

M3C6 Python Assignment 4

Preguntas teóricas.

1.- ¿Para qué usamos Clases en Python?

Las clases son mapeadores de objetos. Dan la habilidad de crear un plano para los objetos. Las clases pueden tener datos dentro de ellas, también funciones y, también comportamiento. El comportamiento se cuida con funciones.

2.- ¿Qué método se ejecuta automáticamente cuando se crea una instancia de una clase?

La función constructora “__init__”. Es una función que se llamará automáticamente cada vez que creamos una instanciación, que es el proceso de creación de un objeto.

Es una función que está disponible para todas las clases.

3.- ¿Cuáles son los tres verbos de API?

Hasta donde yo he visto y aprendido, los verbos como tal se refieren al protocolo o método HTTP en un sistema REST:

GET (listado de recursos),
POST (publicación de recursos),
PUT (modificación total de un recurso),
PATCH (modificación parcial de un recurso) y
DELETE (eliminación de un recurso).

Los hemos visto en la formación sobre Postman.

No se si entiendo bien esta pregunta. Lo de “tres verbos de API” no entiendo muy bien a qué se refiere.

4.- ¿Es MongoDB una base de datos SQL o NoSQL?

MongoDB es una base de datos NoSQL, No solo SQL, orientada a documentos y de código abierto. Soporta datos SQL y, además, guarda estructuras de datos BSON (una especificación similar a JSON) con un esquema dinámico, haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.

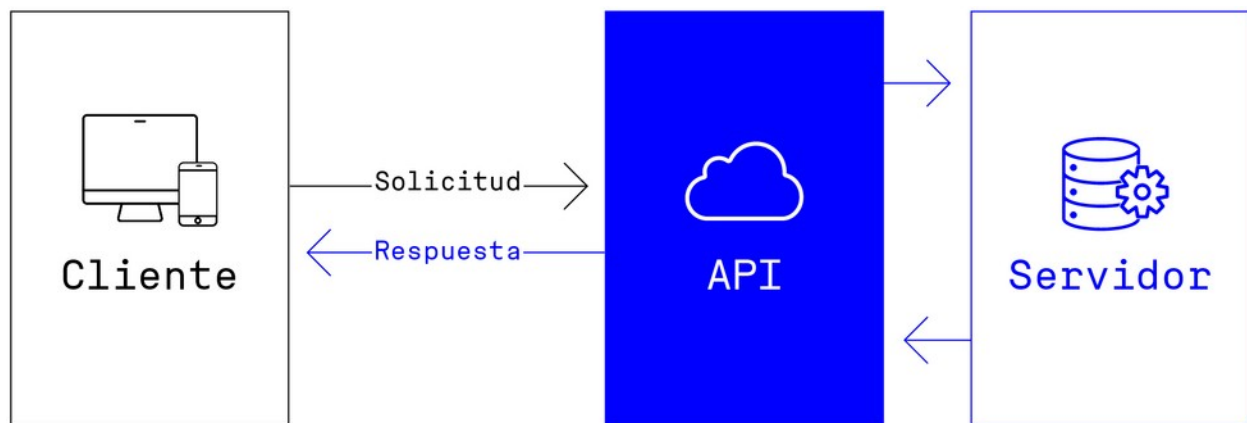
Los desarrolladores utilizan MongoDB por su flexibilidad, escalabilidad, rendimiento y ecosistema: capacidades fundamentales y necesarias para crear y potenciar aplicaciones modernas.

5.- ¿Qué es una API?

API significa interfaz de programación de aplicaciones. Es una forma en que podemos comunicarnos con una aplicación y sin necesidad de implementar un raspador, “scraper”.

Nos da un conjunto de comandos que podemos usar y podemos hacer que nuestra aplicación se comunique con otro servidor en otra aplicación.

Lo que conseguimos con una API es un conjunto de endpoints (puntos finales). Un endpoint es sólo una URL que nos devuelve datos JSON que una aplicación externa pueda usar.



Fuente: gluo.mx

Por ejemplo, si estás creando un React (librería o biblioteca de Javascript que sirve para construir interfaces de usuario), puedes comunicarte con esa URL y recuperar los datos de una forma que puedas anilizarlos y después mostrarlos en pantalla.

En el curso se nos ha enseñado como crear nuestra primera API en la parte del servidor, en Flask, aprovechando el lenguaje Python.

Para ello hemos tenido que haber instalado previamente el entorno “pipenv”, darnos de alta en Postman y utilizar un editor de texto, en mi caso, Visual Studio Code.

6.- ¿Qué es Postman?

Es una herramienta que nos permite establecer comunicación con APIs externas y probarlas.

Este software fue creado en 2012 por Abhinav Asthana, Ankit Sobti y Abhijit Kane en Bangalore, India, para resolver el problema de compartir pruebas API.

Postman hace más eficiente nuestro proceso de codificar. Es sencilla y fácil de usar a la hora de documentar, diseñar y probar las APIs.

7.- ¿Qué es el polimorfismo?

Polimorfismo es un concepto que quiere expresar que un elemento puede tener muchos cambios o muchas formas.

El polimorfismo es una técnica de la programación orientada a objetos que permite a distintos objetos responder de manera diferente a un mismo llamado de método. En Python esto se logra gracias al uso de clases y funciones.

Los objetos de diferentes clases pueden ser accedidos utilizando el mismo interfaz, mostrando un comportamiento distinto según cómo sean accedidos.

En Python no es necesario que los objetos compartan un interfaz, simplemente basta con que tengan los métodos que se quieren llamar.

He descubierto que, para entender el concepto de poliformismo, es importante comprender lo que se denomina “tipado de pato” (duck typing): “si habla como un pato, si camina como un pato,... entonces pato será”.

Aplicado a la programación podemos decir: “si un objeto creado en Python se comporta de una determinada manera, no importa tanto el tipo de objeto como el método que utilizamos para llegar hasta él”.

En lenguaje “patuno”, llegamos al “pato” (objeto) por su “parpeo” y sus “andares” (comportamiento, método).

En Python nos importan más los métodos que los tipos de objetos.

8.- ¿Qué es un método dunder?

Dunder son los métodos que comienzan con guiones bajos dobles, que tienen algunos nombres de métodos como “init” y luego terminan con dos guiones bajos más:

`__init__`

Este método nos permite asignar atributos y realizar operaciones con el objeto en el momento de su creación. También es ampliamente conocido como el constructor.

También son llamados “métodos especiales” o “métodos mágicos” y, los más comunes, pueden ser clasificados según su utilidad:

- Para operaciones aritméticas: `__add__` # `__sub__` # `__mul__` # `__div__`
- Para operaciones de comparación: `__lt__` # `__gt__` # `__eq__`
- Para operaciones del “ciclo de vida”: `__init__` # `__del__`
- Para operaciones de representación: `__str__` # `__repr__`

9.- ¿Qué es un decorador de python?

Es una función que recibe otra función como parámetro, le añade cosas y retorna otra función diferente.

Son muy útiles. Nos permiten envolver una función dentro de otra y modificar el comportamiento de esta última sin modificarla permanentemente.

Los decoradores se aplican utilizando el símbolo @ seguido del nombre de la función decoradora justo encima de la función que deseas decorar:

```
@decorador
```

```
def mi_funcion():
```

```
    # Tu función específica
```

```
    return # El resultado que quieres devolver con tu función
```

Ejercicio práctico.

Cree una clase de Python llamada Usuario que use el método init y cree un nombre de usuario y una contraseña. Crea un objeto usando la clase.