Python

Prof. Guilherme Arthur de Carvalho

@decarvalhogui

@guicarvalho(linkedin e Github)

- O que é Python

- Como e onde devo usar o Python?

O que é Python?

Linguagem baseada na linguagem ABC que era uma linguagem pensada para iniciantes, devido a sua facilidade de aprendizagem e utilização.

Características:

- Linguagem fácil e intuitiva

- código aberto, para que todos possam contribuir

- código tão inteligível quanto o Inglês

- adequada para tarefas diárias, e produtiva.

Python é versátil:

- Tipagem dinâmica e forte

- Multiplataforma e multiparadigma

- Comunidade gigante e ativa

- Curva de aprendizado baixa

- Python não é para mobile

Digite o seguinte comando:

python --version

Este comando retornará a versão do python que está instalada em sua máquina.

Agora digite:

pip --version

Esse comando retornará a versão do pip que está instalada em sua máquina. O pip é o gerenciador de pacote do Python. Com ele você poderá adicionar novas funcionalidades ao seu Python.

instalar as extensões:

Python

autoDocString

IntelliCode

Tipos em Python

|  |  |
| --- | --- |
| **Tipo - descrição** | **Tipo** |
| Texto | str |
| Numérico | int, float, **complex** |
| **Sequência** | **list, tuple, range** |
| **Mapa** | **dict** |
| **Coleção** | **set, fronzenset** |
| Booleano | **bool** |
| **Binário** | **bytes, bytearray, memoryview** |

Comandos para sair do Python:

**quit(), exit(), ctrl+z**

Para executar um arquivo do python sem ter que entrar primeiro no modo do python, colocamos o seguinte comando:

**python –i** <nome do arquivo.py>

O **-i** vai entrar no python já executando o arquivo, mostrando o resultado da execução do mesmo.

**import** <biblioteca>

faz a importação da biblioteca <biblioteca> desejada.

Ex.: **import** math

**dir()**

Sem argumentos, devolve a lista de nomes no escopo local atual. Com um argumento, tentará devolver uma lista de atributos válidos para esse objeto.

**Referência:**

<https://docs.python.org/pt-br/3/library/functions.html#dir>

**import**

Detecta importações de irmãos e as converte em importações relativas.

**Sintaxe:**

**import** <nome da biblioteca>

**Exemplo:**

**Import** math

**help()**

Entra no modo de help, onde para saber o que cada comando faz, é só colocar o que deseja aprender e apertar enter que automaticamente o python vai mostrar todas as informações do que procura.

Para sair do modo do **help()** apertar a tecla **q**

No Python não existe variáveis CONSTANTES, quando se “deseja” uma variável **CONSTANTE¸como boas práticas para o Python, uma constante se usa variável toda em maiúscula.**

**Comentários**

Para inserir um comentário no Python, utilizamos o “**#”** na frente do comentário

**Blocos de comandos**

No python, basta ter um espaço antes da linha que está dentro do bloco para que a linguagem entenda que aquela linha faz parte da estrutura anterior, é fundamental a indentação para o Python.

**Estrutura condicional de Simples escolha**

No Python, temos a estrutura if.

Para a estrutura **if, then**

**If** <condição>**:**

<comandos>

Para a estrutura **if, then** resumido

<variável>**=**<valor> **if** <condição>**:**<valor\_falso>

**Exemplo:**

msg\_idade = " maior de idade " if idade >= 18 else " menor de idade "

Para estrutura **if, then else**

**If** <condição>**:**

<comandos>

**else:**

<comandos>

Para estrutura **if, then elseif else**

**If** <condição>**:**

<comandos>

**elif:**

<comandos>

**else:**

**Estrutura condicional Múltipla Escolha**

No Python, temos a estrutura **match** que é o *“****case”*** nas outras linguagens

**Syntaxe:**

**match** <variável>**:**

**case** <valor 1>**:**

<conjunto de comandos 1>

**case** <valor 2>**:**

<conjunto de comandos 2>

**.**

**.**

**.**

**case** <valor n>**:**

<conjunto de comandos n>

**case \_:**

<conjunto de valores para quando não tiver nenhum dos valores

**Comando List(Range(<valor\_inicial>,<valor\_final>,<intervalo>))**

Gera uma lista de valores de um valor inicial a um valor final onde o **Range** é o range que quer ser gerado.

**Valor\_inicial:** será o primeiro valor a aparecer

**Valor\_Final:**  será a referência para a parada da contagem, podendo ou não aparecer, uma vez que dependerá do valor inicial e do intervalo para que o mesmo apareça ou não.

**Intervalo:** Intervalo que será dado entre cada número

**Exemplos:**

**list(range(4))** Irá gerar uma lista com quatro números {0,1,2,3}, caso não seja informado o início, será adotado por padrão o início como 0

**list(range(5,4))** Não aparecerá nada uma vez que o primeiro é o valor inicial (5), o segundo é o valor final (4), não gerando assim nenhum valor

**list(range(5,15))** Irá gerar uma lista com os números entre 5 até 14. Não aparecerá o número 15, uma vez que eu quero os números que vá de 5 até 15, não contando o 15

**list(range(5,15,3))** Irá gerar uma lista com os números entre 5 até 14 pulando de 3 em 3 até chegar ao 15 {5;8;11;14}. Os demais números não aparecerão.