Sentencia `if/elif/else`

1. ¿Qué son las estructuras de control de flujo?

Todo lo que hemos visto hasta ahora han sido elementos de Python que se ejecutan de manera secuencial y exactamente en el orden especificado.

Sin embargo, con frecuencia un programa necesita cambiar el control del flujo de ejecución de las sentencias y realizar cosas como: saltarse algunas sentencias, ejecutar una serie de sentencias de forma repetitiva, o elegir entre conjuntos alternativos de sentencias a ejecutar.

Aquí es donde entran en juego las estructuras de control. Una estructura de control nos permite modificar el flujo de ejecución de un programa.

2. ¿Qué es la sentencia `if` ?

La sentencia de control de flujo `if` es probablemente una de las estructuras más importantes de cualquier lenguaje de programación, incluido Python.

Esta estructura nos permite implementar sentencias condicionales dentro de nuestro programa.

La sintaxis utilizada para definir la sentencia `if` es la siguiente:

```
if <expresión>:
    <sentencia(s)>
```

`<expresión>` es una expresión evaluada en un contexto booleano. El resultado de evaluar esta expresión debe ser `True` o `False`

`<sentencia(s)>` es el bloque de sentencias en Python que se ejecutará cuando el resultado de evaluar la expresión sea `True` , en caso contrario se omitirá.

In [2]:
```
num1 = 5
num2 = 10
```

In [3]:
```
# Esto sería una expresión
num1 < num2
```

Out[3]:
```
True
```

In [4]:
```
if num1 < num2:
    print("Sentencia 1")
    print("Sentencia 2")
    print("Sentencia 3")
```

```
Sentencia 1
Sentencia 2
Sentencia 3
```

In [5]:
```
num1 = 15
```

In [6]:
```
if num1 < num2:
    print("Sentencia 1")
    print("Sentencia 2")
    print("Sentencia 3")
```

In [7]:
```
num1 < num2
```

Out[7]:
```
False
```

IMPORTANTE: Recuerda que en Python todo el codigo que sea local a una estructura, por ejemplo una función o una estructura de control como `if` , debe esta "identado", es decir, respetar la sangría.

In [8]:
```
if num1 < num2:
    print("Sentencia 1")
    print("Sentencia 2")
    print("Sentencia 3")
print("Sentencia fuera del if")
```

```
Sentencia fuera del if
```

3. ¿Qué expresiones y operadores podemos utilizar con la sentencia `if` ?

Una de las cosas interesantes de esta sentencia de control de flujo es que podemos utilizar casi cualquier tipo de operador, tipo de dato y expresión siempre que el resultado sea un valor booleano (`True` o `False`)

In [9]:
```
lista = ["azul", "amarillo", "verde"]
```

In [10]:
```
if "azul" in lista:
    print("Sentencia 1 en if")
    print("Sentencia 2 en if")
```

```
Sentencia 1 en if
Sentencia 2 en if
```

In [12]:
```
tupla = (1, 2, 3, 4)
```

In [15]:
```
if 6 in tupla:
    print("Sentencia 1 en if")
    print("Sentencia 2 en if")
```

4. La cláusula `else`

En agunas ocasiones nos encontraremos con casos de uso en los que querremos una estructura condicional que ejecute unas sentencias en Python si la expresión es `True` pero además, que ejecute otras sentencias diferentes si la expresión es `False` . Para este tipo de casos se utiliza la cláusula `else` .

La sintaxis que debemos utilizar en Python 3 para definir la cláusula `else` dentro de una estructura `if` es la siguiente:

```
if <expresión>:
    <sentencia(s)>
else:
    <sentencia(s)>
```

In [16]:
```
num1 = 5
num2 = 10
```

In [20]:
```
if num1 < num2:
    print("Sentencias si True")
else:
    print("Sentencias si False")
```

```
Sentencias si True
```

In [21]:
```
if num1 > num2:
    print("Sentencias si True")
else:
    print("Sentencias si False")
```

```
Sentencias si False
```

5. La cláusula `elif`

Otros de los casos de uso que nos encontraremos con frecuencia es la implementación de una estructura condicional que requiera múltiples evaluaciones de varias expresiones para tomar la decisión de qué código ejecutar. Para estos casos específicos podemos utilizar la cláusula `elif` .

La sintaxis que debemos utilizar en Python 3 para definir la cláusula `elif` dentro de una estructura `if` es la siguiente:

```
if <expresión>:
    <sentencia(s)>
elif <expresión>:
    <sentencia(s)>
elif <expresión>:
    <sentencia(s)
...
else:
    <sentencia(s)>
```

Una cosa importante que debemos tener en cuenta es que podemos utilizar tantas cláusulas `elif` como consideremos oportunas.

In [24]:
```
nombres = ["Santiago", "Laura", "Julia"]
```

In [25]:
```
if "Pedro" in nombres:
    print("Hola Pedro")
elif "Juan" in nombres:
    print("Hola Juan")
```

```
elif "Marta" in nombres:
    print("Hola Marta")
elif "Santiago" in nombres:
    print("Hola Santiago")
else:
    print("El nombre no esta en la lista")
```

```
Hola Santiago
```

In [26]:
```
nombres = ["Laura", "Julia"]
```

In [27]:
```
if "Pedro" in nombres:
    print("Hola Pedro")
elif "Juan" in nombres:
    print("Hola Juan")
elif "Marta" in nombres:
    print("Hola Marta")
elif "Santiago" in nombres:
    print("Hola Santiago")
else:
    print("El nombre no esta en la lista")
```

```
El nombre no esta en la lista
```

6. Sentencia `if` en una sola línea

Como curiosidad interesante, Python 3 nos permite utilizar una sintaxis específica para implementar sentencias `if` en una sola línea de código. Sin embargo, en general esta opción no es muy recomendable debido a que puede aumentar la complejidad en la lectura del código fuente.

In [30]:
```
if 4 > 2: print("Sentencia 1");print("Sentencia 2");print("Sentencia 3")
```

```
Sentencia 1
Sentencia 2
Sentencia 3
```

In [33]:
```
if 4 < 2: print("Sentencia 1")
else: print("Sentencia 2")
```

```
Sentencia 2
```

7. Operador condicional

Por último, Python soporta una construcción más relacionada con la sentencia `if` que puede resultarnos muy interesante para determinados casos de uso. A esta construcción se le denomina operador condicional.

La sintaxis que se utiliza para implementar el operador condicional en Python3 es la siguiente:

```
<expresión1> if <expresión_condicional> else <expresión2>
```

Este operador nos permite simplificar el código que utilizamos para determinadas tareas como, por ejemplo, asignarle un valor a una variable en función de una condición.

In [43]:
```
nombre = "Pedro"
```

In [44]:
```
edad = 30 if nombre == "Pedro" else 15
```

In [45]:
```
edad
```

Out[45]:
```
30
```

In [51]:
```
tiempo="lluvia"
```

In [52]:
```
print("Vamos", "a la piscina" if tiempo=="sol" else "al cine")
```

```
Vamos al cine
```

In [ ]:
```

```