

Python: do básico à autonomia intermediária

aula 4

Bernardo Alves Furtado

February 22, 2022

Menu do dia

Lists

Conditionals

Exercises, exercises, exercises

Exercícios, exercícios ...

Exercícios ultima aula

- ▶ <https://py.checkio.org/en/mission/acceptable-password-i/>
- ▶ Turtles?
- ▶ Exercícios, exercícios, exercícios...

list()

- ▶ Listas são sequências de elementos, quaisquer
- ▶ [0, 2, 3, 5, 6]
- ▶ [0, 1, 'a', 'john', 0.5, 'hello', [0, 2]]
- ▶ Comece com [] ou list()
- ▶ Uma lista vazia: a = list()
- ▶ Listas são mutáveis!
- ▶ Descubra! Quais são os métodos de listas?

`l.append()`, `l.pop()`, `l[:5]`

- ▶ Using `random.randint(0, 10)`, generate a list using `append`
- ▶ `print`
- ▶ `pop` one item
- ▶ `pop` another item
- ▶ O que aconteceu?
- ▶ `insert` one item (`index`, `item`)
- ▶ Qual o elemento na posição 5?
- ▶ Teste!
- ▶ `lists_generator.py`

'traversing'/caminhando the/pela list/lista

- ▶ `for i in range(len(a)):`
- ▶ `print(a[i])`

Slicing com :

- ▶ `a[index]`, `a[5]`
- ▶ `a[start:finish]` último valor não incluso
- ▶ `a[0:5]`, vai até a posição **4**, com tamanho **5**
- ▶ `a[-1]`, o último
- ▶ `a[:]`, do primeiro ao último

Some functions accept lists as parameters

- ▶ `sum(lista a)`
- ▶ `max(lista a)`
- ▶ `len(lista a)`
- ▶ Em outros casos, existe um método do objeto lista
- ▶ `a.sort()`
- ▶ `a.reverse()`
- ▶ Exercício: implemente uma função que calcule a média de uma lista
- ▶ Hint1: use um acumulador para ir somando os valores
- ▶ Hint2: use `len(lista a)`

+ Dicas

- ▶ `a = list()`
- ▶ `b = a`
- ▶ A expressão acima não cria um objeto.
- ▶ São duas referências para o mesmo objeto.
- ▶ Ou seja, quando se alterar a ...
- ▶ Use `a is b` para testar ou `id(a)` e `id(b)`
- ▶ `lists_referencing.py`
- ▶ Strings comportam-se como listas. Mas são imutáveis. Necessário fazer cópia nova.
- ▶ Leia Chapter 10

Frame Title

- ▶ Exercício: insira 'maria' em uma lista.
- ▶ Reverta a ordem
- ▶ print o resultado
- ▶ `list_reverting.py`

Operators

- ▶ `>>> 13 == 13`
- ▶ `True`
- ▶ Operators: `and`, `or`, `not`
- ▶ Logic tables?
- ▶ `True and False and not False or True?`
- ▶ `13 != 14`

if, condição and :

- ▶ Assim como `for`, `if` deve terminar a conditional com `:` e dar continuidade ao código com indentation
- ▶ `if x > 0:`
- ▶ `print('x is positive')`

elif, else and :

- ▶ Se quiser testar condições seguidamente:
- ▶ `if x > 0:`
- ▶ `print('x é positivo')`
- ▶ `elif x == 0:`
- ▶ `print('x é nulo')`
- ▶ `else:`
- ▶ `print('x é negativo')`

elif diferente else

- ▶ Se quiser testar condições seguidamente:
- ▶ `if x > 0:`
- ▶ `print('x é positivo')`
- ▶ `elif x == 0:`
- ▶ `print('x é nulo')`
- ▶ `elif x < 0:`
- ▶ `print('x é negativo')`
- ▶ `conditional1.py`

+ exercícios

Escreva um programa que:

1. ... remova elementos duplicados em uma lista
2. ... que verifica se a lista está vazia ou não
3. ... que clona ou copia uma lista
4. ... que identifique as palavras que são mais compridas do que um valor 'n', a partir de uma lista de palavras
5. Vocês: **"check if a triangle is equilateral, isosceles or scalene"** Sucessivamente, três lados iguais, dois iguais e três diferentes. A entrada contém os tamanhos dos lados do triângulo.

while, break, continue

- ▶ **break:** interrompe o loop em que se encontra e continua após o loop... Ou seja, saí do loop
- ▶ **continue:** continua para o próximo elemento do loop... assim, não executa o que está abaixo do continue!

`exemplos_while_break_continue.py`

Teste seus próprios exemplos!

Hierarquia interromper programa

- ▶ `break`
- ▶ `return`
- ▶ `sys.exit()`

Lista. Em grupos.

1. Escreva um programa que ache os números divisíveis por 7 e por 13, entre o seu ano de nascimento e 2701.
2. Escreva um programa que corre os números de 1 a 50 e imprime. Mas, quando for múltiplo de três, imprima 'Oops', quando for múltiplo de 5 imprima 'Doo', quando for de ambos imprima 'OopsDoo'.
3. Escreva um programa que recebe uma letra e identifica se ela é vogal ou consoante.
4. Escreva um programa que, dada uma lista de números [2, 34, 5, 6, 5, 4, 32] qualquer, retorne: o primeiro valor, o número de valores, o último valor, a soma, a média

SENHA

Faça um programa que...

- ▶ 1. Escolha um número aleatório entre 0 e 100
`random.randint(0, 100)`
- ▶ 2. Escreva uma função que teste o valor aleatório contra um input do usuário
- ▶ 3. Use a expressão `while True`: Ela roda indefinidamente. Porém.
- ▶ Quando a condição `aleatório == input`, termine o programa
- ▶ Você também pode dar ao usuário um número limitado de chances.
- ▶ Lembre-se: `guess = int(input('Entre um valor entre 0 e 100'))`
- ▶ Não se esqueça de ajudar o usuário (maior, menor)
- ▶ `senha.py`

Reddit Python Beginner Showcase

Reddit Begginer's showcase

Exercícios próxima aula

- ▶ <https://py.checkio.org/en/mission/first-word-simplified/>
- ▶ <https://py.checkio.org/en/mission/number-length/>
- ▶ <https://py.checkio.org/en/mission/backward-string/>
- ▶ hint: `lista_a =[::-1]`