

APRENDAMOS A PROGRAMAR

Programación Básica.

Clase 04

Python





Contenido

Introducción.....	2
Una condición	2
Estructura de Decisión if	3
Los operadores relacionales	4
Las tablas de verdad	5
Ejercicios	7
Estructura de Decisión if SIMPLE	7
Ejemplo estructura de decisión if SIMPLE.....	8
Práctica estructura de decisión if SIMPLE	8
Estructura de Decisión if DOBLE	9
Ejemplo estructura de decisión if DOBLE.....	9
Práctica condicional doble	9
Estructura de decisión if ANIDADO.....	10
Ejemplo estructura de decisión if ANIDADA	10
Práctica estructura de decisión if ANIDADA.....	11
Estructura de decisión if ANIDADA MÚLTIPLE	11
Práctica estructura de decisión if ANIDADA MÚLTIPLE	12
Lecturas adicionales.....	13



La toma de decisiones es parte de nuestro diario vivir.

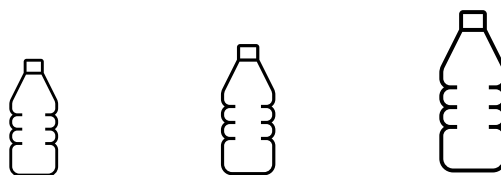
Introducción

Las estructuras de decisión se encuentran en todos los lenguajes de programación y son utilizadas para **dar mayor funcionalidad a los programas**.

Las utilizamos cuando existen varias opciones entre las cuales elegir para realizar una o varias instrucciones y lograr lo que se nos pide.

Por ejemplo: se desea mostrar el precio de un producto. El precio varía según la presentación del producto, de la siguiente manera: la botella de 150ml cuesta 350 colones, la de 600ml cuesta 850 colones y la de 1000ml cuesta 1250 colones.

Para mostrar el precio, necesito solicitar la presentación o tamaño de la botella y según el dato leído asigno el precio a pagar (*luego veremos cómo desarrollar esto a través de un programa en Python*).



En programación, en las estructuras de decisión se utilizan los operadores llamados "operadores relacionales", para crear lo que llamaremos una condición.

Una **condición** es una instrucción que nos permite determinar elegir entre una alternativa u otra.

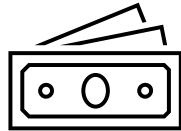




Tomando como **ejemplo** un caso de la vida diaria, podemos decir que:

Si tengo suficiente dinero, puedo comprar una consola de juegos para divertirme, si no debo comprar un entretenimiento de menor costo.

En el ejemplo anterior la condición sería “tener suficiente dinero” y nos permitirá elegir el entretenimiento que debo comprar.



La condición resulta en uno de dos posibles valores:

True (verdadero)

False (falso)

Estructura de Decisión if

La estructura de decisión if, permite tomar una decisión en el flujo del programa, para que siga un camino específico según el resultado de una condición.

En Python se representaría así:

if condición:

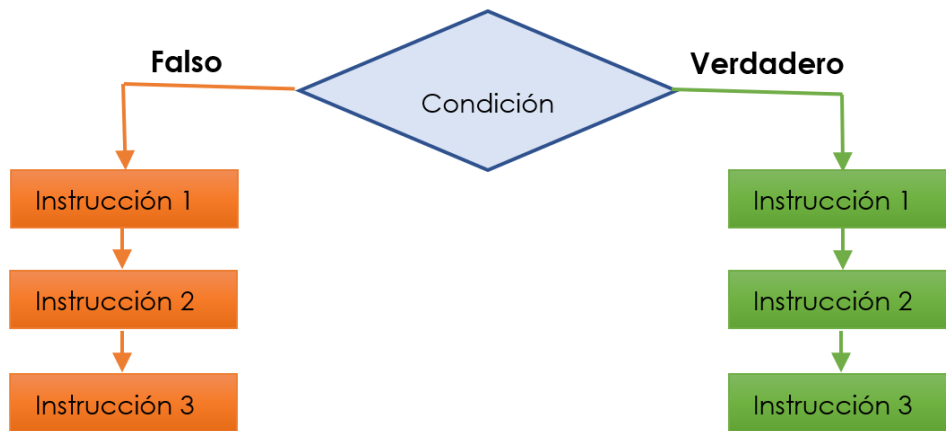
instrucción 1
instrucción 2

else:

instrucción 1
instrucción 2 ...



Para comprenderlo mejor, veámoslo de forma gráfica:



El programa elegirá el camino verdadero o el camino falso y ejecutará sus instrucciones, según sea el resultado de la condición.

En la condición se utilizan los operadores conocidos como “relacionales”.

Los **operadores relacionales** nos permiten establecer una comparación entre dos valores.

Los operadores relacionales son los siguientes:

Operador	Acción	Ejemplo	Significado
>	Mayor que	$A > B$	A es mayor que B
>=	Mayor o igual que	$A >= B$	A es mayor o igual que B
<	Menor que	$A < B$	A es menor que B
<=	Menor o igual que	$A <= B$	A es menor o igual que B
==	Igual	$A == B$	A es igual a B
!=	Diferente	$A != B$	A es diferente a B



Nota: Podemos utilizar en nuestra estructura de decisión uno o más operadores relacionales, considerando que estos deben brindarnos al final un único resultado verdadero o falso. De acuerdo con el lenguaje que se utilice la simbología podría variar, pero el concepto es el mismo.

Operadores Lógicos

En algunas ocasiones las decisiones que debe tomar nuestro algoritmo se basan en más de una comparación lógica, por lo que debemos de unirlos para que, sin importar la cantidad de evaluaciones lógicas, logremos obtener un único valor booleano (verdadero o falso) que nos permita tomar la decisión final. Para eso utilizamos los operadores lógicos, que se detallan a continuación:

Operador	Acción
AND	Requiere que todas las evaluaciones lógicas sean verdaderas para que el resultado final también lo sea
OR	Si al menos una de las evaluaciones lógicas es verdadera, el resultado final también lo será
NOT	Invierte el valor de la evaluación lógica.

Las **tablas de verdad** nos ayudan a comprender la relación y aplicación de los operadores lógicos.

A continuación, se muestra la tabla de verdad para los operadores and, or y not.

Condición	Operador	Condición	Resultado
Verdadero	and	Verdadero	Verdadero
Verdadero		Falso	Falso
Falso		Falso	Falso
Verdadero	or	Verdadero	Verdadero
Verdadero		Falso	Verdadero
Falso		Falso	Falso



Es importante notar que los operadores **and** y **or** nos permiten unir dos condiciones, mientras que el operador **not** tiene como función invertir el resultado de una condición, de tal forma que, si una condición devuelve un resultado verdadero, el operador not la convierte en falsa.

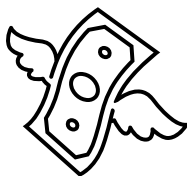
Por ejemplo, tengo la siguiente expresión:

Tengo suficiente dinero para comprar una consola de juegos.

Si agregamos el operador **not** a la expresión anterior:

not (Tengo suficiente dinero, puedo comprar una consola de juegos)

Quiere decir que no tenemos suficiente dinero para comprar la consola.



A continuación, se muestra la tabla de verdad del operador not.

Condición	Operador	Resultado
Verdadero	not	Falso
Falso		Verdadero



Ejercicios

Especifique el resultado de cada expresión, considerando que la variable $A = 0$ y $B = 2$

Operación	Resultado
$(A - B) * -1 > 2$ and $A * -1 == A * 1$	
not $(B ** (B - A) < A + 5)$ or $A < 5$	
not (¡not $(B! = A)$)	
$A * 2 ** 2! = 0$ and $A - B <= 0$	

Estructura de Decisión if SIMPLE

Si la condición que se evalúa en la sentencia if se cumple, se ejecutan ciertas operaciones. Luego se continúa con la secuencia normal del programa.

```
a = 100
if a <= 100:
    print("Hola")
    print("Fuera")
```

Las instrucciones que pertenecen a la estructura de decisión IF, deben ser escritas con una sangría (espacio) respecto a la línea anterior, para indicar que pertenecen a ese bloque de código.



Ejemplo estructura de decisión if SIMPLE

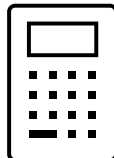
Desarrolle un programa que calcule el total a pagar por la compra de un producto. Debe solicitar al usuario el precio unitario de un producto y la cantidad que desea comprar, **si la compra es de al menos 12 unidades**, aplique un 20% de descuento.

- El texto en verde nos indica que deben aplicarse condicionales.
- Solo se toman acciones cuando la cantidad es al menos 12 unidades.

```
#Se solicita el precio y luego se convierte en un entero
precio = input("Ingrese el precio del producto: ")
precio = int(precio)
#Se solicita la cantidad y se utiliza la función en la misma línea
cantidad = int(input("Ingrese la cantidad: "))
if cantidad >= 12:
    #Si la cantidad es al menos 12
    #se hace un rebajo del 20%
    precio = precio * 0.8
#Se obtiene el valor total de la compra
total = precio * cantidad
#Se muestra el valor al usuario
print("El total a pagar es: ", total)
```

Práctica estructura de decisión if SIMPLE

Desarrolle un programa que muestre el monto a pagar por una compra. Debe tomar en cuenta que si el monto total es superior a \$1000, debe aplicarle un 15% de descuento.





Estructura de Decisión if DOBLE

La estructura de decisión if doble, permite tomar una decisión en el flujo del programa, según el cumplimiento o no de una condición. Si la condición se cumple, toma un camino en el flujo del programa que contiene una o varias operaciones, si la condición no se cumple, tomará otro flujo donde realizarán otras operaciones diferentes. Nunca se realizarán las operaciones de ambos lados.

Ejemplo estructura de decisión if DOBLE

Desarrolle un programa que calcule y muestre el mayor de dos números.

```
num1=int(input("Digite el primer número:"))
num2=int(input("Digite el segundo número:"))
if (num1>num2):
    print("El mayor es:",num1)
else:
    print("El mayor es:",num2)
print("Gracias por utilizar este programa")
```

Práctica condicional doble

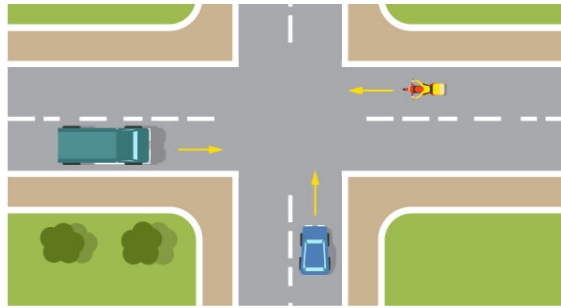
Desarrolle un programa que muestre el precio de la entrada al parque nacional Manuel Antonio, tome en cuenta que un adulto costarricense paga 1600 colones y los niños no pagan.





Estructura de decisión if ANIDADO

En algunas ocasiones requerimos que la toma de decisiones se realice en cascada, es decir, se toma una decisión, se selecciona el camino correspondiente y dentro de este se debe aplicar otro condicional para volver a decidir nuevamente. Este tipo de condiciones se conoce como estructuras de decisión anidadas.



A una condición puede seguirla una instrucción u otra condición.

La estructura de decisión anidada es aquella donde dentro del flujo ya sea del camino verdadero o del camino falso, existe otra condición. Permiten establecer comparaciones, de tal forma que se pueda obtener distintos posibles resultados, ejecutando para cada caso una serie de instrucciones específicas.

Ejemplo estructura de decisión if ANIDADA

```
#Se solicita la nota
nota = int(input("Ingrese su nota: "))
if nota >= 70:
    print("Aprobado")
else:
    if nota >= 60:
        print("Aplazado")
    else:
        print("Reprobado")
print("Nos vemos...!")
```

Salida #1

Salida #2

Salida #3

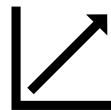


Práctica estructura de decisión if ANIDADA

Una compañía requiere aplicar un aumento de salario a sus empleados, según su categoría, según la tabla que se muestra a continuación.

Categoría	Porcentaje
1	10%
2	12%
3	15%
4	20%

Escriba un programa que calcule el aumento respectivo y muestre el nuevo salario.



Estructura de decisión if ANIDADA MÚLTIPLE

La estructura de decisión anidada múltiple permite la evaluación de múltiples condiciones en forma jerárquica, de tal forma que, si la primera condición no se cumple, se evalúa la siguiente condición y así sucesivamente.

```
#Se solicita la nota
nota = int(input("Ingrese su nota: "))
if nota >= 90:
    print("Aprobado y destacado")
elif nota >= 70:
    print("Aprobado")
elif nota >= 60:
    print("Aplazado")
else:
    print("Reprobado")
print("Nos vemos...!")
```

Salida #1

Salida #2

Salida #3

Salida #4



Práctica estructura de decisión if ANIDADA MÚLTIPLE

1. Un consultorio médico requiere establecer un plan de dieta para sus pacientes. Este plan deben cumplirlo de lunes a jueves en el tiempo del desayuno, según la siguiente tabla:

Día	Dieta
Lunes	Avena y frutas tropicales
Martes	Yogurt y frutas secas
Miércoles	Café y galletas saladas
Jueves	Barra energética y jugo de naranja

Determine la dieta que debe seguir el paciente según el día de la semana.

2. Desarrolle un programa que permita determinar si un jugador puede pasar al siguiente nivel, lo cual se logra completando 5 retos. Si cumplió todos los retos, aparece un mensaje que dice "Siguiendo Nivel". Si un reto no fue cumplido, aparece un mensaje "Realice de nuevo el reto #__" (en el espacio se muestra el número de reto).
3. Una venta de repuestos automotrices está celebrando su aniversario. Por esta razón está ofreciendo descuentos a sus clientes en los productos que ofrece. El descuento se aplica según el tipo de repuesto, de acuerdo con la siguiente tabla:

Tipo	Descuento
Repuestos para el motor	15%
Repuestos accesorios	10%
Repuestos para limpieza del auto	5%
Repuestos para los frenos	3%



Ayuda:

Para la solución se requiere conocer el precio del repuesto y el tipo, por lo cual debe usar la instrucción de lectura para obtener ambos datos. Para leer el tipo es recomendable utilizar opciones, por ejemplo, 1. Repuestos para el motor, 2.



Accesorios, etc. Luego se realiza una operación aritmética para calcular el descuento correspondiente.

Finalmente, muestre el precio a pagar una vez aplicado el descuento.

Lecturas adicionales

eLibro.net

<https://elibro.net/es/ereader/ufidelitas/106404?page=94>

<https://elibro.net/es/ereader/ufidelitas/106404?page=100>

<https://elibro.net/es/ereader/ufidelitas/106404?page=103>