

Olá, tudo bem?

Esperamos que sim! E estamos aqui para deixar ainda melhor.

Nesse material vamos introduzir você a uma das linguagens de programação mais usada no mundo: Python.

Começando...

Você já tem alguma experiência com programação?

Se sim, esse material vai ser uma ótima introdução em Python para você. Mas se você apenas está se aventurando querendo aprender uma linguagem nova, não se preocupe, Python é fácil, te daremos o necessário para começar.



O que é Python?



Python é uma linguagem de programação criada por Guido van Rossum em 1991. Os objetivos do projeto da linguagem eram: produtividade e legibilidade.

A programação em Python pode ser usada para programas simples e rápidos, e também para estruturas de dados complexas, como tuplas, listas e dicionários, estão disponíveis para facilitar o desenvolvimento de algoritmos complexos.

Grandes projetos podem ser feitos usando técnicas de orientação a objetos, que é completamente suportada em Python. Um suporte modesto para programação funcional existe, o que torna a linguagem extremamente expressiva: é fácil fazer muita coisa com poucas linhas de comando.

Instalação do Python

Nada melhor do que aprender praticando.

Então, vamos dar uma pausa nas definições e conceitos formais e vamos botar a mão na massa!

Passo 1: Fazer o download do Python

Primeiramente, precisamos baixar o ambiente de desenvolvimento integrado do Python, chamado de IDLE. Esse é local onde escreveremos nosso programa.

Para isso, basta acessar a página oficial da *Python Software Foundation*, ir até a aba “Downloads” e clicar em Python.

Vamos levantar duas vantagens: Primeira, ele é de licença de uso público, ou seja, é gratuito. E segunda, é multiplataforma, portanto, funciona em vários sistemas operacionais diferentes e até mesmo em aparelhos móveis!



Passo 2: Instalando o software

Abra o arquivo de instalação baixado. Em seguida, clique em “Install Now” e aguarde a mágica acontecer!

Passo 3: Conhecendo o ambiente de desenvolvimento (IDLE)

Ao abrir o aplicativo IDLE, nos damos de cara com o Python Shell. Esse é o principal local de interação com o usuário, é aqui que os resultados dos códigos criados são exibidos.

Uma dica muito útil:

- Para maior compreensão e melhor visualização, recomendamos que você utilize o software **PyCharm**, ele é uma mão na roda para quem não tem muita prática com o mundo da programação.

Meu primeiro código em Python.

Agora que temos tudo pronto, podemos criar alguns códigos para que você possa entender melhor como funciona esse software.

Primeiramente, vamos exibir uma mensagem na tela. Geralmente, esse é o primeiro exercício que se faz no ensino de qualquer linguagem de programação. Para isso, vamos usar a função `print`. Abra um novo arquivo (*File > New File*) e digite o seguinte comando:

```
Print('Hello world')  
-- Hello world
```

1. Ai vai aparecer na sua tela algo impressionante: "Hello World" (sem as aspas!)
2. Vamos explicar o código:
 - O caractere '#' define um comentário, tudo que vem depois dele na linha não é interpretado
 - Print "Hello World" (print é um comando padrão do Python que imprime na tela os argumentos passados.)

Obs: O comando print também aceita números, variáveis, expressões, etc...
Exemplos:

Esconder número das linhas:

```
1 # print expressão  
2 print 5+5
```

Esconder número das linhas:

```
1 # print com múltiplos argumentos  
2 print "5+5=",5+5 # Uma string em python é tudo entre "" (aspas duplas)  
3 # ou " (aspas simples).
```

Esconder número das linhas:

```
1 # print com variáveis  
2 minhavariavel = "Essa é a minha variável" # declaração de variável  
3 #  
4 # Falaremos sobre isso  
5 # no próximo artigo...  
6 print minhavariavel
```

Esconder número das linhas:

```
1 # concatenação no print
2 print "Eu "+"Te "+" Amo" # Concatenamos strings através do operador
    '+'
3 print "Eu","Te","Amo" # Mesma coisa que a linha anterior, porém aqui
4                        # elas só se agrupam na saída.
5                        #
6                        # obs.: o espaçamento é automático.
```

Função de impressão – print()

A função Python print () aceita qualquer número de parâmetros e os imprime em uma linha de texto. Cada um dos itens é convertido para a forma de texto, separados por espaços, e há um único '\n' no final (o caractere de "nova linha"). Quando chamado com zero parâmetros, print () apenas imprime o '\n' e nada mais. No interpretador, a saída padrão é exibida na tela entre os prompts '>>>', então é uma maneira fácil de ver o que print () faz.

```
>>> print(12, 24, -2)
12 24 -2
>>> print ('oi', 'ai', -2)
oi ai -2
>>> print ('mundo') # 1 item, 1 \ n
mundo
>>> print ()      # 0 itens, 1 \ n
```



Variáveis e tipos de dados em Python

Recebendo Dados de usuário – input()

```
nome = input('Qual o seu nome?: ')
print(f'Seja bem-vindo(a) {nome}')
idade = int(input('Qual a sua idade?: '))
print(f'{nome} nasceu em {idade - 2020}')
```

A função `print()` facilita o envio de seus dados como texto para saída padrão. A função `input()` vai na outra direção, permitindo que o usuário digite algo que vai em uma variável Python.

Aqui está um exemplo de `input()` no intérprete para ver como funciona:

```
>>> nome = input('Qual é o seu nome?')
Qual é o seu nome? Guto
>>>
'Guto'
```

O parâmetro `input()` é uma string de prompt que é impressa, perguntando ao usuário o que ele deve digitar.

Conversão de String

O resultado de `input()` é sempre uma string, por isso pode ser necessária uma conversão, como `int()` convertê-lo em um número

```
>>> idade = entrada ('qual é a sua idade?')
Qual é a sua idade? 25
>>> idade
'25'
>>> int (idade)
25
```

Dica importante

As funções `dir` e `help` no *Python* são de grande ajuda, não somente para iniciantes como também para quem já programa em *python*. O propósito dessa função é obter informações sobre determinados tipos, classes ou funções.

Tipo float

Tipo real, tipo decimal

OBS: O separador de casas decimais na programação é o . ponto, não a vírgula

errado:

valor = 1,44

certo:

valor = 1.44

É possível fazer dupla atribuição

Podemos converter um valor float para int
mas ao convertermos perdemos precisão

```
"""
```

```
valor1, valor2 = 1, 44
```

```
print(valor1)
```

```
print(valor2)
```

```
"""
```

```
"""
```

```
valor = 1.44
```

```
print(int(valor))
```

```
print(type(valor))
```

```
"""
```

podemos criar uma variável do tipo complexo, atribuindo o j

```
"""
```

```
variavel = 5j
```

```
print(variavel)
```

Tipo Booleano

Algebra Booleana, criada por George Boole

True -> Verdadeiro

False -> Falso

Ambos com letra maiúscula no Python

```
ativo = True
```

```
print(ativo)
```

```
#Negação (not)
```

```
"""
```

Se o valor Booleano for verdadeiro o resultado será falso, sempre o contrário."""

```
print(not ativo)
```

```
#Ou (or)
```

```
"""
```

É uma operação binária, ou seja, depende de dois valores. Um ou outro deve ser verdadeiro."""

```
print(ativo or logado)
```

```
# E (and)
```

Também é uma operação binária, ou seja, depende de dois valores.

Ambos os

valores devem ser verdadeiros.

True and True -> True

True and False -> False

False and False -> False

Escopo de variáveis

Dois casos de escopo:

1- Variáveis globais:

- São reconhecidas, ou seja, seu escopo compreende todo o programa.

2- Variáveis locais: São reconhecidas apenas no bloco onde foram declaradas, ou seja, ela está limitada ao bloco onde está declarada.

Para declarar variáveis em Python, fazemos:

nome_da_variável = valor_da_variável

*Python é uma linguagem de tipagem dinâmica.

Exemplo em C: Exemplo em Java:

int numero = 42 int numero= 42

```
numero = 42 #exemplo de variável global  
print(numero)
```

```
numero = 5  
if numero > 10: #A variável 'novo', está declarada dentro do bloco if, ou  
seja, é uma variável local  
    novo = numero + 10  
    print(novo)  
print(novo)
```

Estruturas Lógicas e condicionais

Estruturas Lógicas: and (e), or (ou), not (não), is (é)

Operadores unários:

-not

Operadores binários:

-and, or, is

Para o 'and', ambos precisam ser True

para 'or', um ou outro precisa ser True

Para o 'not', o valor do booleano inverte.

Para 'is', um valor é comparado a outro.

ativo = False

logado = True

if ativo and logado:

 print('Seja bem-vinde usuário')

else:

 print('Você precisa ativar seu e-mail')

Condicionais: if, else, elif

idade = int(input('Qual sua idade? '))

if idade < 18: #O bloco inicia quando colocamos : dois pontos

 print('Menor de idade') #Lembrar dos 4 espaços

elif idade > 18:

 print('Maior de idade')

elif idade == 18:

 print('Você tem 18 anos')



Considerações finais

A [referência oficial da linguagem Python](#) oferece explicações claras sobre a sintaxe, as convenções e as práticas de programação recomendadas. Ele permite que você encontre explicações sobre assuntos mais complicados ou sobre problemas que você pode estar tendo. Quando usado corretamente, um guia de referência como esse pode acelerar drasticamente o progresso de aprendizado.

Em caso de dúvidas contate o suporte da Mr. Academy no botão sobre nós