

AT



Engenharia da Computação

Fundamentos Python

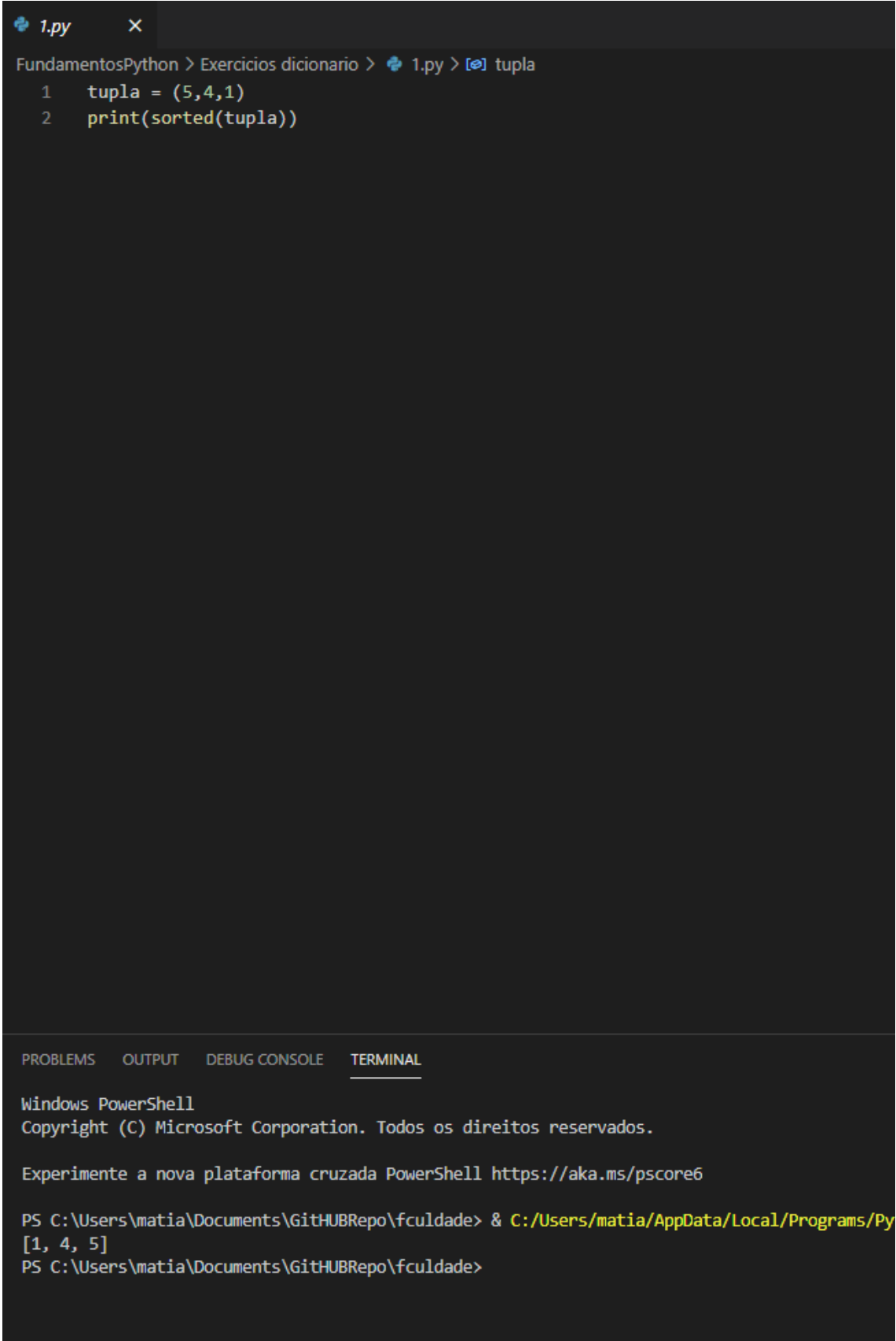
Matias Reinke

Prof^ª.: Thais do Nascimento

Correntina BA

8 de Dezembro de 2020

1)



The image shows a Visual Studio Code editor window with a single file named `1.py` open. The file contains two lines of Python code: `tupla = (5,4,1)` and `print(sorted(tupla))`. The file explorer on the left shows the file is located at `FundamentosPython > Exercicios dicionario > 1.py`. The bottom panel displays the `TERMINAL` tab, which shows the output of running the script: `[1, 4, 5]`. The terminal also displays standard Windows PowerShell startup messages and the current directory path `C:\Users\matia\Documents\GitHubRepo\fculdade`.

```
1.py ×
FundamentosPython > Exercicios dicionario > 1.py > [?] tupla
1  tupla = (5,4,1)
2  print(sorted(tupla))

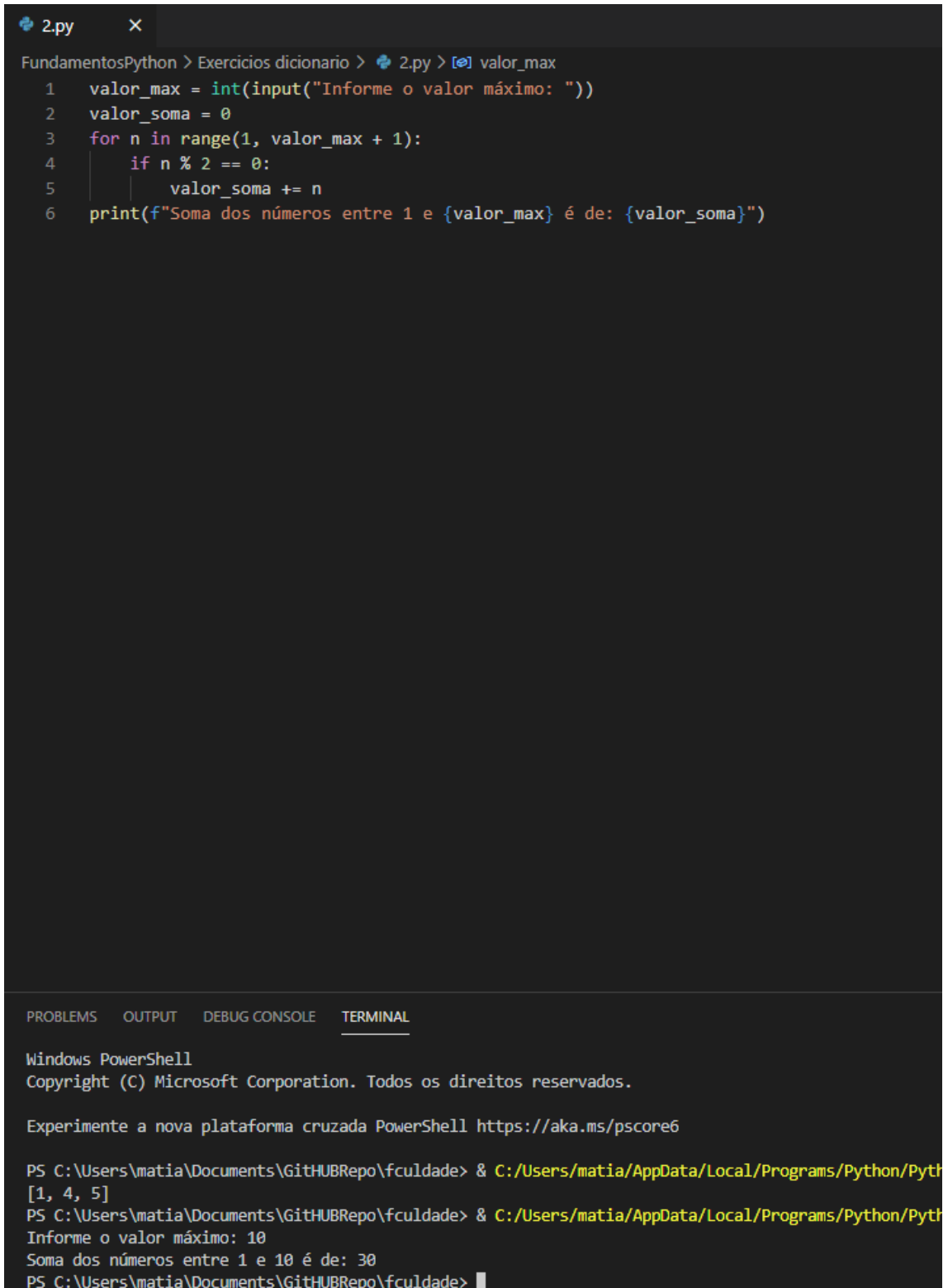
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Experimente a nova plataforma cruzada PowerShell https://aka.ms/pscore6

PS C:\Users\matia\Documents\GitHubRepo\fculdade> & C:/Users/matia/AppData/Local/Programs/Py
[1, 4, 5]
PS C:\Users\matia\Documents\GitHubRepo\fculdade>
```

2)



The image shows a VS Code editor window with a file named `2.py`. The code is a Python script that calculates the sum of even numbers from 1 to a user-defined maximum value. The script is as follows:

```
FundamentosPython > Exercicios dicionario > 2.py > [?] valor_max
1  valor_max = int(input("Informe o valor máximo: "))
2  valor_soma = 0
3  for n in range(1, valor_max + 1):
4      if n % 2 == 0:
5          valor_soma += n
6  print(f"Soma dos números entre 1 e {valor_max} é de: {valor_soma}")
```

Below the editor, the **TERMINAL** tab is active, showing the execution of the script in a Windows PowerShell environment. The output shows that the user entered 10 as the maximum value, and the program correctly calculated the sum of even numbers from 1 to 10, which is 30.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Experimente a nova plataforma cruzada PowerShell https://aka.ms/pscore6

PS C:\Users\matia\Documents\GithUBRepo\fculdade> & C:/Users/matia/AppData/Local/Programs/Python/Python39-6/python.exe C:/Users/matia/AppData/Local/Programs/Python/Python39-6/Scripts/python.exe 2.py
[1, 4, 5]
PS C:\Users\matia\Documents\GithUBRepo\fculdade> & C:/Users/matia/AppData/Local/Programs/Python/Python39-6/python.exe C:/Users/matia/AppData/Local/Programs/Python/Python39-6/Scripts/python.exe 2.py
Informe o valor máximo: 10
Soma dos números entre 1 e 10 é de: 30
PS C:\Users\matia\Documents\GithUBRepo\fculdade>
```

3)

```
3.py x
FundamentosPython > Exercicios dicionario > 3.py > potencia
1 def potencia():
2     base = int(input("Qual o valor da base? "))
3     while base < 0 :
4         base = int(input("Valor inválido, qual o valor da base(maior que zero)? "))
5     expoente = int(input("Qual o valor do expoente? "))
6     while expoente < 0 :
7         expoente = int(input("valor inválido, qual o valor do expoente(maior que zero)? "))
8     resultado = base
9     for n in range(1, expoente):
10        resultado *= base
11    if expoente == 0 :
12        print(f"{base} elevado na {expoente} dá: 1")
13    else:
14        print(f"{base} elevado na {expoente} dá: {resultado}")
15
16 potencia()
```

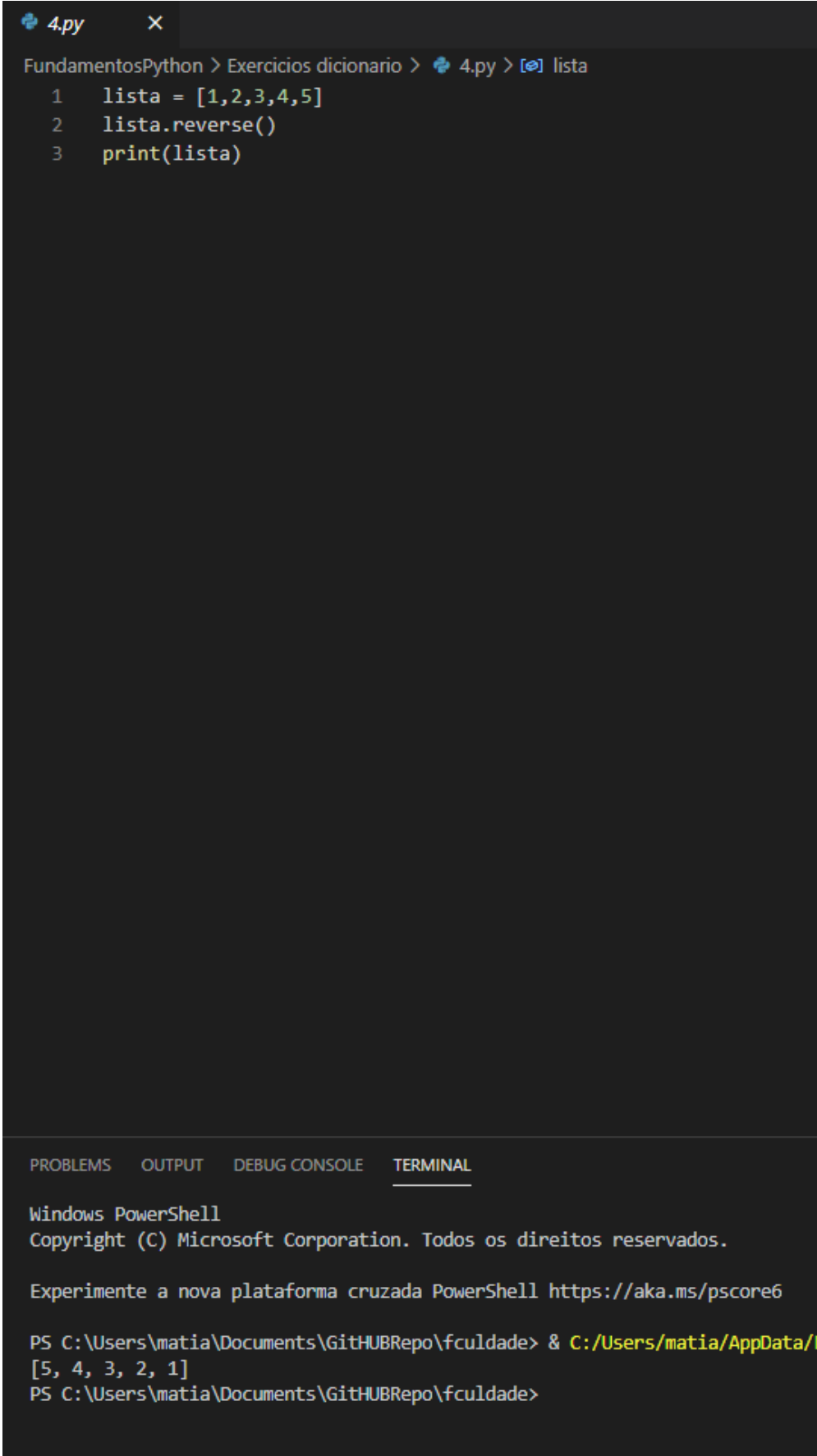
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Experimente a nova plataforma cruzada PowerShell <https://aka.ms/pscore6>

PS C:\Users\matia\Documents\GitHubRepo\fculdade> & C:/Users/matia/AppData/Local/Programs/Python/Python39/python
Qual o valor da base? 5
Qual o valor do expoente? 2
5 elevado na 2 dá: 25
PS C:\Users\matia\Documents\GitHubRepo\fculdade> |

4)



The image shows a Visual Studio Code editor window with a file named `4.py` open. The file contains three lines of Python code: `lista = [1,2,3,4,5]`, `lista.reverse()`, and `print(lista)`. The file explorer on the left shows the path `FundamentosPython > Exercicios dicionario > 4.py`. Below the editor, the `TERMINAL` tab is active, displaying the output of the script: `[5, 4, 3, 2, 1]`. The terminal also shows the Windows PowerShell prompt and the current directory `C:\Users\matia\Documents\GitHUBRepo\fculdade`.

```
4.py X
FundamentosPython > Exercicios dicionario > 4.py > [E] lista
1 lista = [1,2,3,4,5]
2 lista.reverse()
3 print(lista)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Experimente a nova plataforma cruzada PowerShell https://aka.ms/pscore6

PS C:\Users\matia\Documents\GitHUBRepo\fculdade> & C:/Users/matia/AppData/Local/Programs/Python/Python39-64/Python.exe C:\Users\matia\Documents\GitHUBRepo\fculdade\4.py
[5, 4, 3, 2, 1]
PS C:\Users\matia\Documents\GitHUBRepo\fculdade>
```

6)

```
6.py X
FundamentosPython > Exercicios dicionario > 6.py > [?] numeros
1  numeros = (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,30,27,23,45,28)
2  def par_impar(numeros):
3      pares = []
4      impares = []
5      for index, n in enumerate(numeros):
6          if index % 2 == 0:
7              pares.append(n)
8          if n % 2 != 0:
9              impares.append(n)
10     pares = tuple(pares)
11     print(impares, pares)
12     par_impar(numeros)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

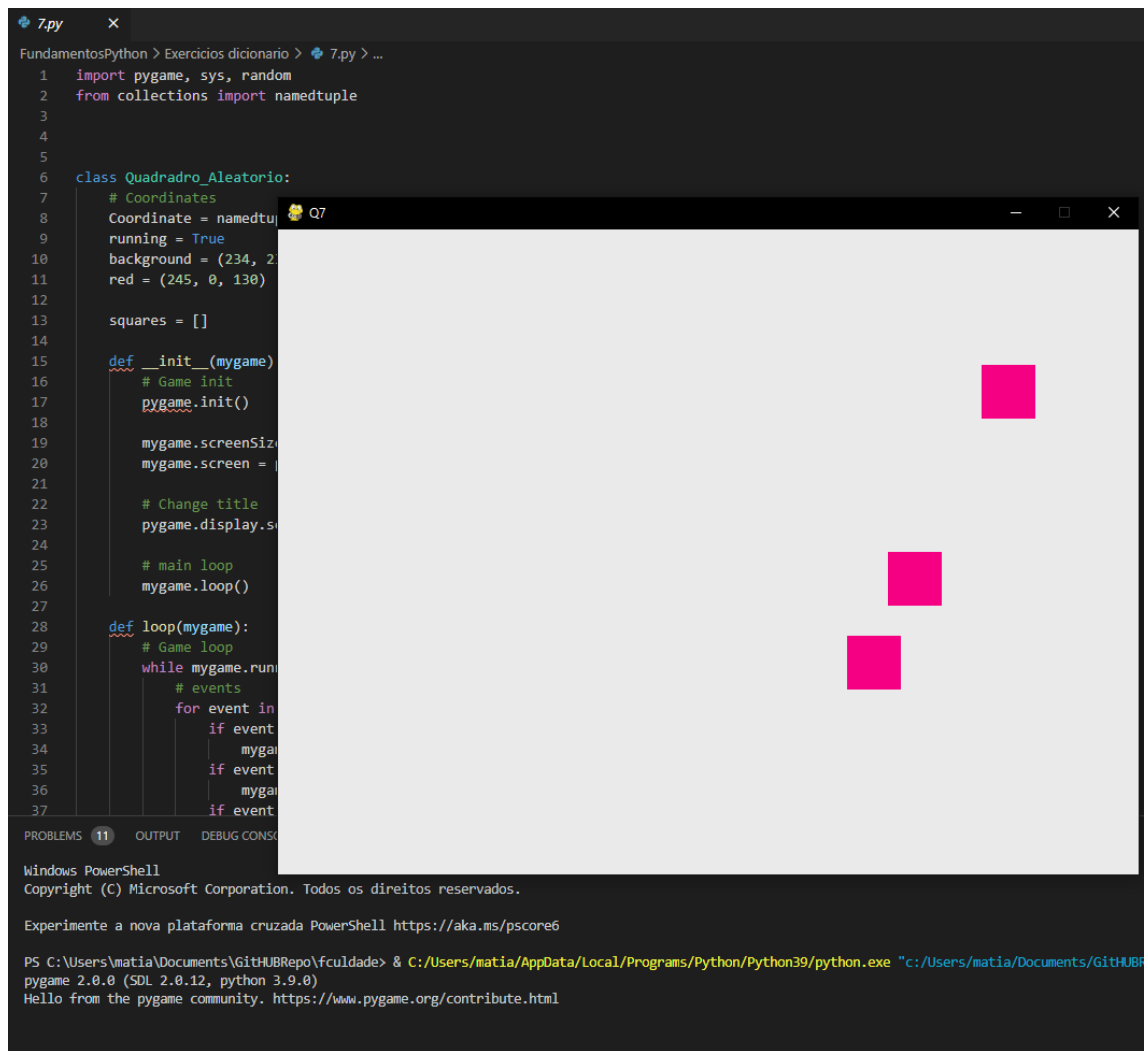
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Experimente a nova plataforma cruzada PowerShell <https://aka.ms/pscore6>

PS C:\Users\matia\Documents\GitHUBRepo\fculdade> & C:/Users/matia/AppData/Local/Programs/Python/Python39-64/Python.exe C:\Users\matia\Documents\GitHUBRepo\fculdade\6.py [1, 3, 5, 7, 9, 11, 13, 15, 27, 23, 45] (1, 3, 5, 7, 9, 11, 13, 15, 30, 23, 28)

PS C:\Users\matia\Documents\GitHUBRepo\fculdade>

7)



The screenshot shows a Python IDE with a file named `7.py`. The code defines a class `Quadrado_Aleatorio` and a function `loop` to create a Pygame window and draw three red squares at random positions. The window is titled `Q7` and displays three red squares on a light gray background. The IDE interface includes a file explorer, a code editor, and a terminal window at the bottom.

```

1 import pygame, sys, random
2 from collections import namedtuple
3
4
5
6 class Quadrado_Aleatorio:
7     # Coordinates
8     Coordinate = namedtuple('Coordinate', 'x y')
9     running = True
10    background = (234, 234, 234)
11    red = (245, 0, 130)
12
13    squares = []
14
15    def __init__(mygame):
16        # Game init
17        pygame.init()
18
19        mygame.screenSize = 400
20        mygame.screen = pygame.display.set_mode(mygame.screenSize)
21
22        # Change title
23        pygame.display.set_caption('Q7')
24
25        # main loop
26        mygame.loop()
27
28    def loop(mygame):
29        # Game loop
30        while mygame.running:
31            # events
32            for event in pygame.event.get():
33                if event.type == pygame.QUIT:
34                    mygame.running = False
35                if event.type == pygame.KEYDOWN:
36                    mygame.running = False
37                if event.type == pygame.MOUSEBUTTONDOWN:
38                    mygame.running = False
39
40            # draw
41            mygame.screen.fill(mygame.background)
42
43            # draw squares
44            for square in mygame.squares:
45                pygame.draw.rect(mygame.screen, mygame.red, square)
46
47            # update
48            pygame.display.update()
49
50            # clock
51            clock = pygame.time.Clock()
52            clock.tick(60)
53
54            # quit
55            if mygame.running == False:
56                pygame.quit()
57                sys.exit()

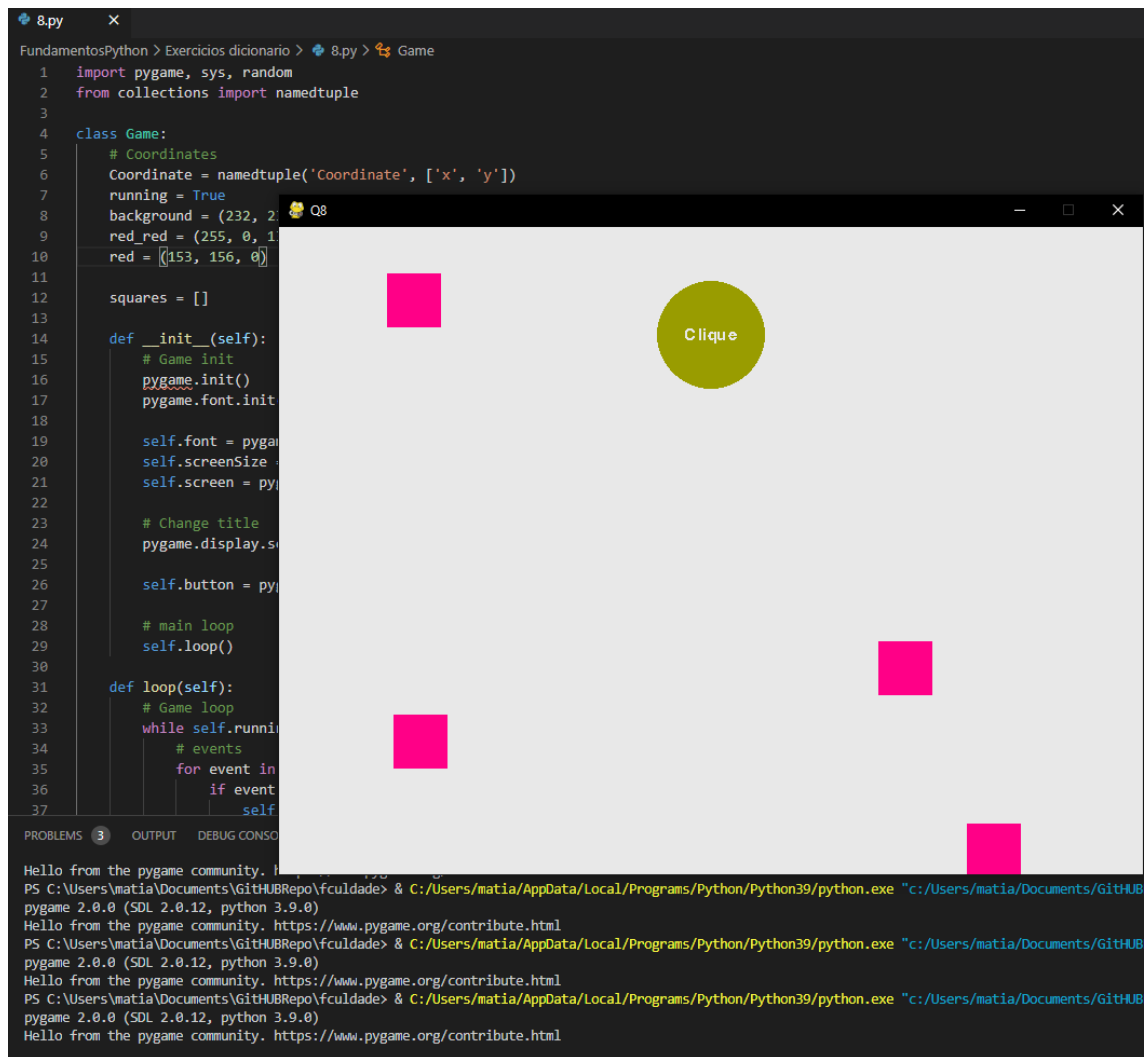
```

Windows PowerShell
 Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Experimente a nova plataforma cruzada PowerShell <https://aka.ms/pscore6>

PS C:\Users\matia\Documents\GitHubRepo\fculdade> & C:/Users/matia/AppData/Local/Programs/Python/Python39/python.exe "c:/Users/matia/Documents/GitHubRepo/7.py"
 pygame 2.0.0 (SDL 2.0.12, python 3.9.0)
 Hello from the pygame community. <https://www.pygame.org/contribute.html>

8)



```

8.py X
FundamentosPython > Exercicios dicionario > 8.py > Game
1 import pygame, sys, random
2 from collections import namedtuple
3
4 class Game:
5     # Coordinates
6     Coordinate = namedtuple('Coordinate', ['x', 'y'])
7     running = True
8     background = (232, 232, 232)
9     red_red = (255, 0, 0)
10    red = [(153, 156, 0)]
11
12    squares = []
13
14    def __init__(self):
15        # Game init
16        pygame.init()
17        pygame.font.init()
18
19        self.font = pygame.font.SysFont('arial', 16)
20        self.screenSize = pygame.display.get_surface().get_size()
21        self.screen = pygame.display.set_mode(self.screenSize)
22
23        # Change title
24        pygame.display.set_caption('Clique')
25
26        self.button = pygame.Rect(100, 100, 100, 100)
27
28        # main loop
29        self.loop()
30
31    def loop(self):
32        # Game loop
33        while self.running:
34            # events
35            for event in pygame.event.get():
36                if event.type == pygame.QUIT:
37                    self.running = False
38                elif event.type == pygame.MOUSEBUTTONDOWN:
39                    # Clicked
40                    x, y = pygame.mouse.get_pos()
41                    # Check if clicked on the button
42                    if self.button.collidepoint(x, y):
43                        # Change the color of the button
44                        self.button.fill(self.red_red)
45
46            # Draw
47            self.screen.fill(self.background)
48
49            # Draw the button
50            pygame.draw.rect(self.screen, self.red_red, self.button)
51
52            # Draw the squares
53            for square in self.squares:
54                pygame.draw.rect(self.screen, self.red, square)
55
56            # Update
57            pygame.display.update()
58
59            # Delay
60            pygame.time.Clock().tick(60)
61
62        # Quit
63        pygame.quit()
64        sys.exit()
65
66    def loop(self):
67        # Game loop
68        while self.running:
69            # events
70            for event in pygame.event.get():
71                if event.type == pygame.QUIT:
72                    self.running = False
73                elif event.type == pygame.MOUSEBUTTONDOWN:
74                    # Clicked
75                    x, y = pygame.mouse.get_pos()
76                    # Check if clicked on the button
77                    if self.button.collidepoint(x, y):
78                        # Change the color of the button
79                        self.button.fill(self.red_red)
80
81            # Draw
82            self.screen.fill(self.background)
83
84            # Draw the button
85            pygame.draw.rect(self.screen, self.red_red, self.button)
86
87            # Draw the squares
88            for square in self.squares:
89                pygame.draw.rect(self.screen, self.red, square)
90
91            # Update
92            pygame.display.update()
93
94            # Delay
95            pygame.time.Clock().tick(60)
96
97        # Quit
98        pygame.quit()
99        sys.exit()
100
101    def loop(self):
102        # Game loop
103        while self.running:
104            # events
105            for event in pygame.event.get():
106                if event.type == pygame.QUIT:
107                    self.running = False
108                elif event.type == pygame.MOUSEBUTTONDOWN:
109                    # Clicked
110                    x, y = pygame.mouse.get_pos()
111                    # Check if clicked on the button
112                    if self.button.collidepoint(x, y):
113                        # Change the color of the button
114                        self.button.fill(self.red_red)
115
116            # Draw
117            self.screen.fill(self.background)
118
119            # Draw the button
120            pygame.draw.rect(self.screen, self.red_red, self.button)
121
122            # Draw the squares
123            for square in self.squares:
124                pygame.draw.rect(self.screen, self.red, square)
125
126            # Update
127            pygame.display.update()
128
129            # Delay
130            pygame.time.Clock().tick(60)
131
132        # Quit
133        pygame.quit()
134        sys.exit()
135
136    def loop(self):
137        # Game loop
138        while self.running:
139            # events
140            for event in pygame.event.get():
141                if event.type == pygame.QUIT:
142                    self.running = False
143                elif event.type == pygame.MOUSEBUTTONDOWN:
144                    # Clicked
145                    x, y = pygame.mouse.get_pos()
146                    # Check if clicked on the button
147                    if self.button.collidepoint(x, y):
148                        # Change the color of the button
149                        self.button.fill(self.red_red)
150
151            # Draw
152            self.screen.fill(self.background)
153
154            # Draw the button
155            pygame.draw.rect(self.screen, self.red_red, self.button)
156
157            # Draw the squares
158            for square in self.squares:
159                pygame.draw.rect(self.screen, self.red, square)
160
161            # Update
162            pygame.display.update()
163
164            # Delay
165            pygame.time.Clock().tick(60)
166
167        # Quit
168        pygame.quit()
169        sys.exit()
170
171    def loop(self):
172        # Game loop
173        while self.running:
174            # events
175            for event in pygame.event.get():
176                if event.type == pygame.QUIT:
177                    self.running = False
178                elif event.type == pygame.MOUSEBUTTONDOWN:
179                    # Clicked
180                    x, y = pygame.mouse.get_pos()
181                    # Check if clicked on the button
182                    if self.button.collidepoint(x, y):
183                        # Change the color of the button
184                        self.button.fill(self.red_red)
185
186            # Draw
187            self.screen.fill(self.background)
188
189            # Draw the button
190            pygame.draw.rect(self.screen, self.red_red, self.button)
191
192            # Draw the squares
193            for square in self.squares:
194                pygame.draw.rect(self.screen, self.red, square)
195
196            # Update
197            pygame.display.update()
198
199            # Delay
200            pygame.time.Clock().tick(60)
201
202        # Quit
203        pygame.quit()
204        sys.exit()
205
206    def loop(self):
207        # Game loop
208        while self.running:
209            # events
210            for event in pygame.event.get():
211                if event.type == pygame.QUIT:
212                    self.running = False
213                elif event.type == pygame.MOUSEBUTTONDOWN:
214                    # Clicked
215                    x, y = pygame.mouse.get_pos()
216                    # Check if clicked on the button
217                    if self.button.collidepoint(x, y):
218                        # Change the color of the button
219                        self.button.fill(self.red_red)
220
221            # Draw
222            self.screen.fill(self.background)
223
224            # Draw the button
225            pygame.draw.rect(self.screen, self.red_red, self.button)
226
227            # Draw the squares
228            for square in self.squares:
229                pygame.draw.rect(self.screen, self.red, square)
230
231            # Update
232            pygame.display.update()
233
234            # Delay
235            pygame.time.Clock().tick(60)
236
237        # Quit
238        pygame.quit()
239        sys.exit()
240
241    def loop(self):
242        # Game loop
243        while self.running:
244            # events
245            for event in pygame.event.get():
246                if event.type == pygame.QUIT:
247                    self.running = False
248                elif event.type == pygame.MOUSEBUTTONDOWN:
249                    # Clicked
250                    x, y = pygame.mouse.get_pos()
251                    # Check if clicked on the button
252                    if self.button.collidepoint(x, y):
253                        # Change the color of the button
254                        self.button.fill(self.red_red)
255
256            # Draw
257            self.screen.fill(self.background)
258
259            # Draw the button
260            pygame.draw.rect(self.screen, self.red_red, self.button)
261
262            # Draw the squares
263            for square in self.squares:
264                pygame.draw.rect(self.screen, self.red, square)
265
266            # Update
267            pygame.display.update()
268
269            # Delay
270            pygame.time.Clock().tick(60)
271
272        # Quit
273        pygame.quit()
274        sys.exit()
275
276    def loop(self):
277        # Game loop
278        while self.running:
279            # events
280            for event in pygame.event.get():
281                if event.type == pygame.QUIT:
282                    self.running = False
283                elif event.type == pygame.MOUSEBUTTONDOWN:
284                    # Clicked
285                    x, y = pygame.mouse.get_pos()
286                    # Check if clicked on the button
287                    if self.button.collidepoint(x, y):
288                        # Change the color of the button
289                        self.button.fill(self.red_red)
290
291            # Draw
292            self.screen.fill(self.background)
293
294            # Draw the button
295            pygame.draw.rect(self.screen, self.red_red, self.button)
296
297            # Draw the squares
298            for square in self.squares:
299                pygame.draw.rect(self.screen, self.red, square)
300
301            # Update
302            pygame.display.update()
303
304            # Delay
305            pygame.time.Clock().tick(60)
306
307        # Quit
308        pygame.quit()
309        sys.exit()
310
311    def loop(self):
312        # Game loop
313        while self.running:
314            # events
315            for event in pygame.event.get():
316                if event.type == pygame.QUIT:
317                    self.running = False
318                elif event.type == pygame.MOUSEBUTTONDOWN:
319                    # Clicked
320                    x, y = pygame.mouse.get_pos()
321                    # Check if clicked on the button
322                    if self.button.collidepoint(x, y):
323                        # Change the color of the button
324                        self.button.fill(self.red_red)
325
326            # Draw
327            self.screen.fill(self.background)
328
329            # Draw the button
330            pygame.draw.rect(self.screen, self.red_red, self.button)
331
332            # Draw the squares
333            for square in self.squares:
334                pygame.draw.rect(self.screen, self.red, square)
335
336            # Update
337            pygame.display.update()
338
339            # Delay
340            pygame.time.Clock().tick(60)
341
342        # Quit
343        pygame.quit()
344        sys.exit()
345
346    def loop(self):
347        # Game loop
348        while self.running:
349            # events
350            for event in pygame.event.get():
351                if event.type == pygame.QUIT:
352                    self.running = False
353                elif event.type == pygame.MOUSEBUTTONDOWN:
354                    # Clicked
355                    x, y = pygame.mouse.get_pos()
356                    # Check if clicked on the button
357                    if self.button.collidepoint(x, y):
358                        # Change the color of the button
359                        self.button.fill(self.red_red)
360
361            # Draw
362            self.screen.fill(self.background)
363
364            # Draw the button
365            pygame.draw.rect(self.screen, self.red_red, self.button)
366
367            # Draw the squares
368            for square in self.squares:
369                pygame.draw.rect(self.screen, self.red, square)
370
371            # Update
372            pygame.display.update()
373
374            # Delay
375            pygame.time.Clock().tick(60)
376
377        # Quit
378        pygame.quit()
379        sys.exit()
380
381    def loop(self):
382        # Game loop
383        while self.running:
384            # events
385            for event in pygame.event.get():
386                if event.type == pygame.QUIT:
387                    self.running = False
388                elif event.type == pygame.MOUSEBUTTONDOWN:
389                    # Clicked
390                    x, y = pygame.mouse.get_pos()
391                    # Check if clicked on the button
392                    if self.button.collidepoint(x, y):
393                        # Change the color of the button
394                        self.button.fill(self.red_red)
395
396            # Draw
397            self.screen.fill(self.background)
398
399            # Draw the button
400            pygame.draw.rect(self.screen, self.red_red, self.button)
401
402            # Draw the squares
403            for square in self.squares:
404                pygame.draw.rect(self.screen, self.red, square)
405
406            # Update
407            pygame.display.update()
408
409            # Delay
410            pygame.time.Clock().tick(60)
411
412        # Quit
413        pygame.quit()
414        sys.exit()
415
416    def loop(self):
417        # Game loop
418        while self.running:
419            # events
420            for event in pygame.event.get():
421                if event.type == pygame.QUIT:
422                    self.running = False
423                elif event.type == pygame.MOUSEBUTTONDOWN:
424                    # Clicked
425                    x, y = pygame.mouse.get_pos()
426                    # Check if clicked on the button
427                    if self.button.collidepoint(x, y):
428                        # Change the color of the button
429                        self.button.fill(self.red_red)
430
431            # Draw
432            self.screen.fill(self.background)
433
434            # Draw the button
435            pygame.draw.rect(self.screen, self.red_red, self.button)
436
437            # Draw the squares
438            for square in self.squares:
439                pygame.draw.rect(self.screen, self.red, square)
440
441            # Update
442            pygame.display.update()
443
444            # Delay
445            pygame.time.Clock().tick(60)
446
447        # Quit
448        pygame.quit()
449        sys.exit()
450
451    def loop(self):
452        # Game loop
453        while self.running:
454            # events
455            for event in pygame.event.get():
456                if event.type == pygame.QUIT:
457                    self.running = False
458                elif event.type == pygame.MOUSEBUTTONDOWN:
459                    # Clicked
460                    x, y = pygame.mouse.get_pos()
461                    # Check if clicked on the button
462                    if self.button.collidepoint(x, y):
463                        # Change the color of the button
464                        self.button.fill(self.red_red)
465
466            # Draw
467            self.screen.fill(self.background)
468
469            # Draw the button
470            pygame.draw.rect(self.screen, self.red_red, self.button)
471
472            # Draw the squares
473            for square in self.squares:
474                pygame.draw.rect(self.screen, self.red, square)
475
476            # Update
477            pygame.display.update()
478
479            # Delay
480            pygame.time.Clock().tick(60)
481
482        # Quit
483        pygame.quit()
484        sys.exit()
485
486    def loop(self):
487        # Game loop
488        while self.running:
489            # events
490            for event in pygame.event.get():
491                if event.type == pygame.QUIT:
492                    self.running = False
493                elif event.type == pygame.MOUSEBUTTONDOWN:
494                    # Clicked
495                    x, y = pygame.mouse.get_pos()
496                    # Check if clicked on the button
497                    if self.button.collidepoint(x, y):
498                        # Change the color of the button
499                        self.button.fill(self.red_red)
500
501            # Draw
502            self.screen.fill(self.background)
503
504            # Draw the button
505            pygame.draw.rect(self.screen, self.red_red, self.button)
506
507            # Draw the squares
508            for square in self.squares:
509                pygame.draw.rect(self.screen, self.red, square)
510
511            # Update
512            pygame.display.update()
513
514            # Delay
515            pygame.time.Clock().tick(60)
516
517        # Quit
518        pygame.quit()
519        sys.exit()
520
521    def loop(self):
522        # Game loop
523        while self.running:
524            # events
525            for event in pygame.event.get():
526                if event.type == pygame.QUIT:
527                    self.running = False
528                elif event.type == pygame.MOUSEBUTTONDOWN:
529                    # Clicked
530                    x, y = pygame.mouse.get_pos()
531                    # Check if clicked on the button
532                    if self.button.collidepoint(x, y):
533                        # Change the color of the button
534                        self.button.fill(self.red_red)
535
536            # Draw
537            self.screen.fill(self.background)
538
539            # Draw the button
540            pygame.draw.rect(self.screen, self.red_red, self.button)
541
542            # Draw the squares
543            for square in self.squares:
544                pygame.draw.rect(self.screen, self.red, square)
545
546            # Update
547            pygame.display.update()
548
549            # Delay
550            pygame.time.Clock().tick(60)
551
552        # Quit
553        pygame.quit()
554        sys.exit()
555
556    def loop(self):
557        # Game loop
558        while self.running:
559            # events
560            for event in pygame.event.get():
561                if event.type == pygame.QUIT:
562                    self.running = False
563                elif event.type == pygame.MOUSEBUTTONDOWN:
564                    # Clicked
565                    x, y = pygame.mouse.get_pos()
566                    # Check if clicked on the button
567                    if self.button.collidepoint(x, y):
568                        # Change the color of the button
569                        self.button.fill(self.red_red)
570
571            # Draw
572            self.screen.fill(self.background)
573
574            # Draw the button
575            pygame.draw.rect(self.screen, self.red_red, self.button)
576
577            # Draw the squares
578            for square in self.squares:
579                pygame.draw.rect(self.screen, self.red, square)
580
581            # Update
582            pygame.display.update()
583
584            # Delay
585            pygame.time.Clock().tick(60)
586
587        # Quit
588        pygame.quit()
589        sys.exit()
590
591    def loop(self):
592        # Game loop
593        while self.running:
594            # events
595            for event in pygame.event.get():
596                if event.type == pygame.QUIT:
597                    self.running = False
598                elif event.type == pygame.MOUSEBUTTONDOWN:
599                    # Clicked
600                    x, y = pygame.mouse.get_pos()
601                    # Check if clicked on the button
602                    if self.button.collidepoint(x, y):
603                        # Change the color of the button
604                        self.button.fill(self.red_red)
605
606            # Draw
607            self.screen.fill(self.background)
608
609            # Draw the button
610            pygame.draw.rect(self.screen, self.red_red, self.button)
611
612            # Draw the squares
613            for square in self.squares:
614                pygame.draw.rect(self.screen, self.red, square)
615
616            # Update
617            pygame.display.update()
618
619            # Delay
620            pygame.time.Clock().tick(60)
621
622        # Quit
623        pygame.quit()
624        sys.exit()
625
626    def loop(self):
627        # Game loop
628        while self.running:
629            # events
630            for event in pygame.event.get():
631                if event.type == pygame.QUIT:
632                    self.running = False
633                elif event.type == pygame.MOUSEBUTTONDOWN:
634                    # Clicked
635                    x, y = pygame.mouse.get_pos()
636                    # Check if clicked on the button
637                    if self.button.collidepoint(x, y):
638                        # Change the color of the button
639                        self.button.fill(self.red_red)
640
641            # Draw
642            self.screen.fill(self.background)
643
644            # Draw the button
645            pygame.draw.rect(self.screen, self.red_red, self.button)
646
647            # Draw the squares
648            for square in self.squares:
649                pygame.draw.rect(self.screen, self.red, square)
650
651            # Update
652            pygame.display.update()
653
654            # Delay
655            pygame.time.Clock().tick(60)
656
657        # Quit
658        pygame.quit()
659        sys.exit()
660
661    def loop(self):
662        # Game loop
663        while self.running:
664            # events
665            for event in pygame.event.get():
666                if event.type == pygame.QUIT:
667                    self.running = False
668                elif event.type == pygame.MOUSEBUTTONDOWN:
669                    # Clicked
670                    x, y = pygame.mouse.get_pos()
671                    # Check if clicked on the button
672                    if self.button.collidepoint(x, y):
673                        # Change the color of the button
674                        self.button.fill(self.red_red)
675
676            # Draw
677            self.screen.fill(self.background)
678
679            # Draw the button
680            pygame.draw.rect(self.screen, self.red_red, self.button)
681
682            # Draw the squares
683            for square in self.squares:
684                pygame.draw.rect(self.screen, self.red, square)
685
686            # Update
687            pygame.display.update()
688
689            # Delay
690            pygame.time.Clock().tick(60)
691
692        # Quit
693        pygame.quit()
694        sys.exit()
695
696    def loop(self):
697        # Game loop
698        while self.running:
699            # events
700            for event in pygame.event.get():
701                if event.type == pygame.QUIT:
702                    self.running = False
703                elif event.type == pygame.MOUSEBUTTONDOWN:
704                    # Clicked
705                    x, y = pygame.mouse.get_pos()
706                    # Check if clicked on the button
707                    if self.button.collidepoint(x, y):
708                        # Change the color of the button
709                        self.button.fill(self.red_red)
710
711            # Draw
712            self.screen.fill(self.background)
713
714            # Draw the button
715            pygame.draw.rect(self.screen, self.red_red, self.button)
716
717            # Draw the squares
718            for square in self.squares:
719                pygame.draw.rect(self.screen, self.red, square)
720
721            # Update
722            pygame.display.update()
723
724            # Delay
725            pygame.time.Clock().tick(60)
726
727        # Quit
728        pygame.quit()
729        sys.exit()
730
731    def loop(self):
732        # Game loop
733        while self.running:
734            # events
735            for event in pygame.event.get():
736                if event.type == pygame.QUIT:
737                    self.running = False
738                elif event.type == pygame.MOUSEBUTTONDOWN:
739                    # Clicked
740                    x, y = pygame.mouse.get_pos()
741                    # Check if clicked on the button
742                    if self.button.collidepoint(x, y):
743                        # Change the color of the button
744                        self.button.fill(self.red_red)
745
746            # Draw
747            self.screen.fill(self.background)
748
749            # Draw the button
750            pygame.draw.rect(self.screen, self.red_red, self.button)
751
752            # Draw the squares
753            for square in self.squares:
754                pygame.draw.rect(self.screen, self.red, square)
755
756            # Update
757            pygame.display.update()
758
759            # Delay
760            pygame.time.Clock().tick(60)
761
762        # Quit
763        pygame.quit()
764        sys.exit()
765
766    def loop(self):
767        # Game loop
768        while self.running:
769            # events
770            for event in pygame.event.get():
771                if event.type == pygame.QUIT:
772                    self.running = False
773                elif event.type == pygame.MOUSEBUTTONDOWN:
774                    # Clicked
775                    x, y = pygame.mouse.get_pos()
776                    # Check if clicked on the button
777                    if self.button.collidepoint(x, y):
778                        # Change the color of the button
779                        self.button.fill(self.red_red)
780
781            # Draw
782            self.screen.fill(self.background)
783
784            # Draw the button
785            pygame.draw.rect(self.screen, self.red_red, self.button)
786
787            # Draw the squares
788            for square in self.squares:
789                pygame.draw.rect(self.screen, self.red, square)
790
791            # Update
792            pygame.display.update()
793
794            # Delay
795            pygame.time.Clock().tick(60)
796
797        # Quit
798        pygame.quit()
799        sys.exit()
800
801    def loop(self):
802        # Game loop
803        while self.running:
804            # events
805            for event in pygame.event.get():
806                if event.type == pygame.QUIT:
807                    self.running = False
808                elif event.type == pygame.MOUSEBUTTONDOWN:
809                    # Clicked
810                    x, y = pygame.mouse.get_pos()
811                    # Check if clicked on the button
812                    if self.button.collidepoint(x, y):
813                        # Change the color of the button
814                        self.button.fill(self.red_red)
815
816            # Draw
817            self.screen.fill(self.background)
818
819            # Draw the button
820            pygame.draw.rect(self.screen, self.red_red, self.button)
821
822            # Draw the squares
823            for square in self.squares:
824                pygame.draw.rect(self.screen, self.red, square)
825
826            # Update
827            pygame.display.update()
828
829            # Delay
830            pygame.time.Clock().tick(60)
831
832        # Quit
833        pygame.quit()
834        sys.exit()
835
836    def loop(self):
837        # Game loop
838        while self.running:
839            # events
840            for event in pygame.event.get():
841                if event.type == pygame.QUIT:
842                    self.running = False
843                elif event.type == pygame.MOUSEBUTTONDOWN:
844                    # Clicked
845                    x, y = pygame.mouse.get_pos()
846                    # Check if clicked on the button
847                    if self.button.collidepoint(x, y):
848                        # Change the color of the button
849                        self.button.fill(self.red_red)
850
851            # Draw
852            self.screen.fill(self.background)
853
854            # Draw the button
855            pygame.draw.rect(self.screen, self.red_red, self.button)
856
857            # Draw the squares
858            for square in self.squares:
859                pygame.draw.rect(self.screen, self.red, square)
860
861            # Update
862            pygame.display.update()
863
864            # Delay
865            pygame.time.Clock().tick(60)
866
867        # Quit
868        pygame.quit()
869        sys.exit()
870
871    def loop(self):
872        # Game loop
873        while self.running:
874            # events
875            for event in pygame.event.get():
876                if event.type == pygame.QUIT:
877                    self.running = False
878                elif event.type == pygame.MOUSEBUTTONDOWN:
879                    # Clicked
880                    x, y = pygame.mouse.get_pos()
881                    # Check if clicked on the button
882                    if self.button.collidepoint(x, y):
883                        # Change the color of the button
884                        self.button.fill(self.red_red)
885
886            # Draw
887            self.screen.fill(self.background)
888
889            # Draw the button
890            pygame.draw.rect(self.screen, self.red_red, self.button)
891
892            # Draw the squares
893            for square in self.squares:
894                pygame.draw.rect(self.screen, self.red, square)
895
896            # Update
897            pygame.display.update()
898
899            # Delay
900            pygame.time.Clock().tick(60)
901
902        # Quit
903        pygame.quit()
904        sys.exit()
905
906    def loop(self):
907        # Game loop
908        while self.running:
909            # events
910            for event in pygame.event.get():
911                if event.type == pygame.QUIT:
912                    self.running = False
913                elif event.type == pygame.MOUSEBUTTONDOWN:
914                    # Clicked
915                    x, y = pygame.mouse.get_pos()
916                    # Check if clicked on the button
917                    if self.button.collidepoint(x, y):
918                        # Change the color of the button
919                        self.button.fill(self.red_red)
920
921            # Draw
922            self.screen.fill(self.background)
923
924            # Draw the button
925            pygame.draw.rect(self.screen, self.red_red, self.button)
926
927            # Draw the squares
928            for square in self.squares:
929                pygame.draw.rect(self.screen, self.red, square)
930
931            # Update
932            pygame.display.update()
933
934            # Delay
935            pygame.time.Clock().tick(60)
936
937        # Quit
938        pygame.quit()
939        sys.exit()
940
941    def loop(self):
942        # Game loop
943        while self.running:
944            # events
945            for event in pygame.event.get():
946                if event.type == pygame.QUIT:
947                    self.running = False
948                elif event.type == pygame.MOUSEBUTTONDOWN:
949                    # Clicked
950                    x, y = pygame.mouse.get_pos()
951                    # Check if clicked on the button
952                    if self.button.collidepoint(x, y):
953                        # Change the color of the button
954                        self.button.fill(self.red_red)
955
956            # Draw
957            self.screen.fill(self.background)
958
959            # Draw the button
960            pygame.draw.rect(self.screen, self.red_red, self.button)
961
962            # Draw the squares
963            for square in self.squares:
964                pygame.draw.rect(self.screen, self.red, square)
965
966            # Update
967            pygame.display.update()
968
969            # Delay
970            pygame.time.Clock().tick(60)
971
972        # Quit
973        pygame.quit()
974        sys.exit()
975
976    def loop(self):
977        # Game loop
978        while self.running:
979            # events
980            for event in pygame.event.get():
981                if event.type == pygame.QUIT:
982                    self.running = False
983                elif event.type == pygame.MOUSEBUTTONDOWN:
984                    # Clicked
985                    x, y = pygame.mouse.get_pos()
986                    # Check if clicked on the button
987                    if self.button.collidepoint(x, y):
988                        # Change the color of the button
989                        self.button.fill(self.red_red)
990
991            # Draw
992            self.screen.fill(self.background)
993
994            # Draw the button
995            pygame.draw.rect(self.screen, self.red_red, self.button)
996
997            # Draw the squares
998            for square in self.squares:
999                pygame.draw.rect(self.screen, self.red, square)
1000
1001            # Update
1002            pygame.display.update()
1003
1004            # Delay
1005            pygame.time.Clock().tick(60)
1006
1007        # Quit
1008        pygame.quit()
1009        sys.exit()
1010
1011    def loop(self):
1012        # Game loop
1013        while self.running:
1014            # events
1015            for event in pygame.event.get():
1016                if event.type == pygame.QUIT:
1017                    self.running = False
1018                elif event.type == pygame.MOUSEBUTTONDOWN:
1019                    # Clicked
1020                    x, y = pygame.mouse.get_pos()
1021                    # Check if clicked on the button
1022                    if self.button.collidepoint(x, y):
1023                        # Change the color of the button
1024                        self.button.fill(self.red_red)
1025
1026            # Draw
1027            self.screen.fill(self.background)
1028
1029            # Draw the button
1030            pygame.draw.rect(self.screen, self.red_red, self.button)
1031
1032            # Draw the squares
1033            for square in self.squares:
1034                pygame.draw.rect(self.screen, self.red, square)
1035
1036            # Update
1037            pygame.display.update()
1038
1039            # Delay
1040            pygame.time.Clock().tick(60)
1041
1042        # Quit
1043        pygame.quit()
1044        sys.exit()
1045
1046    def loop(self):
1047        # Game loop
1048        while self.running:
1049            # events
1050            for event in pygame.event.get():
1051                if event.type == pygame.QUIT:
1052                    self.running = False
1053                elif event.type == pygame.MOUSEBUTTONDOWN:
1054                    # Clicked
1055                    x, y = pygame.mouse.get_pos()
1056                    # Check if clicked on the button
1057                    if self.button.collidepoint(x, y):
1058                        # Change the color of the button
1059                        self.button.fill(self.red_red)
1060
1061            # Draw
1062            self.screen.fill(self.background)
1063
1064            # Draw the button
1065            pygame.draw.rect(self.screen, self.red_red, self.button)
1066
1067            # Draw the squares
1068            for square in self.squares:
1069                pygame.draw.rect(self.screen, self.red, square)
1070
1071            # Update
1072            pygame.display.update()
1073
1074            # Delay
1075            pygame.time.Clock().tick(60)
1076
1077        # Quit
1078        pygame.quit()
1079        sys.exit()
1080
1081    def loop(self):
1082        # Game loop
1083        while self.running:
1084            # events
1085            for event in pygame.event.get():
1086                if event.type == pygame.QUIT:
1087                    self.running = False
1088                elif event.type == pygame.MOUSEBUTTONDOWN:
1089                    # Clicked
1090                    x, y = pygame.mouse.get_pos()
1091                    # Check if clicked on the button
1092                    if self.button.collidepoint(x, y):
1093                        # Change the color of the button
1094                        self.button.fill(self.red_red)
1095
1096            # Draw
1097            self.screen.fill(self.background)
1098
1099            # Draw the button
1100            pygame.draw.rect(self.screen, self.red_red, self.button)
1101
1102            # Draw the squares
1103            for square in self.squares:
1104                pygame.draw.rect(self.screen, self.red, square)
1105
1106            # Update
1107            pygame.display.update()
1108
1109            # Delay
1110            pygame.time.Clock().tick(60)
1111
1112        # Quit
1113        pygame.quit()
1114        sys.exit()
1115
1116    def loop(self):
1117        # Game loop
1118        while self.running:
1119            # events
1120            for event in pygame.event.get():
1121                if event.type == pygame.QUIT:
1122                    self.running = False
1123                elif event.type == pygame.MOUSEBUTTONDOWN:
1124                    # Clicked
1125                    x, y = pygame.mouse.get_pos()
1126                    # Check if clicked on the button
1127                    if self.button.collidepoint(x, y):
1128                        # Change the color of the button
1129                        self.button.fill(self.red_red)
1130
1131            # Draw
1132            self.screen.fill(self.background)
1133
1134            # Draw the button
1135            pygame.draw.rect(self.screen, self.red_red, self.button)
1136
1137            # Draw the squares
1138            for square in self.squares:
1139                pygame.draw.rect(self.screen, self.red, square)
1140
1141            # Update
1142            pygame.display.update()
1143
1144            # Delay
1145            pygame.time.Clock().tick(60)
1146
1147        # Quit
1148        pygame.quit()
1149        sys.exit()
1150
1151    def loop(self):
1152        # Game loop
1153        while self.running:
1154            # events
1155            for event in pygame.event.get():
1156                if event.type == pygame.QUIT:
1157                    self.running = False
1158                elif event.type == pygame.MOUSEBUTTONDOWN:
1159                    # Clicked
1160                    x, y = pygame.mouse.get_pos()
1161                    # Check if clicked on the button
1162                    if self.button.collidepoint(x, y):
1163                        # Change the color of the button
1164                        self.button.fill(self.red_red)
1165
1166            # Draw
1167            self.screen.fill(self.background)
1168
1169            # Draw the button
1170            pygame.draw.rect(self.screen, self.red_red, self.button)
1171
1172            # Draw the squares
1173            for square in self.squares:
1174                pygame.draw.rect(self.screen, self.red, square)
1175
1176            # Update
1177            pygame.display.update()
1178
1179            # Delay
1180            pygame.time.Clock().tick(60)
1181
1182        # Quit
1183        pygame.quit()
1184        sys.exit()
1185
1186    def loop(self):
1187        # Game loop
1188        while self.running:
1189            # events
1190            for event in pygame.event.get():
1191                if event.type == pygame.QUIT:
1192                    self.running = False
1193                elif event.type == pygame.MOUSEBUTTONDOWN:
1194                    # Clicked
1195                    x, y = pygame.mouse.get_pos()
1196                    # Check if clicked on the button
1197                    if self.button.collidepoint(x, y):
1198                        # Change the color of the button
1199                        self.button.fill(self.red_red)
1200
1201            # Draw
1202            self.screen.fill(self.background)
1203
1204            # Draw the button
1205            pygame.draw.rect(self.screen, self.red_red, self.button)
1206
1207            # Draw the squares
1208            for square in self.squares:
1209                pygame.draw.rect(self.screen, self.red, square)
1210
1211            # Update
1212            pygame.display.update()
1213
1214            # Delay
1215            pygame.time.Clock().tick(60)
1216
1217        # Quit
1218        pygame.quit()
1219        sys.exit()
1220
1221    def loop(self):
1222        # Game loop
1223        while self.running:
1224            # events
1225            for event in pygame.event.get():
1226                if event.type == pygame.QUIT:
1227                    self.running = False
1228                elif event.type == pygame.MOUSEBUTTONDOWN:
1229                    # Clicked
1230                    x, y = pygame.mouse.get_pos()
1231                    # Check if clicked on the button
1232                    if self.button.collidepoint(x, y):
1233                        # Change the color of the button
1234                        self.button.fill(self.red_red)
1235
1236            # Draw
1237            self.screen.fill(self.background)
1238
1239            # Draw the button
1240            pygame.draw.rect(self.screen, self.red_red, self.button)
1241
1242            # Draw the squares
1243            for square in self.squares:
1244                pygame.draw.rect(self.screen, self.red, square)
1245
1246            # Update
1247            pygame.display.update()
1248
1249            # Delay
1250            pygame.time.Clock().tick(60)
1251
1252        # Quit
1253        pygame.quit()
1254        sys.exit()
1255
1256    def loop(self):
1257        # Game loop
1258        while self.running:
1259            # events
1260            for event in pygame.event.get():
1261                if event.type == pygame.QUIT:
1262                    self.running = False
1263                elif event.type == pygame.MOUSEBUTTONDOWN:
1264                    # Clicked
1265                    x, y = pygame.mouse.get_pos()
1266                    # Check if clicked on the button
1267                    if self.button.collidepoint(x, y):
1268                        # Change the color of the button
1269                        self.button.fill(self.red_red)
1270
1271            # Draw
1272            self.screen.fill(self.background)
1273
1274            # Draw the button
1275            pygame.draw.rect(self.screen, self.red_red, self.button)
1276
1277            # Draw the squares
1278            for square in self.squares:
1279                pygame.draw.rect(self.screen, self.red, square)
1280
1281            # Update
1282            pygame.display.update()
1283
1284            # Delay
1285            pygame.time.Clock().tick(60)
1286
1287        # Quit
1288        pygame.quit()
1289        sys.exit()
1290
1291    def loop(self):
1292        # Game loop
1293        while self.running:
1294            # events
1295            for event in pygame.event.get():
1296                if event.type == pygame.QUIT:

```


9)

```

9.py
FundamentosPython > Exercicios dicionario > 9.py > Game
1 import pygame, sys, random
2 from collections import namedtuple
3
4
5 class Game:
6     # Coordinates
7     Coordinate = namedtuple('Coordinate', 'x y')
8     running = True
9     background = (232, 232, 232)
10    roxo = (255, 0, 150)
11    bluewater = (120, 130, 150)
12    speed = 5
13
14    squares = []
15
16    def __init__(self):
17        # Game init
18        pygame.init()
19        pygame.font.init()
20
21        self.font = pygame.font.SysFont('arial', 16)
22        self.screenSize = (800, 600)
23        self.screen = pygame.display.set_mode(self.screenSize)
24
25        # Change title
26        pygame.display.set_caption('Q9')
27
28        self.button = pygame.Rect(100, 100, 100, 100)
29
30        # main loop
31        self.loop()
32
33    def loop(self):
34        # Game loop
35        while self.running:
36            # events
37            for event in pygame.event.get():
38                if event.type == pygame.QUIT:
39                    self.running = False
40                elif event.type == pygame.MOUSEBUTTONDOWN:
41                    # Clicked
42                    x, y = pygame.mouse.get_pos()
43                    # Check if clicked on the circle
44                    if (x - 500)**2 + (y - 500)**2 <= 10000:
45                        # Circle clicked
46                        print('Clique')
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Hello from the pygame community. <https://www.pygame.org/contribute.html>

PS C:\Users\matia\Documents\GitHubRepo\fculdade> & C:/Users/matia/AppData/Local/Programs/Python/Python39/python.exe "c:/Users/matia/Documents/GitHubRepo/9.py"

pygame 2.0.0 (SDL 2.0.12, python 3.9.0)

Hello from the pygame community. <https://www.pygame.org/contribute.html>

PS C:\Users\matia\Documents\GitHubRepo\fculdade> & C:/Users/matia/AppData/Local/Programs/Python/Python39/python.exe "c:/Users/matia/Documents/GitHubRepo/9.py"

pygame 2.0.0 (SDL 2.0.12, python 3.9.0)

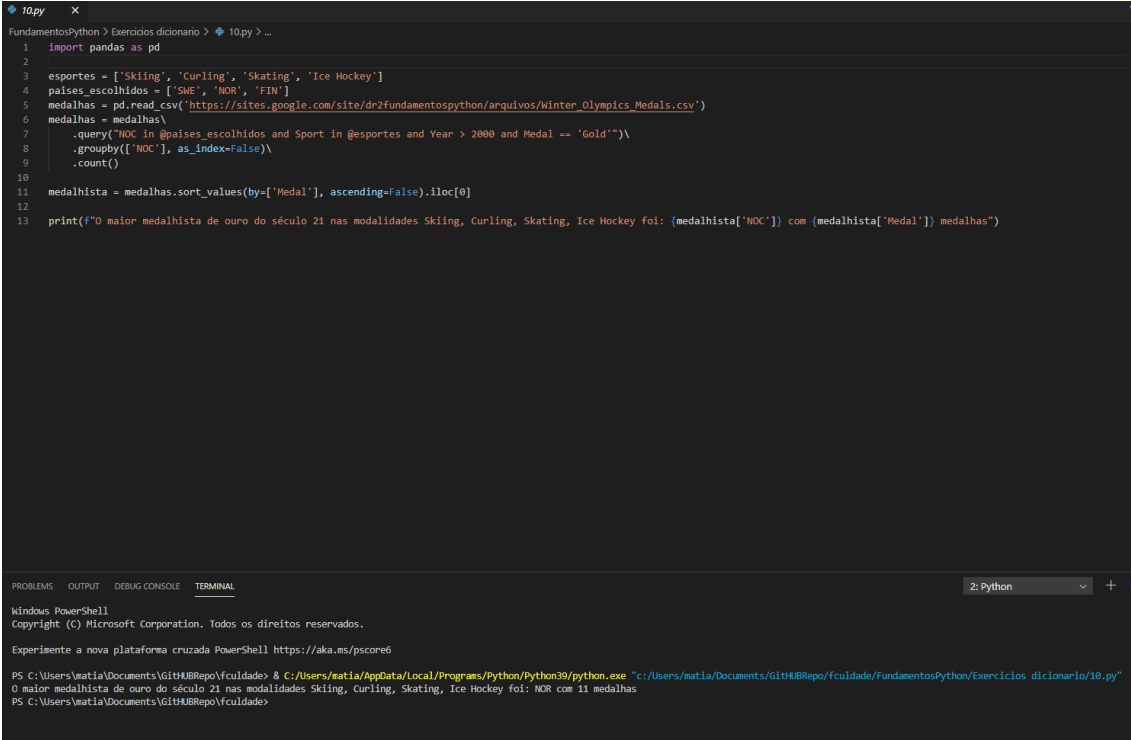
Hello from the pygame community. <https://www.pygame.org/contribute.html>

PS C:\Users\matia\Documents\GitHubRepo\fculdade> & C:/Users/matia/AppData/Local/Programs/Python/Python39/python.exe "c:/Users/matia/Documents/GitHubRepo/9.py"

pygame 2.0.0 (SDL 2.0.12, python 3.9.0)

Hello from the pygame community. <https://www.pygame.org/contribute.html>

10.a)



```
1 import pandas as pd
2
3 esportes = ['Skiing', 'Curling', 'Skating', 'Ice Hockey']
4 paises_escolhidos = ['SWE', 'NOR', 'FIN']
5 medalhas = pd.read_csv('https://sites.google.com/site/dr2fundamentospython/arquivos/Winter_Olympics_Medals.csv')
6 medalhas = medalhas\
7     .query("NOC in @paises_escolhidos and Sport in @esportes and Year > 2000 and Medal == 'Gold'")\
8     .groupby(['NOC'], as_index=False)\
9     .count()
10
11 medalhista = medalhas.sort_values(by=['Medal'], ascending=False).iloc[0]
12
13 print(f"0 maior medalhista de ouro do século 21 nas modalidades Skiing, Curling, Skating, Ice Hockey foi: {medalhista['NOC']} com {medalhista['Medal']} medalhas")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Experimente a nova plataforma cruzada PowerShell <https://aka.ms/pscore6>

PS C:\Users\matia\Documents\GitHubRepo\fculdade> & C:\Users\matia\AppData\Local\Programs\Python\Python39\python.exe "c:/Users/matia/Documents/GitHubRepo/fculdade/fundamentosPython/Exercicios dicionario/10.py"
0 maior medalhista de ouro do século 21 nas modalidades Skiing, Curling, Skating, Ice Hockey foi: NOR com 11 medalhas
PS C:\Users\matia\Documents\GitHubRepo\fculdade>

10.b)

```

10-b.py x
FundamentosPython > Exercicios dicionario > 10-b.py > ...
1  import pandas as pd
2
3  data_frame = pd.read_csv(
4      'https://sites.google.com/site/dr2fundamentospython/arquivos/Winter_Olympics_Medals.csv', sep=',')
5
6
7  medalha_maior = data_frame[(data_frame['NOC'] == 'SWE') | (
8      data_frame['NOC'] == 'DEN') | (data_frame['NOC'] == 'NOR')]
9
10
11 ano_filtro = medalha_maior[(medalha_maior['Year'] >= 2001)]
12
13 sports_filter = ano_filtro[(ano_filtro['Sport'] == 'Curling') | (ano_filtro['Sport'] == 'Skating') | (
14     ano_filtro['Sport'] == 'Skiing') | (ano_filtro['Sport'] == 'Ice Hockey')]
15
16 filter_medals_of_gold = sports_filter[sports_filter['Medal'] == 'Gold']
17
18 maior_medalhista_ouro_filtro = filter_medals_of_gold.groupby(
19     ['NOC'])['Medal'].count().sort_values()
20 print(maior_medalhista_ouro_filtro)

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Experimente a nova plataforma cruzada PowerShell <https://aka.ms/pscore6>

```

PS C:\Users\matia\Documents\GitHubRepo\fculdade> & C:/Users/matia/AppData/Local/Programs/Python/Python39/python.exe "c:/Users/mat
NOC
SWE      6
NOR     11
Name: Medal, dtype: int64
PS C:\Users\matia\Documents\GitHubRepo\fculdade>

```

11)

```

11.py x
FundamentosPython > Exercicios dicionario > 11.py > ...
1 import pandas as pd
2 import numpy as np
3
4 url = "https://sites.google.com/site/dr2fundamentospython/arquivos/Video_Games_Sales_as_at_22_Dec_2016.csv"
5
6 dataframe = pd.read_csv(url, sep=',')
7 frame_genre = dataframe[(dataframe['Genre'] == 'Action') | (dataframe['Genre'] == 'Shooter') | (dataframe['Genre'] == 'Platform')]
8
9 print(f"A: {frame_genre.groupby(['Publisher'])['Publisher'].count().sort_values(ascending=False).head(3)}")
10
11 global_sales = dataframe.groupby(['Publisher'])['Global_Sales'].sum().sort_values(ascending=False)
12
13 print(f"B: {global_sales.head(3)}")

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Experimente a nova plataforma cruzada PowerShell <https://aka.ms/pscore6>

```

PS C:\Users\matia\Documents\GitHubRepo\fculdade> & C:\Users\matia\AppData\Local\Programs\Python\Python39\python.exe "c:/Users/matia/Documents/GitHubRepo/fculdade/11.py"
A: Publisher
Activision      538
Ubisoft         360
Electronic Arts  344
Name: Publisher, dtype: int64
B: Publisher
Nintendo      1788.81
Electronic Arts 1116.96
Activision     731.16
Name: Global_Sales, dtype: float64
C-1: Publisher
Namco Bandai Games 171
Name: Publisher, dtype: int64
C-2: Publisher
Activision      71
Name: Publisher, dtype: int64
C-3: Publisher
Nintendo      21
Name: Publisher, dtype: int64
D-1: Publisher
Namco Bandai Games 13.11
Name: JP_Sales, dtype: float64
D-2: Publisher
Activision      3.85
Name: JP_Sales, dtype: float64
D-3: Publisher
Nintendo      14.44
Name: JP_Sales, dtype: float64
PS C:\Users\matia\Documents\GitHubRepo\fculdade>

```

12)

```
12.py x
FundamentosPython > Exercicios dicionario > 12.py > ...
1 import requests
2 from bs4 import BeautifulSoup
3
4 request = requests.get('https://fgopassos.github.io/pagina_exemplo/estadosCentroOeste.html')
5
6 request.encoding = request.apparent_encoding
7 bs = BeautifulSoup(request.text, "xml")
8
9 tabela = bs.html.body.find('div', {'class': 'tabela'})
10
11 print(tabela.text)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Sigla
Nome
Capital
População
Área

DF
Distrito Federal
Brasília
2977216
5779,999

GO
Goiás
Goiânia
6730848
340111,783

MT
Mato Grosso
Cuiabá
3305531
903378,292

MS
Mato Grosso do Sul
Campo Grande
2651235
357145,532

PS C:\Users\matia\Documents\GitHubRepo\fculdade>

13)

```

13.py X
FundamentosPython > Exercicios dicionario > 13.py > ...
1  import requests, re
2  from collections import Counter
3  from bs4 import BeautifulSoup
4
5  request = requests.get('http://brasil.pyladies.com/about')
6  request.encoding = request.apparent_encoding
7  bs = BeautifulSoup(request.text, "lxml")
8
9  num_ladies = 0
10 total_words = 0
11 words = []
12
13 for elemento in bs.html.body.article.find_all('div'):

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL
A palavra 'segregar' apareceu somente uma vez na página
A palavra 'contrário' apareceu somente uma vez na página
A palavra 'criar' apareceu somente uma vez na página
A palavra 'ambiente' apareceu somente uma vez na página
A palavra 'sejam' apareceu somente uma vez na página
A palavra 'atraidás' apareceu somente uma vez na página
A palavra 'pela' apareceu somente uma vez na página
A palavra 'elas' apareceu somente uma vez na página
A palavra 'vejam' apareceu somente uma vez na página
A palavra 'pensem' apareceu somente uma vez na página
A palavra 'eu' apareceu somente uma vez na página
A palavra 'estar' apareceu somente uma vez na página
A palavra 'aí' apareceu somente uma vez na página
A palavra 'participar' apareceu somente uma vez na página
A palavra 'dojo' apareceu somente uma vez na página
A palavra 'muito' apareceu somente uma vez na página
A palavra 'legal' apareceu somente uma vez na página
A palavra 'experiência' apareceu somente uma vez na página
A palavra 'repetir' apareceu somente uma vez na página
A palavra 'vezes' apareceu somente uma vez na página
A palavra 'além' apareceu somente uma vez na página
A palavra 'aprender' apareceu somente uma vez na página
A palavra 'forma' apareceu somente uma vez na página
A palavra 'bem' apareceu somente uma vez na página
A palavra 'diferente' apareceu somente uma vez na página
A palavra 'rápida' apareceu somente uma vez na página
A palavra 'pude' apareceu somente uma vez na página
A palavra 'amizades' apareceu somente uma vez na página
A palavra 'conhecer' apareceu somente uma vez na página
A palavra 'melhor' apareceu somente uma vez na página
A palavra 'esse' apareceu somente uma vez na página
A palavra 'grande' apareceu somente uma vez na página
A palavra 'divertido' apareceu somente uma vez na página
A palavra 'sara' apareceu somente uma vez na página
A palavra 'maria' apareceu somente uma vez na página

Total de vezes "ladies" na página: 4
PS C:\Users\matia\Documents\GitHUBRepo\fculdade>

```