Condicionales

Los condicionales nos permiten que el programa ejecute un conjunto de sentencias o no según se cumpla una condición sobre el contexto de ejecución del programa. Esta condición debe ser una expresión booleana, es decir, que evalúa a Verdadero o Falso.

Los condicionales nos permiten que el programa tome decisiones según el contexto de ejecución. Es decir, puede comportarse de una manera u otra según el contexto.

A continuación, veremos las diferentes formas que provee Python para utilizar los condicionales.

Ejecución condicional

La ejecución condicional nos permite ejecutar o no un conjunto de sentencias. Para esto Python nos provee la sentencia *if.*

La sintaxis básica para una sentencia if es:

```
if EXPRESION BOOLEANA:
   SENTENCIAS
```

Por ejemplo:

```
x = 2
if x >= 0:
   print(“El número”, x, “es positivo”)
```

En el ejemplo, la variable x tiene el valor 2, con lo cual se imprimirá en pantalla el mensaje “El número 2 es positivo”. Si la variable x tuviera, por ejemplo, el valor -1 entonces no se imprimirá ningún mensaje en pantalla.

Ejecución alternativa

La ejecución alternativa nos permite ejecutar un bloque de código u otro según se cumpla una condición o su complemento.
Para esto Python nos provee las sentencias if...else.
La sintaxis de la ejecución alternativa es:

```
if EXPRESION BOOLEANA:
   SENTENCIAS
else:
   SENTENCIAS
```

Por ejemplo:

```
x = 2
if x >= 0:
   print("El número", x, "es positivo")
else:
   print("El número", x, "es negativo")
```

Este ejemplo, es similar al anterior. Solo que si la variable x tiene asignado un número negativo, por ejemplo, el -1 entonces imprimirá en pantalla el mensaje "El número -1 es negativo".

Condicionales encadenados

Python nos permite hacer condicionales encadenados con la sentencia *elif*. Esta forma de escribir los condicionales nos servirá cuando tengamos más de 2 condiciones disjuntas. Se pueden poner tantas condiciones como uno requiera. Y además, si no se cumple ninguna de las condiciones se puede ejecutar el bloque de código que figura en el *else*.

La sintaxis de los condicionales encadenados es:

```
if EXPRESION BOOLEANA:
   SENTENCIAS
elif EXPRESION BOOLEANA:
   SENTENCIAS
...
elif EXPRESION BOOLEANA:
   SENTENCIAS
else:
   SENTENCIAS
```

Por ejemplo:

```
x = 2
if x > 0:
   print("El número", x, "es positivo")
elif x < 0:
   print("El número", x, "es negativo")
else:
   print("El número es igual a cero")
```

En este ejemplo, dividimos las tres posibilidades escribiendo las dos condiciones, si x es mayor que cero y si x es menor que cero, y si no se cumplen ninguna de esas condiciones es porque x es cero.

Condicionales anidados

Python nos permite anidar los condicionales. Es decir, dentro de las sentencias que se ejecutan dentro del cuerpo del if y/o del else poner otro condicional.
Si bien el lenguaje lo permite, en muchos casos, el código queda más complejo de entender. Dependerá de las condiciones que tengamos que programar, que sea mejor escribirlo de una forma u otra.

La sintaxis para anidar condicionales es:

```
if EXPRESION BOOLEANA:
   if EXPRESION BOOLEANA:
      SENTENCIAS
   else:
      SENTENCIAS
else:
   if EXPRESION BOOLEANA:
      SENTENCIAS
   else:
      SENTENCIAS
```

Por ejemplo:

```
x = 2
if x > 0:
   print("El número", x, "es positivo")
else:
```

```
if x < 0:
   print(“El número”, x, “es negativo”)
else:
   print(“El número es igual a cero”)
```

Condiciones

Dedicamos este apartado a explicar un poco qué se puede poner en las condiciones. Como mencionamos, puede ser cualquier expresión booleana, es decir, una expresión que devuelva un valor booleano.
Con lo cual, podemos utilizar todos los operadores relacionales o de comparación y además formar expresiones compuestas uniendo distintas expresiones booleanas con los operadores lógicos.