1. ***Clase Persona:***

El código define una clase llamada Persona que tiene tres atributos: nombre, edad y pais. Además, la clase incluye varios métodos para imprimir estos atributos.

class Persona:

def \_\_init\_\_(self, nombre, edad, pais):

self.nombre = nombre

self.edad = edad

self.pais = país

* **\_\_init\_\_**: Este es el método constructor que se llama cuando se crea una nueva instancia de la clase. Inicializa los atributos nombre, edad y pais con los valores proporcionados.

**Métodos para Imprimir Atributos**

def imprimir\_nombre(self):

print(self.nombre)

def imprimir\_edad(self):

print(self.edad)

def imprimir\_pais(self):

print(self.pais)

* **imprimir\_nombre**: Imprime el nombre de la persona.
* **imprimir\_edad**: Imprime la edad de la persona.
* **imprimir\_pais**: Imprime el país de la persona.

**Método para Imprimir Todos los Atributos**

def imprimir\_atributos(self):

print(f'Nombre: {self.nombre}')

print(f'Edad: {self.edad}')

print(f'Pais: {self.pais}')

* **imprimir\_atributos**: Imprime todos los atributos de la persona en un formato más legible.

1. ***Clase estudiante***

El código define una clase llamada Estudiante que hereda de la clase Persona. La herencia permite que Estudiante utilice los métodos y atributos de Persona.

class Estudiante(Persona):

def \_\_init\_\_(self, nombre, edad, pais):

super().\_\_init\_\_(nombre, edad, pais)

def imprimir\_atributos(self):

super().imprimir\_atributos()

* **class Estudiante(Persona)**: Define la clase Estudiante que hereda de Persona.
* **def \_\_init\_\_(self, nombre, edad, pais)**: Este es el constructor de la clase [Estudiante](vscode-file://vscode-app/c:/Users/DESIR%C3%88ECAMACHO/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html). Utiliza [super()](vscode-file://vscode-app/c:/Users/DESIR%C3%88ECAMACHO/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) para llamar al constructor de la clase [Persona](vscode-file://vscode-app/c:/Users/DESIR%C3%88ECAMACHO/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) y así inicializar los atributos [nombre](vscode-file://vscode-app/c:/Users/DESIR%C3%88ECAMACHO/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), [edad](vscode-file://vscode-app/c:/Users/DESIR%C3%88ECAMACHO/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) y [pais](vscode-file://vscode-app/c:/Users/DESIR%C3%88ECAMACHO/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ").
* **def imprimir\_atributos(self)**: Este método llama al método [imprimir\_atributos](vscode-file://vscode-app/c:/Users/DESIR%C3%88ECAMACHO/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") de la clase [Persona](vscode-file://vscode-app/c:/Users/DESIR%C3%88ECAMACHO/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) usando [super()](vscode-file://vscode-app/c:/Users/DESIR%C3%88ECAMACHO/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), lo que permite reutilizar la funcionalidad de la clase base.

1. ***Clase Profesor***

La clase Profesor hereda de Persona, lo que significa que Profesor tiene acceso a todos los atributos y métodos de Persona. En el constructor \_\_init\_\_ de Profesor, se llama al constructor de la clase base Persona utilizando super(), pasando los parámetros nombre, edad y pais para inicializar los atributos heredados. Esto asegura que una instancia de Profesor se inicialice correctamente con los mismos atributos que una instancia de Persona.

El método imprimir\_atributos en la clase Profesor también utiliza super() para llamar al método imprimir\_atributos de la clase base Persona. Esto permite que una instancia de Profesor imprima sus atributos de la misma manera que una instancia de Persona, reutilizando el código de la clase base. En resumen, la clase Profesor extiende la funcionalidad de Persona sin necesidad de duplicar código, demostrando una aplicación efectiva de la herencia en programación orientada a objetos.

1. ***Clase Curso***

El código proporcionado define una clase llamada Curso en Python, que modela un curso con dos atributos principales: nombre y duracion. Esta clase encapsula las propiedades y comportamientos asociados con un curso.

El método \_\_init\_\_ es un método especial en Python conocido como el constructor. Se llama automáticamente cuando se crea una nueva instancia de la clase. En este caso, el constructor toma dos parámetros, nombre y duracion, que se utilizan para inicializar los atributos de la instancia self.nombre y self.duracion. La palabra clave self es una referencia a la instancia actual de la clase, lo que permite que el método establezca los atributos específicos de esa instancia.

La clase también incluye dos métodos para imprimir los atributos del curso. El método imprimir\_nombre imprime el atributo nombre, mientras que el método imprimir\_duracion imprime el atributo duracion. Estos métodos proporcionan una forma sencilla de acceder y mostrar los valores de los atributos del curso.

En resumen, la clase Curso es un ejemplo sencillo de una clase en Python que demuestra el uso de un constructor para inicializar atributos de instancia y métodos para interactuar con esos atributos. Esta clase se puede utilizar para crear y gestionar objetos de curso, cada uno con su propio nombre y duración.

1. ***Clase Materia***

El código proporcionado define una clase en Python llamada Materia, que modela una materia o asignatura en un contexto educativo. La clase tiene cuatro atributos principales: nombre, curso, profesor y estudiantes. Estos atributos se inicializan en el método constructor \_\_init\_\_, que toma cuatro parámetros correspondientes a estos atributos. La palabra clave self se utiliza para asignar los valores de estos parámetros a los atributos de la instancia, permitiendo que cada instancia de la clase Materia tenga su propio conjunto único de valores para estos atributos.

La clase también incluye cuatro métodos para imprimir los valores de sus atributos. El método imprimir\_nombre imprime el valor del atributo nombre, el método imprimir\_curso imprime el valor del atributo curso, el método imprimir\_profesor imprime el valor del atributo profesor y el método imprimir\_estudiantes imprime el valor del atributo estudiantes. Estos métodos proporcionan una forma sencilla de acceder y mostrar los valores de los atributos, facilitando la recuperación e impresión de información sobre una materia o asignatura en particular.

En resumen, la clase Materia es un ejemplo sencillo de una clase en Python que demuestra el uso de un constructor para inicializar atributos de instancia y métodos para interactuar con esos atributos. Esta clase se puede utilizar para crear y gestionar objetos de materia, cada uno con su propio nombre, curso, profesor y lista de estudiantes. Los métodos para imprimir los atributos facilitan la recuperación y visualización de información sobre cada materia, lo cual puede ser útil en diversas aplicaciones educativas.

1. ***Clase Grupo***

El código proporcionado define una clase en Python llamada Grupo, que modela un grupo en un contexto educativo. La clase tiene cuatro atributos principales: nombre, curso, profesor y estudiantes. Estos atributos se inicializan en el método constructor \_\_init\_\_, que toma cuatro parámetros correspondientes a estos atributos y los asigna a la instancia de la clase.

class Grupo:

def \_\_init\_\_(self, nombre, curso, profesor, estudiantes):

# Inicializa los atributos del grupo

self.nombre = nombre

self.curso = curso

self.profesor = profesor

self.estudiantes = estudiantes

def imprimir\_nombre(self):

# Imprime el nombre del grupo

print(self.nombre)

def imprimir\_curso(self):

# Imprime el curso del grupo

print(self.curso)

def imprimir\_profesor(self):

# Imprime el profesor del grupo

print(self.profesor)

def imprimir\_estudiantes(self):

# Imprime la lista de estudiantes del grupo

print(self.estudiantes)

* **\_\_init\_\_**: Este es el constructor de la clase Grupo. Se llama automáticamente cuando se crea una nueva instancia de la clase. Inicializa los atributos nombre, curso, profesor y estudiantes con los valores proporcionados como argumentos.
* **imprimir\_nombre**: Este método imprime el valor del atributo nombre de la instancia del grupo.
* **imprimir\_curso**: Este método imprime el valor del atributo curso de la instancia del grupo.
* **imprimir\_profesor**: Este método imprime el valor del atributo profesor de la instancia del grupo.
* **imprimir\_estudiantes**: Este método imprime el valor del atributo estudiantes de la instancia del grupo.

1. ***Clase Salón***

El código proporcionado define una clase en Python llamada Salon, que modela un salón de clases en un contexto educativo. La clase tiene cinco atributos principales: nombre, grupo, curso, profesor y estudiantes. Estos atributos se inicializan en el método constructor \_\_init\_\_, que toma cinco parámetros correspondientes a estos atributos y los asigna a la instancia de la clase.

class Salon:

def \_\_init\_\_(self, nombre, grupo, curso, profesor, estudiantes):

# Inicializa los atributos del salón

self.nombre = nombre

self.grupo = grupo

self.curso = curso

self.profesor = profesor

self.estudiantes = estudiantes

def imprimir\_nombre(self):

# Imprime el nombre del salón

print(self.nombre)

def imprimir\_grupo(self):

# Imprime el grupo del salón

print(self.grupo)

def imprimir\_curso(self):

# Imprime el curso del salón

print(self.curso)

def imprimir\_profesor(self):

# Imprime el profesor del salón

print(self.profesor)

def imprimir\_estudiantes(self):

# Imprime la lista de estudiantes del salón

print(self.estudiantes)

* **\_\_init\_\_**: Este es el constructor de la clase Salon. Se llama automáticamente cuando se crea una nueva instancia de la clase. Inicializa los atributos nombre, grupo, curso, profesor y estudiantes con los valores proporcionados como argumentos.
* **imprimir\_nombre**: Este método imprime el valor del atributo nombre de la instancia del salón.
* **imprimir\_grupo**: Este método imprime el valor del atributo grupo de la instancia del salón.
* **imprimir\_curso**: Este método imprime el valor del atributo curso de la instancia del salón.
* **imprimir\_profesor**: Este método imprime el valor del atributo profesor de la instancia del salón.
* **imprimir\_estudiantes**: Este método imprime el valor del atributo estudiantes de la instancia del salón.

1. ***Clase Escuela***

El código proporcionado define una clase en Python llamada Escuela, que modela una escuela en un contexto educativo. La clase tiene seis atributos principales: nombre, salon, grupo, curso, profesor y estudiantes. Estos atributos se inicializan en el método constructor \_\_init\_\_, que toma seis parámetros correspondientes a estos atributos y los asigna a la instancia de la clase.

class Escuela:

def \_\_init\_\_(self, nombre, salon, grupo, curso, profesor, estudiantes):

# Inicializa los atributos de la escuela

self.nombre = nombre

self.salon = salon

self.grupo = grupo

self.curso = curso

self.profesor = profesor

self.estudiantes = estudiantes

def imprimir\_nombre(self):

# Imprime el nombre de la escuela

print(self.nombre)

def imprimir\_salon(self):

# Imprime el salón de la escuela

print(self.salon)

def imprimir\_grupo(self):

# Imprime el grupo de la escuela

print(self.grupo)

def imprimir\_curso(self):

# Imprime el curso de la escuela

print(self.curso)

def imprimir\_profesor(self):

# Imprime el profesor de la escuela

print(self.profesor)

def imprimir\_estudiantes(self):

# Imprime la lista de estudiantes de la escuela

print(self.estudiantes)

* [**\_\_init\_\_**](vscode-file://vscode-app/c:/Users/DESIR%C3%88ECAMACHO/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Este es el constructor de la clase Escuela. Se llama automáticamente cuando se crea una nueva instancia de la clase. Inicializa los atributos nombre, salon, grupo, curso, profesor y estudiantes con los valores proporcionados como argumentos.
* **imprimir\_nombre**: Este método imprime el valor del atributo nombre de la instancia de la escuela.
* **imprimir\_salon**: Este método imprime el valor del atributo salon de la instancia de la escuela.
* **imprimir\_grupo**: Este método imprime el valor del atributo grupo de la instancia de la escuela.
* **imprimir\_curso**: Este método imprime el valor del atributo curso de la instancia de la escuela.
* **imprimir\_profesor**: Este método imprime el valor del atributo profesor de la instancia de la escuela.
* **imprimir\_estudiantes**: Este método imprime el valor del atributo estudiantes de la instancia de la escuela.

1. ***Clase Universidad***

El código proporcionado define una clase en Python llamada Universidad, que modela una universidad con varios atributos. La clase tiene seis atributos principales: nombre, salon, grupo, curso, profesor y estudiantes. Estos atributos se inicializan en el método constructor \_\_init\_\_, que toma seis parámetros correspondientes a estos atributos y los asigna a la instancia de la clase.

class Universidad:

def \_\_init\_\_(self, nombre, salon, grupo, curso, profesor, estudiantes):

# Inicializa los atributos de la universidad

self.nombre = nombre

self.salon = salon

self.grupo = grupo

self.curso = curso

self.profesor = profesor

self.estudiantes = estudiantes

def imprimir\_nombre(self):

# Imprime el nombre de la universidad

print(self.nombre)

def imprimir\_salon(self):

# Imprime el salón de la universidad

print(self.salon)

def imprimir\_grupo(self):

# Imprime el grupo de la universidad

print(self.grupo)

def imprimir\_curso(self):

# Imprime el curso de la universidad

print(self.curso)

def imprimir\_profesor(self):

# Imprime el profesor de la universidad

print(self.profesor)

def imprimir\_estudiantes(self):

# Imprime la lista de estudiantes de la universidad

print(self.estudiantes)

* **\_\_init\_\_**: Este es el constructor de la clase Universidad. Se llama automáticamente cuando se crea una nueva instancia de la clase. Inicializa los atributos nombre, salon, grupo, curso, profesor y estudiantes con los valores proporcionados como argumentos.
* **imprimir\_nombre**: Este método imprime el valor del atributo nombre de la instancia de la universidad.
* **imprimir\_salon**: Este método imprime el valor del atributo salon de la instancia de la universidad.
* **imprimir\_grupo**: Este método imprime el valor del atributo grupo de la instancia de la universidad.
* **imprimir\_curso**: Este método imprime el valor del atributo curso de la instancia de la universidad.
* **imprimir\_profesor**: Este método imprime el valor del atributo profesor de la instancia de la universidad.
* **imprimir\_estudiantes**: Este método imprime el valor del atributo estudiadontes de la instancia de la universidad.