



GUIA DE LABORATÓRIO 3.4 - EXTRA RESOLUÇÃO DOS EXERCÍCIOS DE REVISÃO

EXERCÍCIOS DE REVISÃO

1.

```
.__next__ (para podermos invocar next(gerador))  
.__iter__ (para podermos invocar iter(gerador))  
.send (para podermos invocar gerador.send(valor))  
.close (para podermos invocar gerador.close())  
.throw (para podermos invocar gerador.throw(TipoExcecao))
```

2. São um forma sucinta de criar listas e geradores. Exemplo(s):

```
>>> squares = []  
>>> for x in range(10):  
...     squares.append(x**2)  
...  
>>> squares  
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

Com uma expressão lista:

```
squares = [x**2 for x in range(10)]
```

3.

<p>a) Dada a seguinte lista de números:</p> <pre>nums = [19, 35, 9, 10, 20, 17, 12, 22, 10]</pre> <p>pretende filtrar para uma lista aqueles que são >= 15</p>	<pre>maiores_que_15 = [] for num in nums: if num >= 15: maiores_que_15.append(num)</pre>	<pre>[num for num in nums if num >= 15]</pre>
---	---	--

<p>b) Dada a seguinte string</p> <pre>txt = 'ALBERTO'</pre> <p>pretende obter o inverso desta string.</p> <p>(não pode utilizar <code>reversed</code>)</p>	<pre>txt2 = [] for i in range(len(txt)-1, -1, -1): txt2.append(txt[i]) txt2 = ''.join(txt2)</pre>	<p>Ver à frente...</p>
<p>c) Dada uma determinada string, pretende determinar se todas as vogais minúsculas aparecem.</p>	<pre>for ch in set(txt): if ch in 'aeiou': count += 1 print(count == 5) --- len(set(txt) & {'a', 'e', 'i', 'o', 'u'}) == 5 OU len(set(txt) & set('aeiou')) == 5 --- not ({'a', 'e', 'i', 'o', 'u'} - set(txt)) OU not (set('aeiou') - set(txt))</pre>	<p>Ver à frente...</p>

2a linha, 2a coluna: `''.join([txt[i] for i in range(-1, -len(txt)-1, -1)])` ou

`''.join(txt[i] for i in range(-1, -len(txt)-1, -1))`

3a linha, 2a coluna: `len(set(ch for ch in txt if ch in 'aeiou')) == 5`

Ver ainda ficheiro com nome parecido com `expressoes_lista_geradoras.py` como complemento deste exercício.

4.

<pre>cores = ['vermelho', 'verde', 'vermelho', 'azul', 'verde', 'vermelho'] d = {} for cor in cores: if cor not in d: d[cor] = 0 d[cor] += 1 print(d)</pre>	<pre>{ 'vermelho': 3, 'verde': 2, 'azul': 1 }</pre>
---	---

<pre> nomes = ('Alberto Antunes', 'Armando Alves', 'Alberto Alves', 'Antonio Almeida', 'Alberto Alexandrino', 'Arnaldo Afonso') x = (n.split('Alberto')[-1] for n in nomes if n.startswith('Alberto')) for a in x: print(a, end='/')</pre>	<p><i>Antunes/ Alves/ Alexandrino/</i></p>
<pre> palavras = ['mesa', 'carro', 'mesa', 'garfo', 'carro', 'mesa', 'bola', 'pá', 'garfo'] d = {} for palavra in palavras: d[palavra] = d.get(palavra, 0) + 1 print(*d, sep=',', end='') print(*d.values(), sep=',')</pre>	<p><i>pá,garfo,bola,mesa,carro,carro=1,2,3,2,1</i></p>
<pre> soma = 0 with open('extra.txt') as fich: for linha in fich: try: partes = linha.split() soma += float(partes[1]) except ValueError: print("ERRO:", linha, end='') else: print(partes[0], '->', partes[1]) print(soma) NOTA: Assuma que o ficheiro extra.txt possui: Alberto 19 Alberto 19 Alberto 19 Armando 24 ... e depois ... Armando x24 ... e depois ... Armando António 150 António 150 António 150</pre>	<p><u>1o Ficheiro:</u></p> <p>Alberto -> 19</p> <p>Armando -> 24</p> <p>António -> 150</p> <p>193</p> <p>--</p> <p><u>2o Ficheiro:</u></p> <p>Alberto -> 19</p> <p>ERRO: Armando x24António -> 150</p> <p>169</p> <p>--</p> <p><u>3o Ficheiro:</u></p> <p>Alberto -> 19</p> <p>Traceback (most recent call last):</p> <p>File "<stdin>", line 1, in <module></p> <p>IndexError: list index out of range</p>

5.

1a solução:

```
d = {}  
for nome in nomes:  
    key = len(nome)  
    if key not in d:  
        d[key] = []  
    d[key].append(nome)
```

2a solução:

```
d = {}  
for nome in nomes:  
    key = len(nome)  
    d.setdefault(key, []).append(nome)
```