Programación orientada a objetos.

Ahora sabiendo los conceptos fundamentales de la POO trasladaremos código nuestra primera ***Clase.***

La sintaxis que usaremos para definir una clase será:

*#Sintaxis para definir una clase*

class NombreClase():

Ahora una vez tengamos definida nuestra clase vamos a comenzar incluyendo las propiedades de nuestra clase.

*#Sintaxis para definir una clase*

class PC():

*#Propiedades de la pc*

**ancho\_gabinete** = 280

**alto\_gabinete** = 480

**largo\_gabinete** = 430

**ventiladores\_gabinete** = 7

Estas son las propiedades que puede llegar a tener una PC, ahora definiremos los comportamientos de esta clase.

Para definir los comportamientos de nuestro objeto haremos uso de ***métodos/funciones.***

Para ello usaremos la siguiente sintaxis:

*#Comportamientos de la pc*

    def encender(self):

Vemos que tiene la misma estructura que una función, la diferencia es que como parámetro le pasamos self, hace referencia al propio objeto perteneciente a la clase, despues lo veremos más a detalle.

*#Sintaxis para definir una clase*

class PC():

*#Propiedades de la pc*

**ancho\_gabinete** = 280

**alto\_gabinete** = 480

**largo\_gabinete** = 430

**ventiladores\_gabinete** = 7

*#Comportamientos de la pc*

    def encender(self):

        pass

    def apagar(self):

        pass

Por el momento no agregaremos funcionamiento a los métodos.

Ahora crearemos el primer objeto perteneciente a la clase previamente creada.

mi\_pc = PC()

De esta manera hemos creado el primer objeto de nuestra clase PC llamado mi\_pc.

Estamos haciendo aquí una ***instancia de clase.***

Para acceder a propiedades y comportamientos de nuestro objeto usaremos la nomenclatura del punto.

print(f"El largo del gabinete es de {mi\_pc.**largo\_gabinete**}mm")

Aquí accedemos a la propiedad del largo del gabinete.



*#Impresión de propiedades de la consola}*

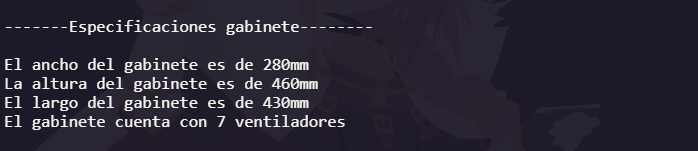
print("-------Especificaciones gabinete--------\n")

print(f"El ancho del gabinete es de {mi\_pc.**ancho\_gabinete**}mm")

print(f"La altura del gabinete es de {mi\_pc.**alto\_gabinete**}mm")

print(f"El largo del gabinete es de {mi\_pc.**largo\_gabinete**}mm")

print(f"El gabinete cuenta con {mi\_pc.**ventiladores\_gabinete**} ventiladores")



Y para acceder a los comportamientos de usaremos la misma sintaxis

Para ello primero creare una nueva propiedad que es:

**encendida** = False

Ahora esto indica para el ejemplo que la pc esta apagada.

Ahora digamos que quiero encenderla para ello usaremos en método encender(), pero como código le pasaremos lo siguiente.

        self.**encendida** = True

Aquí lo que estamos diciendo que es accede a la propiedad de nuestra clase y la reedefina para que sea true.

Ahora si llamamos a este método.

*#Impresion comportamientos de la pc*

mi\_pc.encender()

Entonces aquí llamamos al método ***encender()*** este método al momento de crearlo le especificamos el parámetro self, que es lo que va a recibir, entonces lo que recibirá es el objeto que creamos ***mi\_pc*** por lo tanto lo que en realidad le estamos diciendo aquí es:

    def encender(self):

        self.**encendida** = True

*#mi\_pc.encender = True*

Ahora para comprobar el estado de nuestra PC crearemos un nuevo comportamiento

    def estado(self):

        if (self.**encendida**):

            return "Encendida"

        else:

            return "Apagada"

Por lo tanto al momento de llamar al método encendido le estamos diciendo que encienda el objeto mi\_pc.

*#Impresion comportamientos de la pc*

mi\_pc.encender()

print("La pc esta {}".format(mi\_pc.estado()))



Aquí primero llamamos al método encender que prende la PC, para despues comprobar el estado de la PC.

*#Impresion comportamientos de la pc*

*# mi\_pc.encender()*

print("La pc esta {}".format(mi\_pc.estado()))



En base a este código:

*#Sintaxis para definir una clase*

class PC():

*#Propiedades de la consola*

**ancho\_gabinete** = 280

**alto\_gabinete** = 460

**largo\_gabinete** = 430

**ventiladores\_gabinete** = 7

**encendida** = False

*#Comportamientos de la consola*

    def encender(self):

        self.**encendida** = True

*#mi\_pc.encender = True*

    def apagar(self):

        pass

    def estado(self):

        if (self.**encendida**):

            return "Encendida"

        else:

            return "Apagada"

*#Instancia de la clase PC*

mi\_pc = PC()

*#Impresión de propiedades de la pc*

print("-------Especificaciones gabinete--------\n")

print(f"El ancho del gabinete es de {mi\_pc.**ancho\_gabinete**}mm")

print(f"La altura del gabinete es de {mi\_pc.**alto\_gabinete**}mm")

print(f"El largo del gabinete es de {mi\_pc.**largo\_gabinete**}mm")

print(f"El gabinete cuenta con {mi\_pc.**ventiladores\_gabinete**} ventiladores")

*#Impresion comportamientos de la pc*

print("--------Estado de la PC---------------\n")

*# mi\_pc.encender()*

print("La pc esta {}".format(mi\_pc.estado()))

Podemos decir que tiene:

* 5 propiedades (atributos)
* 3 comportamientos (métodos)
* 1 instancia de la clase (objeto perteneciente a la clase)