



PYTHON

INTRODUÇÃO

Python

O Python se tornou uma das linguagens de programação mais populares do mundo nos últimos anos. Isso se deve, principalmente, à sua versatilidade: É uma linguagem de programação de uso geral utilizada para o aprendizado de máquinas, construção de sites e até para automação de tarefas e testes de softwares.



Como surgiu o Python?

A Linguagem surgiu inicialmente, pelo programador holandês Guido van Rossum, no final da década de 1980.

Sua primeira versão foi lançada só em 1991 no Centrum Wiskunde & Informatica
- CWI (Instituto Nacional de Pesquisa para Matemática e Ciência da Computação)
- Holanda.

A linguagem trouxe grandes avanços para o desenvolvimento de código aberto, utilizando as PEPs (“Propostas de Enriquecimento do Python”).

As PEPs do Python funcionam para descrever mudanças na linguagem ou em suas normas. O público pode avaliá-las e aceitá-las ou rejeitá-las, depois de muitas discussões.

Qualquer pessoa pode escrever e enviar uma PEP para avaliação.





Características

Interpretada

Não precisa passar pelo processo de compilação.

O processo de interpretação é executado dentro de máquinas virtuais, nas quais o código passa por uma camada intermediária que irá traduzir os comandos de programa para código binário.

Sintaxe simples

Fácil de aprender e muito próxima da linguagem falada por nós.

Multiparadigma

Permite programar em vários paradigmas, tais como:

Procedural: Instruções transmitidas ao computador na sequência em que devem ser executadas;

Funcional: Programas construídos aplicando e compondo funções;

Orientação a objetos: Perspectiva do mundo real para a programação.



Hello World

Para exibir uma mensagem na tela utilizamos a função `print()`.

```
print().  
print('Hello World!')  
print(10)
```

Podemos exibir na tela valores de tipos diferentes de forma concatenada, separados por vírgula.

*Para Strings utilizamos aspas simples ou duplas.

```
print('João',30)  
print ('Maria', 20.50)  
print ('elo Horizonte', 4, 'Fevereiro' , 2000)
```

Comentários

O Python também possui formas de incluirmos comentários no código.

Comentário de uma linha (Colocamos o símbolo #).

```
#Imprimi a frase Hello World na tela .  
print('Hello World!')
```

Comentário de várias linhas (Colocamos três pontinhos antes e depois do bloco de texto).

```
...  
Insira aqui o comentário com mais de uma linha.  
...
```

Tipos de dados em Python

Python é uma linguagem dinamicamente tipada, isso significa que não é necessário declarar o tipo de variável ou fazer casting (mudar o tipo de variável), pois o Interpretador se encarrega disso para nós.

Um outro fato interessante é que, se por algum motivo precisarmos alterar o tipo da variável durante a execução do programa, é possível fazer essa mudança. Isso não acontece em linguagem como Java, onde uma variável nasce e morre sendo do mesmo tipo de dado.

Tipos de dados em Python

Inteiro (`int`)

idade = 20

anoNasc = 2000

Ponto Flutuante (`float`)

peso = 78.55

altura = 1.80

salario = 3500.89

String (`str`)

nome = 'Ricardo'

cidade = 'Belo Horizonte'

cpf = '123.545.456-65'

Boolean (`bool`)

casado = True

perfil_ativo = False

A função `type` informa qual é o tipo da variável

```
num = 10
```

```
type(num)
```

Informa que num é do tipo `int`

Tipos de dados em Python

Listas (list)

As listas agrupam um conjunto de elementos variados, como: inteiros, floats, strings, outras listas e outros tipos.

```
alunos = ['Ricardo', 'Ana', 'Maria', 'José', 'Pedro']  
notas = [10.5, 8.5, 5.8, 8.0, 10.0]
```

Tupla (tuple)

A tupla é um tipo que também agrupa um conjunto de elementos, porém sua forma de definição é diferente, utilizamos parênteses e os dados inseridos são imutáveis, ou seja, após sua definição, não podem ser modificados.

```
numeros = (18,20,212,22,52,8,30)  
pontos = (10,20,30,40,50)
```

Tipos de dados em Python

Dicionários (dict)

Os dicionários são utilizados para agrupar elementos através da estrutura de chave e valor, de forma que a chave é o primeiro elemento, seguido por dois pontos e pelo valor.

```
salario= {'Sandra':2500.65,  
          'Elizabeth': 1200.00, 'Roberto': 3555.70  
}
```

```
idade = {'Sandra':35, 'Elizabeth':58,  
         'Roberto':68}
```

Operações Matemáticas

Além das quatro operações já conhecidas: Adição (+), subtração (-), multiplicação (*) e divisão (/) vale a pena destacar:

Potência é definida através do operador `**`

Ex: 2 elevado a 3

`2**3`

Saída: 8

Módulo (%)

Retorna o resto da divisão inteira entre dois números.

Ex: `7%3`

Saída: 1

Divisão inteira (//)

Retorna o resultado da divisão inteira entre dois números

Ex: `7//3`

Saída: 2

Operações Matemáticas

Calculando e mostrando a média do aluno

```
aluno = 'Vitor'  
notaPort = 10  
notaMat = 8.5  
notaIng = 9.5  
notaGeo = 7.5  
notaHist = 8.7
```

```
media = (notaPort+ notaMat+ notaIng+ notaGeo+ notaHist)/5.0
```

```
print('A média do aluno: ', aluno, ' foi: ', media )
```



Manipulando Strings

Python fornece diversas funções para manipulação de Strings.

As funções não realizam **as** mudanças diretamente na String, para alterar o conteúdo da String é preciso fazer com que ela receba o próprio valor atualizado.

```
String com valor de origem  
cidade = ' Bemo Horizontes '  
print(cidade)
```


Manipulando Strings

Retirando o caractere s da String

```
cidade = cidade.replace('s', '' )
```

Sbstituindo o caractere m por l

```
cidade = cidade.replace('m', 'l')
```

Retirando os espaços vazios e colocando os caracteres em letra maiúscula

```
cidade = cidade.strip().upper()
```

Colocando os caracteres em letra maiúscula

```
cidade = cidade.upper()  
print(cidade)
```



Manipulando Strings

O mesmo código poderia ser escrito da seguinte maneira

```
cidade = '  Bemo Horizontes  '
print(cidade)
cidade = cidade.replace('s','').replace('m', 'l').strip().upper()
print(cidade)
```

Entrada de Dados

A entrada de dados via teclado se dá através da função `input()`. Podemos passar uma mensagem como parâmetro para direcionar o usuário.

```
nome = input('Informe o seu nome: ')\nidade = input('Informe a sua idade: ')\nsalario = input('Informe o seu salário: ')
```

A função `input()`, converte os dados de entrada para `String` independente do tipo, sendo assim é preciso realizarmos a conversão dos dados de entrada para o tipo de dados desejável.

```
idade = int (input ('Informe a sua idade: '))\nsalario = float (input ('Informe o salário: '))
```

Formatação do print()

O python fornece diferentes formas de formatar a String de saída
Utilização do f-string

```
nome = 'Ricardo'  
idade = 17
```

```
print(f'O nome da aluno é {nome} e sua idade é {idade} anos.')
```

Formatação do print()

Operador de Formatação %

```
nomeAluno = 'José Antônio'  
print('Nome do aluno: %s' %(nomeAluno))
```

```
nomeAluno = 'Fabricio Daniel'
```

```
idadeAluno = 15
```

```
mediaAluno = 8.45
```

```
print('Nome do aluno é %s, ele tem %d anos e sua média é %.2f.' %(nomeAluno,  
idadeAluno, mediaAluno))
```

Os operadores de formatação de strings com % não funcionam diretamente com valores booleanos. É preciso converter o valor booleano para string antes de usá-los na formatação.

```
x = True
```

```
print("Valor de x: %s" % str(x))
```


Formatação do print()

Format()

É possível usar o método `format()` para fazer a formatação de strings, neste caso, não é necessário os operadores `%`, pois os marcadores são apenas `{ }`.

```
nomeAluno = 'José Antônio'
idadeAluno = 15
mediaAluno = 8.45
```

```
print('Nome do aluno é {}, ele tem {} anos e sua média é {}.'  
      .format(nomeAluno, idadeAluno, mediaAluno))
```

Formatação do print()

Caracteres Especiais

\n para pular uma linha no texto (função do Enter).

```
print("Estudar é um esforço constante,\n É como cultivar uma  
planta,\nPrecisamos de dedicação e paciência,\nPara ver o fruto  
amadurecer.")
```

\t para tabulação.

```
print('Quantidade\tQualidade\n5 amostras\tAlta\n3 amostras\tBaixa')
```

\\ para imprimir uma única barra invertida.

```
print("Caminho do arquivo: C:\\arquivos\\documento.csv")
```

\\" para imprimir aspas duplas.

```
print("Ouvi uma vez \\"Os frutos do conhecimento são os mais doces e  
duradouros de todos.\")
```

\' para imprimir aspas simples.

```
print('Minha professora uma vez disse \'Estudar é a chave do sucesso.\'')
```