01 - ¿Para qué usamos Classes en Python?

Las clases se usan para agrupar objetos y que estos trabajen en conjunto cuando se hace una llamada desde el exterior de la clase. Al hacer una llamada desde el exterior, se le proporciona unos valores y son con estos valores son con los que van a trabajar los objetos que hay dentro de la clase.

Las clases se pueden usar tantas veces como se necesite desde cualquier parte del código, por lo que son reutilizables y se ahorra mucho código y tiempo, tanto para crearlo como para hacer mantenimiento o cambios a lo largo del tiempo de uso.

La estructura básica de una clase es la palabra "class" seguido del nombre de la clase con la primera letra en mayúscula y finalmente dos puntos ":". Usando sangría (Tal como se hace en las condicionales, bucles y funciones) se añaden los objetos que pertenecerán a la clase.

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

El método que de forma automática se pone en funcionamiento es __init__

Este método sirve para crear los atributos que van a poder tener el resto de objetos que hay dentro de la clase y se le proporciona al momento de hacer una llamada desde el exterior de la clase.

La estructura del método __init__ es la siguiente:

- def __init__(self, dato1, dato2):
  - self.dato1 = dato1
  - self.dato2 = dato2

03 - ¿Cuáles son los tres verbos de API?

Los tres verbos principales son GET, POST y DELETE.

- GET: Sirve para pedir y traer información o datos. Por ejemplo, si se desea ver la información de una BBDD o los datos de un usuario se usaría GET.
- POST: Es el contrario de GET, se usa para enviar o modificar datos o información. Por ejemplo, queremos publicar una noticia en nuestro blog o deseamos hacer un cambio en la descripción de un artículo publicado previamente usaríamos POST.
- DELETE: Tal como se intuye por su nombre, este verbo sirve para eliminar datos o información, se puede usar tanto para eliminar un dato en concreto o para eliminar un grupo grande de datos con una sola orden.

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

MongoDB es una BBDD de tipo NoSQL.

Una BBDD del tipo NoSQL quiere decir que no trabaja de forma obligatoria en lenguaje SQL, sino que tienen un uso más flexible y dinámico, permitiendo usar diferentes modelos de datos, como documentos, clave-valor, grafos, entre otros.

Gracias a ello tiene una mejor adaptación a todo tipo de aplicaciones.

05 - ¿Qué es una API?

Una API es un grupo de datos estructurados que logra comunicarse entre diferentes sistemas o aplicaciones, de esta forma se podrá crear un vinculo entre servicios para transportar información o dar diferentes tipos de órdenes.

Por lo general las APIs están estructuradas en JSON.

Por ejemplo, tenemos una web con tienda online y queremos tener por un lado una pasarela de pago por tarjeta bancaria y también avisos al proveedor de material cuando algún artículo esta cerca de terminar, todo esto se enlazaría a través de APIs.

Por un lado la pasarela de pago, desde nuestra web tendría que mandar los datos del comercio, datos del cliente, datos de compra y datos de la tarjeta hacia el banco para que autorice la compra, esa información habría que mandarla a través de una API.

Y por otro lado la comunicación con el proveedor, en nuestra web tendremos que programar que cuando un articulo baje de un número concreto de stock en almacén, mande la información del comercio, del articulo y del número de unidades hacia el sistema del proveedor, eso también se mandaría a través de una API.

06 - ¿Qué es Postman?

Postman es una herramienta para crear, modificar y comprobar APIs. Tiene una interfaz grafica con diferentes opciones para poder crear APIs y probar su funcionamiento.

Tiene capacidad para trabajar con HTTP y HTTPS con lo que ayuda a poder depurar las APIs, también tiene varios métodos de solicitud disponibles, GET, POST o DELETE entre otras.

07 - ¿Qué es el polimorfismo?

El polimorfismo hace que una llamada se pueda comunicar con diferentes clases (Y de ahí a sus objetos) y de esta forma poder recibir una respuesta diferente según se necesite.

Es decir, según el contexto se pueden usar objetos de diferentes clases de manera intercambiable y tendrán comportamientos distintos según la información que se les ha proporcionado.

08 - ¿Qué es un método dunder?

Los métodos Dunder sirven para hacer algún comportamiento especial y concreto en los objetos que hay dentro de una clase.

La estructura es encerrar el nombre entre dos guiones bajos, tanto al principio como al final del nombre, con lo que quedaría de la siguiente forma __nombre__

Ejemplos de métodos Dunder son los siguientes:

- __init__: Este es uno de los más importantes, es el que inicia la clase y trae los datos de la llamada del exterior.
- __repr__: Devuelve una cadena del objeto y se usa mayormente para depurar y desarrollar las clases.
- __str__: Devuelve una cadena de texto del objeto.

09 - ¿Qué es un decorador de python?

Es una función que puede cambiar otra función sin necesidad de modificar el código de la función original. Ayuda a extender y reutilizar funciones de forma sencilla sin tener que añadir nuevas funciones.

Un decorador debe empezar siempre con una arroba @ seguido del nombre de la función a la que se le va a asociar, sin necesidad de poner paréntesis o dos puntos finales.

Debajo del comienzo del decorador se escribe la función en concreto que va a ejecutarse.