A thick black L-shaped frame surrounds the text. It starts at the top left, goes right, then down, then right again at the bottom right.

INTRODUCCIÓN A LA PROGRAMACIÓN

5.- CICLOS DE ITERACIÓN

Rodrigo López A.

rilopez3@uc.cl

Ciclos de iteración (Loops)

- Durante el curso hemos aprendido:
 - *Definir variables*
 - *Construir una secuencia de acciones*
 - *Controlar el flujo de la secuencia*
- Pero que sucede cuando necesito repetir una acción?
 - *Un cantidad determina de veces*
 - *Una cantidad indeterminada de veces*



Funciones útiles

type(x)

- *Nos permite conocer el tipo de variable de x*

round(numero, n)

- *Nos permite redondear un numero real con la cantidad de n decimales que definamos*

range(n)

- *Genera una secuencia de números desde 0 hasta n (sin incluir n)*
- **range(5)** = > 0, 1, 2, 3, 4

range(i, n)

- *Genera una secuencia de números desde i hasta n (sin incluir n)*
- **range(2, 7)** = > 2, 3, 4, 5, 6

range(i, n, step)

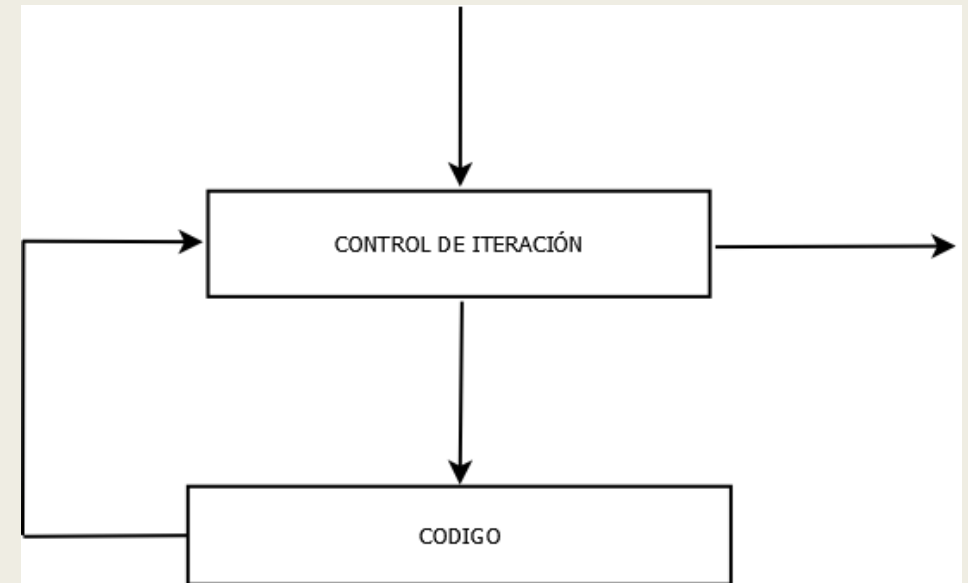
- *Genera una secuencia de números desde i hasta n (sin incluir n), dando step saltos*
- **range(2, 7, 2)** = > 2, 4, 6

Estructura FOR

- La estructura FOR nos permite repetir una acción una cantidad determinada de veces
- Se le puede pasar una secuencia o conjunto de valores a utilizar
- El tamaño de la secuencia define cuantas veces iterara (se repetirá) el ciclo

```
for x in range(6):  
    print(x)
```

➤ 0
➤ 1
➤ 2
➤ 3
➤ 4
➤ 5





```
1 for i in range(5):  
2     print('--ciclo')  
3     print(i)  
4     print('--fin')
```

```
--ciclo  
0  
--ciclo  
1  
--ciclo  
2  
--ciclo  
3  
--ciclo  
4  
--fin
```



```
1 for i in range( 5, 10):  
2     print('--ciclo')  
3     print(i)  
4     print('--fin')
```



```
--ciclo  
5  
--ciclo  
6  
--ciclo  
7  
--ciclo  
8  
--ciclo  
9  
--fin
```

```
[4] 1  for i in range(0, 10, 2):  
    2  |  print('--ciclo')  
    3  |  print(i)  
    4  print('--fin')
```

```
☞ --ciclo  
0  
--ciclo  
2  
--ciclo  
4  
--ciclo  
6  
--ciclo  
8  
--fin
```

```
[5] 1  for i in range(5, 0, -1):  
    2  |  print('--ciclo')  
    3  |  print(i)  
    4  print('--fin')
```

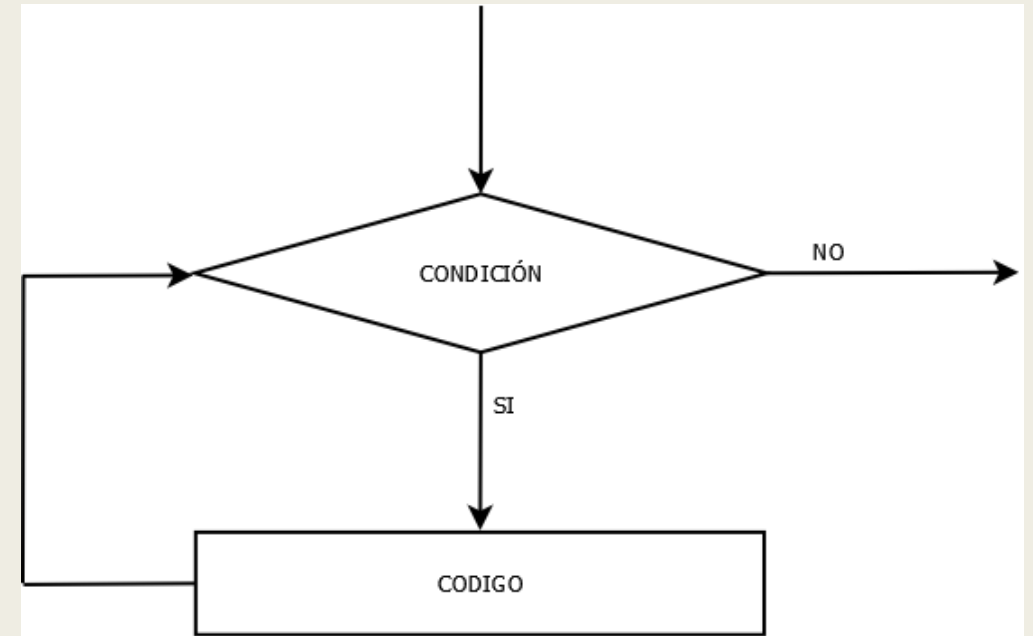
```
--ciclo  
5  
--ciclo  
4  
--ciclo  
3  
--ciclo  
2  
--ciclo  
1  
--fin
```

Estructura WHILE

- La estructura WHILE nos permite repetir una acción una cantidad no determinada de veces, mientras se cumpla la condición definida
- Importante dar fin al ciclo, con una condición de salida

```
x = 0
while(x <= 5):
    print(x)
    x = x + 1
```

➤ 0
➤ 1
➤ 2
➤ 3
➤ 4
➤ 5



```
1 contador = 0
2 while contador < 5:
3     print('--ciclo')
4     print(contador)
5     contador += 1
6 print('--fin')
```

```
--ciclo
0
--ciclo
1
--ciclo
2
--ciclo
3
--ciclo
4
--fin
```

```
[7] 1 todo_ok = False
2     contador = 5
3     while not todo_ok:
4         print('--ciclo')
5         print(contador)
6         contador -= 1
7         if contador == 0:
8             print('--condicion activada')
9             todo_ok = True
10    print('--fin')
11
12
```

```
--ciclo
5
--ciclo
4
--ciclo
3
--ciclo
2
--ciclo
1
--condicion activada
--fin
```


Repaso

- La estructura FOR nos permite repetir una acción una cantidad **determinada** de veces
- La estructura WHILE nos permite repetir una acción una cantidad **no determinada** de veces, mientras se cumpla la condición definida

