

Pregunta **1**

Sin responder
aún

Puntúa como
1,00

La estructura for es una redundancia de Python, igual de cómodo es un while

Seleccione una:

- ☐ a. Sí, la única diferencia son dos números
- ☒ b. Es muy similar si se trata de recorrer números, no si se trata de otros elementos.
- ☐ c. No es una redundancia pero puede que lo eliminen en la próxima versión.
- ☐ d. No, de hecho, while es mejor.

Pregunta **2**

Sin responder
aún

Puntúa como
1,00

for i in range (1,4):

Seleccione una:

- ☐ a. range sobra
- ☐ b. i puede valer de 0 a 3, la resta entre 4 y 1.
- ☐ c. Está mal expresada, da error de tipos, 1,4 es un decimal
- ☒ d. i nunca llegará a valer 4

Pregunta **3**

Sin responder
aún

Puntúa como
1,00

Una lista:

Seleccione una:

- ☐ a. Nunca podrá salir por pantalla por medio de un print. Harán falta tantos como elementos.
- ☐ b. Es una variable que sólo puede tener un tipo de variables. Python es fuertemente tipado.
- ☒ c. Tiene órdenes propias que no comparte con la variable tipo Cadena de caracteres o string
- ☐ d. No se puede alterar durante el programa.

Pregunta **4**

Sin responder
aún

Puntúa como
1,00

for i in range (4,4):

Seleccione una:

- ☒ a. No se ejecutará
- ☐ b. Se ejecuta 4 veces
- ☐ c. Se ejecuta si i es 4
- ☐ d. Se ejecutará una vez

Pregunta **5**

Sin responder
aún

Puntúa como
1,00

Los comentarios

Seleccione una:

- ☒ a. IDLE los saca por pantalla pero en otro color para comentar el programa
- ☐ b. Son un tipo de variable
- ☐ c. Deben explicar lo más detalladamente lo que realiza el programa y cómo lo hace para poder releerlo bien.
- ☐ d. No pueden tener números en sus líneas

Pregunta **6**

Sin responder
aún

Puntúa como
1,00

```
lista=[1,2,3]
```

Seleccione una:

- ☒ a. Es una lista cuyo término 0 es 1
- ☐ b. Las listas así declaradas no se pueden alterar.
- ☐ c. Es una lista cuyo término 1 es 1
- ☐ d. Está mal escrita porque una lista debe contener variables

Pregunta **7**

Sin responder
aún

Puntúa como
1,00

¿Qué realiza la siguiente línea: "while not edad.isdigit():" ?

Seleccione una:

- ☒ a. Repite las órdenes tras while hasta que edad sea un número
- ☐ b. No realiza nada, tiene un error.
- ☐ c. Repite la orden tras while si la edad no tiene un dígito sino varios.
- ☐ d. Repite las órdenes tras while si edad tiene un dígito

Pregunta **8**

Sin responder
aún

Puntúa como
1,00

```
print(str(presentes[elegido-1]))
```

Seleccione una:

- ☐ a. Sacará por pantalla elegido-1 si presente es una Cadena de caracteres
- ☐ b. elegido-1 es lo que sacará por pantalla print si está en la lista presentes
- ☒ c. elegido-1 es el número de orden que ocupa elegido ya que el usuario empieza a contar en 1 y Python en 0
- ☐ d. Da error si presentes[elegido-1] es una Cadena de caracteres

Pregunta **9**

Sin responder
aún

Puntúa como
1,00

En la solución al ejercicio de autoevaluación 4: La orden
`posicion=posicion+1`

Seleccione una:

- ☐ a. Se añade 1 cada vez que se cambia de número en la lista para saber la posición que ocupa
- ☐ b. Se le suma 1 sólo si encuentra el número en la lista
- ☐ c. La variable `posicion` se podía haber sustituido por `i`
- ☒ d. Es propio de malos hábitos en programación.

Pregunta **10**

Sin responder
aún

Puntúa como
1,00

`for i in numeros:`

Seleccione una:

- ☐ a. Sólo se ejecuta si `i`, que es una variable fija, está entre los límites de `numeros`
- ☒ b. `i` va a recorrer la lista `numeros` y, cada vez que esté dentro, se va a ejecutar el `for`
- ☐ c. Va a repetir las órdenes que hay tras el `for` tantas veces como indique `numeros`
- ☐ d. Está mal escrita.