

# Python

**Ambiente de desenvolvimento,  
dados, operadores e variáveis**

*Prof. Me. Lucio Nunes*  
*Prof. MSc. Rafael Maximo*

# Tópicos

Nesta aula iremos falar sobre:

- Preparação do ambiente de desenvolvimento.
- Conceitos básicos de Python:
  - Constantes ou literais;
  - Operadores aritméticos;
  - Variáveis;
  - Identificadores;
  - Palavras chave (reservadas); e
  - Comentários.
- Python Tutor

# Objetivos

Acompanhe, a seguir, os objetivos de aprendizagem para esta aula:

- Preparar o ambiente para a disciplina;
- Acessar as ferramentas básicas utilizadas na disciplina;
- Conceituar e utilizar tipos de dados primitivos, constantes e variáveis;
- Conhecer os operadores aritméticos;
- Manipular o sistema online Python Tutor.

# Preparação do ambiente

## Como preparar o ambiente computacional para acompanhar as aulas:

- Download e instalação do Python;
- IDLE;
- Interpretador interativo (*Shell*);
- Editor de código-fonte.

# Constantes/Literais

**Constantes em Python são símbolos que representam valores escritos diretamente no código e não podem ser alterados em tempo de execução.**

- ***int***: números inteiros;
- ***float***: representação de números reais;
- ***bool***: também conhecidos como valores lógicos;
- ***string***: cadeias de caracteres.

# Operadores

Operadores são símbolos pré-definidos que realizam uma operação sobre um ou mais operandos, produzindo um valor como resultado.

# Operadores aritméticos

Tabela 1: Operadores aritméticos do Python (continua).

Operador	Descrição	Exemplos
<b>**</b>	Exponenciação, eleva o primeiro operando ao segundo.	$3 ** 2 \rightarrow 9$
<b>-</b> (unário)	Inverte o sinal do operando à direita.	$- (3) \rightarrow -3$ $- (-7) \rightarrow 7$
<b>+</b> (unário)	Mantém o sinal do operando à direita. Observação: é a função identidade.	$+ 3 \rightarrow 3$ $+ (-3) \rightarrow -3$
<b>-</b> (binário)	Subtrai o segundo operando do primeiro.	$15 - 5 \rightarrow 10$ $5 - 15 \rightarrow -10$
<b>+</b> (binário)	Soma o primeiro operando com o segundo. <small>Fonte: Elaborado pelo autor.</small>	$7 + 4 \rightarrow 11$

# Operadores aritméticos

Tabela 1: Operadores aritméticos do Python (conclusão).

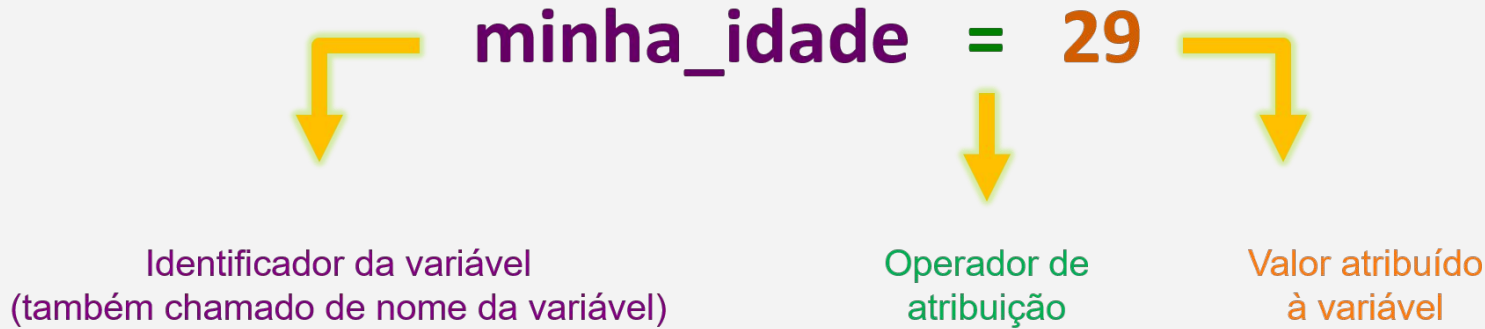
Operador	Descrição	Exemplos
*	Multiplica o primeiro operando pelo segundo.	$3 * 8 \rightarrow 24$
/	Quociente da divisão real do primeiro operando pelo segundo.	$9 / 2 \rightarrow 4.5$ $9.0 / 2 \rightarrow 4.5$
//	Quociente da divisão inteira do primeiro operando pelo segundo.	$9 // 2 \rightarrow 4$ $9.0 // 2 \rightarrow 4.0$
%	Resto da divisão inteira do primeiro operando pelo segundo.	$9 \% 2 \rightarrow 1$ $9.0 \% 2 \rightarrow 1.0$

Fonte: Elaborado pelo autor.



# Variáveis

Figura 1: Esquematização de uso de uma variável.



Fonte: Elaborado pelo autor.

Uma variável é um espaço de memória associado a um identificador, ou seja, um nome, e serve para guardar valores que o programa poderá acessar e modificar.

# Identificadores

Um nome ou identificador de uma variável é formado por uma sequência de um ou mais caracteres, de acordo com as seguintes regras:

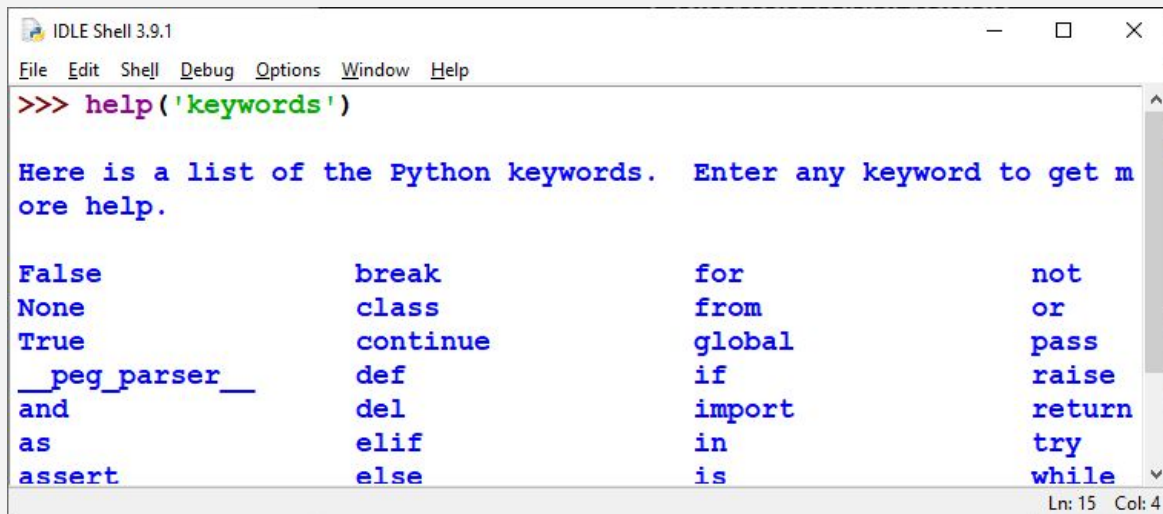
- Pode conter apenas letras, números e o símbolo de sublinhado;
- Não pode começar com dígito;
- Não pode ser uma palavra reservada.

Prefira identificadores concisos, porém significativos para os programadores que lerão seu código.

# Palavras reservadas

O Python possui um conjunto de palavras reservadas, chamadas em inglês de *keywords* (palavras-chave). Não podem ser usadas como identificadores.

Figura 2: Relação parcial das palavras reservadas do Python.



```
IDLE Shell 3.9.1
File Edit Shell Debug Options Window Help
>>> help('keywords')

Here is a list of the Python keywords.  Enter any keyword to get more help.

False      break      for         not
None        class      from        or
True        continue  global      pass
__peg_parser__  def        if          raise
and          del        import      return
as           elif       in          try
assert       else       is          while
```

Fonte: Elaborado pelo autor.

# Comentários

Os comentários são úteis para auxiliar outros programadores (e a nós mesmos) sobre o que está acontecendo no código e, principalmente, a razão.

A linha prefixada com **#** será considerada um comentário e ignorada pelo Python.

```
# Programa feito por: Lucio Nunes  
# Data: 26/02/2021  
# Versão: 1.0  
print('2 + 2 = 4')
```

# Python Tutor

Ambiente gratuito que pode ajudar a entender melhor o que está acontecendo na memória e na saída de dados após a execução de cada instrução do código-fonte.

**Python Tutor**





**Seja um Programador  
(Gustavo Guanabara)**

# Referências

- PYTHON SOFTWARE FOUNDATION. **Lexical analysis.** 2020. Disponível em:  
<[https://docs.python.org/3/reference/lexical\\_analysis.html](https://docs.python.org/3/reference/lexical_analysis.html)>. Acesso em: 21 jan. 2021.
- PYTHON SOFTWARE FOUNDATION. **Expressions.** 2020. Disponível em:  
<<https://docs.python.org/3/reference/expressions.html>>. Acesso em: 21 jan. 2021.
- STURTZ, J. Operators and Expressions in Python. **Real Python,** 2018. Disponível em:  
<<https://realpython.com/python-operators-expressions/>>. Acesso em: 21 jan. 2021.