<CodeCamp/>
<@FEUP/>

14 fevereiro 2026

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

FEUP
ESCOLAS
Programa de captação
e desenvolvimento de talento

acm
Student Chapter

ni

<CodeCamp/>
<@FEUP/>

14
fevereiro
2026

Sessão 1: O meu primeiro emprego!

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

FEUP
ESCOLAS
Programa de captação
e desenvolvimento de talento.

Apoio

acm
Student Chapter

ni

<Strings/>

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

FEUP
ESCOLAS
Programa de captação
e desenvolvimento de talento.

acm
Student Chapter

ni

Strings

Em Python, strings são representados com ‘’ ou com “”. Strings funcionam como arrays, ou seja podemos obter um carácter da seguinte forma:

```
1  palavra = "Texto"
2  print(palavra[0])
```

Alguns métodos de strings:

len() -> retorna o **tamanho** da string

isupper() -> retorna **true** se **todos** os caracteres forem **maiúsculos**

isdigit() -> retorna **true** se **todos** os caracteres forem **dígitos**

isalpha() -> retorna **true** se **todos** os caracteres forem **alfabéticos**

Pedaços de código:

```
palavra = "Texto"

print(palavra[0]) # imprime o primeiro elemento, ”T”

print(‘T’ in palavra) # retorna True se o carácter estiver na palavra

print(‘u’ not in palavra) # retorna True se o carácter não estiver na palavra

print(palavra.replace("T", "S")) # substitui o ‘T’ por ‘S’- Sexto

print(palavra.replace("Tex", "Pei")) # Peito

print(f"A palavra é {palavra}") # format string torna possível dar print a variáveis diretamente na string

ord(“T”) # retoma o valor ASCII respetivo do carácter (84)
```

Houston, we have a problem!

O teu colega fez asneira e agora temos os nossos dados cheios de caracteres estranhos e dígitos em sítios que não deviam estar. E agora? Escreve a função **limpar_dados** que resolve este bicho de sete cabeças e salva o dia de trabalho! Ah e, quanto ao colega… não sei se há algo que o salve!

Dicas: Os dados corretos são apenas letras.

Exemplo:

dado corrompido: G1ir0o!Rápid0o

dado original: Giro Rápido

Tabela ASCII

a	97	0110 0001
b	98	0110 0010
c	99	0110 0011
d	100	0110 0100
e	101	0110 0101
f	102	0110 0110
g	103	0110 0111
h	104	0110 1000
i	105	0110 1001
j	106	0110 1010
k	107	0110 1011
l	108	0110 1100
m	109	0110 1101
n	110	0110 1110
o	111	0110 1111
p	112	0111 0000
q	113	0111 0001
r	114	0111 0010
s	115	0111 0011
t	116	0111 0100
u	117	0111 0101
v	118	0111 0110
w	119	0111 0111
x	120	0111 1000
y	121	0111 1001
z	122	0111 1010

A	65	0100 0001
B	66	0100 0010
C	67	0100 0011
D	68	0100 0100
E	69	0100 0101
F	70	0100 0110
G	71	0100 0111
H	72	0100 1000
I	73	0100 1001
J	74	0100 1010
K	75	0100 1011
L	76	0100 1100
M	77	0100 1101
N	78	0100 1110
O	79	0100 1111
P	80	0101 0000
Q	81	0101 0001
R	82	0101 0010
S	83	0101 0011
T	84	0101 0100
U	85	0101 0101
V	86	0101 0110
W	87	0101 0111
X	88	0101 1000
Y	89	0101 1001
Z	90	0101 1010

Lembras-te do que aconteceu no Louvre?

O CEO não quer que isso aconteça aqui. Por isso, precisas de criar um validador que verifique que as passwords dos clientes cumprem os requisitos de segurança da empresa. Terás de criar uma função, **validar**, que recebe uma lista de potenciais passwords e retorna apenas aquelas que são válidas.

Uma password válida tem de seguir os seguintes requisitos:

- Mínimo de 8 caracteres;
- Pelo menos uma letra maiúscula;
- Pelo menos um número;
- Pelo menos um destes caracteres especiais, **!**, **@**, **#** ou **$**.

<Listas/>

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

FEUP
ESCOLAS
Programa de captação
e desenvolvimento de talento.

acm
Student Chapter

ni

Listas

Como nas strings, consegues aceder aos itens através da indexação, ou seja, o seguinte código imprime também o primeiro elemento da lista:

```
1   lista = [1, 2, 3, 5]
2   print(lista[0])  # 1
3
```

Alguns métodos de listas:

len(lista**)** -> retorna o **tamanho** da lista

lista**.append(**elemento**)** -> **adiciona** um elemento ao **fim** da lista

lista**.insert(**elemento**)** -> **adiciona** um elemento à **posição especificada**

lista**.sort()** -> **ordena** a lista

Pedaços de código:

```
lista = [1,2,3,5]

print(lista[0]) # imprime o primeiro elemento da lista, 1

lista.insert(1,10) # insere o elemento na posição 2, [1,10,2,3,5]

lista.sort() # ordena de forma crescente

lista.sort(reverse=True) # ordena de forma decrescente

print(len(lista)) # imprime o tamanho da lista, 5

def função(n):
    return n*n

lista.sort(key=função) # ordena usando uma função

lista[2] = 9 # modifica o elemento na terceira posição para 9, [1,2,9,5,10]

del lista[2] # remove o elemento na terceira posição, [1,2,5,10]
```

Uma equipa está prestes a começar um novo projeto, mas primeiro precisamos de garantir que temos todos os materiais necessários! Terás de: **verificar** se todos os materiais pedidos estão disponíveis, **fazer** uma lista do que falta e **identificar** que materiais extra temos, mas que não foram pedidos.

Exemplo:

Inventário- [fita cola,agrafador, canetas, papel]

Preciso- [fita cola,agrafador, canetas, papel, computadores, impressora]

<Tuplos/>

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

FEUP
ESCOLAS
Programa de captação
e desenvolvimento de talento.

acm
Student Chapter

ni

Tuplos

Assim como nas Listas ou Strings são indexáveis, isto é, podes aceder a um item específico usando a sua posição (índice). A maior diferença é que, depois de criar um Tuplo, não pode alterar, adicionar ou remover os seus elementos. Fica fixo!

Se realmente precisares de alterar o tuplo, converte-o numa lista temporariamente, faz a alteração, e depois converte-o de volta para um tuplo.

Pedaços de código:

```
tuplo = (1,23,4)
print(tuplo[0]) # primeiro elemento do tuplo, 1
y = list(tuplo) # converte o tuplo numa lista para poder trocar os valores
y.append(7)
tuplo= tuple(y)
print(tuplo) # adicionamos o 7 ao final do tuplo, passa a ser (1,23,4,7)
```

(com este método de conversão para lista podemos aplicar os mesmos métodos das listas aprendidos anteriormente)

```
(Primeiro,Segundo,Terceiro,Quarto) = tuplo
print(Primeiro) # imprime 1
tuplo = (Segundo,Terceiro,Quarto, Primeiro)
print(tuplo) # altera a ordem dos elementos do tuplo, (23, 4, 7, 1)
```

Os Recursos Humanos querem a tua ajuda, pois necessitam de realizar uma avaliação interna, mas não sabem automatizar a ordenação dos funcionários de acordo com a sua avaliação. Sabendo que recebes uma **lista** com os **tuplos** (Nome, Qualidade) **ordena os funcionários em ordem crescente**.

Exemplo:

Recebido- [(‘Ana’,3),(‘Pedro’,4),(‘José’,1)]

Resultado- [‘José’,’Ana’,’Pedro’]

<Exercícios/>

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

FEUP
ESCOLAS
Programa de captação
e desenvolvimento de talento.

acm
Student Chapter

ni

O teu supervisor pediu-te para criares uma lista com todos os funcionários da empresa, enviando-te a lista dos funcionários de cada departamento. No entanto, há funcionários que estão em mais do que um departamento. Remove duplicados!

Exemplo:

Recebido- [['Inês', 'Afonso','Sónia','Miguel'],['Samuel', 'Carla', 'Afonso', 'António'],['Ana', 'Inês', 'Abel']]

Resultado- ['Inês', 'Afonso', 'Sónia', 'Miguel', 'Samuel', 'Carla', 'António', 'Ana', 'Abel']

Os recursos humanos precisam da tua ajuda a criar os emails corporativos para os novos funcionários!

Através de uma lista de nomes gera emails com o primeiro e último nome do funcionário (ex.: anameireles@corporate.com).

Novos Funcionários: ["Ana Faria Meireles", "Pedro José Vidal", "Manuel Silva", "Diogo Moreira Vaz"]

Emails: ['anameireles@corporate.com', 'pedrovidal@corporate.com', 'manuelsilva@corporate.com', 'diogovaz@corporate.com']

A empresa quer começar a fazer vendas para a Inglaterra. Para tal, é preciso converter os preços de € para £ (1£ equivale a 1.15€).

Exemplo:

Preços em euro: [(Pc:900),(Telemóvel:500),(Aspirador:900)]

Preços em Libras: [(Pc:782.61),(Telemóvel:434.78),(Aspirador:782.61)]