

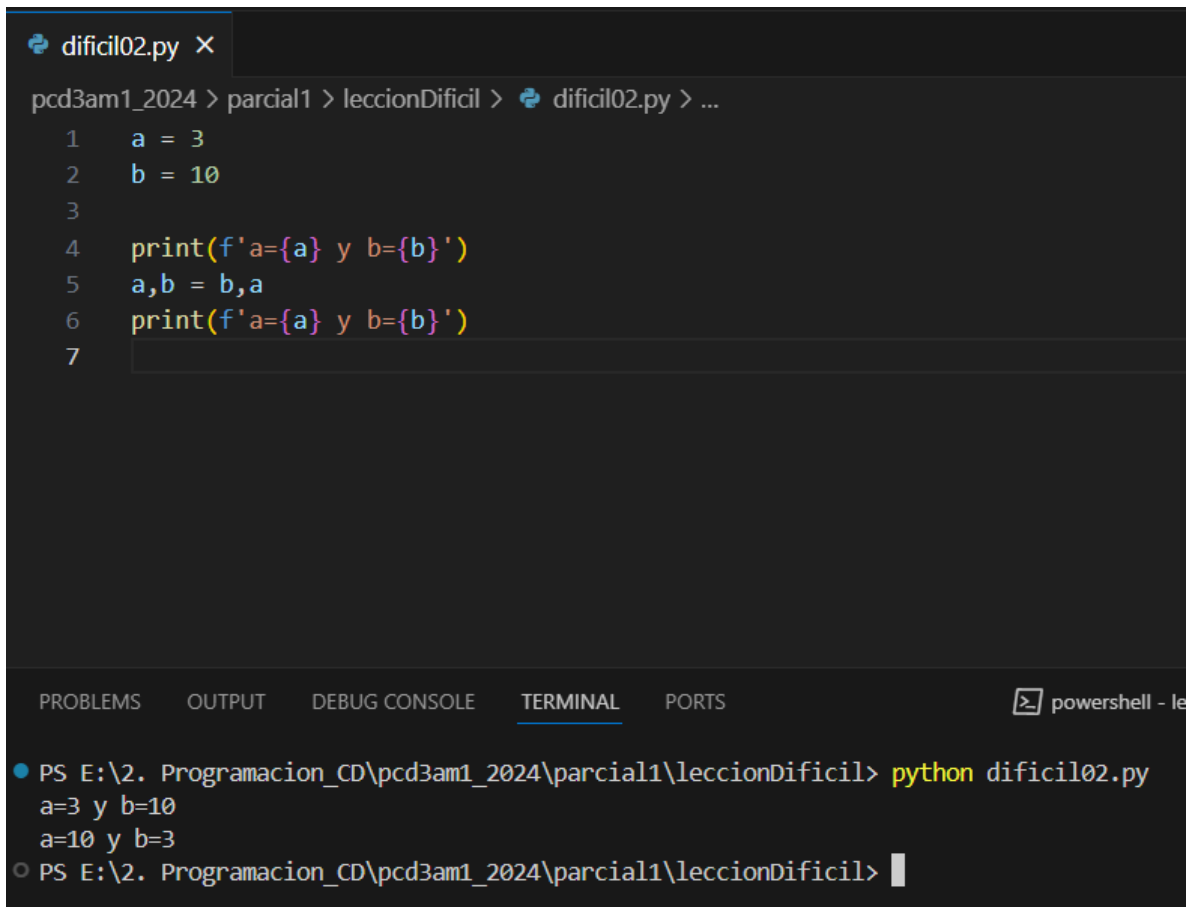
## LECCIÓN DIFÍCIL

difícil01.py

```
difícil01.py ×
pcd3am1_2024 > parcial1 > leccionDifícil > difícil01.py > loMismoPeroConListas
1  def masDeUnDato(x):
2      return x, x**2, x**3
3
4  def loMismoPeroConListas(x):
5      return [x, x**2, x**3]
6
7  x,x2,x3 = masDeUnDato(3)
8  lista = loMismoPeroConListas(3)
9
10 print(x,x2,x3)
11 print(lista)
12

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  powershell
● PS E:\2. Programacion_CD\pcd3am1_2024\parcial1\leccionDifícil> python difícil01.py
3 9 27
[3, 9, 27]
○ PS E:\2. Programacion_CD\pcd3am1_2024\parcial1\leccionDifícil> 
```

difícil02.py



The image shows a code editor window with a file named `difícil02.py`. The code in the editor is as follows:

```
1  a = 3
2  b = 10
3
4  print(f'a={a} y b={b}')
5  a,b = b,a
6  print(f'a={a} y b={b}')
7
```

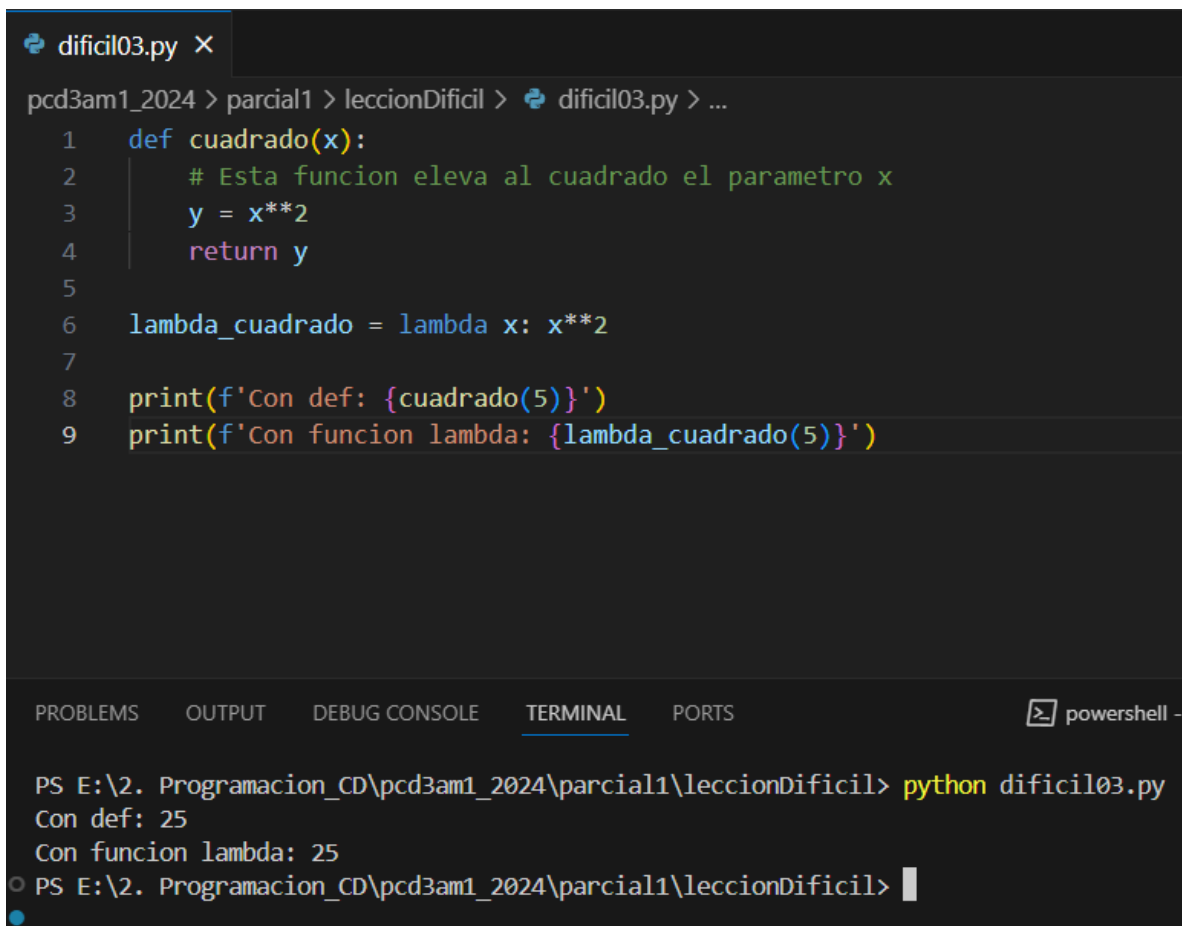
Below the editor is a terminal window. The first command executed is `python difícil02.py`, which produces the output:

```
a=3 y b=10
a=10 y b=3
```

The terminal prompt then returns to the command line.

Aquí se intercambian variables

difícil03.py



The image shows a code editor window with a file named `difícil03.py`. The code defines a function `cuadrado(x)` that squares its input, and a lambda function `lambda_cuadrado` that does the same. Both are used to calculate  $5^2$ . Below the code, a terminal window shows the command `python difícil03.py` being executed, with the output `Con def: 25` and `Con funcion lambda: 25`.

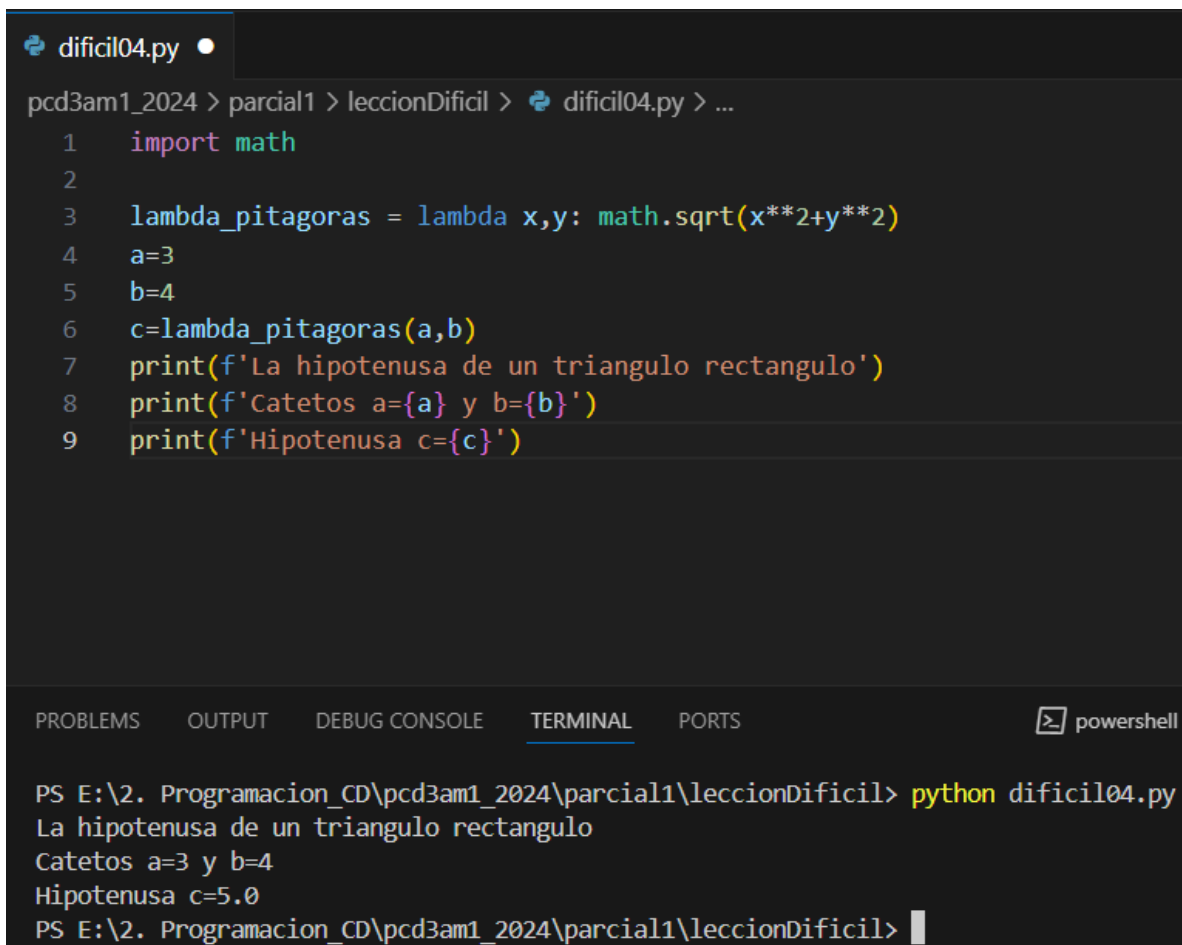
```
pcd3am1_2024 > parcial1 > leccionDifícil > difícil03.py > ...  
1 def cuadrado(x):  
2     # Esta funcion eleva al cuadrado el parametro x  
3     y = x**2  
4     return y  
5  
6 lambda_cuadrado = lambda x: x**2  
7  
8 print(f'Con def: {cuadrado(5)}')  
9 print(f'Con funcion lambda: {lambda_cuadrado(5)}')
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell -

```
PS E:\2. Programacion_CD\pcd3am1_2024\parcial1\leccionDifícil> python difícil03.py  
Con def: 25  
Con funcion lambda: 25  
PS E:\2. Programacion_CD\pcd3am1_2024\parcial1\leccionDifícil>
```

Aquí se hace uso de la función lambda, que reduce la función cuadrado a una sola línea, haciendo que el código sea mas eficiente.

dificil04.py



```

dificil04.py
pcd3am1_2024 > parcial1 > leccionDificil > difcil04.py > ...
1  import math
2
3  lambda_pitagoras = lambda x,y: math.sqrt(x**2+y**2)
4  a=3
5  b=4
6  c=lambda_pitagoras(a,b)
7  print(f'La hipotenusa de un triangulo rectangulo')
8  print(f'Catetos a={a} y b={b}')
9  print(f'Hipotenusa c={c}')

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  powershell

PS E:\2. Programacion_CD\pcd3am1_2024\parcial1\leccionDificil> python difcil04.py
La hipotenusa de un triangulo rectangulo
Catetos a=3 y b=4
Hipotenusa c=5.0
PS E:\2. Programacion_CD\pcd3am1_2024\parcial1\leccionDificil> 
```

Aquí se hace uso de la función lambda, aplicándola para sacar la hipotenusa de dos variables ya declaradas (a, b).

difícil05.py

```
difícil05.py ×
parcial1 > leccionDifícil > difícil05.py > ...
1  import random
2  lambda_paridad = lambda n: 'Par' if n%2==0 else 'Impar'
3
4  lista = [random.randint(1,100) for i in range(5)]
5  print(lista)
6
7  for n in lista:
8      print(f'{n} es {lambda_paridad(n)}')
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```
PS D:\2. Programacion_CD\pcd3am1_2024\parcial1\lecciondifícil> python difícil05.py
[31, 79, 96, 38, 53]
31 es Impar
79 es Impar
96 es Par
38 es Par
53 es Impar
PS D:\2. Programacion_CD\pcd3am1_2024\parcial1\lecciondifícil> █
```

Aquí se hace uso de la librería random, para hacer que genere cinco números aleatorios entre el 1 y el 100, posteriormente se usa la función lambda para saber si esos números generados son pares o impares.

difícil06.py

```
difícil06.py X
parcial1 > leccionDifícil > difícil06.py > ...
1  cadena1 = '31 s3ñ0r d3 l0s 4n1l10s'
2  print(cadena1)
3  print(''.join(['a', 'b', 'c']))
4  x= '3'
5  print(x.isdigit())
6  z='s'
7  print(z.isdigit())
8  c2 = 'hola'
9  for c in c2:
10 |     print(c)
11 print('FIN')
12
13 def sinNumeros(cadena):
14 |     nuevaCadena=''
15 |     for c in cadena:
16 |         if not c.isdigit():
17 |             nuevaCadena = nuevaCadena + c
18 |     return nuevaCadena
19
20 lambda_sinnumeros = lambda cadena: ''.join([c for c in cadena if not c.isdigit()])
21
22 lambda_solonumeros = lambda cadena: ''.join([c for c in cadena if c.isdigit()])
23
24 print(f'cadena sin numeros: {sinNumeros(cadena1)}')
25 print(f'Lambda sin numeros: {lambda_sinnumeros(cadena1)}')
26 print(f'Lambda solo numeros: {lambda_solonumeros(cadena1)}')
27
```

```
PS D:\2. Programacion_CD\pcd3am1_2024\parcial1\lecciondifícil> python difícil06.py
31 s3ñ0r d3 l0s 4n1l10s
abc
True
False
h
h
h
o
h
h
h
o
l
a
FIN
cadena sin numeros: 1 sñr d ls nlls
Lambda sin numeros: 1 sñr d ls nlls
Lambda solo numeros: 33030410
PS D:\2. Programacion_CD\pcd3am1_2024\parcial1\lecciondifícil>
```

difícil07.py

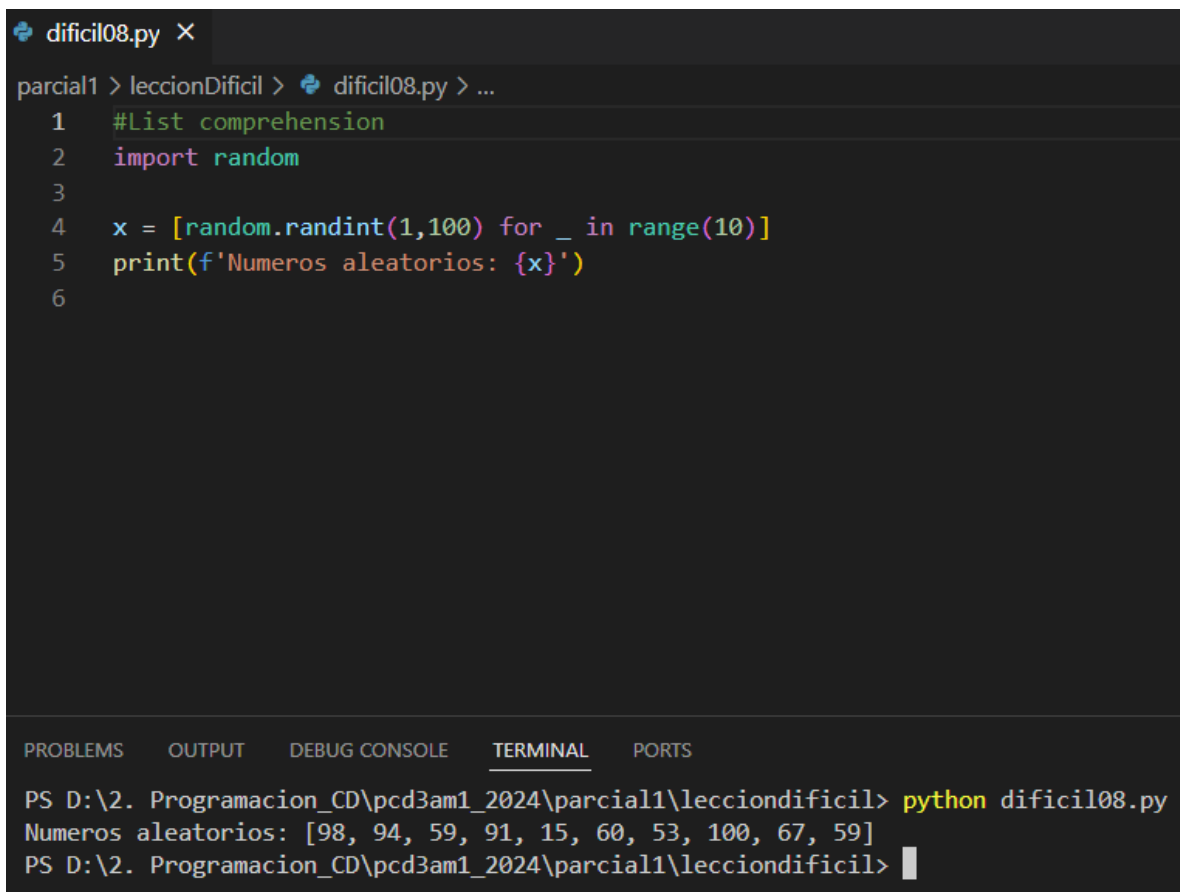
```
difícil07.py
parcial1 > leccionDifícil > difícil07.py > [?] lista
1  lista = [1,2,3,4]
2  print(sum(lista))
3
4  cadena1 = 'abc1de2fghi3jklm4'
5
6  def sumarDigitos(cadena):
7      suma = 0
8      for c in cadena:
9          if c.isdigit():
10             suma = suma + int(c)
11     return suma
12
13     lambda_sumardigitos = lambda cadena: sum([int(c) for c in cadena if c.isdigit()])
14
15     print(f'Sumar digitos: {sumarDigitos(cadena1)}')
16     print(f'Sumar digitos con lambdas: {lambda_sumardigitos(cadena1)}')
17

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  powershell
PS D:\2. Programacion_CD\pcd3am1_2024\parcial1\lecciondifícil> python difícil07.py
10
Sumar digitos: 10
Sumar digitos con lambdas: 10
PS D:\2. Programacion_CD\pcd3am1_2024\parcial1\lecciondifícil> 
```

Primero se declara una lista y posteriormente antes de imprimir se suman los dígitos de esa lista.

Después se declara una cadena de caracteres y primero se implementa una función llamada sumarDigitos que suma los números de esa cadena. Por otro lado, se usa una función lambda para hacer lo mismo que la función pero más eficiente.

difícil08.py



The image shows a code editor window titled 'difícil08.py'. The code is as follows:

```
1  #List comprehension
2  import random
3
4  x = [random.randint(1,100) for _ in range(10)]
5  print(f'Numeros aleatorios: {x}')
6
```

Below the code editor is a terminal window with the following content:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS D:\2. Programacion_CD\pcd3am1_2024\parcial1\lecciondifícil> python difícil08.py
Numeros aleatorios: [98, 94, 59, 91, 15, 60, 53, 100, 67, 59]
PS D:\2. Programacion_CD\pcd3am1_2024\parcial1\lecciondifícil>
```

Se usa la librería `import random` para generar 10 numeros aleatorios entre el 1 y el 100



python09.py

```

difícil09.py X
pcd3am1_2024 > parcial1 > leccionDifícil > difícil09.py > ...
1  # Filter
2  import random
3
4  listaNumeros = [random.randint(1, 100) for _ in range(10)]
5  print(listaNumeros)
6
7  lambda_esPar = lambda x: True if x%2 == 0 else False
8
9  pares = []
10 for x in listaNumeros:
11     #print(x, lambda_esPar(x))
12     if lambda_esPar(x) == True:
13         pares.append(x)
14
15 print(f'Pares: {pares}')
16
17
18 pares2 = list(filter(lambda_esPar, listaNumeros))
19 print(f'Pares filtrados: {pares2}')
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```

PS D:\2. Programacion_CD\pcd3am1_2024\parcial1\leccionDifícil> python difícil09.py
[25, 62, 4, 85, 68, 17, 57, 14, 56, 53]
Pares: [62, 4, 68, 14, 56]
Pares filtrados: [62, 4, 68, 14, 56]
PS D:\2. Programacion_CD\pcd3am1_2024\parcial1\leccionDifícil> 
```

## python10.py

```

difícil10.py M X
pcd3am1_2024 > parcial1 > leccionDifícil > difícil10.py > ...
1  # Map
2  import random
3
4  lambda_cuadrado = lambda x: x**2
5
6  aleatorios = [random.random() for _ in range(5)]
7
8  print(aleatorios)
9  #print(lambda_cuadrado(3))
10 cuadrados1 = []
11 for x in aleatorios:
12     #print(f'{x} : {lambda_cuadrado(x)}')
13     cuadrados1.append(lambda_cuadrado(x))
14
15 print(cuadrados1)
16 cuadrados2 = list(map(lambda_cuadrado, aleatorios))
17 print(cuadrados2)
18

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell
PS D:\2. Programacion_CD\pcd3am1_2024\parcial1\leccionDifícil> python difícil10.py
[0.6236292994521518, 0.18226982895395605, 0.5609806793304355, 0.7844967572701448, 0.10360008240077445]
[0.3889135031351817, 0.03322229054690439, 0.3146993225820369, 0.6154351621673725, 0.010732977073447256]
[0.3889135031351817, 0.03322229054690439, 0.3146993225820369, 0.6154351621673725, 0.010732977073447256]
PS D:\2. Programacion_CD\pcd3am1_2024\parcial1\leccionDifícil>

```

python11.py

```

difcil11.py X
pcd3am1_2024 > parcial1 > leccionDifcil > + difcil11.py > ...
1  import random
2  import functools
3
4  lambda_suma = lambda x,y: x+y
5  print(f'Suma: {lambda_suma(3,5)}')
6
7  numeros = [random.randint(1,10) for _ in range(5)]
8  print(numeros)
9
10 s = functools.reduce(lambda_suma, numeros)
11
12 print(f'Promedio: {s/len(numeros)}')
13

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\2. Programacion_CD\pcd3am1_2024\parcial1\leccionDifcil> python difcil11.py
Suma: 8
[3, 6, 4, 3, 10]
Promedio: 5.2
PS D:\2. Programacion_CD\pcd3am1_2024\parcial1\leccionDifcil> 
```