

Creado por:

Isabel Maniega

Question 1

¿ Cuántos (#) enviará el siguiente fragmento a la consola?

```
var = 1

while var <10:
    print('#')
    var = var << 1
```

- A. uno
- B. ocho
- C. dos
- D. cuatro

Solución 1

```
In [1]: var = 1
while var < 10:
    print('#')
    var = var << 1
# var = 1 --> var * 2 **1 --> 1 * 2 ** 1 --> 2 --> #
# var = 2 --> var * 2 **1 --> 2 * 2 ** 1 --> 4 --> #
# var = 4 --> var * 2 **1 --> 4 * 2 ** 1 --> 8 --> #
# var = 8 --> var * 2 **1 --> 8 * 2 ** 1 --> 16 --> # --> No sigue el

#
#
#
#
```

```
In [2]: # Solución
# D
```

Question 2

¿Cuál es la salida del siguiente fragmento de código?

```
t = [[3-i for i in range(3)] for j in range(3)]
s = 0
for i in range(3):
    s += t[i][i]
print(s)
```

- A. 02
- B. 7
- C. 4
- D. 6

Solución 2

```
In [3]: t = [[3-i for i in range(3)] for j in range(3)] # [[3,2,1], [3,2,1], [3,2,1]]
s = 0
for i in range(3): # 0, 1, 2
    s += t[i][i]
    # 0 --> t[0][0] = 3
    # 1 --> t[1][1] = 2
    # 2 --> t[2][2] = 1
    # 3 + 2 + 1 = 6
print(s)
```

6

```
In [4]: # Solución
# D
```

Question 3

¿Cuál es la salida del siguiente fragmento de código?

```
a = 1
b = 0
c = a & b
d = a | b
e = a ^ b
print(c + d + e)
```

- A. 1
- B. 3
- C. 2
- D. 0

Solución 3

```
In [5]: a = 1
b = 0
c = a & b # 1 & 0 = 0
d = a | b # 1 | 0 = 1
e = a ^ b # 1 ^ 0 = 1

# Resultado : 0 + 1 + 1 = 2
print(c + d + e)
```

2

```
In [6]: # Solución
# C
```

Question 4

¿Cuál es la salida del siguiente fragmento de código?

```
my_list = [1, 2, 3, 4]
print(my_list[-3:-2])
```

- A. [2, 3]
- B. [2, 3, 4]
- C. [2]
- D. []

Solución 4

```
In [7]: my_list = [1, 2, 3, 4]
print(my_list[-3:-2])
```

[2]

```
In [8]: # Solución
# C
```

Question 5

¿Cuáles de los siguientes enunciados son verdaderos?

Selecciona DOS respuestas

```
nums = [1, 2, 3, 4]
vals = nums[-1:-2]
```

- A. vals es más larga que nums
- B. nums es más larga que vals
- C. nums y vals son de la misma longitud
- D. nums y vals son dos listas diferentes

Solución 5

```
In [9]: nums = [1, 2, 3, 4]
vals = nums[-1:-2]
print('nums: ', nums)
print('vals: ', vals)
```

```
nums: [1, 2, 3, 4]
vals: []
```

```
In [10]: # Solución
# B, D
```

Question 6

¿ Cuántos (#) enviará el siguiente fragmento a la consola?

```
var = 0
while var < 6:
    var += 1
    if var % 2 == 0:
        continue
    print('#')
```

- A. tres
- B. dos
- C. uno
- D. cero

Solución 6

```
In [11]: var = 0
while var < 6:
    var += 1
```

```
print("Variable: ", var)
if var % 2 == 0:
    print("Resto 0: ", var, "continua")
    continue
print('imprimo # con variable: ', var)
print('#')
```

```
Variable: 1
imprimo # con variable: 1
#
Variable: 2
Resto 0: 2 continua
Variable: 3
imprimo # con variable: 3
#
Variable: 4
Resto 0: 4 continua
Variable: 5
imprimo # con variable: 5
#
Variable: 6
Resto 0: 6 continua
```

```
In [12]: # Solución
        # A
```

Question 7

Observa el código, y selecciona las sentencias verdaderas:

Selecciona DOS respuestas

```
nums = [1, 2, 3]
vals = nums
del vals[1:2]
```

- A. nums es más larga que vals
- B. nums y vals se refieren a la misma lista
- C. nums y vals son de la misma longitud
- D. nums es replicada y asignada a vals

Solución 7

```
In [13]: nums = [1, 2, 3]
        vals = nums
        del vals[1:2]

        print('nums: ', nums)
        print('vals: ', vals)
```

```
nums: [1, 3]
vals: [1, 3]
```

```
In [14]: # Solución
        # B, C
```

Question 8

¿Qué es IDLE?

- A. Es una versión de Python
- B. Es un acrónimo que significa "Desarrollo Integrado y Entorno de Aprendizaje" para Python
- C. Es un acrónimo que significa "Desarrollo Interactivo y Extensión de Aprendizaje"

Solución 8

<https://es.wikipedia.org/wiki/IDLE>

IDLE (abreviatura de Integrated Development and Learning Environment, en español: «Entorno de desarrollo y aprendizaje integrado») es un entorno de desarrollo integrado para Python

```
In [15]: # Solución
        # B
```

Question 9

¿Qué es CPython?

- A. Un lenguaje compilado utilizado para realizar funciones de programación de alto nivel
- B. Es otro nombre para Cython, un superconjunto del lenguaje de programación Python
- C. La implementación por defecto del lenguaje de programación Python

Solución 9

<https://es.wikipedia.org/wiki/CPython>

CPython es la implementación oficial y más ampliamente utilizada del lenguaje de programación Python. Está escrita en C.

```
In [16]: # Solución  
# C
```

Question 10

¿Cómo llamas a una herramienta que te permite correr tu código paso a paso e inspeccionarlo en cada momento de la ejecución?

- A. Un depurador
- B. Un editor
- C. Una consola

Solución 10

```
In [ ]: # Solución  
# A
```

Question 11

¿Cuáles son los cuatro elementos fundamentales que componen un lenguaje?

- A. Un alfabeto, un léxico, fonética y semántica
- B. Un alfabeto, fonética, fonología y semántica
- C. Un alfabeto, un léxico, una sintaxis y una semántica
- D. Un alfabeto, morfología, fonética y semántica

Solución 11

```
In [ ]: # Solución  
# C
```

Question 12

¿Cómo se llama a un archivo que contiene un programa escrito en un lenguaje de programación de alto nivel?

- A. Un archivo de código
- B. Un archivo de destino
- C. Un archivo fuente
- D. Un archivo de máquina

Solución 12

```
In [ ]: # Solución  
# C
```

Question 13

¿Cómo se llama un intérprete de línea de comandos que te permite interactuar con tu sistema operativo y ejecutar comandos y scripts de Python?

- A. Un editor
- B. Una consola
- C. Un compilador
- D. Jython

Solución 13

```
In [ ]: # Solución  
# B
```

Question 14

El carácter de escape le debe su nombre al hecho de que:

- A. no puede ser atrapado debido a su alta velocidad
- B. cambia el significado del carácter después de él
- C. escapa del archivo fuente hacia la memoria de la computadora

Solución 14

```
In [ ]: # Solución  
# B
```

Question 15

La diferencia más importante entre números enteros y números flotantes es que:

- A. los enteros no pueden ser literales mientras que los flotantes si
- B. se almacenan de manera diferente en la memoria de la computadora
- C. no pueden ser utilizados simultáneamente

Solución 15

```
In [ ]: # Solución  
# B
```

Question 16

Solo una de las siguientes declaraciones es verdadera. ¿Cuál es?

- A. La multiplicación precede la suma
- B. Ninguna de las anteriores
- C. La suma precede la multiplicación

Solución 16

```
In [ ]: # Solución  
# A
```

Question 17

¿Cuál es el resultado del siguiente código?

```
x = 1 / 2 + 3 // 3 + 4 ** 2
print(x)
```

A. 8.5

B. 8

C. 17.5

D. 17

Solución 17

```
In [8]: x = 1 / 2 + 3 // 3 + 4 ** 2
        print(x)
```

17.5

```
In [ ]: # Solución
        # C
```

Question 18

El enlazado del lado izquierdo determina que el resultado de la siguiente expresión:

$$1 // 2 * 3$$

es igual a:

A. 0.0

B. 4.5

C. 0

D. 0.16666666666666666

Solución 18

```
In [9]: 1 // 2 * 3
```

```
Out[9]: 0
```

```
In [ ]: # Solución
        # C
```

Question 19

El significado del parámetro de palabra clave es determinado por:

- A. su valor
- B. su conexión con variables existentes
- C. su posición dentro de la lista de argumentos
- D. el nombre especificado del argumento junto con su valor

Solución 19

```
In [1]: # Solución
        # B

        # https://www.codificandobits.com/curso/python-nivel-basico/20-argumentos
        # https://aulasoftwarelibre.github.io/taller-de-python/Python_Avanzado/Ar
```

Question 20

¿Cuál es el resultado del siguiente fragmento de código si se introduce un 11 y un 4 respectivamente?

```
x = int(input())
y = int(input())

x = x % y
x = x % y
y = y % x

print(y)
```

- A. 3
- B. 1
- C. 2
- D. 4

Solución 20

```
In [4]: x = int(input())
        y = int(input())
```

```
x = x % y # 11 % 4 = 3
x = x % y # 3 % 4 = 3
y = y % x # 4 % 3 = 1

print("Resultado de y: ", y)
```

Resultado de y: 1

```
In [ ]: # Solución
        # B
```

Question 21

¿Cuál es la salida del siguiente fragmento de código?

```
def f(x):
    if x == 0:
        return 0
    return x + f(x - 1)

print(f(3))
```

- A. 1
- B. el código es erróneo
- C. 3
- D. 6

Solución 21

```
In [15]: def f(x):
        if x == 0:
            return 0
        return x + f(x - 1)

        # 3 --> 3 != 0 --> retorna 3 + f(2)
        # 2 --> 2 != 0 --> retorna 3 + 2 + f(1)
        # 1 --> 1 != 0 --> retorna 3 + 2 + 1 + f(0)
        # 0 --> 0 == 0 --> retorna 3 + 2 + 1 + 0
        # retorna 6

print(f(3))
```

6

```
In [ ]: # Solución
        # D
```

Question 22

Una función integrada es una función la cual:

- A. ha sido colocada dentro del código por otro programador
- B. tiene que ser importada antes de ser utilizada
- C. esta oculta a los programadores
- D. viene con Python, y es una parte integral de Python

Solución 22

<https://www.programaenpython.com/miscelanea/funciones-integradas/>

Python tiene funciones integradas que podemos utilizar sin necesidad de importar ningún módulo.

```
In [ ]: # Solución  
        # D
```

Question 23

¿Cuáles de las siguientes afirmaciones son verdaderas?

Selecciona DOS respuestas

- A. El valor None no puede ser empleado fuera de las funciones
- B. El valor None puede ser comparado con otras variables
- C. El valor None puede ser empleado como argumento de operaciones aritméticas
- D. El valor None puede ser asignado a variables

Solución 23

```
In [ ]: # Solución  
        # B, D
```

Question 24

¿Cuál es la salida del siguiente código?

```
try:
    value = input("Ingresa un valor: ")
    print(value/value)
except ValueError:
    print("Entrada incorrecta...")
except ZeroDivisionError:
    print("Entrada errónea...")
except TypeError:
    print("Entrada muy errónea...")
except:
    print("¡Buuu!")
```

- A. Entrada muy errónea...
- B. Entrada errónea...
- C. Entrada incorrecta...
- D. ¡Buuu!

Solución 24

```
In [16]: try:
        value = input("Ingresa un valor: ")
        print(value/value)
    except ValueError:
        print("Entrada incorrecta...")
    except ZeroDivisionError:
        print("Entrada errónea...")
    except TypeError:
        print("Entrada muy errónea...")
    except:
        print("¡Buuu!")
```

Ingresa un valor: 2
Entrada muy errónea...

```
In [5]: value = input("Ingresa un valor: ")
        print(value/value)
```

```

-----
-
TypeError                                Traceback (most recent call last)
/home/isabelmaniega/Documentos/PCEP/PCEP/PCEP_ES-Resuelto.ipynb Cell 136 l
ine 2
      <a href='vscode-notebook-cell:/home/isabelmaniega/Documentos/PCEP/PC
EP/PCEP_ES-Resuelto.ipynb#Y334sZmlsZQ%3D%3D?line=0'>1</a> value = input("I
ngresa un valor: ")
----> <a href='vscode-notebook-cell:/home/isabelmaniega/Documentos/PCEP/PC
EP/PCEP_ES-Resuelto.ipynb#Y334sZmlsZQ%3D%3D?line=1'>2</a> print(value/valu
e)

TypeError: unsupported operand type(s) for /: 'str' and 'str'

```

In []: *# Solución*
A

Question 25

¿Cuál es la salida del siguiente fragmento de código?

```

def any():
    print(var + 1, end=' ')
var = 1
any()
print(var)

```

- A. 21
- B. 11
- C. 12
- D. 22

Solución 25

In [6]: `def any():`
 `print(var + 1, end=' ')`
`var = 1`
`any()`
`print(var)`

2 1

In [7]: *# Solución*
A

Question 26

¿Cuál es la salida del siguiente fragmento de código?

```
dictionary = {'one': 'two', 'three': 'one', 'two':  
             'three'}  
v = dictionary['one']  
for k in range(len(dictionary)):  
    v = dictionary[v]  
print(v)
```

- A. two
- B. ('one', 'two', 'three')
- C. one
- D. three

Solución 26

```
In [18]: dictionary = {'one': 'two', 'three': 'one', 'two': 'three'}  
v = dictionary['one']  
for k in range(len(dictionary)):  
    v = dictionary[v]  
    # Resultado:  
  
    # range(0, 3) --> 0, 1, 2  
    # 0 --> dictionary['two'] --> 'three'  
    # 1 --> dictionary['three'] --> 'one'  
    # 2 --> dictionary['one'] --> 'two'  
  
print(v)
```

two

```
In [ ]: # Solución  
        # A
```

Question 27

Una función definida de la siguiente manera:

Selecciona DOS respuestas

```
def function(x=0):  
    return x
```


- A. debe ser invocada sin argumentos
- B. puede ser invocada exactamente con un argumento
- C. puede ser invocada sin ningún argumento
- D. debe ser invocada exactamente con un argumento

Solución 27

```
In [ ]: # Solución  
# B, C
```

Question 28

¿Cuál es el comportamiento esperado del siguiente programa?

```
try:  
    print(5/0)  
    break  
except:  
    print("Lo siento, algo salió mal...")  
except (ValueError, ZeroDivisionError):  
    print("Mala suerte...")
```

- A. El programa provocará una excepción ZeroDivisión y dará como salida un mensaje de error predeterminado.
- B. El programa provocará una excepción ValueError y dará como salidaa un mensaje de error predeterminado.
- C. El programa provocará una excepción ValueError y dará como salida el siguiente mensaje: Mala suerte...
- D. El programa provocará una excepción de SyntaxError .

Solución 28

```
In [19]: try:  
          print(5/0)  
          break  
except:  
    print("Lo siento, algo salió mal...")  
except (ValueError, ZeroDivisionError):  
    print("Mala suerte...")
```

```
Cell In[19], line 3
      break
      ^
SyntaxError: 'break' outside loop
```

```
In [ ]: # Solución
      # D
```

Question 29

El resultado de la siguiente división:

1 // 2

- A. es igual a 0
- B. es igual a 0.0
- C. no se puede predecir
- D. es igual a 0.5

Solución 29

```
In [20]: 1 // 2
```

```
Out[20]: 0
```

```
In [ ]: # Solución
      # A
```

Question 30

¿Cuáles de los siguientes nombres de variables son ilegales y provocarán una excepción de SyntaxError?

Selecciona DOS respuestas

- A. In
- B. in
- C. print
- D. for

Solución 30

In []: `# Solución`
`# B, D`

Creado por:

Isabel Maniega