



Fundamentos de Programación con Python

Unidad III Control del flujo Sentencias o estructuras Condicionales

Docente: T.S.U Gerardo Alí Ferraro Schelijasch
gerferr83@soltecferr.com
<https://soltecferr.com>

Unidad III – Sentencias condicionales

Resumen

- Definición de flujo de un programa
- Definición de condicional
- Uso y generalidades
- Sintaxis en Python
- If
- If ... else
- If ... elif ... else
- match (switch)
- Condicionales anidados.

Unidad III – Sentencias condicionales

Resumen

- Definición de flujo de un programa.
- Definición de condicional.
- Uso.
- Sintaxis en Python
- If
- If ... else
- If ... elif ... else
- match (switch)
- Condicionales anidados.

Unidad III – Sentencias condicionales

Flujo de un programa:

En programación, el término flujo se refiere a la secuencia y la dirección en que se ejecutan las instrucciones o las operaciones dentro de un programa. El flujo de un programa puede ser controlado por diferentes estructuras y mecanismos que determinan el orden en que se ejecutan las instrucciones. A continuación se explican los principales tipos de flujo en programación:

- **Flujo Secuencial:**

El flujo secuencial es el más simple y se refiere a la ejecución de instrucciones en el orden en que están escritas, de arriba hacia abajo. Cada línea de código se ejecuta una tras otra sin saltos o bifurcaciones.

Unidad II – Sentencias condicionales

- **Flujo Secuencial:**

```
print("Inicio del programa")  
x = 5  
y = x + 10  
print("El valor de y es:", y)  
print("Fin del programa")
```

- **Flujo Condicional:**

El flujo condicional permite que el programa tome decisiones y ejecute diferentes bloques de código dependiendo de ciertas condiciones. Esto se realiza utilizando estructuras de control.

Unidad III – Sentencias condicionales

- **Flujo Condicional:**

```
edad = 20
if edad >= 18:
    print("Eres mayor de edad")
else:
    print("Eres menor de edad")
```

- **Flujo de iteración(Bucles o ciclos):**

El flujo de iteración permite ejecutar un bloque de código repetidamente mientras se cumpla una condición. Se utiliza en estructuras de bucle como for y while.

Unidad III – Sentencias condicionales

- **Flujo de iteración(Bucles o ciclos):**

```
for i in range(5):  
    print(i)
```

```
contador = 0  
while contador < 5:  
    print(contador)  
    contador += 1
```

Unidad III – Sentencias condicionales

Sentencias o estructuras condicionales:

- **Definición:**

Una sentencia condicional es una estructura de control en programación que permite tomar decisiones y ejecutar diferentes bloques de código en función de una o más condiciones. Estas condiciones se evalúan como verdaderas o falsas, y en base a esta evaluación, se determina el flujo de ejecución del programa.

- **Uso:**

Las sentencias condicionales se utilizan para ejecutar ciertos bloques de código solo cuando se cumplen determinadas condiciones. Esto es fundamental para implementar lógica de decisión en los programas.

Unidad III – Sentencias condicionales

Sentencias o estructuras condicionales:

- **Sentencia if.**

Ejecuta un bloque de código si la condición es verdadera.

Un ejemplo sería si tenemos dos valores a y b que queremos dividir. Antes de entrar en el bloque de código que divide a/b, sería importante verificar que b es distinto de cero, ya que la división por cero no está definida. Es aquí donde entran los condicionales if.

```
a = 4
b = 2
if b != 0:
    print(a/b)
```

Unidad III – Sentencias condicionales

Sentencias o estructuras condicionales:

En este ejemplo podemos ver como se puede usar un `if` en Python. Con el operador `!=` se comprueba que el número `b` sea distinto de cero, y si lo es, se ejecuta el código que está indentado. Por lo tanto un `if` tiene dos partes:

La condición que se tiene que cumplir para que el bloque de código se ejecute, en nuestro caso `b!=0`. El bloque de código que se ejecutará si se cumple la condición anterior. Es muy importante tener en cuenta que la sentencia `if` debe ir terminada por `:` y el bloque de código a ejecutar debe estar indentado. Si usas algún editor de código, seguramente la indentación se producirá automáticamente al presionar enter. Nótese que el bloque de código puede también contener más de una línea, es decir puede contener más de una instrucción.

Unidad III – Sentencias condicionales

Sentencias o estructuras condicionales:

Ejemplo: Ecuación de segundo grado

Es necesario crear un programa que permita resolver una ecuación de segundo grado del tipo:

$$x^2+bx+c=0 \text{ --- } ax^2+bx+c=0$$

Determinar x_1 y x_2 como solución de la ecuación.

$$x_1 = ((-1 \pm b) - \sqrt{\Delta}) / (2 \cdot a)$$

$$x_2 = ((-1 \pm b) + \sqrt{\Delta}) / (2 \cdot a)$$

$$\Delta = b^2 - 4 \cdot a \cdot c$$

Unidad III – Sentencias condicionales

Sentencias o estructuras condicionales:

Uso de else y elif

Es posible que no solo queramos hacer algo si una determinada condición se cumple, sino que además queramos hacer algo de lo contrario. Es aquí donde entra la cláusula else. La parte del if se comporta de la manera que ya hemos explicado, con la diferencia que si esa condición no se cumple, se ejecutará el código presente dentro del else. Nótese que ambos bloque de código son excluyentes, se entra o en uno o en otro, pero nunca se ejecutarán los dos.

```
x = 5
if x == 5:
    print("Es 5")
else:
    print("No es 5")
```

Unidad III – Sentencias condicionales

Sentencias o estructuras condicionales:

Uso de else y elif

Hasta ahora hemos visto como ejecutar un bloque de código si se cumple una instrucción, u otro si no se cumple, pero no es suficiente. En muchos casos, podemos tener varias condiciones diferentes y para cada una queremos un código distinto. Es aquí donde entra en juego el elif.

```
x = 5
if x == 5:
    print("Es 5")
elif x == 6:
    print("Es 6")
elif x == 7:
    print("Es 7")
```

Unidad III – Sentencias condicionales

Sentencias o estructuras condicionales:

Uso de else y elif

Con la cláusula elif podemos ejecutar tantos bloques de código distintos como queramos según la condición. Traducido al lenguaje natural, sería algo así como decir: si es igual a 5 haz esto, si es igual a 6 haz lo otro, si es igual a 7 haz lo otro.

Se puede usar también de manera conjunta todo, el if con el elif y un else al final. Es muy importante notar que if y else solamente puede haber uno, mientras que elif puede haber varios.

```
x = 5
if x == 5:
    print("Es 5")
elif x == 6:
    print("Es 6")
elif x == 7:
    print("Es 7")
else:
    print("Es otro")
```


Unidad III – Sentencias condicionales

Sentencias o estructuras condicionales:

Uso del match

En Python, match es una palabra clave introducida en Python 3.10 como parte de la nueva sintaxis de "Structural Pattern Matching" (Coincidencia de Patrones Estructurales). Es una característica avanzada que permite realizar coincidencias complejas en estructuras de datos

Sintaxis

La estructura básica de una declaración match es similar a un switch o case en otros lenguajes de programación, pero con capacidades mucho más avanzadas.

Unidad III – Sentencias condicionales

Sentencias o estructuras condicionales:

Uso del match:

- Cláusula case:

Define un patrón y el bloque de código a ejecutar si el patrón coincide. Puede contener variables para capturar partes del valor de entrada.

- Patrón comodín (_):

Se usa como un caso por defecto que coincide con cualquier valor no capturado por los patrones anteriores.

Unidad III – Sentencias condicionales

Sentencias o estructuras condicionales:

```
.. dia = "martes"
..
.. # Usar match-case para evaluar la variable
.. match dia:
..     case "lunes":
..         print("Es el primer día de la semana.")
..     case "martes":
..         print("Es el segundo día de la semana.")
..     case "miércoles":
..         print("Es el tercer día de la semana.")
..     case "jueves":
..         print("Es el cuarto día de la semana.")
..     case "viernes":
..         print("Es el quinto día de la semana.")
..     case "sábado":
..         print("Es el sexto día de la semana.")
..     case "domingo":
..         print("Es el séptimo día de la semana.")
..     case _:
..         print("No es un día válido.")
```

Unidad III – Sentencias condicionales

Sentencias o estructuras condicionales:

Condicionales anidados:

Las condicionales anidadas en Python se refieren a la práctica de incluir una o más sentencias condicionales (if, elif, else) dentro de otra sentencia condicional. Esto permite verificar múltiples condiciones de manera secuencial y ejecutar diferentes bloques de código según sea necesario.

```
if condición1:
    # Bloque de código si condición1 es verdadera
    if condición2:
        # Bloque de código si condición2 es verdadera
    else:
        # Bloque de código si condición2 es falsa
else:
    # Bloque de código si condición1 es falsa
```

Unidad III – Sentencias condicionales

Sentencias o estructuras condicionales:

Condicionales anidados:

```
nota = 85
proyecto_entregado = True

if nota >= 60:
    if proyecto_entregado:
        if nota >= 90:
            print("Has pasado con una nota excelente.")
        elif nota >= 75:
            print("Has pasado con una buena nota.")
        else:
            print("Has pasado con una nota suficiente.")
    else:
        print("Has pasado el examen, pero no entregaste el proyecto.")
else:
    print("No has pasado el examen.")
```

Unidad III – Sentencias Condicionales

Términos de la licencia.

- This work is licensed under the creative commons Attribution-shareAlike License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA,
- Este trabajo se otorga bajo los términos de la licencia Creative Commons Attribution-shareAlike License. Para obtener una copia de esta licencia visita <http://creativecommons.org/licenses/by-sa/4.0> o envía una carta a la dirección Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.