## **Control Flow**

[**python\_operators\_control\_flow**](https://drive.google.com/drive/folders/1Yys7-QOYZJIrGNUuknbv7yMUW0ZPJpJy)

\*\*\*\*\*

print(“\*”, **end=“ “**) → REPRESENTA LINEALMENTE EL PRINT EN UN FOR LOOP (hace que todo en una interacción se pinte en una misma línea, en vez de una linea por cada interacción)

\*\*\*\*\*

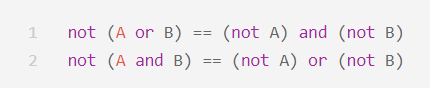
\*\*\*\*\*

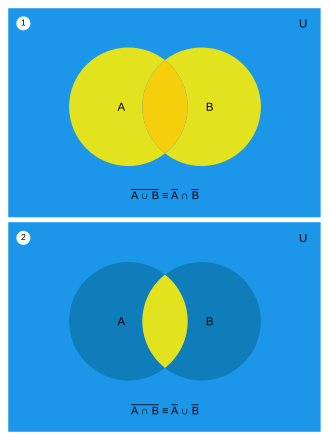
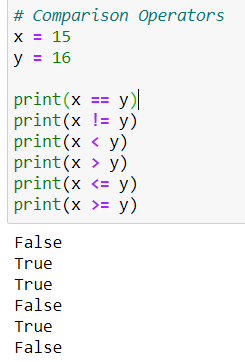
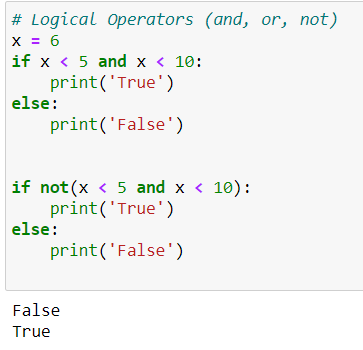
\n → salto de línea

\*\*\*\*\*

* **BOOLEAN CONDITIONS (Comparison Operators)**
* **==**
* **!=**
* **>**
* **>=**
* **<**
* **<=**

**→** se pueden mezclar con **and** y  **or**

****

** **

* **Identity and membership operators**

1. **is** → es igual a

var1 = “hola”

var2 = “hola”

var1 **is** var2

→ true

1. **in** → contiene

var1 = “hola”

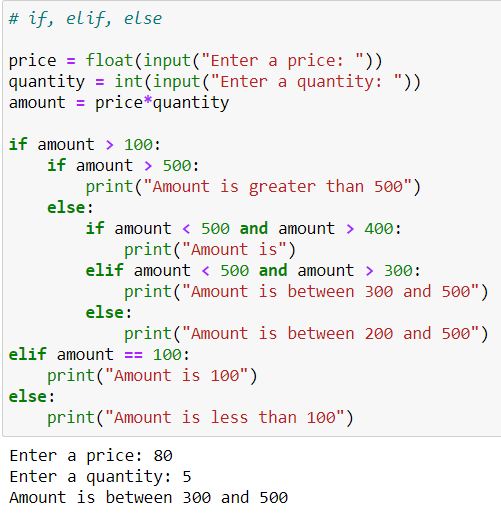
var2 = “holaaaa”

var1 **in** var2

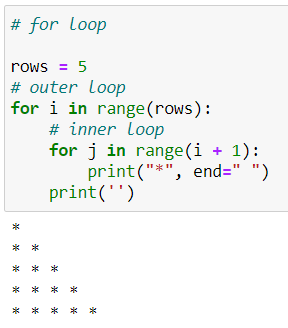
→true

* **Control Flow**
* **if**
* **elif**
* **else**
* **if not**

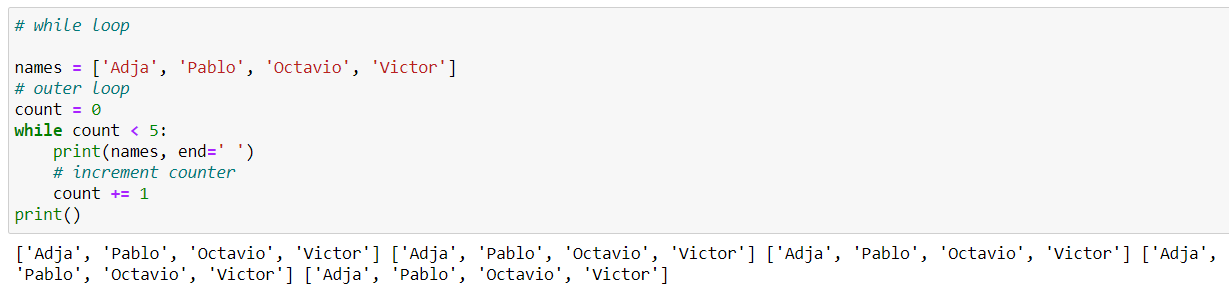
→ después de un else no puede haber un elif, si un if (empezaríamos otra condición nueva)

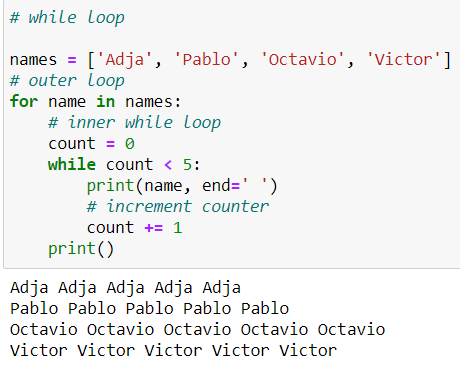


* **FOR:** bucle finito

****

* **WHILE** bucle infinito, para poder hacerlo finito creamos un **contador a cero** y en el **while le ponemos una condición** (es **como si fuese un if**). IMPORTANTE, el contador hay que sumarle un +1





**investigar WHILE LOOP**

CUIDADO con no crear un blucle infino dentro del while

count=0

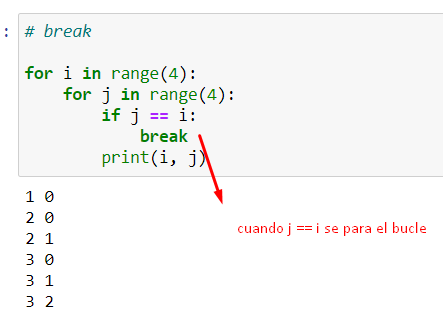
while count <= 0:

print (“hola”)

count += 1

* **break** como cuando en una función se genera el return, que se para la función

→ cuando el break se cumple rompe el loop



→ por eso el 0 0, 1 1, 2 2, 3 3 no están en los resultados

* **continue**

→ cuando el continue se cumple salta la linea (esa linea no ejecuta nada) y continua el loop

