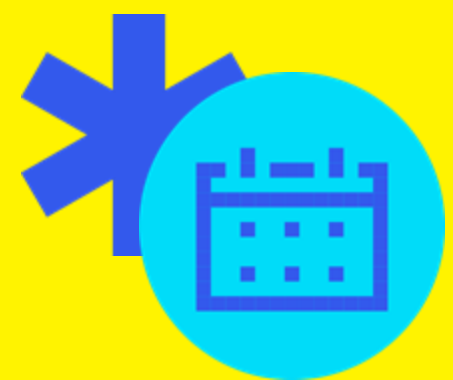




Print, Variáveis, Input e Operadores em Python

< thefutureisblue.me />





Agenda

O que vamos aprender?

Variáveis

Características do Python

Input

Operadores

Prática



Variáveis

- Variáveis são pequenos espaços de memória, utilizados para armazenar e manipular dados. Em Python, os tipos de dados básicos são: tipo inteiro (armazena números inteiros), tipo float (armazena números em formato decimal), e tipo string (armazena um conjunto de caracteres). Cada variável pode armazenar apenas um tipo de dado a cada instante. Em Python, diferentemente de outras linguagens de programação, não é preciso declarar de que tipo será cada variável no início do programa.

Variáveis

➤ Quando se faz uma atribuição de valor, automaticamente a variável se torna do tipo do valor armazenado, como apresentado nos exemplos a seguir:

Exemplos:

```
a = 29
```

A variável a se torna uma variável do tipo inteiro.

```
b = 1.8
```

A variável b se torna uma variável do tipo float

```
c = 'Pelos poderes de  
Grayskull! Eu tenho  
a força!'
```

A variável c se torna uma variável do tipo string.

Características do Python!



Sintaxe elegante, simples e clara.

- * Tipagem dinâmica.

- * Tipagem forte.

- * Blocos de código são delimitados por indentação.

Comando Print!

➤ Print

- A sintaxe da função `print()` é bem simples, basta informar entre parênteses o conteúdo que deseja imprimir na tela do terminal.

➤ Olá mundo!!!!

- Precisa de um delimitador: aspas simples `"` ou aspas duplas `""`.
- Colocar a mensagem dentro de parênteses.

Exemplo: `print('Olá mundo!!!')`

Sintaxe elegante

- Permite fácil legibilidade do código-fonte.
- A simplicidade torna a linguagem menos verbosa.
- Linguagem ideal para scripts.
- Desenvolvimento rápido.

```
print('Olá mundo!!!')
```

```
x = 'Blue'
```

```
print(x)
```

```
Blue
```



Características do Python!

~~* Sintaxe elegante, simples e clara.~~



Tipagem dinâmica.

* Tipagem forte.

* Blocos de código são delimitados por indentação.



Tipagem dinâmica

É uma característica que não exige declarações de tipos de dados, pois são capazes de escolher que tipo utilizar, dinamicamente, para cada variável, podendo alterá-los durante a compilação ou a execução do programa.

Criação de variável inteira

```
x = 1
```

Criação de variável String

```
x = 'A'
```

```
b, t = 10, 20
```

```
print('Resultado:', b + t)
```

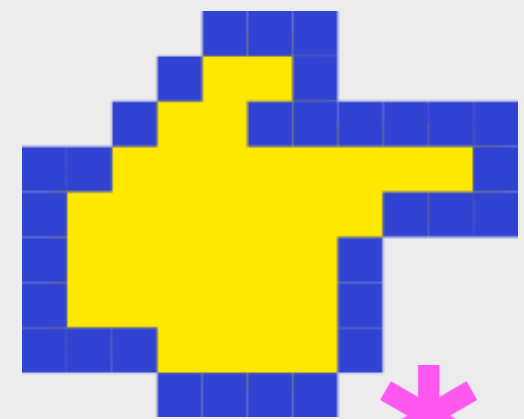
```
Resultado: 30
```



Características do Python!

~~* Sintaxe elegante, simples e clara.~~

~~* Tipagem dinâmica.~~



Tipagem forte.

* Blocos de código são delimitados por indentação.

Tipagem forte

Para o Python, a tipagem forte significa que o interpretador avalia as expressões (evaluate) e não faz conversões automáticas entre tipos não compatíveis (conversões de valores).

```
i, j = 10, 'João'
print(i)
print(j)
print(i + j)
```

10

João

```
-----
TypeError                                Traceback (most
recent call last)
<ipython-input-18-16ad423aa9ac> in <module>()
      2 print(i)
      3 print(j)
----> 4 print(i+j)
TypeError: unsupported operand type(s) for +:
'int' and 'str'
```

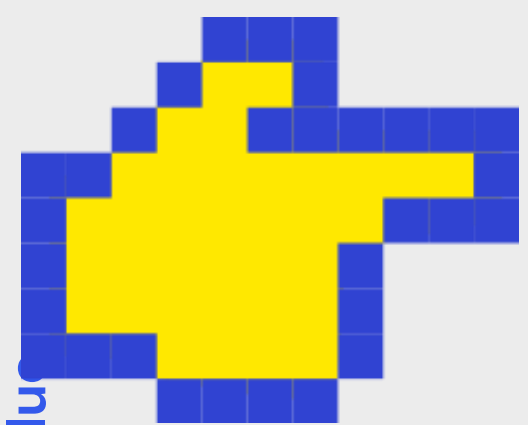


Características do Python!

~~* Sintaxe elegante, simples e clara.~~

~~* Tipagem dinâmica.~~

~~* Tipagem forte.~~



Blocos de código são delimitados por indentação.

Indentação de código

➤ **Indentar** é o recuo do texto em relação à sua margem, ou seja, se antes de escrevermos uma instrução, utilizarmos 4 espaçamentos da margem esquerda até a instrução propriamente dita, podemos dizer que a indentação utilizada possui 4 espaços.

➤ No Python **não usamos** begin e End, nem mesmo { } para delimitar código e seus blocos. Utilizamos somente indentação.

```
print('Nível 1') # Primeiro Nível
```

```
if True:  
    print('Nível 2') # Segundo Nível
```



Comando **Input**

➤ A atribuição de valor para uma variável pode ser feita utilizando o comando `input()`, que solicita ao usuário o valor a ser atribuído à variável.

➤ Exemplo:

```
nome = input('Digite seu nome e sobrenome: ')
```

Comando **Input**

- O comando `input()`, sempre vai retornar uma string. Nesse caso, para retornar dados do tipo inteiro ou float, é preciso converter o tipo do valor lido. Para isso, utiliza-se o `int (string)` para converter para o tipo inteiro, ou `float (string)` para converter para o tipo float.

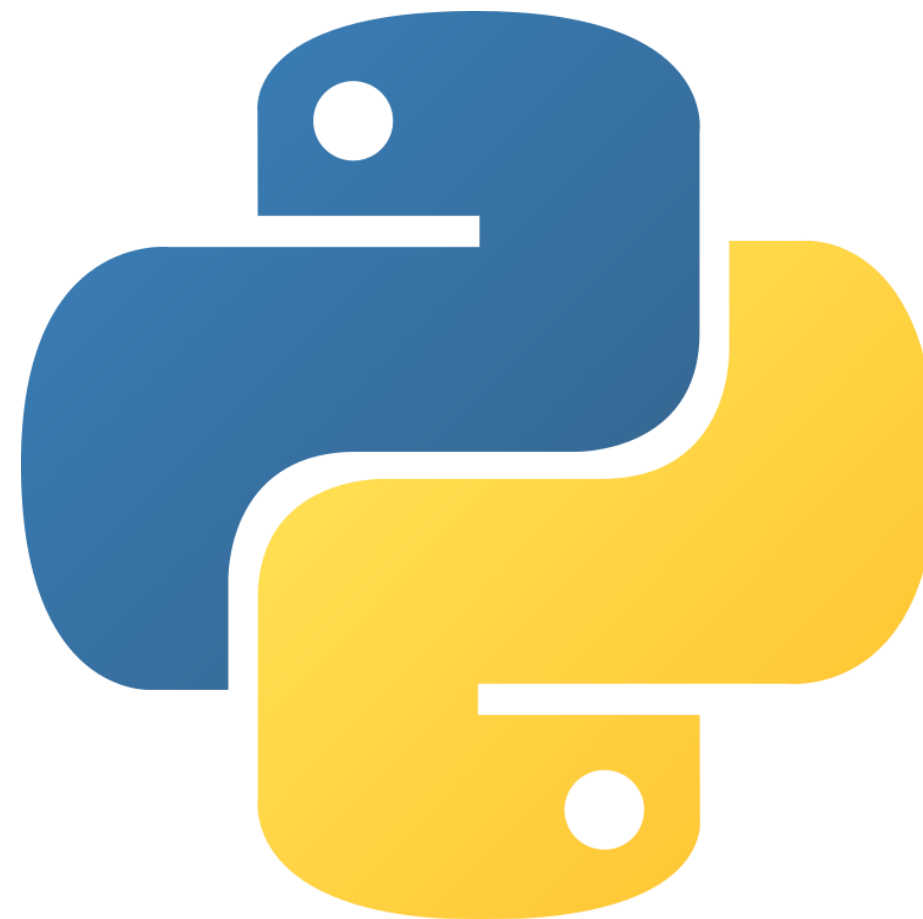
- **Exemplo:**

```
numero_inteiro = int(input('Digite seu número da sorte: '))
```

```
numero_decimal = float(input('Digite sua altura: '))
```

Operadores em Python

- **Operadores Aritméticos**
- Operadores Relacionais
- Operadores Lógicos
- Operadores bit a bit



Operadores Aritméticos

Operador	Descrição	Exemplo
+	Adição	$2 + 3$
-	Subtração	$2 - 2$
*	Multiplicação	$2 * 3$
/	Divisão	$5/4$
//	Divisão Inteira	$4/3$
**	Exponenciação	$2^{**}100$
%	Resto de Divisão(mod)	$10 \% 2$

Operadores em Python

- ~~Operadores Aritméticos~~
- **Operadores Relacionais**
- Operadores Lógicos
- Operadores bit a bit



Operadores Relacionais

Operador	Descrição	Exemplo
>	Maior que	2 > 3
<	Menor que	2 < 4
>=	Maior igual que	2 >= 2
<=	Menor Igual que	4 <= 4
!=	Diferente de	True != False
==	Igual a	True == 1

Operadores em Python

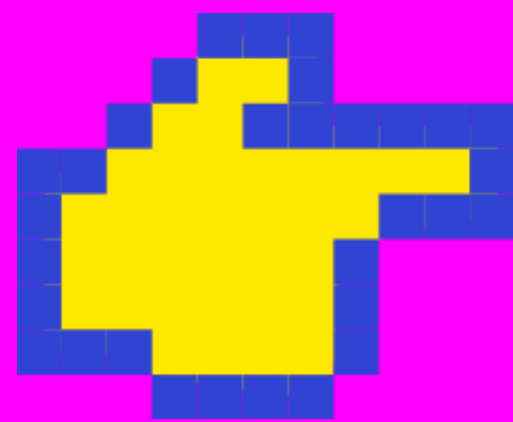
- ~~Operadores Aritméticos~~
- ~~Operadores Relacionais~~
- **Operadores Lógicos**
- Operadores bit a bit



Operadores Lógicos

Operador	Descrição	Exemplo
not	Negação Lógica	not True
and	E lógico	True and False
or	OU lógico	False or True

Operadores em Python



- ~~Operadores Aritméticos~~
- ~~Operadores Relacionais~~
- ~~Operadores Lógicos~~
- Operadores bit a bit

Operadores bit a bit

Operador	Descrição	Exemplo
~	complemento bit a bit	~b
<<	deslocar bits à esquerda	b << 1
>>	deslocar bits à direita	b >> 1
&	E (and) bit a bit	b & 0x01
^	ou exclusivo (XOR) bit a bit	b ^ 0x01
	ou (OR) bit a bit	b 0x01

Booleanos - True e False

➤ Em Python, uma variável pode assumir valor booleano **True** (verdadeiro) ou **False** (falso). Esses valores são úteis para representar, por exemplo, o resultado de uma comparação.

➤ Exemplo:

$1 > 2$

Saída: false

$3 > 2$

Saída: True



Botando para rodar!

Vamos praticar todos operadores conceituados!

Por hoje é só! Obrigado! =)

Até a próxima aula.