

Bucles y Ciclos de Repetición

Utilizando una sentencia While podemos ejecutar el mismo bloque de código una y otra vez mientras que la condición que definamos dé como resultado True.

Para escribir un ciclo While en Python:

1. while -> Palabra reservada
2. Condición lógica (Una expresión que devuelva True o False)
3. : dos puntos
4. Indentación de 4 líneas que delimitan el inicio de un nuevo bloque de código.

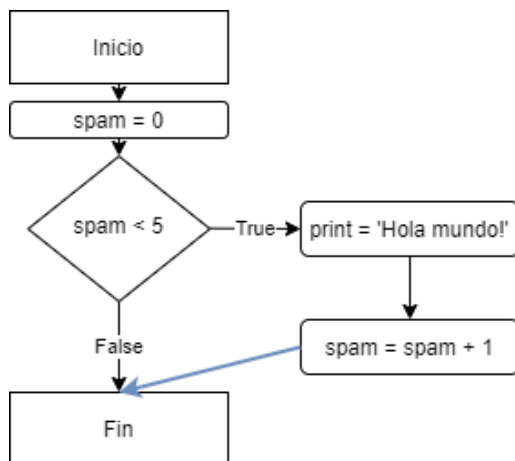
Es decir, una sentencia While es muy similar a un control de flujo if. La diferencia está en que al final de un if, un programa sigue ejecutando el resto de las líneas del programa. Pero cuando un programa llega a la última línea de un bloque de código de While, este salta al principio y vuelve a evaluar la condición que hayamos escrito.

Ejemplo:

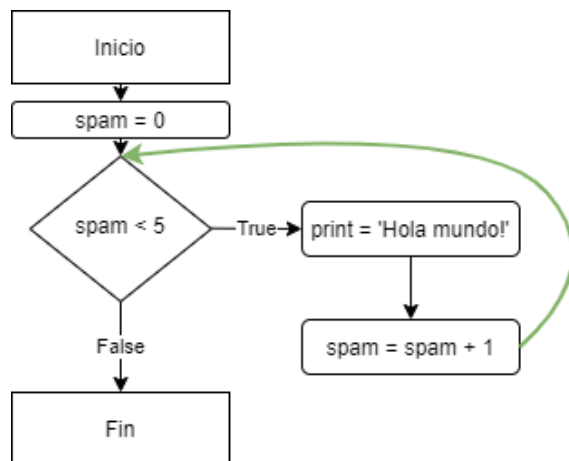
```
spam = 0
if spam < 5:
    print('Hola mundo!')
    spam = spam + 1
```

```
spam = 0
while spam < 5:
    print('Hola mundo!')
    spam = spam + 1
```

Estas dos sentencias son muy similares, pero al ejecutarlas el resultado será muy distinto. Para la que utiliza if el resultado será "Hola mundo!" y nada más. Pero la que utiliza While imprimirá ese mismo mensaje 5 veces.



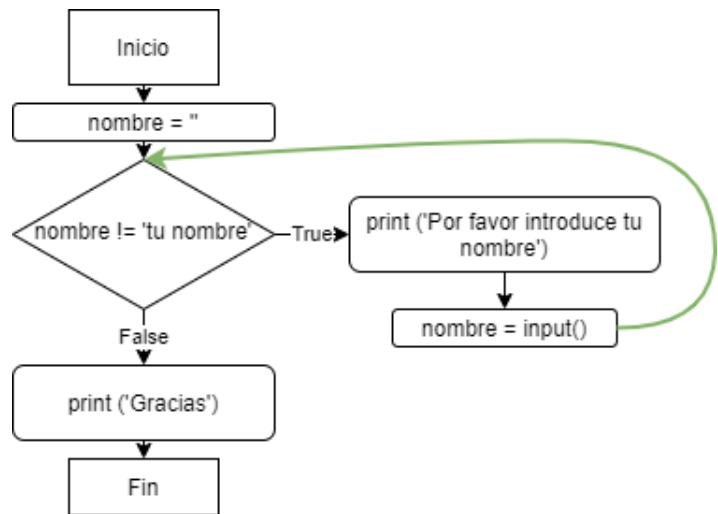
Control de Flujo IF



Ciclo de repetición While

Ejemplo:

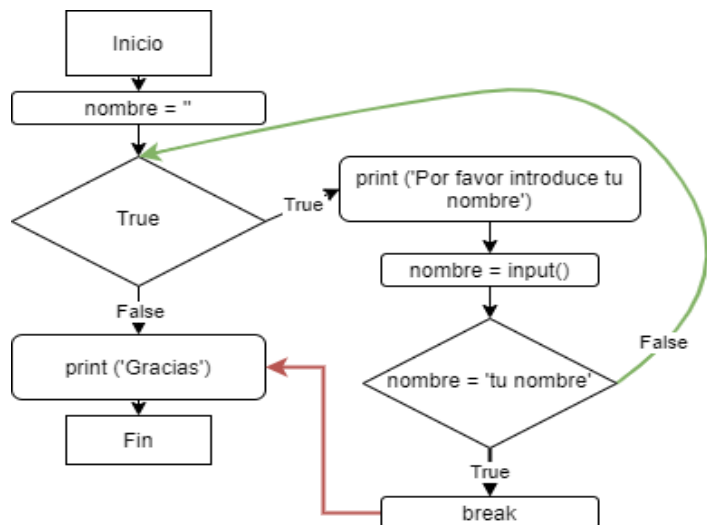
```
nombre=""
while nombre!='tu nombre':
    print('Por favor introduce "tu nombre"')
    nombre=input()
print('Gracias')
```



Break

Existe una manera de hacer que nuestro programa salga de un ciclo de repetición prematuramente. Si el programa ejecuta un **break** inmediatamente saldrá del bucle y continuará con la ejecución del resto del código.

```
while True:
    print('Por favor introduce "tu nombre"')
    nombre=input()
    if nombre=='tu nombre':
        break
print('Gracias')
```

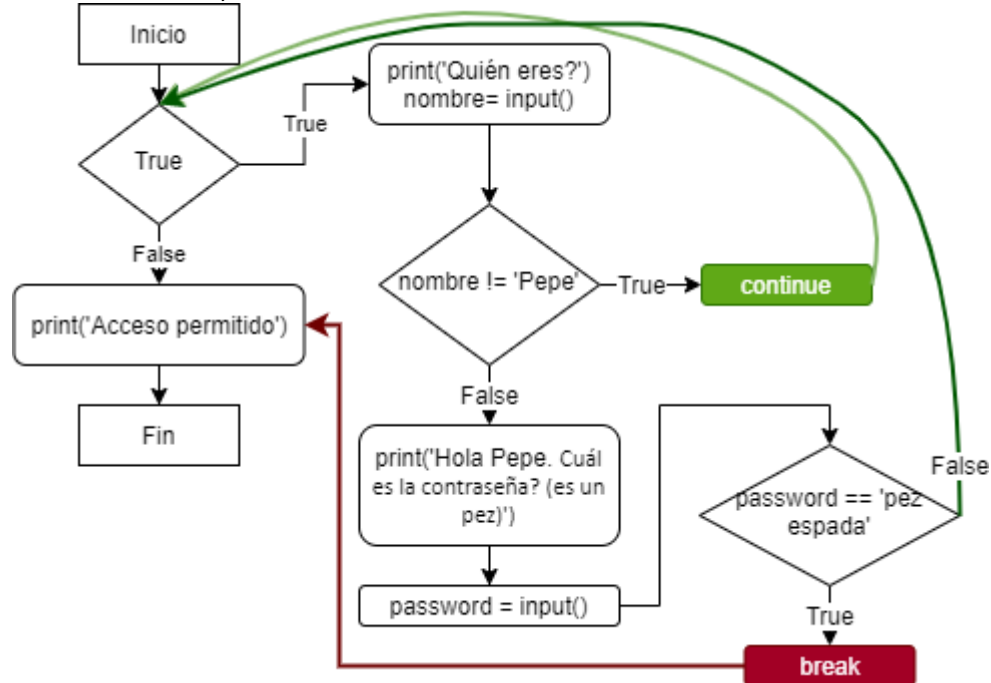


Continue

Al igual que el comando **break**, los **continue** se utilizan dentro de ciclos de repetición. Cuando el programa ejecuta un **continue**, inmediatamente saltará al comienzo del ciclo y evaluará de nuevo la condición lógica del while. Es decir hará lo mismo que como si hubiéramos llegado al final del ciclo.

```
while True:
    print('Quién eres?')
    nombre=input()
    if nombre!='Pepe':
        continue
    print('Hola, Pepe.Cuál es la contraseña? (es un pez)')
```

```
password=input()
if password=='pez espada':
    break
print('Acceso Permitido')
```



Ciclo de repetición For

El ciclo de repetición while sigue repitiéndose mientras que se cumpla una condición, pero en el caso de que queramos que el bloque de código se ejecute una determinada cantidad de veces podremos utilizar un ciclo de repetición **for** junto a la función **range()**

Para escribir un ciclo for en Python:

1. for -> palabra reservada.
2. una variable.
3. in -> palabra reservada.
4. llamada a la función range() -> se le podrán pasar hasta 3 argumentos.
5. : dos puntos.
6. Indentación de 4 líneas que delimitan el comienzo de un nuevo bloque de código.

```
for i in range(5):
    print(i)
```

Función Range

Algunas funciones como **range()**, aceptan más de un argumento y van separadas con una coma. En el caso de **range** nos permiten indicar en que número queremos que empiece el rango, hasta qué número debe llegar la variable, y cuanto será el incremento entre cada ciclo.

```
range(start, stop, step)
```