

Universidad Nacional del Altiplano
Facultad de Ingeniería Estadística e Informática

Docente:

Ing. Coyla Idme Leonel

Alumno:

Ticona Miramira Roberto Angel

Método Operador

En Python se tiene los siguientes métodos operadores:

- `__str__` → Permite mostrar (imprimir)
- `__eq__` → Permite comparar.
- `__gt__` → Permite verificar.
- `__add__` → Permite sumar.
- `__sub__` → Permite restar.

» EJEMPLO 1

Crear el objeto persona y usar los métodos operadores.

- Clase: Persona.
- Atributo: Nombre, Edad.
- Acción: `__str__`, `__eq__`, `__gt__`, `__add__`.
- Objetos: `persona1 = Persona("Carlos",30)`, `persona2 = Persona("Ana",25)`, `persona3 = Persona("Lucia",30)`

Código

```
1 class Persona:
2     def __init__(self, nombre, edad):
3         self.nombre = nombre
4         self.edad = edad
5
6     def __str__(self):
7         return f"{self.nombre}, {self.edad} años"
8
9     def __eq__(self, otra):
10        return self.edad == otra.edad
11
12    def __gt__(self, otra):
13        return self.edad > otra.edad
14
15    def __add__(self, otra):
16        return self.edad + otra.edad
17
18 persona1 = Persona("Carlos",30)
19 persona2 = Persona("Ana",25)
20 persona3 = Persona("Lucia",30)
21
22 print(persona1)
23 print(persona2)
```

```
24 print(persona3)
25
26 print(persona1 == persona2)
27 print(persona1 == persona2)
28
29 print(persona1 > persona2)
30
31 print("Suma de edades: ", persona1 + persona2)
32 print("Suma de edades: ", persona1 + persona3)
```

Ejecución

```
1 Carlos, 30 años
2 Ana, 25 años
3 Lucia, 30 años
4 False
5 False
6 True
7 Suma de edades: 55
8 Suma de edades: 60
```

» EJEMPLO 2

Crear el objeto persona y usar los métodos operadores.

- Clase: Producto.
- Atributo: Nombre, Precio, Stock.
- Acción: `__str__`, `__eq__`, `__add__`.
- Objetos: `prod1 = Producto("Arroz", 3.50, 20)`, `prod2 = Producto("Arroz", 3.50, 15)`, `prod3 = Producto("Azúcar", 4.00, 10)`

Código

```
1 class Producto:
2     def __init__(self, nombre, precio, stock):
3         self.nombre = nombre
4         self.precio = precio
5         self.stock = stock
6
7     def __str__(self):
8         return f"{self.nombre} - S/. {self.precio:.2f} stock: {self.stock}"
9
10    def __eq__(self, otra):
11        return self.nombre == otra.nombre
12
13    def __add__(self, otra):
14        return self.precio + otra.precio
15
16 prod1 = Producto("Arroz", 3.50, 20)
17 prod2 = Producto("Arroz", 3.50, 15)
18 prod3 = Producto("Azúcar", 4.00, 10)
19
20 print(prod1)
21 print(prod2)
22 print(prod3)
23
24 print(prod1 == prod2)
25 print(prod1 == prod3)
26
27 print("Suma de precios: ", prod1 + prod2)
```

Ejecución

```
1 Arroz - S/. 3.50 stock: 20
2 Arroz - S/. 3.50 stock: 15
3 Azúcar - S/. 4.00 stock: 10
4 True
5 False
6 Suma de precios: 7.0
```

» EJEMPLO 3

Crear el objeto CuentaBancaria que permite ingresar el nombre y el saldo.

- Clase: CuentaBancaria.
- Atributo: Titular, Saldo.
- Acción: mostrar(), __sub__.
- Objeto: cuenta1 = CuentaBancaria(nombre, dinero)

Código

```
1 class CuentaBancaria:
2     def __init__(self, titular, saldo):
3         self.titular = titular
4         self.saldo = saldo
5
6     def mostrar(self):
7         print(f"Titular: {self.titular} - Saldo: $ {self.saldo:.2f}")
8
9     def __sub__(self, cantidad):
10        if isinstance(cantidad, (int, float)):
11            if cantidad <= self.saldo:
12                return CuentaBancaria(self.titular, self.saldo - cantidad)
13            else:
14                print("Fondos insuficientes")
15                return self
16
17        else:
18            print("Operador no valido")
19            return self
20
21 nombre = input("Ingrese su nombre: ")
22 dinero = float(input("Ingrese su saldo: "))
23 cuenta1 = CuentaBancaria(nombre, dinero)
24
25 cuenta2 = cuenta1 - 250
26 cuenta2.mostrar()
27
28 cuenta3 = cuenta2 - 700
29 cuenta3.mostrar()
```

Ejecución

```
1 Ingrese su nombre: Luis
2 Ingrese su saldo: 1000
3 Titular: Luis - Saldo: $ 750.00
4 Titular: Luis - Saldo: $ 50.00
```