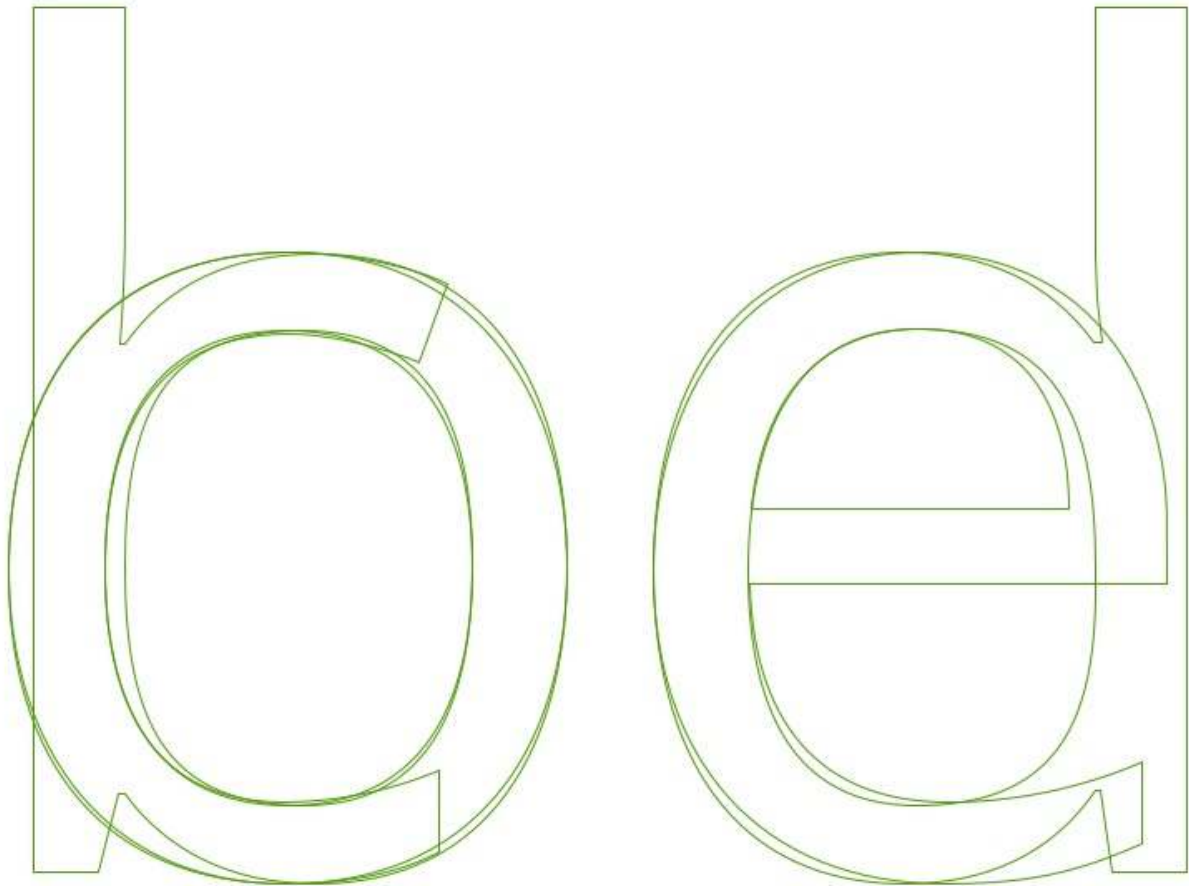


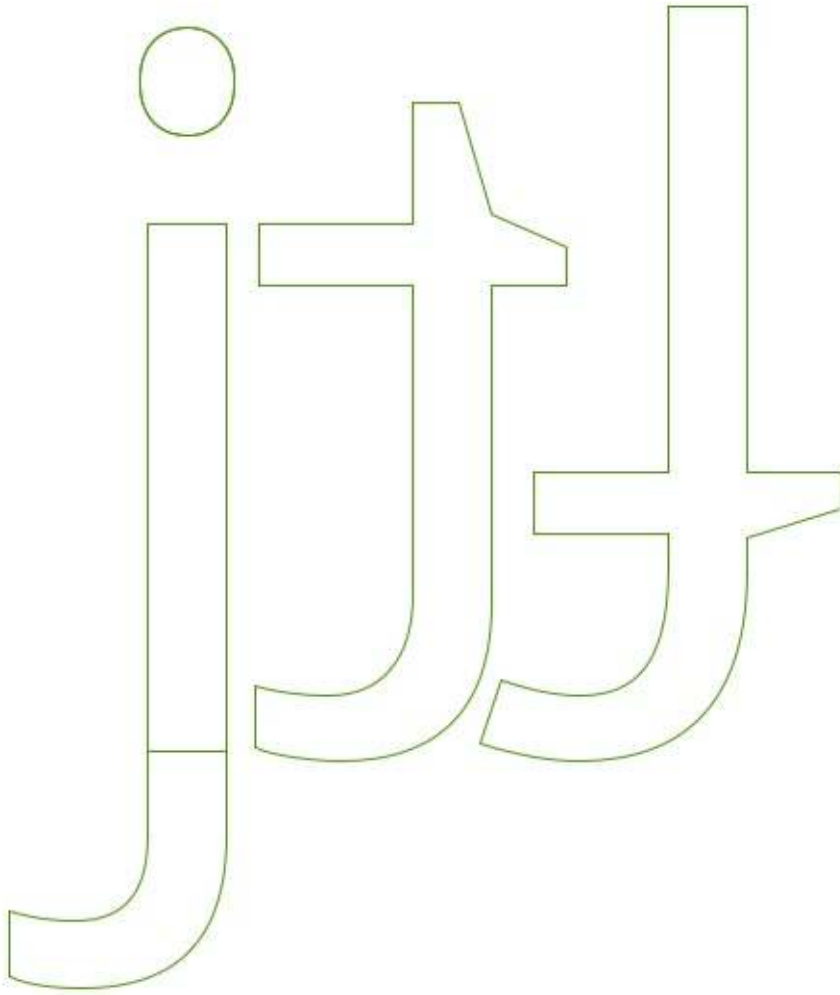


En vous reportant à nouveau au chapitre sur l'espace et les métriques, la conception du caractère 'n' devrait suivre le rythme du processus d'espace entre les caractères 'n' et 'o' ensemble.

Maintenant, en recueillant les méthodes que vous avez utilisées pour créer un caractère 'n' et 'o', vous êtes prêt à développer le jeu de caractères minuscules. Les qualités des composants du fût et de la courbe du 'n' et du 'o' indiqueront la façon dont vous pourrez former les autres caractères. En étudiant les caractères ci-dessous de [Open Sans](#), nous pouvons voir les relations entre les formes de caractères séparés et comment elles peuvent être répétées, avec quelques ajustements, pour former les composants de notre police.



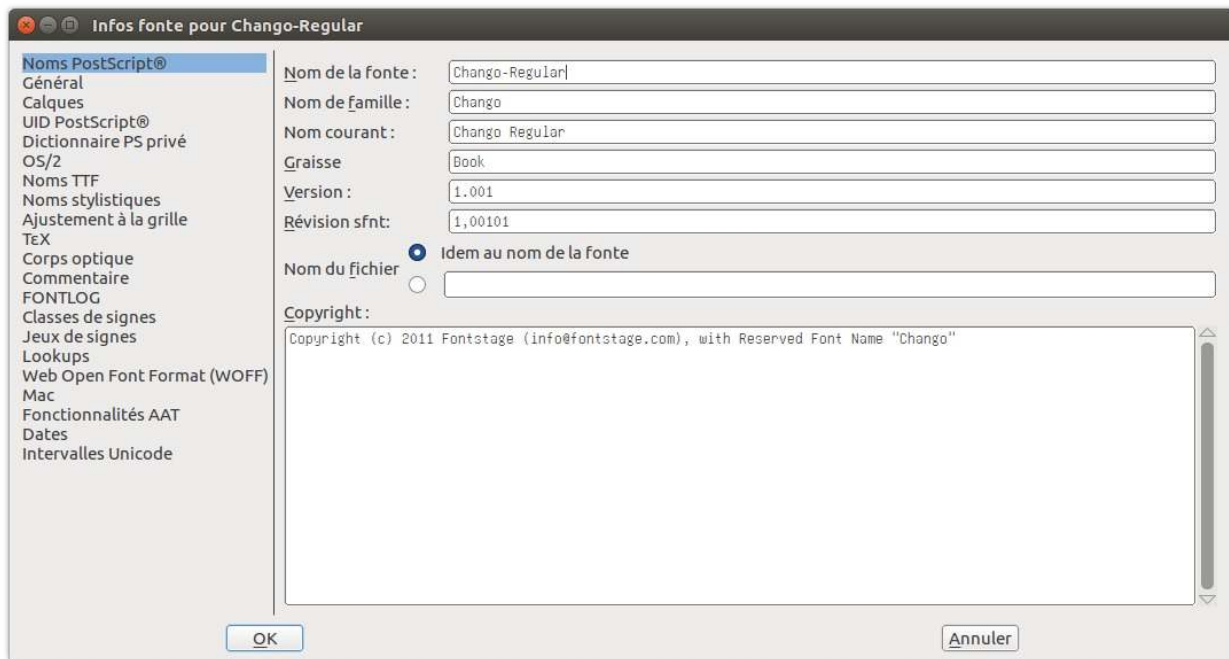




## Infos fonte et métadonnées

### Élément, Infos fonte

La fenêtre Infos fonte est omniprésente dans les éditeurs de polices et FontForge suit de près la spécification OpenType. Cela peut sembler cryptique au début, mais l'utiliser peut aider à apprendre le format OpenType, et à son tour la lecture sur le format OpenType rend la boîte de dialogue plus accessible.



### Numérotation des versions

Les développeurs de logiciels aiment utiliser la [gestion sémantique de version](#) pour leurs programmes, et c'est aussi une bonne idée pour vos polices. D'une certaine manière, les polices sont une "API" pour que le texte accède à certains sentiments associatifs chez les lecteurs.

Une version MAJEURE serait après une refonte complète. Comparez [Exo](#) et [Exo 2](#). Si vous avez un document utilisant Exo, vous ne voulez pas tomber dans Exo 2, parce que le sentiment évoqué, le «ton» ou «saveur», est (subtilement) différent. Prendre en charge une ou plusieurs nouvelles écritures qui sont assez similaires en hauteur, ou ajouter un nombre substantiel de langues, pourrait également constituer une révision MAJEURE, comme toute autre chose qui modifie substantiellement les métriques verticales ou horizontales. Cependant, si un design complémentaire est fait pour 2 écritures, il peut être préférable d'émettre 2 ou 3 familles, une avec chaque écriture mise à l'échelle de façon appropriée et les autres écritures mises à l'échelle comme des polices secondaires pour la typographie simple de textes multilingues.

Une version MINEURE serait tout ce qui modifie subtilement les métriques, telles que les métriques verticales, les approches horizontales ou l'amélioration du crénage, ou l'apport de corrections mineures à certains glyphes, car de telles mises à jour entraîneront la repagination des documents utilisant la police (bien que subtilement dans de nombreux cas). Voici un exemple (en anglais) de <http://www.fastcodesign.com/3033126/roboto-rebooted-why-google-plans-to-update-its-font-like-the-rest-of-its-products>:



L'ajout de quelques glyphes ou d'une douzaine d'entre-eux pour "compléter" la couverture d'un jeu de caractères précédemment planifié ou l'ajout de la prise en charge de quelques langues supplémentaires est probablement MINEUR, surtout si cela ne change pas les métriques verticales.

Un changement de type CORRECTIF serait tout ce qui améliore la police sans changer les métriques ou ne modifie pas le design d'un glyphe de manière visible, sans affecter la disposition finale du texte. Votre version 1.001 pourrait ne pas avoir le champs fsType mis à 0 ou ne pas être exécuté encore par fontcrunch. La modification de ces deux choses dans une version 1.0.1 ne sera pas visible ou ne créera aucune repagination. Malheureusement, le troisième numéro de version de type CORRECTIF n'est pas disponible dans les champs de métadonnées des polices OpenType. Au lieu de cela, incrémenter le numéro de version MINEUR pour des modifications au hinting ou aux métadonnées.

De plus, la version ne doit pas contenir plus de 3 décimales, alors qu'elle peut être représentée par 5 dans un fichier TTX. Par exemple 2.001 est typique, et peut apparaître comme 2.00099 dans un fichier XML TTX.

Si vous créez de nouvelles versions de polices libres, les fonctions de [Github Releases](#) sont très utiles.

## Nom de famille

Microsoft travaille dur pour s'assurer qu'un programme écrit pour une version antérieure de Windows continuera à s'exécuter sur les dernières versions, incitant les gens à se mettre à niveau. Cela signifie que le modèle de police TrueType de base présent dans Windows 3 existe toujours, et Windows ne prend pas en charge les familles de polices avec plus de 4 styles de base (Régulier, Italique, Gras, Gras Italique.)

Cela signifie pour les concepteurs de polices que nos noms de famille de polices doivent être configurés de manière à ce que nos polices puissent être utilisées dans tous les systèmes d'exploitation. Le format OpenType le permet, en complétant les valeurs de nom de famille et de style avec les valeurs "Nom de famille préféré" et "Nom de style préféré" qui auront priorité dans un logiciel interprétant le format OpenType.

Cette feuille de calcul de [Google Docs](#), [Family Naming](#) est basée sur des informations partagées par l'expert en polices polonais Adam Twardoch et discutées dans le [forum Fontlab](#). Elle remplace l'[exemple de spécification OpenType](#).



## Inter-mot

Il peut sembler drôle de prêter une attention particulière à l'espace entre les mots, mais le blanc est l'une des parties les plus couramment utilisées dans la conception de police. Un espace de mots trop large ou trop étroit peut ruiner le design d'une police. Il n'est pas trop tôt pour commencer à considérer l'espacement des mots dès que vous avez conçus vos premiers caractères. Le choix que vous faites à ce stade doit être progressivement ajusté pendant que vous progressez dans la conception de la police.

L'inter-mots ici est trop serré...

The concept of produsage highlights that within the communities which engage in the collaborative creation and extension of information and knowledge that we examine on this site, the role of consumer and even that of end user have long disappeared, and the distinctions between producers and users of content have faded into comparative insignificance. In many of the spaces we encounter here, users are always already necessarily also producers of the shared knowledge base, regardless of whether they are aware of this role - they have become a new, hybrid, produser.

Et ici, c'est trop large...

The concept of produsage highlights that within the communities which engage in the collaborative creation and extension of information and knowledge that we examine on this site, the role of consumer and even that of end user have long disappeared, and the distinctions between producers and users of content have faded into comparative insignificance. In many of the spaces we encounter here, users are always already necessarily also producers of the shared knowledge base, regardless of whether they are aware of this role - they have become a new, hybrid, produser.

Maintenant c'est bien équilibré...



The concept of produsage highlights that within the communities which engage in the collaborative creation and extension of information and knowledge that we examine on this site, the role of consumer and even that of end user have long disappeared, and the distinctions between producers and users of content have faded into comparative insignificance. In many of the spaces we encounter here, users are always already necessarily also producers of the shared knowledge base, regardless of whether they are aware of this role - they have become a new, hybrid, producer.

Si votre police est destinée à être utilisée à de plus grandes tailles, alors l'inter-mots peut être réduit — et *vice-versa* si elle doit être utilisée à de très petites tailles.

La recherche a montré qu'un inter-mots qui est trop grand est plus tolérable que celui qui est trop petit, donc si vous n'êtes pas certain, vous devriez pencher dans cette direction.

**Remarque:** Des études similaires ont montré que les enfants plus jeunes bénéficient en particulier d'inter-mots plus grands que ce qui est considéré normal pour les lecteurs adultes.

Linda Reynolds et Sue Walker (2004) – 'You can't see what the words say: word spacing and letter spacing in children's reading books', *Journal of Research in Reading*, vol 27, no.1, pp. 87-98.

## Création de l'ADN de votre police

Après avoir terminé un solide design ainsi que l'espacement entre le 'o' et le 'n', la prochaine chose à faire est de commencer à remplir la police avec des lettres dont les caractéristiques structurales fournissent une base utile pour faire beaucoup d'autres lettres dans la police.

Il peut être tentant de se précipiter et de peupler votre police aussi rapidement que possible avec toutes les lettres — résister à cette envie! Alors que le 'o' et le 'n' fournissent un excellent point de départ à la fondation du design, nous devons établir le reste. Une expansion rapide avant que cela ne soit fait signifie que le projet complet deviendra plus difficile à gérer — et prendra plus de temps qu'il n'en faudrait.

De quoi avons-nous besoin pour la fondation de notre design? — Tout d'abord, regardons ce que nous avons avec nos 'n' et nos 'o'.

Bien que le 'o' soit particulièrement utile pour calculer l'espacement de base, cela ne nous aidera pas à concevoir d'autres caractères — pas même nécessairement le 'b' ou le 'd'. La lettre 'n', d'autre part, est très utile car elle aide à faire le 'm', 'h', et 'u'. L'autre facteur que nous devons peser lors du choix des lettres pour notre fondation est la fréquence à laquelle la lettre est utilisée. Une lettre qui est utilisée beaucoup nous aidera à faire des mots de test. Certaines lettres peuvent être choisies presque exclusivement pour cette raison particulière.

Les lettres que vous choisissez ne doivent pas nécessairement être celles suggérées ici. Elles devraient simplement avoir les caractéristiques qui sont discutées. Ainsi, par exemple, vous pouvez utiliser "a d h e s i o n" pour commencer. Cet ensemble de lettres est ce qui est utilisé dans le cours de Maîtrise de l'Art sur la conception de polices à l'Université de Reading, au Royaume-Uni. Une alternative est "v i d e o s p a n" qui est utilisée par la fonderie *Type Together* pour démarrer leurs projets dans leurs ateliers de conception de police. L'un ou l'autre ensemble a assez d'ADN pour être significatif, et les deux sont petits, ainsi ils sont faciles à changer 'globalement'.

Bien qu'il peut être plus facile d'utiliser simplement l'un des ensembles ci-dessus de lettres, vous pouvez également construire le vôtre. Demandez-vous quel ensemble de lettres vous devriez choisir pour ajouter à 'n' et 'o'. Considérez les options suivantes:

- 'a' — la lettre 'a' est aussi un choix de départ très courant. Le 'a' peut également être utile pour anticiper à quoi ressembleront les terminaisons du 's'.
- 'd' — la forme du 'd' peut vous apprendre beaucoup de choses sur la conception de 'b', 'p' et 'q'.
- 'e' — en anglais et dans bien d'autres langues, la lettre 'e' est particulièrement courante — ce qui la rend particulièrement précieuse. La forme du 'e' peut également être utilisée pour commencer la conception du 'c'.
- 'h' — tandis que 'h' peut être construit assez rapidement à partir du 'n', il donne également de la variété à la gamme que vous souhaitez tester en offrant une ascendante.
- 'i' — comme le 'e', la lettre 'i' est assez courante et a l'avantage de vous proposer un peu la forme du 'j'. La forme du 'i' est aussi partiellement dérivée de la forme du 'n'.
- 's' — la lettre 's' est un bon choix pour commencer à dessiner car elle ajoute une variété visuelle à la gamme des lettres que vous testez. La lettre 's' est également exceptionnellement difficile à réussir. Alors la commencer tôt, vous permettra de passer suffisamment de temps sur celle-ci pour qu'elle soit parfaite à la fin du projet. Les terminaisons du 's' peuvent parfois être utiles pour anticiper ce que pourraient être les terminaisons des 'a', 'c', 'f', 'j' et 'y'.
- 'v' — la lettre 'v' est utile pour anticiper ce que peuvent être les 'y' et 'w'.

Une fois que vous avez ces lettres, il est bon de passer du temps à les affiner en testant des mots qui sont faits à partir d'elles. Comme précédemment avec le 'n' et 'o', il faut accorder une grande attention à l'espacement des lettres et aux relations des contrepoinçons avec ces espaces.

Voici un fil de discussion de TypeDrawers pour déterminer les hauteurs d'ascendantes et de descendantes:

<http://typedrawers.com/discussion/1620/ascender-descenders-in-latin-type-design>

## Création d'un texte d'essai

Il existe de nombreuses ressources disponibles en ligne pour construire rapidement votre texte de test fictif:

- [LibreText](#) est une solution de logiciel libre.
- [Adhesion Text](#), réalisé par Miguel Sousa était la première ressource de ce genre.
- [JAF Generator](#), par Just Another Type Foundry.
- [Typable](#), par Ondrej Job

## Utiliser du texte réel

Ces *textes fictifs* sont en quelque sorte des textes "à l'aveugle"; vous ne pouvez pas vraiment les lire.

Mais une fois que vous avez une douzaine de lettres, vous pouvez construire du texte **réel** pour expérimenter une *lecture immersive*. Ceci est essentiel pour une compréhension approfondie de la façon dont le design se déroule. Il est donc bon d'arriver à cette étape rapidement.

Mais attention à l'effet de **texte surutilisé**. Si vous réutilisez le même texte réel pour tester votre police de caractères, il peut devenir si familier que vous perdez certaines capacités de perception afin de voir comment la police de caractères exerce ses fonctions.

Une fois que vous avez un alphabet complet, afin de s'assurer de texte frais, vous pouvez définir votre police de système d'exploitation, d'application de courrier électronique ou la police par défaut du navigateur Web comme étant votre police.

Ceci a été discuté sur le forum Type Drawers dans le fil [sur l'utilisation des textes à l'aveugle](#).

# Lettres capitales

Faire les lettres capitales devrait suivre un modèle très semblable à la conception des lettres minuscules. Vous commencez par concevoir des lettres-clés dont les formes et les caractéristiques se prêtent à la conception de caractères qui partagent une forme commune. Tout comme pour les lettres minuscules, la fréquence d'utilisation des lettres reste également un facteur important dans le choix des lettres.

Les deux premières lettres à dessiner sont "H" et "O". La conception de ces lettres ne doit pas être seulement en relation avec les unes avec les autres, mais devrait également se rapporter aux lettres minuscules existantes.

C'est à ce stade que vous déterminez la proportion de la minuscule à la capitale. Vous voudrez peut-être ajuster les ascendantes et descendantes de votre minuscule ou ajuster vos capitales à la minuscule pour créer la proportion qui correspondent au but de votre conception.

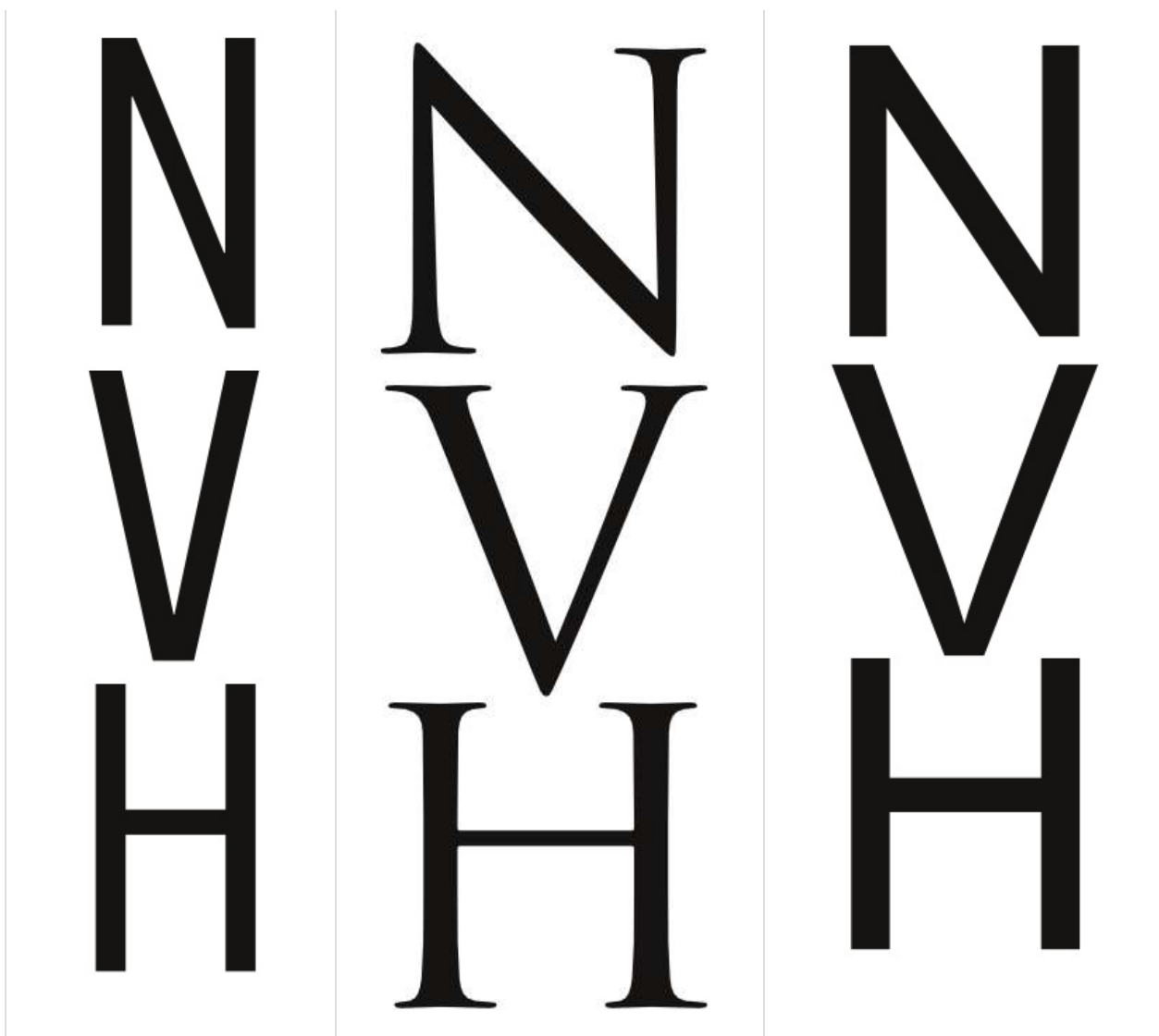
La graisse des traits dans le cas des majuscules doit souvent être un peu plus lourde que les traits de minuscules. Vous devrez peut-être expérimenter par interpolation pour trouver rapidement combien plus lourde elles devraient être.

Le prochain jeu de lettres à envisager d'ajouter est A, E, S, I, N et soit P ou D et peut-être V.

Selon le style de la police que vous faites, vous pouvez constater que les lettres capitales exigent plus de variation de largeur qu'il y a dans les lettres minuscules. La largeur des E S et P peut être sensiblement plus étroite que celle du H ou peut être similaire.

Généralement, les N et les V sont généralement semblables à H, mais légèrement plus larges.

--	--	--



Le D peut être semblable à H ou un peu plus large.



La forme du O peut vous en dire beaucoup sur les C, G et Q. La forme du H vous en dit un peu plus sur le I et le J et sur le côté gauche des B D E F K L P R. Il vous en apprend aussi un peu sur le T et le U. La forme du A peut vous en dire beaucoup sur la forme du V.

--	--	--

O O C C G G O O

O O C C G G O O

O O C C G G Q Q



La forme et les proportions du V vous indiquent un peu comment concevoir les Y W X. La forme du Z est distinctive.



V W Y X

V W Y X

V W Y X

V W Y X

# Interligne

Lorsque vous avez l'inter-mots et un jeu de n et o, vous pouvez commencer à regarder l'interligne. Toutefois, une décision complète et définitive sur l'interligne est impossible sans avoir les lettres capitales et de la ponctuation.

## Pensez à l'interligne intentionnellement

Comme c'est le cas avec l'espacement des lettres et des mots, avoir trop ou trop peu d'interligne peut rendre votre police difficile à utiliser dans le monde réel. Par-dessus tout, trouver le juste équilibre entre les lignes est une question nécessitant réflexion et le test d'une gamme d'options avant de prendre une décision finale.

Règle générale, la plupart des nouveaux concepteurs de police ont tendance à mettre trop peu d'interligne dans leur police. Donc si vous n'êtes pas sûr, ajouter de l'espace supplémentaire est généralement une bonne idée.

Vous devriez également considérer la portée linguistique de votre projet lorsque vous considérez l'interligne. Si vous tester l'interligne de votre police uniquement avec des caractères non accentués, vous êtes susceptible de choisir une valeur d'interligne qui ne laisse aucune place pour les accents. Si vous êtes certain que votre police ne sera jamais utilisée avec des caractères accentués, cela pourrait être acceptable — mais les chances sont que votre police sera utilisée pour composer un texte accentué. Dans ce cas, un interligne trop petite entraînera un empiètement des accents sur le bas des glyphes de la ligne supérieure, et laissera au lecteur la tâche difficile (voire impossible) de lire le texte.

Une stratégie pour tester si l'interligne de votre police est appropriée pour les caractères accentués consiste à utiliser des exemples de texte à partir de plusieurs langues.

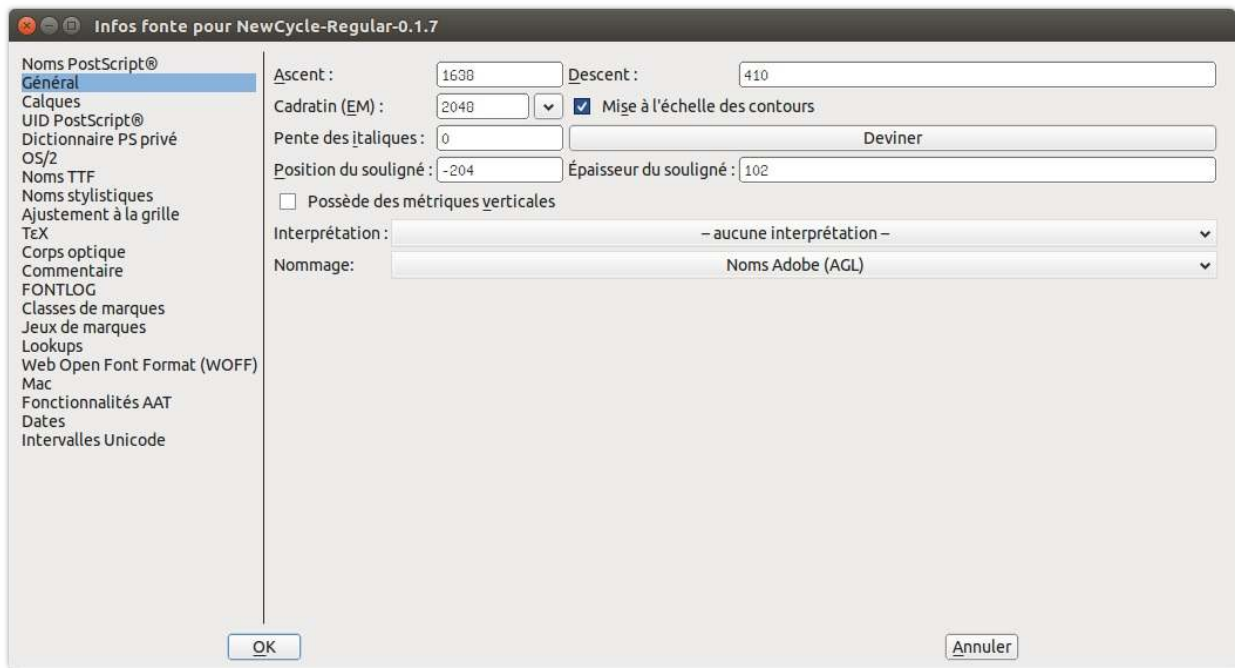
**Je to fotka, která nadchne všechny sportovní nadšence. Je to fotka, kterou můžete mít doma! Stačí se zapojit do charitativní dražby, jejíž výtěžek získá projekt Blesk Srdce pro děti.**

**It's a picture that inspires all sports enthusiasts. It's a picture that you can take home! Just get involved in a charity auction, the proceeds of which the project will flash heart for children.**

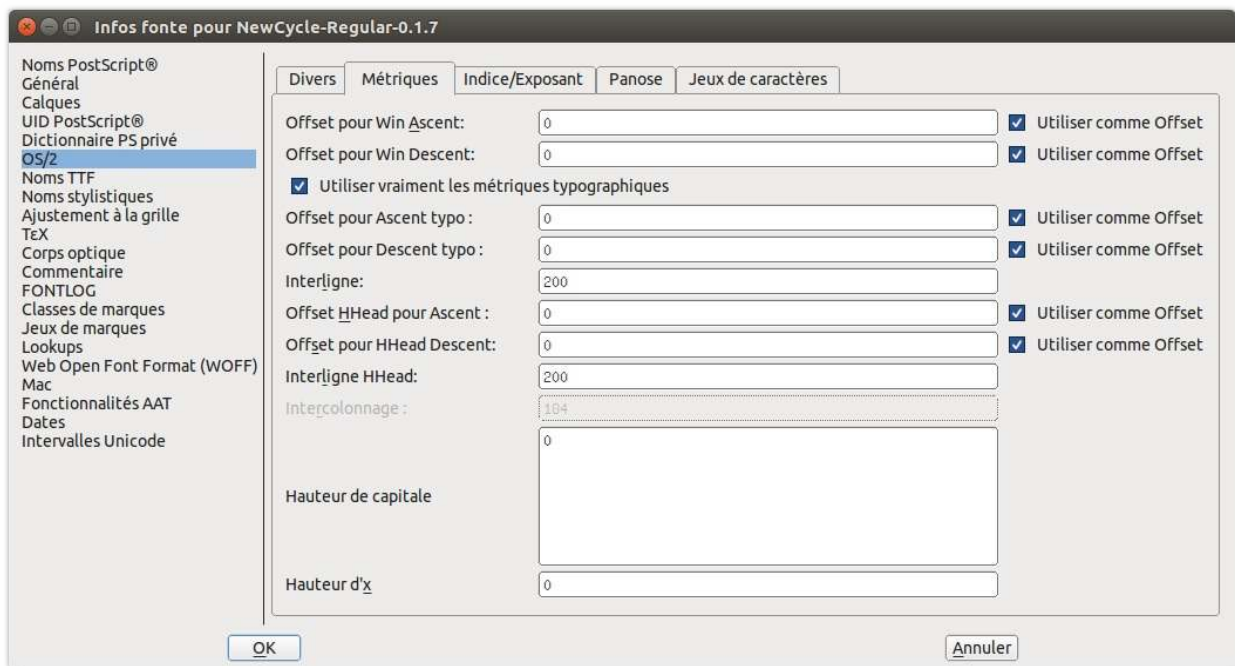
Pour les langues chargées de signes diacritiques (comme le tchèque), l'interligne doit être plus grande que pour les langues qui n'utilisent pas de signes diacritiques. Les exemples ci-dessus montrent le tchèque (ci-dessus) et l'anglais avec le même interligne assez large.

## Expérimentez avec l'interligne de votre police dans FontForge

Dans FontForge, vous pouvez définir et ajuster l'interligne de votre projet de polices à partir de la fenêtre Infos fonte. Ouvrez cette fenêtre en choisissant *Infos fonte* dans le menu "Élément", puis cliquez sur l'onglet Général. Notez les valeurs que FontForge a répertoriées pour Ascent et Descent. À moins que vous ayez déjà apporté des modifications manuelles, ces deux nombres, lorsqu'ils sont additionnés, devraient être égaux à la valeur de la taille en cadratin (Em) indiquée sur la ligne en dessous.



Passez maintenant à l'onglet "OS/2". Sur presque tous les ordinateurs, l'interligne de votre police sera déterminée par les valeurs d'ascent et de descent que vous entrez dans cet onglet, sous l'en-tête Métriques.



Il existe trois jeux de valeurs: Win Ascent et Descent, ascent et descent typographique, et ascent et descent HHead. Vous devez définir tous les ascents comme étant identiques à la valeur de l'ascent que vous avez notée dans l'onglet Général. Ensuite, vous devez définir tous les descents comme étant identiques à la valeur de descent que vous avez notée dans l'onglet Général, à une exception près: vous devez rendre le descent typographique *négatif*. Laissez la valeur identique, mais mettez un signe moins devant. Enfin, décochez toutes les options "Utiliser comme offset".

Ces paramètres vous donneront un bon point de départ. Vous pouvez maintenant tester votre police avec cette interligne et effectuer des ajustements incrémentaux jusqu'à ce que vous obteniez un résultat agréable à l'œil.

Si vous trouvez que votre interligne est trop serrée et que vous ne voulez pas ou ne pouvez pas rendre les métriques verticales plus importantes, vous pouvez diminuer la taille des glyphes pour gagner plus d'espace pour l'interligne.



## Ponctuation et symboles

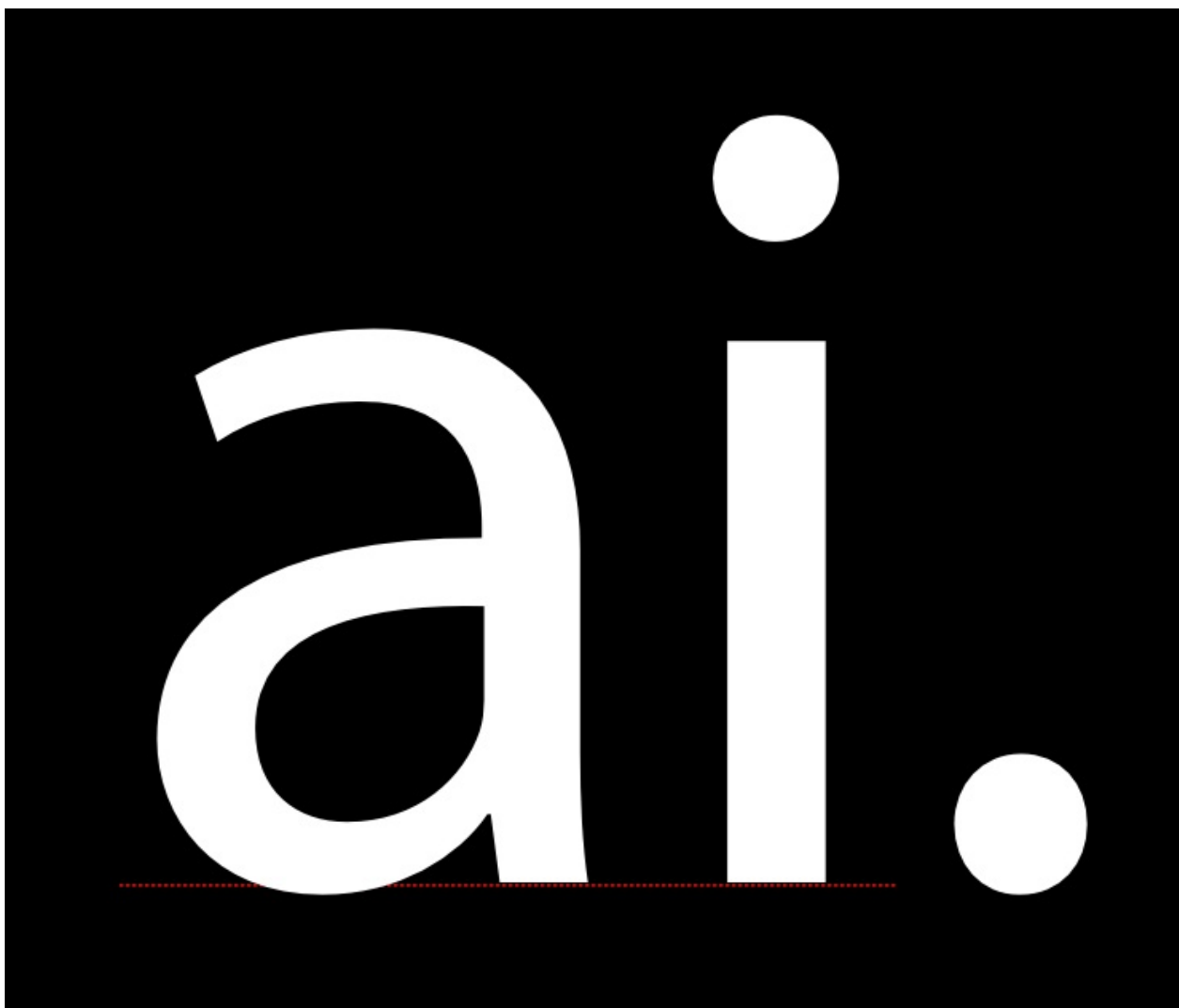
La ponctuation et autres symboles typographiques ont une histoire propre, distincte et à part du développement de l'alphabet. Mais vous constaterez que le même processus de conception s'applique toujours, y compris la réutilisation et l'adaptation de composants, et le test itératif de vos choix de design.

### Glyphes de ponctuation simples

La première chose à faire lors du design de la ponctuation est de créer le caractère '.', qui est connu comme le point.

La forme de ce glyphe est souvent prise du point sur le 'i,' qui est parfois appelé point suscrit. Après avoir copié le point, vous pouvez vouloir l'agrandir. Il est conseillé de tester plusieurs tailles de texte imprimé ou à l'écran.

Une fois que vous établissez une taille dont vous êtes satisfait, ce point peut être utilisé comme base pour une grande variété d'autres ponctuations, y compris ces glyphes: ; : ? ! ; ; · ...



Le prochain glyphe à faire est la virgule. La forme de la virgule peut varier de façon surprenante. Il peut être utile de regarder de nombreux designs de virgule avant de concevoir le vôtre.

L'image ci-dessous montre deux des formes les plus courantes que la virgule peut prendre.



Le sommet de la virgule est souvent légèrement plus adouci que celui du point, car si il est identique, celle-ci peut paraître trop lourde. Dans l'image d'exemple, la virgule à droite est un bon exemple où cette compensation est appliquée. Une autre erreur commune à surveiller avec ce glyphe est de le faire trop court.

Quand vous avez votre virgule, il est assez facile de faire le point-virgule (;).

## Point d'exclamation et point d'interrogation

Le point d'exclamation peut être trompeur en ce qu'il *semble* simple à faire. Si vous regardez une gamme de caractères, vous verrez que parfois le design est en effet assez simple.

Toutefois, il s'agit d'un glyphe qui a une étonnante occasion de s'exprimer dans le design. Il arrive souvent que même dans une police qui a très peu de contraste, la barre au-dessus du point est un peu plus lourde au sommet qu'à la base. La forme du point d'exclamation se rapporte généralement à la conception de la virgule dans une certaine mesure.



Le point d'interrogation peut également être assez difficile à faire, car il vous oblige à équilibrer une courbe ouverte sur le point en dessous.

Comme avec le point d'exclamation, il est conseillé de regarder et même de tester une gamme de solutions différentes avant d'en choisir une pour votre design.



Le design des glyphes c, C, G, s et S peut fournir une certaine base pour la conception de ce glyphe, mais vous pouvez décider de choisir une forme qui est distincte aussi.

## Symboles additionnels

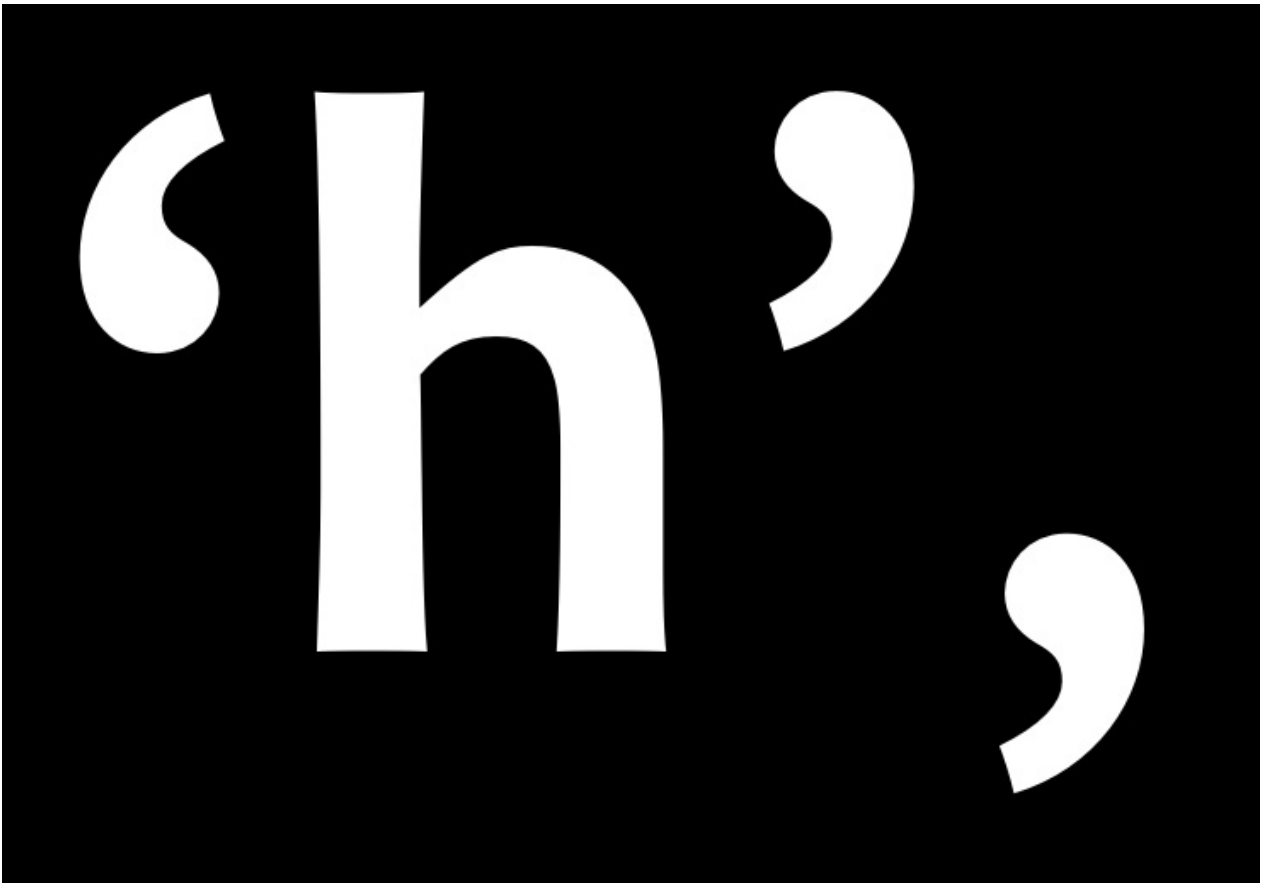


Les guillemets et apostrophes dactylographique — ' et " — sont distincts des guillemets typographiques: ‘ ’ et “ ”, „ .

Les apostrophes peuvent suivre la forme de la barre au-dessus du point dans le point d'exclamation, mais ils peuvent également être conçus séparément.

Généralement, les guillemets typographiques sont étroitement liés à la virgule, mais ils doivent être plus longs que la virgule et sont souvent plus courbés.





Les crochets [ ] sont relativement simples à faire car ils sont de forme carrée. Néanmoins, leur conception doit refléter les choix que vous avez faits dans le reste de la police.

La question principale à résoudre est comment grands et profonds ils seront. La convention est qu'ils doivent dépasser la hauteur des capitales très légèrement et descendre au-dessous de la ligne de base à environ 3/4 de la profondeur de vos descendantes de minuscules.

Ces choix seront également reflétés dans le design des parenthèses ( ) et des accolades { }. La graisse des fûts sur ces trois symboles doit être inférieure à la graisse des fûts des capitales et des lettres minuscules.

Soyez averti: lorsque vous testez les caractères [ ] # dans la fenêtre de métriques, ils peuvent ne pas s'afficher. C'est parce qu'ils sont réservés par le programme. Au lieu de taper [ ] et #, vous devez taper `/bracketleft /bracketright` et `/numbersign`.



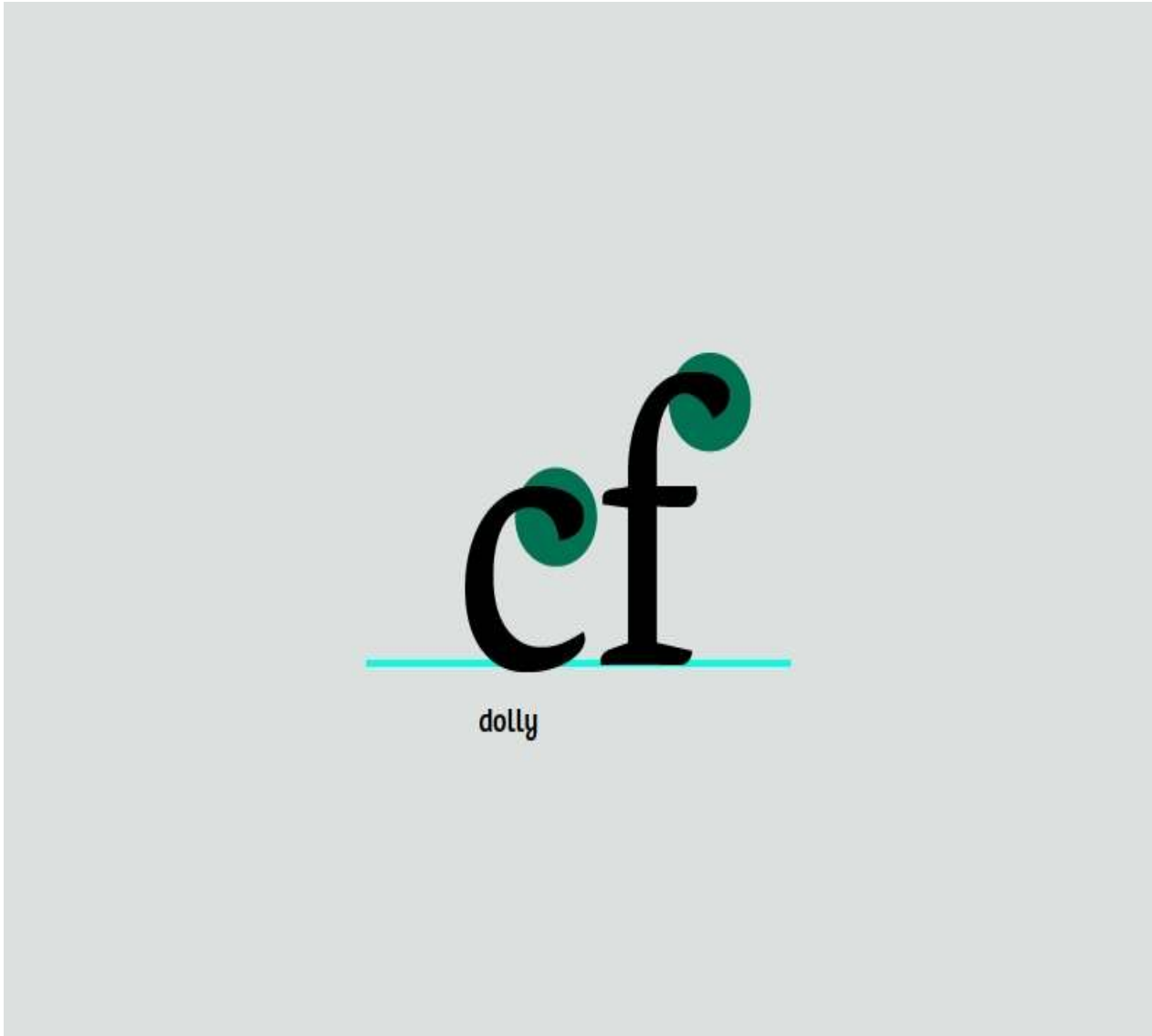
Les parenthèses devraient s'appuyer sur le design des formes connexes, telles que le D, le C et le G.

Les accolades sont remarquables par leur variation de design. Les accolades ont ceci en commun avec le point d'interrogation. La répartition de la graisse dans les accolades peut être comme la répartition de la graisse des nombres, dans la mesure où cela peut parfois violer les règles que vous suivez dans le reste du design.

## Finalisation des minuscules

Vous avez peut-être remarqué dans des polices que, tandis que chaque lettre a sa propre forme, elles se rapportent toutes les unes aux autres. En déconstruisant soigneusement quelques glyphes, vous obtenez les blocs de construction de presque tous les autres.

Noter la similitude entre les terminaisons supérieures sur ce c et ce f :

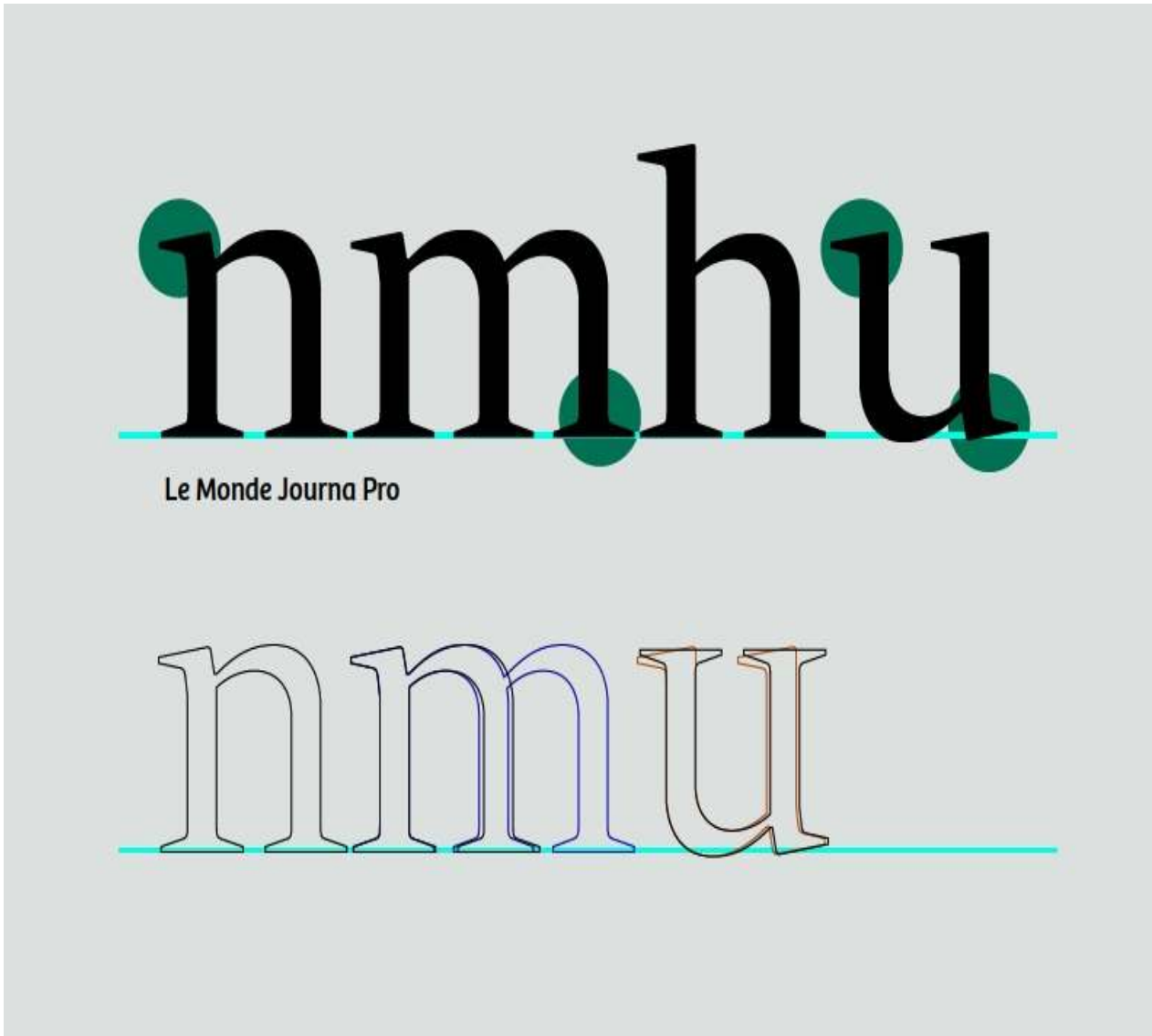


Leurs formes indiquent qu'elles appartiennent au même groupe, même si elles sont subtilement différentes. Les terminaisons sont l'un des traits d'identification d'une police, et sont généralement répétées sur de nombreuses formes de lettres.

Cependant, une dépendance excessive à la modularité laisse des marques dans un design, et donc devrait être évitée — sauf si c'est le look que vous cherchez.

## Travailler avec les autres lettres minuscules

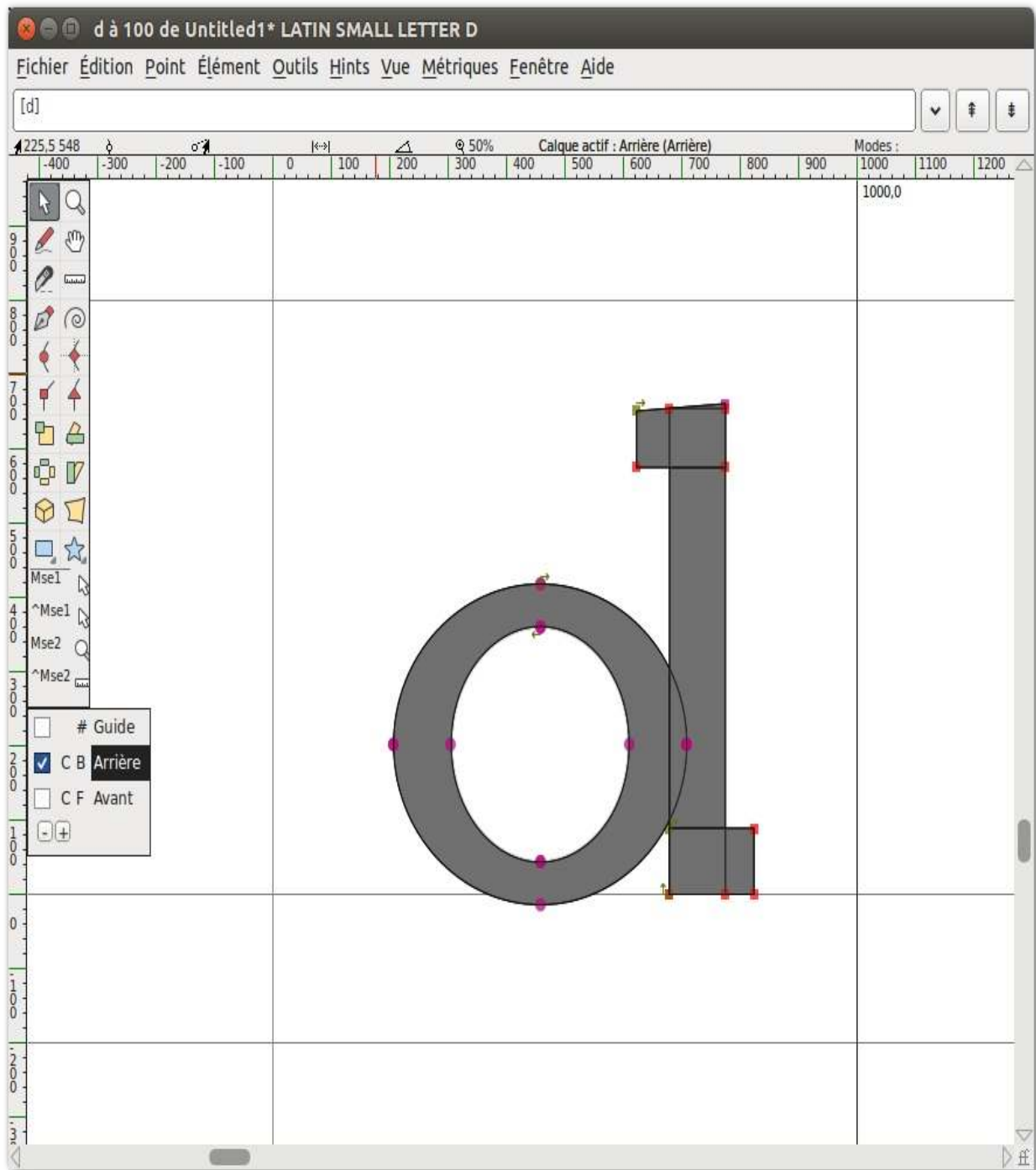
Vous avez déjà fait votre lettre 'n.' De ce fait, vous pouvez facilement dériver les m, h et u par clonage, étirement et rotation. Il ya des changements subtils dans l'espacement des fûts dans le m et le u. Le 'u' a changé non seulement son espacement mais ses empattements. Cela ne se produit pas automatiquement; c'est à vous d'y aller et de déplacer les points.



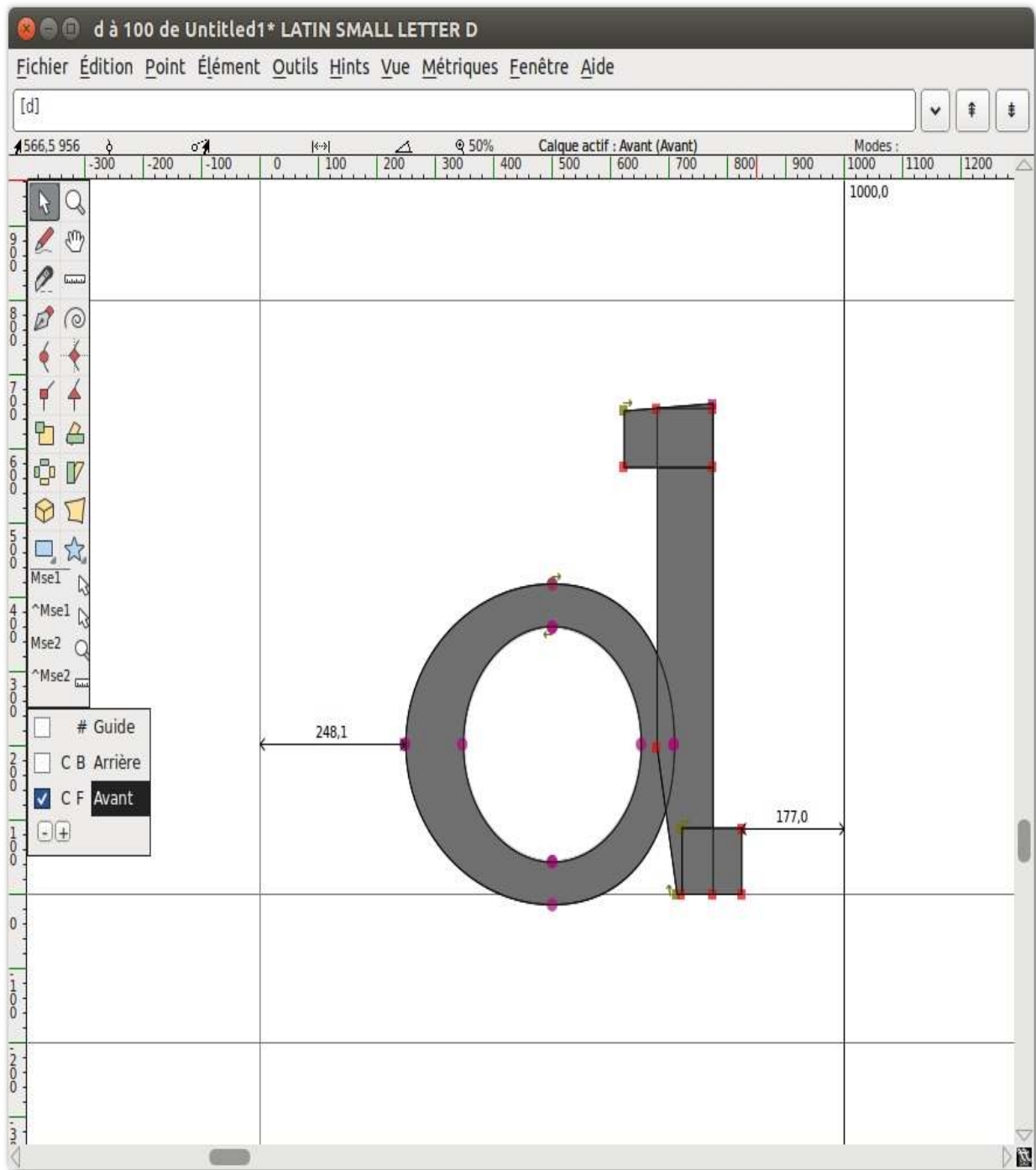
Le 'i' peut être dérivé du fût du 'n.' Le 'l' peut être fait à partir du fût du 'n' avec quelques ajustements.

### Faire le d à partir du fût du h et du o

Ouvrez la fenêtre de glyphe de la lettre 'd' en double-cliquant en dessous de 'd' dans la fenêtre de fonte. À partir de la fenêtre de fonte, copiez le 'o' et le coller dans la fenêtre de glyphe de la lettre 'd'. Ensuite, faites la même chose pour le 'h'. À ce stade, vous pouvez supprimer la partie du h que vous n'allez pas utiliser. Positionner les pièces restantes ensemble afin que cela ressemble à un 'd'.

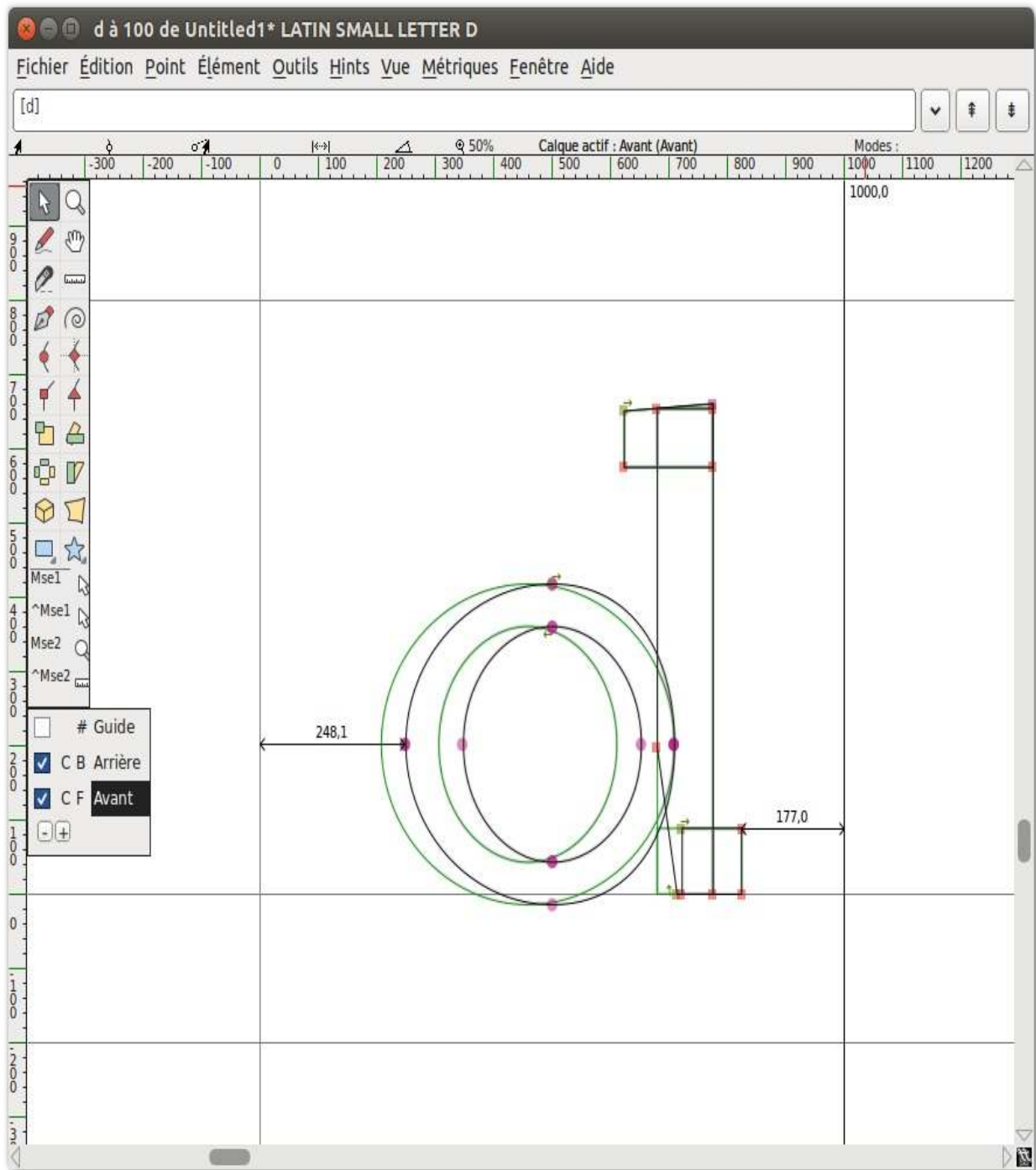


De toute évidence, il y a encore du travail à faire ici. Nous allons faire quelques ajustements. Affinez le côté droit du o où il rencontre le fût.



Pour améliorer l'espacement optique et permettre à la forme de paraître plus équilibré, faites un peu de place à l'empattement en ajoutant un point au fût et en déplaçant les points du bas vers la droite.

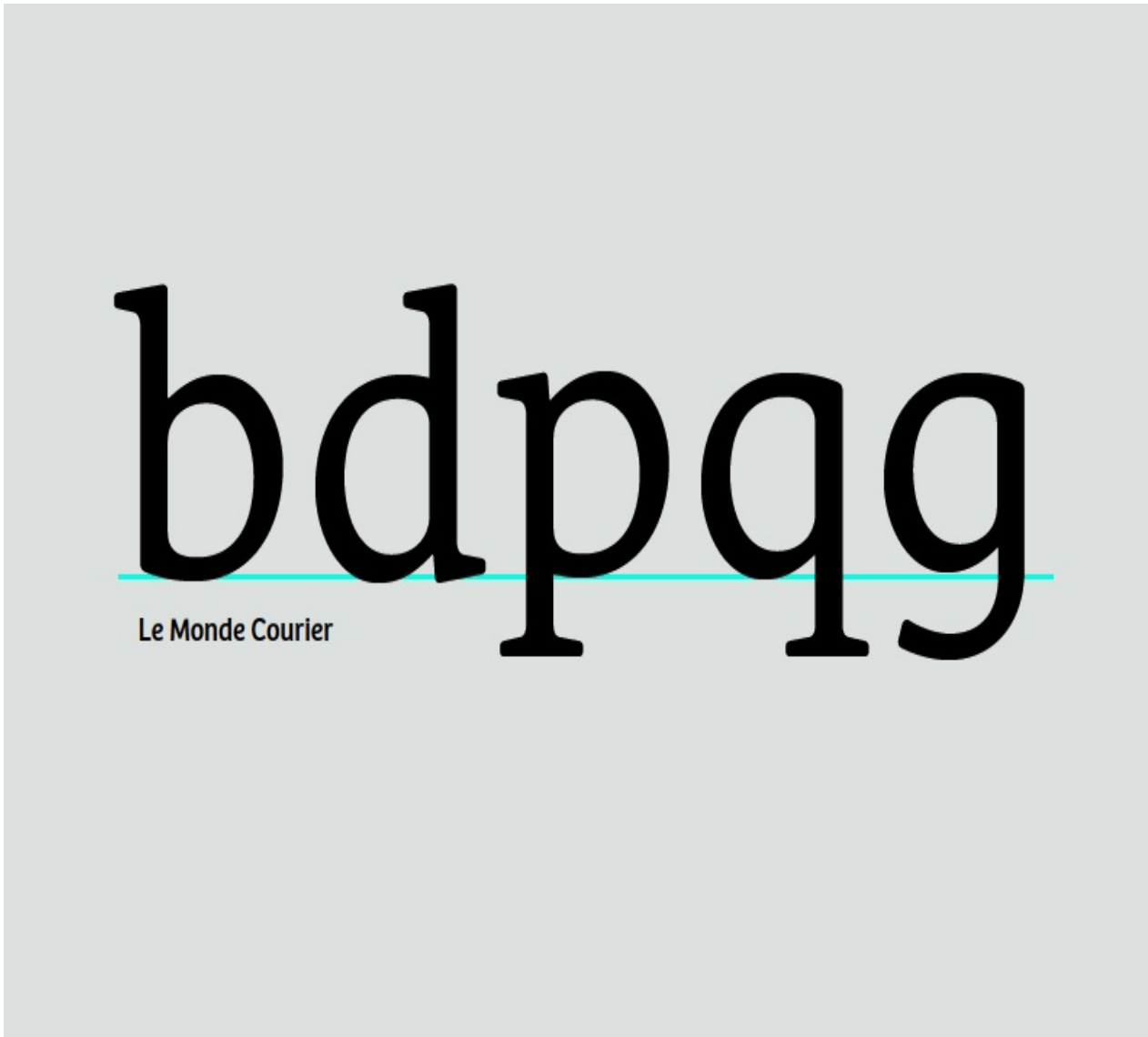
Ci-dessous, vous voyez une superposition de la forme de départ et de la nouvelle forme.



Maintenant que vous savez comment assembler à partir de pièces existantes, vous pouvez faire d'autres lettres semblables. Gardez à l'esprit les subtilités qui font chaque lettre individuelle, mais qui sont partie d'une famille.

## Dériver le b, le p et le q

Maintenant que vous avez le d, en l'inversant et en le tournant vous pouvez faire de raisonnables b, p et q. Encore une fois, être conscient de la façon dont les empattements et le contraste diffèrent dans chaque lettre. Votre police ne doit pas être faite exactement de la même manière, mais c'est une des choses que vous devriez penser.



## Faire le g

Vous pouvez commencer par le q, l'étirement et la modification de la queue, pour faire un g à panse unique. Aucune forme ne ressemble au g binoculaire. Le g binoculaire doit habituellement être nettement plus léger afin de paraître correct quand il est utilisé avec d'autres lettres.





Passer au f et au t

Le t a une ascendante, mais elle est généralement plus court que les ascendantes des autres lettres minuscules. Par comparaison, le f est beaucoup plus grand et empiète habituellement sur l'espace de la lettre à côté d'elle. Ils ont tous deux des traverses qui sont généralement à la même hauteur, largeur et épaisseur. Souvent, vous pouvez les copier de l'un à l'autre.

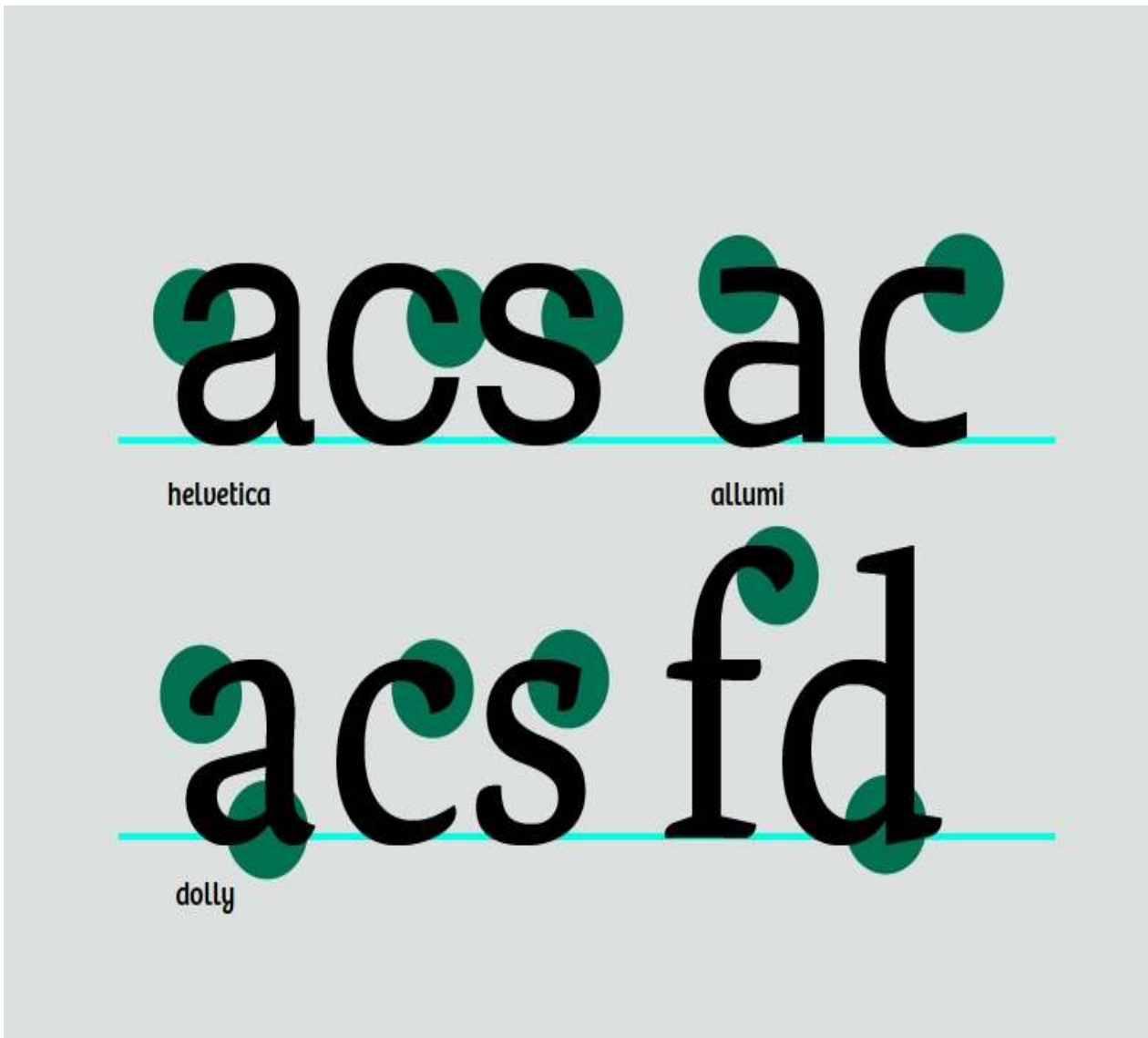


### Maintenant faire le e

Le e sera librement basé sur le o. La traverse du e est inférieure à celle du t, mais a la même épaisseur. Le crochet au bas du e sera inspiré par le bas du t.

### Du e vient le c

La création de c à partir de e implique la suppression de la traverse et l'ajout d'une terminaison en haut. La terminaison supérieure du c peut être semblable aux terminaisons supérieures d'autres lettres telles que le a, le f et le r. Les terminaisons du c peuvent également constituer la base du s. Le e peut également influencer sur les proportions de votre a.



### **v, w, x, y, et z**

Ces lettres sont un peu difficiles parce qu'elles n'ont pas de formes qui sont liées aux autres lettres. Cela signifie que vous devez juste vous y atteler et dessiner le v. Faites le trait descendant aussi épais que vos fûts épais, et faites le trait ascendant aussi mince que les traits les plus minces dans vos autres lettres. Une fois que vous avez le v, vous avez un plan de base pour le w et le y. Pour le x et le y, concentrez-vous sur faire correspondre le contraste au reste du design tout en compensant les illusions qui se produisent dans les formes diagonales et en croisant les diagonales.

## Diacritiques et accents

Un diacritique est un signe (ou une marque selon la terminologie anglaise) ajouté ou combiné à une lettre, souvent utilisé pour modifier la valeur sonore de la lettre à laquelle le signe est ajouté. Certaines signes diacritiques (telles que «aigu» et «grave») sont souvent appelées accents. Les signes diacritiques peuvent apparaître au-dessus ou au-dessous d'une lettre, en son sein ou entre deux lettres.

à â ã ä å ç ċ ġ ö

### Quelques exemples de diacritiques

à

'a accent grave' minuscule (unicode u + 00e0). Créé dans une police en combinant le glyphe 'a' minuscule (unicode u+0061) et le glyphe 'diacritique accent grave' (unicode u+0300).

â

'a accent circonflexe' minuscule (unicode u+00e2). Créé dans une police en combinant le glyphe 'a' minuscule (unicode u+0061) et le glyphe 'diacritique accent circonflexe' (unicode u+0302).



'a ogonek' minuscule (unicode u+0105). Créé dans une police en combinant le glyphe 'a' minuscule (unicode u+0061) et le glyphe 'diacritique ogonek' (unicode u+0328).



'c cédille' minuscule (unicode u+00e7). Créé dans une police en combinant le glyphe 'c' minuscule (unicode u+0063) et le glyphe 'diacritique cédille' (unicode u+0327).



'o double accent aigu' minuscule (unicode u+0151). Créé dans une police en combinant le glyphe 'o' minuscule (unicode u+006f) et le glyphe 'diacritique double accent aigu' (unicode u+030b).

---

FontForge peut automatiquement créer des caractères accentués de 2 façons;

1. FontForge contient des informations rudimentaires sur l'endroit où placer les signes diacritiques, donc peut construire

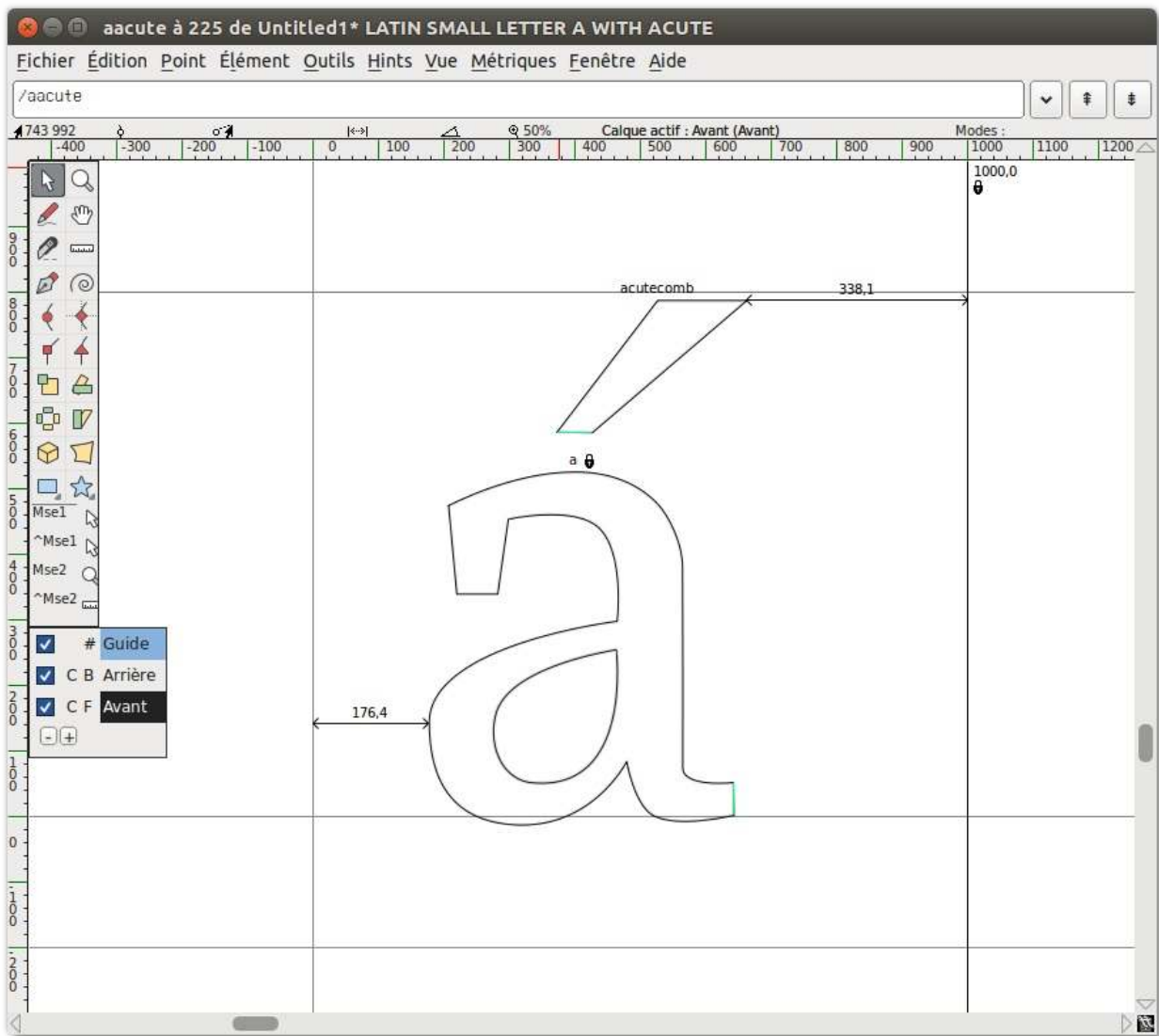
automatiquement la plupart des caractères accentués.

2. Pour plus de contrôle du placement diacritique, FontForge peut placer des signes diacritiques en fonction de la position des points d'ancrage créés par l'utilisateur.

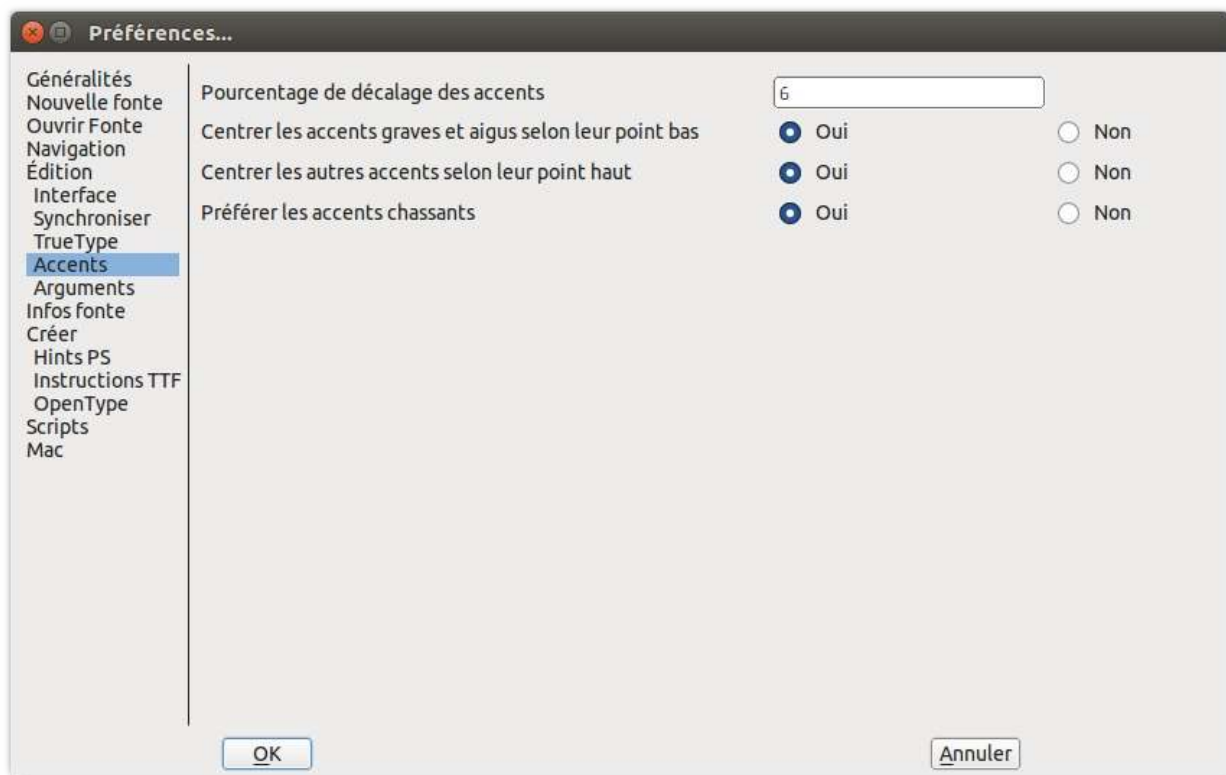
Notez que si vous n'utilisez pas les ancrages et les tables de lookup pour positionner les signes diacritiques, alors si le glyphe d'un signe diacritique particulier n'est pas présent dans votre police, FontForge utilisera plutôt un accent chassant similaire en place. Par exemple, si le signe diacritique 'accent aigu combiné' (u + 0301) n'est pas présent, FontForge utilisera le caractère 'accent aigu' (u+00b4) standard lorsqu'il construira automatiquement des glyphes accentués aigus. Si l'accent aigu combiné est présent, alors FontForge l'utilisera toujours, sauf si vous forcez spécifiquement FontForge à utiliser des accents chassants pour construire des glyphes accentués.</P>

## Positionnement automatique de base des signes diacritiques dans FontForge.

Dans le menu 'Elément' de FontForge, il y a une fonction appelée 'Assembler' qui peut être utilisée pour créer des caractères accentués, certains caractères composites et certains caractères en double. Pour construire automatiquement des caractères accentués FontForge utilise la fonction 'Elément> Assembler> Créer les glyphes accentués'. Cette fonction peut également être effectuée avec la touche 'Ctrl + Maj + a'. Ainsi, à l'aide de l'exemple de construction du caractère 'a accent aigu' (u+00e1), il faudrait avoir déjà créé le 'a' minuscule (u+0061) et le glyphe 'accent aigu combiné' (u+301). Ensuite, en sélectionnant l'emplacement de caractères 'a accent aigu' et en utilisant la fonction 'Elément> Assembler> Créer les glyphes accentués', FontForge placera une référence au glyphe 'a' minuscule et une référence au glyphe 'accent aigu' dans l'emplacement de caractères 'a accent aigu' (voir ci-dessous).



Ce positionnement automatique des signes diacritiques peut être réglé selon les préférences, qui se trouvent dans la section 'Accents' du menu des préférences de FontForge 'Fichier > Préférences > Accents' (voir ci-dessous).



'Préférer les accents chassants' — l'activation de cette option obligera FontForge à créer des glyphes accentués avec des caractères chassant, même si les caractères combinants appropriés sont présents. Cette option est ignorée lors de l'utilisation d'ancres pour positionner les signes diacritiques.

'Pourcentage de décalage des accents' contrôle la quantité d'espace vertical entre le glyphe de base et celui du diacritique. La valeur saisie ici est un pourcentage du cadratin de la police. Donc, une valeur de '6' va décaler le glyphe du signe à partir du glyphe de base de 6% du cadratin de la police.

Les préférences pour le positionnement horizontal du glyphe de signe peuvent également être définies. En mettant à 'Oui' la préférence 'Centrer les accents graves et aigus selon leur point bas', le glyphe d'accent sera centré au point le plus bas du glyphe de base.

Si vous mettez à 'Oui' l'option 'Centrer les autres accents selon leur point haut', l'accent sera centré sur le point le plus élevé du glyphe de base.

En mettant les deux préférences ci-dessus à 'Non', l'accent sera centré sur la largeur du glyphe de base. Si vous mettez les deux préférences ci-dessus à 'Oui', l'accent sera centré sur la largeur de la case de caractère.

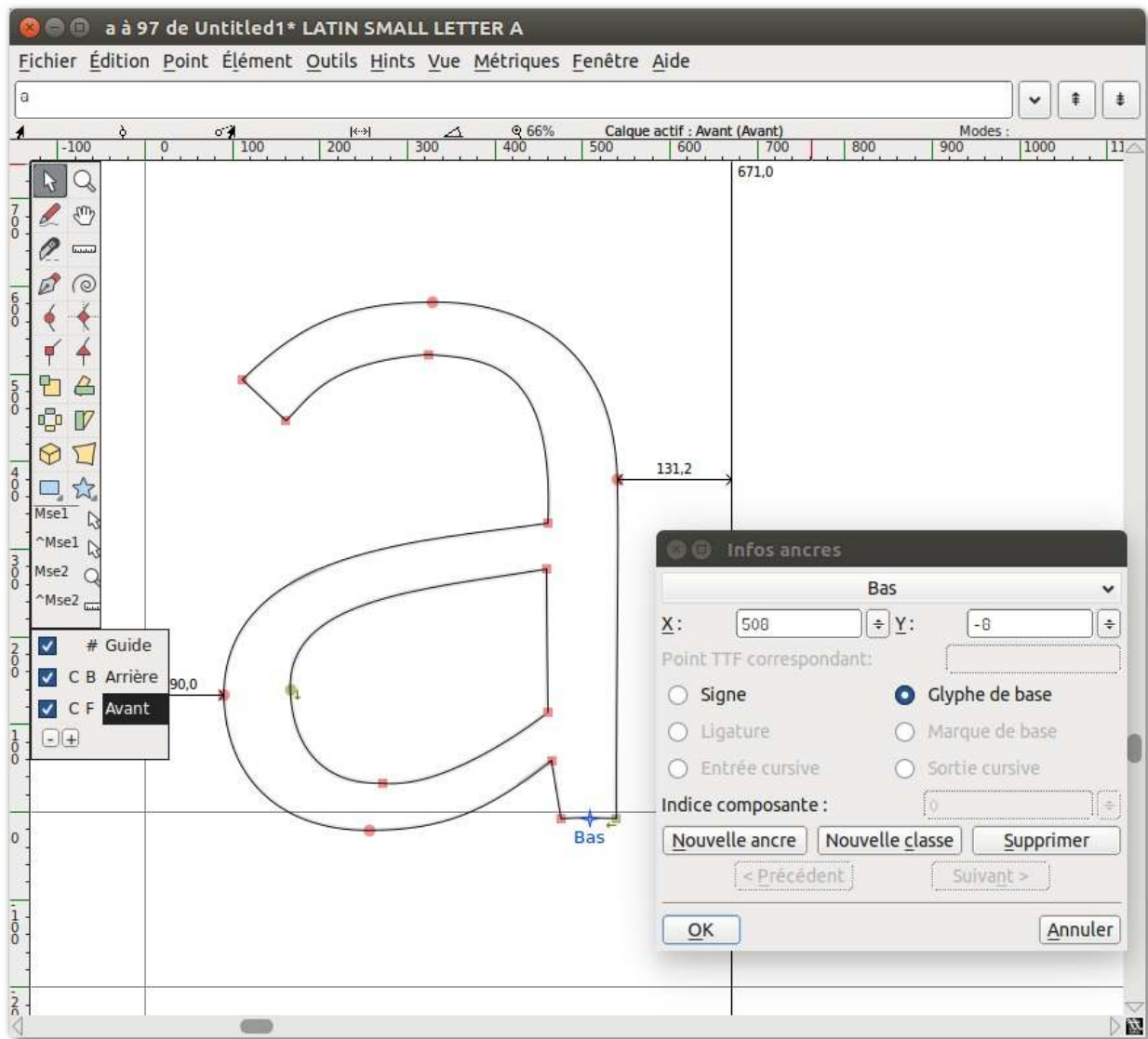
## Utilisation des points d'ancrage pour placer des signes diacritiques

La méthode la plus précise et la plus efficace pour créer des caractères accentués dans FontForge consiste à utiliser des points d'ancrage.

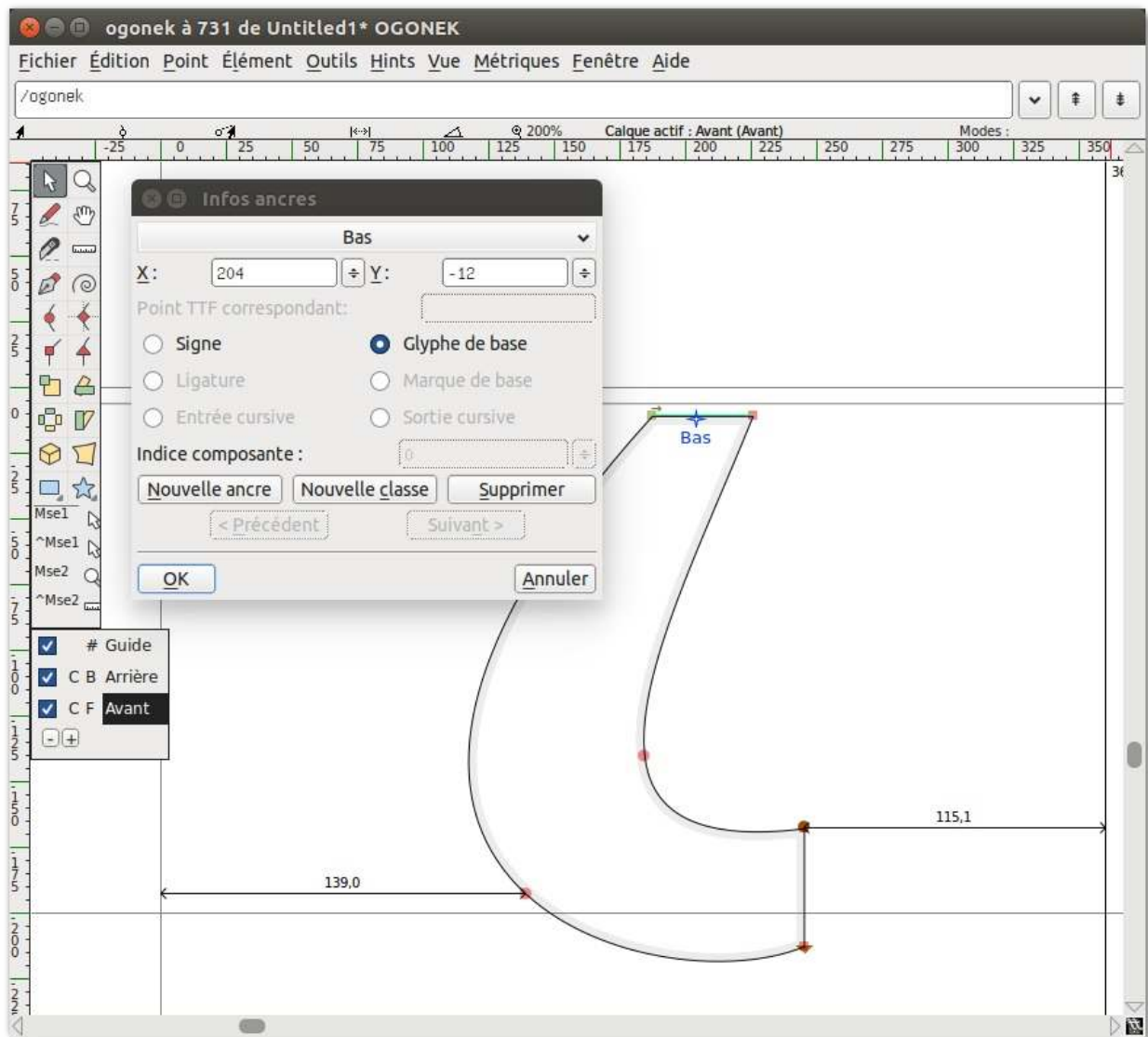
Les points d'ancrage permettent un contrôle précis du positionnement du signe diacritique par rapport à chaque glyphe de base dans les caractères accentués. Ainsi, dans le cas du caractère 'a ogonek', le glyphe 'a' est le glyphe de base et il sera placé normalement. Le glyphe 'ogonek' est le 'glyphe de signe' et sera positionné de sorte que le point d'ancrage du 'glyphe de signe' coïncide avec le point d'ancrage sur le glyphe de base.

Dans l'exemple ci-dessous, en créant un caractère 'a ogonek', une classe d'ancrage a été créée et appelée 'bas'. Dans le glyphe 'a' minuscule, l'ancrage du 'bas' est placé au bas du fût du 'a' sous la forme d'une ancre de 'glyphe de base'. (voir ci-dessous)

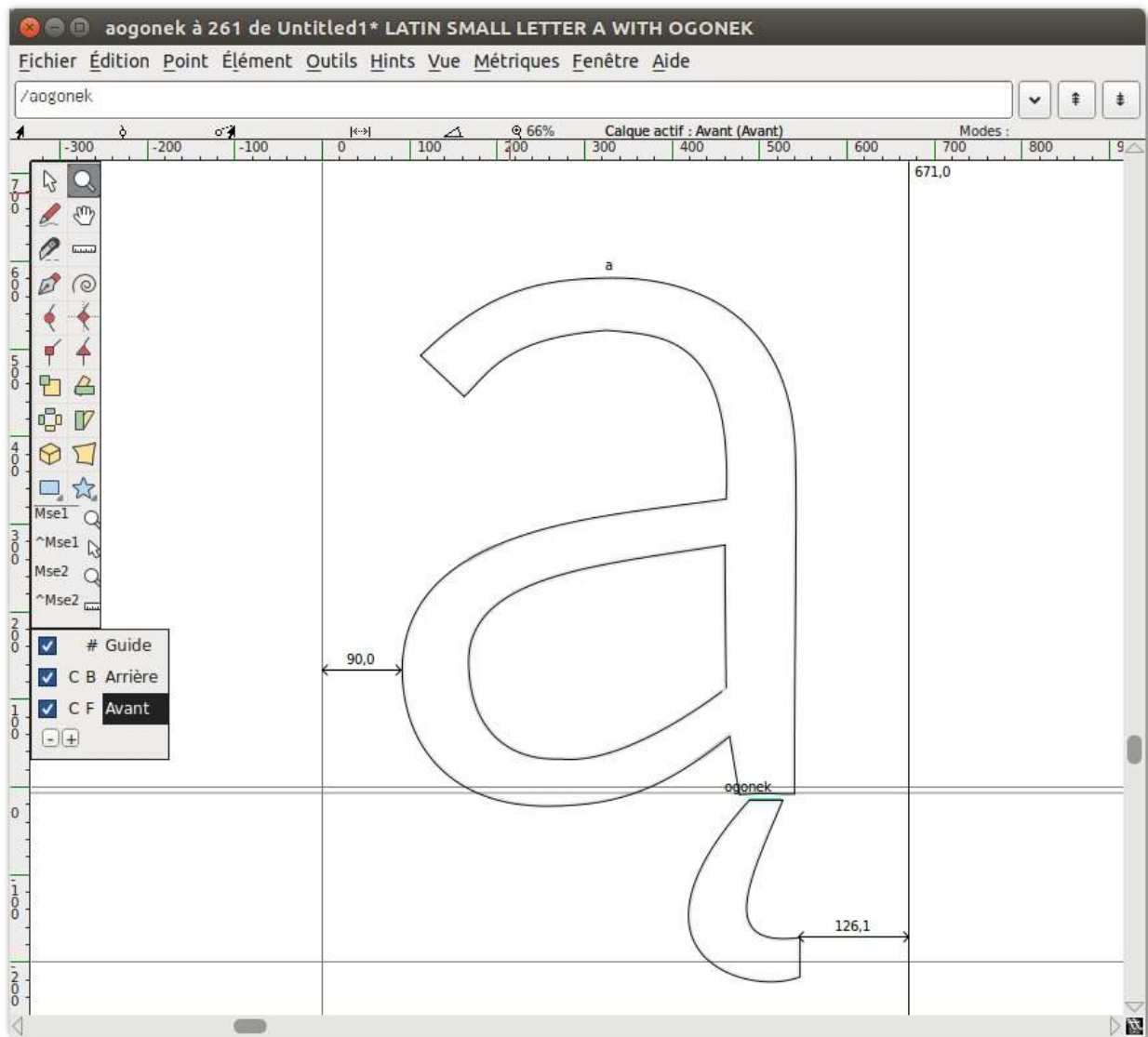




Dans le glyphe 'ogonek', l'ancre du bas est placée au sommet du glyphe ogonek, sous la forme d'une ancre de 'signe'. (voir ci-dessous)



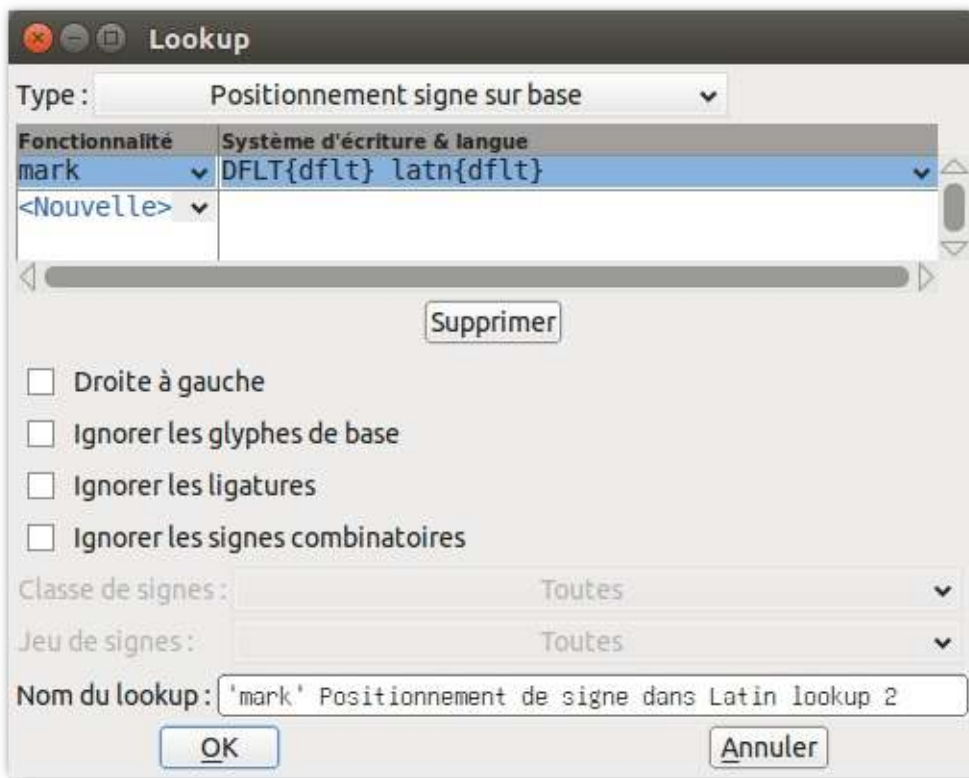
Ensuite, lorsque le caractère 'a ogonek' est construit (en utilisant la fonction 'Créer les glyphes accentués'), le point d'ancrage du signe inférieur sera placé au même emplacement que le point d'ancrage de base du 'bas', assurant que le glyphe ogonek indiqué est placé correctement au pied du fût du glyphe 'a' indiqué (voir ci-dessous). Ce placement exact et automatique n'aurait pas été possible sans utiliser de points d'ancrage pour positionner les glyphes de base et de signe.



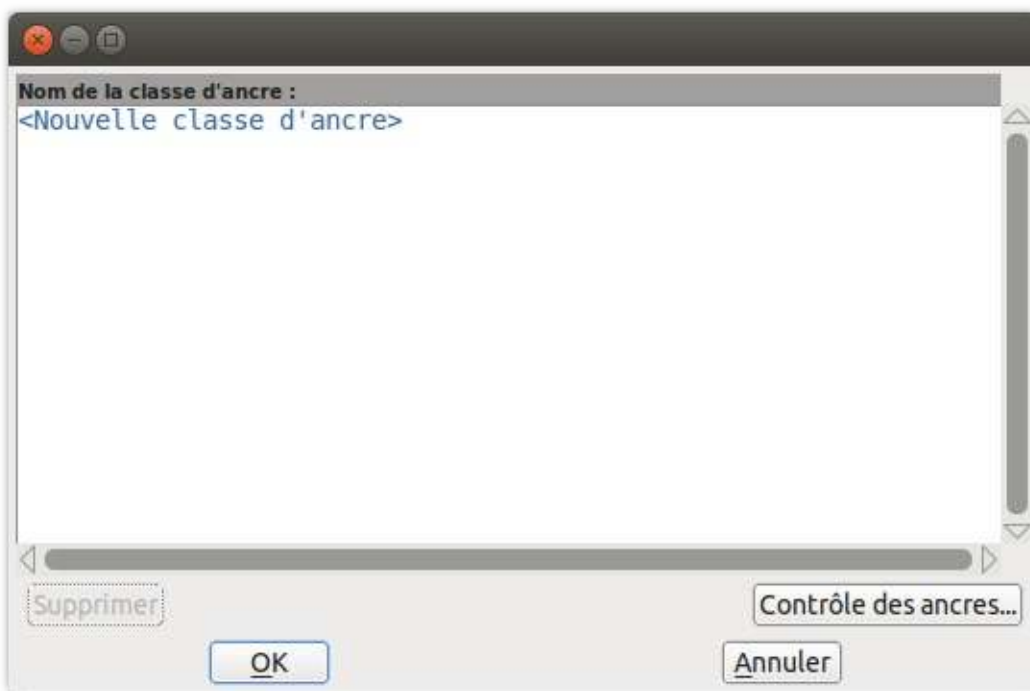
## Création de points d'ancrage pour le placement de signes diacritiques (positionnement signe sur base)

FontForge utilise des fonctionnalités de recherche (lookup) appelées 'signe sur base' pour créer et positionner des points d'ancrage. Ces lookups de signe sur base peuvent être créés et édités dans la section Lookup GPOS des informations sur la fonte de votre police ('Élément> Infos fonte> Lookups> GPOS').

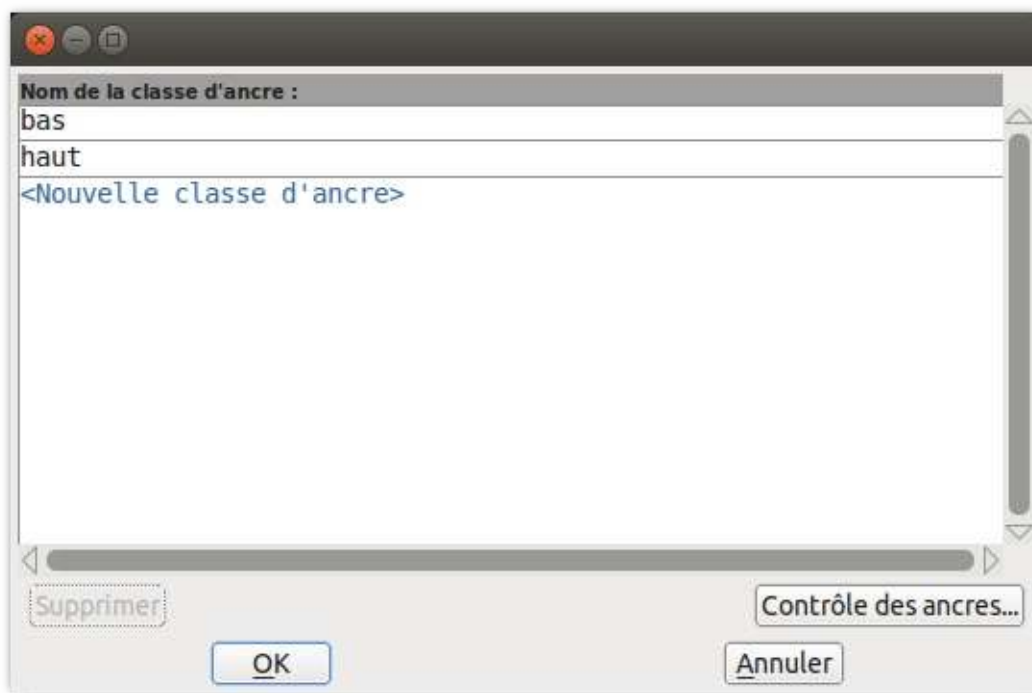
Dans la fenêtre Lookups GPOS, cliquez sur 'Nouveau lookup' et choisissez 'Positionnement signe sur base', puis choisissez 'Positionnement par signe' dans la colonne 'Nouveau' du volet Fonctionnalité (voir ci-dessous). Cliquez sur 'OK' pour fermer la fenêtre.



Lorsque le nouveau lookup est sélectionné, cliquez sur 'Nouvelle sous-table'. Dans la fenêtre résultante (voir ci-dessous), vous pouvez créer vos classes d'ancrage.



Dans cet exemple (ci-dessous), deux classes d'ancrage ont été créées, 'haut' et 'bas'. La classe d'ancrage du 'haut' sera utilisée pour positionner les signes diacritiques placés au-dessus des glyphes, et l'ancrage du 'bas' sera utilisée pour positionner les signes sous les glyphes.



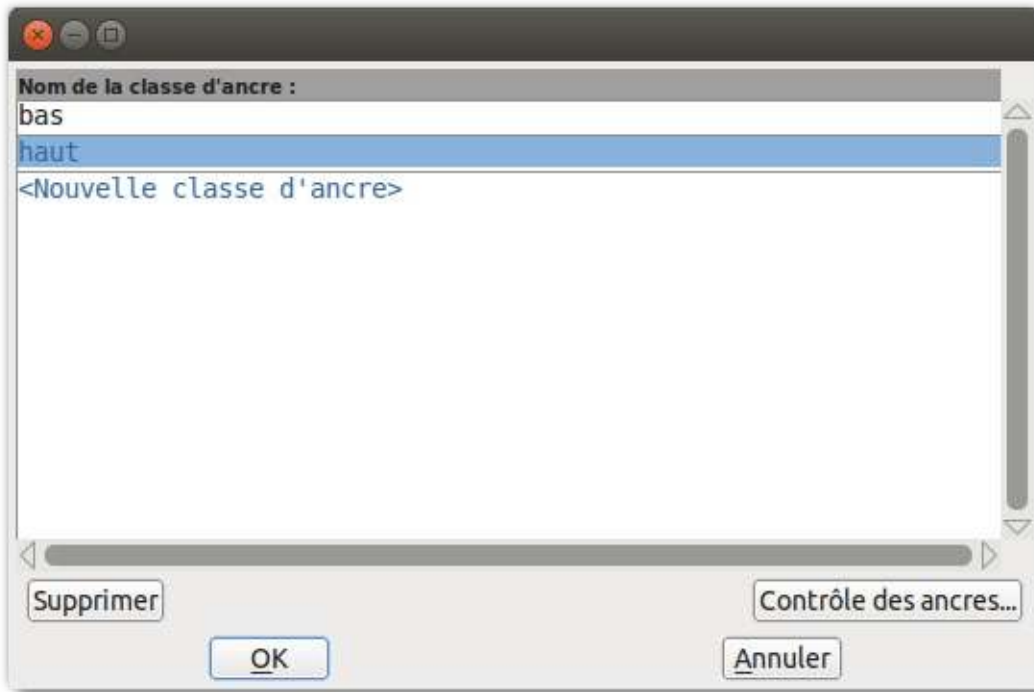
Pour placer une ancre avec un glyphe, il suffit d'utiliser le bouton droit de la souris dans une fenêtre d'édition de glyphe et de sélectionner la fonction 'Ajouter ancre' dans le menu contextuel.

La boîte de dialogue qui s'affiche vous permet d'indiquer si l'ancre est un ancrage sur base ou sur signe. La position de l'ancre peut également être affinée à partir de cette boîte de dialogue. Alternativement, l'ancre peut être déplacé en étant déplacé à la position avec la souris, ou déplacé en utilisant les touches haut, bas, gauche et droite. Le point d'ancrage peut également être modifié en cliquant avec le bouton droit de la souris sur le point d'ancrage et en choisissant «Infos...» dans le menu de la souris.

## Contrôle des classes d'ancrage

FontForge contient également une interface graphique utile pour contrôler la position de classes entières de points d'ancrage, permettant à l'utilisateur d'affiner la position de, par exemple, tous les accents aigus à la fois dans une police, ou tous les ancres dans une classe contenue dans, par exemple, les caractères qui font référence à la minuscule 'e'. Dans les exemples ci-dessous, nous pouvons voir comment utiliser cette interface graphique pour affiner la position de tous les accents aigus dans une police et pour affiner une classe d'ancre sur tous les caractères qui font référence au glyphe 'e' minuscule.

Une fois que vous avez créé des classes d'ancrage dans vos lookups de positionnement 'signe sur base' et ajouté des ancres à certains glyphes, vous pouvez contrôler ces classes à partir de "Élément>Infos fonte>Lookups> GPOS" puis modifier une sous-table contenant les classes d'ancrage. Vous verrez alors cette fenêtre;



À partir de là, sélectionnez la classe que vous souhaitez modifier et cliquez sur le bouton 'Contrôle des ancrés'. Une interface graphique de cette classe vous sera ensuite présentée. Dans les exemples ci-dessous, nous éditons le contrôle de la classe 'haut'. Dans le premier exemple (ci-dessous), la minuscule 'e' a été sélectionnée dans la section 'Bases' du menu déroulant. Lorsqu'un glyphe de base est sélectionné, tous les caractères qui font référence à ce glyphe et contiennent un ancrage 'haut' de base s'affichent dans le volet d'aperçu. Nous pouvons alors ajuster la position de l'ancre de base 'haut' pour voir comment elle affecte la position de tous les glyphes qui contiennent l'ancrage de signe 'haut'.



Dans le deuxième exemple ci-dessous, le glyphe 'aigu' a été sélectionné dans la section 'Signes' du menu déroulant. Lorsqu'un glyphe de signe est sélectionné, tous les glyphes qui font référence au glyphe sélectionné et contiennent un point d'ancrage 'haut' s'affichent dans l'aperçu.



## Autres ressources (en anglais)

- <http://urtd.net/projects/cod/about>
- <http://ilovetypography.com/2009/01/24/on-diacritics/>
- <http://diacritics.typo.cz/>
- <http://scripts.sil.org/ProbsOfDiacDesign>
- <http://www.microsoft.com/typography/developers/fdsspec/diacritics.htm>
- <https://twitter.com/fostertype/status/610292546971893760>

## Chiffres

Les chiffres sont souvent difficiles pour les concepteurs de polices — et pour plusieurs raisons. L'une est que les chiffres ont un très grand nombre de courbes. Une autre est que les chiffres utilisent souvent des conventions dans leurs formes qui sont différentes (ou sont même en violation de) des conventions visuelles vues dans le reste du design de la police. En outre, les chiffres peuvent avoir un très grand nombre de traits (comme 8 et 5 par exemple), ou ils peuvent avoir de grands espaces blancs (comme 1, 7, et parfois 2 et 4). Les deux situations peuvent être difficiles à gérer. Enfin, il ya le problème de s'assurer que votre zéro est différent de la lettre capitale O.

Il peut être utile de regarder les chiffres d'une grande variété de polices pour mieux connaître les façons dont les concepteurs font face à ces problèmes.

Dans ces chiffres avec un nombre dense de traits (comme 8), vous pouvez constater que les concepteurs permettent aux largeurs de trait d'être un peu plus minces que ceux des lettres dans la police. Une approche similaire peut être observée sur le design du g avec descendante en boucle.

À l'inverse, pour compenser les chiffres avec de grandes proportions d'espace blanc, certains traits sont susceptibles de devenir plus épais que normal.

Afin de distinguer entre le zéro et la lettre capitale O, il existe une large gamme de solutions — comme faire le zéro plus étroit que le O, ou un zéro qui est parfaitement rond, ou peut-être (surtout dans une police à chasse fixe) mettre un barre oblique sur le zéro.

Avoir le zéro plus étroit que le O capital tout en ayant la même hauteur est l'approche commune. Cette approche est typique des chiffres alignés. Les chiffres alignés sont ceux dont le style est le plus courant. Des exemples de polices qui utilisent cette approche comprennent: de nombreux Garamonds, Futura et la police Web Google Open Sans. Ci-dessous, la police Open Sans montrant le zéro, le O capital, le zéro et d'autres chiffres.

000123456789

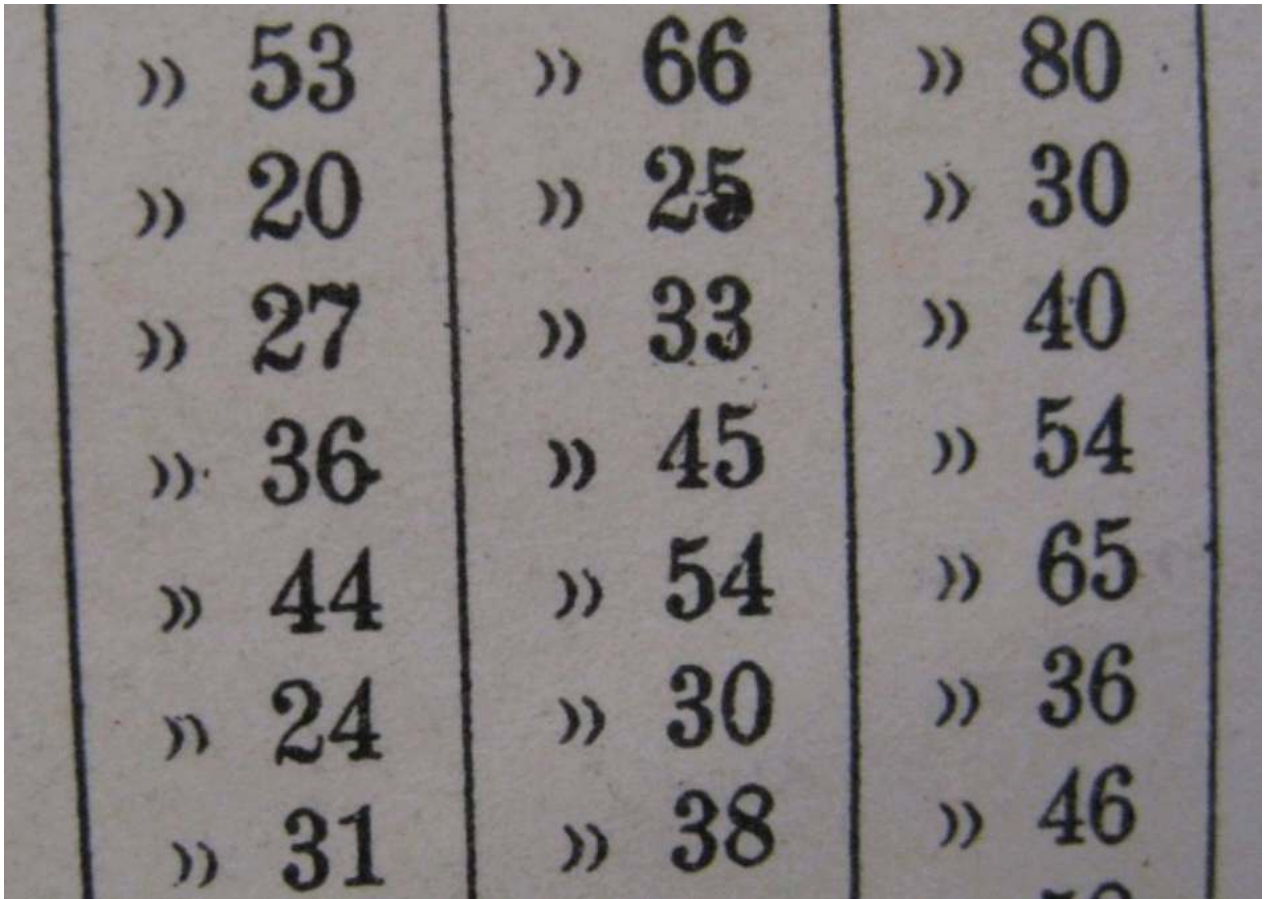
Un cercle parfaitement rond ou presque parfaitement rond est moins commun, mais il existe. Des exemples de polices qui utilisent cette approche comprennent la police web Google Vollkorn ainsi que d'autres types commerciaux tels que Mme Eaves, Vendeta et Fleischman BT Pro. Les polices qui utilisent des chiffres proportionnels de style ancien sont plus susceptibles de présenter cette approche. Parfois, un zéro à hauteur d'*x* mais qui est plus étroit sera également vu.

Jusqu'à 11 styles identifiables permettent de classer les chiffres quand vous incluez les fractions, les exposants et les indices. Nous allons examiner les 5 plus courants.

### Chiffre de style aligné

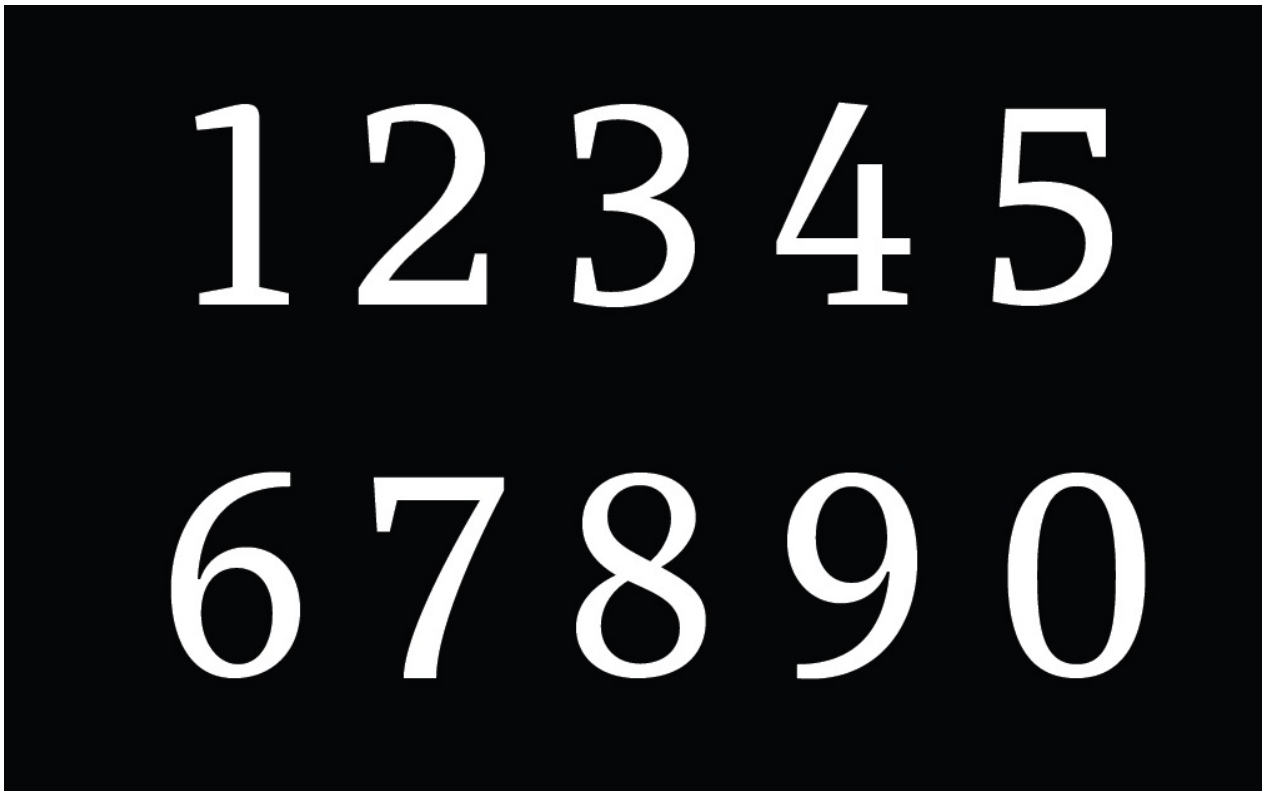
Les styles les plus courants de chiffres que l'on retrouve dans les polices sont l'alignement tabulaire et l'alignement proportionnel. L'alignement se rapporte aux hauteurs que les nombres utilisent. Si c'est un style aligné, les hauteurs de tous les nombres seront optiquement les mêmes. La différence entre les chiffres alignés de façon tabulaire et les chiffres alignés de façon proportionnelle est que dans l'alignement tabulaire tous les chiffres ont également la même largeur. Ce style est utile pour les feuilles de calcul et tout autre usage où il est utile pour les chiffres de rester empilés dans des lignes épurées à la fois horizontalement et verticalement.



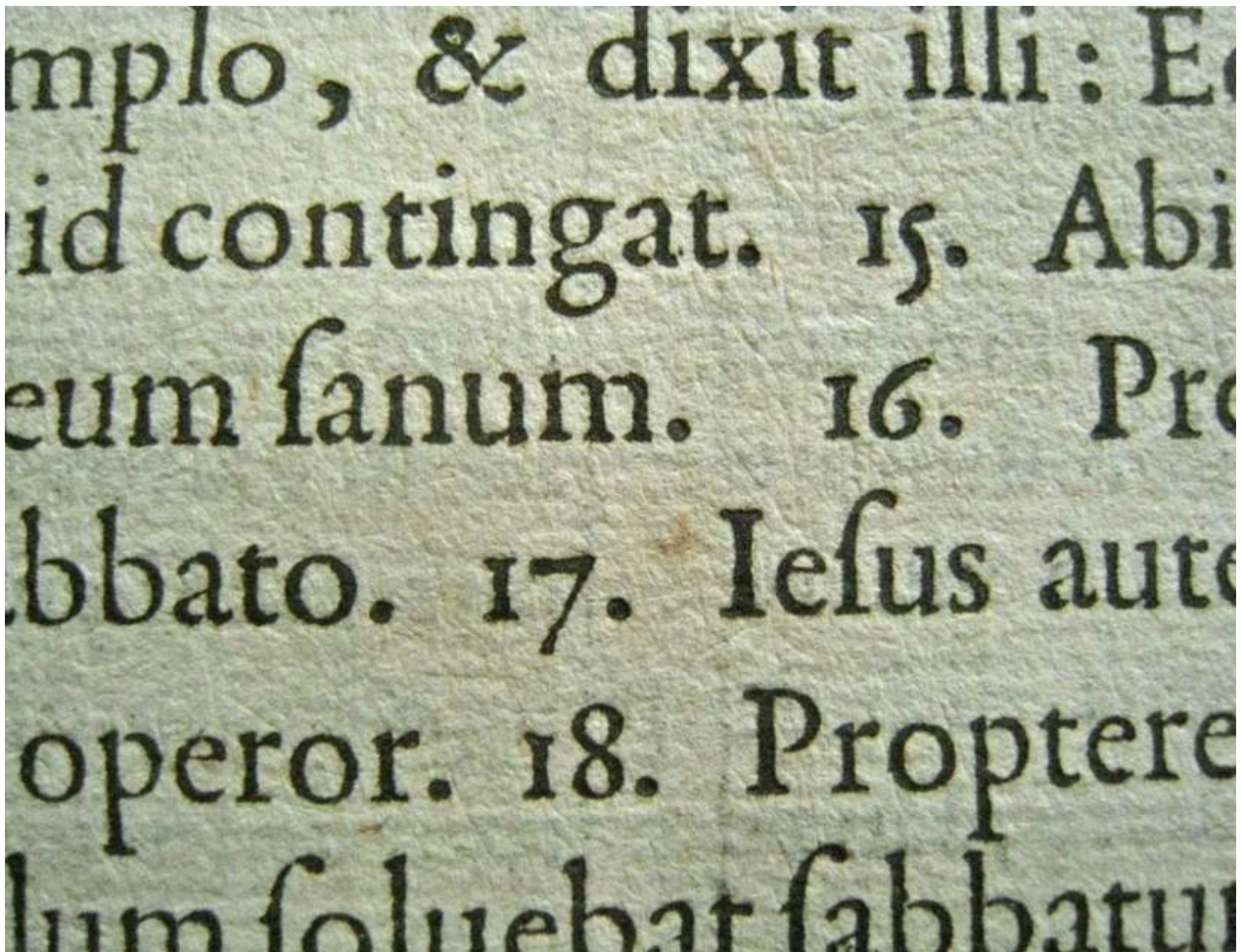


» 53	» 66	» 80
» 20	» 25	» 30
» 27	» 33	» 40
» 36	» 45	» 54
» 44	» 54	» 65
» 24	» 30	» 36
» 31	» 38	» 46

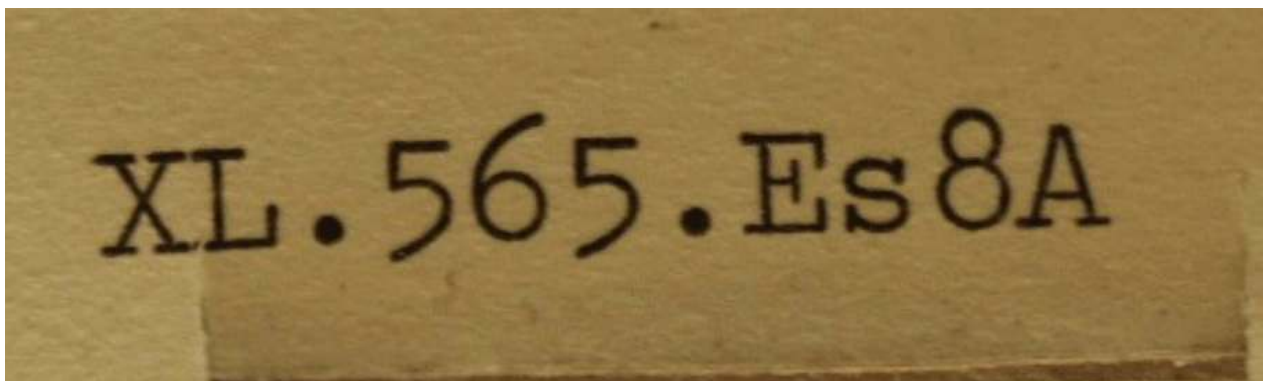
Les chiffres avec alignement proportionnel ont l'avantage d'être visuellement égaux parce que les formes et l'espacement des nombres varient pour compenser la densité de trait différente.



**Chiffres à style variant ou à l'ancienne**



Les chiffres tabulaires sont une invention relativement nouvelle en termes historiques. Avant d'exister, il y avait des nombres proportionnels à l'ancienne. Les chiffres de style ancien sont utiles si vous souhaitez que les nombres se mélangent et partagent le style d'un texte.



Les chiffres tabulaires à l'ancienne sont assez rares. Ils peuvent être utiles dans un rapport annuel où vous voulez donner une impression de style ancien mais l'espacement tabulaire typique de ce type de document. L'image ci-dessus provient d'une carte de catalogue de bibliothèque fait à la machine à écrire.

## Chiffres de style hybride

Les chiffres hybrides ne partagent pas la hauteur de capitale ou la hauteur x de la police, mais occupent leur propre hauteur. Le terme "hybride" se réfère à un mélange de chiffres à l'ancienne et alignés. Des exemples de polices qui utilisent des numéros de style hybride incluent les polices Georgia et les polices Google web Merriweather et Donegal. Les glyphes zéro, O capital, zéro, 1, 2, 3, etc de Merriweather sont montrés ci-dessous.

000123456789

## Gras

Lorsque nous parlons du style "gras", nous parlons vraiment d'une variable plus large, qui est la graisse. La graisse peut inclure tout, depuis très très fines lettres minces aux lettres énormément épaisses. Cette variable est utilisée dans la typographie de texte pour créer une forte séparation entre les corps de texte et elle est utilisée dans la conception graphique soit pour attirer l'attention sur un mot ou des textes courts, ou pour donner un sentiment à un texte spécifique (plutôt que de le contraster avec d'autres textes).

Alors que vous pouvez faire bien de choses avec la graisse, il est probable que votre première expérience avec l'ajustement de celle-ci sera d'essayer de créer un style gras pour accompagner votre texte de style régulier.

Parce que vous utilisez FontForge vous avez un avantage distinct. Contrairement à de nombreux programmes d'édition de polices, les résultats obtenus à partir du filtre de style dans FontForge peuvent en fait être convenables — plus que ceux que vous obtenez dans le logiciel de conception de type commercial. C'est parce que l'algorithme qu'il utilise est exceptionnellement sophistiqué.

La création d'une version en caractères gras d'une police peut être approximée rapidement en exécutant le filtre appelé *Changer la graisse* (que vous trouverez dans le menu Élément > Style) pour ajouter de la graisse à vos glyphes.

La nature automatique et la vitesse relativement élevée de ce processus le rend idéal pour tester quelle graisse vous voulez pour votre gras. Vous pouvez essayer d'exécuter ce filtre plusieurs fois et d'enregistrer plusieurs versions à comparer en texte à côté de votre style régulier. Cela dit, vous devrez peut-être modifier le résultat après avoir exécuté le filtre ou ajuster manuellement des glyphes individuels afin d'obtenir un résultat satisfaisant.

Il est également intéressant de rappeler que les glyphes qui n'ont pas une densité de traits (tels que 1, i, l, L, j et J) peuvent avoir besoin d'être plus épais, alors que les glyphes qui ont une densité de traits (tels que a, e, g, x, B, R, 8 et &) devront être moins épais que les autres glyphes.

## Interpolation de fonte

FontForge a une fonction d'interpolation entre des fontes séparées (voir la fonction *Interpoler fontes* dans le menu Élément). L'interpolation des fontes est une technique qui peut être utilisée pour créer des graisses intermédiaires à partir de deux autres graisses. Par conséquent, une façon de décider de la graisse de votre gras est de créer un gras qui est certainement plus lourd que vous avez besoin, puis d'interpoler plusieurs graisses différentes entre ce design trop gras et votre design régulier.

En utilisant cette technique, vous pouvez trouver plus rapidement la graisse que vous estimez la mieux adaptée à votre projet. La même technique peut être appliquée pour aider à décider des graisses encore plus lourdes tels que dans le style "extra-gras" et "noir", ainsi que les plus légers comme "book (ou régulier)" et "mince". Vous pouvez également définir des valeurs négatives à l'interpolation, par exemple vous obtiendrez un style "gras" si vous interpoler un "régulier" avec "mince" à -50%.

Cette logique de faire une fonte très mince et une fonte extra-gras, puis de générer tout ce dont vous avez besoin à partir de celles-ci, peut sembler la meilleure et la plus efficace façon de faire une graisse de style régulier et toutes les autres graisses que vous pourriez avoir besoin. Cependant, le résultat de cette approche est susceptible d'être excessivement fade. Au lieu de cela, et bien souvent, chaque changement significatif dans la graisse exigera son propre design maître à partir duquel d'autres graisses moyennes peuvent être faites.

## Lecture complémentaire (en anglais)

- <http://bigelowandholmes.typepad.com/bigelow-holmes/2015/07/on-font-weight.html>



## Italique

Le style italique est probablement celui qui est le plus mal compris dans le design de polices, mais il est celui avec le plus grand potentiel d'enthousiasme et de plaisir en raison du grand nombre de variables avec lesquelles vous, le concepteur, pouvez jouer.

Les italiques sont différentes du gras en ce sens qu'ils ne sont pas censés paraître avoir une graisse différente du style régulier. Au lieu de cela, ils sont destinés à offrir une *texture* différente du style régulier. Une plus grande intensité dans cette différence signifie que l'italique est particulièrement utile pour créer un sens de contraste avec le régulier. Cet effet plus fort est utile pour mettre en évidence des mots simples ou des passages courts de texte. En revanche, une texture moins différente est souvent utile dans des situations où vous définissez plusieurs lignes, des paragraphes entiers ou même des pages en italique.

## Inclinaison

La variable la plus fréquemment associée à l'italique est l'inclinaison. En effet, lorsqu'un navigateur Web demande un italique dans une règle CSS et qu'il n'y a pas d'italique, il incline simplement le style régulier pour créer un faux italique ou synthétique. Il n'est probablement pas surprenant que lorsque les gens commencent à concevoir des polices qu'ils envisagent également cette approche. Les origines de cette idée remontent au milieu du XXe siècle avec le modernisme tel qu'il a été appliqué à la conception. C'est pourquoi les premières italiques vues dans les polices de caractères comme Helvetica étaient également des versions inclinées des régulières.

### Combien d'inclinaison?

Certains italiques n'ont pas d'inclinaison. Pas vraiment! Ces italiques sont appelés italiques droits. Toutefois, il est probable que si votre design a seulement un italique, vous choisirez que cet italique ait un certain degré d'inclinaison. En général, les italiques tendent à s'incliner entre 4 et 14 degrés. La plupart des polices contemporaines s'inclinent entre 6-9 degrés.

Bien que l'inclinaison peut être importante pour la conception d'une italique, est facile à remarquer, et peut même être fait avec un succès limité en utilisant un filtre automatique, ce n'est pas la seule variable que vous pouvez utiliser pour aider à séparer votre style italique de votre style régulier. Vous pouvez envisager d'utiliser une ou plusieurs des variables suivantes en plus de l'inclinaison.

## Construction italique

Le terme "italique" ne se réfère pas en fait à l'inclinaison vu dans dans de nombreux designs d'italiques, mais se réfère plutôt à un style d'écriture qui est devenu populaire dans l'Italie du 14ème siècle. Ce style d'écriture était une forme plus rapide et connectée qui utilise une construction différente pour ses lettres que celle que l'on voit dans le style régulier. Cette construction différente ou modèle de traits est ce que les concepteurs de polices veulent dire quand ils disent qu'ils ont conçu un "vrai" italique. Cette construction a beaucoup de sous-caractéristiques que vous pouvez choisir d'inclure dans un design d'italique.

### Contrepoinçons triangulaires

La plus évidente de ces caractéristiques est le contrepoinçon triangulaire créée par les lettres avec des jointures. Ces lettres comprennent a, b, d, g, h, m, n, p, q et u. Cette variable est puissante, en partie parce que le contrepoinçon est une variable puissante, mais aussi en raison du grand nombre de lettres avec cette caractéristique. Ce fait, combiné à la fréquence élevée de leur utilisation dans la plupart des langues, est également un très grand (et probablement le plus grand) facteur.

Lors de la conception de votre italique, vous pouvez effectivement régler l'effet que votre italique donne en faisant des ajustements relativement faible à la hauteur des jointures. Des changements subtils peuvent donner des résultats surprenants. Néanmoins, toutes les polices italiques ne profitent pas de cette variable.

## Traits entrant et sortant

De nombreuses fontes italiques utilisent des empattements asymétriques, sous forme de traits entrant ou sortant, ou les deux. Lorsqu'un seul est utilisé, il est plus fréquent d'utiliser le trait sortant et d'appliquer un style droit où le trait entrant aurait dû être. L'intensité de l'effet fournie par trait entrant et sortant peut être contrôlée par la graisse des traits et en ajustant combien longs ils sont. Comme les contrepointings triangulaires, une grande partie de leur utilité et de leur puissance vient du fait que plusieurs lettres les utilisent.

## Condensation

Les styles italiques sont normalement un peu moins large ou plus condensés que le style régulier. Parce que la condensation est une caractéristique vue dans toutes les lettres en italique, c'est une variable très puissante. Cette variable peut être employée à la fois d'une manière grossière et subtile. Si vous choisissez d'utiliser cette variable, il est nécessaire d'ajuster la graisse des traits pour rendre la graisse de l'italique la même ou presque la même que celle du design régulier. Plus votre italique est condensé, plus vous devrez effectuer cet ajustement.

## Mélange de variables

La plupart des italiques utilisent toutes les variables énumérées ci-dessus dans diverses proportions. Il est utile de regarder une gamme de designs d'italiques et d'analyser quelles variables sont utilisées et dans quelle force. Lorsque vous faites cela, vous remarquerez qu'aucune des variables ne sont utilisées à pleine puissance. Au lieu de cela, une des variables tend à dominer, avec un usage plus limité des autres. Plus l'utilisation des variables est forte, plus votre style italique sera contrasté avec le style régulier.

Il faut remarquer que dans les dix dernières années, nous avons vu un nombre croissant de concepteurs de polices choisir d'offrir non pas seulement un italique dans leur de familles de polices, mais deux ou même plus. Remarquez également que les dictionnaires utilisent parfois plus d'un style d'italique.

Lorsqu'ils ont été faits originalement pour pour l'impression, les italiques n'étaient pas considérés comme faisant partie du même design de police ou famille de police. Cette idée est devenue une norme au cours des 19e et 20e siècles. Même l'idée de mélanger le style italique avec le style régulier ne faisait pas partie de l'idée originale derrière cette écriture. Les premières fontes italiques ont été utilisées pour concevoir des livres entiers, au lieu du style romain droit. Il est probablement prudent de supposer que le rôle de l'italique va continuer d'évoluer.

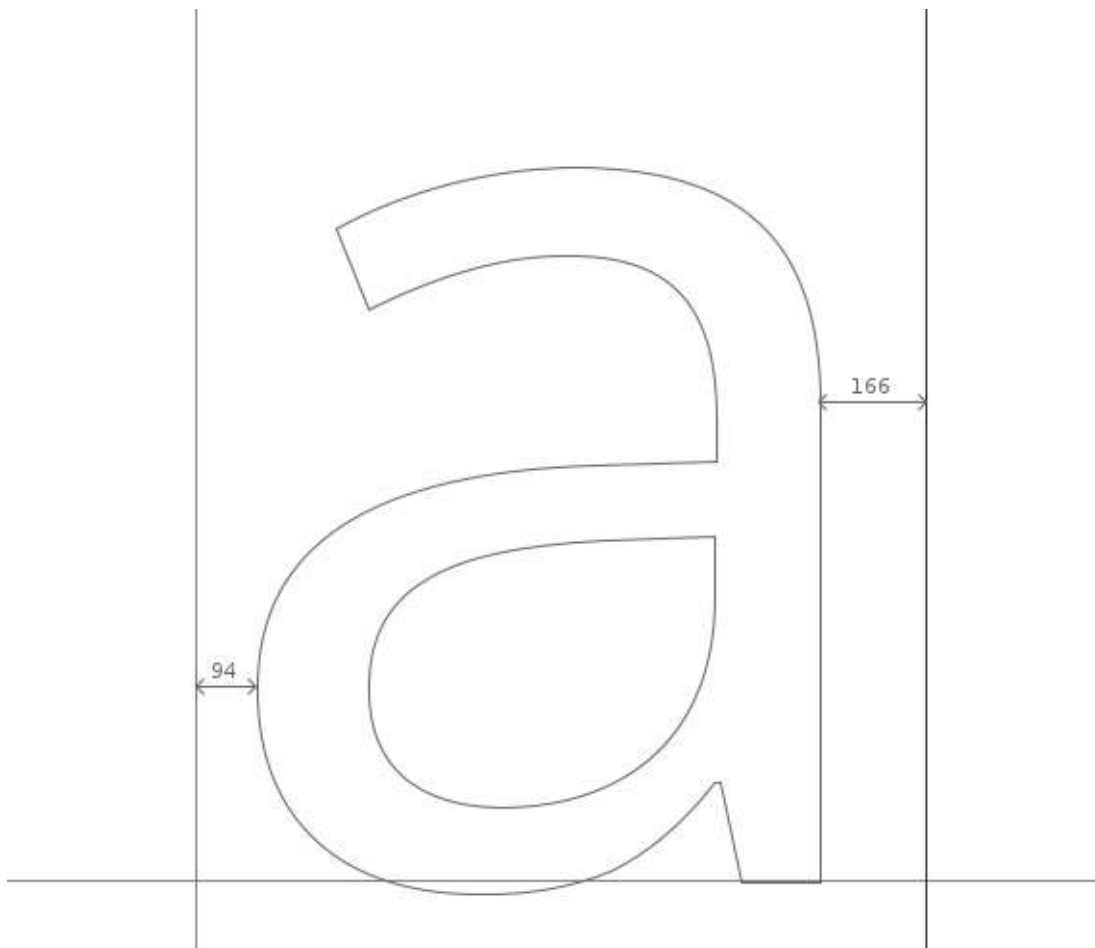
## Espace, métriques et crénage

Les espaces entre les caractères sont une partie importante et intégrale de la conception d'une police.

La conception de l'espacement des lettres d'une police doit faire partie intégrale du processus de design d'une police. Un bon espacement est nécessaire pour le bon fonctionnement de la police.

Dans FontForge, la fenêtre de métriques vous permet de concevoir les métriques de votre police, de modifier l'espacement entre les glyphes et leur apparence ensemble. La fenêtre de métriques peut être ouverte à partir du menu 'Fenêtre', ou en utilisant la commande Control-k.

L'espace entre deux glyphes a deux composantes; l'espace après le premier glyphe, et l'espace avant le deuxième glyphe. Ces espaces entre les glyphes sont composés 'd'approches latérales' entre chaque paire de glyphes. Chaque glyphe a une approche gauche et une approche droite. Dans l'exemple ci-dessous de la lettre minuscule 'a' de Open Sans, l'approche droite a une valeur de 166 unités et l'approche gauche a une valeur de 94 unités.



## Fonctions de base de la fenêtre de métriques

Les approches latérales des caractères peuvent être éditées dans la fenêtre de métrique de FontForge de 5 manières;

- Déplacer manuellement chaque limite d'approche latérale.
- Déplacer manuellement un caractère. Notez cependant que faire glisser un caractère n'aura d'effet que sur la valeur du côté gauche.
- Les valeurs des approches latérales peuvent être modifiées en éditant directement leur valeur dans les tables de métriques de la fenêtre de métriques.



- La valeur des approches latérales peut être incrémentée / décrétementée à l'aide du clavier.
- Utiliser des commandes dans le menu Métriques de la fenêtre des métriques.

### Réglage des valeurs d'approches latérales avec le clavier.

Une méthode d'ajustement rapide et précis des valeurs métriques dans FontForge consiste à utiliser les touches haut, bas, gauche et droite du clavier. Les touches haut et bas sont utilisées pour incrémenter / décrétement les valeurs et alt+haut, alt+bas, alt+gauche et alt+droit sont utilisées pour naviguer autour des différents champs de valeur de la fenêtre de métriques.

## Principes généraux

En principe, les caractères symétriques tels que 'A' 'H' 'I' 'M' 'N' 'O' 'T' 'U' 'V' 'W' 'X' 'Y' 'o' 'v' 'w' 'x' auront des approches symétriques, par exemple les approches gauche et droite d'un 'H' auront la même valeur. Notez cependant que ce n'est pas une règle absolue, mais générale.

Au fur et à mesure que vous espacez les caractères de votre design, vous devriez faire confiance à vos yeux. L'essentiel est de 'concevoir — regarder — ajuster — regarder à nouveau'.

Pour le parfait débutant; ne supposez pas que l'on obtient des résultats fiables en s'appuyant sur l'espace mesuré. Par exemple, alors que les mesures entre deux caractères peuvent être inégales, l'œil peut les voir comme égales. Un exemple évident de ceci peut être vu en essayant d'espacez les caractères 'H' et 'O'. Donc, dans l'exemple ci-dessous, les approches latérales du 'H' et du 'O' sont égales, mais semblent inégales. Dans la ligne inférieure, les approches latérales ne sont pas égales mais l'espacement semble plus équilibré.



H O O O H H H H O H H H O O O O  
H O O O H H H H O H H H O O O O

Un outil pour générer de tels textes est disponible à cette adresse: <http://tools.ninastoessinger.com/>

## Commandes du menu Métriques pour l'édition de métriques

'Centrer en largeur' — Cela centre le glyphe actuel dans sa largeur actuelle.

'Type de fenêtre' — La fenêtre de métriques de FontForge peut être configurée pour se comporter de 2 manières pour l'ajustement des métriques;

- 'Chasse seulement' — dans ce mode, la fenêtre des métriques ne peut être utilisée que pour ajuster la chasse des glyphes.
- 'Les deux' — Dans ce mode, la fenêtres des métriques ajustera soit la chasse, soit les valeurs de crénage.

'Définir chasse' — cette commande vous permet de changer la largeur du glyphe courant.

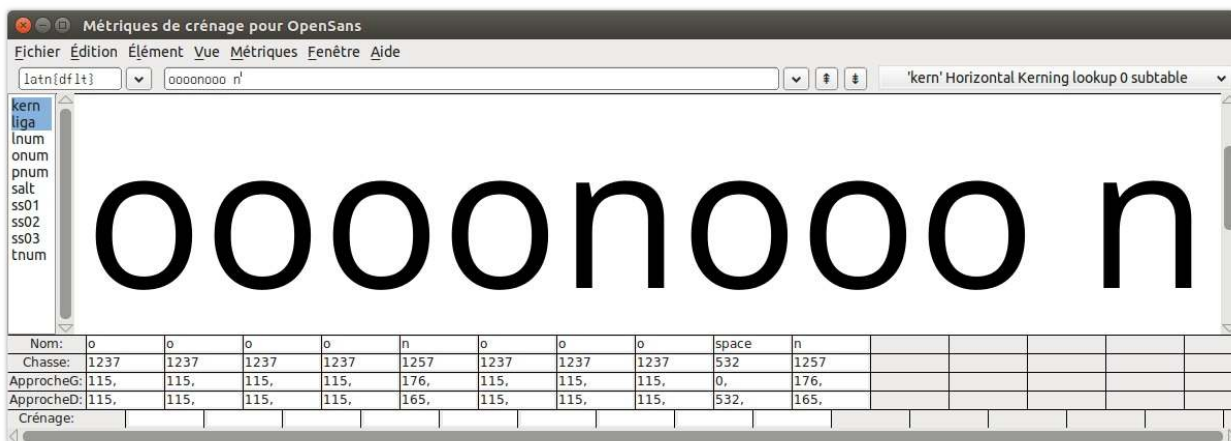
'Définir approche gauche' — vous permet de changer la valeur de l'approche gauche.

'Définir approche droite' — vous permet de changer la valeur de l'approche droite.

## Une approche de base pour l'espacement

La méthode suivante est conçue pour vous aider à concevoir efficacement les métriques de votre police.

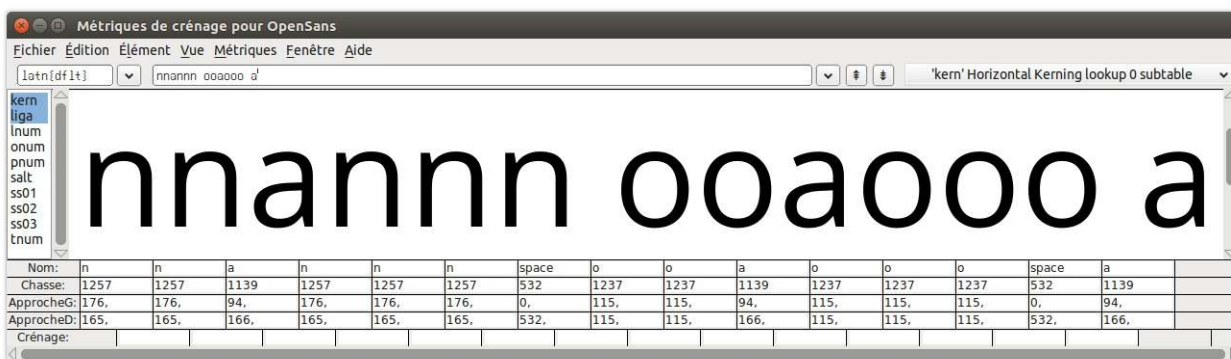
En commençant par une chaîne de lettres minuscules 'o' dans la fenêtre de métriques, les approches gauche et droite peuvent être ajustées jusqu'à ce que l'espacement des caractères ait bonne apparence. Une façon d'obtenir cette 'bonne apparence' est d'équilibrer l'espace entre les 'o' avec l'espace blanc à l'intérieur des 'o'. En général, à l'exception des polices inclinées ou italiques, les côtés gauche et droit d'un 'o' minuscule doivent être de valeur égale. Une fois que vous êtes satisfait de l'espacement de votre chaîne de caractères 'o', introduisez le caractère 'n' de votre police (voir ci-dessous). Ensuite ajustez les approches latérales du 'n' de sorte que son espacement s'inscrit dans la balance de la chaîne de caractères 'o' (voir ci-dessous). Notez qu'en raison de la nature de la façon dont nos yeux voient, l'approche latérale droite du 'n' aura toujours une valeur plus petite que celle de l'approche gauche et les approches latérales du 'o' seront plus petites que les approches latérales du 'n'.



Une fois que le 'n' et le 'o' sont adéquatement espacés, leurs approches peuvent être utilisées pour créer les approches pour un ensemble d'autres caractères, par exemple;

- L'approche gauche du 'o' peut être utilisée pour l'approche gauche du 'c', 'd', 'e', et 'q'.
- L'approche droite du 'o' peut être utilisée pour l'approche droite du 'b' et 'p'.
- L'approche droite du 'n' peut être utilisée pour l'approche droite du 'h' et 'm'.
- L'approche gauche du 'n' peut être utilisée pour l'approche gauche du 'b', 'h', 'k', 'm', 'p' et 'r'

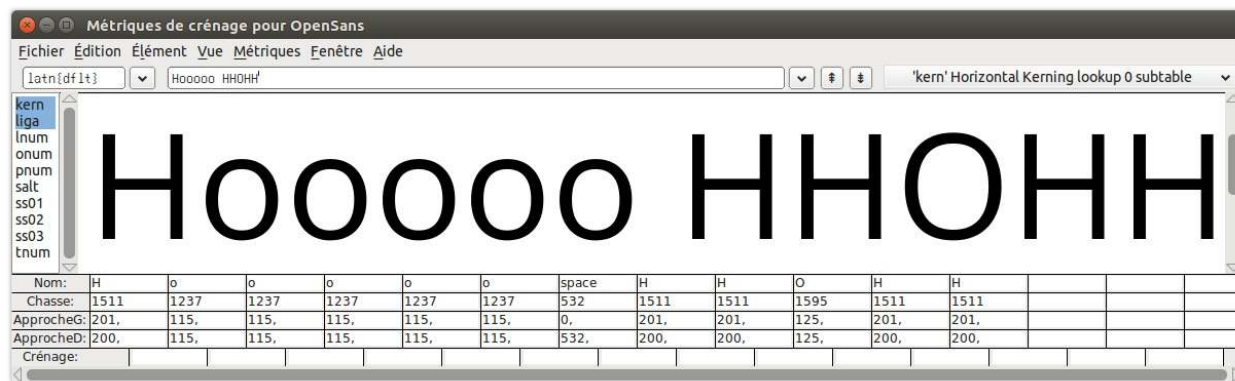
Note — Ce qui précède doit être utilisé comme un guide uniquement. Ce guide fournit un point de départ efficace pour trouver les valeurs correctes des approches latérales.



A partir de là, il est logique d'espacer ensuite le reste des approches latérales des caractères minuscules contre des chaînes de caractères 'n' et 'o', comme on peut le voir dans le diagramme ci-dessus. Encore une fois, faites confiance à vos yeux pour atteindre l'équilibre correct des caractères.

## Lettres majuscules

Les lettres majuscules peuvent être espacées en utilisant les mêmes principes que ci-dessus. Par exemple, commencez par la chaîne 'Hooooo' et ajustez l'approche droite du 'H' jusqu'à ce qu'il soit équilibré par rapport à la chaîne de caractères 'o'. L'approche gauche du 'H' étant égale à l'approche droite, le 'O' majuscule peut alors être espacé contre le 'H' (voir ci-dessous).



De là, tous les autres caractères peuvent être espacés par rapport à des caractères qui ont déjà été espacés. Il convient de noter que cette méthode peut être utilisée comme un bon point de départ pour l'espacement d'une police, mais il est vraisemblable que des ajustements plus minutieux de l'espacement seront également nécessaires pour atteindre des niveaux plus élevés de qualité. D'autres chaînes de caractères telles que 'naxna', 'auxua', 'noxno' et 'Hxndo' peuvent être utiles pour cela.

## Crénage

Le crénage est le réglage de l'espacement entre des paires de caractères spécifiques. Le crénage permet l'espacement individuel des paires de caractères qui est appliqué en plus de l'espacement fourni par les approches latérales d'un caractère. Des exemples courants de paires de caractères où le crénage est souvent nécessaire pour améliorer l'espacement seraient 'WA', 'Wa', 'To', 'Av'. Dans les exemples ci-dessous, nous pouvons voir que sans crénage, l'espacement entre les lettres 'T-o' and 'V-a' est trop large, alors qu'avec crénage, l'espace entre ces paires de caractères est beaucoup plus équilibré avec l'impression d'espacement du reste de la police.

Toned Avenue  
Toned Avenue

La fenêtre de métriques de FontForge peut être utilisée pour concevoir les approches latérales et les valeurs de crénage. Les valeurs de crénage peuvent être appliquées à une police de plusieurs façons dans FontForge. Deux de celles-ci sont présentées ci-dessous: le crénage par classes et le crénage par paires individuelles.

## Le menu Métriques de FontForge

'Type de fenêtre' — La fenêtre de métriques de FontForge peut être configurée pour se comporter de 2 manières pour l'ajustement du crénage;

- 'Crénage seulement' — dans ce mode, la fenêtre des métriques ne peut être utilisée que pour ajuster le crénage.
- 'Les deux' — Dans ce mode, la fenêtres des métriques ajustera soit la chasse, soit les valeurs de crénage.

'Crénage par classes' — Cette commande fournit à l'utilisateur une boîte de dialogue pour manipuler les classes de crénage.

'Crénage manuel' — Cette commande fournit à l'utilisateur une boîte de dialogue à partir de laquelle vous pouvez ajuster des paires déjà créées ou créer de nouvelles paires (voir ci-dessous).

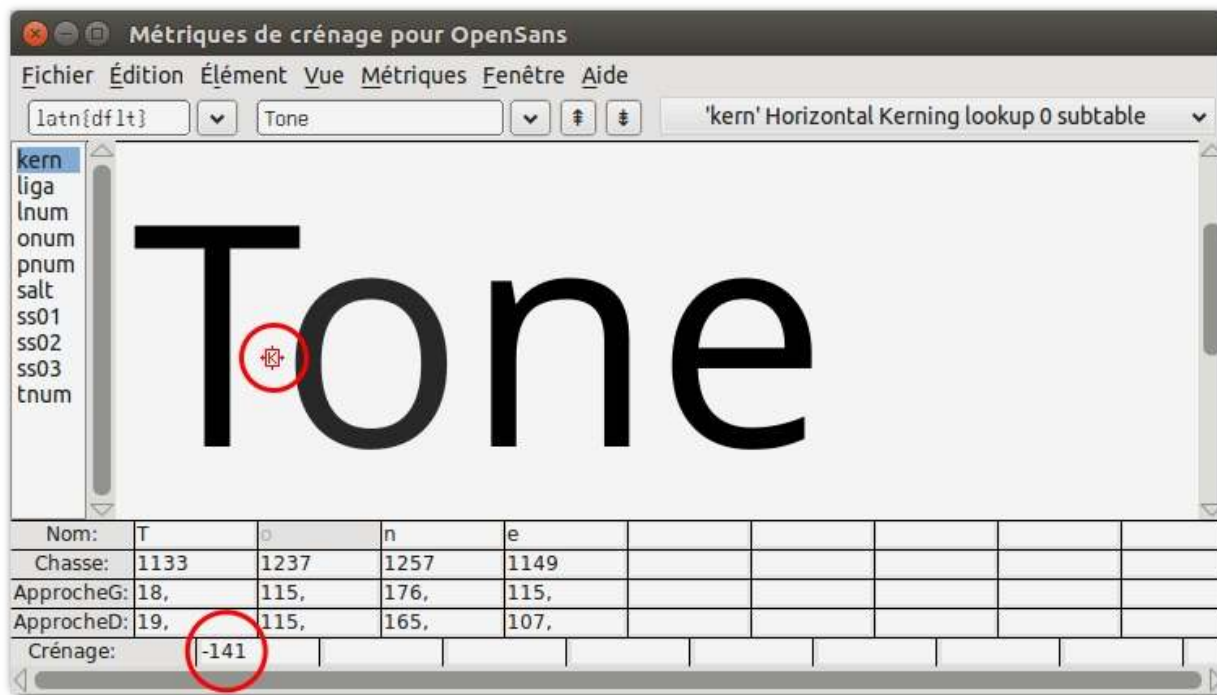


## Réglage des valeurs de crénage avec le clavier

Tout comme pour ajuster les valeurs des approches latérales, les valeurs de crénage peuvent être modifiées rapidement et avec précision dans FontForge en utilisant les touches 'haut', 'bas', 'gauche' et 'droit' d'un clavier. Les touches 'haut' et 'bas' sont utilisées pour incrémenter / décrémenter les valeurs et 'alt+haut', 'alt+bas', 'alt+gauche' et 'alt+droit' sont utilisées pour naviguer autour des différents champs de valeur de la fenêtre de métriques.

## Crénage de paires individuelles

C'est le niveau le plus basique de création de paires de crénage dans FontForge. Dans la fenêtre de métrique, la valeur de crénage entre 2 caractères peut être ajustée manuellement soit en faisant glisser le caractère de droite vers ou à partir du caractère de gauche ou en modifiant la valeur de crénage directement dans la table de métriques de la fenêtre. Pour modifier les valeurs de crénage en faisant glisser des caractères, utilisez la poignée d'outil de crénage qui apparaît lorsque le curseur de la souris est placé entre 2 caractères (voir la capture d'écran ci-dessous). La valeur de crénage dans la table de métriques peut être modifiée en saisissant manuellement des valeurs ou en incrémentant / décrémentant la valeur à l'aide des touches haut / bas de votre clavier.

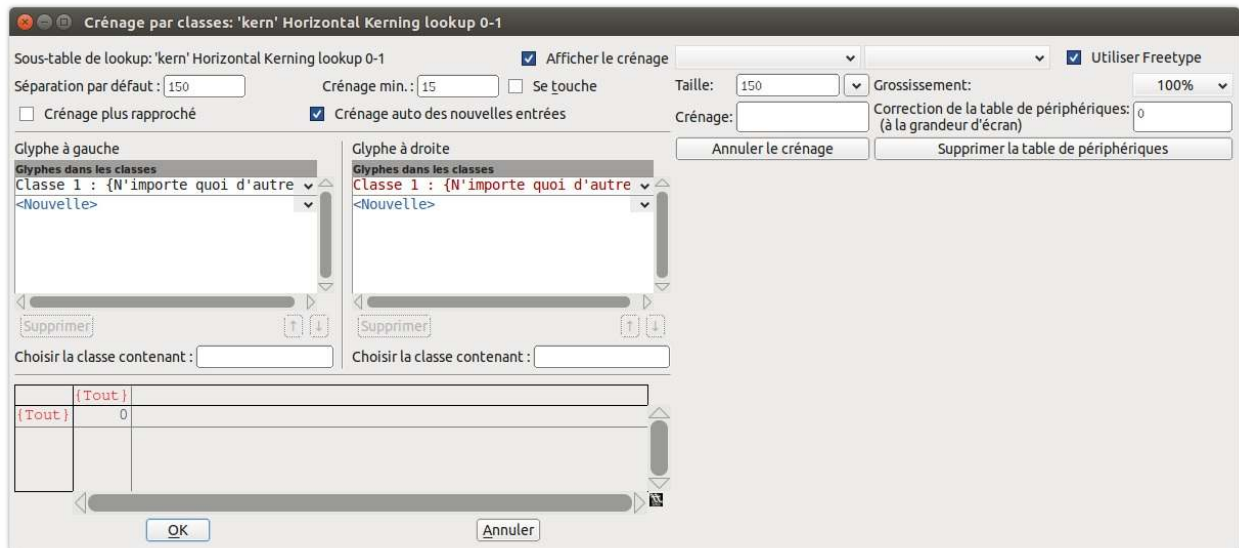


## Crénage par classes

Une 'classe de crénage' dans FontForge peut être créée pour construire des groupes de caractères qui auront tous la même valeur de crénage appliquée. Par exemple une classe peut être créée, appelons-la 'o\_panse\_gauche' dans laquelle les caractères 'o', 'c', 'D', 'e' et 'q' auront toujours la même valeur de crénage précédée, par exemple, du caractère 'T'. Le 'T' pourrait également être lui-même membre d'une autre classe qui comprendrait probablement d'autres caractères tels que Tcaron et Tbar. Effectivement, le crénage par classe peut vous faire gagner beaucoup de temps.

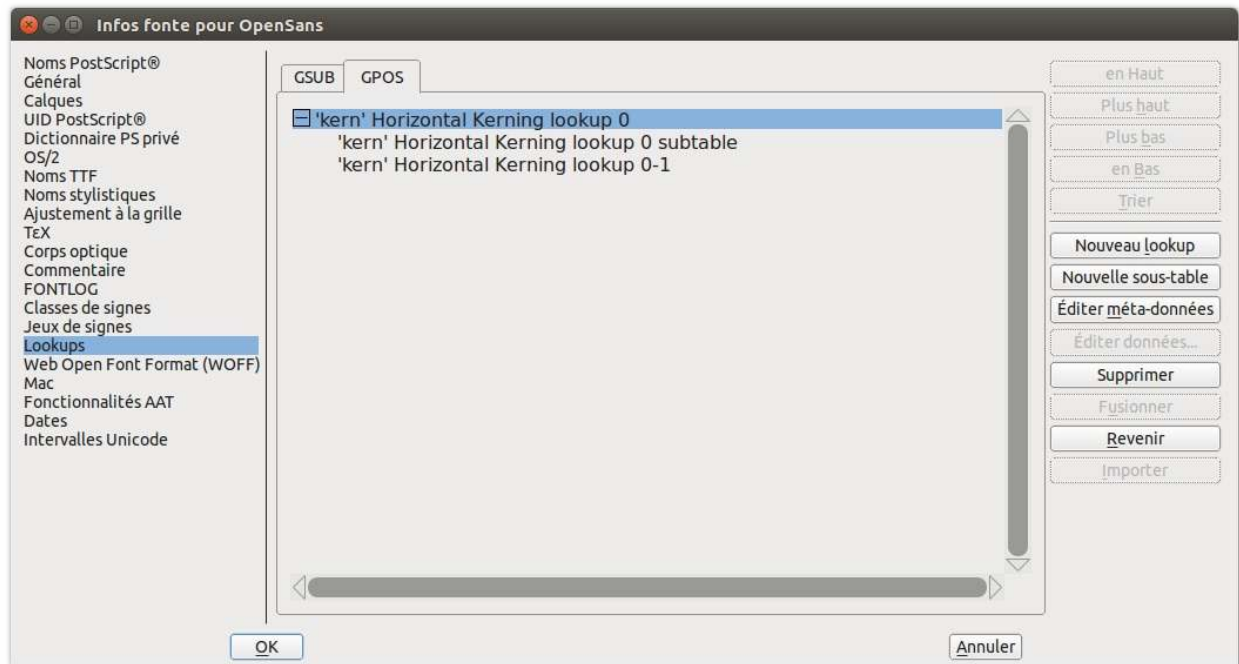
Le moyen le plus direct de créer des classes de crénage est à partir de l'élément «Crénage par classes» dans le menu «Métriques» de FontForge.

- Sélectionnez "Crénage par classes" et la fenêtre "lookup" vous sera présenté.
- Cliquez sur le bouton "Ajouter" et une autre fenêtre apparaîtra, où vous pourrez créer un lookup de de crénage.
- Choisissez l'élément "Positionnement par paire (crénage)" dans le menu déroulant "Type".
- Maintenant, cliquez sur la flèche vers le bas à côté de "Nouvelle" dans la colonne "Fonctionnalité" et choisissez "Crénage horizontal" dans le menu déroulant.
- Cliquez sur "OK". Vous pouvez conserver les noms par défaut que FontForge crée pour vous.

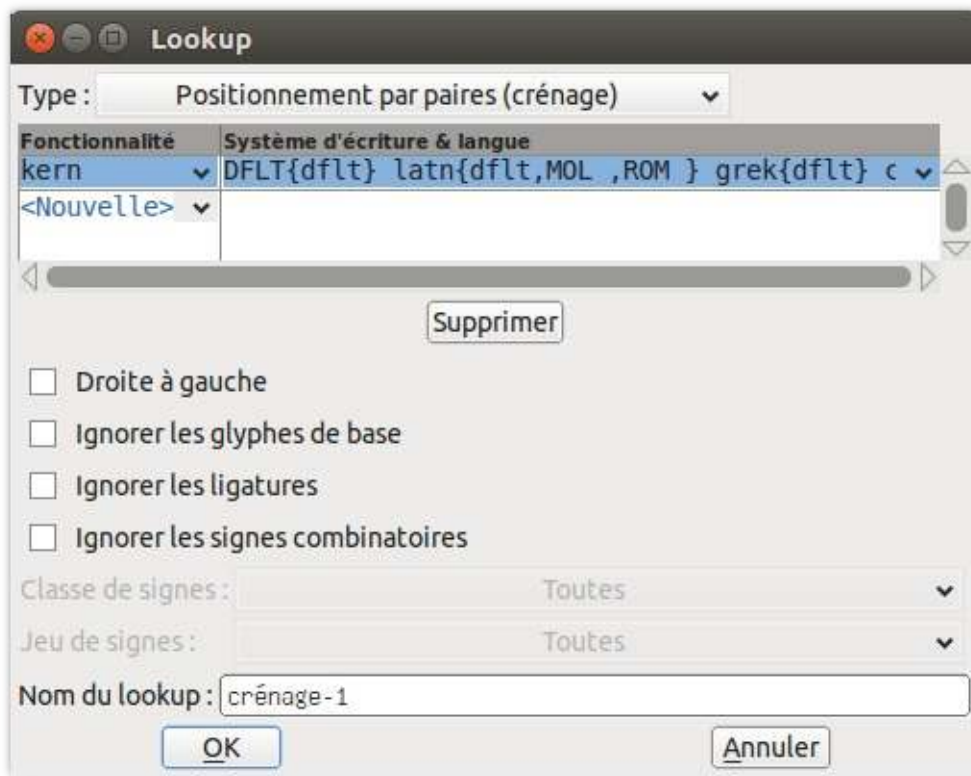


Maintenant, vous est présentée la fenêtre où vous pouvez construire vos propres classes de crénage (voir ci-dessus). Le premier caractère d'une paire de crénage sera choisi dans la colonne de gauche et le second caractère d'une paire sera choisi dans la colonne de droite.

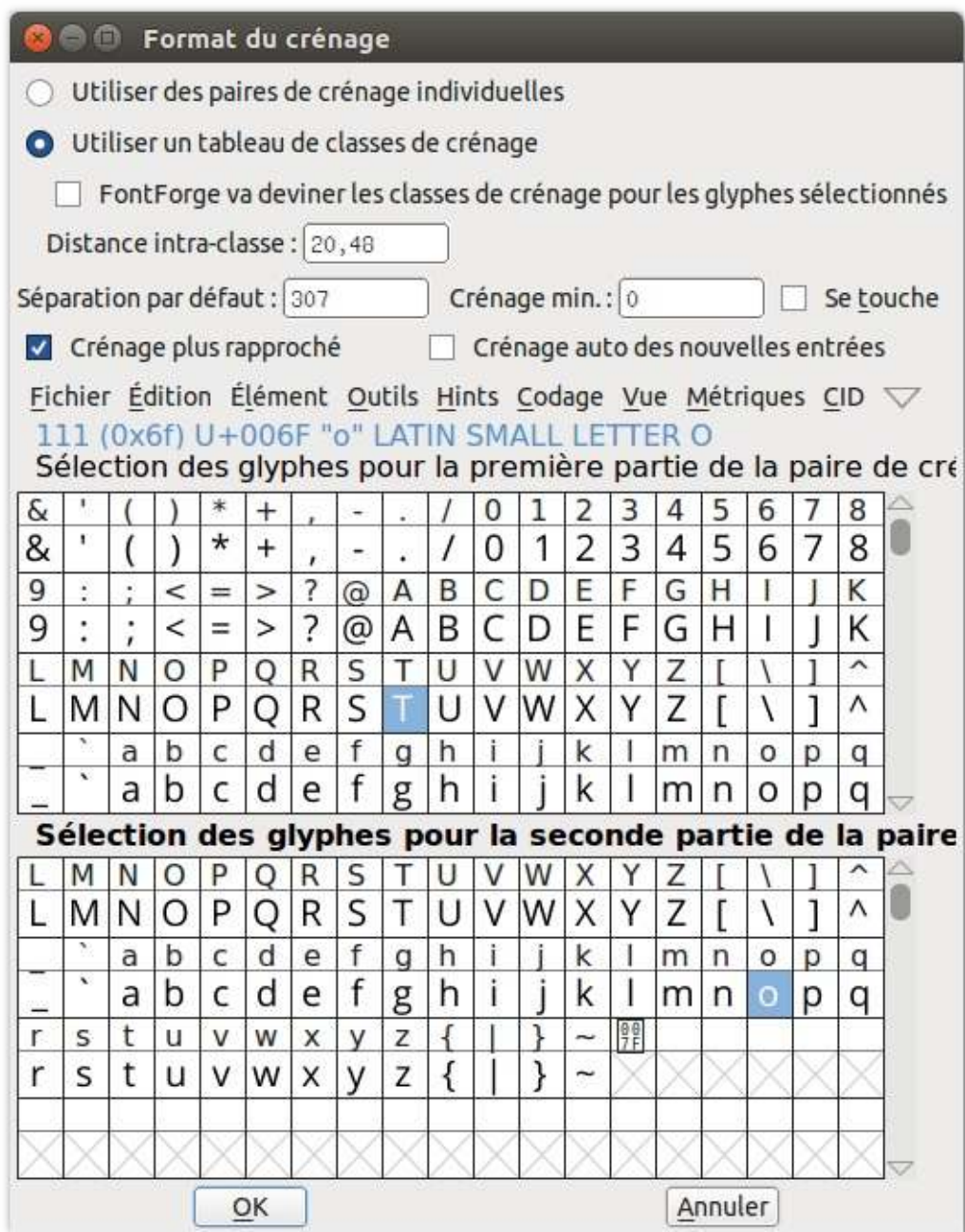
L'onglet Lookups dans Élément > Infos fonte > fournit une interface au crénage par classe dans FontForge. Cette interface affiche un dialogue montrant tous les lookups GPOS (dont le crénage est un) et leurs sous-tables. Voir la capture d'écran ci-dessous;



Pour créer un nouveau lookup de crénage, cliquez sur «Nouveau lookup» et choisissez 'Positionnement par paire (crénage)' comme type de lookup et donnez au lookup un nom unique (voir ci-dessous).

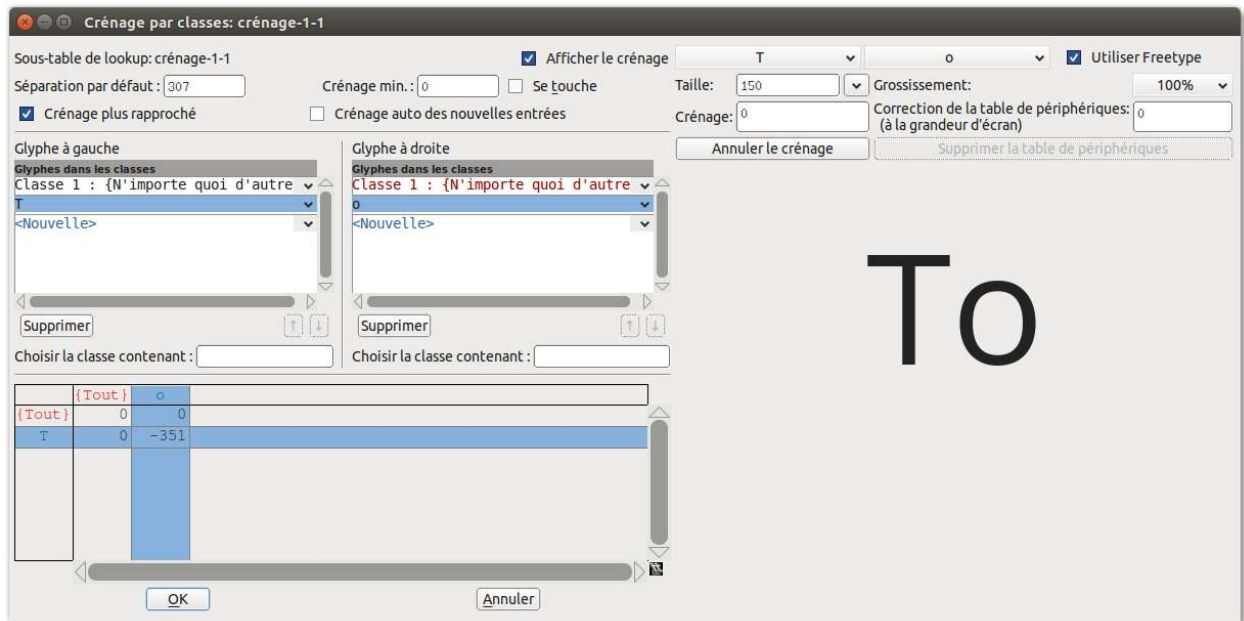


Chaque ensemble de classes de crénage existe dans sa propre sous-table. Pour créer une sous-table, cliquez sur 'Nouvelle sous-table'. Lorsque vous créez une sous-table de crénage, on vous demandera si vous voulez un ensemble de paires de crénage individuelles ou un tableau de classes de crénage. Si vous choisissez les classes, vous obtiendrez la boîte de dialogue suivante dans lequel vous pouvez créer vos classes. Notez que vous pouvez choisir d'activer FontForge pour 'deviner' ou 'auto-crénage' les valeurs de crénage entre les classes que vous créez dans la boîte de dialogue. Si vous utilisez FontForge pour deviner les valeurs de crénage, vous aurez sans doute besoin d'une certaine quantité d'essais et d'erreurs pour y parvenir, mais il est logique d'utiliser la fonction autokern comme point de départ pour le crénage de votre police.



Par exemple, dans la capture d'écran ci-dessus, 2 classes ont été créées; une classe contenant le caractère 'T' et une classe contenant le caractère 'o'. En cliquant sur 'ok' dans la boîte de dialogue ci-dessus, vous obtiendrez la fenêtre suivante où vous pouvez affiner la quantité de crénage entre ces deux classes 'T' et 'o'.





## Crénage manuel

Si les valeurs auto-crénées doivent être ajustées (et elles le devront)! Alors, cela peut être fait de plusieurs façons.

- Via la boîte de dialogue 'crénage par classes'.
- En utilisant la fenêtre de métriques.
- En utilisant la commande 'Crénage manuel' du menu Métriques.

## Lecture complémentaire (en anglais)

[Strategies for determining letter spacing](#)

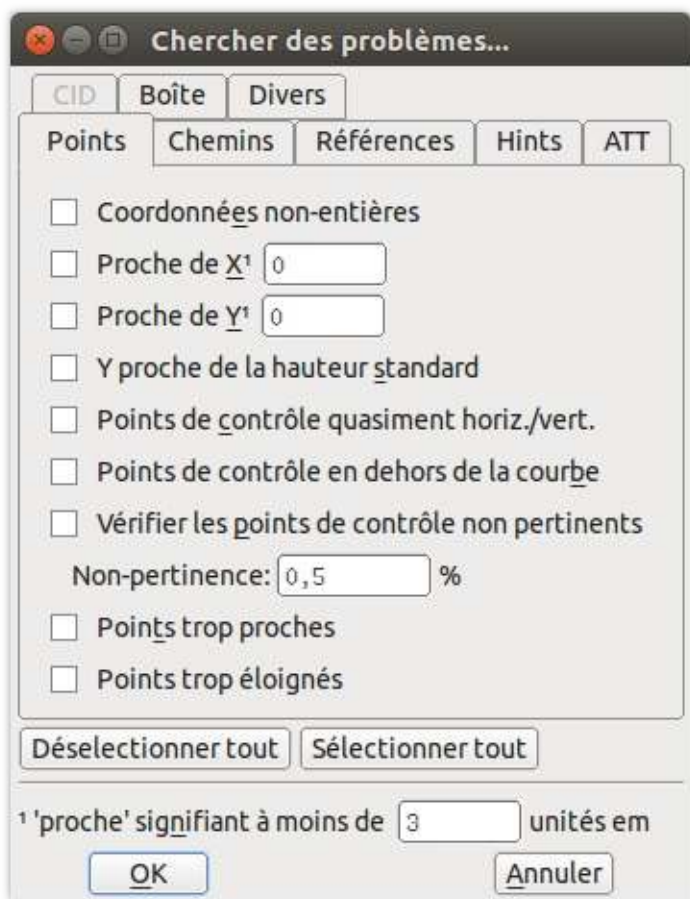
## S'assurer que votre police fonctionne

Dans un monde parfait, votre police serait prête à construire et à installer sur n'importe quel ordinateur moderne sans effort particulier, mais la réalité est plus compliquée—en particulier au cours du processus de conception. Les polices peuvent comporter des erreurs techniques qui les empêchent de fonctionner ou de s'afficher correctement. Par exemple, les courbes qui se croisent ne seront pas rendues correctement parce qu'elles n'ont pas «d'intérieur» et «d'extérieur». Les différents formats de fichiers de polices s'attendent également à ce que les glyphes adhèrent à certaines règles qui simplifient le placement du texte à l'écran. Les polices qui enfreignent les règles peuvent provoquer des problèmes inattendus. Un exemple de ce type de problème est que tous les points sur une courbe doivent avoir des coordonnées qui sont des entiers. Enfin, il y a des erreurs stylistiques qui ne sont pas techniquement incorrectes, mais que vous voudrez réparer — telles que des lignes qui sont destinées à être parfaitement horizontale ou verticale, mais sont accidentellement légèrement inclinées.

FontForge offre des outils que vous pouvez utiliser pour localiser (et, dans de nombreux cas, réparer) les trois catégories de problème. La validation de votre police pour éliminer ces erreurs permettra non seulement de s'assurer qu'elle peut être installée et appréciée par les utilisateurs, mais fera en sorte que le projet fini brille.

## Trouvez les problèmes

Le premier outil est appelé *Recherche de problèmes* et se trouve sous le menu Élément. Vous devez d'abord sélectionner un ou plusieurs glyphes, soit dans la fenêtre de fonte, soit dans la fenêtre de glyphe, ou dans la fenêtre de métriques, puis ouvrez l'outil Rechercher des problèmes. L'outil vous présente un assortiment de problèmes potentiels dans huit onglets distincts.



Vous pouvez sélectionner les problèmes qui vous intéressent en cochant la case à côté de chacun et, dans certains cas, fournir une valeur numérique à vérifier. Lorsque vous cliquez sur le bouton OK, l'outil examine tous les glyphes sélectionnés et signale tous les problèmes qu'il trouve dans une boîte de dialogue.

Les problèmes que l'outil recherche de problème peut vérifier sont triés dans ces huit groupes:

- Problèmes liés aux points
- Problèmes de chemins et de courbes
- Problèmes de références
- Problèmes avec le hinting
- Problèmes avec ATT
- Problèmes spécifiques aux polices CID-keyed
- Problèmes avec les boîtes englobantes
- Divers autres problèmes

Pas tous les contrôles sont nécessaires. Certains s'appliquent uniquement à des écritures ou des langages spécifiques (comme ceux de l'onglet "CID"), tandis que d'autres s'appliquent uniquement à des fonctions de police facultatives spécifiques (telles que les vérifications dans l'onglet des références). Mais vous devez vérifier que votre police passe les tests qui examinent les glyphes pour les fonctionnalités requises et plusieurs tests qui recherchent le comportement facultatif mais généralement attendu. Plusieurs des autres tests vous fournissent une rétroaction et des conseils au cours du processus de design, et sont à explorer pour cette raison.

## Tout d'abord: testez les fonctionnalités requises

Dans l'onglet "Points", sélectionnez le test *Coordonnées non-entières*. Ce test vérifie que tous les points de chaque glyphe (y compris les points sur la courbe et les points de contrôle) ont des coordonnées entières. Pas tous les formats de sortie de police requiert ce comportement, mais certains le demandent.

Dans l'onglet "Chemins", sélectionnez les options *Chemins ouverts* et *Chemins extérieurs dans le sens horaire*. Ce sont les deux fonctions obligatoires dans toutes les polices. La première recherche des courbes qui ne sont pas des formes fermées et la seconde assure que les courbes extérieures de chaque glyphe sont tracées dans le sens des aiguilles d'une montre. C'est une très bonne idée de vérifier les *chemins sécants* en même temps; bien que les formats de polices modernes puissent prendre en charge deux chemins croisés, les courbes qui s'entrecroisent ne sont pas autorisées. De plus, si un glyphe a des chemins auto-intersectés, FontForge ne peut pas effectuer le test *Chemins extérieurs dans le sens horaire*.

Dans l'onglet «Références», sélectionnez tous les six tests. Ces vérifications concernent toutes des références, dans lesquelles un glyphe inclut des chemins d'un autre glyphe. Par exemple, une lettre accentuée comprend une référence à la lettre originale (non accentuée), plus une référence au caractère accent. Tous les tests de l'onglet "Références" sont obligatoires pour au moins un format de sortie commun, et tous sont de bonnes idées.

De même, sélectionnez tous les tests dans l'onglet "ATT". Ces tests recherchent des noms de glyphes manquants, des règles de substitution qui font référence à des glyphes inexistantes et d'autres problèmes liés aux noms de glyphe ou aux fonctionnalités OpenType. Les problèmes dont ils nous protègent sont rares, mais cela peut provoquer que la police sera considérée comme invalide par un ou plusieurs systèmes informatiques, donc ils valent la peine d'être inclus.

## Rendez la vie plus facile à vos utilisateurs: testez le bon fonctionnement

Les tests énumérés ci-dessus feront en sorte que votre police s'installe et s'affiche correctement selon les règles établies par les différents formats de police, mais il existe une poignée d'autres tests que vous devriez envisager d'ajouter — surtout à la fin du processus de design — simplement parce qu'ils vérifient les conventions communes suivies par la plupart des typographies modernes.

Dans l'onglet "Points", sélectionnez *Points de contrôle en dehors de la courbe*. Ce test recherchera des points de contrôle situés au-delà des extrémités du segment de courbe sur lequel ils résident. Il ya rarement une raison pour laquelle un point de contrôle devrait se trouver en dehors de la courbe, de sorte que ces cas signifient généralement des accidents. Il est également judicieux de sélectionner *Points trop éloignés*, qui recherche des points situés à plus de 32767 unités du point

le plus proche. Cette distance est plus grande que la plupart des ordinateurs peuvent traiter, et un point qui est loin est presque certainement involontaire (à titre de comparaison, un glyphe simple a tendance à être dessiné sur une grille d'environ 1000 unités), donc enlever ces points est important.

Dans l'onglet "Chemins", les tests *Vérifier les extrema manquants* et *Plus de points que [val]* peuvent être bénéfiques. Le premier cherche des points aux extrema — c'est-à-dire le point le plus haut, le point le plus bas, et les points les plus à gauche et à droite du glyphe. Les formats de police modernes suggèrent fortement que chaque chemin ait un point à chacun de ses extrema horizontaux et verticaux. Cela rend la vie plus facile lorsque la police est rendue à l'écran ou sur la page. Cette vérification recherchera des points extrema manquants. Le deuxième test est une vérification sur le nombre de points au sein d'un glyphe. La valeur par défaut de FontForge pour cette vérification est de 1 500 points, ce qui correspond à la valeur proposée par la documentation PostScript, et elle est suffisante pour presque toutes les polices.

Comme son nom l'indique, l'onglet "Divers" répertorie les tests qui ne correspondent pas aux autres catégories. Parmi ceux-ci, les trois derniers sont utiles: *Codes Unicode répétés*, *Noms répétés* et *Incohérences nom/code Unicode*. Ils recherchent des erreurs de métadonnées dans le mappage entre les noms de glyphe et les cases Unicode.

## Aidez-vous: exécutez des tests qui peuvent aider au design

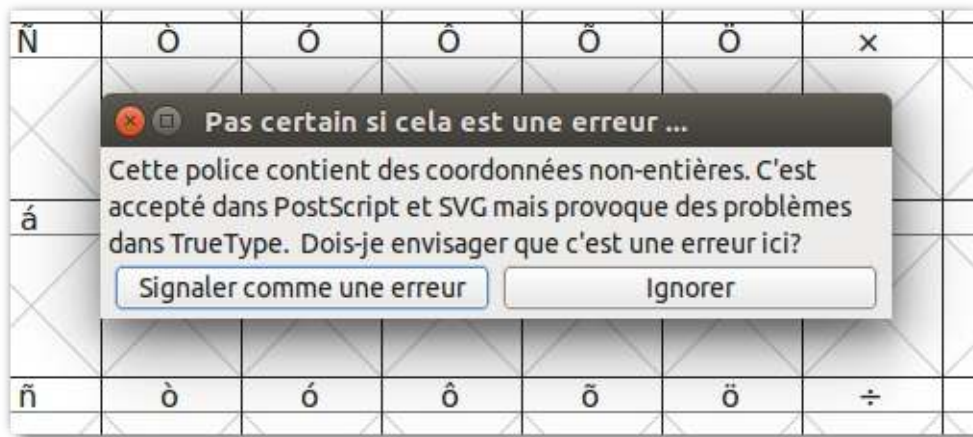
La plupart des autres tests de l'outil de recherche de problèmes peuvent être utiles pour trouver et localiser des incohérences dans votre collection de glyphes; des choses qui ne sont pas mauvaise ou invalide, mais que vous en tant que concepteur voudrez améliorer. Par exemple, le test *Y proche de la hauteur standard* dans l'onglet "Points" compare les glyphes à un ensemble de mesures verticales utiles: la ligne de base, la hauteur du glyphe "x", le point le plus bas de la descendante sur la lettre "p", et ainsi de suite. Dans une police de caractères cohérente, la plupart des lettres vont adhérer à au moins quelques-unes de ces mesures standard, donc les chances sont qu'un glyphe qui est très éloigné de celles-ci ait besoin de beaucoup de travail.

Le test *Bords presque horizontaux/verticaux/italiques* dans l'onglet "Chemins" recherche des segments de ligne qui sont presque exactement horizontaux, verticaux ou à l'angle de l'italique de la police. Faire vos lignes presque verticales parfaitement verticales signifie que les formes apparaîtront nettes lorsque la police est utilisée, et ce test est un moyen fiable pour traquer les segments pas tout à fait-droit qui pourraient être difficiles à repérer à l'œil nu.

Vous pouvez utiliser d'autres tests pour localiser des points de courbe qui sont trop proches les uns des autres pour être significatif, pour comparer les approches latérales des glyphes de même forme et pour effectuer une gamme d'autres tests qui révèlent les glyphes ayant des bizarreries. Une partie du processus de raffinement est de prendre vos designs initiaux et de les rendre plus précis; comme les autres aspects de la conception de police, c'est une tâche itérative, donc utiliser les outils intégrés réduit une partie de la répétition.

## Validez la police

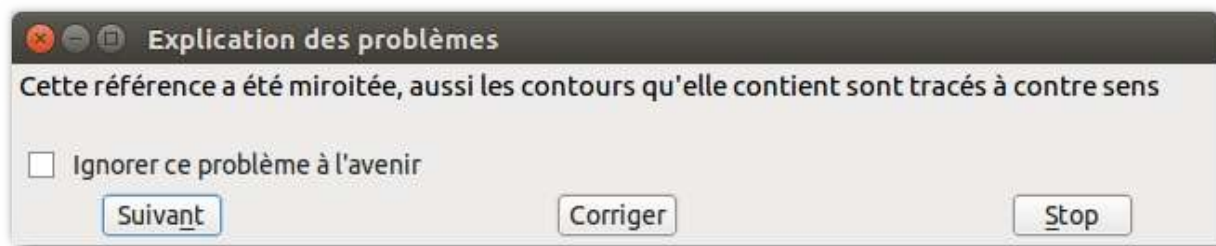
L'autre outil de validation de FontForge est le valideur de la police, qui exécute une batterie de tests et vérifie la police en entier. Parce que le valideur est utilisé pour examiner une police complète, vous ne pouvez le démarrer qu'à partir de la fenêtre de fonte. Vous le trouverez dans le menu Élément, sous le sous-menu Validation. Le valideur est conçu pour exécuter uniquement les tests qui examinent la police pour l'exactitude technique, essentiellement les tests décrits dans la section "testez les fonctionnalités requises" ci-dessus. Mais il exécute les tests contre la police en entier, et il le fait beaucoup plus rapidement que vous pouvez le faire vous-même glyphe par glyphe à l'aide de l'outil de recherche de problèmes.



La première fois que vous exécutez le validateur au cours d'une session particulière, une boîte de dialogue apparaîtra vous demandant si oui ou non il doit signaler les coordonnées de point non entier comme une erreur. La réponse prudente est de choisir "Signaler comme une erreur", car s'en tenir à des coordonnées entières est une bonne pratique de conception. Lorsque le validateur termine son scan de la police (ce qui ne prendra que quelques secondes), il ouvrira une nouvelle boîte de dialogue intitulée Validation de *quel que soit le nom de votre police*. Cette fenêtre répertorie tous les problèmes trouvés par le validateur, présentés dans une liste triée par glyphe.



Mais cette fenêtre n'est pas simplement une liste d'erreurs: vous pouvez double-cliquer sur chaque élément de la liste, et FontForge ira au glyphe correspondant et mettra en évidence le problème exact, avec une explication de texte dans sa propre fenêtre. Vous pouvez ensuite résoudre le problème dans l'éditeur de glyphe, et l'élément d'erreur associé disparaît immédiatement de la liste d'erreurs du validateur. Dans de nombreux cas, l'erreur sera quelque chose que FontForge peut automatiquement réparer. Dans ces cas, la fenêtre d'explication affichera en bas un bouton "Corriger". Vous pouvez cliquer dessus et effectuer la réparation sans efforts supplémentaires.



Pour certains problèmes, il n'y a pas de correction automatique, mais voir le problème à l'écran vous aidera à le corriger immédiatement. Par exemple, une courbe auto-sécante a un endroit spécifique où le chemin se croise — il peut être trop petit pour que vous puissiez le remarquer d'un coup d'œil, mais le grossissement vous permettra de remodeler le chemin et d'éliminer le problème.

Pour d'autres problèmes, il peut ne pas y avoir un point spécifique auquel l'erreur est localisée. Par exemple, si une courbe est tracée dans la mauvaise direction (c'est-à-dire dans le sens inverse des aiguilles d'une montre), toute la courbe est affectée. Dans les cas où FontForge ne peut pas résoudre automatiquement le problème et il n'y a pas de point spécifique sur le glyphe pour le validateur à mettre en évidence, vous pourriez avoir à rechercher afin de corriger manuellement le problème.

Enfin, il existe des tests effectués par le validateur qui pourraient ne pas être un problème pour le format de sortie final que vous avez à l'esprit — par exemple, le test des coordonnées non entières mentionné précédemment. Dans ces cas, vous pouvez cliquer sur la case à cocher "Ignorer ce problème dans le futur" dans la fenêtre d'explication d'erreur, et supprimer ce message d'erreur particulier lors de futures exécutions de validation.

## Résolvez les problèmes lors de l'édition

La plupart des erreurs que l'outil de recherche de problème et le validateur de la police entière recherchent peuvent être corrigées au cours du processus d'édition. Vous n'avez donc pas besoin de reporter la résolution de problèmes à la fin. Par exemple, le sous-menu *Vue > Afficher* comporte des options qui mettent en évidence les zones problématiques lors de l'édition. Le menu élément contient des commandes comme *Ajouter des extrema* qui ajouteront les points extrema attendus dans la plupart des formats de fichier de sortie et des coches pour indiquer si le chemin sélectionné est orienté dans le sens horaire ou antihoraire. Si vous retournez une forme (horizontalement ou verticalement) dans l'éditeur de glyphe, vous remarquerez que sa direction est inversée automatiquement aussi. Si vous cliquez sur la commande *Corriger direction* dans le menu *Élément*, FontForge corrigera immédiatement l'orientation. Prendre l'habitude de faire de petits correctifs comme ceux-ci lorsque vous travaillez vous permettra d'économiser du temps au cours de la phase de validation finale.

## Le design fonctionne-t-il?

Les polices de caractères peuvent être plus ou moins 'fonctionnelles' dû à ces deux facteurs: facilité de lecture et lisibilité.

La lisibilité signifie que les designs des glyphes sont suffisamment distincts pour être instantanément reconnus. Voici quelques paires qui sont souvent trop similaires:

- La lettre "l" et le chiffre "1"
- La lettre "O" et le chiffre "0"
- La lettre "Z" et le chiffre "2"
- Les chiffres "1" et "7"

La facilité de lecture signifie que tous les glyphes fonctionnent bien ensemble pour une expérience de lecture familière et confortable. La création de documents de test est la meilleure façon de s'assurer de cela. Si vous avez un alphabet complet, vous pouvez composer du texte réel — par exemple en utilisant [FontFriend](#) pour glisser et déposer votre police dans un long article que vous souhaitez lire, puis l'imprimer.

Toutefois, si la police ne contient qu'une fraction de l'alphabet, vous pouvez utiliser un générateur de texte de test tel que [LibreText.org](https://libretext.org/) et n'importe quel traitement de texte, application de publication assistée par ordinateur ou programme d'illustration générale (comme [Inkscape](https://inkscape.org/)) pour créer des documents de test.

## Testez la police dans différents environnements

Lorsque vous testez des polices sur Microsoft Windows, l'extension [Propriétés de police](#) peut être utile pour examiner rapidement les métadonnées internes de la police, telles que les numéros de version.

Si vous installez des polices en développement qui font que Windows se comporte de manière erratique, [John Hudson](#) a décrit comment effacer ces polices corrompues sur TypeDrawers:

Redémarrez Windows en mode console de récupération. Dans la console, accédez au dossier Windows/Fonts et supprimez toutes les entrées de la police Rhodium. Ensuite, naviguez vers Windows/System32 et supprimez le fichier 'FNTCACHE.DAT' (**pas le .dll**), puis redémarrez Windows. Le fichier de cache de police .dat sera reconstruit, puis vous pouvez alors réinstaller une copie propre de la police Rhodium et voir si cela fonctionne. (Ne vous inquiétez pas si vous obtenez toujours un message indiquant que la police est déjà installée: à ce stade, Windows vous ment.)

## Le produit final, génération des polices

Bien que vous puissiez effectuer un large éventail de tests dans FontForge, vous devrez générer des fichiers de polices installables afin d'effectuer des tests réels pendant le processus de développement. En outre, votre objectif ultime est bien sûr de créer une police que vous pouvez rendre disponible dans un format de sortie pour d'autres personnes qui l'installeront et l'utiliseront. Vous utiliserez l'outil *Générer fonte(s)* (qui se trouve dans le menu Fichier) pour compiler une police utilisable, que ce soit pour vos propres tests ou pour publication au grand public. Mais vous devrez employer quelques étapes supplémentaires lors de la compilation du produit fini.

FontForge peut exporter vos polices vers différents formats, mais en pratique seulement deux sont importants: TrueType (qui a une extension de fichier *.ttf*) et OpenType CFF (qui a une extension *.otf*). Techniquement, le format OpenType peut englober toute une gamme d'autres options, mais le format CFF est celui qui est largement utilisé.

### Génération expéditive pour les tests

Pour construire un fichier de police à des fins de test — pour examiner l'espacement dans un navigateur Web par exemple — vous devez seulement vous assurer que votre police passe les tests de validation requis.

Vous pouvez utiliser l'outil *Validation* dans le menu Élément (voir le chapitre sur la validation des polices pour une explication plus détaillée) ou sélectionner tous les glyphes (appuyez sur Control-A ou choisissez "Sélectionner" -> "Tout sélectionner" dans le menu "Édition") puis exécutez quelques commandes pour appliquer des modifications de base en vrac. N'oubliez pas de sauvegarder votre travail avant de continuer. Cependant, certaines des modifications requises pour valider votre police pour l'exportation va modifier les formes de vos glyphes de façon subtile.

Pour les polices OpenType, corrigez d'abord la direction de tous vos chemins. Appuyez sur Contrôle-Maj-D ou choisissez "Corriger direction" dans le menu "Elément". Ensuite, vérifiez que vous n'avez pas laissé de chemins non fermés. Choisissez "Validation>Problèmes..." dans le menu «Elément», sélectionnez l'option *Chemins ouverts* dans l'onglet «Chemins» et cliquez sur OK pour lancer le test. Une fois que votre police passe le test sans erreurs, vous êtes prêt à générer une sortie OpenType.

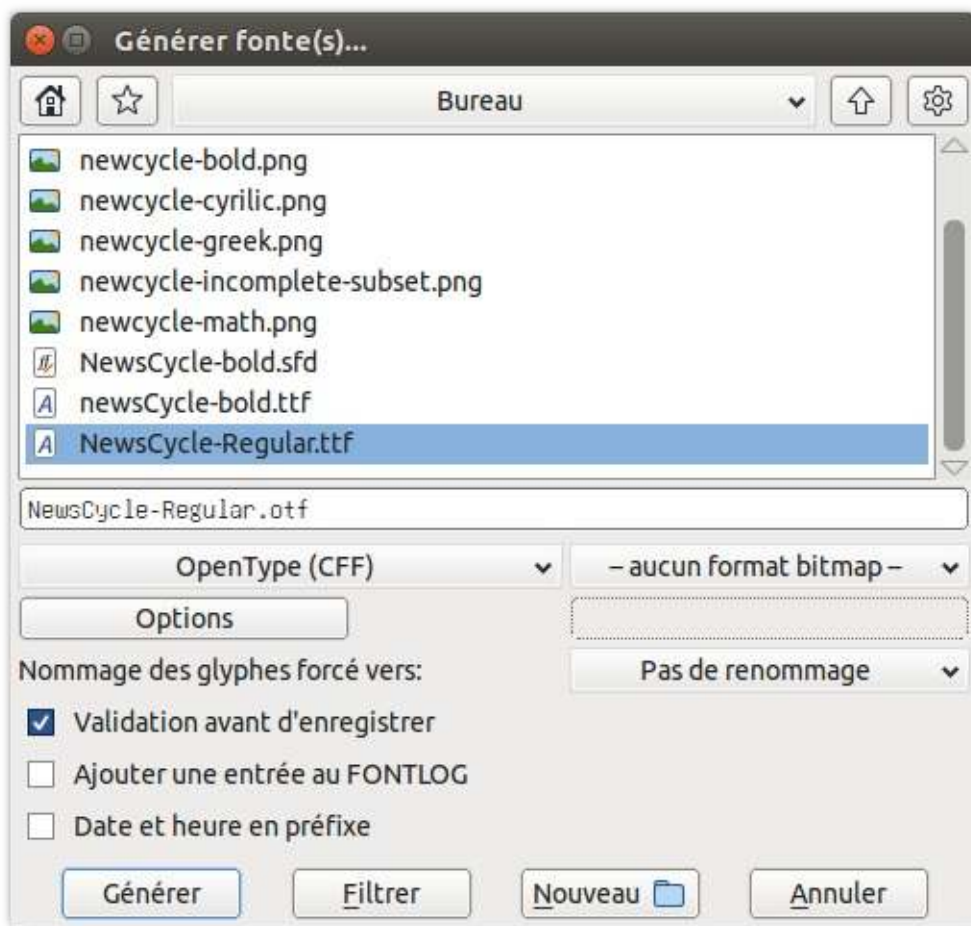
Pour les polices TrueType, quelques étapes supplémentaires sont nécessaires. Vous devez d'abord corriger la direction de tous vos chemins comme décrit ci-dessus. Ensuite, ajustez tous les points pour qu'ils aient des coordonnées entières: soit appuyez sur Control-Maj- (soulignement), ou choisissez *à l'entier* dans le menu "Element" -> "Arrondir". Enfin, ouvrez l'outil "Recherche de problèmes", sélectionnez le test *Chemins ouverts* comme décrit ci-dessus, puis sélectionnez tous les tests dans l'onglet "Références".

Afin que vous puissiez exécuter ces tests sans erreurs, vous devriez alors convertir vos chemins en courbes quadratiques. Ouvrez la fenêtre "Infos fonte" dans le menu "Élément". Cliquez sur l'onglet "Calques" et cochez l'option *Tous les calques quadratiques*. Cliquez sur OK en bas de la fenêtre et vous êtes prêt à générer une sortie TrueType.

### Compilation des fichiers de polices

Ouvrez la fenêtre *Générer fonte(s)* en la choisissant dans le menu "Fichier". La moitié supérieure de la fenêtre affiche des options familières de sélection de fichier, une liste des fichiers trouvés dans le répertoire courant, une zone de texte pour entrer un nom de fichier et des boutons pour naviguer vers d'autres dossiers et répertoires si nécessaire. C'est strictement un moyen pour vous aider à trouver rapidement le bon endroit pour enregistrer votre fichier de sortie, ou de choisir un fichier de police existant si vous avez l'intention de remplacer une sauvegarde précédente. Toutes les options que vous devez examiner se trouvent dans la moitié inférieure de la fenêtre.





Sur la gauche, un menu déroulant vous permet de sélectionner le format de la police que vous souhaitez générer. Vous devez choisir *TrueType* ou *OpenType (CFF)*, comme indiqué précédemment. Sur le côté droit, assurez-vous que *Aucun format Bitmap* est sélectionné. Sur la ligne en-dessous, assurez-vous que *Pas de renommage* est sélectionné pour l'option "Nomme des glyphes forcé vers:". Vous pouvez cocher l'option "Validation avant d'enregistrer" si vous le souhaitez (pour potentiellement attraper d'autres erreurs), mais cela est facultatif. Laissez les options "Ajouter une entrée au FONTLOG" et "Date et heure en préfixe" désactivées.

Cliquez sur le bouton "Générer", et FontForge va compiler votre fichier de polices. Vous pouvez charger la police dans d'autres applications et exécuter des tests, mais quand vous êtes prêt à revenir à l'édition, n'oubliez pas de rouvrir la version enregistrée de votre police que vous avez créée avant de générer votre *.ttf* ou *.otf*</em>.

## Génération de la version finale

La conception de votre police est un processus itératif, mais le jour viendra où vous devrez déclarer votre police terminée — ou au moins prête pour distribution au public. À ce stade, vous allez à nouveau générer un fichier de sortie *.ttf* ou *.otf* (peut-être même les deux), mais avant cela, vous devrez effectuer quelques étapes supplémentaires pour créer la version la plus conforme aux normes et la plus conviviale de votre fichier de police.

Tout d'abord, suivez les mêmes étapes de préparation que décrites dans la section sur la génération expéditive pour les tests. En particulier, n'oubliez pas de changer votre police à *Tous les calques quadratiques* si vous créez un fichier TrueType.

## Supprimer les chevauchements

Comme vous le savez, il est judicieux de conserver vos lettres sous forme de combinaisons de composants distincts que vous concevez: fûts, panses, empattements et autres morceaux de chaque glyphe. Mais bien que cette technique soit idéale pour la conception et le raffinage des formes, vous voulez que votre police finale et publiée ait des glyphes avec contours simples à la place. Cela réduit la taille du fichier un peu, mais surtout réduit les erreurs de rendu.

FontForge dispose d'une commande *Supprimer les recouvrements* qui combinera automatiquement les composants séparés d'un glyphe en un seul contour. Sélectionnez un glyphe (ou même sélectionnez tous les glyphes avec Control-A), puis appuyez sur Control-Maj-O ou choisissez Enlever les recouvrements dans le menu "Élément" -> "Recouvrements". Cependant, une mise en garde: FontForge ne peut pas fusionner des formes si l'une des formes est tracée dans la mauvaise direction (c'est-à-dire si le chemin le plus à l'extérieur est dans le sens anti-horaire). Un chemin tracé dans la mauvaise direction est une erreur en elle-même que vous devriez réparer de toute façon.

## Simplifier les contours et ajouter des points extrema

Vous devriez également simplifier vos glyphes si possible — sans éliminer les détails, mais éliminer les points redondants. Cela réduit la taille de chaque glyphe, ce qui réduit considérablement la taille de l'ensemble des caractères de la police.

Dans le menu "Élément", choisissez "Simplifier" -> *Simplifier* (ou appuyez sur Control-Maj-M). Cette commande fusionnera les points redondants sur la courbe dans tous les glyphes sélectionnés. Dans certains cas, seulement quelques points seront enlevés, dans d'autres il peut y en avoir beaucoup. Mais la simplification devrait se faire sans modifier sensiblement la forme des glyphes. Si vous constatez qu'un glyphe particulier est trop modifié par *Simplifier*, n'hésitez pas à annuler l'opération. Vous pouvez également expérimenter avec la commande *Simplifier d'avantage* située dans le même menu; elle offre des paramètres modifiables qui pourraient s'avérer utiles.

Dans tous les cas, après avoir terminé l'étape de simplification, vous devez ajouter tous les points extrema absents. Choisissez *Ajouter des extrema* dans le menu "Élément" (ou appuyez sur Contrôle-Maj-X). Comme indiqué précédemment, il est judicieux de placer des points sur courbe à l'extrémité de chaque glyphe durant sa création. Néanmoins, vous devriez tout de même effectuer cette étape lors de la préparation de la génération de sortie finale, car l'étape *Simplifier* éliminera occasionnellement un point extrema.

## Tout arrondir à des coordonnées entières

L'étape de préparation finale à exécuter consiste à arrondir tous les points (points de courbe et points de contrôle) à des coordonnées entières. Ceci est obligatoire pour générer une sortie TrueType, mais est également fortement recommandé pour la sortie OpenType. Il peut en résulter un rendu plus précis et un meilleur ajustement à la grille lorsque les polices sont affichées, sans effort de design supplémentaire.

Pour arrondir tous les points à des coordonnées entières, choisissez "Élément" -> "Arrondir" -> à *l'entier*.

Dès que cette opération est terminée, vous pourriez remarquer quelque chose de déroutant. Parfois, simplement en raison des particularités des courbes impliquées, les processus d'arrondissement aux coordonnées entières, de simplification des glyphes et d'ajout d'extrema manquants peuvent être opposés. Cela peut se produire par exemple quand un bord extérieur incurvé a un point de contrôle qui se trouve juste devant l'horizontale ou la verticale. Dans cette situation, l'arrondissement à des coordonnées entières peut décaler la courbe légèrement et changer la position des extrema.

Il n'y a pas de solution unique à ce dilemme. Le seul correctif garanti est de répéter le cycle d'étapes pour les glyphes affectés jusqu'à ce qu'ils se stabilisent à un point où les trois opérations n'interfèrent plus les uns avec les autres. Cela peut prendre plusieurs cycles, mais c'est rare.

## Valider

Votre police doit passer les tests de validation nécessaires avant de générer votre sortie finale. Comme dans le cas de l'étape de l'arrondissement des coordonnées à des entiers, les autres opérations préparatoires peuvent parfois introduire des erreurs. Il est donc toujours judicieux d'exécuter le validateur de la police entière à ce stade avant de compiler la sortie finale. Le chapitre sur les outils de validation de FontForge vous donnera plus de détails sur ce qu'il faut vérifier.

## Un mot sur le hinting

Le hinting fait référence à l'utilisation d'instructions mathématiques pour afficher les courbes vectorielles dans une police de telle sorte qu'elles s'alignent bien avec la grille de pixels du dispositif de sortie (que cette grille soit composée de points d'encre ou de toner sur papier, ou luminescents sur un moniteur d'ordinateur).

FontForge vous permet d'ajouter des hints à votre police (et même fournit une fonction *Autohint*), mais en pratique cette étape n'est pas strictement nécessaire. Les systèmes d'exploitation modernes ont souvent une meilleure fonctionnalité d'ajustement à la grille dans leurs moteurs de rendu de texte que celle fournie par les hints que vous auriez créé avec beaucoup de temps et d'efforts. En fait, Mac OS X et Linux *ignorent* tous les hints incorporés dans le fichier de police. Si vous décidez que votre police requière du hinting pour les utilisateurs de Windows, il est préférable de construire la police sans hint, puis d'utiliser une application spécialisée telle que **ttfautohint** pour ajouter du hinting après le fait.

Pour le hinting CFF, consultez [cette vidéo](#) d'Adobe chez RoboThon.

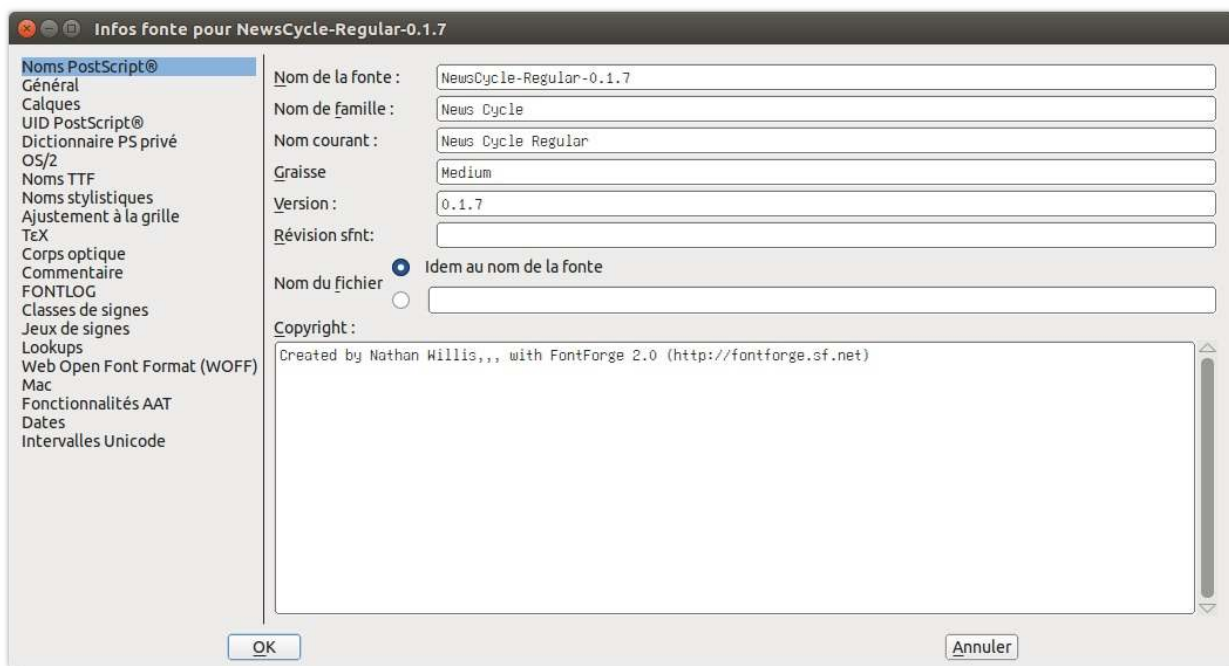
Configurer le hinting PS avec Python est possible: `private` est une liste de tuples (Merci [Sungsi!](#))

```
font.private['BlueValues'] = (-20, 0, 600, 620, 780, 800, 810, 830)
font.private['OtherBlues'] = (-225, -210)
font.private['StdHW'] = 100,
font.private['StdVW'] = 137,
```

## Vérifiez vos métadonnées

Enfin, une fois que votre police a été soigneusement préparée techniquement pour l'exportation, vous devriez prendre une pause et mettre à jour les métadonnées, en veillant à ce que des informations importantes soient incluses et à jour.

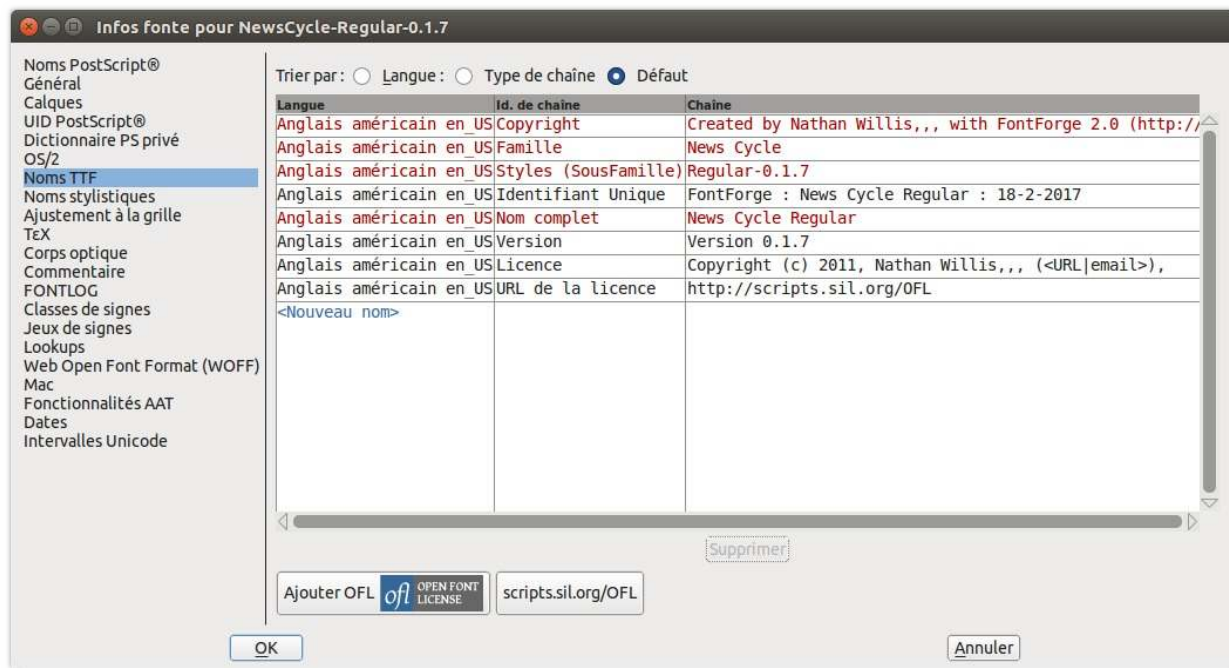
Tout d'abord, s'il s'agit de la version initiale de votre police, ouvrez la boîte de dialogue *Infos fonte* de la fenêtre "Élément" et sélectionnez l'onglet "Noms Postscript". Remplissez d'abord le nom de famille et la graisse de la fonte, puis copiez ces informations dans la case "Nom courant". Bien que l'utilisation de numéros de version ne soit pas nécessaire, il est extrêmement utile pour vous en tant que concepteur de différencier entre les différentes révisions de votre travail. Entrez "1.0" comme numéro de "Version" si vous n'êtes pas sûr. Ensuite, visitez l'onglet "Noms TTF" et entrez les mêmes informations.



Comme c'est le cas pour les numéros de version, il est utile, à long terme, de créer des entrées de journal pour chaque révision. Allez dans l'onglet "FONTLOG" et écrivez une ou deux phrases pour expliquer les changements éventuels dans la révision que vous créez pour distribution. Si c'est votre première entrée de journal, vous devriez également décrire votre

fonte et son but dans une phrase ou deux.

Les polices, comme toutes les œuvres de création, doivent disposer d'une licence, afin que les utilisateurs sachent ce qu'ils sont et ne sont pas autorisés à faire. FontForge a un bouton dans l'onglet "Noms TTF" appelé "Ajouter Open Font License". La licence Open Font (OFL) est une licence de police conçue pour vous permettre de partager votre police avec le public avec très peu de restrictions sur la façon dont elle est utilisée, tout en vous protégeant en tant que concepteur d'autres prenant le crédit pour votre travail ou des dérivés créatifs de votre police qui seront confondus pour l'original. Cliquez sur le bouton pour ajouter les chaînes "Licence" et "URL de la licence" aux métadonnées des noms TTF. Si vous avez une autre licence que vous préférez utiliser au lieu de l'OFL, saisissez-la dans le champ "Licence" à la place.



Si vous avez apporté des modifications importantes à d'autres fonctionnalités de votre police, il est judicieux de vérifier les autres paramètres dans la fenêtre Infos fonte et de vous assurer que tout est toujours à jour. Les informations d'espacement linéaire, par exemple, se trouvent dans l'onglet "OS/2" sous "Métriques".

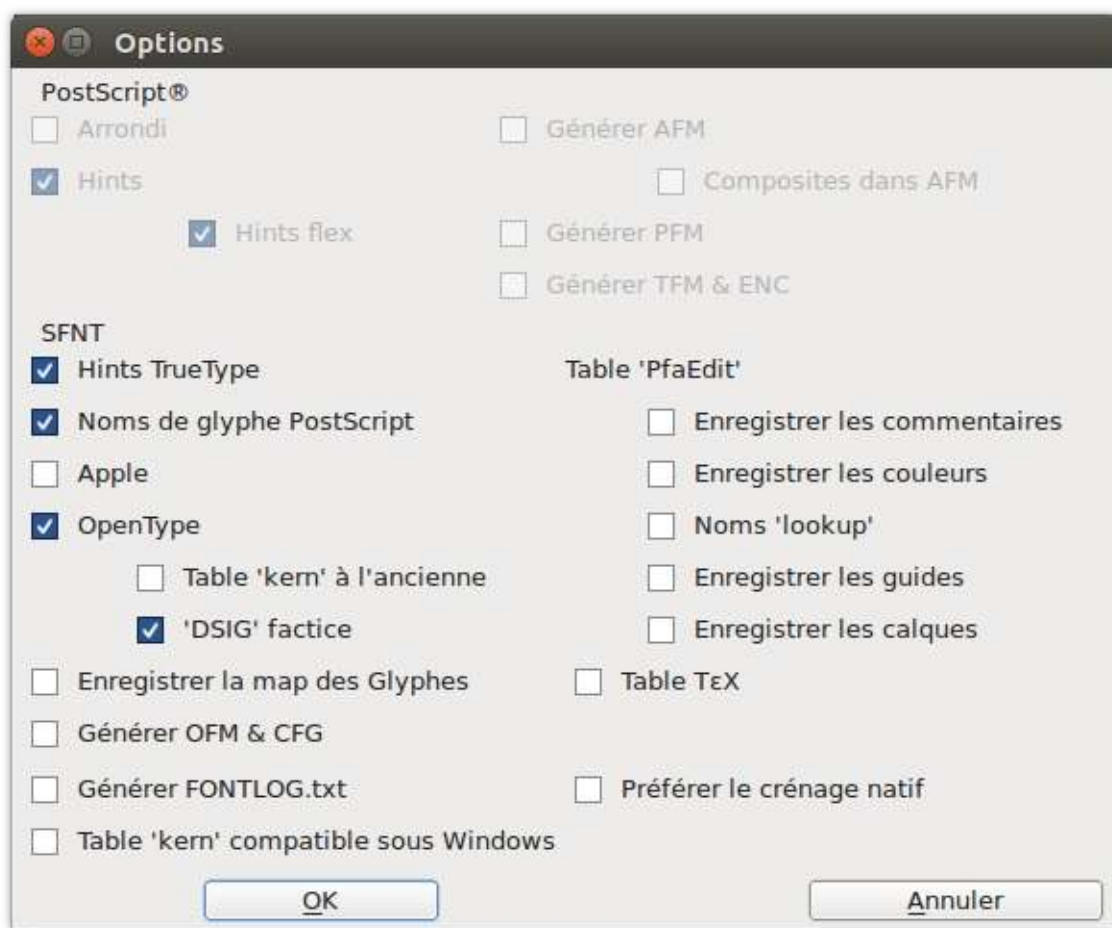
## Compilation des fichiers de polices

Le processus de génération des fichiers de police est le même lorsque vous créez la version finale comme c'est le cas lorsque vous créez une copie expéditive pour le test, mais vous voudrez accorder une plus grande attention à certaines des options.

Ouvrez la fenêtre *Générer fonte(s)* en la choisissant dans le menu "Fichier". Encore une fois, la moitié supérieure de la fenêtre vous permet de choisir le répertoire et le nom du fichier à donner à votre fichier de sortie — faites attention de ne pas écraser une sauvegarde précédente.

Dans le menu déroulant de gauche, sélectionnez le format de la police que vous générez, *TrueType* ou *OpenType (CFF)*, comme indiqué précédemment. Sur le côté droit, assurez-vous que *aucun format Bitmap* est sélectionné. Sur la ligne en-dessous, assurez-vous que *Pas de renommage* est sélectionné pour l'option "Nommage des glyphes forcé vers:". Vous pouvez cocher l'option "Validation avant d'enregistrer" si vous le souhaitez (pour potentiellement attraper d'autres erreurs), mais cela est facultatif. Laissez les options "Ajouter une entrée au FONTLOG" et "Date et heure en préfixe" désactivées.

Ensuite, cliquez sur le bouton "Options". Sélectionnez les options *Noms de glyphe Postscript*, *OpenType* et *DSIG factice* dans la fenêtre qui s'affiche et désélectionnez tout le reste.



Cliquez sur le bouton "Générer", et FontForge va compiler votre fichier de polices. Un dernier mot: il est important de ne pas écraser la version sauvegardée de votre travail dans FontForge avec les modifications que vous avez apportées dans cette section uniquement pour générer votre sortie *.ttf* ou *.otf*. Par exemple, vous perdez beaucoup de composants de glyphes individuels lorsque vous effectuez l'opération *Supprimer les recouvrements*. Mais la prochaine fois que vous reprendrez le travail sur votre police, vous voudrez certainement reprendre où vous avez laissé dans la version originale, la version remplie de glyphes avec composants individuels.

Par conséquent, si vous décidez d'enregistrer la version modifiée de votre fichier FontForge, assurez-vous que vous la renommez de manière mémorable, comme *MaPolice-TTF.sfd* ou *MaPolice-OTF.sfd* *</Em>*. *Mais vous n'avez pas forcément besoin de sauvegarder ces variations faites pour la sortie de votre fichier — en pratique, la prochaine fois que vous réviserez votre travail original dans FontForge, vous travaillerez de nouveau à la préparation de la sortie.*

Félicitations! Vous avez maintenant créé votre première police. Tout ce qui vous reste à faire désormais est de partager votre travail: le télécharger sur le web, l'afficher sur votre blog, et le dire à vos amis.

Sans doute, vous reviendrez et continuerez à réviser et à affiner votre police de caractères — après tout, comme vous l'avez vu, le design des polices est un processus très itératif. Mais assurez-vous de faire une pause et de profiter de ce que vous avez accompli d'abord.

## Quand les choses tournent mal avec FontForge

FontForge est développé sur Github. L'équipe de FontForge utilise la section *Issues* de Github pour discuter des problèmes, des erreurs et des idées d'améliorations, puis quelqu'un développe une solution et la propose en tant que *Pull Request*.

Pour en savoir plus sur Github, consultez les [bonnes ressources pour l'apprentissage de Git et GitHub](#) (en anglais).

## Payer pour du soutien

Cela pourrait être une surprise, mais il est à la fois possible, et encouragé de payer pour du soutien lorsque les choses tournent mal.

Alors que d'autres éditeurs de polices avec des fonctionnalités complètes semblables coûtent des centaines de dollars américains, si nous payions chacun un montant similaire aux développeurs de FontForge pour obtenir que nos bogues les plus ennuyeux soit corrigés, FontForge deviendrait de mieux en mieux.

[FreedomSponsors.org](#) offre un système transparent pour inciter les développeurs de FontForge à résoudre rapidement les problèmes qui vous préoccupent.

1. Créez un problème (issue) sur la page FonForge de Github décrivant ce que vous désirez voir changer (voir ci-dessous). Copiez l'URL du problème dans le presse-papiers.
2. Visitez FreedomSponsors et parrainez un nouveau problème, en utilisant l'URL que vous avez copiée plus tôt.
3. Révisez le problème sur Github et ajoutez en commentaire le lien vers la page de problèmes de FreedomSponsors, avec une note personnelle que vous offrez une prime payée pour que ce problème soit fermé

## Signaler un bogue

1. Connectez-vous à Github (après avoir créé un compte, si vous n'en avez pas encore) et visitez la page de [suivi des problèmes de FontForge](#).
2. Dans la zone de recherche des problèmes, essayez de rechercher des problèmes similaires, pour voir si le problème auquel vous êtes confronté a déjà été signalé. Si c'était le cas, et que votre problème est lié, mais pas tout à fait le même, veuillez commenter ce problème avec votre propre opinion sur la question.
3. Si ce n'est pas déjà signalé, ouvrez un nouveau problème. Cliquez sur le bouton vert "New Issue", puis décrivez votre question, ce que vous avez fait pour déclencher un plantage ou votre idée d'amélioration.

Inclure les détails pertinents, tels que:

- Votre système d'exploitation et sa version;
- La version de FontForge et d'où elle provient;
- **Ce qui se passe, étape par étape, pour reproduire le problème;**
- **Quels messages d'erreur vous voyez;**
- **Le comportement auquel vous vous attendez**

Vous pouvez glisser-déposer des captures d'écran ou d'autres images directement dans la page de problème pour les inclure.

Une façon facile de signaler des problèmes est d'enregistrer des vidéos d'écran où vous faites la narration des choses qui vous intéressent comme elles se produisent, puis le télécharger sur YouTube et inclure un lien vers votre vidéo

Pour reproduire le problème, il peut être utile de partager avec la communauté des développeurs les fichiers avec lesquels vous travaillez. Si vous pouvez faire un fichier petit qui ne contient que ce qui est nécessaire pour reproduire le problème, s'il vous plaît créez un fork du dépôt de fontforge et ajoutez ces fichiers à /tests/fonts et soumettez un "pull request". Vous

pouvez également placer des fichiers sur votre propre site Web ou sur un service de partage de fichiers temporaires (comme MegaUpload, DropBox, Google Drive, etc.). Enfin, si vous ne souhaitez pas rendre vos fichiers accessibles au public, vous pouvez fournir une adresse e-mail pour un développeur de FontForge qui vous contactera pour obtenir une copie privée du fichier.

S'il vous plaît ne pas fermer les problèmes des autres — demandez-leur de fermer le problème s'il est résolu à leur satisfaction.

## Comment signaler un plantage

Le processus est le même pour signaler un plantage ou d'autres types de bogues que pour les nouvelles fonctionnalités ou questions. L'envoi d'un bon rapport de plantage aux développeurs de FontForge peut vraiment les aider beaucoup à améliorer la stabilité du programme pour tout le monde! N'hésitez pas à signaler de tels problèmes, car un plantage qui n'est pas signalé est un plantage qui est beaucoup moins susceptible d'être corrigé.

Si vous voyez FontForge planter en cours d'utilisation, créez un rapport de problème comme ci-dessus. Si vous avez un fichier de police particulier (SFD, UFO, OTF, TTF, etc.) qui déclenche le plantage, vous pouvez soit le télécharger dans un nouveau dépôt Github vous-même (ou Dropbox ou autre) et inclure un lien, ou soit publier votre email et demandez à un développeur de vous envoyer un courriel pour obtenir une copie en privé.

Avec votre description, les développeurs de FontForge essayeront de reproduire le plantage. S'ils peuvent le faire, ils seront en mesure de déterminer où le code va mal, et de créer un correctif.

Une fois que la *pull request* résolvant le problème fusionné, vous aurez besoin d'obtenir une nouvelle version. Vous pouvez effectuer l'une des opérations suivantes:

- Recompiler à partir du dernier code source de Github (voir [Installation de Fontforge](#)),
- Vérifier si une compilation quotidienne est disponible (souvent possible pour [Mac OS X](#)), ou
- Attendre la prochaine version (souvent en quelques semaines).

## Les meilleurs rapports de plantage

Pour aider les développeurs à découvrir ce qui ne va pas et **vraiment** comprendre comment le réparer, vous pouvez faire un peu plus de travail et faire un *backtrace*. Un backtrace inclut une liste des fonctions du programme qui ont appelé les autres pour arriver là où le programme a cessé de fonctionner. Un backtrace est très utile s'il contient également les numéros de ligne des fonctions.

Pour faire un backtrace, vous devrez installer à partir de source avec *information de débogage* incluse. Utilisez les commandes `type` et `nm` pour trouver le chemin et l'état de vos binaires fontforge. Exemple:

```
$ type -all fontforge;
fontforge is /usr/bin/fontforge
$ nm /usr/bin/fontforge;
nm: /usr/bin/fontforge: no symbols
$
```

Dans cet exemple, nous voyons `no symbols`, nous devons donc mettre à jour notre installation pour y inclure des informations de débogage.

## Installer les informations de débogage sur Fedora

Fedora propose dans le dépôt standard une commande permettant d'installer facilement des informations de débogage pour FontForge. (Mais notez que cela peut nécessiter des centaines de mégaoctets de téléchargement si vous ne disposez pas déjà des nombreux paquets debuginfo dépendants). Pour l'installer, lancez:

```
debuginfo-install fontforge;
```

A FAIRE: expliquer comment inclure des informations de débogage à partir de sources compilées

Un backtrace est généré à l'aide du débogueur de projet GNU, `gdb`. Vous pouvez soit attacher `gdb` à FontForge déjà en cours, soit démarrer FontForge à l'intérieur de la session `gdb` elle-même. Voici un exemple de cette dernière méthode:

```
$ gdb fontforge;
GNU gdb (GDB) Fedora (7.3.50.20110722-16.fc16)
Copyright (C) 2011 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-redhat-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /usr/local/bin/fontforge...done.
```

Ensuite, une fois que vous lancez le débogueur avec la commande `run`, FontForge s'ouvrira à l'écran:

```
(gdb) run
Starting program: /usr/local/bin/fontforge
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib64/libthread_db.so.1".
Copyright (c) 2000-2012 by George Williams.
  Executable based on sources from 14:57 GMT 31-Jul-2012-ML-TtfDb-D.
  Library based on sources from 14:57 GMT 31-Jul-2012.
```

De là, vous pouvez utiliser FontForge de la manière habituelle, mais avec l'avantage d'être en mesure de capturer efficacement et de signaler tous les problèmes que FontForge peut avoir.

L'avantage majeure d'exécuter FontForge à l'intérieur de `gdb` est de savoir comment un plantage est arrivé. Sans `gdb`, lorsque FontForge plante, il disparaîtra de votre écran. Lorsque vous exécutez FontForge à l'intérieur de `gdb` cependant, un FontForge planté restera ouvert avec ses fenêtres et son interface utilisateur.

Si vous constatez que votre interface ne répond pas, retournez au terminal où vous avez exécuté `gdb` et vous pourrez voir quelque chose comme `SIGSEGV` dans le texte suivi de l'invite `(gdb)`. Si vous voyez l'invite `(gdb)`, FontForge ne s'exécute plus.

Vous pouvez maintenant (enfin!) utiliser la commande `bt` pour obtenir un backtrace, puis utiliser la commande `gdb quit` pour quitter `gdb` et fermer la session plantée de FontForge. Voici un exemple:



```

Program received signal SIGSEGV, Segmentation fault.
0x00007ffff74a7c01 in ?? () from /lib/x86_64-linux-gnu/libc.so.

(gdb) bt
#0 0x00007ffff74a7c01 in ?? () from /lib/x86_64-linux-gnu/libc.so.6
#1 0x00007ffff6389a80 in copy (str=0x900000000) at memory.c:82
#2 0x00007ffff7a4aeb5 in KCD_AutoKernAClass (kcd=kcd@entry=0xe80c40, index=2, is_first=is_first@entry=1)
  at kernclass.c:236
#3 0x00007ffff7a51405 in KCD_FinishEdit (g=0xeb0fe0, r=1, c=, wasnew=1) at kernclass.c:2020
#4 0x00007ffff5effe2d in GME_SetValue (gme=gme@entry=0xeb0fe0, g=0xe94760) at gmatrixedit.c:988
#5 0x00007ffff5f00554 in GME_FinishEdit (gme=0xeb0fe0) at gmatrixedit.c:997
#6 0x00007ffff5f01c1a in GMatrixEditGet (g=g@entry=0xeb0fe0, rows=rows@entry=0x7ffffffcf78)
  at gmatrixedit.c:2214
#7 0x00007ffff7a4ea3c in KCD_Expose (event=0x7fffffffd1e0, pixmap=0x83ae00, kcd=0xe80c40)
  at kernclass.c:1446
#8 kcd_e_h (gw=0x83ae00, event=0x7fffffffd1e0) at kernclass.c:1762
#9 0x00007ffff5eabe8f in _GWidget_Container_eh (gw=gw@entry=0xe7f040, event=event@entry=0x7fffffffd1e0)
  at gcontainer.c:269
#10 0x00007ffff5eac385 in _GWidget_TopLevel_eh (event=0x7fffffffd1e0, gw=0xe7f040) at gcontainer.c:734
#11 _GWidget_TopLevel_eh (gw=0xe7f040, event=0x7fffffffd1e0) at gcontainer.c:606
#12 0x00007ffff5ef86ce in GXDrawRequestExpose (gw=0xe7f040, rect=0xef72b0, doclear=)
  at gxdraw.c:2687
#13 0x00007ffff5eea075 in gtextfield_focus (g=0xef72a0, event=0x7fffffffd2e0) at gtextfield.c:1888
#14 0x00007ffff5eaa857 in _GWidget_IndicateFocusGadget (g=0xe94760, mf=mf@entry=mf_normal)
  at gcontainer.c:143
#15 0x00007ffff5eaac97 in GWidgetIndicateFocusGadget (g=) at gcontainer.c:155
#16 0x00007ffff5f02b1e in GME_StrSmallEdit (event=0x7fffffffd670, str=0xe10e60 "A", gme=0xeb0fe0)
  at gmatrixedit.c:890
#17 GMatrixEdit_StartSubGadgets (gme=gme@entry=0xeb0fe0, r=1, c=c@entry=0, event=event@entry=0x7fffffffd670)
  at gmatrixedit.c:1472
#18 0x00007ffff5f03d69 in GMatrixEdit_MouseEvent (event=0x7fffffffd670, gme=0xeb0fe0) at gmatrixedit.c:1499
#19 matrixeditsub_e_h (gw=, event=0x7fffffffd670) at gmatrixedit.c:1735
#20 0x00007ffff5eabd98 in _GWidget_Container_eh (gw=0xe7f040, event=0x7fffffffd670) at gcontainer.c:393
#21 0x00007ffff5ef6555 in dispatchEvent (gdisp=gdisp@entry=0x769a50, event=event@entry=0x7fffffffd9b0)
  at gxdraw.c:3475
#22 0x00007ffff5ef7d1e in GXDrawEventLoop (gd=0x769a50) at gxdraw.c:3574
#23 0x00007ffff7ad353a in fontforge_main (argc=, argv=) at startui.c:1196
#24 0x00007ffff736676d in __libc_start_main () from /lib/x86_64-linux-gnu/libc.so.6
#25 0x0000000004006e1 in _start ()
(gdb) quit
A debugging session is active.

        Inferior 1 [process 19196] will be killed.

Quit anyway? (y or n) y

```

Un développeur peut voir dans cet exemple de backtrace que FontForge a planté à l'intérieur de la fonction `copy ()`. La fonction `copy ()` a elle-même été appelée à partir de la fonction `KCD_AutoKernAClass`. Le backtrace indiquera à un développeur les lignes exactes de ces appels et suggérera que le paramètre passé à `copy ()` était non valide (hors limites) pour déterminer ce que le code ne fait pas correctement.

# Conception de polices Devanagari

*Merci à Adam Twardoch, Erin McLaughlin, Neelakash Kshetrimayum, Dan Reynolds, Pooja Saxena et au Dr Girish Dalvi pour leur contribution à beaucoup des idées de cette page*

La conception d'une police [Devanagari](#) nouvelle et originale suit un processus similaire au processus de design d'une police latine. L'avantage unique du *libre* dans les polices libres est que vous pouvez les modifier et les réutiliser à des fins nouvelles à lesquelles leurs créateurs initiaux n'ont jamais pensées — par exemple, concevoir une police Devanagari et adapter une police latine existante pour l'accompagner.

## Les glyphes Devanagari

Les polices Devanagari contiennent ces différents types de glyphes:

- Consonnes (36)
- Voyelles indépendantes (28)
- Voyelles matras
- Espace(s) de mots
- Chiffres Devanagari (10)
- Les chiffres latins (nouveaux, ou si déjà présents puis ajustés pour fonctionner dans le texte pur Devanagari)
- Diacritiques nuqtā
- Demi-glyphes
- Conjonctions (glyphes de ligature uniques)
- Voyelles "I" matras de différentes longueurs
- Ponctuation Devanagari, marques et symboles
- La ponctuation latine, les marques et les symboles (nouveaux ou ajustés si déjà présents)
- Lettres latines

Consultez le [chapitre 12 d'Unicode sur les écritures indiennes](#) (en anglais), ([la page Devanagari d'Unicode](#)), ainsi que la ([page de développement de polices OpenType Devanagari de Microsoft](#) (en anglais) pour en savoir plus sur ces glyphes et comment le moteur de mise en forme indien fonctionne.

Il est utile de faire de la calligraphie ou d'étudier de près les manuels d'écriture pour apprendre comment l'écriture fonctionne, de sorte que vous compreniez quelles lettres devraient être comme quelles autres lettres dans la structure. Ces [2 pages du Manuel de calligraphie de Devanagari d'Aksharaya](#) peuvent être utilisées comme référence pour l'angle de la plume et les proportions de lettre.

## Que faire d'abord

Lors de la conception d'une police de caractères Devanagari et latine, il est important de commencer par dessiner le latin à côté du Devanagari. Dans les premiers stades, il faut concevoir les glyphes "clés", pour établir la personnalité de la police de caractères à travers les formes fondamentales et l'espacement (qui en latin peut être 'adhesion' ou 'videospans').

Concevez les glyphes les plus bas et les plus hauts "en hauteur" tôt dans le processus.

Vous aurez besoin de nombreuses voyelles pour commencer à tester la texture et l'échelle.

Le professeur de typographie à IIT Bombay, le Dr Girish Dalvi, a écrit dans sa thèse de doctorat,

Grâce aux résultats de cette étude, nous pouvons déduire que les dix lettres अ इ ए ख त भ द ध थ ष peuvent presque capturer toutes les propriétés formelles des lettres Devanagari restantes. Dans ces lettres, les lettres अ इ ख भ द ध ष sont les plus critiques car elles définissent des caractéristiques pour la majorité des lettres. Nous pouvons donc suggérer qu'en concevant ces lettres en premier, le processus de conception de la police Devanagari peut être simplifié pour les étudiants et concepteurs de polices puisque les lettres restantes peuvent être dérivées de celles-ci.

Erin McLaughlin a suggéré ces glyphes comme une progression initiale: पाव + किमीनुफू + भरसगदह + मॉं डू (extrêmes de hauteur) + यथघआछड ... continuez le jeu de caractères et a suggéré de se concentrer sur le signe de voyelle "Au" + reph + combinaison anusvara! Le Ma est juste là pour la postérité.

Les glyphes de hauteur extrême vous permettent de déterminer les métriques verticales, et comment mettre à l'échelle les deux systèmes d'écriture pour travailler ensemble. Adobe publie des familles de polices très volumineuses qui couvrent des orthographes très différentes. Celles-ci sont divisées en familles avec des proportions communes partagées; Myriad Pro a le latin, le grec et le cyrillique, mais les designs hébreus et arabes sont dans un pack de familles séparées qui incluent des latins **modifiés**.

Voici Myriad Pro Latin et Myriad Arabic juxtaposés:

Tout le texte est à 48pt  
 dans InDesign CS6  
 Myriad Pro  
 Myriad Arabic  
 Test مرحل اعلا test اب ح ر م  
 xxnnaaftt  
 HxHxXxxx

(Remarquez la décision habile des concepteurs d'Adobe: la hauteur de capitale du latin en Myriad Arabic est la hauteur d'x du Myriad Pro Latin.)

Notez que dans le jeu de caractères Lohit, les glyphes les plus bas sont des formes, destinées à aller en dessous des caractères qui descendent très loin en dessous de la ligne de base:

(Vattu est la forme sous-base de reph. Voir la [terminologie Microsoft](#) pour plus de détails)

Idéalement, ceux-ci devraient s'empiler au-dessous de votre empilage de conjonction la plus basse, comme dans l'exemple sur la gauche (Lohit, qui n'est pas tout à fait ajusté verticalement, est sur la droite):

## Approche d'espacement

La conception de polices latines implique généralement une série de chaînes d'espacement comme ceci:

HHxHOHOxOO  
 nnXnonoXoo

Où le X représente la lettre sur laquelle vous vous concentrez à espacer, le concept étant d'examiner cette lettre près d'un caractère au côté assez plat et d'un caractère rond.

Pa et Va ou Da sont les équivalents Devanagari:

पपXपवपवXवव

पपXपदपदXदद

Lorsque vous débutez un projet, commencez par remplir une page entièrement avec Pa afin d'obtenir le bon équilibre entre l'épaisseur du trait, la taille du contrepoinçon et l'espacement.

पपपपपपपपपपपपपपपपपप

Une fois que le Pa a la bonne "couleur", vous pouvez commencer à ajouter ces autres caractères de base communs:

पपववपपपपववपववपवव (va, aléatoire)

पपपापपपापापा (Aa maatra, aléatoire)

पपपदपपपपदपदपद (da, aléatoire)

Ensuite, vous pouvez commencer à utiliser les chaînes d'espacement ci-dessus, afin d'ajouter des glyphes supplémentaires:

पपरपदपदरदद

पपकपदपदकदद

पपलपदपदलदद

पपपीपदपदपीदद

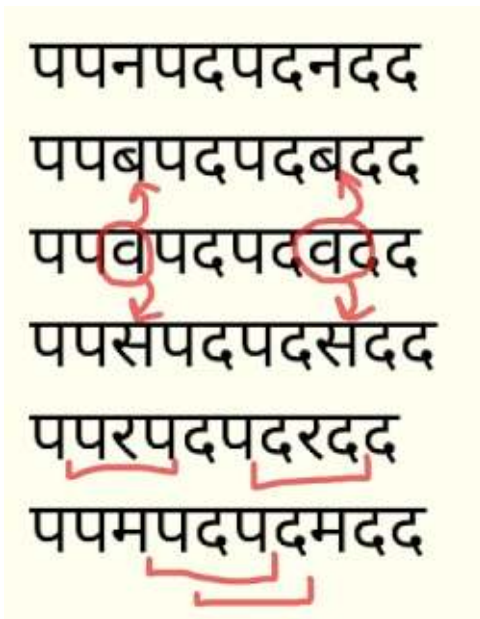
et ainsi de suite!

Vous devrez les regarder dans une longue liste comme ça pour les comparer d'un glyphe à l'autre, en même temps que vous les faites défiler vers le bas — à la fois à l'écran et sur papier. Une vérification verticale est plus efficace qu'une simple longue ligne de texte continu. Voici pourquoi:

Lorsque vous regardez les chaînes d'espacement dans les colonnes verticales, vous pouvez facilement comparer l'espacement avec les lignes qui les précèdent et en dessous du caractère actuel. De la même façon que nous pouvons aisément reconnaître les "rivières" dans un texte pleinement justifié et mal réglé, il sera plus facile de voir des blancs ou des taches noires dans l'espacement si vous comparez avec une chaîne d'espacement qui reste constante.

La chaîne d'espacement ci-dessus vous permet de comparer des formes très disparates, de sorte que l'espacement soit plus uniforme (au lieu de tous les caractères ronds étant trop espacés ou trop serrés)

Les quatre glyphes dans le milieu, Pa/Da/Pa/Da vous permettent de comparer le caractère testé contre deux ensembles de trois, si vous regardez juste Pa/Da/Pa ou Da/Pa/Da.



Après avoir dessiné et espacé une poignée de voyelles et de consonnes, vous serez en mesure de faire un nombre limité de mots avec seulement ces lettres, et commencer à tester votre conception avec du texte réel.

## Plan de travail

Dans tout projet de conception de police, c'est une excellente idée d'esquisser un plan de travail.

Pour quelqu'un de très expérimenté, il est possible de concevoir les graisses initiales Léger et Gras d'une police Devanagari en environ 4-6 mois.

Voici un exemple de calendrier pour une famille interpolée de 9 graisses, verticale et inclinée, d'un design 'sans' assez simple, faite par un designer très expérimenté:

Semaine	Objectif	Glyphes
1	Établir le design de 7-10 glyphes clés	10
2	Affiner, concevoir les glyphes les plus hauts, faire correspondre les hauteurs et les graisses au latin dans Régulier & Gras, tester le rendu à l'écran avec ttfautohint	20
3	Affiner les proportions avec les commentaires des lecteurs natifs	40
4	Obtenir des commentaires de lecteurs natifs, affiner et ajouter d'autres conjonctions	100
5	Obtenir les commentaires de lecteurs natifs, affiner et ajouter des conjonctions	200
6	Obtenir des commentaires de lecteurs natifs, affiner et ajouter des conjonctions	300
7	Obtenir des commentaires de lecteurs natifs, affiner et ajouter d'autres conjonctions	400
8	Obtenir des commentaires de lecteurs natifs, affiner et ajouter d'autres conjonctions	500
9	Obtenir les commentaires de lecteurs natifs, affiner et ajouter des ensembles	600
10	Obtenir des commentaires de lecteurs natifs, affiner et ajouter d'autres ensembles	700
11	Obtenir les commentaires de lecteurs natifs, affiner et ajouter d'autres ensembles	800
12	Obtenir des commentaires de lecteurs natifs, affiner et ajouter d'autres conjonctions	900
13	Dériver le Gras	1 800
14	Raffinements, crénage, tests avec commentaires des lecteurs natifs	1800
15	Extrapolation et nettoyage des graisses minces et noirs, génération et nettoyage des styles inclinés	3600
16	Raffinement des styles interpolés	3600
17	Raffinement général de l'espacement, du crénage et tests dans tous les styles	3600
18	Finalisation	3600

Vous voudrez peut-être travailler avec une police qui n'a pas de sources disponibles, seulement les tables binaires OpenType GPOS/GSUB. Il existe quelques outils qui peuvent les convertir dans la syntaxe Adobe FEA, y compris FontForge, mais la sortie de chaque outil nécessitera un retraitement à la main.

Le FDK d'Adobe contient un outil 'spot', qui peut être utilisé comme ceci:

```
Spot -t GSUB=7 Font.otf> GSUB.fea
```

Le projet noto possède un [dump\\_otl.py](#)

Les applications 'Fontlab Studio' et 'OpenType Master' possèdent également des convertisseurs.

## Ressources utiles

### Introductions

- <http://www.linotype.com/6896/devanagari.html>

### Où chercher de l'inspiration et des idées

Regardez les polices Devanagari sur le site <http://indiantypefoundry.com>, et ceux qui viennent de sortir via Google Fonts, car les formes de lettres peuvent varier.

Un autre bon endroit pour rechercher de l'Hindi sont les sites de journaux "e-paper" hindi pour voir les polices réelles en cours d'utilisation — les publicités ont généralement plus de diversité dans les polices. <http://epaper.jagran.com> est un e-papier indien largement diffusé.

Flickr est également une bonne source d'idées par ses images:

- <https://www.flickr.com/groups/devanagari-script/>
- <https://www.flickr.com/groups/37703106@N00/>
- <https://www.flickr.com/groups/indicscripts/>
- <https://www.flickr.com/photos/pauldhunt/sets/72157603715699186>

## Sources historiques

Mettez la main sur des copies de *Introduction to the Devanagari Script* par H. M. Lambert, Oxford University Press 1953 et *Typography of Devanagari* (trois volumes) par B. S. Naik, Direction des Langues, Bombay 1971.

Au-delà de cela, il existe au moins deux sources générales de fontes du 19e siècle venant d'Europe qui vaut la peine d'être regardées: les polices de caractères de la Grande-Bretagne et celles de l'Allemagne (principalement de Leipzig). Ces fontes ont plus été utilisées pour la mise en texte sanskrit que pour les textes hindi.

Essayez également de trouver des échantillons de caractères de texte de 19e et 20e siècles provenant de fonderies indiennes. Ils sont nettement moins européanisés, comme vous pouvez l'espérer. Il y a des choses stupides qui se voient dans les fontes académiques européennes sanskritiques du 19ème siècle qui ne semble pas du tout apparaître dans la typographie indienne du 20ème siècle. Ces sources indiennes sont probablement plus difficiles à trouver dans les bibliothèques occidentales, mais peut-être qu'Erin McLaughlin a plus d'une piste. Linotype Devanagari de Matthew Carter, 1970, est basé sur des fontes de la fonderie Nirnaya Sagar, par exemple. Les échantillons de leurs polices et les polices de la fonderie de Bombay devraient être accessibles dans certaines universités et/ou bibliothèques nationales occidentales. Je recommanderais également de regarder le Devanagari Monotype et le Devanagari Linotype (la version des années 1970 et la mise à jour des années 80/90s, pas l'original 1935, qui portait seulement le même nom).

Il n'y a pas de police Devanagari dans Typefounders aux Pays-Bas (Charles Enschede, Harry Carter, 1978). Quoi que vous fassiez, ne regardez pas les polices Bodoni de son manuel de 1818.

Une certaine police Devanagari faite en Allemagne par H. Berthold AG peut être vue dans *Alphabete und Schriftzeichen des Morgen- und des Abendlandes*, du Reichsdruckerei, Berlin 1924, p. 45-47.

## Articles

Sarang Kulkarni a écrit "[Issues with Devanagari Display Type \(PDF\)](#)"

Yashodeep Gholap a écrit [Designing a Devanāgarī text font for newspaper use \(PDF\)](#)

La dissertation de Vaibhav Singh dans MA Typeface Design, [Devanagari in multi-script typography](#)

## Lohit2 Devanagari

Le Lohit2 Devanagari peut être utilisé comme base pour les nouvelles polices OFL en utilisant sa liste de glyphes et son code de disposition OpenType. Il est disponible en tant que [source originale de FontForge](#) ou en [téléchargement zip](#)

## Disposition OpenType

[Page de développement de la police Microsoft Devanagari OpenType](#)

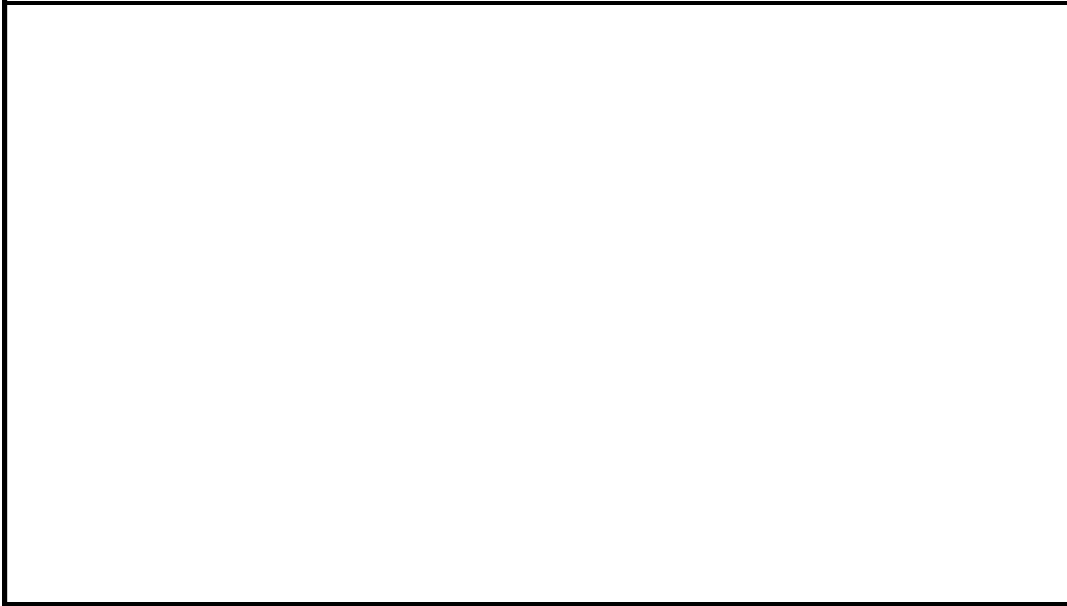
## Anatomie du Devanagari

[Grammaire d'écriture Devanagari du TDIL \(Technology Development for Indian Language\) \(PDF\)](#)

[2 pages du manuel de calligraphie Devanagari d'Aksharaya](#), which can be used as a reference for pen angle and letter proportions.

Le Professeur Girish Davli de l'IIT Bombay IDC (comparable au MIT Media Lab des Etats-Unis) a publié cet [article sur l'anatomie du Devanagari \(PDF\)](#)

Si vous êtes nouveau dans l'écriture Devanagari, il est important de prendre conscience de la pression de la plume calligraphique traditionnelle, qui est différente de celle du latin. Voici une démonstration rapide de l'angle d'attaque et de la façon traditionnelle d'appliquer la graisse aux courbes. Votre conception sera plus réussie, et apparaîtra moins «latinisé», si vous dessinez vos courbes selon ces principes de graisse, plutôt que de couper-et-coller des parties de lettres de l'alphabet latin.



La page [Devanagari Unicode](#) montre les lettres de base, mais pas les conjonctions.

## Outils de test

Le support d'Adobe InDesign pour les polices OpenType pour les écritures non européennes n'est devenu fiable qu'avec Creative Cloud et l'[option de paragraphe World Ready Composer doit être activée](#). Même la dernière version peut être améliorée. Les implémentations libre harfbuzz et Microsoft OpenType sont les plus complètes. Vous devez donc tester vos polices dans les dernières versions de Chrome, Firefox et Microsoft Word, sur Windows et Mac OS X, pour vous assurer que les erreurs de mise en forme sont dues aux polices et non au moteur sous-jacent.

La [page d'essai de Devanagari de Pablo Impallari](#) (avec [sources sur github!](#)) fournit des schémas de tests prêts à l'emploi, et vous pouvez glisser-déposer votre OTF ou TTF directement dans la page pour le charger.

Les [outils de police de Pooja Saxena](#) (avec [sources sur github!](#)) génère des textes de test avec combinaisons de lettres.

Le [Adhesion Text Devanagari](#) est une version spéciale de l'outil construit par Miguel Sousa pour faire du texte fictif composé uniquement des mots possibles pour écrire avec les glyphes que vous avez déjà dessinés. Insérez les glyphes (अआईईउऊ etc) que vous avez dessinés, et il apparaîtra quelques exemples de mots à utiliser dans vos tests.

Le Huerta Tipografica [Devanaguide](#) est un outil open source pour voir et comparer les différentes polices devanagari. Il vous permet également de taper un texte et de le prévisualiser dans toutes les polices en même temps. Le Devanaguide comprend également une liste de mots [Devanagari](#) qui est utile pour la conception de texte d'essai.

## Autres liens

[Books & References for Devanagari typography Projects](#) est une liste de ressources qui répondra probablement à la plupart des besoins de recherche de beaucoup de projets. Compilé par le Professeur Girish Davli de IIT Bombay IDC.

## Forum de discussions

Typophile

- [Adobe Devanagari](#) (en anglais)



### Répertoire des discussion Google Fonts (en anglais)

- [A note on conventions for the placement of shoulder line, descenders, ascenders and so forth in relation to the latin?](#)
- [Starting from Lohit2 and pdf](#)
- [Adobe Glyph Lists](#)
- [rVocalic and rrVocalic](#)
- [Devanagari Character Priorities](#)

# Importation de glyphes d'autres programmes

Il est possible de dessiner des glyphes dans une application d'illustration (tel qu'Inkscape, Adobe Illustrator, etc.) et de les importer comme EPS ou SVG.

## SVG codé à la main

### Comment le préparer

- Le fichier SVG a pas besoin de `viewBox="0 0 1000 1000"`
- La largeur n'a pas d'importance, tant qu'elle est plus large que votre glyphe. Mais une hauteur de 1000 est important pour faciliter l'importation.
- `y=0` sera la ligne de l'ascendante et `y=1000` sera la ligne de la descendante.
- (Il peut y avoir quelques glyphes qui vont au-delà de ces lignes. Peut-être que FontForge fera la bonne chose, mais nous ne l'avons pas testé.)
- Par défaut, FontForge configurera votre ligne de base à `y=800`. Dans le système de coordonnées FontForge, la ligne de base se trouve à '0' sur l'ordonnée.
- Pour définir la ligne de base où vous la voulez dans FontForge, prenez la coordonnée y pour votre ligne de base en SVG. Ce sera le point vertical de FontForge pour la ligne d'ascendante dans son système de coordonnées.  $(1000 - y)$  pour la descendante. Allez dans `Élément, Infos fonte` et dans le menu Général, placez la valeur de l'ascendante dans l'entrée "Ascent" et celle de la descendante dans l'entrée "Descent". Les deux seront positifs. Le cadratin (la taille Em) doit rester à 1000 (comme c'est la hauteur en unités SVG).
- Lors du dessin du glyphe, il est préférable d'utiliser des coordonnées relatives. Commencez le glyphe avec `<path d="M Xvalue,Yvalue`. Si vous pouvez dessiner le glyphe à partir d'un point tout à gauche, alors le Xvalue sera l'approche gauche par défaut (LeftBearing) que FontForge utilise. Vous pouvez ajuster cela facilement après l'importation du glyphe et devriez avoir besoin de le faire de toute façon après avoir testé la police. Pour le Yvalue, quand vous pouvez commencer à dessiner à partir de la ligne de base, utilisez cette valeur de ligne de base pour le Yvalue.
- Toujours terminer l'attribut du chemin d avec un z. Le SVG importera sans cela, mais le glyphe ne s'affichera pas directement dans la fenêtre principale jusqu'à ce que vous redémarriez fontforge si vous oubliez de mettre un z après le dernier point du chemin.
- Lorsque vous dessinez des trous (comme pour la lettre P), ne commencez pas un nouveau nœud de chemin. Il suffit d'utiliser z à la fin du premier chemin et commencez de nouveau avec `mNewX,NewY` pour commencer à dessiner le trou. Utilisez l'attribut `fill-rule="evenodd"` pour le chemin et cela fonctionnera correctement.

### Flux de travail

Utilisez un navigateur Web pour afficher le SVG sur lequel vous travaillez. Vous pouvez utiliser un fichier nommé "template.svg" qui est 1200 par 1200, mais s'affiche à 800 par 800 afin qu'il ne défile dans la fenêtre du navigateur.

Dans ce modèle, tracez les lignes directrices à `y=100, y=1100, y=(100 + {ligne de base, hauteur de capitale, etc.}, x=100, x=1100`

Puis importez le glyphe SVG sur lequel vous travaillez dans ce document avec `<image xlink:href="LC_p.svg" x="100" y="100" width="1000" height="1000" />`

Vous pouvez maintenant coder manuellement votre lettre dans une fenêtre et actualiser le navigateur dans l'autre pour le voir apparaître au-dessus des lignes directrices.

## Listes de glyphes personnalisées

1. Créez un fichier `namelist.txt`, en utilisant par exemple une feuille de calcul pour répertorier les points de code et les noms de glyphes unicode. Par exemple:

```
0xEC00 octDotDhe
0xEC01 octDotDheDb1
0xEC02 octDotDheTrp1
0xEC03 octDotDheQdrp1
0xEC04 octDotLik
0xEC05 octDotLikDb1
0xEC06 octDotLikTrp1
0xEC07 minirLik
0xEC08 minirDhe
0xEC09 minirBawah
0xEC0A soroganDhe
0x-001 soroganLik
```

Pour les glyphes sans point Unicode, utilisez un point de code de -1, comme dans la dernière ligne de l'exemple ci-dessus.

Puis ouvrez FontForge et allez dans `Codage > Ouvrir une liste de noms` puis utilisez `Renommer les glyphes` (Puisque `Ouvrir une liste de noms` ajoute seulement la liste de noms personnalisée à l'ensemble des options disponibles dans les commandes de renommage subséquentes).

# Ajout de glyphes à une police arabe

## Introduction

Dans certains cas, il peut manquer un glyphe dans une police qui est essentiel à son utilisation dans votre application. Les polices arabes présentent des problèmes particuliers ici, car la forme du glyphe dépend non seulement de sa position dans le mot, mais aussi des attributs de la lettre elle-même. Ainsi, à l'aide de la séquence sans signification *babab*, la lettre *beh* a trois formes différentes selon qu'elle vient initialement, médialement ou finalement. Cependant (à l'aide de la séquence sans signification *dadad*), la lettre *dal* n'a qu'une seule forme, peu importe où elle se trouve dans le mot.

Les polices sous licences ouvertes (par exemple [GPL](#) ou [OFL](#) permettent à l'utilisateur de faire des modifications. Si vous adaptez une police qui était à l'origine sous licence libre et que vous la distribuez, vous devez conserver les avis de droits d'auteur et les informations sur les licences de l'auteur original, bien que vous puissiez ajouter une note à la fin de l'avis de copyright couvrant votre contribution.

ب beh بَب  
د dal دَد

Ce chapitre décrit pas à pas l'ajout d'un glyphe à une police arabe. La police que nous allons utiliser est [Graph](#), et le glyphe que nous ajouterons est *peh* (U+067E), qui ne se trouve pas en arabe lui-même, mais désigne *p* dans certaines langues pour lesquelles l'écriture arabe est utilisée (pour une liste complète des glyphes disponibles pour l'écriture arabe, voir les [diagrammes Unicode](#)).

پ peh

## Faire une copie de travail de la police

Téléchargez la police de la page Web et décompressez-la. Lancez FontForge et chargez la police. Enregistrez-la en tant que fichier *sfd*, modifiez le nom suggéré pour lire **GraphNew.sfd** avant d'enregistrer.

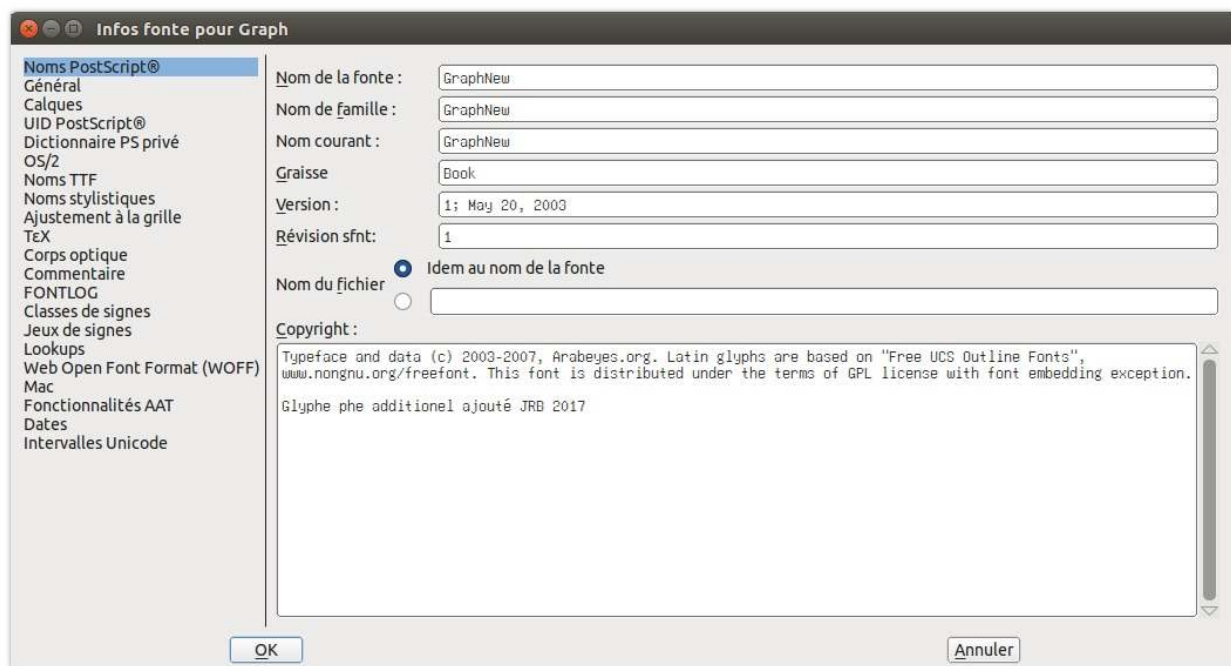
## Renommer la police

### Pourquoi devrais-je renommer la police?

Si vous ne renommez pas la police, votre police adaptée ne sera pas installée séparément de l'original – vous devrez d'abord désinstaller la police d'origine. Il est également judicieux de renommer la police si vous allez distribuer vos adaptations – si l'auteur original de la police a réservé le nom de la police sous le mécanisme réservé de nom de police (RFN), ce nom d'origine peut uniquement être utilisé avec la version d'origine de l'auteur de la police.

### Modifier les données de nom

Sélectionnez **Element** → **Infos fonte**, et dans le volet *Noms PostScript*, changez le *nom de la fonte*, le *nom de famille* et le *nom courant* à **GraphNew**.



Si vous le souhaitez, vous pouvez placer un message «Ajout de glyphes supplémentaires par» après le texte déjà entré pour le *Copyright*.

Dans le volet *Noms TTF*, les noms de *famille* et *complet* proviennent des entrées *Noms PostScript* et doivent déjà afficher *GraphNew* (vous ne pouvez pas les modifier directement). Modifiez les entrées pour *Famille préférée* et *Compatibilité totale* à **GraphNew**. Ces changements de nom vous permettront maintenant d'installer cette police à côté de l'original si vous le souhaitez.

Si vous le souhaitez, vous pouvez placer un message 'Ajout de glyphes supplémentaires' après le texte déjà entré pour *Designer*.

Cliquez sur **OK** pour enregistrer ces modifications. Vous recevrez un message sur la génération d'un nouveau UniqueID (XUID) pour la police – Cliquez sur **Changer**.

## Ajouter le glyphe pour la forme isolée *peh*

Allez dans la section arabe du tableau de fonte: sélectionnez **Vue** → **Aller au glyphe**, cliquez sur la liste déroulante et sélectionnez **Arabe**, puis cliquez sur **OK**.

Cliquer sur une cellule dans le graphique de la police affichera son numéro Unicode et son nom en bleu en haut de l'application. Allez à la position 1662, qui apparaîtra en bleu comme *1662 (0x67e) U+067E "uni067E" ARABIC LETTER PEH*. La cellule sous le glyphe de référence contient un X gris, montrant que la police n'inclut pas ce glyphe.



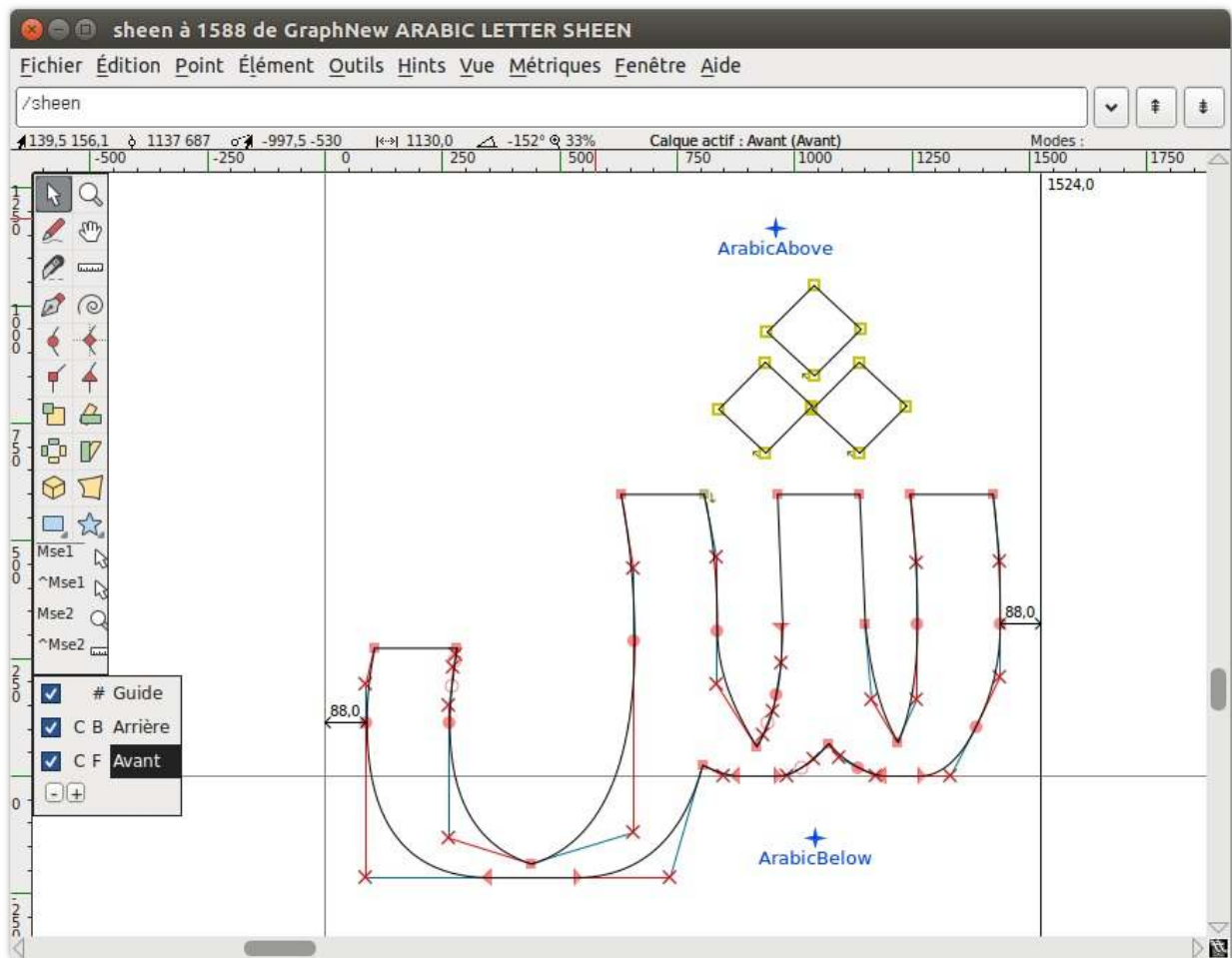
Nous ferons *peh* en copiant *beh* (U+0628) et en changeant son point unique pour trois points.

Cliquez sur la cellule *beh* (position 1576), puis cliquez avec le bouton droit de la souris et sélectionnez **Copier**. Puis cliquez avec le bouton droit de la souris sur la cellule *peh* et sélectionnez **Coller**. Maintenant que *beh* est copié dans la cellule *peh*, la prochaine chose est de changer le point.



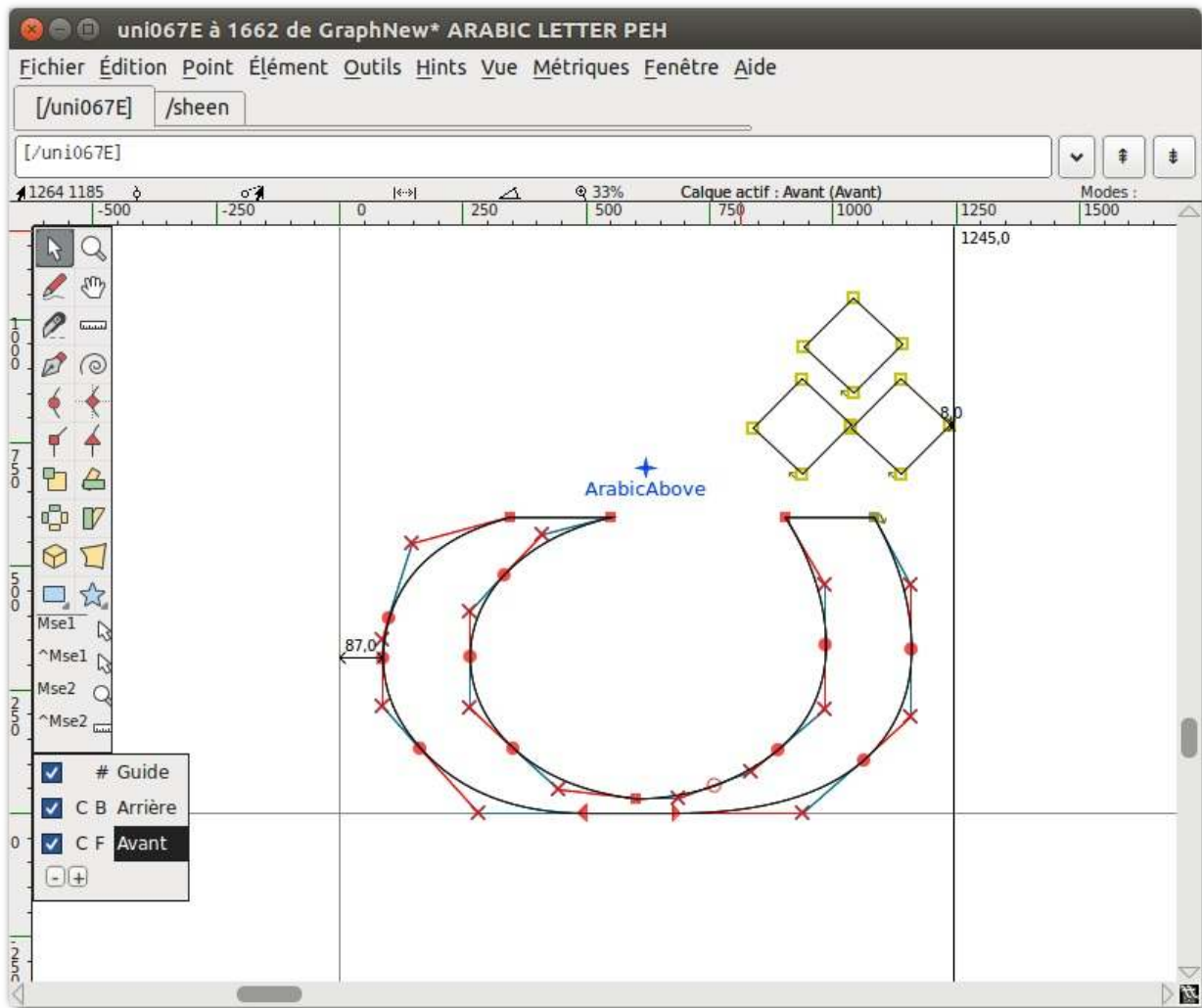
Trouver un glyphe avec trois points – *sheen* (position 1588, U+0634) suffira. Double-cliquez sur la cellule – cela ouvrira la fenêtre de conception de glyphe. Appuyez sur **V** pour vous assurer que l'outil pointeur (tête de flèche) dans la boîte à outils est sélectionné, puis appuyez sur **Z** et agrandissez la fenêtre pour avoir une bonne vue du glyphe.

Cliquez et faites glisser afin que les nœuds des trois points au-dessus de *sheen* changent de couleur de rose à beige. Si vous avez accidentellement inclus ou omis un nœud, désélectionnez-le ou sélectionnez-le en appuyant sur **Maj** et en cliquant dessus. Appuyez sur **Alt+C** pour copier.

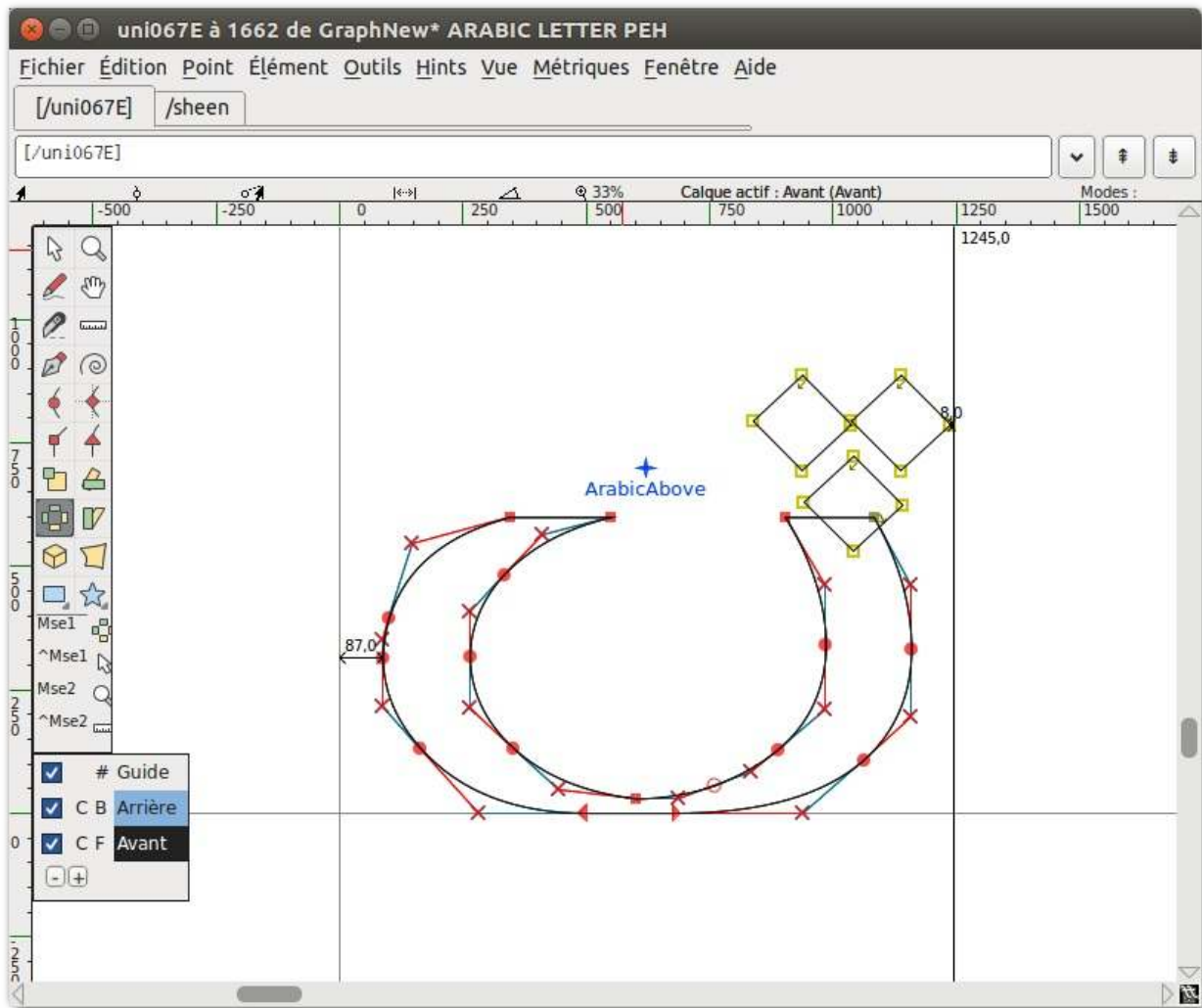


Revenez à la fenêtre de fonte et double-cliquez sur la cellule *peh* – cela va charger *peh* dans un autre onglet dans la fenêtre de conception de glyphe, à côté de l'onglet *sheen*.

Cliquez et faites glisser pour sélectionner le point en dessous de *peh*, puis appuyez sur **Supprimer**. Appuyez sur **Alt+V** pour coller les trois points, qui apparaîtront probablement au-dessus du corps de *peh*. Laissez les nœuds de points en surbrillance afin que vous puissiez les inverser et les déplacer plus facilement.

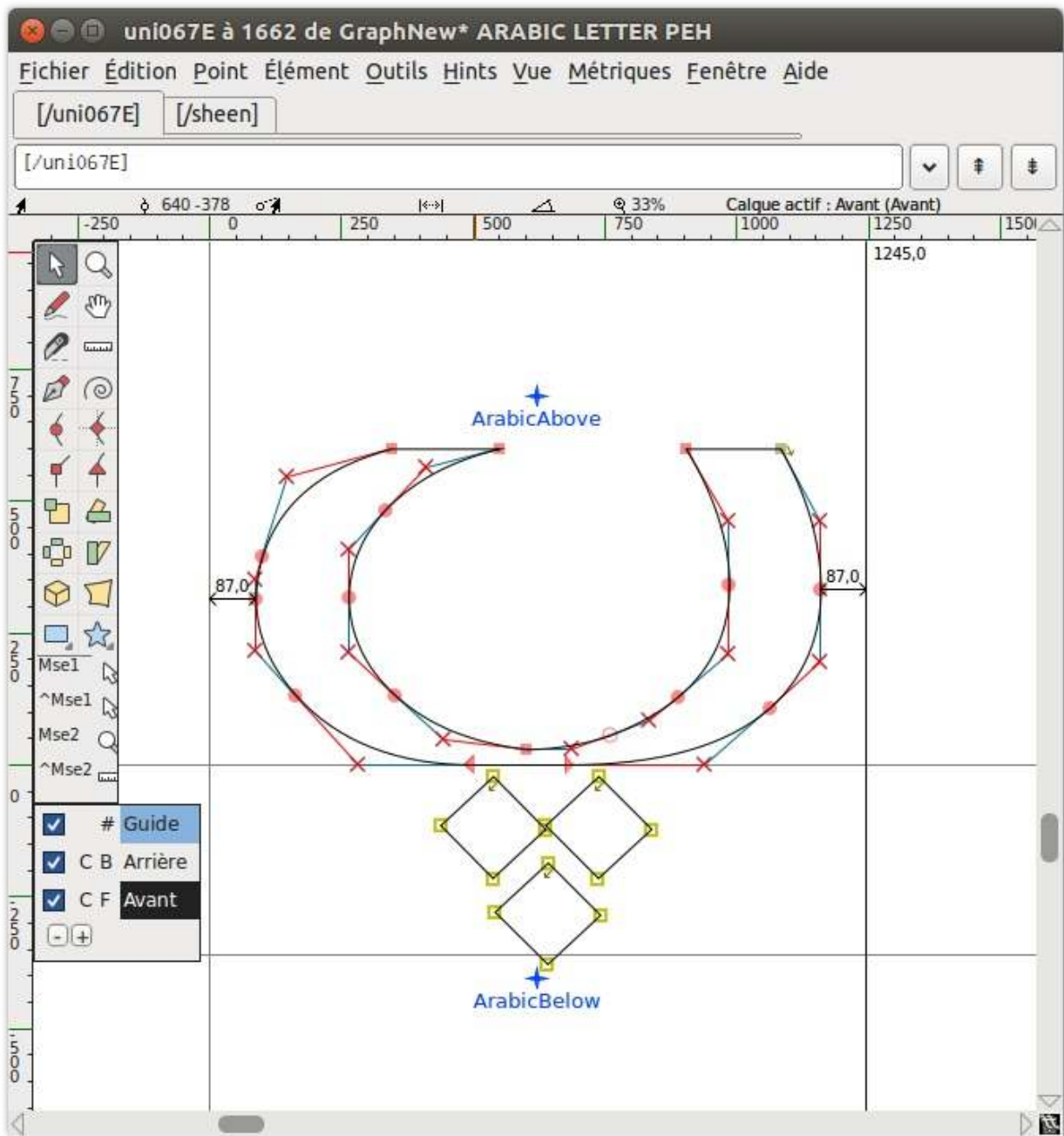


Inverser les points: sélectionnez l'outil symétrie (quatre rectangles jaune et vert) de la boîte à outils. (Alternativement, cliquez avec le bouton droit de la souris dans le milieu des points et sélectionnez **Appliquer une symétrie à la sélection** dans la fenêtre contextuelle.) Cliquez sur l'un des nœuds de points et faites glisser la souris légèrement vers la gauche ou vers la droite.



Déplacer les points inversés: appuyez sur **V** pour sélectionner à nouveau l'outil de pointeur, cliquez sur un des nœuds de points et faites-le glisser sous le corps du glyphe. Positionnez-les au centre, au-dessus de la marque *ArabicBelow*.





Fermez le panneau de conception du glyphe. Il devrait maintenant y avoir un nouveau glyphe pour *peh* dans le tableau. Enregistrez la police adaptée (**Fichier** → **Enregistrer**).



**Ajoutez les glyphes pour les formes connectées de *peh***

Cependant, ce n'est que la forme isolée (autonome) du glyphe. Si vous essayez d'utiliser votre police adaptée, vous constaterez que les formes initiales, médianes et finales ne sont pas disponibles. Celles-ci doivent être créées séparément. "Ces formes sont construites comme des glyphes non codés (glyphes dont le codage est -1 dans les conventions FontForge). Elles n'ont pas de cases prédéfinies." (Khaled Hosny)

Sélectionnez **Codage** → **Ajouter des cases de codage** et entrez le nombre de glyphes souhaités – dans ce cas **3**. FontForge ajoutera le même nombre d'emplacements à la fin de la police, et la vue se déplacera à cet endroit dans le tableau de la police. Les trois dernières cellules (positions 65537, 65538, 65539) ont un point d'interrogation comme glyphe de référence, et c'est dans ces cellules que vous allez ajouter les glyphes non codés en répétant le processus ci-dessus.

←	↑	→	↓	■	◇	FF EF
		?	?	?	?	
		□				

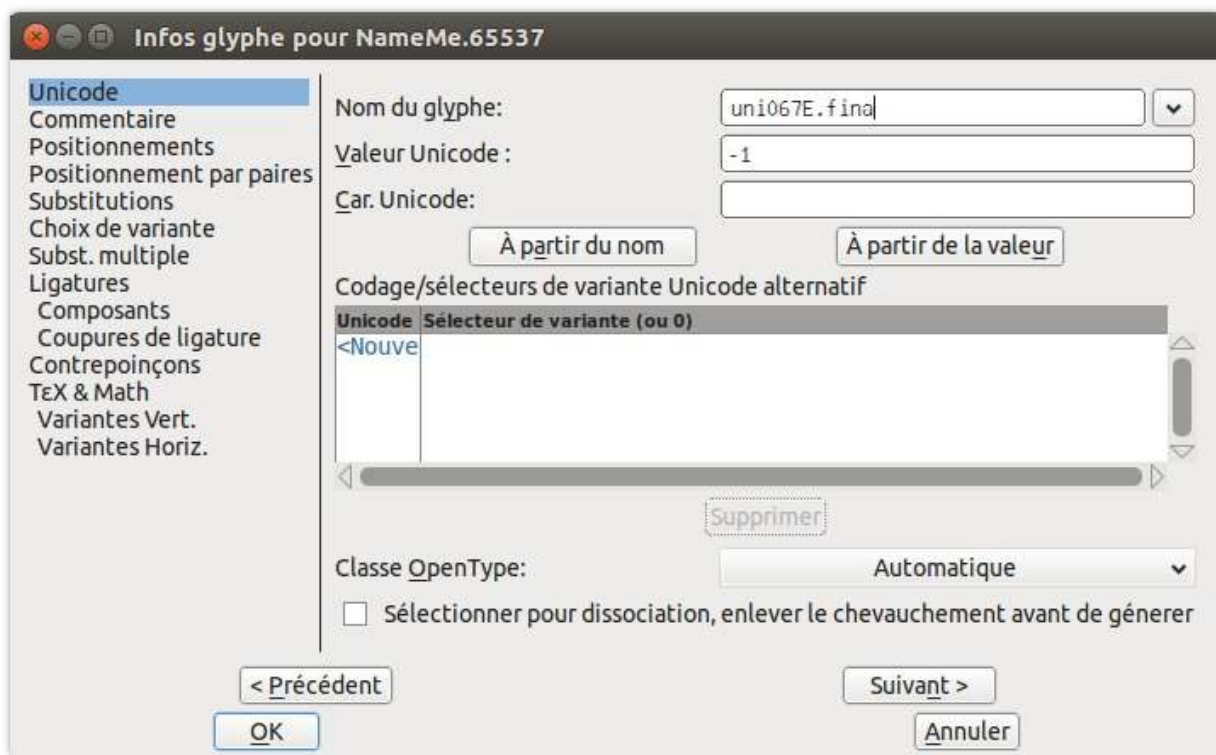
Notez que si par erreur vous commencez à taper quand le focus est toujours sur le tableau de la police, vous serez déplacé à la section européenne en haut. Pour revenir au bas, sélectionnez *Vue* → *Aller au glyphe*, cliquez sur la liste déroulante et sélectionnez *Remplaçant pour NON unicode*, puis cliquez sur *OK*.

## Créer la forme finale

Remontez le tableau de police un peu jusqu'à ce que vous arriviez à un ensemble de glyphes arabes à la position 65152 (U+FE80). À U+FE90 (position 65168), vous verrez un glyphe *behfinal* – cliquez dessus et appuyez sur **Ctrl+C** pour le copier. Faites défiler jusqu'à la troisième dernière cellule du tableau (position 65537), cliquez dessus et appuyez sur **Ctrl+V** pour y coller le glyphe *behfinal*.

ل	و	و	ي	ي	ة
ل	و	و	ي	ي	ة
و	و	و	و	و	و
و	و	و	و	و	و

Faites un clic droit sur la cellule et sélectionnez **Infos glyphe**. La convention de dénomination consiste à utiliser le nombre du glyphe isolé + un suffixe pour la forme. Changez le *Nom du glyphe* en **uni067E.fina**, puis cliquez sur **OK**. Le point d'interrogation dans la cellule de référence devient *peh*.



Obtenez les trois points: double-cliquez sur *sheen* (U+FE91) pour le charger dans la fenêtre de conception de glyphe, sélectionnez les trois points et appuyez sur **Ctrl+C**.

Double-cliquez sur le nouveau *pehfinal* pour le charger dans la fenêtre de conception du glyphe, cliquez et faites glisser pour mettre en surbrillance les nœuds du point et appuyez sur **Supprimer**.

Ctrl+V pour insérer les trois points de *sheen*, retournez-les et placez-les en position sous le corps du glyphe. Appuyez sur **Ctrl+S** pour enregistrer le tableau révisé.

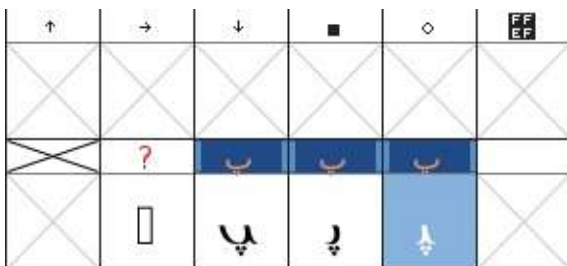
## Créer les formes initiales et médiales

Copier la forme initiale U+FE91 (position 65169) à l'avant-dernière cellule (position 65538), supprimer le point unique et coller-y les trois points.

Cliquez avec le bouton droit sur la cellule, sélectionnez **Infos glyphe**, changez le *Nom du glyphe* à **uni067E.init**, puis cliquez sur **OK**.

Copiez la forme médiane U+FE92 (position 65170) dans la dernière cellule (position 65539), supprimez le point unique et collez-y les trois points.

Cliquez avec le bouton droit de la souris sur la cellule, sélectionnez **Infos glyphes**, changez le *Nom du glyphe* à **uni067E.medi**, puis cliquez sur **OK**.



Sélectionnez **Fichier** → **Enregistrer** pour enregistrer le tableau de polices révisé.

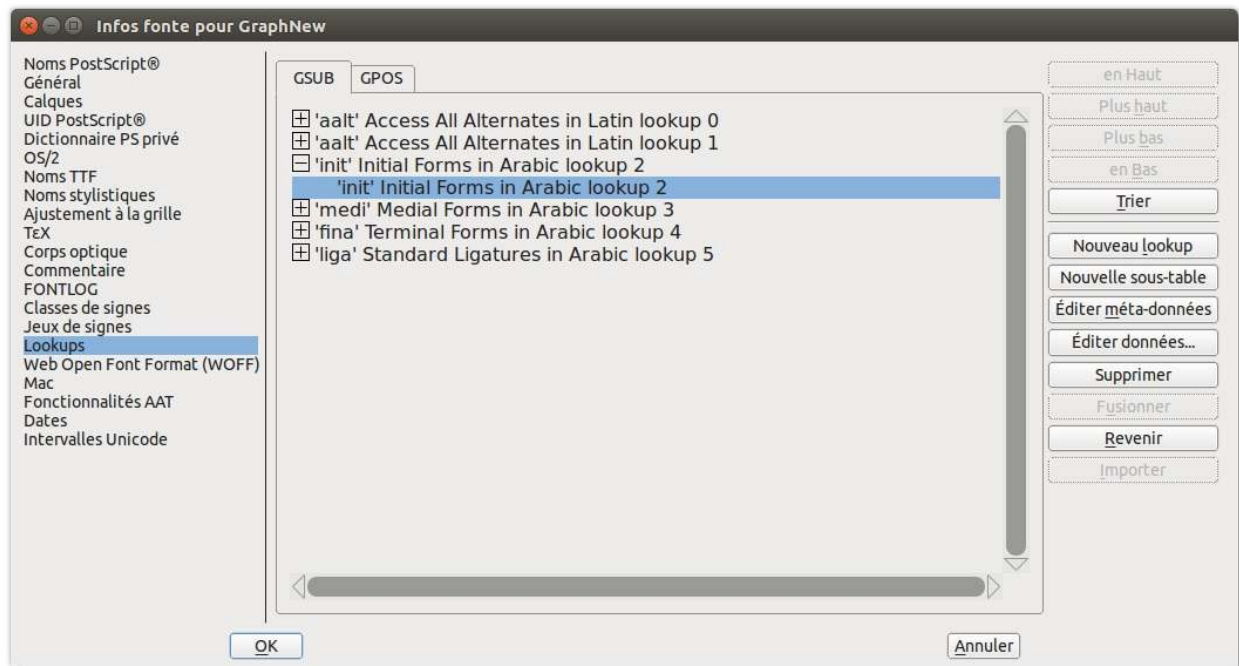
## Ajouter les lookups

La forme isolée doit être mappée (liée) à ses formes initiale, médiane et finale.

Sélectionnez **Élément** → **Infos fonte** → **Lookups**.

Cliquez sur le **+** à côté de l'entrée *'init' Initial Forms in Arabic lookup 2*. Cela ouvrira un sous-menu du même nom. Cliquez sur ce sous-menu.

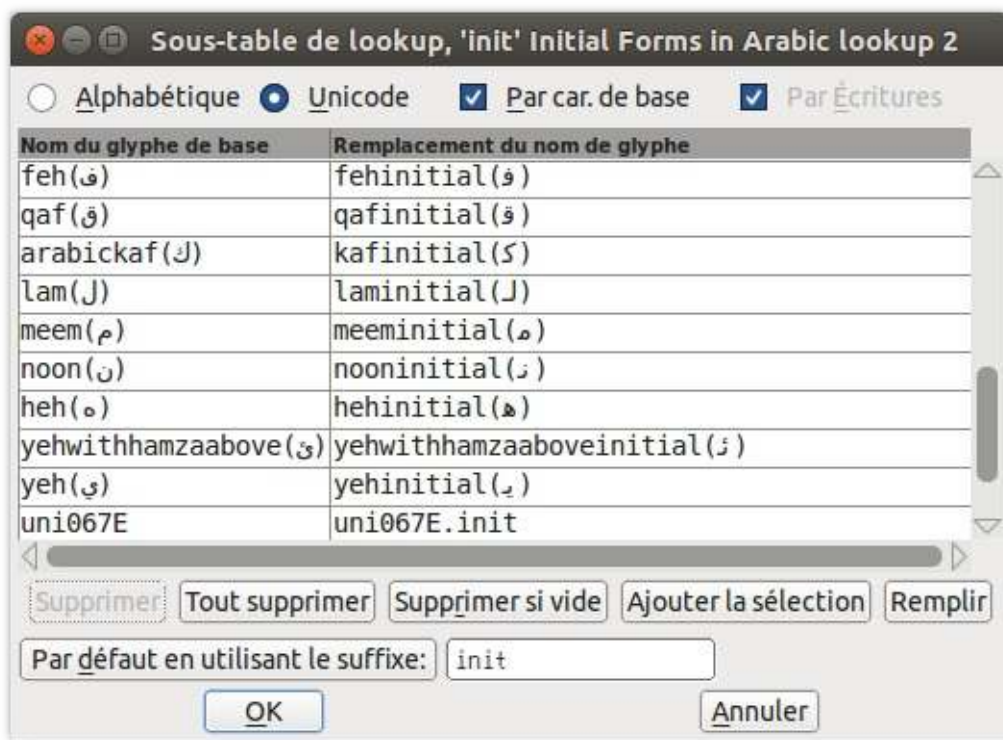
Le bouton *Éditer données* à droite deviendra disponible maintenant – cliquez dessus.



Dans la *Sous-table de lookup* affichée, vérifiez que le bouton *Unicode* est coché. Faites défiler la liste de caractères vers le bas jusqu'à ce que vous arriviez à la fin.

Dans l'encadré à côté de *Par défaut en utilisant le suffixe*, saisissez le suffixe correspondant (dans ce cas, **init**), puis cliquez sur **Par défaut en utilisant le suffixe**.

Un nouveau mappage sera ajouté à la liste des caractères, de uni067E (la forme isolée de *peh*) à uni067E.init (la forme initiale). Cliquez sur **OK**.



Faites la même chose pour les sous-menus sous les entrées *medi'* *Medial Forms in Arabic lookup 2* et *fina'* *Terminal Forms in Arabic lookup 2*, en choisissant *medi* et *fina* comme suffixe pertinent.

Cliquez à nouveau sur **OK** pour fermer le volet et enregistrez le tableau de police (**Ctrl+S**).

Notez que *Par défaut en utilisant le suffixe* semble fonctionner uniquement sur les glyphes du bloc Unicode 06 (*Arabe*) – les glyphes dans Unicode 07 (*Arabe supplémentaire*), par ex. *ain* avec deux points, peut être ajouté manuellement en cliquant sur la ligne marquée *Nouvelle* et en tapant les noms.

## Générer la police adaptée

Sélectionnez **Fichier** → **Générer fonte(s)**.

Dans la liste déroulante affichant *PS Type 1 (Binaire)*, sélectionnez **TrueType** et vérifiez que le nom de fichier lit *GraphNew.ttf*.

Accédez à l'endroit où vous souhaitez enregistrer la police, puis cliquez sur **Générer**. Cliquez sur **Oui** et **Générer** sur les deux messages d'information qui s'affichent.

Vous pouvez ensuite utiliser votre procédure normale d'installation de polices pour installer la police adaptée. Le nouveau glyphe *peh* peut alors être utilisé aux côtés des glyphes existants dans les mêmes chaînes sans signification qu'au début de ce chapitre:

بَبَب

دَدَد

يَیَی

Notez que si vous utilisez une police dans LibreOffice et apportez des modifications à cette police, vous devrez redémarrer LibreOffice pour qu'elle affiche les modifications. – sinon elle utilisera la version précédente de la police, et non celle avec les nouvelles modifications.</P>

Merci à [Khaled Hosny](#) pour ses conseils sur l'utilisation de FontForge pour modifier les glyphes arabes.

## Lecture complémentaire (en anglais)

- La page <http://lists.nongnu.org/archive/html/freetype-devel/2015-08/msg00016.html> a un conseil sur la façon de dessiner les parties se chevauchant des glyphes arabes.

# Lectures complémentaires

## Vidéos

- [Optical Compensation](#) par Thomas Phinney and FontLab
- [Type Review Videos](#)
- [Lato: The Making Of](#)

## Sites Internet

### Autres listes de ressources

- [Thomas Phinney's Type Design Resources](#)

### Mathématiques de la courbe

- <http://pomax.github.io/bezierinfo/> Excellent guide pour le calcul des courbes de Bézier
- <http://raph.levien.com/phd> Doctorat de Raph Levien sur les courbes Spiro

### Affaires

- [ILT: Why Did I Start A Foundry](#)
- [ILT Leonidas on Typeface Design practice](#)

### Typographie

- [Typekit: Practice](#)

### Latin

Les guides de la fonderie MyFonts sont fantastiques : [foundry.myfonts.com/guides](http://foundry.myfonts.com/guides)

- [Frere Jones: Typeface Mechanics 1](#)
- [Frere Jones: Typeface Mechanics 2](#)
- [Letter-by-letter tutorials](#)
- [Adobe: Designing Multiple Master Typefaces](#)
- [University of Reading Type Design Intensive Review](#)
- [On Tilde](#)
- [ILT: On Diacritics](#)
- [Twardoch: Polish Diacritics](#)
- [Logotype: Quora logo](#)
- [Victor Gaultney: Latin Diacritics](#)

Exemples de projets de *I Love Typography*:

- [Novel](#)
- [Questa](#)
- [Sindelar](#)
- [GT Sectra](#)
- [FF Tundra](#)

- [Acorde](#)
- [Ode](#)
- [Biome](#)
- [Vesper](#) (voir aussi [github.com/motaitalic/vesper-libre](https://github.com/motaitalic/vesper-libre))

## Grec

- [Greek Typefaces Worth Studying](#)

## Cyrillique

- [Learn Cyrillic](#)
- [Serbian Cyrillic](#)
- [Bulgarian Cyrillic](#)
- <http://www.lettersoup.de/what-shall-be-done-for-bulgarian-cyrillic-loclbgr/>
- [Paratype Language Help — Codages cyrilliques](#)
- [Cyrillic Typography on Facebook](#)
- <http://luc.devroye.org/cyrillic.html>
- [La liste de diffusion ATypI Cyrillic SIG](#)

## Arabe

- [Khaled Hosney's FontForge Guide](#)
- [Example Project: Arabic Zapfino](#)
- [Parachut: Interview with Hasan Abu Afash](#)
- [Arabic Typefaces Worth Studying](#)
- [Arabic Type Anatomy](#)

## Hébreu

- [Adi Stern's MATD Dissertation](#)

## Devanagari

- [Microsoft Specifications for Devanagari Fonts](#)
- [South Asia Language Resource Center The University of Chicago](#)
- [Professor Mahendra Patel](#)
- [Dhruvi Tolia Graduation Report](#)
- [Sanchit Sawaria Project 3 Document](#)
- [Yashodeep Gholap — Designing a Devanagari text font for newspaper use](#)
- [Mustafa Saifee — Devanagari Font for Optical Character Recognition](#)
- [Vaibhav Singh — Devanagari in multi-script typography](#)
- [Claviers virtuels: lexilogos et branah](#)

## Livres

- [University of Reading MA Typeface Design Summer Reading List](#)
- <http://www.type-library.com> a une liste étendue de livres

Titre: Detail in Typography (Paperback)

Auteur: Jost Hochuli (Auteur)

Titre: ISBN: 9780907259343



Titre: Adrian Frutiger — Typefaces: The Complete Works

Auteurs: Heidrun Osterer, Philipp Stamm, Swiss Foundation Type and Typography

Titre: ISBN: 9783764385811

---

Titre: Printing Types: Their History, Forms, and Use: A Study in Survivals (with ILLUSTRATIONS)

Auteur: D. B Updike, Daniel Berkeley (Auteurs)

---

Titre: Creative Characters Format: Flexibound

Auteur: Jan Middendorp (Editor)

Titre: ISBN: 9789063692247

---

Titre: Now Read This: The Microsoft Cleartype collection

Auteur: John D. Berry (Auteur)

---

Titre: The Stroke: Theory of Writing (Paperback)

Auteur: Gerrit Noordzij (Auteur)

Titre: ISBN: 9780907259305

---

Titre: Shaping Text: Type, Typography and the Reader

Auteur: Jan Middendorp (Auteur)

Titre: ISBN: 9063692234

---

Titre: Thinking with Type: A Critical Guide for Designers, Writers, Editors, & Students (Design Briefs)

Auteur: Ellen Lupton (Auteur)

Titre: ISBN: 9781568984483

---

Titre: LETTERS OF CREDIT: A view of type design

Auteur: Tracy, Walter (Auteur)

Titre: ISBN: 0879236361 / 0-87923-636-1

---

Titre: The Elements Of Typographic Style: Version 3.1

Auteur: Bringhurst, Robert (Auteur)

Titre: ISBN: 0881792063 / 0-88179-206-3

---

Titre: Type: The Secret History of Letters

Auteur: Simon Loxley (Auteur)

Titre: ISBN: 1845110285

---

Titre: Type Designs

Auteur: AF Johnson (Auteur)

---

Titre: Typography: Macro & Microaesthetics

Auteur: Willy Kunz (Auteur)

Titre: ISBN: 3721203488

---

Titre: Fonts and encodings

Auteur: Yannis Haralambous (Auteur), P Scott Horne (Translator)

Titre: ISBN-10: 0596102429 | ISBN-13: 978-0596102425

---

Titre: [The Unicode Standard, Version 7.0.0](#)

Auteur: The Unicode Consortium (Auteur)

ISBN: 978-1-936213-09-2

---

# Glossaire

## A

### Abjad

Abjad est le terme technique pour le type de système d'écriture utilisé par les langues sémitiques (hébreu, arabe, etc.), où il y a des glyphes pour toutes les consonnes, mais le lecteur doit être prêt à deviner quelle voyelle ajouter entre deux consonnes.

L'hébreu et l'arabe ont des signes de voyelles facultatifs et sont appelés abjads "impurs". Le phénicien antique n'avait rien d'autre que des consonnes et un abjad "pur".

Voir aussi: alphabet, abugida, syllabaire et [l'article pertinent sur Wikipedia](#).

### Abugida

Un abugida est quelque part entre un alphabet et un syllabaire. Les systèmes d'écriture indiens sont probablement les abugidas les plus connus.

Dans la plupart des abugidas, il existe des glyphes indépendants pour les consonnes, et chaque consonne est implicitement suivie d'un son vocale par défaut. Toutes les voyelles autres que la valeur par défaut seront marquées soit par des signes diacritiques, soit par une autre modification de la consonne de base.

Un abugida diffère d'un syllabaire en ce sens qu'il y a un thème commun aux images représentant une syllabe commençant par une consonne donnée (c'est-à-dire le glyphe pour la consonne), tandis que dans un syllabaire chaque syllabe est distincte même si deux de celles-ci commencent par une consonne commune.

Un abugida diffère d'un abjad en ce que les voyelles (autres que celles par défaut) doivent être marquées dans l'abugida.

Voir aussi: alphabet, abjad, syllabaire et [l'article pertinent sur Wikipedia](#).

### Accent

Voir Diacritiques

### Ajustement à la grille

Avant que les glyphes TrueType ne soient rasterisés, ils passent par un processus d'ajustement à la grille où un programme minuscule (associé à chaque glyphe) est exécuté, ce qui déplace les points sur les contours du glyphe jusqu'à ce qu'ils correspondent mieux à la grille de pixels.

### Alphabet

Un système d'écriture où il y a des glyphes pour tous les phonèmes — consonnes et voyelles semblables — et (en théorie de toute façon) tous les phonèmes dans un mot seront marqués par un glyphe approprié.

Voir aussi: abjad, abugida, syllabaire et [l'article pertinent sur Wikipedia](#).

### AAT (Apple Advanced Typography)

L'extension d'Apple aux polices TrueType de base. Comprend les substitutions contextuelles, les ligatures, le crénage, etc. Inclut également les polices déformables.

## Approche droite

La distance horizontale de l'extrémité la plus à droite d'un glyphe à l'extrémité de la chasse du glyphe. Cela peut être positif ou négatif.

## Approche gauche

La distance horizontale de l'origine d'un glyphe à son extrême gauche. Cela peut être négatif ou positif.

## Ascendante

Le jambage sur une lettre minuscule qui s'étend au-dessus de la hauteur d'x. "l" a une ascendante.

Voir aussi hauteur d'x, hauteur de capitale, descendante, overshoot, ligne de base.

## Classe d'ancrage

Utilisé pour spécifier les positionnement de signe sur base et les sous-tables GPOS cursives.

## Ascent

Dans la typographie traditionnelle, l'ascent (en anglais) d'une police était la distance du haut d'un bloc de fonte à la ligne de base.

Sa signification précise dans la typographie moderne semble varier avec les différents définisseurs.

## ATSUI

Le système typographique avancé d'Apple. Appelée également *Apple Advanced Typography*.

## B

### Bidi

Texte bidirectionnel. Il s'agit d'une section de texte contenant des écritures de gauche à droite et de droite à gauche. Texte en anglais citant l'arabe, par exemple. Les choses deviennent encore plus complexes avec des citations imbriquées. La norme Unicode contient un algorithme pour la mise en page du texte Bidi. Voir aussi: Boustrophédon.

### Black letter (lettre noire)

N'importe quelle famille de polices basée sur l'écriture médiévale.

Voir aussi Gothique.

### Bopomofo

Un alphabet chinois (mandarin) moderne (~ 1911) utilisé pour fournir la translittération phonétique des idéogrammes Han dans les dictionnaires.

### Boustrophédon

Écriture à la manière du bœuf marquant les sillons dans un champ, allant de droite à gauche puis de gauche à droite. Les premiers alphabets (ancien cananéen, et les écrits grecs très anciens (et étonnamment, le fupark) l'ont utilisé. Souvent, les glyphes de droite à gauche seraient des miroirs de ceux de gauche à droite. Pour autant que je sache, aucun système d'écriture moderne n'utilise cette méthode (ni OpenType ne le supporte). Voir également Bidi.

## C

### Cadratin

Le cadratin, en typographie, est une unité de mesure de longueur des espaces. Sa traduction anglaise em (de la lettre M) est utilisée comme symbole de l'unité. FontForge utilise le mot cadratin et unité-em avec la même signification.

Voir em, Unité em

### Caractère

Un caractère est un idéal platonicien réifié en au moins un glyphe. Par exemple la lettre "s" est un caractère qui est réifié en plusieurs glyphes différents: "S", "s", "ſ", "ß", long-s, etc. Notez que ces glyphes peuvent apparaître assez différents les uns des autres, cependant, bien que le glyphe d'un signe intégral puisse être le même que le glyphe de long-s, ce sont en fait des caractères différents.

### CFF (Compact Font Format)

Format de police le plus souvent utilisé dans les polices PostScript OpenType, mais il est valide même sans un wrapper SFNT. Il s'agit du format de police natif pour les polices dotées de charstrings PostScript Type2.

### Chasse

La distance entre le début d'un glyphe et le début du glyphe suivant. Parfois appelée la largeur du glyphe. Voir aussi Chasse verticale.

### Chasse fixe (monospace)

Une police dans laquelle tous les glyphes ont la même largeur. Celles-ci sont parfois appelés polices de machine à écrire.

### Chasse verticale

Le texte CJC est souvent écrit verticalement (et parfois horizontalement), de sorte que chaque glyphe CJC a une chasse verticale ainsi qu'une chasse horizontale.

### CID (Character Identifier)

Identifiant de caractère, un nombre. Dans certaines polices PostScript CJC, les glyphes ne sont pas nommés mais sont désignés par un numéro CID.

### CJC

Chinois, japonais, coréen. Ces trois langues requièrent des polices avec un grand nombre de glyphes. Toutes les trois partagent un système d'écriture basé sur des idéogrammes chinois (bien qu'ils aient subi une évolution distincte dans chaque pays, en fait, les polices chinoises continentales sont différentes de celles utilisées à Taiwan et à Hong Kong).

Le japonais et le coréen ont également des syllabaires phonétiques. Les Japonais ont deux syllabaires, Hiragana et katakana qui ont environ 60 syllabes. Les Coréens ont un syllabaire, hangul avec des dizaines de milliers de syllabes.

### CJCV

Chinois, japonais, coréen, vietnamien. Ces quatre langues nécessitent des polices avec un grand nombre de glyphes.

### Codage

Un codage est un mappage à partir d'un ensemble d'octets sur un jeu de caractères. C'est ce qui détermine quelle séquence d'octets représente quel caractère. Les mots "codage" et "jeu de caractères" sont souvent utilisés de façon synonyme. La spécification pour ASCII spécifie à la fois un jeu de caractères et un codage. Mais les jeux de caractères CJC ont souvent des codages multiples pour le jeu de caractères (et plusieurs jeux de caractères pour certains codages).

Dans les cas plus complexes, il est possible d'avoir plusieurs glyphes associés à chaque caractère (comme en arabe où la plupart des caractères ont au moins 4 glyphes différents) et le programme client doit choisir le glyphe approprié pour le caractère dans le contexte actuel.

## Condensée

Une police condensée est celle où l'espace entre les fûts des glyphes et la distance entre les glyphes eux-mêmes a été réduite.

## Contrepoinçon

Le contrepoinçon d'un glyphe est la partie blanche qui est totalement ou partiellement fermée. Les o et n ont tous deux des contrepoinçons. Le i et l n'en ont pas. Les e et le d ont tous deux des contrepoinçons. Le B a deux contrepoinçons.

## Courbe (Spline)

La version originale anglaise de FontForge utilise le terme « spline » pour définir un segment de ligne courbe. En réalité, les splines utilisées dans FontForge sont des courbes de Bézier de deuxième degré (quadratique) ou de troisième degré (cubique) ou des clothoïdes de Raph Levien.

Pour éviter cette ambiguïté, la version française n'utilise que le terme "courbe".

## Courbe de Bézier ou splines de Bézier

Les courbes de Bézier sont décrites en détail dans la section Bézier du manuel principal (en anglais).

## Crénage

Lorsque l'espacement par défaut entre deux glyphes est inapproprié, la police peut inclure des informations supplémentaires pour indiquer que, lorsqu'un glyphe donné (par exemple "T") est suivi d'un autre glyphe (par exemple "o"), la chasse du "T" soit ajustée d'un certain montant pour rendre l'affichage plus agréable.

À l'époque des fontes de métal, le métal devait être limé pour fournir un bon ajustement. Par exemple, le "F" avait eu un peu de métal enlevé de sorte qu'une lettre minuscule pouvait se blottir plus près de celui-ci.

## Crénage par classes

Les glyphes de la police sont divisés en classes de glyphes et il ya une grande table qui spécifie le crénage pour chaque combinaison possible de classes. Généralement, cela sera plus petit que l'ensemble équivalent de paires de crénage car chaque classe contiendra plusieurs glyphes.

## D

### Descendante

Un jambage sur une lettre minuscule qui s'étend au-dessous de la ligne de base. "p" a une descendante. Voir aussi Hauteur'dx, Hauteur de capitale, Ascendante, Overshoot, Ligne de base

### Descent

Dans la typographie traditionnelle, la descent (en anglais) d'une police était la distance entre le bas d'un bloc de police et la ligne de base.

Son sens dans la typographie moderne est devenu moins précis.

## Diacritiques

Beaucoup de langues utilisent des lettres qui ont des signes au-dessus ou en dessous d'eux ou même qui traversent les lettres. Ces signes sont appelées diacritiques. Ils sont également appelés «accents» bien que ce soit un terme moins précis. Voici quelques exemples de ces lettres: Å à å Ä Ü ü Ø ø Ç ç.

## E

### Écriture

Une écriture est un ensemble de caractères et ses règles associées pour mettre des caractères ensemble. Le latin, l'arabe, le katakana et le hanja sont tous des écritures.

Voir aussi Alphabet.

### em

Une unité linéaire égale à la taille en points de la police. Dans une police de 10 points, le em sera de 10 points. Un em-space est espace blanc qui est aussi large que la taille en point. Un em-dash est une barre horizontale qui est aussi large que la taille en point.

Un em-square est un carré, un em de chaque côté. Dans la typographie traditionnelle (quand chaque lettre était coulée dans le métal) le glyphe devait être dessiné dans le em-square.

## Empattement (Serif)

Il y a deux mille ans, quand les Romains gravaient leurs lettres sur des monuments de pierre, ils ont découvert qu'ils pouvaient réduire le risque de fissure en ajoutant des lignes fines aux extrémités des fûts principaux d'un glyphe.

Ces lignes fines ont été appelées empattements. Les premiers typographes les ajoutent à leurs fontes pour des raisons esthétiques plutôt que fonctionnelles.

A la fin du XIXe et au début du XXe siècle, les typographes ont commencé à concevoir des fontes sans empattements. Celles-ci ont d'abord été appelées grotesques parce que leur forme paraissait étrange. Elles sont maintenant généralement appelées sans-serif.

D'autres systèmes d'écriture (l'hébreu par exemple) ont leurs propres empattements. Les empattements hébreus sont assez différents des empattements latins, grecs ou cyrilliques car ils apparaissent seulement au sommet des glyphes.

### en

La moitié d'un "em"

## Eth — Edh

L'ancienne lettre germanique "ð" pour le son (anglais) "th" (le son dans «this» — la plupart des anglophones ne sont même pas conscients que "th" en anglais a deux sons qui lui est associé, mais c'est le cas. Voir aussi Thorn.

## Espace blanc

L'espace blanc d'un design de police comprend l'espace entre les lignes de texte, l'espace entre les lettres, l'espace inter-mot et les espaces à l'intérieur des lettres. C'est un terme large et englobant.

## Étendue

Une police étendue est celle où l'espace entre les fûts des glyphes, et la distance entre les glyphes eux-mêmes a été augmentée.

## Extremum (*plurel: Extrema*)

Un extremum est le point sur une courbe mathématique où la courbe atteint sa valeur maximale ou minimale. Sur une courbe continue, cela peut se produire aux extrémités ou à  $dx/dt=0$  ou à  $dy/dt=0$ .

Dans le design de police, les extrema d'un glyphe sont les points les plus hauts et les plus bas du contour, ainsi que ses points les plus à gauche et les plus à droite. Il est important de s'assurer qu'un glyphe a des points sur la courbe à tous ses extrema, car cela simplifie le rendu du texte lorsque la police est utilisée.

## F

### Famille de fontes, ou Famille

Une collection de polices associées. Souvent incluant les styles réguliers, italiques et gras.

### Fichier de fonctionnalité (feature file)

Il s'agit d'une syntaxe de texte conçue par Adobe pour décrire les fonctionnalités OpenType. Il peut être utilisé pour déplacer des informations de fonctionnalité et de lookup d'une police à l'autre.

### Fonctionnalité (OpenType)

Lors de la création des fontes pour les écritures complexes (et même pour celles qui le sont moins), diverses transformations (comme les ligatures) doivent être appliquées aux glyphes avant de les afficher. Ces transformations sont identifiées comme fonctionnalités de fonte et sont étiquetées avec une balise de 4 lettres (OpenType) ou un identifiant de 2 chiffres (Apple). La signification de ces fonctionnalités est prédéfinie par Microsoft et Apple. FontForge vous permet d'étiqueter chaque lookup avec une ou plusieurs fonctionnalités lorsque vous le créez (ou plus tard).

### Fonctionnalité/Paramètres (Apple)

Ceux-ci sont à peu près les équivalents des fonctionnalités OpenType décrit précédemment. Ils sont définis par Apple.

## Fonte

Une fonte est l'ensemble des caractères correspondant *aux mêmes* caractéristiques de corps, graisse et italique au sein d'une même police. Donc plusieurs fontes peuvent être incluses dans une police de caractères.

La fonte est une collection de glyphes, généralement avec au moins un glyphe associé à chaque caractère du jeu de caractères de la fonte, souvent avec un codage.

Une fonte contient une grande partie des informations nécessaires pour transformer une séquence d'octets en un ensemble d'images représentant les caractères spécifiés par ces octets.

Dans la typographie traditionnelle une fonte était une collection de petits blocs de métal chacun avec une image gravée d'une lettre sur ceux-ci. Traditionnellement, il existait une fonte différente pour chaque taille de point.

Voir Police de caractères.



## FreeType

Une librairie pour rasteriser les polices. Utilisé abondamment dans FontForge pour comprendre le comportement des polices TrueType et pour faire une meilleure rasterisation.

## Fupark (Futhark)

La vieille écriture runique germanique.

## Fût

Le fût est la partie de la lettre qui est verticale. Les l et l sont tous composés de fûts, à l'exception des empattements. Le H se compose de deux fûts et d'une traverse. D'autres glyphes avec fûts comprennent B b F f K k P p R r 1 et 4.

## G

### Glyphe

Un glyphe est une image, souvent associée à un ou plusieurs caractères. Ainsi, le glyphe utilisé pour tracer "f" est associé au caractère f, tandis que le glyphe de la ligature "fi" est associé à f et i. Dans les polices latines simples, l'association est souvent de un pour un (il y a exactement un glyphe pour chaque caractère), tandis que dans les polices ou écritures plus complexes, il peut y avoir plusieurs glyphes par caractère (dans l'imprimerie de la Renaissance, la lettre "s" avait deux glyphes associés: le long-s qui a été utilisé initialement et médialement, et le court-s qui a été utilisé seulement à la fin des mots). Et dans les ligatures un glyphe est associé à deux ou plusieurs caractères.

Les fontes sont des collections de glyphes avec une certaine forme de correspondance entre les caractères et les glyphes.

### Graisse

La graisse d'une police est l'épaisseur (sombre) des fûts des glyphes. Traditionnellement la graisse est nommée, mais récemment des nombres ont été appliqués aux graisses.

Nom	Nombre
Mince	100
Extra-léger	200
Léger	300
Normal, régulier	400
Moyen	500
Demi-gras	600
Gras	700
Extra-gras	800
Noir	900
Extra-noir	
Ultra-noir	

### Gras

Un style de police commun. Les fûts des glyphes sont plus larges que dans la police normale, donnant aux lettres une impression plus foncée. Le gras est l'un des rares styles LGC qui se transposent facilement à d'autres écritures.

## Tables graphite

Graphite est une extension de TrueType qui intègre plusieurs tables dans une police contenant des règles pour la mise en forme contextuelle, les ligatures, le réordonnancement, les glyphes divisés, la bidirectionnalité, les diacritiques empilés, le positionnement complexe, etc.

Cela ressemble plutôt à OpenType — sauf que OpenType dépend des routines de mise en page de texte qui en savent beaucoup sur les glyphes impliqués. Cela signifie que les polices OpenType ne peuvent pas être conçues pour une nouvelle langue ou un nouvelle écriture sans avoir à expédier une nouvelle version du système d'exploitation. Alors que les tables Graphite contiennent toutes ces informations cachées.

La typographie avancée d'Apple offre une meilleure comparaison, mais les tables Graphite sont censées être plus faciles à construire.

SIL International fournit gratuitement un compilateur Graphite.

## Grotesque

Voir aussi Sans-serif.

## Gothique

Les moines allemands à l'époque de Gutenberg utilisaient un style d'écriture à lettre noire, et il copia leur écriture manuscrite dans ses caractères d'imprimerie. Les designers de police italiens (après que l'imprimerie s'étendit vers le sud) se moquèrent du style, préférant les designs de police laissées par les Romains. Comme terme de mépris, ils ont utilisé le mot gothique, le style des Goths qui ont aidé à détruire l'empire romain.

## H

### Han

Les caractères idéographiques utilisés en Chine, au Japon et en Corée (et, je crois, dans divers autres pays asiatiques aussi (Vietnam?)), tous basés sur le style d'écriture qui a évolué en Chine.

### Hangul

Le syllabaire coréen. Le seul syllabaire (que je connais de toute façon) basé sur un alphabet — les lettres de l'alphabet ne paraissent jamais seules, mais seulement comme des groupes de deux ou trois qui forment une syllabe.

### Hanja

Le nom coréen des caractères Han.

## Hauteur de capitale

La hauteur d'une lettre majuscule au-dessus de la ligne de base (une lettre avec un sommet plat comme "I" par opposition à une avec une courbe comme "O").

Voir aussi Hauteur d'x, Ascendante, Descendante, Overshoot, Ligne de base

## Hauteur de fonte

À l'époque des fontes de métal, c'était la hauteur du morceau de métal — la distance de la surface d'impression à la plateforme sur laquelle il reposait.

## Hauteur d'x

La hauteur d'une lettre minuscule au-dessus de la ligne de base (avec un sommet plat comme "x" ou "z" ou "v" par opposition à une avec sommet courbé comme "o" ou une avec une ascendante comme "l").

Voir aussi Hauteur de capitale, Ascendante, Descendante, Overshoot, Ligne de base.

## Hints

Ceux-ci sont décrits en détail dans le manuel principal. Ils aident le "rasterizer" à bien dessiner un glyphe à de petites tailles en points.

## Hints contradictoires

Si un glyphe contient deux hints où le point de départ ou de fin de l'un est dans la plage de l'autre, alors ces hints sont en conflit. Ils ne peuvent pas être actifs simultanément.

## Hints fantômes

Parfois, il est important d'indiquer qu'un bord horizontal est bien horizontal. Mais le bord n'a pas de bord correspondant avec lequel faire un fût normal. Dans ce cas, un hint spécial est utilisé avec une largeur de -20 (ou -21). Un hint fantôme (ghost en anglais) doit se trouver entièrement dans un glyphe. Si il est en haut d'un contour, utilisez une largeur de -20, si il est au bas, utilisez -21. Des hints fantômes devraient également se trouver dans les BlueZones.

(La spécification mentionne également des hints fantômes verticaux, mais comme il n'y a pas de bluezone verticale, il n'est pas clair comment ils devraient être utilisés).

## Hiragana

Un des deux syllabaires japonais. Le Hiragana et le Katakana ont les mêmes sons.

## I

## Idéogramme

Un caractère unique qui représente un concept sans l'expliquer. Généralement utilisé pour désigner les caractères Han (chinois).

## Interligne

La distance entre les lignes successives d'une police.

## Italique

Le style oblique d'une police, souvent utilisé pour mettre l'emphase.

L'italique diffère de l'oblique en ce que la transformation de la forme régulière à la forme inclinée implique plus que simplement incliner les lettres. Généralement, les minuscules a changent en *a*, les empattements sur les lettres minuscules comme *i* (*i*) changent, et la police gagne généralement un sentiment plus fluide.

## J

## Jamo

Les lettres de l'alphabet coréen. Celles-ci ne sont presque jamais seules, apparaissant généralement en groupes de trois dans une syllabe Hangul. Les Jamos sont divisés en trois catégories (avec un chevauchement considérable entre la première et la troisième), le choixong — consonnes initiales, les voyelles jungseong — médiales et les consonnes jongseong — finales. Une syllabe se compose en plaçant un glyphe de choixong en haut à gauche d'un em-square, un jungseong en haut à droite, et éventuellement un jongseong dans la partie inférieure du carré.

## Jeu de caractères

Un jeu de caractères est un jeu de caractères non ordonné.

## K

### Kanji

Le nom japonais des caractères Han.

### Katakana

Un des deux syllabaires japonais (modernes). Le Hiragana et le Katakana ont les mêmes sons.

### Knuth, Donald

Un mathématicien qui en a eu tellement marre de la mauvaise composition dans les années 1970 & 80s qu'il a créé son propre système de design de police et programme de mise en page typographique appelé, respectivement, MetaFont et TeX.

## L

### Largeur

C'est un terme légèrement ambigu et parfois utilisé pour désigner la chasse (la distance entre le début du glyphe et le début du glyphe suivant), et parfois utilisé pour signifier la distance de l'approche gauche à l'approche droite.

### Lémur

Un genre monotypique de primates prosimiens, qu'on ne retrouve plus qu'à Madagascar mais à l'origine (il y a environ 50 millions d'années) les membres de cette famille étaient beaucoup plus étendus.

### Ligature

Un glyphe unique qui se compose de deux glyphes adjacents. Un exemple courant dans l'écriture latine est la ligature "œ" qui est plus agréable à l'œil que que la séquence.

### Ligne de base

La ligne de base est la ligne horizontale sur laquelle les lettres (latines, grecques, cyrilliques) se trouvent. La ligne de base sera probablement dans un endroit différent pour les différentes écritures. Dans les écritures indiennes, la plupart des lettres descendent sous la ligne de base. Dans les écritures CJC, il existe également une ligne de base verticale généralement au milieu du glyphe. Les tables BASE et bsln vous permettent de spécifier comment les lignes de base des différentes écritures doivent être alignées les unes par rapport aux autres.

Voir aussi, Hauteur d'x, Hauteur de capitale, Ascendane, Descandante, Overshoot

## Liste de noms

Un mappage à partir du point de code Unicode vers le nom du glyphe.

## LGC

Latin, grec, cyrillique. Ces trois alphabets ont évolué côte à côte au cours des derniers millénaires. Les formes de lettres sont très semblables (et certaines lettres sont partagées). De nombreux concepts tels que "minuscules", "italiques" s'appliquent à ces trois alphabets et à aucun autre. (OK, l'Arménien a aussi des lettres minuscules).

## M

### Machine à écrire

Voir Chasse fixe (monospace)

### Machine d'état

Une machine d'état est un très petit programme simple. Ces programmes sont utilisés sur le Mac pour effectuer des substitutions contextuelles et le crénage. La boîte de dialogue de la machine d'état est accessible à partir d'Élément->Infos fonte>Lookups.

La "machine d'état" se compose d'une table d'états, chaque état à son tour se compose d'une série de transitions potentielles (aux mêmes ou différents états) en fonction de l'entrée. Dans les machines d'état dans les polices, la machine démarre dans un état spécial appelé état de démarrage et lit le flux de glyphe du texte. Chaque glyphe individuel provoquera une transition d'état. Lorsque ces transitions se produisent, la machine peut également spécifier des modifications du flux de glyphe (substitutions conditionnelles ou crénage).

### Manyogana

Une ancienne écriture japonaise, ancestrale à l'hiragana et au katakana. Le Manyogana a utilisé le kanji pour ses sons phonétiques, et au fil des ans ces kanjis ont été simplifiés en hiragana et katakana.

### Masques de Hint

À tout point donné sur un contour des hints ne peuvent pas entrer en conflit. Cependant, des points différents d'un glyphe peuvent nécessiter des hints contradictoires. Donc de temps en temps un contour va changer les hints qui sont actifs. Chaque liste de hints actifs est appelée un masque de Hint.

### Matrice de bitmaps

Une taille particulière de police. Plus précisément, une matrice de bitmaps représente la taille en pixel de la police. Par analogie avec la typographie traditionnelle, les matrices sont de petits morceaux de cuivre rouge que l'on veut frapper avec les caractères qui ont été gravés en relief sur un poinçon. Tout comme les bitmaps qui sont de taille fixe (non vectorielle), de nouvelles matrices devaient être frappées pour créer des fontes de tailles différentes.

## O

### Oblique

Un style incliné de police, généralement utilisé pour l'accent.

L'oblique diffère de l'italique en ce sens que la transformation de la forme simple à la forme inclinée implique une déformation mathématique ou mécanique des lettres.

## Ogham

La vieille écriture des inscriptions celtiques.

## OpenType

Un type de police. OpenType est une tentative de fusionner postscript et TrueType dans une seule spécification. Une fonte OpenType peut contenir une fonte TrueType ou Postscript.

Une fonte OpenType contient les mêmes tables de données pour des informations telles que les codages qui étaient présentes dans les fontes TrueType.

Confusément, ce type est également utilisé pour désigner les tables typographiques avancées d'Adobe et de Microsoft (mais pas d'Apple) qui ont été ajoutées à TrueType. Ces tables comprennent des informations comme les ligatures contextuelles, le crénage contextuel, la substitution de glyphes, etc.

Et MS Windows l'utilise pour désigner une police avec une table de signature digitale 'DSIG'.

## Overshoot

Pour que la forme incurvée du "O" apparaisse à la même hauteur que le sommet plat du "I", elle doit dépasser (ce qu'on appelle un "overshoot") la hauteur de capitale ou la hauteur d'x, ou dépasser (ce qu'on appelle un "undershoot") la ligne de base d'environ 3% de la hauteur de capitale ou de la hauteur d'x. Pour une forme triangulaire (telle que le "A"), le dépassement est encore plus grand, peut-être 5%.

Ces directives sont basées sur la façon dont l'œil fonctionne et les illusions d'optique qu'il génère selon *Digital Formats for Typefaces*, p. 26 de Peter Karow.

Le dépassement dépend également de la taille en point de la police, plus la taille est grande, plus le dépassement doit être petit. Généralement, les polices modernes seront utilisées à plusieurs tailles. Dans certaines familles de polices, il existe plusieurs faces pour les différentes tailles de points, et dans ce cas le dépassement va varier d'une face à l'autre.

Voir aussi Hauteur d'x, Hauteur de capitale, Ascendante, Descendante, ligne de base

## P

### Paire de crénage

Une paire de glyphes pour lesquels les informations de crénage ont été spécifiées.

### Panose

Système de description des polices. Voir [Panose Font Classification System Metrics Guide](#). Il existe également une extension appelée [Panose 2.0](#).

FontForge ne connaît que le système de classification des polices latines. D'autres systèmes existent pour d'autres écritures.

### Panse

La partie ronde de la lettre.

### PfaEdit

C'est l'ancien nom de FontForge. La conception originale ne permettait d'éditer que des polices ASCII de type 1 (d'où le nom). Elle s'est rapidement métamorphosée au-delà de ce point, mais il a fallu trois ans pour renommer l'application.

## Points fantômes

Dans une police TrueType, il y a quelques points ajoutés à chaque glyphe qui ne sont pas sur les contours qui composent le glyphe. Ces points sont appelés points fantômes. Un de ces points représente l'approche gauche, et l'autre la chasse du glyphe. Les instructions TrueType (hints) sont autorisées à déplacer ces points tout comme n'importe quels autres points peuvent être déplacés — changeant ainsi l'approche gauche ou la chasse. Les premières versions de TrueType n'incluaient que ces deux points fantômes, les versions plus récentes incluent également un point pour l'approche supérieure et un point pour la chasse verticale.

## PCS

Plan complémentaire spécialisé (0xE0000-0xEFFFF) d'Unicode. Peu utilisé pour quoi que ce soit. Voir également

- PMB: Plan multilingue de base (0x00000-0x0FFFF)
- PMC: Plan multilingue complémentaire (0x10000-0x1FFFF)
- PIC: Plan idéographique complémentaire (0x20000-0x2FFFF)

## PIC

Plan idéographique complémentaire (0x20000-0x2FFFF) d'Unicode. Utilisé pour les caractères Han rares (la plupart ne sont plus d'usage courant). Voir également:

- PMB: Plan multilingue de base (0x00000-0x0FFFF)
- PMC: Plan multilingue complémentaire (0x10000-0x1FFFF)
- PCS: Plan complémentaire spécialisé (0xE0000-0xEFFFF)

## Pica

Une unité de longueur définie (aux États-Unis du moins) à 35/83 cm (ou environ 1/6 de pouce). Cela a été utilisé pour mesurer la longueur des lignes de texte (comme "30 picas et 4 points de long"), mais pas pour la mesure des hauteurs de police.

Dans la typographie de la Renaissance, avant qu'il y ait des points, les tailles de fonte ont des noms, et "pica" a été utilisé dans ce contexte: "Great Canon", "Double Pica", "Great Primer", "English", "Pica", "Primer", "Small Pica", "Brevier", "Nonpareil" et "Pearl" (chaque nom représentant une taille de police progressivement plus petite). Voir le [spécimen de composition de fontes de Caslon sur Wikipedia](#). En France l'équivalent de ces tailles est: Gros-Canon (44 pt), Palestine (24 pt), Gros-Romain (18 pt), Saint-Augustin (14 pt), Cicéro (12 pt), Petit-romain (10 pt), Petit-texte (8 pt), Nompaille (6 pt) et Parisienne (5 pt)

## Point

Un point est une unité de mesure. Il y avait (au moins) trois différentes définitions de "point" en usage avant l'avènement des ordinateurs. Celui utilisé dans le monde de l'impression anglo-saxonne était le "point pica" avec 72,27 points par pouce (2,85pt/mm), tandis que celui utilisé en Europe continentale était le point Didot avec 62 2/3 points par 23,566mm (2,66pt/mm ou 67,54pt/pouce) et les Français ont parfois utilisé le point médian (72,78 points par pouce, 2,86pt/mm).

Les points didot et pica étaient disposés de telle sorte que le texte à une taille en point donnée aurait approximativement la même hauteur de capitale dans les deux systèmes, le point didot avait un espace blanc supplémentaire au-dessus des capitales pour contenir les accents présents dans la plupart des écritures non-anglaise en latin.

Un effet secondaire intéressant d'une police conçue pour l'usage européen est d'avoir une plus petite proportion de cadratin vertical donné au corps de texte. Je crois que les polices informatiques ont tendance à ignorer cela, donc présumément les imprimantes européennes mettent plus d'espace en tête.

Autant que je sache, les ordinateurs ont tendance à travailler dans des approximations de points pica (mais c'est peut être parce que je suis aux États-Unis). PostScript utilise une unité de 1/72ème de pouce.

À l'origine les polices n'étaient pas décrites par taille en point, mais par nom. Ce n'est qu'en 1730 que Pierre Fournier créa le système de points pour spécifier les hauteurs des polices. Celle-ci fut plus tard améliorée par François-Ambroise Didot (d'où le nom du point). En 1878, le Chicago Type Foundry utilisa la première un système de points aux États-Unis. En 1886, le point américain a été normalisé — le pica a été défini comme étant 35/83cm, et le point pica défini comme étant 1/12ème de cela.

## Point d'inflexion

Le point de la courbe où elle change de concave vers le bas à concave vers le haut (ou vice versa). Ou en termes mathématiques (pour les courbes continues) où  $d^2y/dx^2=0$  ou infini. Les splines cubiques peuvent contenir des points d'inflexion, les splines quadratiques n'en ont pas.

## Point Pica

Le point anglo-américain. Avec 72,27 points par pouce (2,85 pt/mm).

## Police CID-keyed

Une police PostScript dans laquelle les glyphes sont indexés par CID et non par nom.

## Police déformable

Voir Multi-maître

## Police de caractères

Ensemble de glyphes, représentations visuelles de caractères d'une même famille, qui regroupe *tous* les corps et graisses d'une même famille, dont le style est coordonné, afin de former un alphabet, ou la représentation de l'ensemble des caractères d'un langage, complet et cohérent. Une police de caractères peut donc inclure plusieurs fontes.

Voir fonte.

## Polices multicouches

(Le propre terme de FontForge) Les polices PostScript type 3 et les polices SVG permettent plus de possibilités de dessin que les polices normales. Les polices normales ne peuvent être remplies que par une seule couleur héritée de l'environnement graphique. Ces deux polices peuvent être remplies de plusieurs couleurs différentes, être tracées, inclure des images, des remplissages de dégradé, etc.

## Police multi-maître

Une police multi-maître est un schéma de police PostScript qui définit un nombre infini de polices associées. Plusieurs polices principales peuvent varier le long de plusieurs axes, par exemple vous pouvez avoir un multi-maître qui définit à la fois des graisses différentes et des largeurs différentes d'une famille de polices afin de générer mince, régulier, demi-gras, gras, condensé, étendu, gras-condensé, etc.

Adobe ne développe plus ce format. Apple a un format qui a le même effet, mais n'a pas produit de nombreux exemples.

FontForge prend en charge les deux.

## PostScript

PostScript est un langage de mise en page utilisé par de nombreuses imprimantes. Le langage contient les spécifications de plusieurs formats de police différents. Le manuel principal de FontForge contient une section décrivant comment PostScript diffère de TrueType.

- Type 1: Il s'agit de l'ancienne norme pour les polices PostScript. Une telle police a généralement l'extension .pfb (ou



.pfa). Une police de type 1 est limitée à un codage d'un octet (c'est-à-dire que seulement 256 glyphes peuvent être codés).

- Type 2/CFF: C'est le format utilisé dans les polices OpenType. Il est presque identique à Type 1, mais a quelques extensions et un format plus compact. Il est généralement à l'intérieur d'un wrapper CFF, qui est généralement à l'intérieur d'une police OpenType. Le format de police CFF ne permet à nouveau qu'un codage de 1 octet, mais le wrapper OpenType étend cela pour fournir des types de codage plus complexes.
- Type 3: Ce format permet un postscript complet dans la police, mais cela signifie qu'aucun hints n'est autorisé, de sorte que ces polices ne seront pas aussi jolies à de petites tailles en points. En outre, la plupart des rasterizers (écrans) sont incapables de les traiter. Une police de type 3 est limitée à un codage d'un octet (c'est-à-dire que seulement 256 glyphes peuvent être codés).
- Type 0: Ce format est utilisé pour regrouper de nombreuses sous-polices (de Type 1, 2 ou 3) en une seule grande police avec un codage multi-octet, et a été utilisé pour les polices CJC ou Unicode.
- Type 42: Une police TrueType enveloppée dans PostScript. En quelque sorte l'opposé de OpenType.
- CID: Ce format est utilisé pour les polices CJC avec un grand nombre de glyphes. Les glyphes eux-mêmes sont spécifiés soit comme format de glyphe type1 ou type2. La police CID elle-même n'a pas de codage, juste un mappage de CID (un nombre) vers un glyphe. Un ensemble de fichiers CMAP externes est utilisé pour fournir les codages appropriés au besoin.

## PMB (Plan multilingue de base)

Les premiers 65536 points de code d'Unicode. Ceux-ci contiennent la plupart des caractères ordinaires dans le monde moderne. Voir également

- PMC: Plan multilingue complémentaire (0x10000-0x1FFFF)
- PIC: Plan idéographique complémentaire (0x20000-0x2FFFF)
- PCS: Plan complémentaire spécialisé (0xE0000-0xEFFFF)

## PMC

Plan multilingue complémentaire (0x10000-0x1FFFF) d'Unicode. Utilisé pour les alphabets anciens et artificiels et syllabaires — comme Linéaire B, gothique et Shavian. Voir également:

- PMB: Plan multilingue de base (0x00000-0x0FFFF)
- PIC: Plan idéographique complémentaire (0x20000-0x2FFFF)
- PCS: Plan complémentaire spécialisé (0xE0000-0xEFFFF)

## Python

Un langage de programmation informatique qui met l'accent sur la lisibilité du code.

## R

### Référence

Une référence est un moyen de stocker les contours d'un glyphe dans un autre (par exemple en glyphes accentués). Parfois appelé composant.

### Règle de remplissage de la tangente tournante (Non-Zero Winding Rule)

Pour déterminer si un pixel doit être rempli en utilisant cette règle dessiner une ligne d'ici à l'infini (dans n'importe quelle direction) et compter le nombre de fois que les contours traversent cette ligne. Si le contour traverse la ligne dans le sens horaire, ajoutez 1, si le contour croise dans le sens anti-horaire, soustrayez-en un. Si le résultat est différent de zéro, remplissez le pixel. S'il est zéro, laissez-le vide. Cette méthode est utilisée pour rasteriser les polices par TrueType et plus anciennes postscripts (avant version 2).

Voir aussi Règle de remplissage pair-impair.

## Règle de remplissage pair-impair

Pour déterminer si un pixel doit être rempli à l'aide de cette règle, tracez une ligne du pixel à l'infini (dans n'importe quelle direction) puis comptez le nombre de fois où les contours traversent cette ligne. Si ce nombre est impair, remplissez le point, si il est pair alors ne remplissez pas le point. Cette méthode est utilisée pour les polices par les rasterizers postScript après le niveau 2.0 de PostScript. Voir aussi règle de la tangente tournante.

## S

### Sans Serif

Voir Empattement.

### SFD (Spline Font Database)

Base de données SplineFont. Il s'agit du format propriétaire de FontForge pour représenter des polices. Les fichiers sont ASCII et vaguement lisibles, le format est décrit ici. Depuis le 14 mai 2008, le format a été enregistré avec l'IANA pour un type MIME: application/vnd.font-fontforge-sfd.

D'autres personnes utilisent l'acronyme 'sfd' aussi. (Malheureusement)

- Tops-10, sur l'ordinateur Digital PDP-10 a utilisé sfd pour signifier "Sub File Directory". Tops-10 a fait une distinction entre les répertoires de niveau supérieur (home), appelés "répertoires de fichiers utilisateur" et les sous-répertoires.
- TeX l'utilise pour signifier "Sub Font Definition" où un fichier TeX sfd contient des informations sur la façon de diviser une grande police CJC ou Unicode en petites sous-polices, chacune avec un codage de 1 octet dont TeX (ou les versions plus anciennes de TeX) avait besoin.

### SFNT

Le nom du format de police générique qui contient TrueType, OpenType, Apple bitmap only, X11 bitmap only, les polices 'typ1' obsolètes et les polices SING d'Adobe (et sans doute d'autres). Le format SFNT décrit comment les tables de polices doivent être disposées dans un fichier. Chacun des formats ci-dessus suit cette idée générale, mais inclut des exigences plus spécifiques (quelles tables sont nécessaires, et le format de chaque table).

### Style

Il existe plusieurs variantes classiques d'une police. Dans probablement n'importe quel système d'écriture, l'épaisseur des fûts des glyphes peut varier, c'est ce qu'on appelle la graisse d'une police. Les graisses communes sont régulier et gras.

Dans les alphabets LGC, un style italique (ou oblique) a surgi et est utilisé pour mettre l'accent.

Les polices sont souvent compressées en un style condensé, ou élargi dans un style étendu.

Divers autres styles sont utilisés occasionnellement: souligné, barré, contour, ombre.

### SVG

Scalable Vector Graphics (en français graphique vectoriel adaptable). Un format XML utilisé pour dessiner des images vectorielles. Il inclut un format de police.

### Syllabaire

Un syllabaire est un système d'écriture phonétique comme un alphabet. Contrairement à un alphabet, l'unité sonore qui est écrite est une syllabe plutôt qu'un phonème. En japonais Katakana le son "ka" est représenté par un glyphe. Les syllabaires ont tendance à être plus gros que les alphabets (le Katakana japonais nécessite environ 60 caractères différents, alors que le Hangul coréen nécessite des dizaines de milliers).

Voir aussi: Abjad, Abugida, Alphabet et [l'article pertinent sur Wikipédia](#).

## T

### Table de périphérique

Un concept en OpenType qui vous permet d'entrer des réglages d'espacement adaptés à la rasterisation à des tailles de pixels particulières. Si une valeur de crénage qui fonctionne la plupart du temps conduit à une juxtaposition laide de glyphes sur une police de 12 pixels de haut, alors vous pouvez ajouter un réglage spécial à l'espacement qui est seulement applicable à 12 pixels (et un autre à 14 et 18, ou tout ce qui est nécessaire). Une fonctionnalité similaire est nécessaire pour les signes ancrés.

### Tables OpenType

Chaque police OpenType contient une collection de tables contenant chacune certains types d'informations.

### Tables TrueType

Chaque police truetype contient une collection de tables contenant chacune un certain type d'informations.

### Taille en points

Dans la typographie traditionnelle une police de 10pt était celle où le bloc de métal pour chaque glyphe était de 10 points de haut. La taille en points d'une police est la distance minimale entre les ligne de base "sans plomb".

### Terminaison

La terminaison d'un glyphe est la partie où le tracé se termine. Le haut du f a une terminaison. Le s a deux terminaisons. Quand un glyphe a des empattements, les empattements sont considérés comme différents des terminaisons. Parce que le bas du f a un empattement s'il est dans un style serif, le bas n'est pas considéré comme une terminaison. Le bas des j et y sont cependant considérés comme des terminaisons. De même le 3 a deux terminaisons, une au sommet et une au bas. Le milieu est considéré comme une jointure plutôt qu'une terminaison. La classification de ces parties est peut-être plus déterminée par convention que par une logique stricte.

### TeX

Un système logiciel de composition de documents.

### Thorn

La lettre germanique "þ" utilisée en anglais comme consonne fricative dentale sourde "th" (comme dans le mot "thorn"). Je crois que c'est approximativement la même valeur sonore que Thêta en grecque. Actuellement une version corrompue de ce glyphe survit comme "ye" pour le "the". Voir aussi Eth.

### Traverse

Partie horizontale d'un caractère. Il y en a 3 dans la lettre capitale E.

### True Type

Un type de police inventé par Apple et partagé avec Microsoft. Il précise les contours avec des courbes de Bézier de second degré (quadratique), contient des contrôles innovants pour le hinting, et une série de tables extensibles pour contenir toute information supplémentaire importante à la police.

Apple et Adobe/Microsoft ont étendu ces tables de différentes façons pour inclure des fonctions typographiques avancées nécessaires pour les écritures non latines (ou pour les écritures latines complexes).

## Type 1

Un type de police PostScript.

## Type 2

Un type de police PostScript utilisés dans les wrappers de polices OpenType.

## Type 3

Un type très général de police PostScript.

## Type 0

Type de police PostScript.

# U

## Unicode

Un jeu de caractères/codage qui tente de couvrir tous les caractères utilisés actuellement dans le monde, et beaucoup d'anciens aussi. Voir le [consortium Unicode](#).

- PMB: Plan multilingue de base (0x00000-0x0FFFF)
- PMC: Plan multilingue complémentaire (0x10000-0x1FFFF)
- PIC: Plan idéographique complémentaire (0x20000-0x2FFFF)
- PCS: Plan complémentaire spécialisé (0xE0000-0xEFFFF)

## Undershoot

Voir Overshoot.

## UniqueID

Identifiant unique. Il s'agit d'un champ dans une police PostScript. Il a été utilisé à l'origine comme mécanisme d'identification unique de police. Adobe a décidé qu'il n'était pas suffisant et a créé le champ XUID (Extended Unique ID). Adobe a maintenant décidé que les deux sont inutiles.

Il existe un champ très similaire dans la table TrueType 'name'.

## Unité em

Dans une police vectorielle, le "em" est subdivisé en unités. Dans une police postscript il ya habituellement 1000 unités pour les em. Dans une police TrueType, il peut y avoir 512, 1024 ou 2048 unités. Dans une police Ikarus, il ya 15 000 unités. FontForge utilise ces unités comme base de son système de coordonnées.

Voir Cadratin

## UseMyMetrics

C'est un concept truetype qui force la largeur d'un glyphe composite (par exemple une lettre accentuée) à être de la même largeur que la largeur d'un de ses composants (par exemple la lettre de base étant accentuée).

## **X**

### **XUID**

Extended Unique ID. Identifiant unique étendu dans une police PostScript. Maintenant quelque peu obsolète. Voir UniqueID.