Programación con Python nivel avanzado

Reactivos

Pregunta 1

En Python, ¿cuál de los siguientes operadores se utiliza en Python para evaluar si dos valores son iguales?

@ Revisa la explicación del tema 1.

Opciones:

a. <
b. >
c. !=
d. ==

Pregunta 2

En Python, ¿cuál de los siguientes operadores lógicos evalúa si al menos una de las condiciones es verdadera?

@ Revisa la explicación del tema 1

Opciones:

a. not
b. and
c. or
d. xor

Pregunta 3

¿Cuál de las siguientes afirmaciones sobre Python es verdadera?

@ Revisa la explicación del tema 1.

Opciones:

a. Python no puede ser utilizado en Windows.
b. Python puede ser utilizado en cualquier plataforma o sistema operativo existente.
c. Python requiere al menos 2 GB de RAM para funcionar.
d. Python no puede ser utilizado en Linux.

Pregunta 4

¿Cómo se ejecuta un programa de Python desde la terminal?

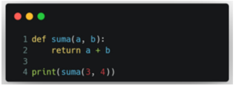
@ Revisa la explicación del tema 1.

Opciones:

a. Usando el comando python nombre_del_programa.py.
b. Usando el comando run python.
c. Usando el comando execute python.
d. Usando el comando start python.

Pregunta 5

Supón que tienes el siguiente código en Python:



```
def suma(a, b):
    return a + b

print(suma(3, 4))
```

¿Cuál será la salida al ejecutar este código?

@ Revisa la explicación del tema 1

Opciones:

a. Error
b. 1
c. 12
d. 7

Pregunta 6

¿Cuál de las siguientes estructuras de datos en Python es inmutable?

@ Revisa la explicación del tema 2

Opciones:

a. Conjuntos (set)
b. Diccionarios
c. Listas
d. Tuplas

Pregunta 7

En Python, ¿cuál de las siguientes funciones se utiliza para determinar la longitud de una lista?

@ Revisa la explicación del tema 2

Opciones:

a. append()
b. count()
c. len()
d. insert()

Pregunta 8

¿Cuál de las siguientes afirmaciones sobre las funciones en Python es verdadera?

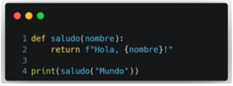
@ Revisa la explicación del tema 2

Opciones:

a. Las funciones no pueden ser reutilizadas.
b. Las funciones permiten desarrollar programas más simples y fáciles de entender, probar y corregir.
c. Las funciones no permiten que el código sea más legible.
d. Las funciones no pueden devolver un resultado.

Pregunta 9

Supón que tienes el siguiente código en Python:



```python
def saludo(nombre):
    return f"Hola, {nombre}!"

print(saludo("Mundo"))
```

¿Cuál será la salida al ejecutar este código?

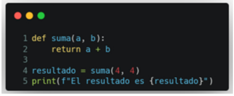
@ Revisa la explicación del tema 2

Opciones:

a. Hola, Mundo!
b. Hola, nombre!
c. Hola, Python!
d. Error

Pregunta 10

Supón que tienes el siguiente código en Python:



```python
def suma(a, b):
    return a + b

resultado = suma(4, 4)
print(f"El resultado es {resultado}")
```

¿Cuál será la salida al ejecutar este código?

@ Revisa la explicación del tema 2

Opciones:

a. El resultado es 13
b. El resultado es 1
c. Error
d. El resultado es 8

Pregunta 11

En Python, ¿qué es un módulo?
@ Revisa la explicación del tema 3

Opciones:

a. Un tipo de error.
b. Un tipo de estructura de dato.
c. Un bloque de código que no puede ser reutilizado.
d. Un conjunto de funciones y estructuras existentes que se pueden importar para optimizar el tiempo de codificación.

Pregunta 12

En Python, ¿cómo se importa un módulo?
@ Revisa la explicación del tema 3

Opciones:

a. Usando el comando "include".
b. Usando el comando "export".
c. Usando el comando "import".
d. Usando el comando "module".

Pregunta 13

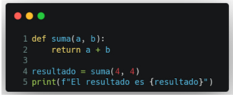
En Python, ¿qué es un paquete?
@ Revisa la explicación del tema 3

Opciones:

a. Un bloque de código que no puede ser reutilizado.
b. Un conjunto de módulos.
c. Un tipo de error.
d. Un tipo de estructura de datos.

Pregunta 14

Supón que tienes el siguiente código en Python:



```python
def suma(a, b):
    return a + b

resultado = suma(4, 4)
print(f"El resultado es {resultado}")
```

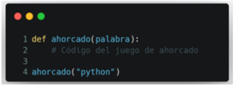
¿Qué hace este código?
@ Revisa la explicación del tema 3

Opciones:

a. Define una función llamada “ahorcado” y luego la llama con el argumento "python".
b. Define una variable llamada “ahorcado” y luego la imprime.
c. Define una clase llamada “ahorcado” y luego crea una instancia dentro de la clase.
d. Define una función llamada python y luego la llama con el argumento "ahorcado".

Pregunta 15

Supón que tienes el siguiente código en Python:



```python
def ahorcado(palabra):
    # Código del juego de ahorcado

ahorcado("python")
```

¿Cuál será el resultado al ejecutar este código?

@ Revisa la explicación del tema 3

Opciones:

a. Cuenta el número de palabras que se incluyen en la lista.
b. Ordena las palabras de la lista "palabras".
c. Imprime todas las palabras de la lista "palabras".
d. Selecciona una palabra al azar de la lista "palabras".

Pregunta 16

¿Qué es un objeto en la programación orientada a objetos en Python?

@ Revisa la explicación del tema 4

Opciones:

a. Un bloque de código que no puede ser reutilizado.
b. Un tipo de error en Python.
c. Una instancia de una clase.
d. Una función que se puede reutilizar.

Pregunta 17

En Python, ¿qué es el método init()?

@ Revisa la explicación del tema 4

Opciones:

a. Es un método que se utiliza para calcular la longitud de una lista.
b. Es un método que se invoca cada vez que se instancia un nuevo objeto.
c. Es u método que se utiliza para agregar un elemento a una lista.
d. Es un método que se utiliza para eliminar un elemento de una lista.

Pregunta 18

¿Qué es la herencia en la programación orientada a objetos en Python?

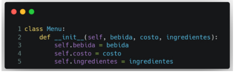
@ Revisa la explicación del tema 4

Opciones:

a. Un proceso por el cual una clase puede obtener los atributos y métodos de otra clase.
b. Un proceso por el cual una clase puede eliminar los atributos y métodos de otra clase.
c. Un proceso por el cual una clase puede modificar los atributos y métodos de otra clase.
d. Un proceso por el cual una clase puede ocultar los atributos y métodos de otra clase.

Pregunta 19

Supón que tienes el siguiente código en Python:



```
class Menu:
    def __init__(self, bebida, costo, ingredientes):
        self.bebida = bebida
        self.costo = costo
        self.ingredientes = ingredientes
```

¿Qué hace este código?

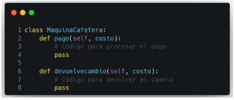
@ Revisa la explicación del tema 4

Opciones:

a. Crea una máquina de café.
b. Define una función llamada "Menu".
c. Crea un menú de comidas.
d. Define una clase llamada "Menu" con atributos para bebida, costo e ingredientes.

Pregunta 20

Supón que tienes el siguiente código en Python:



```python
class MaquinaCafetera:
    def pago(self, costo):
        # Código para procesar el pago
        pass

    def devuelvecambio(self, costo):
        # Código para devolver el cambio
        pass
```

Si deseas procesar un pago por $12.80, ¿cómo llamarías al método correspondiente?

@ Revisa la explicación del tema 4

Opciones:

a. MaquinaCafetera.devuelvecambio(12.80)
b. MaquinaCafetera(12.80)
c. MaquinaCafetera.pago(12.80)
d. pago(12.80)

Pregunta 21

En Python, ¿qué hace el código conn.commit()?

@ Revisa la explicación del tema 5

Opciones:

a. Crea una nueva base de datos.
b. Guarda los cambios en la base de datos.
c. Elimina una base de datos.
d. Recupera datos de la base de datos.

Pregunta 22

En Python, ¿qué hace el código cursor.execute()?

@ Revisa la explicación del tema 5

Opciones:

a. Ejecuta una consulta SQL.
b. Crea una nueva base de datos.
c. Guarda los cambios en la base de datos.
d. Elimina una base de datos.

Pregunta 23

En Python, ¿qué hace el código conn.close()?

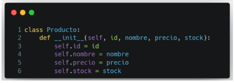
@ Revisa la explicación del tema 5

Opciones:

a. Guarda los cambios en la base de datos.
b. Ejecuta una consulta SQL.
c. Crea una nueva base de datos.
d. Cierra la conexión con la base de datos.

Pregunta 24

Supón que tienes el siguiente código en Python:



```python
class Producto:
    def __init__(self, id, nombre, precio, stock):
        self.id = id
        self.nombre = nombre
        self.precio = precio
        self.stock = stock
```

¿Qué hace este código?

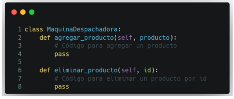
@ Revisa la explicación del tema 5

Opciones:

a. Crea una máquina de café.
b. Define una clase llamada "Producto" con atributos para id, nombre, precio y stock.
c. Define una función llamada "Producto".
d. Crea un producto con id, nombre, precio y stock.

Pregunta 25

Supón que tienes el siguiente código en Python:



```python
class MaquinaDespachadora:
    def agregar_producto(self, producto):
        # Código para agregar un producto
        pass

    def eliminar_producto(self, id):
        # Código para eliminar un producto por id
        pass
```

Si deseas agregar un producto a la máquina despachadora, ¿cómo llamarías al método correspondiente?

@ Revisa la explicación del tema 5

Opciones:

a. MaquinaDespachadora(producto)
b. agregar_producto(producto)
c. MaquinaDespachadora.agregar_producto(producto)
d. MaquinaDespachadora.eliminar_producto(producto)

Pregunta 26

En Python, ¿qué hace la función read()?

@ Revisa la explicación del tema 6

Opciones:

a. Lee el contenido de un archivo.
b. Crea un nuevo archivo.
c. Elimina un archivo.
d. Escribe en un archivo.

Pregunta 27

En Python, ¿qué hace la función readlines()?

@ Revisa la explicación del tema 6

Opciones:

a. Escribe en un archivo.
b. Elimina un archivo.
c. Crea un nuevo archivo.
d. Lee el contenido de un archivo línea por línea.

Pregunta 28

¿Qué es JSON en el contexto de la programación?

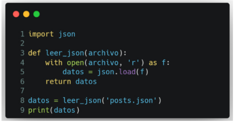
@ Revisa la explicación del tema 6

Opciones:

a. Un tipo de error en Python.
b. Un lenguaje de programación.
c. Un formato para el intercambio de datos.
d. Un bloque de código que no puede ser reutilizado.

Pregunta 29

Supón que tienes el siguiente código en Python:



```python
import json

def leer_json(archivo):
    with open(archivo, 'r') as f:
        datos = json.load(f)
    return datos

datos = leer_json('posts.json')
print(datos)
```

¿Qué hace este código?

@ Revisa la explicación del tema 6

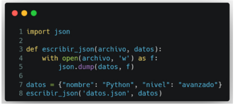
Opciones:

a. Escribe datos en un archivo JSON.
b. Lee datos de un archivo JSON y los imprime.
c. Crea un nuevo archivo JSON.

d. Elimina un archivo JSON.

Pregunta 30

Supón que tienes el siguiente código en Python:



```python
import json

def escribir_json(archivo, datos):
    with open(archivo, 'w') as f:
        json.dump(datos, f)

datos = {"nombre": "Python", "nivel": "avanzado"}
escribir_json('datos.json', datos)
```

¿Qué hace este código?

@ Revisa la explicación del tema 6

Opciones:

a. Escribe datos en un archivo JSON.
b. Lee datos de un archivo JSON.
c. Crea un nuevo archivo JSON.
d. Elimina un archivo JSON.

Pregunta 31

En Flask, ¿cómo se declara una ruta con más de un argumento o parámetro?

@ Revisa la explicación del tema 7

Opciones:

a. Los argumentos se declaran separados por puntos.
b. Los argumentos se declaran separados por espacios.
c. Los argumentos se declaran separados por comas.
d. Los argumentos se declaran separados por diagonales y en el orden deseado.

Pregunta 32

En Flask, ¿qué hace el código render_template()?

@ Revisa la explicación del tema 7

Opciones:

a. Elimina un archivo HTML.
b. Crea un nuevo archivo HTML.
c. Renderiza una plantilla HTML.
d. Escribe en un archivo HTML.

Pregunta 33

En Flask, ¿qué hace el código app.route()?

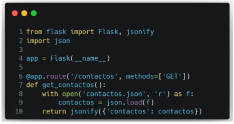
@ Revisa la explicación del tema 7

Opciones:

a. Crea una nueva base de datos.
b. Define una ruta en la aplicación web.
c. Elimina una base de datos.
d. Escribe en una base de datos.

Pregunta 34

Supón que tienes el siguiente código en Python:



```python
from flask import Flask, jsonify
import json

app = Flask(__name__)

@app.route('/contactos', methods=['GET'])
def get_contactos():
    with open('contactos.json', 'r') as f:
        contactos = json.load(f)
    return jsonify({'contactos': contactos})
```

¿Qué hace este código?

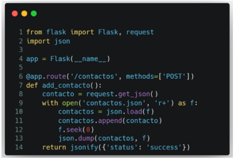
@ Revisa la explicación del tema 7

Opciones:

a. Lee datos de un archivo JSON y los devuelve en una ruta de Flask.
b. Escribe datos en un archivo JSON.
c. Crea un nuevo archivo JSON.
d. Elimina un archivo JSON.

Pregunta 35

Supón que tienes el siguiente código en Python:



```python
from flask import Flask, request
import json

app = Flask(__name__)

@app.route('/contactos', methods=['POST'])
def add_contacto():
    contacto = request.get_json()
    with open('contactos.json', 'r+') as f:
        contactos = json.load(f)
        contactos.append(contacto)
        f.seek(0)
        json.dump(contactos, f)
    return jsonify({'status': 'success'})
```

¿Qué hace este código?
@ Revisa la explicación del tema 7

Opciones:

a. Elimina un contacto de un archivo JSON.
b. Lee datos de un archivo JSON.
c. Escribe datos en un archivo JSON.
d. Añade un nuevo contacto a un archivo JSON.

Pregunta 36

¿Qué es Pytest?
@ Revisa la explicación del tema 8

Opciones:

a. Un tipo de error en Python.
b. Un tipo de estructura de datos en Python.
c. Una biblioteca para realizar pruebas unitarias.
d. Un bloque de código que no puede ser reutilizado.

Pregunta 37

En Python, ¿cómo se ejecutan las pruebas unitarias con Pytest?
@ Revisa la explicación del tema 8

Opciones:

a. Usando el comando python test.
b. Usando el comando pytest.
c. Usando el comando python pytest.
d. Usando el comando run pytest.

Pregunta 38

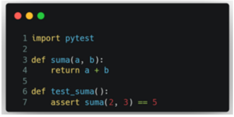
En Python, ¿cómo se agrupan las pruebas en clases con Pytest?
@ Revisa la explicación del tema 8

Opciones:

a. El nombre de la clase debe iniciar con la palabra *Test*
b. El nombre de la clase debe iniciar con la palabra *Class*.
c. El nombre de la clase debe iniciar con la palabra *Pytest*.
d. El nombre de la clase debe iniciar con la palabra *Unit*.

Pregunta 39

Supón que tienes el siguiente código en Python:



```python
import pytest

def suma(a, b):
    return a + b

def test_suma():
    assert suma(2, 3) == 5
```

¿Qué hace este código?
@ Revisa la explicación del tema 8

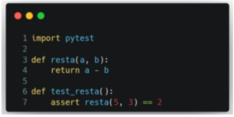
Opciones:

a. Define una función de resta y una prueba de rendimiento para la función.
b. Define una función de suma y una prueba de rendimiento para la función.
c. Define una función de resta y una prueba unitaria para la función.

d. Define una función de suma y una prueba unitaria para la función.

Pregunta 40

Supón que tienes el siguiente código en Python:



```python
import pytest

def resta(a, b):
    return a - b

def test_resta():
    assert resta(5, 3) == 2
```

Si ejecutas este código con Pytest, ¿cuál sería el resultado?

@ Revisa la explicación del tema 8

Opciones:

a. Una falla porque la prueba unitaria test_resta está mal definida.
b. Una falla porque la función resta está mal definida.
c. Éxito, porque la prueba unitaria test_resta pasa.
d. Éxito, porque la prueba unitaria test_resta falla.