

Métodos de clase en Python

Ejemplo completo de uso de métodos de clase

[crayon-6a9d787c8c4f6116082803/]

Definición de la clase Persona

```
class Persona:  
    def __init__(self, nombre, edad):  
        self.nombre = nombre  
        self.edad = edad
```

1. **Clase Persona:** Se define una clase llamada Persona.
2. **Método `__init__`:** Este es el constructor de la clase, el cual se llama cuando se crea una nueva instancia de Persona. Recibe dos parámetros:
3. **Atributos de instancia:** Los parámetros nombre y edad se asignan a los atributos de la instancia (`self.nombre` y `self.edad`).

Métodos de clase

Los métodos de clase están decorados con `@classmethod`, lo que significa que reciben la clase (`cls`) como primer argumento en lugar de la instancia (`self`).

```
@classmethod  
def desde_nombre(cls, nombre):  
    return cls(nombre, 0)
```

1. **Método `desde_nombre`:** Este método permite crear una instancia de Persona proporcionando solo el nombre.

```
@classmethod
def vacio(cls):
    return cls("Desconocido", 0)
```

2. **Método vacio:** Este método permite crear una instancia de Persona sin proporcionar ningún parámetro.

Representación en String

```
def __str__(self):
    return f"Persona(nombre={self.nombre}, edad={self.edad})"
```

3. **Método __str__:** Este método define cómo se debe representar una instancia de Persona como una cadena de texto.

Ejemplos de uso

```
# Ejemplos de uso
persona1 = Persona("Juan", 30)
persona2 = Persona.desde_nombre("Ana")
persona3 = Persona.vacio()
print(persona1) # Persona(nombre=Juan, edad=30)
print(persona2) # Persona(nombre=Ana, edad=0)
print(persona3) # Persona(nombre=Desconocido, edad=0)
```

1. **Crear una instancia directamente:**
2. **Crear una instancia usando el método de clase desde_nombre:**
3. **Crear una instancia usando el método de clase vacio:**

Salida

Cuando se imprimen estas instancias:

```
print(persona1) # Persona(nombre=Juan, edad=30)
print(persona2) # Persona(nombre=Ana, edad=0)
print(persona3) # Persona(nombre=Desconocido, edad=0)
```

Cada llamada a `print` invoca el método `__str__` de la instancia correspondiente, lo que resulta en la representación de cadena definida en `__str__`, mostrando los valores de nombre y edad de cada instancia.