

# Python: do básico à autonomia intermediária

## aula 2

Bernardo Alves Furtado

February 15, 2022

# Menu do dia

Help and strings

help(), dir()

Objeto, métodos

Instances

Documentation

Exemplo

For loop

for, range

# Continuação última aula

- ▶ **zfill**: `i = 789; s1 = f'{i:08d}'`
- ▶ `'789'.zfill(8)`
- ▶ Instalações?
- ▶ Abriram Pense Python?
- ▶ Exercícios PyCheckio?
  - ▶ `is_even`
  - ▶ `all_upper`

# Continuação continuação última aula

- ▶ right justify 3.1? `exercise3_1.py`
- ▶ Exercício: area triângulo
- ▶ `function7_function.py`

# Objetos em python: ainda funções

► `do_four.py`

# Relembrando

- ▶ `int()`, `str()`, `type()`, `'f{object}'`, `print()`, `input()`
- ▶ Terminal, Console, PyCharm, Conda
- ▶ `//`, `/`, `**`, `%`
- ▶ `if __name__ == '__main__':`
- ▶ `import`
  - ▶ `import math` # Uso: `p = math.pi`
  - ▶ `from math import pi` # Uso: `p = pi`
  - ▶ `import math as m` # Uso: `p = m.pi`

# Relembrando

- ▶ `int()`, `str()`, `type()`, `'f{object}'`, `print()`, `input()`
- ▶ Terminal, Console, PyCharm, Conda
- ▶ `//`, `/`, `**`, `%`
- ▶ `if __name__ == '__main__':`
- ▶ `import`
  - ▶ `import math` # Uso: `p = math.pi`
  - ▶ `from math import pi` # Uso: `p = pi`
  - ▶ `import math as m` # Uso: `p = m.pi`
- ▶ `def`: Escreva uma função que receba três argumentos `b`, `h`, `c` e retorna o volume de um cubo. Qual é o volume do cubo com dimensões (`b x h x c`): `.5`, `.34`, `7.12`?

# console

- ▶ digite `a = 'hello'`
- ▶ `type(a)`
- ▶ `dir(a)`
- ▶ O que acontece?
- ▶ Teste pelo menos 5 métodos `'str'` em `a`
- ▶ Você modificou `a`?



# console

- ▶ digite `a = 'hello'`
- ▶ `type(a)`
- ▶ `dir(a)`
- ▶ O que acontece?
- ▶ Teste pelo menos 5 métodos `'str'` em `a`
- ▶ Você modificou `a`?
- ▶ `'immutable'`

# help!

- ▶ digite `a = 'hello'`
- ▶ `dir(a)`
- ▶ Escolha uma função
- ▶ `help(a.upper)` # Não coloque parênteses depois de `upper`
- ▶ O que o console retorna?

# help!

- ▶ digite `a = 'hello'`
- ▶ `dir(a)`
- ▶ Escolha uma função
- ▶ `help(a.upper)` # Não coloque parênteses depois de `upper`
- ▶ O que o console retorna?
- ▶ E: <https://stackoverflow.com/>, claro

## ponto, parênteses

- ▶ Você notou o uso do **ponto**?
- ▶ `a.upper()`
- ▶ `a.lower()`
- ▶ `a.capitalize()`
- ▶ e do **parênteses**?
- ▶ o que eles significam? qual a diferença?

# instância x classe, bool

- ▶ `a = 'hello'`
- ▶ `b = 'world'`
- ▶ `c = 25`
- ▶ `d = True`

# Relevância

- Comente para você, no futuro!



# Documentation II

- ▶ Revisitando `function5_comentada.py`
- ▶ Triple quotes `""" """`
- ▶ `#`
- ▶ Non-obvious
- ▶ Boas práticas: sentido geral, parâmetros e tipo, retornos e tipo
- ▶ P A U S A – para começar **Loops**

# For, indentation, dois pontos

- ▶ For + iterator + iterable + :
- ▶ indentation: 4 espaços!
- ▶ Tente:  

```
for i in range(5):  
    print(i)
```
- ▶ O que se espera?
- ▶ Qual foi o número da primeira aula?
- ▶ `my_first_loop.py`



# range

- ▶ iterator: "An iterator is an object that contains a countable number of values. An iterator is an object that can be iterated upon, meaning that you can traverse through all the values"<sup>1</sup>
- ▶ alternatively: `for i in [0, 1, 2, 3, 4]:`
- ▶ Exercise: print 'hello world', uma letra por vez.
- ▶ Em maiúsculas?

---

<sup>1</sup>[https://www.w3schools.com/python/python\\_iterators.asp](https://www.w3schools.com/python/python_iterators.asp)

# Exercícios próxima aula

- ▶ <https://py.checkio.org/en/mission/first-word-simplified/>
- ▶ <https://py.checkio.org/en/mission/acceptable-password-i/>