

Errata* do livro *Programação em Python – Fundamentos e Resolução de Problemas*

*Nota importante: Algumas correções já foram consideradas na reimpressão de outubro de 2016.

* pg. 32, listagem 1.3

Problema de alinhamento do código:

```
def raizquad(a):
    """Cálculo da raiz quadrada de um número positivo pelo método de Newton."""
    print(("Raiz quadrada de: ",a))
    x = eval(input("Valor inicial sff: "))
    for i in range(10):
        x = (1/2) * (x + (a/x))
    return x

if __name__ == '__main__':
    print(raizquad(2))
```

pg.71, secção 3.2.3

Em vez de “ O Quadro 3.3” deve estar “0 Quadro 3.2”.

pg.73, listagem de código (linhas 4 até 11)

Na listagem de código no início da página onde se lê ">>> decimalz.Decimal(1953)..." (linha 6) deveria ler-se ">>> decimal.Decimal(1953)...", sem o “z” no final da palavra decimal.

Onde se lê "Este pequeno exemplo mostra..." deveria ler-se “Este pequeno exemplo mostra como se pode ter precisão ilimitada”.¹

¹Esta informação é verdadeira se os métodos usados forem os métodos do tipo decimal.Decimal.

pg.85, listagem de código (a partir da linha 14 até linha 23)

As plicas estão mal colocadas. O que deve estar:

```
>>> nome = 'Ernesto J. F. Costa'
>>> nome.center(50)
'                Ernesto J. F. Costa                '
>>> titulo = nome.center(30)
```

```
>>> titulo
'    Ernesto J. F. Costa    '
>>> titulo.strip()
'Ernesto J. F. Costa'
>>> titulo.upper()
'    ERNESTO J. F. COSTA    '
```

pg.128, primeira listagem de código

Nas duas menções de "sys.sdout" deveria ler-se "sys.stdout", a palavra é "stdout" e não "sdout".

pg.150, última listagem de código

No final da página, na caixa de código, onde se lê "<instrução>" deveria ler-se "<instrução>", com indentação, que é obrigatória em Python:

```
for <nome> in <iterável>:
    <instrução>
```

pg.151, última listagem de código

No final da página, na caixa de código, onde se lê "<instruções>" deveria ler-se "<instruções>", com indentação, que é obrigatória em Python:

```
while <condição>:
    <instruções>
```

pg.164, listagem 5.25

Retirar continue:

```
def password(lista_passw):
    """Três tentativas para introduzir corretamente uma password """
    conta=3
    while conta:
        codigo=input("Insira o seu código sff:")
        if codigo in lista_passw:
            print "Bem-vindo"
            break
        print ("Código errado.")
        conta= conta - 1
```

```
else:
    print( "Acabaram as suas tentativas!!!")
```

pg.165, última listagem

Retirar ":", linha 3:

```
def nada(x):
    if cond1(x):
        f1(x)
    elif cond2(x):
        pass
    else:
        fn(x)
```

pg.169

Na nota de rodapé 15, onde se lê "...permanente no disco externo..." deveria ler-se "...permanente em memória ...".

pg.238, listagem 7.7

Incluir a instrução `plt.title(tit)` em mostra:

```
import matplotlib.pyplot
plt = matplotlib.pyplot

def ler(ficheiro):
    with open(ficheiro, 'r') as f_ent:
        dados_car = f_ent.read().split()
        dados = []
        for elem in dados_car:
            dados.append(float(elem))
    return dados

def mostra(xetiq, yetiq, tit, x, y):
    plt.xlabel(xetiq)
    plt.ylabel(yetiq)
    plt.title(tit)
    plt.plot(x, y)

def main(ficheiro):
```

```

dados = ler(ficheiro)
mostra('Meses','Temperatura','',range(1,13),dados)
plt.show()

if __name__ == '__main__':
    main('/data/dados_simples.txt')

```

pg.239, última listagem
Retirar no final f_ent_close():

```

import matplotlib.pyplot
plt = matplotlib.pyplot

def le_todas_temperaturas(fich):
    """
    Extrai os dados de temperaturas relativos a Portugal.
    """
    with open(fich,'r') as f_ent:
        portugal = list()
        dados = le_uma_temperatura(f_ent)
        while dados != -1:
            portugal.append(dados)
            dados = le_uma_temperatura(f_ent)
    return portugal

```

pp. 240-241, listagem 7.10
Retirar f_ent_close()(linha 11) e indentar corretamente plt.plot(cidade)
(linha 29):

```

import matplotlib.pyplot
plt = matplotlib.pyplot

def le_todas_temperaturas(fich):
    """
    Extrai os dados de temperaturas relativos a Portugal.
    """
    with open(fich,'r') as f_ent:
        portugal = list()
        dados = le_uma_temperatura(f_ent)
        while dados != -1:
            portugal.append(dados)

```

```

        dados = le_uma_temperatura(f_ent)
    return portugal

def le_uma_temperatura(f_ent):
    """
    Ler dados da temperatura de uma cidade.
    Devolve -1 se fim de ficheiro
    """
    linha = f_ent.readline()
    while (linha != '') and (linha == '\n'):
        linha = f_ent.readline()
    if linha == '':
        return -1
    else:
        linha = linha[:-1].split()
        return [float(dado) for dado in linha]

def mostra_todas(xetiq, yetiq, tit, dados):
    plt.xlabel(xetiq)
    plt.ylabel(yetiq)
    plt.title(tit)
    for cidade in dados:
        plt.plot(cidade)

def main(ficheiro):
    dados = le_todas_temperaturas(ficheiro)
    mostra_todas('Meses', 'Temperatura', 'Temperaturas Médias das Cidades'
    plt.show()

if __name__ == '__main__':
    main('/data/temperaturas.txt')

```

pg.241

A Figura 7.4 está trocada com a Figura 7.6 da página 245.

pg.243

No 1.º parágrafo (2.ª linha), onde se lê "dozes" deveria ler-se "doze".

pg.247, segunda listagem de código

Retirar `fich_close()`(linha 10), indentar corretamente a linha 9:

```

import csv

def le_csv(nome_fich):
    """ Lê um ficheiro em formato csv."""
    with open(nome_fich) as fich:
        csv_reader = csv.reader(fich)
        dados = []
        for linha in csv_reader:
            dados.append(linha)
        return dados

```

pg.248, primeira e terceira listagem (linha 8 e linha 6, respetivamente)
Eliminar nome_fich.close()

```

import csv

def insere_linha_csv(fich , linha):
    """ Insere uma linha no fim do ficheiro."""
    with open(fich , 'a') as nome_fich:
        csv_writer = csv.writer(nome_fich)
        csv_writer.writerow(linha)

e csv_fich.close()

def escreve_csv(fich , dados, delimitador):
    """ Escreve um ficheiro em formato csv."""
    with open(fich , 'w') as csv_fich:
        csv_writer = csv.writer(csv_fich , delimiter=delimitador)
        csv_writer.writerows(dados)

```

pg.249, primeiro parágrafo

Onde se lê "(linhas 3 a 5)" deveria ler-se "(linhas 3 e 4)".
 Onde se lê "(linhas 6 a 11)" deveria ler-se "(linhas 6 a 9)".
 Onde se lê "(linha 13)" deveria ler-se "(linha 12)".

pg.253, último parágrafo

Onde se lê "...o original não pode estar em formato de texto, mas antes binário." deveria ler-se "...o original não pode estar em formato de texto, mas antes binário com o

Python até à versão 3.3. A partir dessa versão, inclusive já é aceite o formato de texto.”

pg.256, nota de rodapé 8

Como o *link* referido foi retirado e, portanto, não está acessível, deverá fazer o *download* do ficheiro com código python no *site* da FCA em <https://www.fca.pt/pt/catalogo/informatica/programacao/programacao-em-python/>

*** pg. 421, primeira fórmula**
Corrigir a 2.ª linha:

$$M(n) = 2 * M(n - 1) + 1$$

*** pg. 488, problema 12.20**
Retirar a expressão final “de procura”:

... numa dada árvore binária.

*** pg. 524, primeira listagem.**
Problema de alinhamento a seguir a “for...”. Substituir por:

```
class Predador(Robot):
```

```
    def __init__(self, nome, mundo, pos_x=0, pos_y=0, energia=100):
        super().__init__(nome, mundo, pos_x, pos_y)
        self._energia = energia
```

```
    def come(self):
        vizinhos = [(0,-1),(1,0),(0,1),(-1,0)]
        x,y = self.obtem_posicao()
        for d_x, d_y in vizinhos:
            n_x = x + d_x
            n_y = y + d_y
            tamanho = self._mundo.obtem_tamanho()
            if (0 <= n_x < tamanho) and (0 <= n_y < tamanho):
                conteudo = self._mundo.obtem_conteudo(n_x,n_y)
                if isinstance(conteudo,Presas):
                    self._mundo.limpa_registo(x,y)
                    self.define_posicao(n_x,n_y)
                    self._mundo.regista(self)
                    self._energia += conteudo.obtem_energia()
```

```
        break
    else:
        continue
else:
    print('Impossível comer.')
```

```
class Presa(Robot):
    def __init__(self, nome, mundo, pos_x=0, pos_y=0, energia=100):
        super().__init__(nome, mundo, pos_x, pos_y)
        self._energia = energia
```

*** pg. 526, primeira linha de texto**
Corrigir a gralha:

Com esta situação ao chamarmos ...

*** pg. 556, listagem de código**
A linha 5:
dados = texto.get('0.0', END)

deve estar alinhada com a linha seguinte.

*** pg. 559, primeira listagem de código**
O código da primeira listagem na linha 6:

```
super().__init__()
```

deve estar alinhado com o código da linha 7. (Ver, p.f., segundo exemplo da mesma página que está correto)