

Introducción a Python

Módulo 2

Bucle «*while*»

Bucle «while»

Los bucles son otra herramienta que permiten **modificar el flujo** de un programa.

Vimos que los condicionales pueden diferir la ejecución del código según se cumpla una condición u otra. Asimismo, los bucles repiten un bloque de código en tanto en cuanto se cumpla una condición.

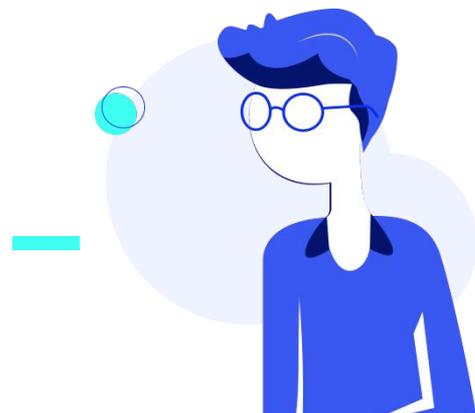
El bucle **while** en particular repite una porción de código siempre que una expresión sea verdadera.

Observemos la sintaxis con el siguiente ejemplo:

```
a = 1
while a < 5:
    print("Hola mundo")
```

Ejecutando esto en Geany, veremos que nuestro programa imprime “**Hola mundo**” infinitamente. Esto ocurre porque el bloque de código del bucle (es decir, `print("Hola mundo")`) se ejecuta siempre y cuando la condición (`a < 5`) sea verdadera. Puesto que `a` es `1` y no cambia su valor en todo el programa, la condición siempre es verdadera y el bucle se ejecuta infinitamente.

Para que un bucle sea útil debe terminar en algún momento. Hay dos formas de conseguirlo: que la condición se vuelva **falsa**, o ejecutar una instrucción para forzar al bucle a que termine.



Comencemos por la primera. Debemos hacer que `a < 5` sea falso en algún momento para que el bucle termine. Podemos lograrlo vía el siguiente código:

```
a = 1
while a < 5:
    print("Hola mundo")
    a = a + 1
```

Ejecutemos este programa y veremos que el resultado ahora son cuatro “**Hola mundo**” impresos en pantalla.

La lógica de este comportamiento se comprende muy fácilmente si entendemos cómo avanza un bucle **while**:

1. Chequear que la condición sea verdadera.
2. Si es verdadera, ejecutar el bloque de código.
3. Si es falsa, terminar el bucle (y también estos pasos).
4. Volver al paso **1**.

Así, nuestro bucle se ejecuta hasta que `a` se convierte en **5**, y en consecuencia la condición `a < 5` (esto es, `5 < 5`) se vuelve falsa.

El segundo método para finalizar un bucle es vía la palabra reservada **break**.

Cuando Python se encuentra con esta instrucción dentro de un bucle, éste termina abruptamente, por más que la condición siga siendo verdadera.

Nuestro código anterior puede replicarse usando **break** del siguiente modo:

```
a = 1
while True:
    if a < 5:
        print("Hola mundo")
        a = a + 1
    else:
        break
```

Poner **True** en lugar de una condición hace sencillamente que el bucle se ejecute **de forma infinita**. Luego nos encargaremos de terminarlo manualmente vía **break** cuando la condición (**a < 5**) sea falsa.



**¡Sigamos
trabajando!**