

Fundamentos de Programação de Computadores
Licenciatura em Sistemas de Informação para Gestão
1º Semestre 2021/2022

Ficha de trabalho 8

Objetivos:

- Desenvolver competências Python usando listas e dicionários

Dicionários

Dicionários são coleções de dados, mas: associam uma **chave** a um **valor**

{chave: valor, chave: valor, chave: valor,}

As chaves de um dicionário são imutáveis enquanto os valores são mutáveis.

Os dicionários têm uma série de **métodos** que permitem a **manipulação dos mesmos**.

<u>Método</u>	<u>Sintaxe</u>	<u>Descrição</u>
keys	<code>mydict.keys()</code>	Retorna as chaves do dicionário em forma de lista
values	<code>mydict.values()</code>	Retorna os valores do dicionário em forma de lista
items	<code>mydict.items()</code>	Retorna os elementos do dicionário em forma de uma lista de tuplos
get	<code>mydict.get(key)</code>	Retorna o valor da key indicada
has_key	<code>mydict.has_key(key)</code>	Verifica se existe a key no dicionário

Tendo em conta estas características dos dicionários resolve os seguintes exercícios utilizando o IDLE ou o Pycharm.

- 1- Dada a seguinte tabela que corresponde ao menu de um snack-bar, crie um dicionário:

Produto	Preço
Salgados	0,60 €
Lanche	1,20 €
Sumo	1,50 €
Refrigerante	1,00 €
Bolo	0,70 €

- 1.1- Se o cliente pedisse um lanche, um bolo e um sumo de laranja qual o preço que ele deveria pagar. (Utiliza o dicionário que criaste para efetuar o cálculo.)
- 1.2- Desenvolve um programa que questione o pedido do cliente, faça o cálculo e calcule o troco a devolver.
- 1.3- Faça a média dos preços dos produtos existentes no snack-bar.

- 2- Escreve uma função `traduz` que recebe como argumentos uma lista de palavras e um dicionário e devolve uma nova lista de palavras traduzidas usando a tradução do dicionário. Caso a lista de palavras contenha uma palavra que não exista no dicionário, essa palavra deverá ficar por traduzir.

```
>>> pt_en = {'hoje' : 'today',
             'esta' : 'is',
             'enevoado' : 'cloudy'}
>>> traduz(['hoje', 'esta', 'muito', 'enevoado'], pt_en)
['today', 'is', 'muito', 'cloudy']
```

Tenta construir a frase completa para mostrar ao utilizador sem ser em formato lista.

- 3- O código Morse associa cada letra do alfabeto a uma sequência de “pontos” e “traços”

A	.-	B	-...	C	-.-.	D	-..	E	.	F	..-.
G	--.	H	I	..	J	.-.-	K	--.-	L	.-..
M	--	N	-.	O	---	P	.-.-	Q	---.-	R	.-.
S	...	T	-	U	...-	V	...-	W	.-.-	X	-. -.
Y	-. -.	Z	--..								

Escreve uma função `morse(txt)` que converte as letras numa sequência de caracteres para Morse; o resultado deve ser uma cadeia com pontos e traços. Usa um espaço para

separar sequências correspondentes às letras. Os caracteres do texto original que não forem letras maiúsculas devem ser ignorados. Exemplos:

```
>>> morse('ABC')  
'._ -... _..'  
>>> morse('ATTACK AT DAWN')  
'._ - _ .- -. -.- -.- .- - ... ._ .-- _.'
```

Sugestão: Começa por definir a tabela de código Morse como o seguinte dicionário:

```
code = {"A": "-.", "B": "-...", "C": "-.-", "D": "-..", "E":
".", "F": "..-.", "G": "--.", "H": "....", "I": "..", "J":
".----", "K": "-.-", "L": "-...", "M": "--", "N": "-.", "O":
"----", "P": "-.-.", "Q": "--.-", "R": "-.", "S": "...",
"T": "-", "U": "..-.", "V": "-.---", "Z": "--.."}

```

Vê o código morse do teu nome 😊

- 4- No campeonato nacional de futebol uma vitória conta 3 pontos, um empate 1 ponto e uma derrota 0 pontos. Em cada jogo, ganha a equipa que marcar mais golos, havendo um empate se o número de golos for o mesmo.
- 4.1- Implemente a função `futebol(scores)` que retorna um dicionário com a pontuação de cada equipa no final da temporada. O parâmetro `scores` é uma lista de dicionários com os resultados da jornada. Cada dicionário tem nome de um clube na chave e o número de golos no valor. Por exemplo,

```
futebol ([{"Vitória SC":2, "Boavista":1}, {"Gil Vicente":1, "Rio Ave":1}, {"Famalicão":3, "Sporting":2}, {"FC Porto":0, "Benfica":0}, {"Tondela":2, "Santa Clara":3}])
```

Deverá retornar o resultado:

```
{'Vitória SC': 3, 'Boavista': 0, 'Gil Vicente': 1, 'Rio Ave': 1,
'Famalicão': 3, 'Sporting': 0, 'FC Porto': 1, 'Benfica': 1,
'Santa Clara': 3, 'Tondela': 0}
```

- 4.2- Acrescenta o código seguinte ao que acabaste de desenvolver e verifica o que acontece

```
res = scores(futebol)
import operator
res = dict( sorted(res.items(), key=operator.itemgetter(1),
reverse=True))
print(res)
```

- 5- Nos anos 50 do século passado, no campeonato de Fórmula 1 a pontuação de cada prova era atribuída com base na posição de cada corredor no final, de acordo com a seguinte tabela:

Posição	Pontos
1	8
2	6
3	4
4	3
5	2

- 5.1- Implemente a função `formula1(scores)` que, dada uma lista com a lista ordenada dos cinco melhor classificados, para cada prova de uma época, retorna um dicionário com a pontuação de cada corredor. Por exemplo,

```
formula1([
    ['Sainz', 'Verstappen', 'Hamilton', 'Ricciardo', 'Massa'],
    ['Bottas', 'Verstappen', 'Raikkonen', 'Stroll', 'Vettel'],
    ['Perez', 'Raikkonen', 'Verstappen', 'Hamilton', 'Vettel']
])
```

deverá retornar

```
{'Sainz': 8, 'Verstappen': 16, 'Hamilton': 7, 'Ricciardo': 3, 'Massa': 2,
 'Bottas': 8, 'Raikkonen': 10, 'Stroll': 3, 'Vettel': 4, 'Perez': 8}
```

- 5.2- Acrescenta o código do exercício 4.2 para obteres o mesmo objetivo.

- 6- Utilizando dicionários e listas, faz o inventário de uma loja de informática. O inventário tem que conter no mínimo 10 produtos. O dicionário deve conter: nome do produto, quantidade e preço.

Sugestão: Se não recordares de material informático pesquisa na internet.

Inventa uma operação a ser realizada sobre a estrutura de dados que criaste.

Bom Trabalho.