

CHECKPOINT6

1.¿Para qué usamos Clases en Python?

- Una clase en Python sirve para definir un conjunto de métodos y atributos que describen un objeto o entidad.

Para definirlos se utiliza la palabra `class` delante del nombre que le ponemos a la clase y posterior al nombre ponemos el símbolo `:` y luego el contenido que se define dentro de la clase.

Por ejemplo:

```
class Person:

def __init__(self,name,age):
    self.name=name
    self.age=age
```

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

- Se inicia el método `__init__` para iniciar una instancia.

Como en el propio ejemplo anterior :

```
class Person:

def __init__(self,name,age):
    self.name=name
    self.age=age
```

3.¿Cuáles son los tres verbos de API?

- Application Programming Interfaces es el significado de las 3 siglas de API.

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

- MongoDB es una base de datos NoSQL de código abierto, almacena objetos de datos en colecciones y documentos. NoSQL es un sistema de base de datos más nuevo que no utiliza un lenguaje de consulta estándar pero emplea documentos JSON para el almacenamiento de datos.

Por ejemplo, en este link se puede apreciar los diferentes métodos para insertar documentos en mongo :

https://www.adrformacion.com/knowledge/programacion/como_insertar_documentos_en_mongodb.html

5.¿Qué es una API?

- Las API son mecanismos que permiten a dos componentes de software comunicarse entre sí mediante un conjunto de definiciones y protocolos.
- Por ejemplo, el sistema de software del instituto de meteorología contiene datos meteorológicos diarios. La aplicación meteorológica de su teléfono “habla” con este sistema a través de las API y le muestra las actualizaciones meteorológicas diarias en su teléfono.

Por ejemplo, con la API Web de Twitter se puede escribir un programa en un lenguaje como Python o Javascript que recoja metadatos sobre twits.

6.¿Qué es Postman?

- Postman es una plataforma de API para crear y utilizar API. Simplifica cada paso del ciclo de vida de la API y agiliza la colaboración para poder crear mejores API, más rápido.

Postman surgió originariamente como una extensión para el navegador Google Chrome. A día de hoy dispone de aplicaciones nativas para MAC y Windows y están trabajando en una aplicación nativa para Linux.

Está compuesto por diferentes herramientas y utilidades gratuitas:

-Creación de peticiones a APIs internas o de terceros.

-Elaboración de tests para validar el comportamiento de APIs.

-Posibilidad de crear entornos de trabajo diferentes (con variables globales y locales).

Todo ello con la posibilidad de ser compartido con otros compañeros del equipo de manera gratuita (exportación de toda esta información mediante URL en formato JSON).

Aquí en este enlace podemos encontrar más información:

<https://openwebinars.net/blog/que-es-postman/>

7. ¿Qué es el polimorfismo?

- El polimorfismo es la capacidad que tienen ciertos lenguajes para hacer que, al enviar el mismo mensaje desde objetos de diferentes clases, cada uno de esos objetos pueda responder a ese mensaje diferente forma.

Es decir, dos objetos de diferentes clases pueden tener métodos con el mismo nombre, y ambos métodos pueden ser llamados con el mismo código, dando respuestas diferentes.

Aquí podemos ampliar la información y apreciarlo mejor mediante los ejemplos.

<https://ellibrodepython.com/polimorfismo-en-programacion>

8. ¿Qué es un método dunder?

- Métodos Dunder o los denominados también métodos Mágicos o especiales. Se utilizan para emular el comportamiento de las funciones integradas.

Sus nombres empiezan y terminan en __ doble barra baja. Por ejemplo init.

```
class user:
```

```
    def __init__(self,age):
```

```
        self.age=age
```

Aquí podemos encontrar más información:

https://barcelonageeks.com/dunder-o-metodos-magicos-en-python/#google_vignette

9. ¿Qué es un decorador de python?

- Los decoradores son una característica de Python que permite modificar, reducir las líneas de código duplicadas, hacer nuestro código más legible y fácil de testear.

Aquí podemos ver más ejemplos:

<https://ellibrodepython.com/decoradores-python>

EJERCICIO

-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.

class Usuario:

```
def __init__(self, username, password):
```

```
    self.username = username
```

```
    self.password = password
```

```
user = Usuario("ainara", "asdfg")
```

```
print (user.username, user.password)
```