

## Declarando un método constructor en Python 25/37



Curso de Programación Orientada a Objetos: POO



→|APLICANDO HERE...

En Python encontrarás un concepto denominado **Métodos Mágicos**, estos métodos son llamados automática y estrictamente bajo ciertas reglas. El método constructor en Python forma parte de esta familia de métodos y como aprendimos en la clase anterior lo declaramos usando `__init__`, aunque si nos ponemos estrictos este método no construye el objeto en sí. El encargado de hacer esto es `__new__` y el método `__init__` se encargará de personalizar la instanciación de la clase, esto significa que lo que esté dentro de `__init__` será lo primero que se ejecute cuando se cree un objeto de esta clase.

Para nuestro proyecto tenemos la necesidad de que algunas variables se inicialicen obligatoriamente cuando ocurra la instanciación. Así que declaremos el método `__init__` en las clases de nuestro proyecto con las propiedades obligatorias.

Para la clase `Account` quedaría algo así, notarás que usamos la palabra clave **self**, esta es muy parecida a lo que venimos trabajando a otros lenguajes con **this**. Y como su nombre lo dice hace referencia a los datos que componen la clase, en este caso `self.name` está llamando al atributo `name` que se encuentra en la línea 3 de la clase y, le está asignando el dato que se pasa en el método `__init__` de la línea 8.

```
1  class Account:
2      id          = int
3      name        = str
4      document    = str
5      email       = str
6      password    = str
7
8      def __init__(self, name, document):
9          self.name      = name
10         self.document   = document
```



Ahora veamos la clase Car:

```
1  from account import Account
2
3  class Car:
4      id          = int
5      license     = str
6      driver      = Account("", "")
7      passenger   = int
8
9      def __init__(self, license, driver):
10         self.license = license
11         self.driver   = driver
```



Lo que notarás de diferente es que cambiamos el tipo de dato de **driver**, ahora es de tipo `Account` y como ves está solicitando los dos datos obligatorios para instanciar un objeto de este tipo. Esto lo verás más en acción en el próximo fragmento de código del archivo `main.py`. Además, mucho ojo, en la primera línea observamos que es importante importar la clase para poderla usar.

Nuestro archivo `main.py` ahora se verá así:

```
1  from car import Car
2  from account import Account
3
4  if __name__ == "__main__":
5      print("Hola Mundo")
6
7      car = Car("AMS234", Account("Andres Herrera", "ANDA876"))
8      print(vars(car))
9      print(vars(car.driver))
10
11
```

Observa que estamos importando las dos clases que usaremos y las estamos instanciando en los métodos constructores.

Los resultados serán los siguientes:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  1: bash
MacBook-Pro-de-Anahi:Python anahisalgado$ python main.py
Hola Mundo
{'driver': <account.Account instance at 0x104be4ef0>, 'license': 'AMS234'}
{'document': 'ANDA876', 'name': 'Andres Herrera'}
MacBook-Pro-de-Anahi:Python anahisalgado$
```

El código de este ejemplo lo encuentras en este enlace:

<https://github.com/anncode1/Curso-POO-Platzi/tree/3.2.ConstructorPython>

## Reto 3.

Ahora que ya viste cómo creamos un método constructor en Python, mira esta clase

<https://platzi.com/clases/2034-php-poo/32133-polimorfismo/> y hazlo también para PHP.

Compártenos tus resultados en la sección de discusiones.