

Python: do básico à autonomia intermediária

aula 5

Bernardo Alves Furtado

March 7, 2022

Menu do dia

Fase 1 completa. Exercícios

Dictionaries

try, except

Recursion

is_palindrome: retorne True

Exercícios check

1. Três últimos slides última aula:
 - ▶ Números divisíveis por 7 e 13, período longo: lista, loop, função, parâmetro, script
 - ▶ Oops, OopsDoo (múltiplo 3 e 5). Conditional intermediário
 - ▶ Vogal ou consoante: string, conditional, função, parâmetro, script
 - ▶ Primeiro, ultimo, soma, média: Listas, listas, slicing, built-in functions
2. Programa senha: tudo junto e misturado: input, parâmetros, while loop, conditional message
3. Reddit showcase?
4. py.checkio.org: slice, lista avançado

Basics

- ▶ Dicionários são jeitos fáceis do computador guardar (e localizar) informações
- ▶ Funcionam no sistema key (chave), valor (value)
- ▶ `d = dict()` ou
- ▶ `d = {}`
- ▶ A referência à key é feita como na lista index:
 1. `d['key1'] = 258`
 2. `d['key23'] = [852, 5]`
- ▶ ou ainda `d = {'nome': 'maria', 'idade': 29}`
- ▶ Note: chaves e dois pontos, separados por vírgula
- ▶ [dic_basics.py](#)

+ detalhes

- ▶ Dicionários são NÃO-ORDENADOS (pythons mais novos, eles mantêm a ordem, porém, dicionários não são feitos para lidar com ordem, use listas nesse caso)
- ▶ Exemplo de dicionário como histogramas
- ▶ função `d.get('a', 0)`
- ▶ Default value para keys inexistentes!
- ▶ `dic_histogram.py` – Introduce optional parameters in functions!
- ▶ `dic_sorted.py`

Exercícios com dicionários – breakout rooms?

1. Faça um dicionário com três elementos
2. Adicione uma chave, valor ao dicionário
3. Acesse informações individuais do dicionário, de acordo com a chave
4. Imprima os pares key-value
5. Use a função `update`
6. Verifique se uma chave está presente no dicionário. Use `in`
7. Some os valores de dois dicionários quando tenham a mesma chave

Exercício part1: `more_dictionaries.py` – breakout rooms?

- ▶ 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

Exercise Dictionaries

Teste sua compreensão de dicionários

1. Como gerar um dicionário/base de dados com informações de um empregado (nome, cargo, departamento, data de admissão, salário)? Gere-o
2. Como identificar as 'keys' do dicionário?
3. Como print a data de admissão de um funcionário específico
4. Mude o cargo de um funcionário
5. Incremente o salário por 15% (`d['salario'] *= .15`)

Exercício part2

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

Exercise2. Probability with Python. Rascunho de uma simulação numérica...

Quantos alunos são necessários para que uma turma tenha probabilidade quase 1 de ter dois aniversariantes no mesmo dia? Rodando simulações, chegamos próximos a probabilidades. Por exemplo:

1. Considere o ano com 365 dias.
2. Sorteie dias para aniversários (aumentando o número de alunos) (`random.randint()`)
3. Com os dias dos aniversários em uma lista, verifique se há algum duplicado. (várias possibilidades). Se não conseguir, use `set()`
4. Faça várias vezes (100). Conte quantos houve pelo menos um duplicado.

Exercise1. Nested lists.

Escreva uma função `nested_sum` que aceita uma lista de listas e adiciona os elementos de todas as listas.

A lista de inputs deve ser bem comportada (sempre listas de listas) – > Veja `nested_lists2!`

```
>>> t = [[1, 2], [3], [4, 5, 6]]
```

```
>>> nested_sum(t)
```

```
21
```

Hint1: use a função built-in do Python `sum()`

Hint2: use um contador e loop sobre as nested listas

`is_palindrome: retorne True`

Exercise 4: é palíndromo?

`recursion.py`

1. Escreva uma função que retorne a primeira letra da palavra.
2. Escreva uma que retorne a última
3. Escreva uma função que retorne a string entre a primeira e a última. slicing
4. A função principal, testa a palavra:
Se tem comprimento menor que 2 (primeira e última), retorna True;
Testa se a primeira != última e retorna Falso
Senão, retorna a própria função (recursive)

is_palindrome: retorne True

NestedLists2

Extra1: e no caso de ter que retornar a lista como uma lista

Extra2: e no caso de listas não comportadas? use try, except e recursion

is_palindrome: retorne True

Py.Checkio.Org

Todos as funções do

- ▶ Initiation

- ▶ Home (quase)

- ▶ Mine:

<https://py.checkio.org/en/mission/best-stock/>

is_palindrome: retorne True

Python Challenges

- ▶ Vocês estão aptos a fazer os challenges 1 e 2
- ▶ Hint. Challenge 1
- ▶ Use `from string import ascii_lowercase as letters`
- ▶ `mapping = letters[2:] + letters[:2]`
- ▶ Hint. Challenge 2
- ▶ Use `Inspect Element` no seu navegador
- ▶ Use um dicionário contador para descobrir os itens raros no input

is_palindrome: retorne True

Challenge 3

<http://www.pythonchallenge.com/pc/def/ocr.htm>

Hint. Use o contador de dictionaries que já fizemos em aula.