

Ciclos en Python

En Python tenemos dos estructuras para crear ciclos

- `while`
- `for`

Ciclo While

Podemos ejecutar un conjunto de instrucciones siempre que una condición sea verdadera

```
i = 1
while i < 1000:
    print(i)
    i = i + 1
```

Python cuenta con un tipo de datos especial que nos ayuda con el ciclo for, el cual nos devuelve una lista de `[0,...,n-1]`

```
a = list( range(1000) )
print( a )
```

Ciclo for

El ciclo for se usa para iterar sobre una secuencia (es decir, una lista, una tupla, un diccionario, un conjunto o una cadena).

Ejemplo 1

Iterar en un rango(lista)

```
for i in range(1001):
    print(i)
```

Ejemplo 2

Iterar lista

```
frutas = ["manzana", "limon", "mango"]
for x in frutas:
    print(x)
```

Ejemplo 3

Iterar una cadena

```
for x in "manzana":  
    print(x)
```

Ejemplo 4

La función range() por defecto es 0 como valor inicial, sin embargo, es posible especificar el valor inicial agregando un parámetro: range(2, 6) , lo que significa valores del 2 al 6 (pero sin incluir el 6):

```
for x in range(2, 6):  
    print(x)
```

La función range() por defecto incrementa la secuencia en 1, sin embargo, es posible especificar el valor del incremento agregando un tercer parámetro: range(4, 100, 3):

```
for x in range(4, 100, 3):  
    print(x)
```

Break y Continue

La instrucción **break** nos permite detener un ciclo while o for.

```
i = 1  
while i <= 12:  
    print(i)  
    if i == 3:  
        break  
    i += 1
```

```
fruits = ['manzana', 'banana', 'fresa', 'mango']  
for x in fruits:  
    if x == "banana":  
        break  
    print(x)
```

La instrucción `continue` nos permite detener la iteración actual y saltar a la que sigue.

```
i = 1
while i <= 12:
    i += 1
    if i == 3:
        continue
    print(i)
```

```
fruits = ['manzana', 'banana', 'fresa', 'mango']
for x in fruits:
    if x == "banana":
        continue
    print(x)
```