

## PRUEBA MÓDULO 3

¡Bien hecho! Has llegado al final del Módulo 3 y has completado una meta importante en tu educación de programación en Python. He aquí un breve resumen de los objetivos que has cubierto y con lo que te has familiarizado en el Módulo 3:

- Valores booleanos para comparar diferentes valores y controlar las rutas de ejecución usando las instrucciones if e if-else;
- la utilización de bucles (while and for) y cómo controlar su comportamiento utilizando las instrucciones break y continue;
- la diferencia entre operaciones lógicas y bit a bit;
- el concepto de listas y procesamiento de listas, incluyendo la iteración proporcionada por el bucle for loop, y las rebanadas;
- la idea de los arreglos multidimensionales.

Ahora estás listo para tomar la **Prueba del Módulo**, que te ayudará a evaluar lo que has aprendido hasta ahora.

La siguiente prueba se basa en lo que acabas de aprender. Hay veinte preguntas en total y debes obtener al menos un 70% para aprobar.

¡Buena suerte!

### Pregunta 1

Un operador que puede verificar si dos valores son iguales se codifica como:

===



==

=

!=

### Pregunta 2

El valor asignado finalmente a x es igual a:

```
1 x = 1
2 x = x == x
3
```

0

False

1



True

### Pregunta 3

¿Cuántos (■) enviará el siguiente fragmento de código a la consola?

```
1 i = 0
2 while i <= 3 :
3     i += 2
4     print("■")
5
```

1

0

3

✓ 2



### Pregunta 4



¿Cuántos (■) enviará el siguiente fragmento de código a la consola?

```
1 i = 0
2 while i <= 5 :
3     i += 1
4     if i % 2 == 0:
5         break
6     print("■")
7
```

0

✓ 1

3

2



### Pregunta 5

¿Cuántos (■) enviará el siguiente fragmento de código a la consola?

```
1 for i in range(1):
2     print("■")
3 else:
4     print("■")
5
```

1

0

✓ 2

3



## Pregunta 6

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

```
1 var = 0
2 while var < 6:
3     var += 1
4     if var % 2 == 0:
5         continue
6     print("##")
7
```

1

2

0

✓ 3

## Pregunta 7

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

```
1 var = 1
2 while var < 10:
3     print("##")
4     var = var << 1
5
```

8

2

1

✓ 4

## Pregunta 8

¿Qué valor será asignado a la variable x?

```
1 z = 10
2 y = 0
3 x = y < z and z > y or y > z and z < y
4
```

False

1

✓ True

0

### Pregunta 9

¿Cuál es la output del siguiente fragmento de código?

```
1 a = 1
2 b = 0
3 c = a & b
4 d = a | b
5 e = a ^ b
6
7 print(c + d + e)
8
```



2

1

3

0

### Pregunta 10

¿Cuál es la output del siguiente fragmento de código?

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

-1

-2



1

3

### Pregunta 11

¿Cuál es la output del siguiente fragmento de código?

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

1

[2, 3, 4]

[2, 3]



[2]

## Pregunta 12

La segunda asignación:

```
1 vals = [0, 1, 2]
2 vals[0], vals[2] = vals[2], vals[0]
3
```



invierte la lista

mantiene la lista igual

extiende la lista

acorta la lista



## Pregunta 13



Después de la ejecución del siguiente fragmento de código, la suma de todos los elementos `vals` será igual a:

```
1 vals = [0, 1, 2]
2 vals.insert(0, 1)
3 del vals[1]
4
```

3



4

5

2



## Pregunta 14

Observa el código, y selecciona las sentencias **verdaderas**: (Selecciona **dos** respuestas)

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



`nums` y `vals` son de la misma longitud



`nums` y `vals` se refieren a la misma lista

`nums` es replicada y asignada a `vals`

`nums` es más larga que `vals`



## Pregunta 15

¿Cuáles de los siguientes enunciados son **verdaderos**? (Selecciona **dos** respuestas)

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

`vals` es más larga que `nums`

`nums` y `vals` son de la misma longitud

☒ `nums` y `vals` son dos listas diferentes

☒ `nums` es más larga que `vals`

## Pregunta 16

¿Cuál es la output del siguiente fragmento de código?

```
1 my_list_1 = [1, 2, 3]
2 my_list_2 = []
3 for v in my_list_1:
4     my_list_2.insert(0, v)
5 print(my_list_2)
6
```

`[1, 1, 1]`

`[1, 2, 3]`

`[3, 3, 3]`

☒ `[3, 2, 1]`

## Pregunta 17

¿Cuál es la output del siguiente fragmento de código?

```
1 my_list = [1, 2, 3]
2 for v in range(len(my_list)):
3     my_list.insert(1, my_list[v])
4 print(my_list)
5
```

`[1, 2, 3, 1, 2, 3]`

☒ `[1, 1, 1, 1, 2, 3]`

`[1, 2, 3, 3, 2, 1]`

`[3, 2, 1, 1, 2, 3]`

### Pregunta 18

¿Cuántos elementos contiene la lista `my_list`?

```
1 my_list = [i for i in range(-1, 2)]
2
```

1

4



3

2

### Pregunta 19

¿Cuál es la output del siguiente fragmento de código?

```
1 t = [[3-i for i in range (3)] for j in range (3)]
2 s = 0
3 for i in range(3):
4     s += t[i][i]
5 print(s)
6
```



6

2

7

4

### Pregunta 20

¿Cuál es la output del siguiente fragmento de código?

```
1 my_list = [[0, 1, 2, 3] for i in range(2)]
2 print(my_list[2][0])
3
```



el fragmento generará un error de ejecución

2

0

1



Has obtenido 100%.

Felicidades, has acreditado el quiz.

