

# Python para Modelagem Baseada em Agentes

## aula 4

Furtado, Bernardo Alves

March 27, 2019

# Menu do dia

Lists

Dictionaries

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

# 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



## + 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

# key, value pair

- ▶ Dicionários são jeitos fáceis do computador guardar (e localizar) informações
- ▶ Funcionam no sistema key (chave), valor (value)
- ▶ `d = dict()`
- ▶ A referência à key é feita como na lista index, por exemplo:
- ▶ `d['key1'] = 258`
- ▶ `d['key23'] = [852, 5]`
- ▶ ou ainda `d = {'nome': 'maria', 'idade': 29}`
- ▶ Façam um dicionário com detalhes da turma. [toy database: age, birthday, affiliation]

## + detalhes

- ▶ Dicionários são NÃO-ORDENADOS
- ▶ diferentemente de listas
- ▶ Exemplo de dicionário como histogramas
- ▶ função `d.get('a', 0)`
- ▶ Default value para keys inexistentes!
- ▶ `dic_histogram.py`
- ▶ `dic_sorted.py`

# Exercício part1

- ▶ Organize os dados dos alunos em um dicionário
- ▶ Comece com
- ▶ `from collections import defaultdict`
- ▶ `my_dict = defaultdict(list)`
- ▶ 1. Entre as informações em listas
- ▶ 2. Os nomes serão as **chaves**?
- ▶ 3. Como vamos guardar 3 infos no dicionário?
- ▶ Hint1. Aproveite que as listas são do mesmo tamanho e utilize um `for loop` que traverse o comprimento das listas

## Exercício part2

- ▶ `Hint2. range(len(listas))`
- ▶ Imprima: chave + informações para cada um
- ▶ Calcule a média das idades
- ▶ `more_dictionaries.py`