



Recordando el método input

El método `input()` permite el ingreso de información por parte del usuario, como se ve en el ejemplo anterior, el programa pregunta por un carácter para ejecutar una acción.

Puede ser una palabra, números o frases completas, sin embargo el método `input()` recibe esta información como una cadena de texto, es importante tenerlo presente porque al momento de requerir un número o un flotante se debe hacer la conversión de tipo.

Los datos ingresados pueden ser almacenados en una variable para gestionarla dentro del código.

```
[ ] edad = int(input('Ingrese su edad: '))  
  
print(f'Usted tiene {edad} años')
```

```
Ingrese su edad: 30  
Usted tiene 30 años
```

if

Ahora que se conocen los operadores condiciones, es importante conocer las estructuras de control de flujo. Se debe tener un mecanismo que le permita hacer algo si se cumple una condición, y no hacerlo si no se cumple.

Por ejemplo, sales a caminar si el clima es bueno, o te quedas en casa si está húmedo y frío.

Se habla de control de flujo, al utilizar la sentencia `if`.

```
if true_or_not:  
    do_this_if_true
```

- Si la expresión `true_or_not` es verdadera (es decir, su valor no es igual a cero), las sentencias `do_this_if_true` se ejecutarán.
- Si la expresión `true_or_not` es falsa (es decir, su valor es igual a cero), las sentencias con sangría se omitirán (ignorado), y la siguiente instrucción ejecutada será la siguiente al nivel de la sangría original.



El bloque de control puede complementarse con `else` y con `elif` para preguntar en repetidas ocasiones si una condición se cumple o cuál condición se cumple. La sentencia `else` es opcional.

El condicional admite diferentes operadores de comparación para obtener un `True` o un `False`

Ahora que se habla del bloque condicional, hay que tener presente que Python funciona con indentación, es decir, una especie de sangría para indicar que una línea pertenece a un bloque. Esa indentación o espaciado se recomienda hacerlo a 4 espacios, algunos editores de código ya cuentan por defecto con esta configuración.

Ejercitación

Escriba con una sentencia `if` la siguiente expresión

```
si el clima es bueno, saldremos a caminar  
  
después, almorzaremos
```

- Podría mejorar las ejercitaciones anteriores (vistas con operadores de comparación) utilizando el bloque `if` para entregar una respuesta?
- Aquí explicar que almorzar está por fuera del condicional, lo que indica que es independiente del resultado de la condición.

A estas condiciones, donde solo se tiene un camino de acciones a realizar en caso de que la condición se haya cumplido, se le conocen también como condicionales simples.



if-else

Partiendo de la frase anterior, se puede observar que en caso de que el clima no sea bueno, no hay ninguna acción al respecto, algo como por ejemplo

```
Si el clima es bueno, saldremos a caminar, de lo contrario, iremos al cine.
```

Ahora, la nueva frase contempla un plan B, el cual se realiza cuando el plan A no se puede llevar a cabo. Su estructura es similar a la sentencia if, solo que ahora agrega otro bloque dentro de una estructura llamada else.

```
if true_or_false_condition:  
    perform_if_condition_true  
else:  
    perform_if_condition_false
```

Como se puede apreciar, el else no lleva una condición, es decir que en esta estructura if-else en caso de no ocurrir lo primero, no importa que haya ocurrido, se ejecuta lo segundo.

Si la condición se evalúa como verdadera, la instrucción `perform_if_condition_true` se ejecuta, y la sentencia condicional llega a su fin;

Si la condición se evalúa como falsa, la instrucción `perform_if_condition_false` se ejecuta, y la sentencia condicional llega a su fin.

Ejercitación

Transcribir la frase



Si el clima es bueno, saldremos a caminar, de lo contrario, iremos al cine.

después, almorzaremos

A un bloque de código utilizando **if-else**

Un elemento importante adicional, es que los condicionales se pueden anidar, esto quiere decir que dentro de un condicional if o else puedo tener más condicionales, siempre respetando la indentación del bloque.

```
if true_or_false_condition:
    perform_if_condition_true
    if another_condition_true:
        run_this()
    else:
        another_false()
else:
    perform_if_condition_false
```

Ejercitación

Qué tal ahora escribir la siguiente frase utilizando condicionales anidados.

Si hay buen clima, saldremos a caminar. Si encontramos un buen restaurante, almorzaremos allí. De lo contrario, vamos a comer un sandwich.

Si hay mal clima, iremos al cine. Si no hay boletos, iremos de compras al centro comercial más cercano.



¿Qué tal si ahora se agrega un operador condicional lógico, puede identificar en donde se encuentra? y cuál sería el resultado de la operación?

Deben recordar que el else pertenece al if que se encuentra al mismo nivel de bloque, sin importar que dentro de ellos se encuentren más estructuras anidadas.

```
[ ] temperature = 24

if temperature < 20:
    if temperature < 10:
        print('Nivel azul')
    else:
        print('Nivel verde')
else:
    if (temperature < 30 and temperature >25): #Uso de operador condicional lógico
        print('Nivel naranja')
    else:
        print('Nivel rojo')

#Si por algún motivo el bloque condicional no tiene una instrucción a ejecutar,
#puede utilizarse la palabra reservada `pass` para evitar que la aparición de un error
if(temperature == 24):
    pass
```

Ejercitación

Haga que el valor de la temperatura sea ingresada por el usuario. Puede también definir los valores máximos o límites de comparación solicitando que sean ingresados por el usuario.

elif

elif se usa para verificar más de una condición, y para detener cuando se encuentra la primera sentencia verdadera.

Algo común, es que se requiera preguntar por varios casos o condiciones hasta encontrar una que coincida con el criterio, otros lenguajes implementan una estructura llama switch, en Python, se debe implementar algo similar utilizando bloques if-elif-else.

Suponga que se quiere presentar un saludo a un usuario dependiendo del rango de su edad. La lógica de su funcionamiento es, inicio con el bloque if, si la condición definida no es verdadera por ejemplo.



```
edad = 38

if(edad > 18 and edad < 25)
    print("Hola joven")
else:
    print("Hola viejo")
```

Se puede comprobar que el valor de edad no entra en el rango, pero se se crea un else que diga "Hola viejo", se estaría encasillando un rango muy amplio que puede no ser correcto.

Si se quiere utilizar condiciones anidadas, se tendría algo como lo siguiente:

```
edad = 38

if(edad >= 18 and edad < 25):
    print("Hola joven")
else:
    if(edad >= 25 and edad < 36):
        print("Hola adulto joven")
    else:
        if(edad >= 36 and edad < 50):
            print("Hola señor")
        else:
            if(edad >= 50 and edad < 70):
                print("Hola adulto mayor")
            else:
                ...
```

Puede no ser algo tan fácil de leer, utilizando elif no requiere crear la anidación.

```
if(edad >= 18 and edad < 25):
    print("Hola joven")
elif(edad >= 25 and edad < 36):
    print("Hola adulto joven")
elif(edad >= 36 and edad < 50):
    print("Hola señor")
elif(edad >= 50 and edad < 70):
    print("Hola adulto mayor")
else:
    print("Fuera de aquí")
```



Ejercitación clima

Ahora, utilizando if-elif-else como quedaría la siguiente frase:

```
si hay buen clima, saldremos a caminar, de lo contrario,  
si obtenemos entradas, iremos al cine, de lo contrario,  
si hay mesas libres en el restaurante, vamos a almorzar;  
si todo falla, regresaremos a casa y jugaremos ajedrez.
```

- No debes usar else sin un if precedente;
- else siempre es la última rama de la cascada, independientemente de si has usado elif o no;
- else es una parte opcional de la cascada, y puede omitirse;
- Si hay una rama else en la cascada, solo se ejecuta una de todas las ramas;
- Si no hay una rama else, es posible que no se ejecute ninguna de las opciones disponibles.

```
[ ] a = 4  
    b = 2  
  
if(a == b):  
    print('a es igual a b');  
elif(a < b):  
    print('a es mejor que b')  
else:  
    print('a es mayor que b')  
  
#También existen el atajo para el if y el atajo para el if ... else  
print("A") if a > b else print("B")  
if a > b: print("a es mayor que b")
```

Ejercitación Par o impar?

Cree un código que permita identificar si un número es par, impar o si es 0, si el número es impar, determine si es divisible entre 3.