

✓ **¡Felicitaciones! ¡Aprobaste!**

Calificación recibida 100 % Calificación del último envío 100 % Para Aprobar 80 % o más

[Ir al siguiente elemento](#)

1. Completa el espacio en blanco: Si usas código Python para reducir el esfuerzo manual necesario para gestionar una lista de control de acceso, este es un ejemplo de \_\_\_\_\_.

1 / 1 punto

- ☐ reasignación
- ☐ análisis de datos
- ☐ depuración
- ☒ automatización

✓ Correcto

2. El propósito del siguiente código es imprimir un mensaje "Intentando conexión" mientras el valor de la variable count sea inferior a 10. El valor de count debe aumentar en 1 con cada iteración del bucle. ¿Qué está mal en el código? Selecciona todas las opciones que correspondan.

1 / 1 punto

```
count = 1

while count < 10:

print("Intentando conexión")

count = count + 1
```

- ☐ La línea con `count = 1` no tiene sangría
- ☒ La línea con `count = count + 1` no tiene sangría.

✓ Correcto

- ☐ La línea con `while count < 10:` no tiene sangría.
- ☒ La línea con `print("Intentando conexión")` no tiene sangría.

✓ Correcto

3. Completa el espacio en blanco: Los datos de cadena \_\_\_\_\_.

1 / 1 punto

- ☐ deben incluir un punto decimal
- ☐ deben ir entre corchetes
- ☐ deben ir entre paréntesis
- ☒ deben ir entre comillas

✓ Correcto

4. ¿Qué tipo de datos tiene siempre un valor `True` (verdadero) o `False` (false)?

1 / 1 punto

- ☐ Lista
- ☐ Flotante
- ☒ Booleano
- ☐ Cadena

✓ Correcto

5. ¿Cómo se asigna el valor de cadena "xtp3426" a una variable llamada `device_id`?

1 / 1 punto

☐ device\_id = "xtp3426"

- ☐ `device_id(rtp3426)`
- ☐ `device_id = rtp3426`
- ☐ `device_id("rtp3426")`
- ☒ `device_id = "rtp3426"`

✓ Correcto

6. ¿Qué hará este código cuando lo ejecutes?

1 / 1 punto

```
var2 = ["a", "b", "c"]  
  
var2_type = type(var2)  
  
print(var2_type)
```

- ☐ Cambiar el tipo de datos de `var2`
- ☐ Imprimir la cadena `"var2_type"` en la pantalla
- ☐ Imprimir los caracteres `"a"`, `"b"` y `"c"` en la pantalla
- ☒ Indicar que `var2` contiene datos de lista

✓ Correcto

7. Deseas verificar la cadena almacenada en una variable `update_status`. Cuando contiene un valor de `"incompleto"`, deseas imprimir un mensaje de `"programar actualización"`. En este momento, esta sentencia condicional no es correcta. ¿Cuáles son los problemas de esta sentencia condicional? Selecciona todas las opciones que correspondan.

1 / 1 punto

```
if update_status != "incompleto"  
print("programar actualización")
```

- ☐ Debería haber comillas alrededor de la variable `update_status`.
- ☒ El operador no debería ser `!=`, sino `==`.

✓ Correcto

- ☐ La línea con `print("programar actualización")` no debería tener sangría.
- ☒ Faltan dos puntos `:` al final del encabezado condicional.

✓ Correcto

8. Escribiste el siguiente código:

1 / 1 punto

```
if operating_system == "OS 3":  
  
print("Se necesitan actualizaciones")
```

Deseas agregar que imprima un mensaje `"No se necesitan actualizaciones"` siempre que el valor de `operating_system` no sea `"OS 3"`. ¿Qué líneas de código tienen la sintaxis correcta para hacer esto?

- ☐ `else`  
  
`print("No se necesitan actualizaciones")`
- ☐ `else operating_system != "OS 3":`  
  
`print("No se necesitan actualizaciones")`
- ☒ `else:`  
  
`print("No se necesitan actualizaciones")`
- ☐ `elif operating_system == "OS 3":`  
  
`print("No se necesitan actualizaciones")`

✓ Correcto

9. ¿Qué hace esta sentencia iterativa?

1 / 1 punto

```
for i in [0, 5]:
```

```
    print(i)
```

- ☒ Muestra los números enteros 0 y 5
- ☐ Muestra el número entero 0
- ☐ Muestra los números enteros 0, 1, 2, 3, y 4
- ☐ Muestra los números enteros 0, 1, 2, 3, 4, y 5

✓ Correcto

10. ¿Cuántas veces imprimirá el mensaje "advertencia" el siguiente código?

1 / 1 punto

```
count = 1
```

```
while count < 5:
```

```
    print("advertencia")
```

```
    count = count + 1
```

- ☐ 0
- ☐ 1
- ☒ 4
- ☐ 5

✓ Correcto