

OFICINA DE PROGRAMAÇÃO EM PYTHON

Cronograma

Introdução

Apresentação da oficina

O que é programação?

O que é Python?

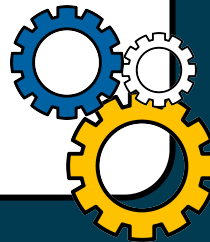
Fundamentos de programação

Funções

Variáveis e tipos de dados

Operadores

Condicionais



Python

Como instalar?

IDEs

Algoritmos

Como escrever e executar programas?

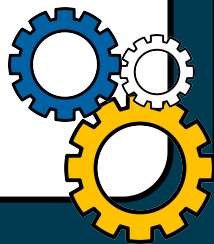
Exemplo prático



Apresentação

Wiki Movimento Brasil

O Wiki Movimento Brasil (WMB) é uma associação brasileira sem fins lucrativos afiliada à Fundação Wikimedia. Fundado em 2013, o WMB busca, entre outros objetivos: ampliar, qualificar e diversificar o conteúdo e a comunidade nos projetos da Wikimedia, em especial a Wikipédia; e apoiar a atuação de organizações sociais no ecossistema do conhecimento livre.



Contexto desse espaço

Em 2023, o WMB lançou publicamente sua estratégia para o triênio de 2023 a 2025. Dentro da estratégia, temos 5 eixos norteadores, cada um com uma intencionalidade e direcionamentos para ações.

Esta oficina está sendo orientada principalmente pelo eixo **(Re)imaginar a infraestrutura sociotécnica do Movimento Wikimedia**, especificamente o direcionamento Fortalecer uma cultura de documentação e compartilhamento de processos e aprendizados, com a criação de espaços formativos

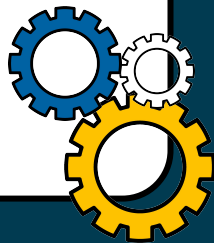
Apresentação

Éder Porto

Eu sou Gestor de Produtos e Tecnologia na equipe profissional do WMB, e sou responsável por desenvolver novas tecnologias e otimizar processos dentro da infraestrutura das atividades e programas que o grupo conduz.

Sou matemático aplicado, designer gráfico e desenvolvedor, baseado em São Paulo.

Meu projeto favorito é o Wikidata. ❤️



A Oficina de programação em Python

Contexto

Este curso foi criado como uma iniciativa de capacitação da comunidade Wikimedista em programação em Python. Durante as oficinas, os participantes serão apresentados aos conceitos básicos de programação e aprenderão como aplicá-los na linguagem Python, além da integração com os projetos Wikimedia. Os exemplos e exercícios práticos que os participantes irão realizar para consolidar seus aprendizados serão voltados aos seus interesses dentro dos projetos Wikimedia que atuam.

O que podemos esperar do curso?

Espera-se que os participantes estejam motivados a aprender e dediquem-se a acompanhar o ritmo do curso, tirando dúvidas, assistindo às oficinas e participando das atividades e realizando os exercícios e projetos propostos.

Da nossa parte, espera-se que os materiais sejam claros e acessíveis, que os exercícios sejam relevantes aos contextos dos participantes e que sejam desafiadores e que estejamos disponíveis para tirar dúvidas e fornecer feedback sempre que necessário.

A Oficina de programação em Python

Dinâmica

As oficinas serão espaços de apresentação teórica e prática. **Só se aprende fazendo!** Dúvidas são mais que bem-vindas, são incentivadas!

Após as oficinas, vocês receberão orientações para exercícios práticos que deverão ser executados no período entre oficinas.

Um mês após a realização da oficina, eu abrirei uma chamada para resolução de dúvidas sobre as atividades para quem as tiver. Mas podem mandar mensagens por e-mail ou Slack com dúvidas também.

Certificados

Os certificados serão emitidos em dezembro, imediatamente após a última oficina e contabilizarão a quantidade de horas do curso.

Caso não consigam mais participar das oficinas, pedimos que enviem uma mensagem avisando, pois temos uma lista de espera.

Materiais

Todos os materiais estarão presentes na página da oficina: <https://w.wiki/6cDn>

Rodada de apresentação

(15 min)

Questionário de expectativas

Expectativas

O que poderemos fazer depois do curso?

Espera-se que os participantes possam aplicar os conhecimentos adquiridos durante as oficinas para criar programas em Python para facilitar suas atividades Wiki. Isso inclui a criação de scripts para automação de tarefas, manipulação de dados para análise e a criação de interfaces gráficas de usuário. Além disso, espera-se que os participantes sejam capazes de contribuir com a comunidade Wikimedia, compartilhando seus conhecimentos e aprendizados.



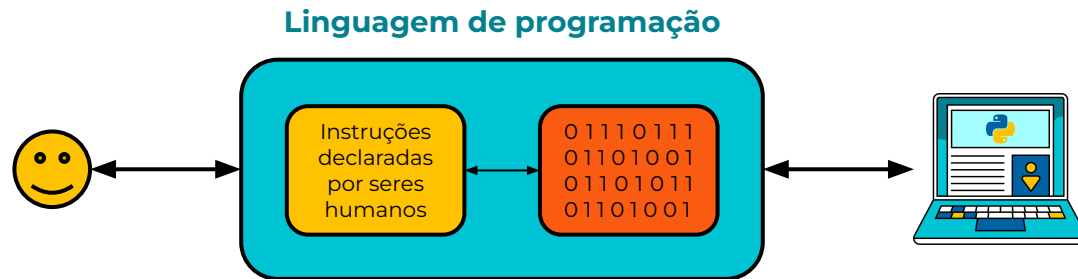
Definições

Definições

O que é programação?

Programação é a criação de um conjunto de comandos para um computador, de modo que ele execute uma tarefa específica.

Porém, os computadores são máquinas que se comunicam em números binários (0 e 1). Para nos comunicarmos de maneira mais fluida, utilizamos as linguagens de programação, que facilitam a integração humano-máquina e máquina-humano.



Definições

O que é Python?

Python é uma linguagem de programação de alto nível, interpretada e orientada a objetos. Possui uma sintaxe simples e é fácil de aprender. É utilizada para uma grande variedade de aplicativos e finalidades.



Alto nível = Mais próxima da linguagem natural humana

Baixo nível = Mais próxima da linguagem de máquina

Interpretada = Não é necessário um “tradutor”, o código é traduzido conforme é executado.

Compilada = É necessário um “tradutor” antes de executar o programa.

Orientada a objetos = modelo que organiza o código em torno de entidades com propriedades e comportamentos específicos.

Estruturada = modelo que organiza o código em estruturas mais simples, como sequências, decisões e iterações.

Programação na Wiki

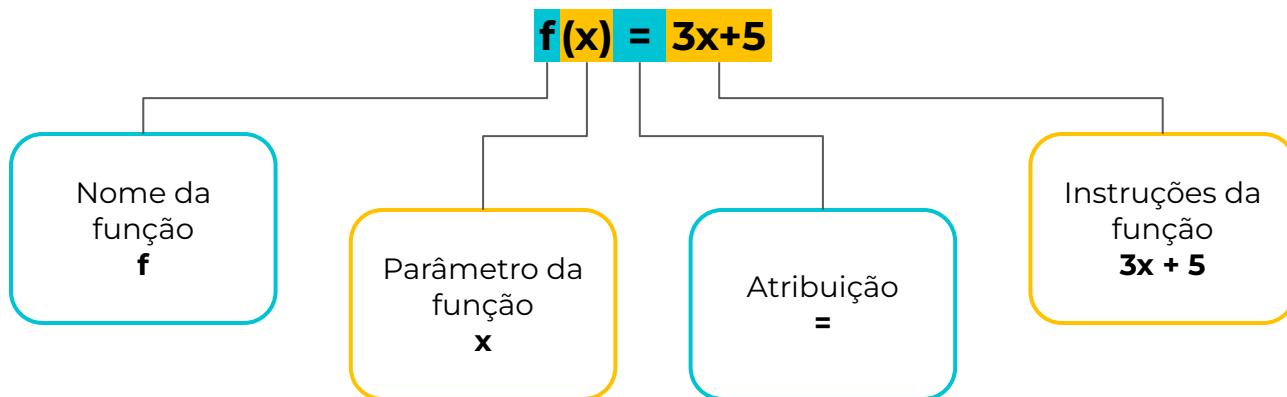
```
N I ↻ ↺ ↻ ↻ ↻ Edição avançada > Caracteres especiais > Ajuda
Cabeçalho ▾ Formatar ≡ ≡ ≡ ↶ A^ A^ X^ X^ Inserir ↗ ↘ ↗
1 <includeonly>{{Info
2 <!-- Identificação -->
3 | título           = {{{nome|{{PAGENAME}}}} {{#if:{{{nome_nativo}}}|<br />{{nom
4 | subtítulo       = {{{subtítulo|}}}
5 | imagem          = {{{imagem|{{{foto|{{{imagem|}}}}}}}}
6 | imagem_tamanho = {{{imagem_tamanho|{{{tamanho|200px}}}}}}
7 | legenda        = {{{legenda|{{{imagem_legenda|{{{desc|}}}}}}}}
8 <!-- Estilos CSS -->
9 | estilo          = width: 270px; font-size: 88%;
10 | cabeçalho-classe = {{{cabeçalho-classe|}}}
11 | cabeçalho-estilo = {{{cabeçalho-estilo|}}}
12 | título-classe   = {{{título-classe|}}}
13 | título-estilo   = text-align:center; font-size:larger; background-color:{{{bgco
{{{título-estilo|}}}
14 | tópico-estilo   = {{{tópico-estilo|}}}
15 | rótulo-estilo   = font-weight:bold;
16 | rodapé-estilo   = padding:4px 0 4px 4px;
17 | classe         = infobox_v2
18 <!-- Informações -->
19 <!-------Básico: Pessoal----->
20 | rótulo1         = Nome completo
21 | dados1          = {{{nome_completo|{{{nome_de_batismo|}}}}}}
22 | rótulo2         = [[Pseudônimo|Pseud{{{Info/lang| {{{lang|}}} |ô|ó| ptx={{{pseudóni
{{{nomes artísticos|}}} }}nimo(s)}}
23 | dados2          = {{{Info/lang| {{{lang|}}} | {{{pseudônimo|}}}{{{nome_artístico|}'
```

```
<> ≡ ≡ ≡ ↶ ↷ ↶ ↷ ↶ ↷
391 -- data type is a wikibase item:
392 if props[1].mainsnak.datatype == "wikibase-item" then
393 -- it's wiki-linked value, so output as link if possible
394 for k, v in pairs(props) do
395 if filter(v) and ((onlysrc == false) or sourced(v)) then -- has valid re
396 if v.mainsnak.snaktype == "somevalue" then -- check for value is unk
out[#out + 1] = "Desconhecido"
397 elseif v.mainsnak.snaktype == "novalue" then -- check for value is n
-- out[#out + 1] = "No value" -- don't return a value for this
398 else
399 local qnumber = "Q" .. v.mainsnak.datavalue.value["numeric-id"]
400 local sitelink = mw.wikibase.sitelink(qnumber)
401 local label = labelOrId(qnumber, frame.args.idioma or "pt-br")
402
403
404
405 local gender = frame.args.gender or ""
406 if gender ~= "" and entity.claims["P21"] then
407 local genderID = entity.claims["P21"][1].mainsnak.datavalue.
408 if genderID == "F" or genderID == "Q6581072" or genderID ==
409 local genderEntity = mw.wikibase.getEntityObject(qnumber
410 if genderEntity.claims["P2521"] then
411 for k2,v2 in pairs(genderEntity.claims["P2521"]) do
412 if filter(v2) and (v2.mainsnak.datavalue.value["
413 label = v2.mainsnak.datavalue.value["text"]
414 end
415 end
416 end
417 end
418 end
419
```

Fundamentos de programação

Funções

Funções são blocos de código que executam uma tarefa específica e podem ser chamadas (ou invocadas) em outras partes do programa. Geralmente recebem valores de entrada (chamados de *parâmetros*) e retornam valores de saída. Exemplo:



Funções

Em Python, escrevemos funções de forma semelhante.

def f(x):

resultado = 3*x + 5

return resultado

def: indicador de definição de função

return: comando de retorno do resultado da função

Variáveis em Python não precisam ser definidas previamente.

O código em python é escrito de forma indentada. O início de um bloco é marcado pela presença dos dois pontos ao final da linha.

O que acontece se x não for um número?

Funções

Existem diversos tipos de entradas e saídas. Esses valores são chamados de **variáveis**.

Cadeia de caracteres (strings), Números inteiros (int) e com casas decimais (float), valores lógicos (booleanos), listas (list), dicionários (dict) etc.

Operadores

Para fazer operações com variáveis, é necessário conhecer os **operadores matemáticos** básicos além de outros:

Multiplificação: *	3 * 5	“três vezes cinco”
Divisão: /	6 / 4	“seis dividido por quatro”
Soma: +	5 + 7	“cinco mais sete”
Subtração: -	8 - 6	“oito menos seis”
Exponenciação: **	7 ** 9	“sete elevado a nove”

Operadores

def f(x):

resultado = 3*x + 5

return resultado

Para $x = 3$, $f(x)$ retorna o valor 14 (int)

Para $x = 2.5$ $f(x)$ retorna o valor 12.5 (float)

Operadores

Operadores de comparação

Igualdade: ==

“texto” == “Texto”

“texto é igual a Texto?”

Diferença: !=

“texto” != “Texto”

“texto é diferente de Texto?”

Maior que: >

5 > 7

“cinco é maior que sete?”

Maior ou igual que: >=

8 >= 6

“oito é maior ou igual a seis?”

Menor que: <

7 < 9

“sete é menor que nove?”

Menor ou igual que: <=

7 <= 9

“sete é menor ou igual a nove?”

Operadores

def f(x):

resultado = 3*x + 5

comparacao = resultado > 13

return comparacao

Para $x = 3$, $f(x)$ retorna o valor True (bool)

Para $x = 2.5$ $f(x)$ retorna o valor False (bool)

Operadores

Operadores lógicos

E: **and**

“texto” == “texto” and 3 < 5

“texto é igual a texto e

3 é menor que 5?”

Ou: **or**

6 != 4 or 6 < 9

“seis é diferente de quatro ou

seis é menor que nove?”

Não: **not**

not 5 > 7

“Não ‘cinco é maior que sete?’”

Operadores

```
def e_maior_ou_menor(x):
```

```
    resultado = 3*x + 5
```

```
    se not resultado:
```

```
        return "Deu erro"
```

```
    return resposta
```

Para $x = 3$, $f(x)$ retorna o valor True (bool)

Para $x = 2.5$ $f(x)$ retorna o valor False (bool)

Operadores

Operadores de atribuição

Atribuição combinada de soma: +=

x += 6 “xis é igual a xis mais seis”

Atribuição combinada de subtração: -=

x -= 3 “xis é igual a xis menos três”

Atribuição combinada de multiplicação: *=

x *=2 “xis é igual a xis vezes dois”

Atribuição combinada de divisão: /=

x /=2 “xis é igual a xis sobre dois”

Operadores

def f(x):

resultado = 3*x + 5

resultado += 2

return resultado

Para $x = 3$, $f(x)$ retorna o valor 17 (int)

Para $x = 2.5$ $f(x)$ retorna o valor 14.5 (float)

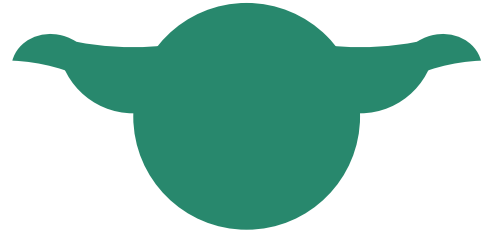
Dúvidas?

Se e caso contrário

if e **else** são instruções condicionais que permitem que o programa execute diferentes blocos de código a depender do resultado de uma expressão booleana (*verdadeiro* ou *falso*).

Lembremos que computadores só fazem o que pedimos que façam e não tomam decisões por si próprios.

A programação exige, portanto, o desenvolvimento de uma habilidade de previdência dos cenários possíveis.



“Computadores decisões próprias não têm, lógica e instruções humanas seguir eles devem”

Python

Python: Instalação e utilização

Existem vários jeitos de interagir com a linguagem Python.

Download direto (utilização pelo Terminal): <https://www.python.org/downloads>

Ambientes de Desenvolvimento Integrado (IDEs):

- Pycharm: <https://www.jetbrains.com/pt-br/pycharm>
- Jupyter: <https://jupyter.org/install>
- Spyder: <https://docs.spyder-ide.org/current/installation.html>
- Visual Studio Code: <https://code.visualstudio.com/download>

Ambiente de desenvolvimento baseado na nuvem:

- Hub-PAWs: <https://hub-paws.wmcloud.org>



Vamos programar!

Exemplo: Pedra, Papel, Tesoura!

Brincadeira clássica, mas como funciona?

Dois jogadores esticam ***simultaneamente*** as mãos, na qual cada um formou um símbolo representando uma pedra, um papel ou uma tesoura, respectivamente 🖐️, 🖐️ e ✌️.

Regras:

- Pedra ganha de tesoura;
- Tesoura ganha de papel;
- Papel ganha de pedra.

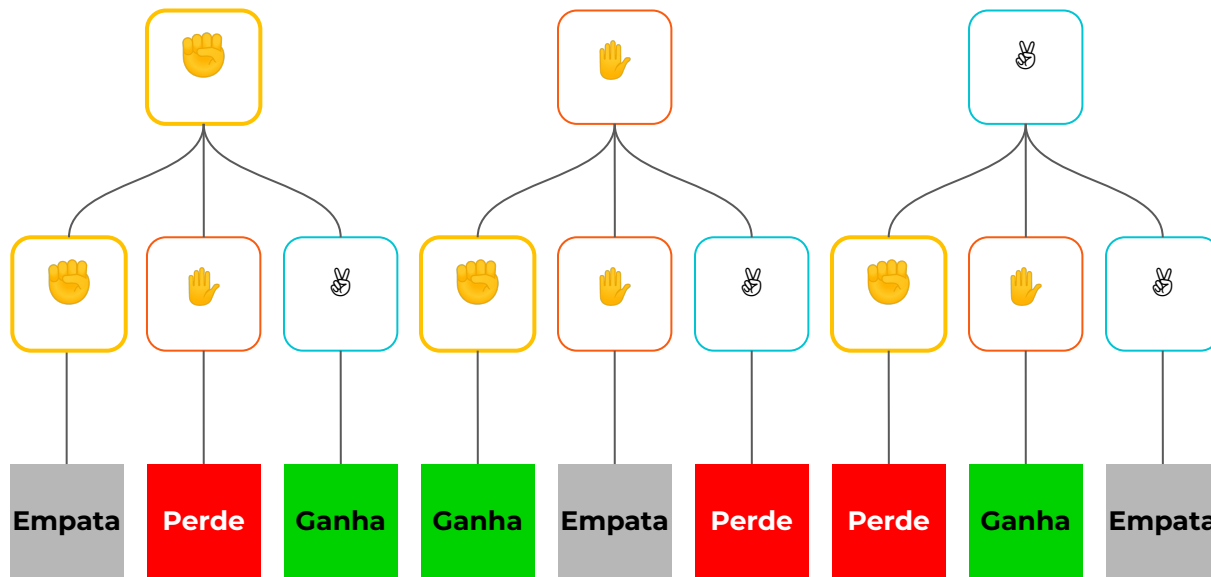
Como você transformaria esta brincadeira em um algoritmo para determinar o vencedor?

Exemplo: Pedra, Papel, Tesoura!

Suas escolhas possíveis

Escolhas possíveis do computador

Resultados para você



Exemplo: Pedra, Papel, Tesoura!

1. Observe o símbolo escolhido pelo usuário;
2. Observe o símbolo escolhido pelo computador;
3. Se ambos os símbolos forem iguais, diga que houve empate;
4. Se o símbolo do usuário for pedra e o símbolo do computador for tesoura, diga que o usuário ganhou;
5. Se o símbolo do usuário for pedra e o símbolo do computador for papel, diga que o usuário perdeu;
6. Se o símbolo do usuário for tesoura e o símbolo do computador for papel, diga que o usuário ganhou;
7. Se o símbolo do usuário for tesoura e o símbolo do computador for pedra, diga que o usuário perdeu;
8. Se o símbolo do usuário for papel e o símbolo do computador for pedra, diga que o usuário ganhou;
9. Se o símbolo do usuário for papel e o símbolo do computador for tesoura, diga que o usuário perdeu;
10. Se alguém que alguém não entendeu a brincadeira esticar a mão com um “joinha”, por exemplo, ao invés das opções predeterminadas, diga que a opção é inválida.

[Vamos implementar isso?!](#)

Éder Porto
Ederporto | EPorto (WMB)
eder.porto@wmnobrasil.org

Grupo de usuários
WMB
Wiki Movimento Brasil