

[GIM] Curso de Férias - Python

Luiz Eduardo Barros e Victor Matheus Castro

Conteúdo do Curso

- Variáveis;
- Operadores;
- Entrada e Saída de dados;
- Scripts Executáveis;
- Estruturas de Controle;
- Listas
- Funções
- Strings
- Manipulação de Arquivos
- Orientação a Objetos

Certificado

- Participar das atividades desenvolvidas;
- Mínimo 75% de presença no curso.

O Que é Python?

"Python é uma linguagem de programação poderosa e fácil de aprender. Ela possui estruturas de dados de alto nível e uma simples mas eficiente abordagem da programação orientada a objetos. Sua elegante sintaxe e tipagem dinâmica juntamente com seu interpretador nativo fazem dela a linguagem ideal para scripting e desenvolvimento rápido de aplicações em diversas áreas sob várias plataformas".

Guido Van Rossum

Em que tipo de aplicações eu usaria Python?

Python é uma linguagem de uso geral que pode ser empregada em vários tipos de problemas. A biblioteca padrão inclui módulos para processamento de texto e expressões regulares, protocolos de rede (HTTP, FTP, SMTP, POP, XML-RPC, IMAP), acesso aos serviços do sistema operacional, criptografia, interface gráfica etc. Além da biblioteca padrão, existe uma grande variedade de extensões adicionais para todo tipo de aplicação.

Mercado

- Empresas:

Globo.com, YouTube, Netflix, Google, **Evolux**, etc.

- Salários:

R\$ 1.500 a R\$ 7.000

Instalação

- No Linux e Mac OS, por padrão, o interpretador Python já vem instalado.
- Para verificar a versão do python:

```
python -v
```

- Instalação no Linux (Ubuntu 14.04):

```
sudo apt-get install python3.4  
sudo apt-get install idle-python3.4
```

- No Windows: Baixar a versão do Python no site:
<https://www.python.org/>

Controle de Bloco por Identação

```
if(true):  
    # bloco de código  
    #(ao fim da seção identada  
    # termina o bloco)  
# próxima instrução (após o if)
```


Modo Interativo

O interpretador interativo é uma característica diferencial da linguagem, porque há a possibilidade de testar e modificar funções antes de incluí-las nos programas. Isso permite um grau de aprendizado relativamente alto em relação às outras linguagens. Por essa e outras características é que Python é uma linguagem intuitivamente fácil de se aprender.

```
$ python
```

```
>>> 1+1
```

```
2
```

```
>>> a = 1 + 1
```

```
>>> print a
```

```
2
```

Variáveis - nada a declarar!

Python é uma linguagem dinamicamente tipada. Isso quer dizer que não é necessário tipar as variáveis para usa-las. Por um processo chamado "binding", atribui-se um objeto a um nome e este nome incorpora o tipo do objeto. Assim, para atribuirmos o valor 2, que é um inteiro, ao nome "valor1", só precisamos fazer `valor1=2`, e "valor1" será automaticamente tipado por Python como inteiro.

- É um nome que se refere a um valor;
- Armazena dados utilizados no programa;
- Possui tipos diferentes;
- Usem sempre letras ou _ para começar nomes de variáveis e **JAMAIS** utilize espaços;
- Python 3 permite acentos;
- **IMPORTANTE:**
Use nomes significativos para suas variáveis.

Teste no seu terminal:

```
>>> import keyword  
>>> print (keyword.kwlist)
```

Tipos Primitivos

Existem três tipos primitivos de dados, são eles:

- **Númericos** Armazena números inteiros ou em ponto flutuante.
- **String** Armazena caracteres/textos.
- **Lógico** Armazena valores lógicos.

Operadores

Operadores Aritméticos

Nome	Operador	Descrição	Exemplo
Adição	+	Soma dos valores de duas variáveis	$a + b$
Subtração	-	Subtração do valor de uma variável pelo valor de outra	$a - b$
Multiplicação	*	Multiplicação de uma variável por outra	$a * b$
Exponenciação	**	Valor de uma variável elevado ao valor de outra	$a ** b$
Divisão	/	Divisão inteira de uma variável por outra	a/b
Módulo	%	Resto da divisão de uma variável por outra	$a \% b$

Operadores de Atribuição

Nome	Operador	Descrição
Atribuição	=	Atribui um valor a uma variável
Atribuição com soma	+=	Atribui a uma variável o valor dela somado com outro valor
Atribuição com subtração	-=	Atribui a uma variável o valor dela subtraído de outro valor
Atribuição com multiplicação	*=	Atribui a uma variável o valor dela multiplicado com outro valor
Atribuição com divisão	/=	Atribui a uma variável o valor dela dividido com outro valor
Atribuição com módulo	%=	Atribui a uma variável o resto da divisão do valor dela com outro valor

Operadores Relacionais

Nome	Operador	Descrição	Exemplo
Igualdade	==	Verifica se dois valores são iguais	a == b
Diferença	!=	Verifica se dois valores são diferentes	a != b
Maior	>	Verifica se um valor é maior que o outro	a > b
Menor	<	Verifica se um valor é menor que o outro	a < b
Maior ou igual	>=	Verifica se um valor é maior ou igual a outro	a >= b
Menor ou igual	<=	Verifica se um valor é menor ou igual a outro	a <= b

Operadores Lógicos

Nome	Operador	Descrição	Exemplo
AND	and	Verifica se duas operações são verdadeiras	a and b
OR	or	Verifica se pelo menos uma das operações é verdadeira	a or b
NOT	not	Resulta na operação inversa de uma dada operação	not a

Entrada e Saída de Dados

Entrada de Dados

- Possibilita a obtenção de dados externos em tempo de execução;
- Comando **input** e **raw_input**:
Comando usados para solicitação dos dados;
Recebe como parâmetro uma mensagem a ser exibida;
Retorna o dado obtido.

```
>>> variável = input(" Mensagem a ser exibida: ")
```

IMPORTANTE:

- Em algumas versões do Python **input** e **raw_input** retornam um elemento do tipo String por padrão, assim, podemos (e devemos) fazer conversões de tipo (type casting) explicitamente.

```
>>> number = int(input(" Digite um número:"))
```

Saída de dados

- Permite exibir os resultados do programa;
- Comando **print**:
 - Comando usado para exibir os resultados;
 - Recebe como parâmetro uma variável ou um texto (String) a ser exibido;
 - Retorna para o usuário o valor da variável ou texto (String) passado como parâmetro.

```
>>> print "%d" %number
```

```
2
```

```
>>> print("Texto a ser exibido")
```

```
Texto a ser exibido
```


Scripts Executáveis

- Conjunto de comandos relacionados;
- Scripts podem ser codificados em qualquer editor de texto que suporte edição em UTF-8;
- O arquivo deve ser salvo com a extensão .py;
- Para executar o script digite o seguinte comando no terminal:
`python nomedoprograma.py`

Estruturas de Controle

Comandos condicionais

- Instruções **if**, **else**, **elif**:

```
if x > 2 :  
    print ( "maior que dois" )  
elif x == 2 :  
    print ( "dois" )  
else :  
    print ( "menor que dois" )
```

Comandos de Repetição

Laço While

- Estrutura:

```
while True:
```

```
    #Bloco de instrução
```

Comandos break e continue

- break:
Faz o laço terminar imediatamente.
- continue:
Reinicia uma nova iteração imediatamente, interrompendo apenas a iteração atual.

Função range()

- Gera uma lista de números em um determinado intervalo de valores.

Nome	Descrição	Exemplo
Chamada com 1 parâmetro	Retorna uma lista de números de 0 a y-1.	<code>range(y)</code>
Chamada com 2 parâmetros	Retorna uma lista de números de x a y-1.	<code>range(x,y)</code>
Chamada com 3 parâmetros	Retorna uma lista de números de x a y-1 com um passo z.	<code>range(x,y,z)</code>

Comando for

- Estrutura:

```
>>> for x in range(5):  
    print(x)  
0  
1  
2  
3  
4
```

Curiosidades do dia

Teste Unitário

- Consiste em realizar alguns testes de uma função que foi criada;
- Python oferece ferramentas para realizar esses testes de forma eficiente;
- Doctest e Unittest (abordaremos apenas o primeiro).
- Biblioteca doctest;
- Escreve-se a função em um arquivo .py e adiciona-se algumas linhas extras contendo os resultados esperados e as funções da biblioteca necessárias, utilizamos o `import doctest` e após isso `doctest.testmod()`

Depuração de Programas

O Que é depuração?

- Ao desenvolver um programa, erros irão ocorrer, e não há como fugir disso;
- Alguns serão simples de encontrar, outros nem tanto;
- Localizar e corrigir tais erros é chamado de depuração;