

3 - Sentencias de repetición

Horas de Libre Configuración

Curso 2023- 2024



Recuerda



- ▶ No hay carácter especial para delimitar los bloques, simplemente se introduce una sangría (espacios o tabuladores).
- ▶ En Python se recomienda una sangría de 4 caracteres.

while



- ▶ Ejecuta un bloque de instrucciones *mientras* se cumpla una condición que se valida antes de cada iteración.

```
x = 0
```

```
while x < 10:
```

```
    print(x)
```

```
    x += 1
```

for



- ▶ Permite recorrer elementos iterables (string, lista, tupla, conjunto, diccionario)

```
# Ejemplo con un conjunto
datos = {1, 2, 3}
for d in datos:
    print(d)
```

range

- ▶ Crea un rango de números para usar en el bucle for.

```
range(inicio, fin, paso)
```

- ▶ Crea un rango de valores desde inicio hasta fin-1 con un incremento entre valores indicado en paso.

range

- ▶ inicio es opcional, su valor por defecto es 0.
- ▶ paso es opcional, su valor por defecto es 1.

EJEMPLO	INTERVALO CREADO
<code>range(5)</code>	<code>[0, 1, 2, 3, 4]</code>
<code>range(1, 5)</code>	<code>[1, 2, 3, 4]</code>
<code>range(1, 10, 2)</code>	<code>[1, 3, 5, 7, 9]</code>
<code>range(9, 0, -2)</code>	<code>[9, 7, 5, 3, 1]</code>

Variable contador

- ▶ Si no se necesita utilizar la variable contador, se puede obviar.

```
for _ in range(5):  
    print("Estoy en un bucle")
```

break y continue



- ▶ Tienen el mismo comportamiento que en otros lenguajes.
 - ▶ break finaliza la ejecución del bucle (intenta evitarlo).
 - ▶ continue salta a la siguiente iteración del bucle.

else



- ▶ Bloque de instrucciones que se ejecuta tras dejar de cumplirse el invariante.
- ▶ Si el bucle finaliza por un break, el bloque de finalización no se ejecuta.
- ▶ Se puede usar tanto en instrucciones while como en instrucciones for.

Bloque de finalización

EJEMPLO WHILE	EJEMPLO FOR
<pre>while x < 10: print(x) x += 1 else: print("Fin")</pre>	<pre>datos = {1, 2, 3} for d in datos: print(d) else: print("Fin")</pre>