Ámbito de variables: local y global

Nombre: _____

Fecha: _____

Puntaje: _____

1.

¿Qué es una variable global?

1. Una variable declarada dentro de una función
2. Una variable que solo puede ser usada en la función donde fue creada
3. Una variable declarada fuera de cualquier función

2.

¿Qué es una variable local?

1. Una variable declarada dentro de una función
2. Una variable declarada al inicio del programa
3. Una variable declarada con la palabra clave 'global'
4. Una variable que no puede ser modificada

3.

¿Cuál es la palabra clave que permite modificar una variable global dentro de una función en Python?

Respuesta: _____

4.

¿Qué ocurre si intentas acceder a una variable local fuera de la función donde fue definida?

1. Se produce un error (NameError)
2. Se devuelve el valor None
3. Se obtiene el valor de la variable global con el mismo nombre
4. La variable es accesible sin problema

5.

Si dentro de una función existe una variable local con el mismo nombre que una global, ¿cuál prevalece?

1. La local
2. La global
3. Ambas coexisten con el mismo valor

6.

¿Cuál es la salida del siguiente código? x = 10; def func(): x = 5; func(); print(x)

Respuesta: _____

7.

¿Cómo se modifica una variable global dentro de una función en Python?

1. Usando la palabra clave 'global'
2. Usando la palabra clave 'nonlocal'
3. Asignándole directamente un nuevo valor
4. Usando la palabra clave 'static'

8.

En Python, si declaras una variable dentro de una función sin usar la palabra 'global', ¿cómo se trata esa variable?

1. Como variable local
2. Como variable global
3. Como variable de clase

9.

¿Cuál es la salida del siguiente código? y = 20; def test(): print(y); test()

Respuesta: _____

10.

En Python, ¿cuál es el ámbito de una variable declarada dentro de un bucle for?

1. Solo dentro del bucle
2. Dentro de la función que contiene el bucle
3. Global
4. De clase

11.

¿Qué hace la palabra clave 'nonlocal' en Python?

1. Se refiere a la variable global
2. Se refiere a la variable del ámbito de la función externa
3. Se refiere a la variable local actual
4. Declara una variable estática

12.

¿Cuál es la salida del siguiente código? a = 1; def outer(): a = 2; def inner(): a = 3; print(a); inner(); outer()

Respuesta: _____

13.

Si dentro de una función asignas un valor a una variable local que tiene el mismo nombre que una global, ¿qué sucede con la global?

1. Permanece sin cambios
2. Se actualiza al nuevo valor
3. Se produce un error de compilación

14.

¿Cuál es la salida del siguiente código? x = 1; def f(): x = 2; f(); print(x)

1. 1
2. 2
3. 0
4. Error de ejecución

15.

¿Cuál de las siguientes afirmaciones sobre variables locales y globales es correcta?

1. Las variables locales pueden ser accedidas desde cualquier parte del programa
2. Las variables globales solo son accesibles dentro de las funciones
3. Las variables locales sombrean a las globales del mismo nombre dentro de la función
4. Ambos tipos de variables se almacenan en la misma memoria

16.

¿Qué error se produce en el siguiente código? x = 10; def f(): print(x); x = 5; f()

1. NameError
2. UnboundLocalError
3. SyntaxError
4. No se produce error, imprime 5

17.

¿Cuál es la salida del siguiente código? x = 10; def f(): global x; x = 5; x = 10; f(); print(x)

1. 10
2. 5
3. Error
4. None

18.

¿Cuántas variables diferentes existen en el siguiente código? num = 1; def g(): num = 2; g()

1. 1
2. 2
3. 3
4. 4

19.

En el siguiente código, ¿qué valor se imprime? def f(): return x + 1; x = 2; print(f())

Respuesta: _____

20.

¿Cuál de las siguientes opciones define correctamente el término 'ámbito' (scope) de una variable?

1. La parte del programa donde la variable es accesible
2. El tamaño en memoria que ocupa la variable
3. El tipo de dato de la variable
4. El valor actual de la variable