



Curso de Python: Comprehensions, Funciones y Manejo de Errores

¡No te rindas!

Necesitas una **calificación mínima de 9.0** para aprobar.

Vuelve a intentarlo en 05 horas, 37 minutos, 01 segundos

7.5

Calificación

18 / 24

Aciertos

Resumen



1. ¿Cuál de los siguientes es un principio de The Zen of Python?

Plano es mejor que anidado.



2. ¿Cuál de los siguientes es un principio de The Zen of Python?

Bello es mejor que feo.



3. ¿Cuál de las siguientes es una propiedad principal CORRECTA de los Sets o Conjuntos en Python?

No pueden tener duplicados.



4. ¿Cuál de los siguientes es un Set o Conjunto en Python?

```
python_set = {'Nicolas', 'Miguel', 'Juan'}
```



5. ¿Cuál será el resultado del siguiente bloque de código?

```
a = {1,2}
b = {2,3}
print(a | b)
```

```
{1,2,3}
```



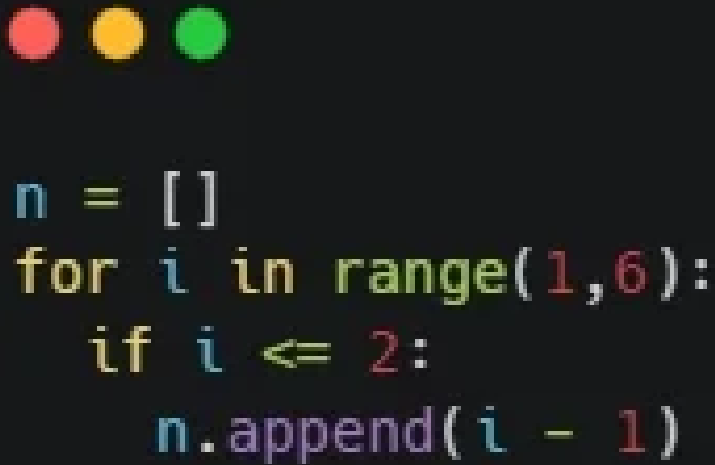
6. ¿Cuál será el resultado del siguiente bloque de código?

```
a = {1,2}
b = {2,3}
print(a & b)
```

```
{2}
```



7. Dado el siguiente bloque de código:



```
n = []  
for i in range(1,6):  
    if i <= 2:  
        n.append(i - 1)
```



¿Cuál de las siguientes respuestas con List Comprehension nos permite obtener el mismo resultado con una sintaxis más corta?

`i = [i - 1 for n in range(1,6) if i <= 2]`

[REPASAR CLASE](#)

8. Dado el siguiente bloque de código:

```
d = {}  
for e in range(1,6):  
    if e <= 2:  
        d[e] = e - 1
```



¿Cuál de las siguientes respuestas con Dictionary Comprehension nos permite obtener el mismo resultado con una sintaxis más corta?

`d = { e: e - 1 for e in range(1,6) if e <= 2 }`

[REPASAR CLASE](#)

9. ¿Cuál de las siguientes estructuras de datos es mutable?

List

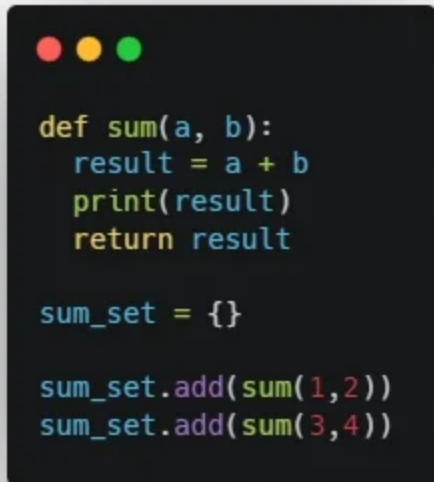


10. ¿Cuál de las siguientes estructuras de datos nos permite ordenar sus elementos?

Tuple



11. Creaste una función sum que recibe 2 parámetros y los suma entre ellos. Ahora debes almacenar el resultado de esas sumas en un conjunto. ¿Cómo lo harías?



```
def sum(a, b):  
    result = a + b  
    print(result)  
    return result  
  
sum_set = {}  
  
sum_set.add(sum(1,2))  
sum_set.add(sum(3,4))
```

Platzi

12. ¿Cuál estructura de datos usa Python para retornar varios valores en una función?

List

[REPASAR CLASE](#)

13. La función sum(a, b) retorna 3 elementos: 1) la suma entre a y b; 2) a; y 3) b. Debes guardar los 3 elementos retornados en nuevas variables por fuera de la función. ¿Cómo lo harías?

```
sum_return = sum(1,2) result = sum_return[0] a = sum_return[1] b = sum_return[2]
```

[REPASAR CLASE](#)

14. Dado el siguiente bloque de código:

```
def sum(x, y):  
    return x + y
```

¿Cuál de las siguientes respuestas nos permite obtener el mismo resultado, pero con una sintaxis más corta?

sum = lambda x, y : x + y



15. ¿Cuál de las siguientes afirmaciones sobre las funciones lambda es correcta?

Todas las respuestas son correctas.



16. ¿Cuál de las siguientes funciones SIEMPRE devuelve la misma cantidad de elementos de la lista original?

map



17. Dado el siguiente bloque de código:

```
original = [1, 2, 3, 4, 5]
```

```
new = []
```

```
for x in original:
```

```
    new.append(x * 2)
```

¿Cuál de las siguientes respuestas es la sintaxis más corta para obtener el mismo resultado?

new = list(map(lambda x: x * 2, original))



18. ¿La función Map en Python puede procesar dos listas al mismo tiempo?

Sí, pero solo iterará en ambas listas hasta la cantidad de elementos de la lista más pequeña.



19. ¿Cómo se importa el módulo functools (nativo de Python)?

```
import functools
```



20. Creaste una función sum dentro del archivo math.py. Ahora desde el archivo main.py (en la misma carpeta que math.py) debes importar ese módulo y usar la función sum. ¿Cómo lo harías?

```
import math math.sum(1,2)
```



21. ¿Cuál es la función del siguiente bloque de código: if __name__ == '__main__'?

Nos informa si el archivo está cargando dependencias de otros módulos loc [REPASAR CLASE](#)

22. ¿Cuál de los siguientes bloques de código provoca un error en Python?

Todas las respuestas arrojan un error.



23. Dado el siguiente bloque de código: print(10 / user_option). Y siendo user_option una variable que puede recibir cualquier valor numérico (incluyendo 0). ¿Cómo manejarías la excepción de división sobre cero para que el código siga ejecutando a pesar del error?

```
try:
    print(10 / user_option)
except ZeroDivisionError as error:
    print(error) ````
```



24. ¿Cuál es la principal característica de una función Lambda o anónima?

Que no hace falta invocarlas para ser utilizadas.

[REPASAR CLASE](#)

[Ver menos](#)

Siguiente curso que podría interesarte



Curso de Python: PIP y Entornos Virtuales

Por: Nicolas Molina



Curso de FastAPI

Por: Eduardo Álvarez

[Curso de Python: PIP y Entornos Virtuales](#) →

[Volver al Home](#)