


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

## Bucles for

### Resumen

- Estrenaremos el paquete `advanced/`
- Crearemos un fichero `loops.py` que hará un bucle `for` básico en Python dentro de una función

### Descripción

Vamos a pasar de trabajar en el paquete `intro/` a `advanced/`.

En `advanced/`, **añade un nuevo fichero `loops.py`**

### Listas en Python [ ]

Aquí no hay `String[]`, `ArrayList<String>`, ....

**Todas** las listas son iguales, y contienen elementos de *cualquier* tipo.

Ejemplos:

```
lista1 = [1, 5, 6]
otro_array = ["Pepe", "Alice"]
lista_vacia = []
lista_heterogenea = [1, "nombre", None]
lista_de_listas = [ [1, 5], [7, 8] ]
```

### Bucles for en Python

En Python **no** existe el bucle `for` clásico, tipo C ó Java:

```
// NO EXISTE
for (int i = 0; i < numero; i++)
```

**Todos los bucles** son del estilo `for-each`.

### ¿Entonces cómo *recorro* una lista?

Así:

```
lista1 = [1, 5, 6]

for elemento in lista1:
    print(elemento)
```

### Un momento, un momento

### ¿Y si quiero recorrer índices del **1** al **50**, por ejemplo?

Puedes hacerlo usando la utilidad `range`:

```
# Imprime 0 1 2 3 4 5 6 7 8 9
for i in range(10):
    print(i)
```

### ¡Pruébalo!

**Añade** un fichero `loops.py` al **paquete que estrenaremos**:

- `advanced/`

Escribe una nueva función llamada `simple_for`, que **no** reciba parámetros.

- Dentro de `def simple_for():`, escribe un bucle `for` como el de arriba, que imprima los números de `0` al `61`

Puedes escribir un `if __name__ == '__main__':` para probarlo.

## Por último

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

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



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