



Taller de Programación Web

Estructuras de control de flujo - condicionales



Subsecretaría de
Empleo
Chaco Gobierno de todos



Ministerio de
Producción, Industria y Empleo
Chaco Gobierno de todos



CHACO
Gobierno de todos

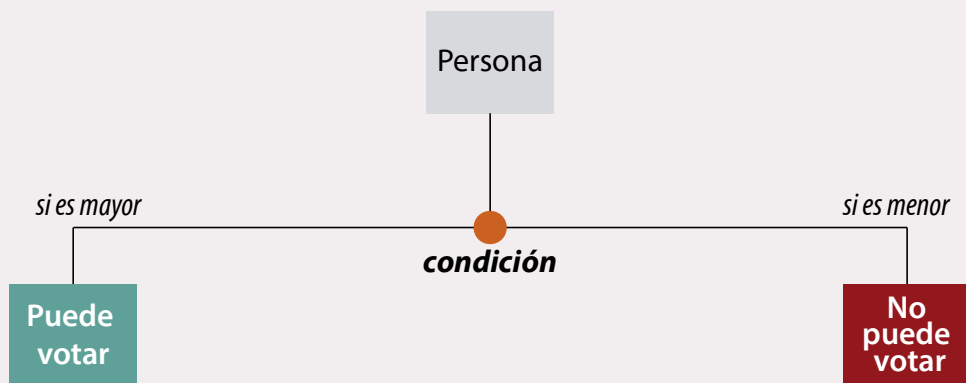


INTRODUCCIÓN

Como vimos hasta ahora, estuvimos ejecutando instrucciones de código de forma plana, esto quiere decir que, el código se estaba leyendo línea por línea una tras otra, de forma secuencial. Pero, ¿qué pasa cuando queremos el programa “tome una decisión” de realizar una acción u otra acción en base a instrucciones que le damos?

Por ejemplo: queremos saber si una persona es mayor de edad o menor de edad para votar o no.

En un caso así, necesitamos que el programa haga una cosa si se cumple que la persona es mayor de edad o, que realice otra cosa si la persona es menor de edad.



Como podés ver, mediante una **condición**, la cual contiene una pregunta, el programa va a tomar una forma de avanzar u otra. Es decir, gracias a las **estructuras de control de flujo** el programa va a saber como “actuar” ante una determinada situación, e incluso, repetir una tarea si es necesario.

Los tipos de **estructuras de control** que veremos son:

CONDICIONALES

REPETITIVAS



ESTRUCTURAS DE CONTROL CONDICIONALES

Estas estructuras realizan la evaluación de una condición y, según el resultado, el programa va a realizar una determinada acción.

Para dar estas condiciones, utilizamos expresiones lógicas. La estructura que veremos es:

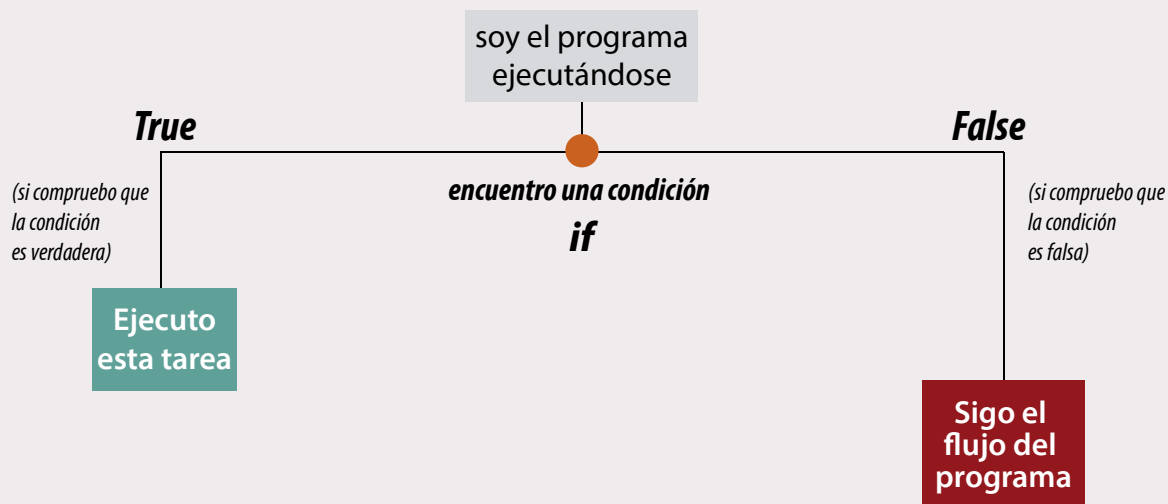
if

```
1  if True:
2      print('Voy a hacer esta tarea.')
3      print('Se cumplió esta condición.')
4
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL

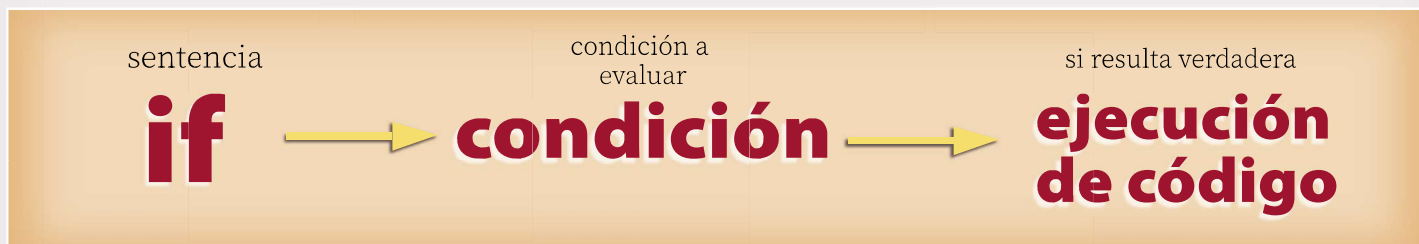
Voy a hacer esta tarea.
Se cumplió esta condición.

Como podés ver, para que la condición se cumpla, la expresión que se comprueba tiene que ser verdadera (True). Si la expresión resulta falsa (False), es decir, no se cumple, el programa sigue su recorrido sin ejecutar lo que hay dentro del **if**





Entonces, analizando la estructura de **if** (que, por si aún no te diste cuenta, significa SI en inglés) podemos ver que tiene 3 partes:



Así que ahora que lo comprendés mejor vamos a darte el primer ejemplo en código:

```
1 edad = 19
2 if edad >= 18:
3     print('Usted puede votar.')
4
```

Bloque de código

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL

Usted puede votar.

Como ves, si el valor de la variable “edad” es mayor o igual a 18 (es decir, True, como en este caso), se ejecuta el bloque de código que hay dentro de **if**. Si esto no fuese verdad (es decir False), no se ejecutaría lo que hay dentro del **if** y el programa seguiría con las siguientes líneas de código, pasando por alto todo el bloque de código dentro del **if**. Además, si notaste, usamos un operador comparación para la condición.

Otra parte importante en la sintáxis para que todo sea ejecutado correctamente, es la indentación

Este espacio que hay entre el comienzo de la línea de código y el código que se escribe, se llama indentación

```
1 edad = 19
2 if edad >= 18:
3     print('Usted puede votar.')
4
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL

Usted puede votar.



La importancia de la **indentación** es fundamental para que el código dentro del bloque se pueda ejecutar. Si no se indenta el bloque de código dentro de la sentencia, dará un error y no se podrá continuar con el programa.

correcto

```
1 edad = 19
2 if edad >= 18:
3     print('Usted puede votar.')
4
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL

Usted puede votar.

incorrecto

```
1 edad = 19
2 if edad >= 18:
3     print('Usted puede votar.')
4
```

PROBLEMAS 1 SALIDA CONSOLA DE DEPURACIÓN TERMINAL

IndentationError: expected an indented block after 'if' statement on line 2
PS C:\Users\PC\Desktop\Info2023> & "C:/Program Files/Python311/python.exe"
as_de_control.py
File "c:\Users\PC\Desktop\Info2023\estructuras_de_control.py", line 3
 print('Usted puede votar.')
 ^

Y así podemos observar, si no indentamos correctamente, nos dará un error y el programa dejará de ejecutarse.

if encadenados

Otra forma en la que podemos usar if es encadenándolo. Esta opción es muy específica para ciertos casos, ya que hay otras funciones que ofrece if (que veremos en un momento).

```
1 edad = 17
2
3 if edad >= 18:
4     print('Usted puede votar.')
5 if edad < 18:
6     print('Usted NO puede votar.')
7
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL

Usted NO puede votar.

encadenado

Podemos observar que estamos “forzando” a que se ejecute uno de los bloques, según la condición que estamos dando.

Como el valor de nuestra variable es 17, cuando el programa verifica la condición en el primer if, obtiene un valor False, por lo que no ejecuta el bloque de código que se encuentra adentro y sigue con las siguientes líneas de código.

Se encuentra con otro if y vuelve a verificar la condición. Al obtener un valor True, ejecuta el bloque de código que encuentra dentro de este segundo if.



if anidados

Otra opción para usar if es la de anidar. Esto quiere decir, poner un if dentro de un bloque de código de otro if. Esto sucede cuando necesitamos hacer dos o más comparaciones para brindar mensajes distintos, pero a la vez relacionados, como en el ejemplo que veremos.

```
1  edad1 = 18
2  edad2 = 21
3
4  if edad1 >= 18:
5      print('Usted puede votar.')
6      if edad2 >= 21:
7          print('Usted puede viajar fuera del país.')
```

anidados

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL

Usted puede votar.
Usted puede viajar fuera del país.

Podemos observar que en ambos if obtenemos un valor True, y como tal, ambos mensajes se muestran. Si, en tal caso, en el if que está adentro del bloque de código del primer if (o podríamos decir, del if más a la izquierda según la indentación) obtuviésemos False, el segundo mensaje no se mostraría, y, si en el primer if obtuviésemos False, ninguno de los mensajes se mostraría. ¿Lo vas comprendiendo mejor? Copiá el código en tu editor y comenzá a cambiar los valores de las variables para practicar y entender mejor. Algo importante a destacar, es el uso de la **indentación**, ya que si no estuviese correctamente indentado, no funcionaría este bloque de código. ¡Probalo!

Uso de operadores lógicos con if

En el caso anterior, necesitábamos brindar mensajes distintos, según la condición, pero que estaban relacionados de todas formas. Sin embargo, hay casos en los que necesitamos hacer más de una comparación o necesitamos más de una condición para que se realice una acción. Vamos a tomar uno de los ejemplos más cotidianos que solemos ver.

```
1  usuario = 'Informatario'
2  contrasena = 'Info2023'
3
4  if usuario == 'Informatario' and contrasena == 'Info2023':
5      print('Acceso correcto. Bienvenido')
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL

Acceso correcto. Bienvenido



Como pudiste ver, así funcionaría básicamente un acceso de usuario y contraseña. Usando un operador lógico, en este caso **and**, para verificar dos condiciones a la vez. Esto no se limita a solo dos condiciones, pueden haber más, pero siempre tenés que respetar la estructura que te damos en el ejemplo. También podés usar el operador **or** o el operador **not**, o podés ir combinando todos. Lo importante es pensar en la condición que queremos dar, pensar en lo que queremos que evalúe el programa según nuestra necesidad y en base a eso, armar las condiciones que correspondan. ¡Con práctica lo vas a comprender mejor! Y para que sigas practicando te dejamos la tabla de operadores lógicos y los resultados que se obtienen según las combinaciones.

AND		
VARIABLES		RETORNO
α	b	y
TRUE	TRUE	TRUE
TRUE	FALSE	FALSE
FALSE	TRUE	FALSE
FALSE	FALSE	FALSE

OR		
VARIABLES		RETORNO
α	b	y
FALSE	FALSE	FALSE
TRUE	TRUE	TRUE
TRUE	FALSE	TRUE
FALSE	TRUE	TRUE

NOT	
VARIABLES	RETORNO
α	y
TRUE	FALSE
FALSE	TRUE

if - else - elif

El **condicional alternativo o doble**, es una solución que te va a ser muy útil y ¡vas a usarla muchísimo! Mirá este ejemplo y tratá de entenderlo... luego profundizamos.

```
1 usuario = 'Informatorio'
2 contrasena = 'contraseña_incorrecta'
3
4 if usuario == 'Informatorio' and contrasena == 'Info2023':
5     print('Acceso correcto. Bienvenido')
6 else:
7     print('Ingresaste mal tu usuario o contraseña. Reiniciá el programa y volvé a intentar.')
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL Python + - [] [X]

Ingresaste mal tu usuario o contraseña. Reiniciá el programa y volvé a intentar.



¿Pudiste entender que pasó? Usando la sentencia **else** (que lo traducimos como sino en español), pudimos realizar una acción u otra acción, es decir, sino se ejecutaba un bloque de código, se ejecutaría el otro bloque de código, y esto gracias a **else**.

Podemos ver que la indentación de **else**, está a la izquierda, a la altura de **if**, y el bloque de código, de **else**, también se encuentra a la altura del bloque de código del **if**.

De esta manera, el programa lee la sentencia **if**, y si tiene un retorno **False** ejecuta inmediatamente lo que se encuentra dentro del bloque de código de **else**.

Entonces, podemos decir que cuando utilizamos un condicional doble, uno de los dos bloques de código se va a ejecutar. El bloque de código de **if** (si se retorna un valor **True** de la condición), o el bloque de código de **else** (si se retorna un valor **False** de la condición **if**).

Algo para hacer notar, **else**, no lleva condicional, ya que la condición se realiza en el primer parte, en el **if**.

¿Y qué pasa si necesitamos condiciones múltiples (elif)?

Te vamos a mostrar tres ejemplos y volvemos a profundizar.

Ejemplo 1: se ingresan tres notas, se saca el promedio y se da un mensaje al usuario (para este caso Desaprobado).

```
1  nota1 = 1
2  nota2 = 3
3  nota3 = 5
4  promedio_notas = (nota1 + nota2 + nota3) / 3
5
6  if promedio_notas < 6:
7      print(f'Por lo visto no estás leyendo los apuntes ni practicando. Desaprobaste. Tu promedio es: {promedio_notas}')
8  elif promedio_notas < 8:
9      print(f'¡Aprobaste! ¡Pero con más práctica vas a lograr más! Tu promedio es: {promedio_notas}')
10 else:
11     print(f'¡Excelente! Se nota que estás leyendo los apuntes y practicando. Tu promedio es: {promedio_notas}')
12
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN **TERMINAL** Python + v [icon] [icon]

Por lo visto no estás leyendo los apuntes ni practicando. Desaprobaste. Tu promedio es: 3.0

En este ejemplo podemos observar que el promedio de notas es de 3.0, por lo que según las condiciones que tenemos, el bloque de código que se ejecuta es el primero ya que el rango que tenemos como condición es entre 0 y 5. Por lo que el programa verifica el primer condicional y, como retorna un valor **True** (ya que se cumple la condición del rango que pedimos) ejecuta el bloque de código que se encuentra adentro. El resto de los bloques no se ejecutarán porque se cumplió la primer condición.



Ejemplo 2: se ingresan tres notas, se saca el promedio y se da un mensaje al usuario (para este caso Aprobado pero...).

```
1  nota1 = 5
2  nota2 = 7
3  nota3 = 9
4  promedio_notas = (nota1 + nota2 + nota3) / 3
5
6  if promedio_notas < 6:
7      print(f'Por lo visto no estás leyendo los apuntes ni practicando. Desaprobaste. Tu promedio es: {promedio_notas}')
8  elif promedio_notas < 8:
9      print(f'¡Aprobaste! ¡Pero con más práctica vas a lograr más! Tu promedio es: {promedio_notas}')
10 else:
11     print(f'¡Excelente! Se nota que estás leyendo los apuntes y practicando. Tu promedio es: {promedio_notas}')
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL Python + - □ □

¡Aprobaste! ¡Pero con más práctica vas a lograr más! Tu promedio es: 7.0

En este segundo ejemplo vemos (que el alumno estudió más 😊) que el promedio de notas es de 7.0, por lo que según las condiciones que tenemos, el bloque de código que se ejecuta es el segundo (donde ya tenemos la sentencia `elif`) ya que el rango que tenemos como condición en este caso es entre 6 y 7.99. Por lo que el programa verifica el primer condicional y retorna un valor `False`, así que pasa a la siguiente condición y esta retorna un valor `True`, así que ejecuta ese bloque de código. Si te estás preguntando si se pueden seguir agregando más `elif`, la respuesta es sí!

Elif que sería la combinación entre `if` y `else` (interpretándolo del inglés sería como “sino si”) es una condición intermedia entre `if` y `else`. Se puede poner la cantidad que necesitemos, aunque siempre es conveniente mirar desde otra perspectiva el problema y, tratar de reducir y simplificar el código de la forma más óptima posible, ya que mientras más código escrito, más complicado es leerlo y con una sintaxis deficiente (por uso excesivo de condicionales muchas veces), es aún peor. Así que sí, se puede usar la cantidad de `elif` que queramos, pero hay que ser precavidos.

Ejemplo 3: se ingresan tres notas, se saca el promedio y se da un mensaje al usuario (para este caso Excelente).

```
1  nota1 = 10
2  nota2 = 10
3  nota3 = 10
4  promedio_notas = (nota1 + nota2 + nota3) / 3
5
6  if promedio_notas < 6:
7      print(f'Por lo visto no estás leyendo los apuntes ni practicando. Desaprobaste. Tu promedio es: {promedio_notas}')
8  elif promedio_notas < 8:
9      print(f'¡Aprobaste! ¡Pero con más práctica vas a lograr más! Tu promedio es: {promedio_notas}')
10 else:
11     print(f'¡Excelente! Se nota que estás leyendo los apuntes y practicando. Tu promedio es: {promedio_notas}')
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL Python + - □ □

¡Excelente! Se nota que estás leyendo los apuntes y practicando. Tu promedio es: 10.0



En este último caso, las dos primeras condiciones retornan un valor False, por lo que “por defecto” va a ingresar al bloque de código de **else** y ejecuta lo que encuentra adentro.

Como podemos ver en estos 3 ejemplos, el programa va a ingresar al condicional **if**, donde si este retorna un valor False, va a seguir al **elif** y si este retorna un valor False, va a ejecutar el bloque de código que se encuentra en **else**, pero si alguna de las anteriores condiciones retorna un valor True, ejecutará el código dentro del bloque donde obtuvo el retorno True y saldrá del condicional **if**.

Ahora que conocés la estructura de control condicional te invitamos a practicar, hacer tus propios programas, e incluso, ya podés hacer un programa que pida usuario y contraseña para dar un mensaje (mejorando el código de ejemplo que te dimos!).

En la siguiente lección vas a aprender sobre

ESTRUCTURAS DE DATOS