

Introducción a Python

Módulo 1

Condicionales

Observemos el siguiente código de ejemplo y analicemos parte por parte.

```
a = 12
if a > 10:
    print("a es mayor que 10.")
```

La primera línea define una variable **a** con el valor **12**, y la segunda línea indica que el código a continuación sólo debe ejecutarse cuando la condición **a > 10** sea verdadera. En efecto, si cambiamos **a = 12** por **a = 7**, veremos que, al correr el programa, la tercera línea será ignorada.

El bloque de código que será ejecutado según el resultado de la condición debe tener aplicada una sangría de cuatro espacios (esto también se conoce como «*indentación*», que es una transliteración del inglés *indentation*).

La sangría es necesaria para poder diferenciar el bloque que se encuentra dentro del condicional del resto del código. Por ejemplo:

```
a = 12
if a > 10:
    print("a es mayor que 10.")
print("Hola mundo")
```

Condicionales

Dijimos que el **código** de un programa es **ejecutado** por Python de **izquierda a derecha y de arriba hacia abajo**.

Los **condicionales** nos permiten diferir la ejecución de una porción de código -es decir, una o más líneas- según se cumpla una condición u otra; o, lo que es lo mismo, hacer que un bloque de código se ejecute únicamente cuando se cumpla una condición.

Es decir, **los condicionales nos permiten modificar el flujo del programa**.



En el ejemplo anterior, le estamos indicando a **Python** que únicamente la tercera línea debe ejecutarse si `a > 10`. En cambio, la cuarta está por fuera del condicional por cuanto no tiene sangría, y por ende se ejecute siempre independientemente del resultado de la condición.

Los **condicionales**, entonces, se constituyen por:

- La palabra reservada **if**.
- Una condición o expresión cuyo resultado debe ser **True** o **False** seguida de dos puntos.
- Una o más líneas con **sangría**.

En **Geany** la sangría se aplica automáticamente a la siguiente línea al presionar **Enter** luego de indicar los dos puntos al final de la condición. También podemos insertar manualmente sangría presionando la tecla **Tab** o cuatro veces la **barra espaciadora**.



La cantidad de código que podemos incluir dentro de un condicional es **ilimitada**. Incluso puede contener otras definiciones de variables y otros condicionales:

```
a = 12
if a > 10:
    print("a es mayor que 10.")
    b = 5
    if b == 5 or b == 3.14:
        print("b es 5 o 3.14.")
```

Ahora bien, consideremos esta variante:

```
a = 12
if a > 10:
    print("a es mayor que 10.")
else:
    print("a es menor o igual que 10.")
```

La sintaxis es bastante clara: la tercera línea se ejecutará cuando **a > 10** sea verdadero; y la quinta, cuando sea falso.

Nótese que las palabras reservadas **if** y **else** deben estar alineadas, es decir, deben tener la misma sangría (si bien en este ejemplo no tienen sangría en absoluto).

Por lo dicho es evidente que el código anterior es similar a los siguientes códigos:

```
if a > 10:
    print("a es mayor que 10.")
if not a > 10:
    print("a es menor o igual que 10.")
```

```
a = 12
if a > 10:
    print("a es mayor que 10.")
if a <= 10:
    print("a es menor o igual que 10.")
```

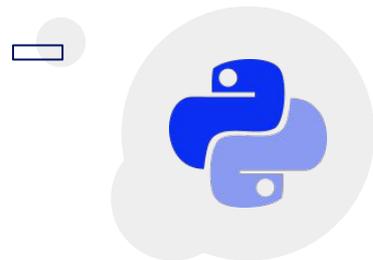
Nos resta la palabra reservada **elif**, que nos permite agregar más de una condición en un mismo condicional:

```
a = 5

if a == 1:
    print("a es 1.")
elif a == 2:
    print("a es 2.")
elif a > 2 and a < 10:
    print("a es mayor que 2 y menor que 10.")
else:
    print("a es mayor o igual que 10.")
```

Así como en un condicional del tipo `if/else` únicamente se ejecuta un bloque de código, del mismo modo ocurre aquí. Cuando tenemos un condicional con múltiples condiciones (en donde la primera se escribe con `if` y el resto con `elif`), todas son evaluadas por Python (por “evaluar” entendemos determinar si son falsas verdaderas o falsas) en el orden en el que están dispuestas en el código.

De este modo, si `a == 1` es verdadero, se ejecuta `print("a es 1.")` y el condicional termina; es decir, el resto de las condiciones no son consideradas. Pero si `a == 1` es falso, entonces se considera la segunda condición, `a == 2`, y así sucesivamente hasta llegar al `else` en caso de que todas las condiciones anteriores sean falsas.



**¡Sigamos
trabajando!**