

# Python para Modelagem Baseada em Agentes

## aula 4

Furtado, Bernardo Alves

June 3, 2020

# Menu do dia

Lists

Conditionals

Exercícios, exercícios ...

Exercises, exercises, exercises

# 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')`

## + exercícios

1. Write a Python program to remove duplicates from a list
2. Write a Python program to check a list is empty or not
3. Write a Python program to clone or copy a list.
4. Write a Python program to find the list of words that are longer than n from a given list of words
5. Write a Python program to create the multiplication table (from 1 to 10) of a number.
6. Write a Python program to check a triangle is equilateral, isosceles or scalene. An equilateral triangle is a triangle in which all three sides are equal. A scalene triangle is a triangle that has three unequal sides. An isosceles triangle is a triangle with (at least) two equal sides. Input lengths of the triangle sides.

# while, break, continue

`exemplos_while_break_continue.py`  
Teste seus próprios exemplos!

## Exercício: 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 e um input do usuário
- ▶ 3. Use a expressão `while True:` debaixo do `if __name__ == '__main__':` Ela roda indefinidamente. Porém.
- ▶ Quando a condição `aleatório == input`, use:  
`sys.exit()`
- ▶ Lembre-se: `guess = int(input('Entre um valor entre 0 e 100'))`
- ▶ Não se esqueça de ajudar o usuário (maior, menor)