

# ASIMOV

---

## Números e mais em Python!

Nesta aula aprenderemos sobre os números em Python e sobre como usá-los.

Aprenderemos sobre os seguintes tópicos:

1.) Tipos de Números em Python    2.) Aritmética básica    3.) Atribuição de Objeto em Python

### Tipos de números

Python tem vários "tipos" de números (literais numéricos). Nós nos concentraremos principalmente em números inteiros e números de ponto flutuante.

Inteiros são apenas números inteiros, positivos ou negativos. Por exemplo: 2 e -2 são exemplos de números inteiros.

Os números de pontos flutuantes em Python são notáveis porque eles têm um ponto decimal neles, ou usam uma exponencial (e) para definir o número. Por exemplo, 2.0 e -2.1 são exemplos de números de ponto flutuante. 4E2 (4 vezes 10 para o poder de 2) também é um exemplo de um número de ponto flutuante em Python.

Ao longo deste curso, trabalharemos principalmente com números inteiros ou tipos simples de números de flutuação.

Aqui está uma tabela dos dois tipos principais que passaremos a maior parte do tempo trabalhando com alguns exemplos:

Exemplo	Tipo
1,2, -5, 1000	Inteiros
1.2, -0.5, 2e2, 3E2	Numero Ponto-flutuante

Agora vamos começar com alguma aritmética básica.

### Aritmética básica

```
In [1]: # Soma
2+1
```

```
Out[1]: 3
```

```
In [2]: # Subtração
2-1
```

Out[2]: 1

```
In [3]: # Multiplicação  
2*2
```

Out[3]: 4

```
In [4]: # Divisão  
3/2
```

Out[4]: 1.5

```
In [5]: # Specifying one of the numbers as a float  
3.0/2
```

Out[5]: 1.5

```
In [6]: # Works for either number  
3/2.0
```

Out[6]: 1.5

```
In [7]: # Quadrado  
2**3
```

Out[7]: 8

```
In [8]: # Outra forma de fazer raiz quadrada  
4**0.5
```

Out[8]: 2.0

```
In [9]: # Ordem das operações em Python igual na matemática normal  
2 + 10 * 10 + 3
```

Out[9]: 105

```
In [10]: # Você pode usar parêntesis para especificar a ordem  
(2+10) * (10+3)
```

Out[10]: 156

## Atribuições de variáveis

Agora que vimos como usar números em Python como calculadora, vejamos como podemos atribuir nomes e criar variáveis.

Usamos um único sinal de igual para atribuir rótulos às variáveis. Vejamos alguns exemplos de como podemos fazer isso.

```
In [11]: # Vamos criar um objeto chamado "a" e atribuir o número 5  
a = 5
```

Agora, se eu chamar *a* no meu script Python, o Python tratará isso como o número 5.

```
In [12]: # Somando objetos  
a+a
```

Out[12]: 10

O que acontece na redefinição? Python nos deixa escrever sobre os valores anteriores?

```
In [13]: # Redefinição  
a = 10
```

```
In [14]: # Checa  
a
```

Out[14]: 10

```
In [15]: # Usa a para redefinir a  
a = a + a
```

```
In [16]: # Checa  
a
```

Out[16]: 20

Os nomes que você usa ao criar esses rótulos precisam seguir algumas regras:

1. Os nomes não podem começar com um número.
2. Não pode haver espaços no nome. Use `_` em vez disso.
3. Não é possível usar nenhum desses símbolos: `'', <> /? | \ () ! @ # $ % ^ & * ~ - +`
3. É considerada a melhor prática (PEP8) que os nomes são minúsculos.

Usar nomes de variáveis pode ser uma maneira muito útil de acompanhar diferentes variáveis em Python. Por exemplo:

```
In [17]: # Use nomes de objetos para manter uma melhor relação com o que está acontecendo  
my_income = 100  
  
tax_rate = 0.1  
  
my_taxes = my_income*tax_rate
```

```
In [18]: # Mostra as taxas  
my_taxes
```

Out[18]: 10.0

Então, o que aprendemos? Aprendemos alguns dos conceitos básicos dos números em

Python. Também aprendemos a fazer aritmética e usar Python como uma calculadora básica. Em seguida, o envolvemos com o aprendizado sobre a atribuição variável em Python.

Em seguida, aprenderemos sobre Strings!