Definir una función en Python

Nombre: _____

Fecha: _____

Puntaje: _____

1.

¿Cuál es la palabra clave utilizada para definir una función en Python?

1. func
2. define
3. def
4. lambda

2.

¿Cuál de las siguientes es la sintaxis correcta para definir una función llamada 'saludar' que imprime 'Hola'?

1. def saludar(): print('Hola')
2. function saludar(): print('Hola')
3. def saludar: print('Hola')
4. def saludar() print('Hola')

3.

Después de la línea de cabecera de una función (por ejemplo, def saludar():), ¿qué debe venir a continuación?

1. Una línea en blanco
2. Un bloque de código indentado
3. Un comentario

4.

Para llamar a la función 'saludar', debes escribir: __ (incluyendo paréntesis)

Respuesta: _____

5.

¿Qué imprime el siguiente código?

```
def saludar():
print('Hola')
saludar()
```

1. Nada
2. Hola
3. Error de sintaxis
4. saludar

6.

Si defines una función pero nunca la llamas, ¿qué ocurre cuando ejecutas el programa?

1. La función se ejecuta automáticamente
2. No ocurre nada hasta que la llames
3. Ocurre un error
4. El programa imprime la definición de la función

7.

¿Qué valor devuelve por defecto una función que no contiene una sentencia return?

Respuesta: _____

8.

¿Cuál es la forma correcta de documentar una función usando un docstring?

1. Esta función imprime un saludo
2. """Esta función imprime un saludo""" (colocado como primera línea dentro de la función)
3. 'Esta función imprime un saludo'
4. // Esta función imprime un saludo

9.

¿Cuál de los siguientes nombres de función sigue la convención de estilo snake_case?

1. saludoBienvenida
2. saludo-bienvenida
3. saludo_bienvenida
4. SaludoBienvenida

10.

Si olvidas los dos puntos al final de la cabecera de una función (ej. def saludar()), ¿qué tipo de error se produce?

Respuesta: _____

11.

¿Qué sucede si llamas a una función como 'saludar' sin los paréntesis?

1. La función se ejecuta
2. Se obtiene una referencia al objeto función
3. Ocurre un error
4. No pasa nada

12.

¿Cuál de las siguientes funciones mostrará 'Hola' en pantalla al ser llamada?

1. def f(): return 'Hola'
2. def f(): print('Hola')
3. def f(): 'Hola'
4. def f(): print('Hola') y return

13.

¿Cuál de las siguientes definiciones de función es válida si se desea incluir dos instrucciones?

1. def f(): print(1); print(2)
2. def f(): print(1) print(2)
3. def f(): print(1), print(2)
4. def f(): print(1) then print(2)

14.

La indentación estándar recomendada en Python es de _____ espacios.

Respuesta: _____

15.

¿Cuál de las siguientes define correctamente una función que imprime 'Buenos días'?

1. def buenos_dias(): print('Buenos días')
2. def buenos_dias: print('Buenos días')
3. def buenos_dias(): print(Buenos días)
4. def buenos_dias() print('Buenos días')

16.

Para definir una función, el bloque de código dentro de la función debe estar __ (una palabra).

Respuesta: _____

17.

Si llamas a una función exactamente una vez, ¿cuántas veces se ejecuta el cuerpo de la función?

1. 0
2. 1
3. 2

18.

En Python, los dos puntos al final de la cabecera de una función indican:

1. El final de la función
2. El inicio de un bloque indentado
3. Un comentario
4. Un separador opcional

19.

¿Puede una función contener más de una instrucción print?

1. Sí
2. No
3. Solo si usa return

20.

¿Cuál de los siguientes NO es un nombre de función válido en Python?

1. _saludar
2. saludar1
3. 1saludar
4. saludar_nuevo