
✓ Fundamentos de Programação em Python

Pontifícia Universidade Católica de Campinas

Prof. Dr. Denis Mayr Lima Martins


Aula 2: Variáveis e Tipos de Dados em Python

Bem-vindo à nossa primeira aula prática de **Fundamentos de Programação em Python!**

Nesta aula, exploraremos os conceitos fundamentais de **variáveis**, **tipos de dados** e **operadores matemáticos**.

O que veremos hoje?

- O que são **variáveis** e como armazenam informações em Python.
- Os principais **tipos de dados**: inteiros (`int`), números decimais (`float`), textos (`str`), valores booleanos (`bool`), entre outros.
- Como utilizar **operadores matemáticos** para realizar cálculos simples.
- A importância da conversão entre tipos de dados.

 **Dica:** Python é uma linguagem dinâmica, ou seja, não precisamos especificar o tipo da variável ao declará-la! Vamos entender isso melhor na prática.

Objetivos de Aprendizagem

Ao final desta aula, você será capaz de:

- Compreender o conceito de **variáveis** e seu papel na programação.
 - Identificar e utilizar corretamente os principais **tipos de dados** em Python (`int`, `float`, `str`, `bool`).
 - Aplicar **operadores matemáticos** (`+`, `-`, `*`, `/`, `//`, `%`, `**`) para realizar cálculos e manipular dados.
 - Diferenciar e converter tipos de dados utilizando funções como `int()`, `float()`, `str()`, entre outras.
-

✓ Variáveis em Python

Em programação, uma variável é um espaço na memória que armazena um valor. Ou seja, as variáveis são contêineres usados para armazenar valores de dados que podem mudar à medida que o programa é executado.

Em Python, uma variável pode ser declarada sem a necessidade de definir seu tipo explicitamente.

O **comando de atribuição** cria novas variáveis e dá a elas valores:

```
1 nome = "Maria"
2 idade = 25
```

```
1 print(nome)
```

```
1 print(idade)
```

```
1 # 0 que vai aparecer aqui?
2 a = 1
3 a = 2
4 print(a)
5 # Isso acontece pois os dois comandos são recebidos, mas o último que fica c
```

▼ Nomes de variáveis

- **Nomes Significativos:** Programadores escolhem nomes descritivos para variáveis, pois isso facilita a compreensão do código.
- **Comprimento do Nome:** Nomes de variáveis podem ser longos e conter letras e números, mas devem começar com uma letra (ou com o caractere `_`).
- **Maiúsculas e Minúsculas:** Embora letras maiúsculas sejam permitidas, convencionou-se o uso de minúsculas. Python diferencia `Idade` e `idade`, pois é *case-sensitive*.
- **Uso do Sublinhado (`_`):** O caractere `_` pode ser usado para separar palavras em nomes de variáveis, como `meu_nome` ou `preco_do_produto_na_china`.
- **Erros de Sintaxe:** Atribuir um nome inválido a uma variável resultará em um erro de sintaxe.

```
1 number_of_participants = 30
2 print(number_of_participants)
```

```
1 numberOfParticipants = 20
2 print(numberOfParticipants)
```

Observe que os nomes das variáveis são **sensíveis a maiúsculas e minúsculas**.

Portanto, `number_of_participants` é diferente de `Number_of_Participants`.

```
1 # Crie uma nova variável chamada Number_of_Participants
2 Number_of_Participants = 500
```

```
3 print(Number_of_Participants)
```

```
1 print(number_of_participants)
```

Lembrete: `print()` pode mostrar vários valores separados por vírgulas.

```
1 print('Com letra minúscula é', number_of_participants)
2 print('Com letra maiúscula é', Number_of_Participants)
```

```
1 print(1, 2, "Test")
```

Exercício 1:

1. Declare três variáveis: `idade`, `peso` e `altura`.
2. Exiba seus valores usando `print()`.
3. Calcule o Índice de Massa Corporal usando as variáveis `peso` e `altura`.

$$IMC = \frac{peso}{altura^2}$$

```
1 # Adicione sua resposta aqui
```

Exercício 2:

1. Calcule a área de um retângulo usando valores armazenados nas variáveis `lado_1` e `lado_2`.
2. Exiba o valor da área calculada usando `print()`.

```
1 # Adicione sua resposta aqui
```

✓ Palavras Reservadas

Em Python, palavras reservadas são identificadores especiais que fazem parte da sintaxe da linguagem e não podem ser usados como nomes de variáveis, funções ou identificadores personalizados. Elas são essenciais para a estrutura do código, representando comandos e instruções como `print`, `if`, `while`, `def`, `return`, entre outras. Como Python é *case-sensitive*, palavras reservadas devem ser usadas **exatamente** como definidas.

```
1 print("Hello")
```

✓ Tipos de dados

Python possui diversos tipos de dados. Os principais são:

- `int` → números inteiros (ex: 10, -25)
- `float` → números de ponto flutuante, decimais (ex: 3.14, -0.75)
- `str` → textos (ex: "Python", 'Curso')
- `bool` → valores lógicos (ex: True, False)

Exemplos no código:

```
1 numero_inteiro = 42
2 numero_decimal = 3.14
3 texto = "Hello, Python!"
4 verdadeiro = True
```

No Python as variáveis são fracamente tipadas. É possível mudar o tipo da variável conforme vai executando o código, mas se reescrever a variável e mudar o tipo dela você pode ter problemas com seu código.

```
1 # Verificar o tipo de dados inteiro (int)
2 type(1)
3 # É possível digitar type para descobrir o tipo do seu dado
```

```
1 # type(5)
```

```
1 # Verificar o tipo de dados float
2 # type(3.14)
```

```
1 # Qual é esse tipo de dados?
2 type(123812983718)
```

```
1 # Qual é esse tipo de dados?
2 type(231231232.189029)
```

```
1 # Verificar o tipo de dados string
2 type("Hello")
```

```
1 # Verificar o tipo de dados booleanos
2 type(False)
```

Exercício 2:

Verifique os valores produzidos pelas seguintes operações (o resultado da operação), e também seus respectivos tipos de dados:

1. `10*2`
2. `2**10`
3. `7 / 3`

3. `1 / 3`
4. `7 // 3`
5. `'bom' + 'dia'`

```
1 # Adicione sua resposta aqui
```

✓ Conversão de Tipos (Type Casting)

Às vezes, é necessário converter um tipo de dado para outro. Python permite conversões com:

- `int()` → Converte para inteiro
- `float()` → Converte para número decimal
- `str()` → Converte para string
- `bool()` → Converte para booleano

```
1 # Verifique o tipo de dado. Observe as aspas
2 idade = "20"
```

```
1 # Converter string para número inteiro (int)
2 idade = "20"
3
```

```
1 # Converter string em float
2 pi = "3.14"
```

```
1 # Verifique o tipo de dados
2 num_produtos_no_estoque = 10_000
```

```
1 # Converter inteiro em string (str)
2 num_produtos_no_estoque = 10_000
```

```
1 # Converter float em inteiro
2 pi = 3.14
3
```

```
1 # Converter inteiro em booleano
2 verdadeiro = 1
3
```

Exercício 3:

Realize as seguintes operações.

1. Converta `15.0` em `int`.

2. Converta 1.234 em str.

3. Converta 35 em float.

```
1 # Adicione sua resposta aqui
```

▼ Função input()

A função `input()` em Python é utilizada para receber dados do usuário durante a execução do programa. Ela permite que o usuário digite um valor, que é retornado como uma string por padrão. Se necessário, o dado pode ser convertido para outros tipos, como inteiro (`int()`) ou decimal (`float()`).

É comum usar `input()` para interações básicas, como solicitar nomes, números ou escolhas em um menu.

```
1 # Desenvolva um programa que pergunte o seu nome
2 nome = input('Qual é o seu nome? ')
3 # Depois de rodar o código, lembre de colocar a informacao pedida e apertar
4 print("Olá,", nome + "!")
```

Exercício 4:

Peça ao usuário para digitar o ano de nascimento e calcule sua idade atual.

Passos:

1. Solicite o ano de nascimento com `input()`.
2. Converta a entrada para um número inteiro.
3. Calcule a idade subtraindo o ano de nascimento do ano atual.
4. Exiba a idade na tela.

```
1 # Adicione sua resposta aqui
```

Exercícios para Praticar:

- **1 Nível Fácil:** Média de três números
 1. Peça ao usuário três números.
 2. Calcule e exiba a média aritmética.
- **2 Nível Médio:** Crie um programa que converta uma temperatura de Celsius para Fahrenheit.
 1. Peça ao usuário a temperatura em graus Celsius com `input()`.
 2. Converta o valor para float, caso o usuário insira números decimais.
 3. Use a fórmula $F = (C \times 9/5) + 32$ para converter.

4. Exiba a temperatura convertida.

- **3 Nível Avançado:** Dias para anos e meses

1. Peça ao usuário um número de dias e converta para anos, meses e dias restantes.

2. Considere 1 ano = 365 dias e 1 mês = 30 dias.

```
1 # Adicione suas respostas aqui
```

✓ Conclusão

Parabéns por concluir esta aula!

🎯 O que aprendemos hoje?

- O que são variáveis e como usá-las.
- Principais tipos de dados em Python.
- Como converter tipos de dados corretamente.
- Como interagir com o usuário de maneira simples.

Agora você já pode criar pequenos programas e manipular dados com Python!

📌 Próximos Passos

- Resolva os problemas na seção "Exercícios para Praticar".
- Caso tenha dúvidas, revise os exemplos e experimente modificá-los. A programação se aprende na prática!

Conteúdo adicional

🎉 Explore o Python Tutor: <https://pythontutor.com/python-compiler.html>

Parabéns pela dedicação! Nos vemos na próxima aula!

Double-click (or enter) to edit

