


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

Método constructor

Resumen

- Veremos el método especial `__init__`
- Comprenderemos cuándo es invocado

Descripción

En Java hemos manejado el concepto de [método constructor](#).

Recordemos que se trata de un *método* cuyo código se ejecuta **al crear una nueva instancia**.

O sea, que si en Python *pudiéramos* implementar un *método constructor* para nuestro `ChatBot`, **esperaríamos** que su código se ejecutase **en la siguiente línea de `run_class_example.py`**:

```
if __name__ == '__main__':  
    my_object = ChatBot() # Aquí se instancia el objeto, por lo tanto, tiene efecto el método constructor
```

¿Podemos *implementar* un método constructor en Python?

¡Sí!

¿Cómo?

Como *cualquier* otro método, pero **usando el nombre especial `__init__`** (dos barras bajas (`_`) a cada lado).

También debe declarar el parámetro `self`:

```
class UnaClase:  
  
    def __init__(self):  
        # Este código se ejecuta al instanciar la clase  
        pass
```

La tarea

Se pide que añadas un **método** constructor a `ChatBot`:

- Únicamente, hará un *print* indicando `"ChatBot encendido"`

Si re-lanzas `run_class_example.py` sin cambiar nada en ese fichero, ¿qué ves en la consola?

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