

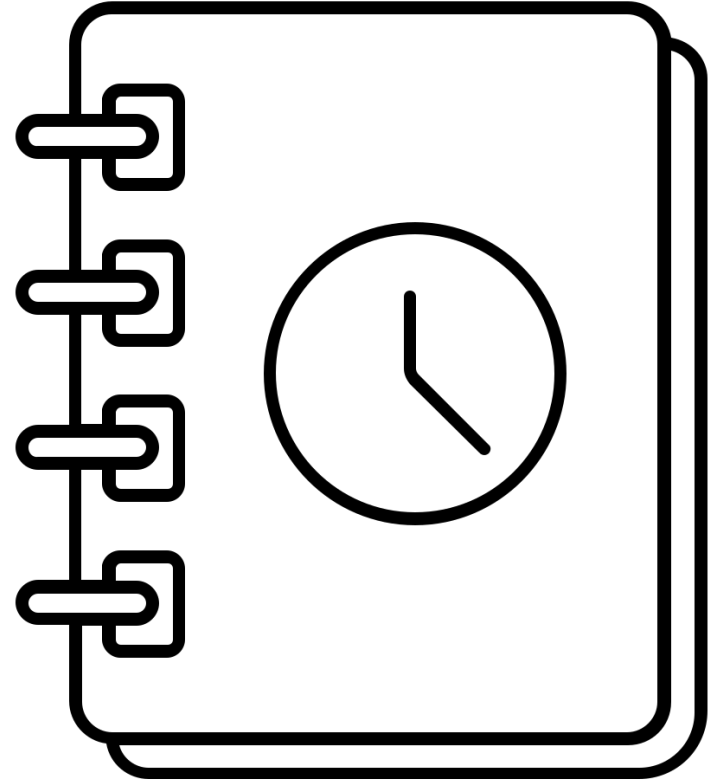
# Programación Básica Clase 04

# Agenda

Estructuras de decisión

Introducción

- Condicional Simple (si – entonces, if – then)
- Condicional doble (si – entonces – sino, if – then – else)





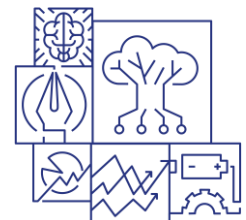
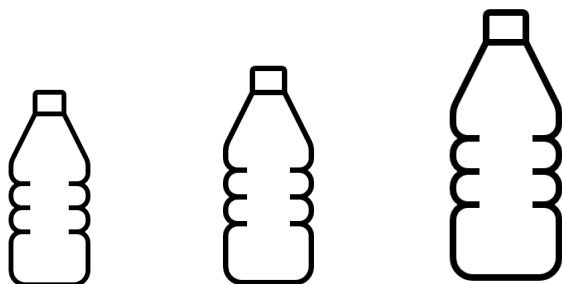
# Introducción a las estructuras de decisión

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).



# ¿Qué es una condición?

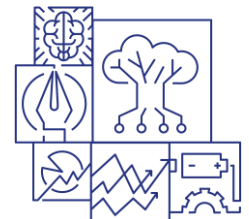


# La condición resulta en uno de dos posibles valores

True  
(verdadero)

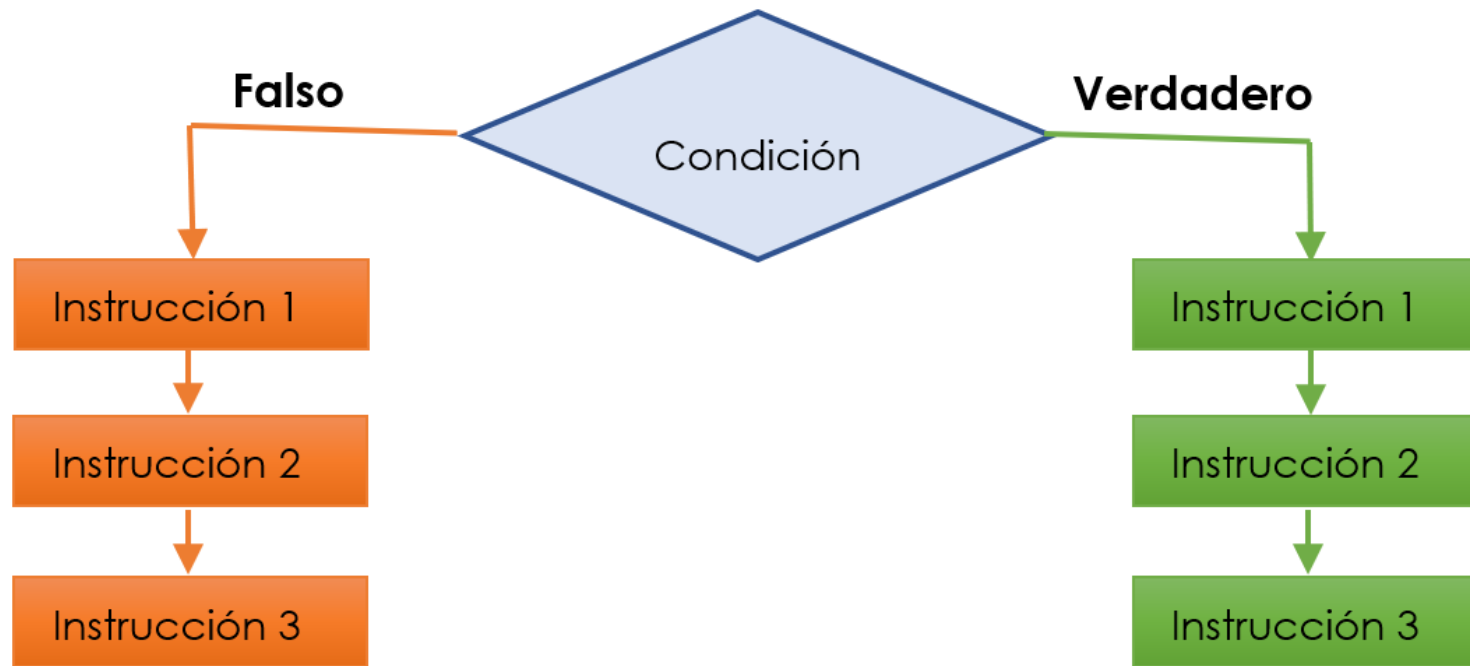


False  
(falso)



# 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.



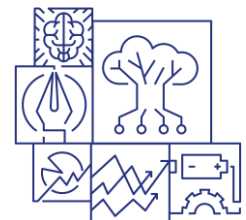
# ¿Qué son operadores relacionales?





Operador	Acción	Ejemplo
>	Mayor que	$A > B$
>=	Mayor o igual que	$A \geq B$
<	Menor que	$A < B$
<=	Menor o igual que	$A \leq B$
==	Igual	$A == B$
!=	Diferente	$A != B$

***Se debe considerar que de acuerdo con la herramienta y al lenguaje que se utilice la simbología podría variar, pero el concepto lógico es el mismo.***



# Ejemplo

Si consideremos las variables

$$A = 0$$

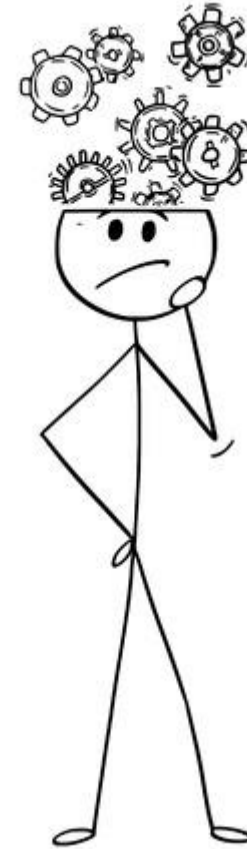
$$B = 2$$

¿Qué resulta de?

$$A > B$$

$$A \neq B$$

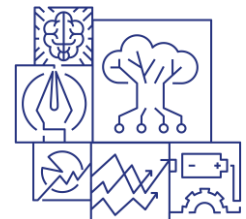
$$B > A$$



# ¿Qué son operadores lógicos?

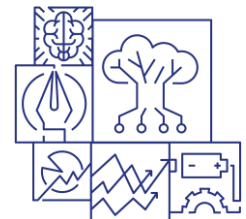


Operador	Acción
AND	Requiere que todas las evaluaciones lógicas sean verdaderas para que el resultado final también lo sea
OR	Si al menos una de las evaluaciones lógicas es verdadera, el resultado final también lo será
NOT	Invierte el valor de la evaluación lógica.



## Ejemplo:

Vamos de paseo para el Volcán Irazú todos en familia en buseta la entrada del parque vale ₡1500 pero si los miembros de la familia mayores de 65 años ● menores de 3 años no paga la entrada.





# Estructuras de decisión SIMPLE

Permite que el flujo del programa siga por un camino específico si se cumple una condición (True) o conjunto de condiciones.

El espacio del tabulador indica que esta instrucción pertenece al bloque de instrucciones previas

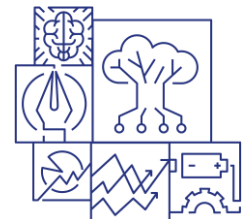
```
a = 100
if a <= 100:
    print("Hola")
    print("Fuera")
```



# Ejemplo de estructura de decisión SIMPLE

Desarrolle un programa que calcule el monto a pagar por una compra, tome en cuenta que las a compras de **12 unidades o más se les** aplica un 20% de descuento.

*El texto en verde nos indica que deben utilizarse estructuras de decisión ya que el descuento depende de la cantidad de unidades compradas.*



```
#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)
```

Un posible resultado

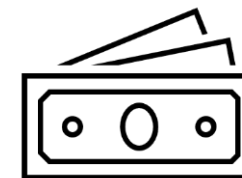
```
Ingrese el precio del producto: 1000
Ingrese la cantidad: 12
El total a pagar es: 9600.0
```



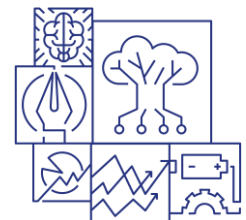
# Práctica de estructura de decisión SIMPLE

Desarrolle un programa que calcule el salario de un colaborador. Si este es inferior a \$1000 aplíquese un 15% de incremento.

Si su salario es igual o superior a \$1000 no recibe incremento



**Tiempo aproximado 15 minutos**



# Estructuras de decisión DOBLE

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.

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





# Ejemplo Estructuras de decisión DOBLE

Desarrolle un programa que muestre si un estudiante aprobó o no un curso.  
**Si la nota es mayor o igual** 70, muestre un mensaje de “Aprobado” **en caso contrario** muestre un mensaje “Reprobado”.

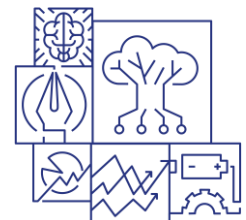
*El texto en verde nos indica que deben utilizarse estructuras de decisión*



```
#Se solicita la nota
nota = int(input("Ingrese su nota: "))
if nota >= 70:
    print("Ha aprobado")
else:
    print("Ha reprobado")
print("Siempre pasa por aquí")
```

Un posible resultado

Ingrese su nota: 85  
Ha aprobado  
Siempre pasa por aquí



# Práctica Estructuras de decisión DOBLE

Desarrolle un programa que calcule el salario de un colaborador si este es superior a \$1000 aplíquese un 15% de incremento de lo contrario aplíquese un 20% de incremento. En ambos casos muestre al usuario su nuevo salario.



Tiempo aproximado 20 minutos



# Estructura de decisión if ANIDADO

En algunos casos una decisión depende de una condición previa y se requiere lo cual causa 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.

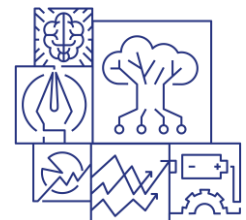


```
#Se solicita la nota
nota = int(input("Ingrese su nota: "))
if nota >= 70:
    print("Aprobado")
else:
    if nota >= 60:
        print("Aplazado")
    else:
        print("Reprobado")
print("Nos vemos...!")
```

Salida 1

Salida 2

Salida 3





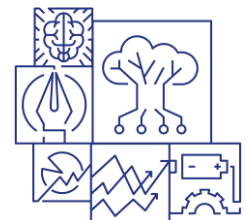
# Práctica Estructura de decisión if ANIDADO

Desarrolle un programa que calcule el salario de un colaborador, según su categoría se le aplica el aumento correspondiente.

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



Tiempo aproximado 25 minutos



# Otra práctica

Desarrolle un programa que muestre si un estudiante aprobó o no un curso. **Si la nota es mayor o igual 70**, muestre un mensaje de “Aprobado” **en caso contrario** muestre un mensaje “Reprobado”.

*El texto en verde nos indica que deben utilizarse estructuras de decisión*

**Nota  $\geq 90$  aprobado destacado**  
**Nota  $\geq 70$  aprobado**  
**Nota  $\geq 60$  aplazado**



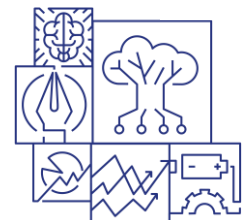
```
#Se solicita la nota
nota = int(input("Ingrese su nota: "))
if nota >= 90:
    print("Aprobado y destacado")
elif nota >= 70:
    print("Aprobado")
elif nota >= 60:
    print("Aplazado")
else:
    print("Reprobado")
print("Nos vemos...!")
```

Salida 1

Salida 2

Salida 3

Salida 4



# Así se vería

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

Estructura de  
decisión simple

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

Estructura de  
decisión doble

```
a = 100
b = 10
if a <= 100:
    if b > a:
        print("Hola")
    else:
        print("¡Qué tal!")
else:
    print("Adiós")
print("Fuera")
```

Estructura de  
decisión anidado



1



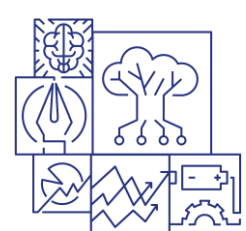
Desarrolle un programa que muestre el mayor de dos números.



Nota suponga que los números son diferentes entre sí.

Modifique el programa para que funcione para 4 números.

**Tiempo aproximado 15 minutos**





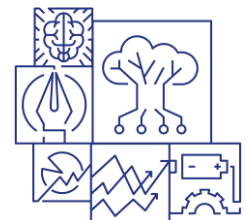
2



Desarrolle un programa que identifique si un numero es par o impar.



Tiempo aproximado 15 minutos



3

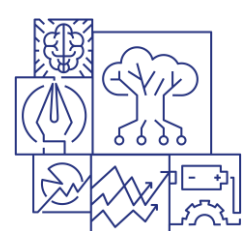


Desarrolle un programa que le indique si su año de nacimiento es en año bisiesto.

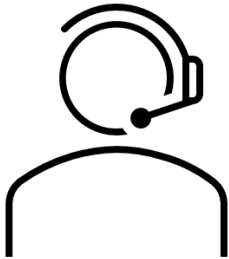


Considere que un año bisiesto es aquel que es divisible entre 4 pero que no es divisible entre 100, excepto el que es divisible entre 400.

**Tiempo aproximado 20 minutos**



Los condicionales nos permiten crear programas que se comporten de formas diferentes de acuerdo a diferentes circunstancias.

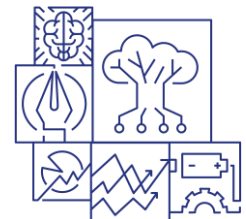


Lecturas adicionales:

<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>





**¡Nos vemos la próxima semana!**

