

# Pourquoi programmer ?

## Chapitre 1

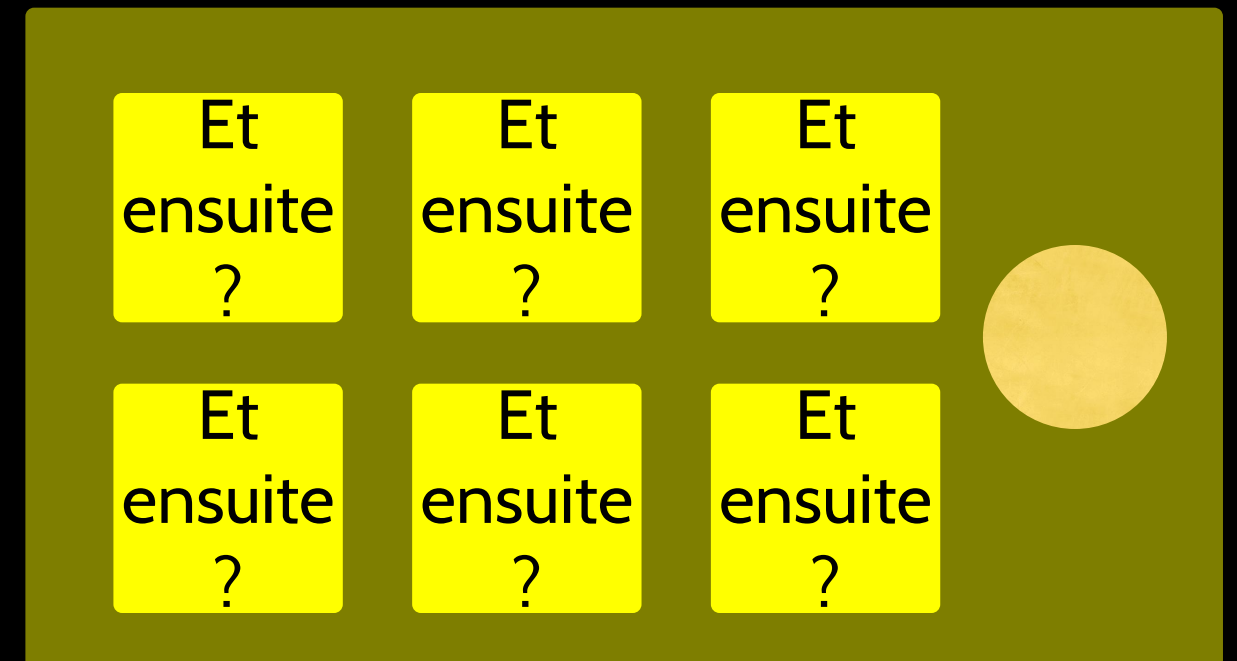
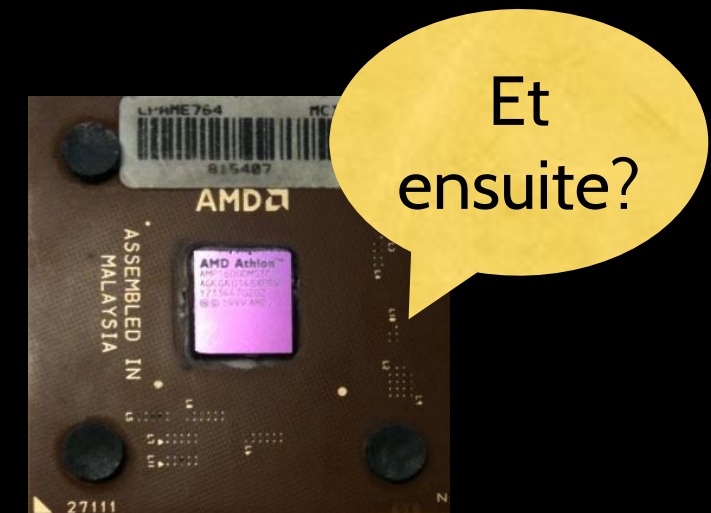


Python for Informatics: Exploring Information  
[www.pythonlearn.com](http://www.pythonlearn.com)



# Les ordinateurs veulent nous être utiles...

- Les ordinateurs sont construits dans un objectif - faire des choses pour nous
- Mais nous avons besoin de parler leur langue pour qu'ils puissent faire ce que nous voulons
- C'est facile pour les utilisateurs - quelqu'un a déjà intégré beaucoup de programmes différents dans l'ordinateur et les utilisateurs n'ont qu'à choisir les programmes qu'ils veulent faire marcher



# Les programmeurs anticipent les besoins

- Les applications pour iPhone sont un marché
- On compte plus de trois milliards de téléchargements d'applications pour iPhones
- Des programmeurs ont quitté leur poste pour devenir des développeurs d'iPhone à plein temps
- Les programmeurs connaissent **le fonctionnement du programme**



Choisi  
Moi!

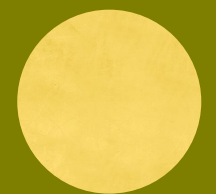
Choisi  
Moi!

Choisi  
Moi!

Choisi  
Moi!

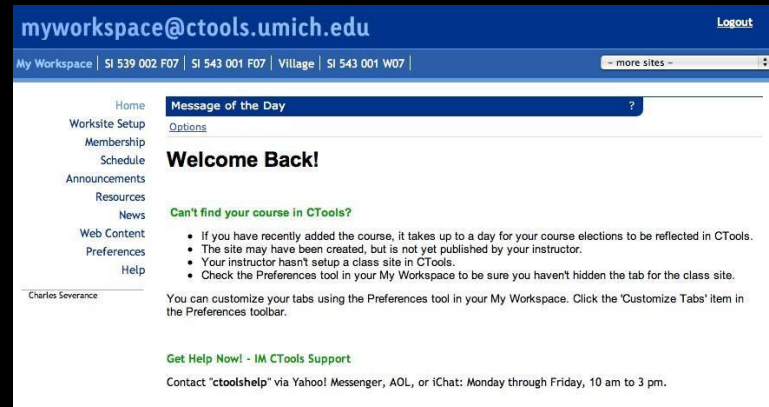
Choisi  
Moi!

Achète  
Moi!



# Les utilisateurs contre les Programmeurs

- Les utilisateurs voient les ordinateurs comme un ensemble d'outils - traitement de texte, tableur, carte, liste "à faire", etc.
- Les programmeurs apprennent le fonctionnement de l'ordinateur et son langage
- Les programmeurs disposent de certains outils qui leur permettent d'en créer de nouveaux
- Parfois les programmeurs écrivent des outils pouvant servir à plusieurs utilisateurs. D'autres fois, ils écrivent des petits "utilitaires", pour eux-même, qui leur permettent d'automatiser une tâche.



Utilisateur



Programmeur



....



Les programmeurs créent le logiciel. L'utilisateur final est le "maître" - qu'ils veulent satisfaire. Souvent celui-ci paye quand il est satisfait du logiciel. Mais les données, l'information, et les réseaux sont les problèmes auxquels sont confrontés les programmeurs. Ils ont pour alliés le matériel et le logiciel.

# Pourquoi être programmeur ?

- Pour que certaines tâches soient faites - nous sommes l'utilisateur et le programmeur
  - › Filtrer des données provenant de questionnaires
- Pour produire des informations à exploiter - c'est le travail du programmeur
  - › Résoudre un problème de performance dans le logiciel Sakai
  - › Ajouter un livre d'or à un site web

# Qu'est-ce que du code ? Un logiciel ? Un programme ?

- Une suite d'instructions stockées
  - › C'est une petite partie de notre intelligence dans l'ordinateur
  - › C'est une petite partie de notre intelligence que nous pouvons donner aux autres - nous résolvons quelque chose, nous l'encodons, puis nous le donnons à d'autres pour leurs épargner une perte de temps et d'énergie
- Un morceau d'art créatif - en particulier quand nous faisons du bon travail sur l'interface utilisateur (l'expérience utilisateur)



# Des programmes pour humains...



<http://www.youtube.com/watch?v=vlzwuFkn88U>  
<http://www.youtube.com/watch?v=sN62PAKoBfE>



# Des programmes pour humains...

Avec la musique:

Main gauche vers l'extérieur et en haut

Main droite vers l'extérieur et en haut

Retournez la main gauche

Retournez la main droite

Main gauche sur l'épaule droite

Main droite sur l'épaule gauche

Main gauche derrière la tête

Main droite derrière la tête

Main gauche sur la manche droite

Main droite sur la manche gauche

Main gauche sur fesse gauche

Main droite sur fesse droite

Se déhancher

Se déhancher

Sauter



<http://www.youtube.com/watch?v=vlzwwFkn88U>

<http://www.youtube.com/watch?v=sN62PAKoBfE>

# Des programmes pour humains...

Avec la musique:

Main gauche vers l'extérieur et en haut

Main droite vers l'extérieur et en haut

Retournez la main gauche

Retournez la main droite

Main gauche sur l'épaule droite

Main droite sur l'épaule gauche

Main gauche derrière la tête

Main droite derrière la tête

Main gauche sur la manche droite

Main droite sur la manche gauche

Main gauche sur fesse gauche

Main droite sur fesse droite

Se déhancher

Se déhancher

Sauter



<http://www.youtube.com/watch?v=vlzwwFkn88U>

<http://www.youtube.com/watch?v=sN62PAKoBfE>

# Des programmes pour humains...

Avec la musique:

Main gauche vers l'extérieur et en haut

Main droite vers l'extérieur et en haut

Retournez la main gauche

Retournez la main droite

Main gauche sur l'épaule droite

Main droite sur l'épaule gauche

Main gauche derrière la tête

Main droite derrière la tête

Main gauche sur la hanche droite

Main droite sur la hanche gauche

Main gauche sur fesse gauche

Main droite sur fesse droite

Se déhancher

Se déhancher

Sauter



<http://www.youtube.com/watch?v=vlzwwFkn88U>

<http://www.youtube.com/watch?v=sN62PAKoBfE>





le clown couru après la voiture et la voiture fonça sur la tente et  
la tente tomba sur le clown et la voiture

Programmes pour Python...



Programmes pour Python...

```
name = raw_input('Enter file:')
handle = open(name, 'r')
text = handle.read()
words = text.split()

counts = dict()
for word in words:
    counts[word] = counts.get(word,0) + 1
bigcount = None
bigword = None

for word,count in counts.items():
    if bigcount is None or count >
bigcount:
        bigword = word
        bigcount = count
print bigword, bigcount
```

**python words.py**  
**Fichier à utiliser: words.txt**  
**to 16**

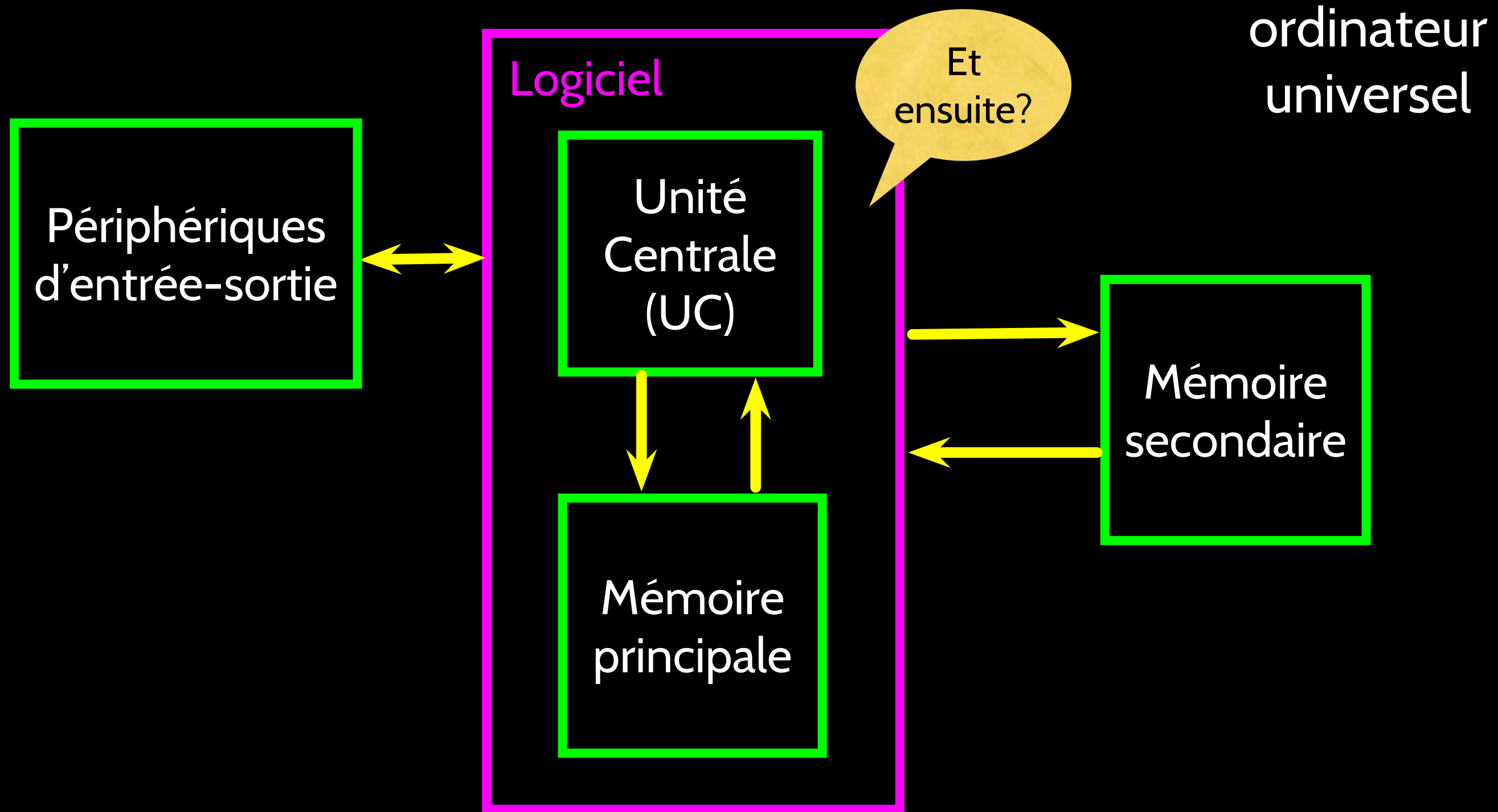
**python words.py**  
**Fichier à utiliser: clown.txt**  
**the 7**



# L'architecture matérielle

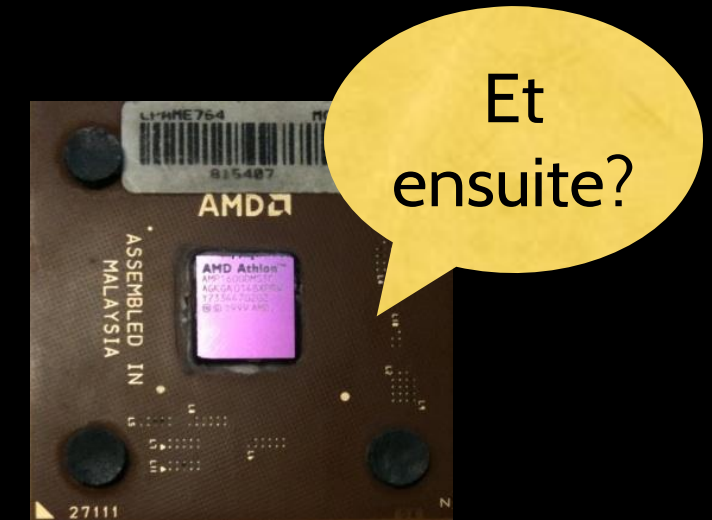


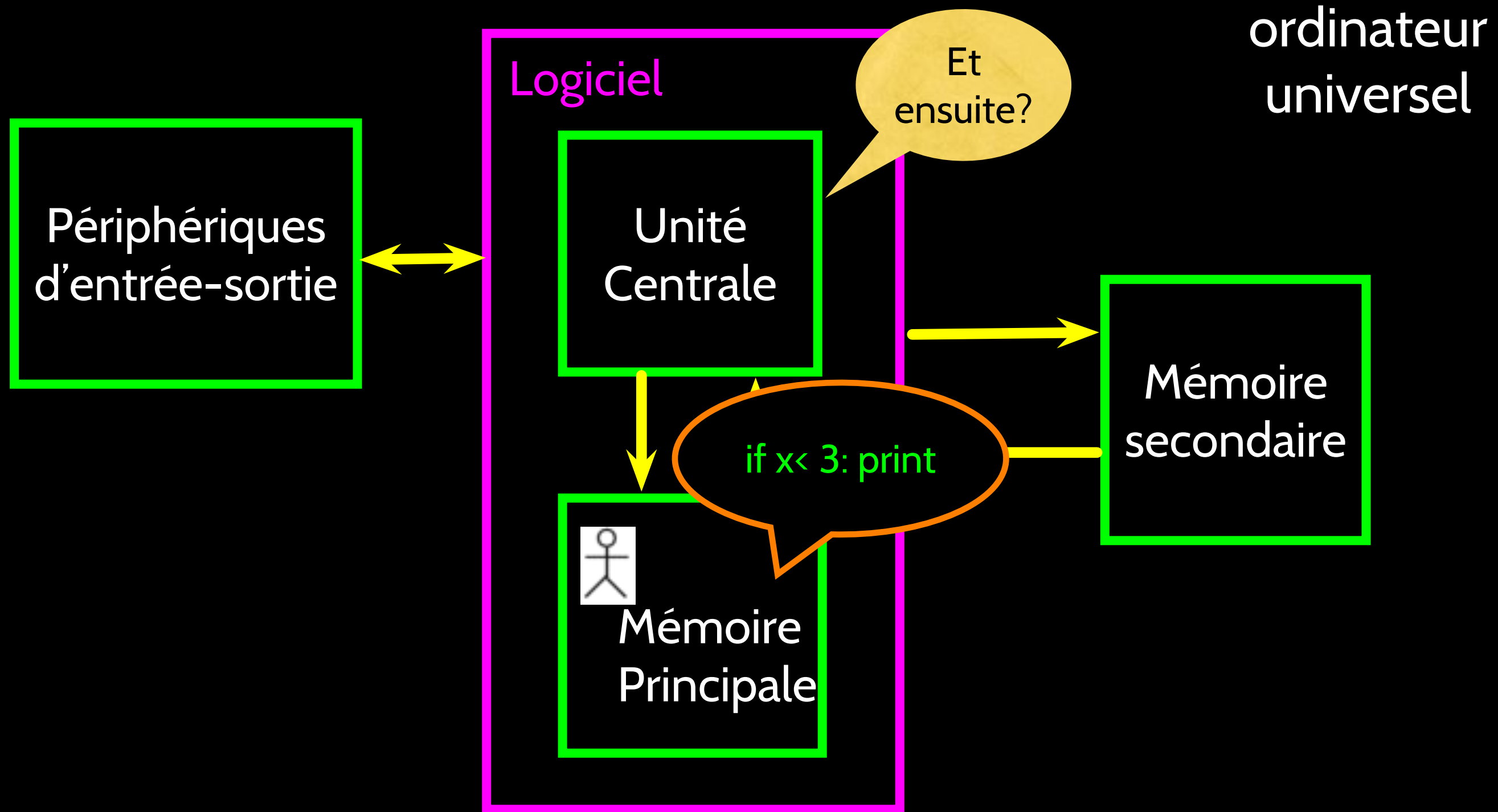
<http://upload.wikimedia.org/wikipedia/commons/3/3d/RaspberryPi.jpg>

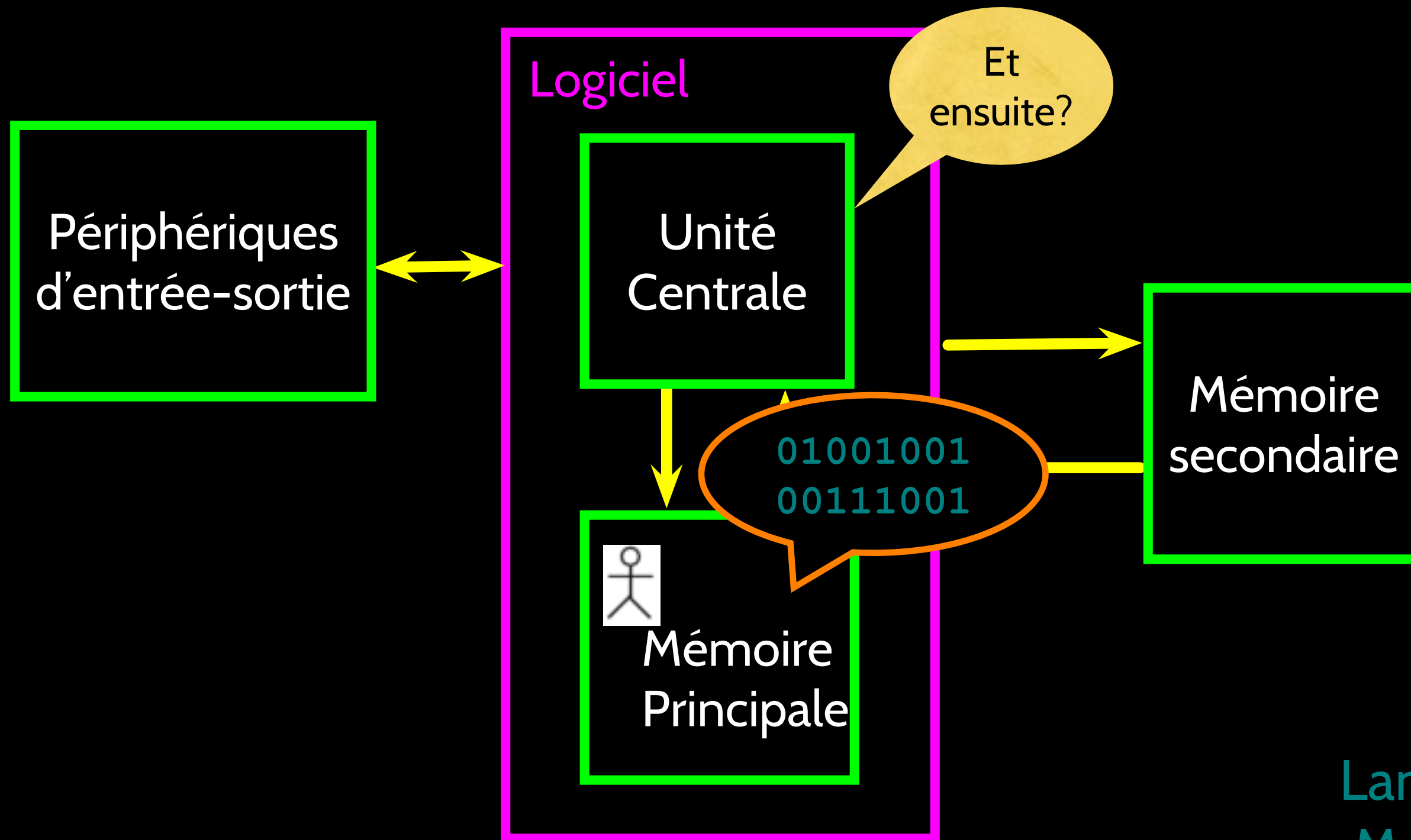


# Définitions

- **Unité centrale:** Exécute le programme - Le CPU demande toujours “Que dois-je faire ensuite”? Pas exactement comme un cerveau - plutôt stupide mais très très rapide
- **Périphériques d'entrée:** Clavier, Souris, Ecran tactile
- **Périphériques de sortie:** Ecran, Haut-parleurs, Imprimante, Graveur DVD
- **Mémoire principale:** Petite mémoire de stockage volatile et rapide - perdue au redémarrage (RAM)
- **Mémoire secondaire:** Grande mémoire de stockage permanente - dure jusqu'à ce qu'elle soit effacée - disque dur / carte mémoire



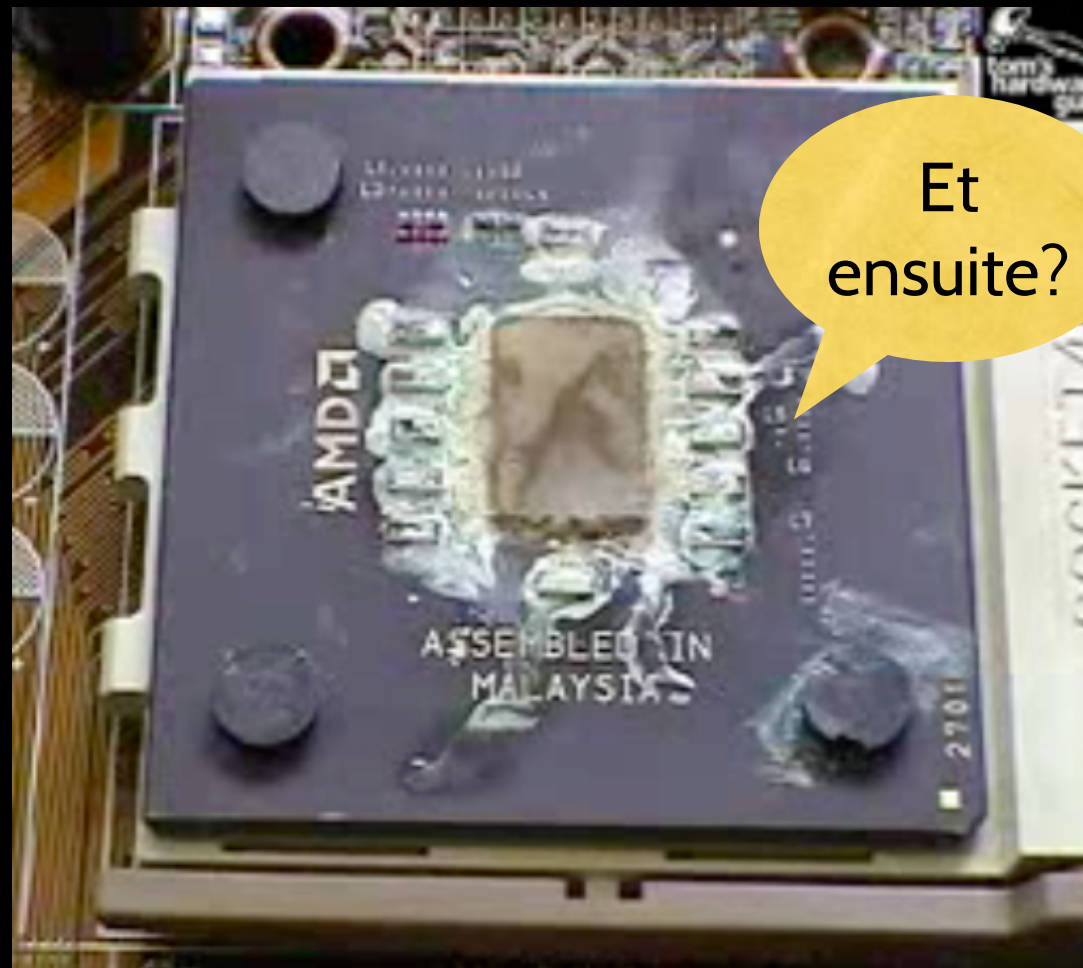




Language  
Machine

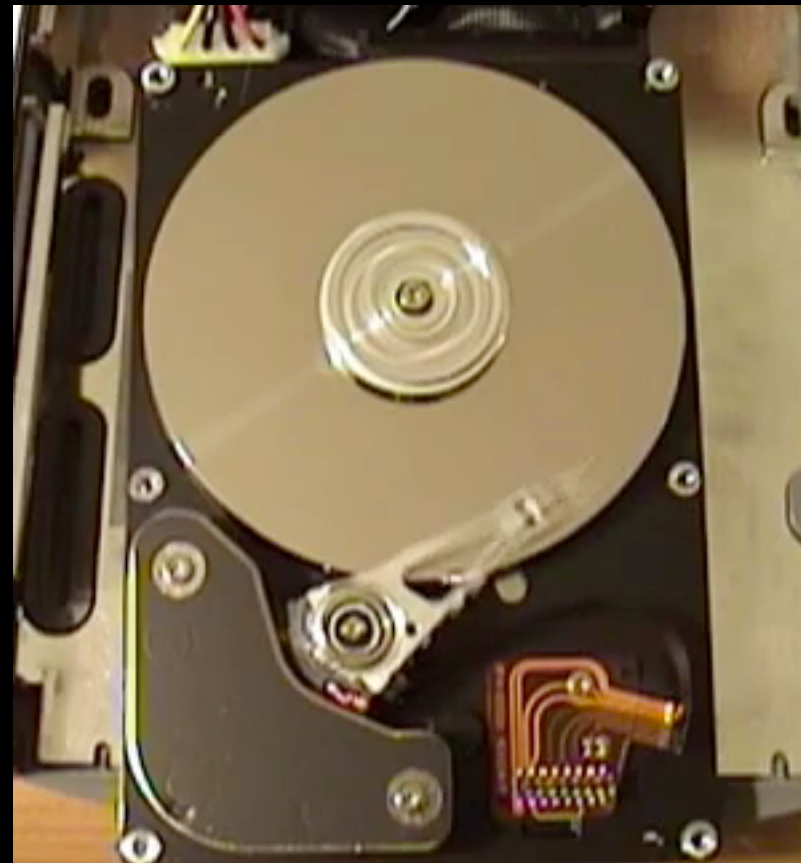


# Surchauffe de l' UC



<http://www.youtube.com/watch?v=y39D4529FM4>

# Disque dur en Action



<http://www.youtube.com/watch?v=9eMWG3fwiEU>

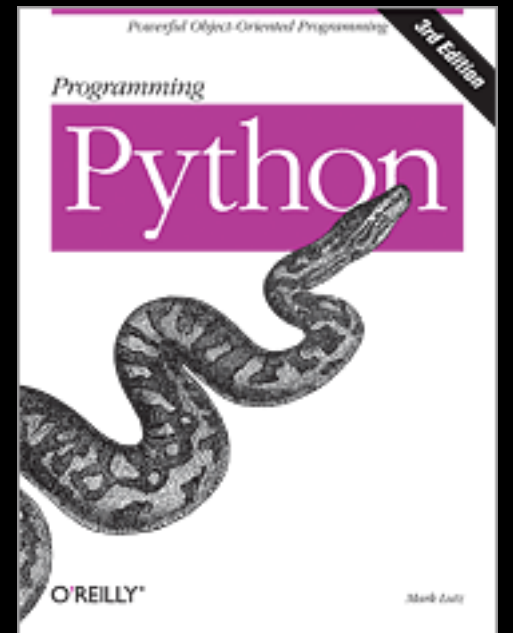
# Le langage Python

Le **Fourchelang** est le langage des serpents et de ceux qui peuvent parler avec eux. Une personne qui peut parler le **Fourchelang** est appelée un(e) **Fourchelang**. C'est une compétence très singulière, qui semble être héréditaire. Presque tous les **Fourchelangues** connus sont les descendants de Salazar Serpentard.



<http://harrypotter.wikia.com/wiki/Parseltongue>

Python est le langage de l'Interpréteur Python et de ceux qui peuvent parler avec lui. Une personne qui peut parler en Python est appelée un Pythoniste. C'est une compétence très singulière, qui semble être héréditaire. Presque tous les Pythonistes connus utilisent le logiciel développé initialement par Guido van Rossum.



# L'apprenant débutant: Syntax Errors

- Nous devons apprendre le **language Python** afin de pouvoir communiquer nos instructions à Python. Au début nous ferons beaucoup d'erreurs et nous parlerons le charabia comme les petits enfants.
- Quand vous faites une erreur, l'ordinateur ne pense pas que vous êtes "futé". Il dit "**syntax error**" - affirmant qu'il *\*connait\** le langage et que vous avez juste à l'apprendre. C'est comme si Python était cruel et insensible.
- Vous devez vous rappeler que *\*vous\** êtes intelligent et que vous *\*pouvez\** apprendre - l'ordinateur est simplement très rapide - mais il ne peut apprendre - donc il **est plus facile pour vous d'apprendre le Python que pour l'ordinateur d'apprendre le Français...**



Parler à Python

```
csev$ python
```

```
Python 2.5 (r25:51918, Sep 19 2006, 08:49:13)
```

```
[GCC 4.0.1 (Apple Computer, Inc. build 5341)] on darwin
```

```
Type "help", "copyright", "credits" or "license" for more  
information.
```

```
>>>
```

← Et ensuite?

```
csev$ python
```

```
Python 2.5 (r25:51918, Sep 19 2006, 08:49:13)
```

```
[GCC 4.0.1 (Apple Computer, Inc. build 5341)] on darwin
```

```
Type "help", "copyright", "credits" or "license" for more  
information.
```

```
>>> x = 1
```

```
>>> print x
```

```
1
```

```
>>> x = x + 1
```

```
>>> print x
```

```
2
```

```
>>> exit()
```

Ceci est un bon test afin de s'assurer que vous  
avez correctement installé Python.

Remarquez qu'il est aussi possible de taper  
quit() pour terminer la session interactive.afi

# Allons parler à Python...

```
Default
dr-chuck2:~ csev$ python
Python 2.6.1 (r261:67515, Jun 24 2010, 21:47:49)
[GCC 4.2.1 (Apple Inc. build 5646)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print "hello world"
hello world
>>> █
```

```
Administrator: C:\Windows\system32\cmd.exe - C:\Python27\python.exe
Microsoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>C:\Python27\python.exe
Python 2.7.2 (default, Jun 12 2011, 15:08:59) [MSC v.1500 32 bit (Intel)] on win
32
Type "help", "copyright", "credits" or "license" for more information.
>>> print "hello world"
hello world
>>> _
```

Que disons-nous?

# Éléments de Python

- **Vocabulaire / Mots** - Variables et mots réservés (Chapitre 2)
- **Structure de phrase** - Modèles de syntaxe valide (Chapitres 3-5)
- **Structure de l'histoire** - Programmer dans un but



```
name = raw_input('Enter file:')
handle = open(name, 'r')
text = handle.read()
words = text.split()

counts = dict()
for word in words:
    counts[word] = counts.get(word,0) + 1
bigcount = None
bigword = None

for word,count in counts.items():
    if bigcount is None or count > bigcount:
        bigword = word
        bigcount = count
print bigword, bigcount
```

Une “nouvelle” ayant  
pour propos le  
comptage des mots  
dans un fichier Python

**python words.py**  
**Fichier à utiliser: words.txt**  
**to 16**

# Mots réservés

- Vous ne pouvez pas utiliser des **mots réservés** en tant que noms de variables / identificateurs

and del for is raise assert elif from  
lambda return break else global  
not try class except if or while  
continue exec import pass yield  
def finally in print as with

# Phrases ou Lignes

x	=	2	←	Instruction d'assignation		
x	=	x	+	2	←	Assignation d'une expression
print	x	←	Instruction Print (sur écran)			

Variable

Opérateur

Constante

Mot réservé

Programmation de paragraphes

# Scripts Python

- Python en mode interactif est pratique pour expérimenter avec des programmes longs de 3-4 lignes
- La plupart des programmes sont beaucoup plus longs, c'est pourquoi nous les écrivons dans un fichier afin que python exécute les commandes à partir de ce fichier
- D'une certaine manière nous "donnons à Python un script"
- Comme convention, nous ajoutons le suffixe ".py" à la fin de ces fichiers pour indiquer qu'ils contiennent des instructions Python

Ecrire un programme simple

# Interactif contre script

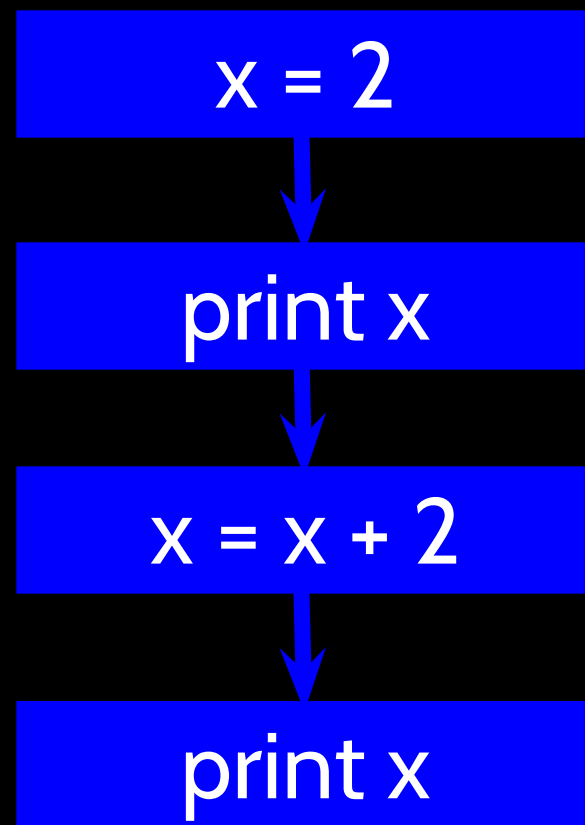
- Interactif
  - › Vous tapez directement dans Python une ligne après l'autre et vous obtenez la réponse
- Script
  - › Vous écrivez une suite d'instructions (lignes) dans un fichier en vous servant d'un éditeur de texte et dites à Python d'exécuter les instructions du fichier

# Etape de programme ou Flux de programme

- Tout comme une recette ou des instructions d'installation, un programme est une séquence de pas à faire dans l'ordre
- Certaines étapes sont conditionnelles - elles peuvent être "sautées"
- Parfois une étape ou un groupe d'étapes doivent être répétés
- Parfois nous stockons un ensemble d'étapes qui doivent être réitérées au besoin à différents endroits du programme (Chapitre 4)



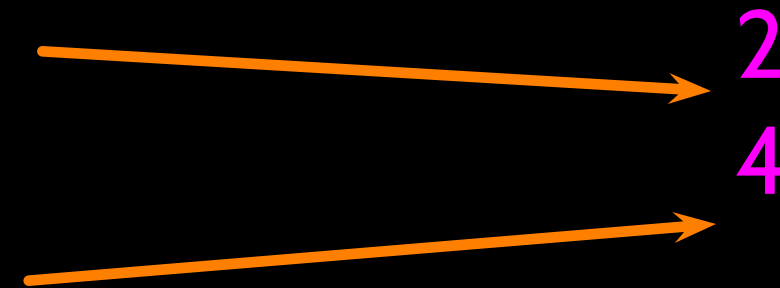
# Etapes Sequentielles



Programme:

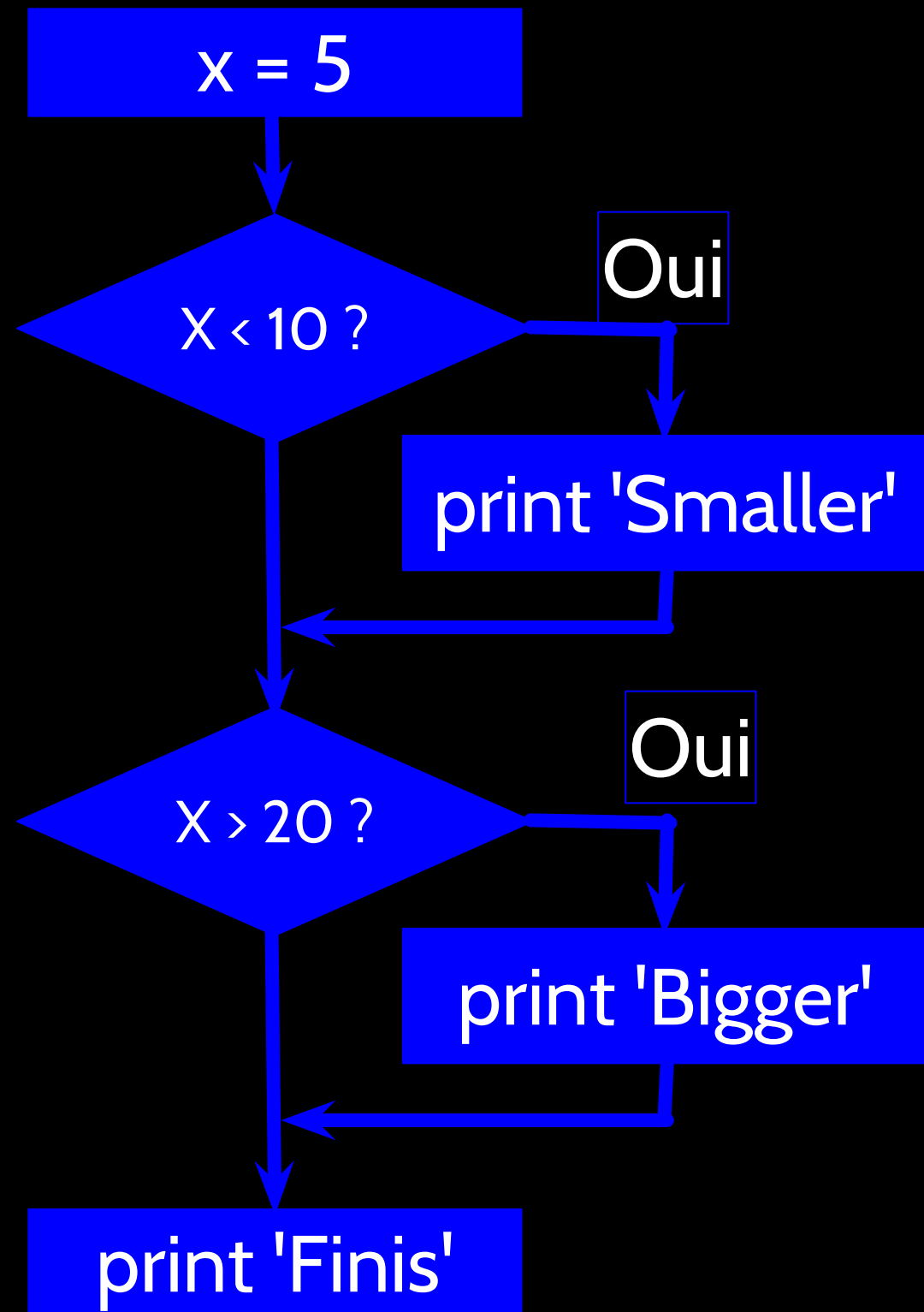
```
x = 2  
print x  
x = x + 2  
print x
```

Sortie:



Quand un programme s'exécute, il passe d'une étape à la suivante. En tant que programmeurs, nous mettons en place des "chemins" que le programme doit suivre.

# Etapes Conditionnelles



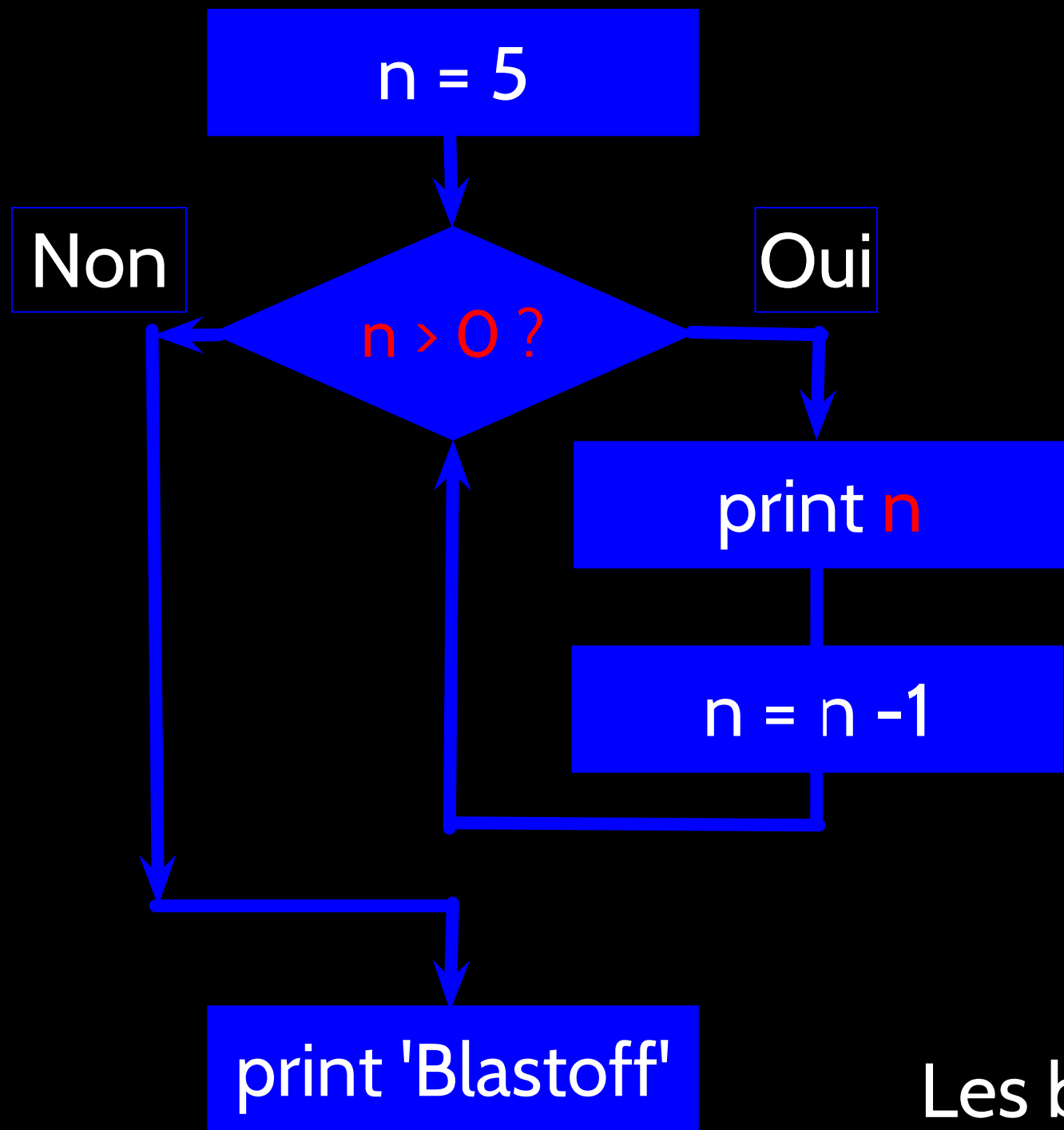
Programme:

```
x = 5
if x < 10:
    print 'Smaller'
if x > 20:
    print 'Bigger'
print 'Finis'
```

Sortie:

Smaller  
Finis

# Étapes Répétées



Programme:

```
n = 5
while n > 0:
    print n
    n = n - 1
print 'Blastoff!'
```

Sortie:

5  
4  
3  
2  
1  
Blastoff!

Les boucles (étapes répétées) ont des **variables d'itération** qui changent à chaque cycle de boucle. Souvent ces **variables d'itération** suivent une séquence de nombres.

# Résumé

- Ceci est un rapide aperçu du Chapitre 1
- Nous reverrons ces concepts dans le cours
- Pour l'instant concentrez-vous sur la vue d'ensemble et non sur les détails



# Remerciements / Contributions



These slides are Copyright 2010- Charles R. Severance ([www.dr-chuck.com](http://www.dr-chuck.com)) of the University of Michigan School of Information and [open.umich.edu](http://open.umich.edu) and made available under a Creative Commons Attribution 4.0 License. Please maintain this last slide in all copies of the document to comply with the attribution requirements of the license. If you make a change, feel free to add your name and organization to the list of contributors on this page as you republish the materials.

Initial Development: Charles Severance, University of Michigan School Of Information

Translation: Frederic Foiry  
Revision : Stephanie Kamidian

...