



# VARIÁVEIS

Variáveis são contêineres para armazenar valores de dados.

## Criação de variáveis

Python não tem comando para declarar uma variável.

Uma variável é criada no momento em que você atribui um valor a ela pela primeira vez.

### Exemplo

```
x = 5
y = "João"
print(x)
print(y)
```

As variáveis não precisam ser declaradas com nenhum *tipo* específico e podem até mesmo mudar de tipo após terem sido definidas.

```
x = 4          # x é do tipo int
x = "Solo"     # x agora é do tipo str
print(x)
```

## Casting

Se você deseja especificar o tipo de dados de uma variável, isso pode ser feito com o casting.

### Exemplo

```
x = str(3)     # x será '3'
y = int(3)     # y será 3
z = float(3)   # z será 3.0
```

## Obtenha o tipo da variável

Você pode obter o tipo de dados de uma variável com a `type()` função.

### Exemplo

```
x = 5
y = "João"
print(type(x))
print(type(y))
```

## Aspas simples ou duplas?

Variáveis de string podem ser declaradas usando aspas simples ou duplas, desde que sejam pares combinados. Se abrir aspas duplas, deve fechar com aspas duplas, assim também para aspas simples.

### Exemplo

```
x = "João"
# é o mesmo que
x = 'João'
```

## Maiúsculas e Minúsculas

Os nomes das variáveis diferenciam maiúsculas de minúsculas.

É o que chamamos de *case-sensitive*.

### Exemplo

```
a = 4
A = "Solo"
#A não substituirá a
```

## Nomes de Variáveis

Uma variável pode ter um nome curto (como `xey`) ou um nome mais descritivo (`idade`, `carname`, `total_volume`). Regras para variáveis Python:

- O nome de uma variável deve começar com uma letra ou o caractere de sublinhado

- O nome de uma variável não pode começar com um número
- Um nome de variável pode conter apenas caracteres alfanuméricos e sublinhados (A-z, 0-9 e \_)
- Os nomes das variáveis diferenciam maiúsculas de minúsculas (idade, Idade e IDADE são três variáveis diferentes)

## Exemplo

```
myvar = "João"
my_var = "João"
_my_var = "João"
myVar = "João"
MYVAR = "João"
myvar2 = "João"

#Nomes de variáveis ilegais:
2myvar = "João"
my-var = "João"
my var = "João"
```

## Nomes de variáveis com várias palavras

Nomes de variáveis com mais de uma palavra podem ser difíceis de ler.

Existem várias técnicas que você pode usar para torná-los mais legíveis:

### Camel Case

Cada palavra, exceto a primeira, começa com uma letra maiúscula:

```
myVariableName = "João"
```

### Pascal Case

Cada palavra começa com uma letra maiúscula:

```
MyVariableName = "João"
```

### Snake Case

Cada palavra é separada por um caractere de sublinhado:

```
my_variable_name = "John"
```

## Muitos Valores para Variáveis Múltiplas

Python permite que você atribua valores a várias variáveis em uma linha:

### Exemplo

```
x, y, z = "Laranja", "Banana", "Cereja"
print(x)
print(y)
print(z)
```



**Nota:** Certifique-se de que o número de variáveis corresponda ao número de valores, ou então você obterá um erro.

## Um valor para múltiplas variáveis

E você pode atribuir o *mesmo* valor a várias variáveis em uma linha:

```
x = y = z = "Laranja"
print(x)
print(y)
print(z)
```

## Descompacte uma coleção

Se você tem uma coleção de valores em uma list, tuple, etc. Python permite que você extraia os valores em variáveis. Isso é chamado de *descompactação*.

### Exemplo

```
fruits = ["maçã", "banana", "cereja"]
x, y, z = frutas
print(x)
print(y)
print(z)
```

## Variáveis de saída

A instrução `print` costuma ser usada para gerar variáveis.

Para combinar texto e variável, usamos o operador aritmético `+`:

### Exemplo

```
x = "maravilhoso"
print("Python é " + x)

#Você também pode usar o +caractere para adicionar uma variável a outra variável:
x = "Python é"
y = "maravilhoso"
z = x + y
print(z)

#Para números, o + funciona como um operador matemático:
x = 5
y = 10
print(x + y)

#Se você tentar combinar uma string e um número, o Python apresentará um erro:
x = 5
y = "João"
print(x + y)
```

## Variáveis globais

Variáveis que são criadas fora de uma função (como em todos os exemplos acima) são conhecidas como variáveis globais e podem ser usadas em qualquer parte do código, tanto dentro quanto fora das funções.

### Exemplo

```
x = "maravilhoso"
def myfunc():
    print("Python é " + x)
    myfunc()
```

Se você criar uma variável com o mesmo nome dentro de uma função, essa variável será local e só pode ser usada dentro da função. A variável global com o mesmo nome permanecerá como era, global e com o valor original.

### Exemplo

```
x = "maravilhoso"
def myfunc():
    x = "fantástico"
    print("Python é " + x)
    myfunc()
    print("Python é " + x)
```

## A palavra-chave global

Normalmente, quando você cria uma variável dentro de uma função, essa variável é local e só pode ser usada dentro dessa função.

Para criar uma variável global dentro de uma função, você pode usar a palavra - chave `global`.

### Exemplo

```
def myfunc():
    global x
    x = "fantástico"
    myfunc()
    print("Python é " + x)
```



**Nota:** Podemos usar a palavra-chave `global` se quisermos alterar uma variável global dentro de uma função.

```
x = "maravilhoso"
def myfunc():
    global x
    x = "fantástico"
    myfunc()
    print("Python é " + x)
```