


Open · Opened 4 days ago by  **Rubén Montero**

Atributos

Resumen

- Veremos cómo se declaran los *atributos* en una clase Python
- Añadiremos en nuestro constructor el atributo `self.name`

Descripción

En Java solíamos especificar atributos de la clase **fuera** de los métodos:

```
public class JavaClass {  
    private String name;  
    private int age;  
}
```

En Python, si hacemos *algo parecido a eso*, lo estamos haciendo **mal**.

Debemos conocer que la **siguiente sintaxis existe**, pero **no** la usaremos:

```
class PythonClass:  
    name = "Something" # NO es lo que esperas  
    age = 42
```

Sirve para definir variables que [son comunes a todas las instancias de la clase](#). **No** es lo que esperamos. Son más bien como **constantes** o **atributos estáticos** de Java.

Entonces, ¿cómo se definen atributos de instancia en Python?

Dentro del método constructor.

¿Cómo?

Se declaran como *variables de self*.

Y ya está. Una vez hecho esto, estarán disponibles para *tu instancia* durante toda su vida.

¿Por ejemplo?

Sí tenemos un clase `Circle`, podemos darle un atributo `radius` así:

```
class Circle:  
  
    def __init__(self):  
        self.radius = 5
```

¿El método constructor puede recibir parámetros?

Sí. Basta con declararlos.

```
class Circle:  
  
    # En Python NO puede haber más de un método constructor  
    def __init__(self, radius):  
        self.radius = radius
```

¿Para qué sirven los atributos?

Como ya sabes, la utilidad de los atributos es almacenar información de una instancia.

Puede ser usada más tarde o servir a multitud de propósitos.

```
class Circle:

    def __init__(self, radius):
        self.radius = radius

    def calculate_circumference(self):
        return 2 * 3.1416 * self.radius
```

La tarea

Se pide que añadas *un atributo* al método constructor que ya existe en `ChatBot` :

- La firma del constructor *seguirá siendo* `__init__(self)` , sin parámetros
- Además del `print` que ya se hace, se **inicializará** el atributo `self.name` a `"Atenea"`

Por último

Verifica que tu código pasa el *test* asociado a la tarea.

Haz `commit` y `push` para subir los cambios al repositorio.



Rubén Montero @ruben.montero changed milestone to [%Sprint 3](#) 4 days ago