CONDICIONALES

If, if/else, if/elif/else

Los programas que hemos aprendido a construir hasta el momento presentan siempre una misma secuencia de acciones:

1. **Se piden datos al usuario (asignando a variables valores obtenidos con *input*).**
2. **Se efectúan cálculos con los datos introducidos por el usuario, guardando el resultado en variables (mediante asignaciones).**
3. **Se muestran por pantalla los resultados almacenados en variables (mediante la sentencia print).**

Estos programas se forman como una serie de líneas que se ejecutan una tras otra, desde la primera hasta la ultima y siguiendo el mismo orden con el que aparecen en el archivo: *el flujo de ejecución del programa es estrictamente secuencial*.

No obstante, es posible alterar el flujo de ejecución de los programas para hacer que:

a) **tomen decisiones a partir de los datos y/o resultados intermedios y, en función de estas, se ejecuten ciertas sentencias y otras no;**
b) **tomen decisiones a partir de los datos y/o resultados intermedios y, en función de ´estas, ejecuten ciertas sentencias mas de una vez.**

El primer tipo de alteración del flujo de control se efectúa con sentencias condicionales o de selección *y el segundo tipo con sentencias iterativas o de repetición.*

Las sentencias que permiten alterar el flujo de ejecución se engloban en las denominadas *estructuras de control de flujo (que abreviamos con el termino <<estructuras de control>>).*

Estructuras de Control:

Son estructuras que afectan el flujo normal de un programa, estas se dividen en Selectivas o condicionales y Repetitivas (Ciclos).

<<Estructuras Selectivas>>

Para poder escribir programas útiles, casi siempre vamos a necesitar la capacidad de comprobar condiciones o expresiones lógicas o booleanas y cambiar el flujo del programa de acuerdo a ellas. Las estructuras selectivas o condicionales nos proporcionan esa capacidad.

De forma mas precisa las estructuras selectivas o condicionales: permiten determinar un sentido de acción en el flujo del programa sobre la base de la evaluación de una determinada condición, *se usan para tomar decisiones lógicas*, en ellas se evalúa una expresión lógica y en función del resultado de la misma se ejecutan o no una secuencia de pasos en el programa.

Una expresión es un trozo de código Python que produce o calcula un valor (resultado).

***Una expresión booleana o expresión lógica* es una expresión que produce o bien True o bien False.**

Python provee las llamadas *expresiones relacionales o de comparación* que sirven para *comparar valores entre sí*. Estas expresiones involucran el uso de los llamados operadores relacionales que son:

Operador	*Comparación*
==	es igual que
!=	es diferente de
<	es menor que
<=	es menor o igual que
>	es mayor que
>=	es mayor o igual que

Para ilustrar lo anterior veamos la siguiente tabla

Expresión	Significado
a == b	a es igual a b
a != b	a es distinto de b
a < b	a es menor que b
a <= b	a es menor o igual que b
a > b	a es mayor que b
a >= b	a es mayor o igual que b

Si la expresión se cumple el resultado será True de no ser así False

Operadores lógicos

De la misma manera que se puede operar entre números mediante las operaciones de suma, resta, etc., también existen tres operadores lógicos para combinar expresiones relacionales o de comparación: and (y), or (o) y not (no).

Siendo el significado de estos operadores el siguiente, sean a y b dos expresiones relacionales:

Expresión	Significado
`a` **`and`** `b`	El resultado es `True` solamente si `a` es `True` y `b` es `True` de lo contrario el resultado es `False`
`a` **`or`** `b`	El resultado es `True` si `a` es `True` o `b` es `True` de lo contrario el resultado es `False`
`not` `a`	El resultado es `True` si `a` es `False` de lo contrario el resultado es `False`

Orden de precedencia de los operadores.

La precedencia de los operadores queda establecida en la siguiente tabla:

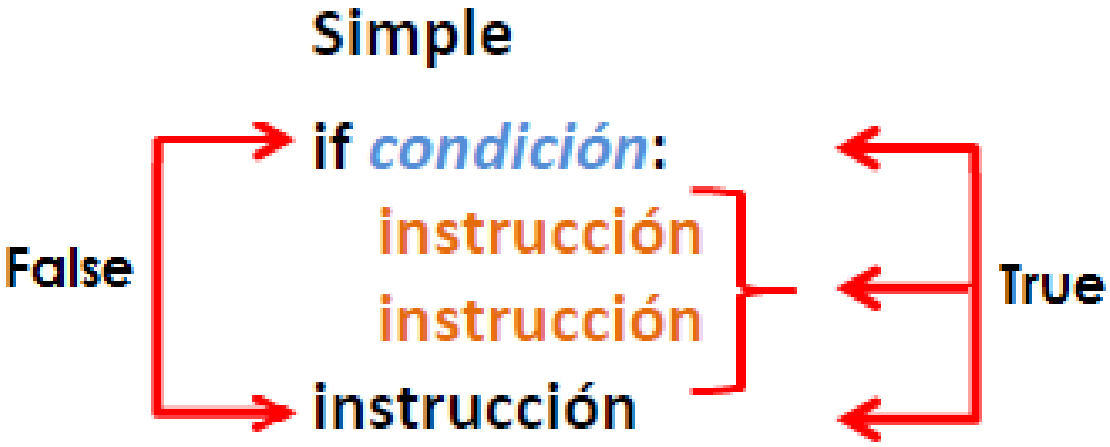
Prioridad	Operador
1	**
2	-
3	*, /, //, %
4	+, -
5	<, >, <=, >=
6	==, !=
7	=, %=, //=, /=, *=, -=, +=, **=
8	and, or, not

Estructura selectiva IF

El Python provee tres tipos de estructuras selectivas:

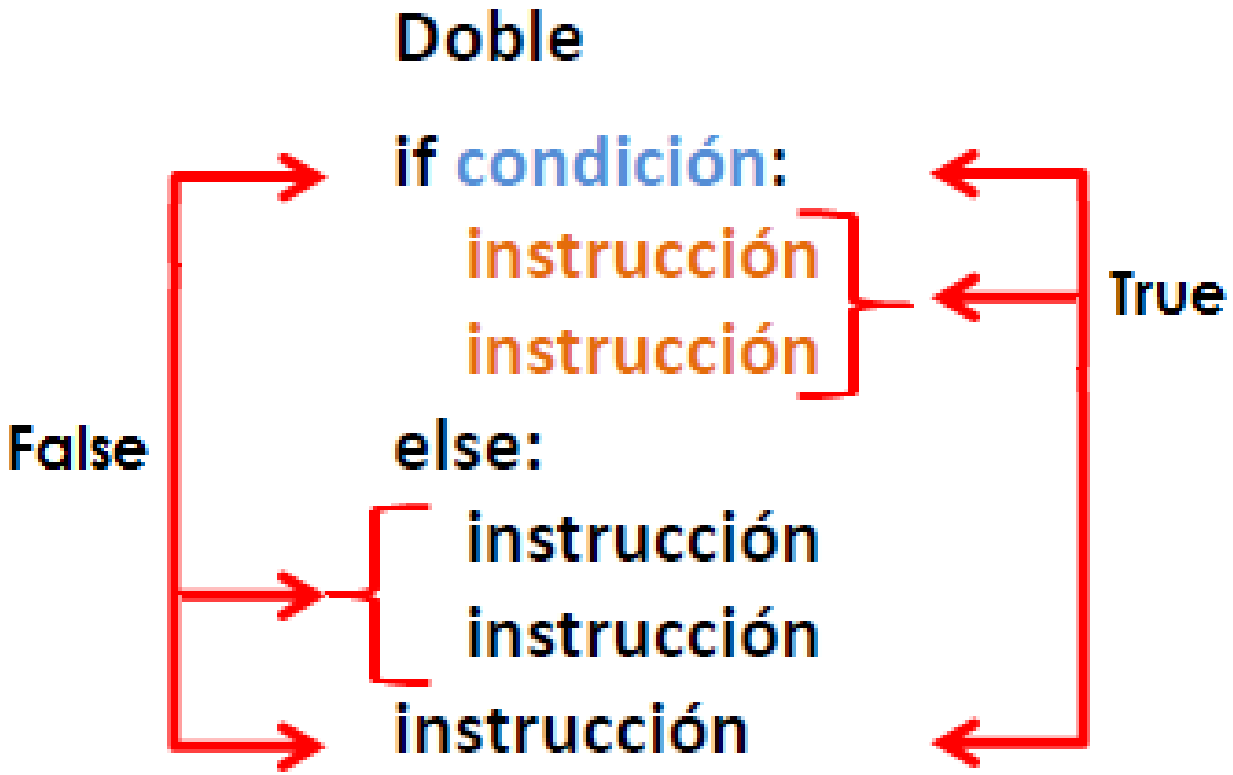
1. **If (la simple).**
2. **If/else (la compuesta)**
3. **If/elif/else (la multiple)**

1. **La primera ejecuta (selecciona) una acción si una condición es TRUE o salta (ignora) la acción si la condición es falsa. Esta es una estructura de selección simple porque selecciona o ignora una simple acción.**
2. **La segunda ejecuta una acción si la condición es TRUE o ejecuta una acción diferente si es FALSE. Esta es una estructura de selección doble o compuesta porque seleccione entre dos diferentes acciones.**
3. **La tercera ejecuta una de diferentes acciones dependiendo de la veracidad o falsedad de varias condiciones. Esta es una estructura de selección múltiple porque selecciona la acción a ejecutar de diferentes acciones.**



Identación o Sangría:

Para hablar de estructuras de control en Python debemos hablar de Identación, que significa hacer espacios en blanco hacia la derecha para mover una línea de código, en Python se aplica la identación para indicar que las instrucciones identadas forman un *bloque de código asociado a una misma estructura de control. Al escribir dos puntos (:) al final de una línea, el IDE sangrará automáticamente las líneas siguientes. Para terminar un bloque, basta con volver al principio de la línea.*



Ejecuta una acción si la condición es TRUE o ejecuta una acción diferente si es FALSE. Esta es una estructura de selección doble o compuesta porque seleccione entre dos diferentes acciones.

Múltiple

```
if condición_1:
    bloque_1
elif condición_2:
    bloque_2
elif condición_3:
    bloque_3
    .
    .
    .
else:
    bloque_n
instrucción
```

Ejecuta una de diferentes acciones dependiendo de la veracidad o falsedad de varias condiciones. Esta es una estructura de selección múltiple porque selecciona la acción a ejecutar de diferentes acciones.

Un persona desea saber cuánto dinero se genera por concepto de intereses sobre la cantidad que tiene en inversión en el banco. El decidirá reinvertir los intereses siempre y cuando estos excedan a Bs. 70.000, y en ese caso desea saber cuánto dinero tendrá finalmente en su cuenta.

```
#Inicializar variables
capital = interes = 0.0
tasa_interes
#Entrada de datos
capital float(input("de el capiltal disponible en el banco. "))
tasa_interes = float(input("Indique la tasa de interes anual: "))
interes = capital*tasa_interes/1200
if interes > 70000:
    capital += interes
print("Capital: ",capital)
```

Crear un programa en Python que lea dos números enteros y determine cual de ellos es el mayor.

```
#Inicializar variables
mayor = n1 = n2 = 0
# Entrada de datos
n1 = int(input("De un numero entero: "))
n2 = int(input("De otro  numero entero: "))
if n1 > n2:
    mayor = n1
else:
    mayor = n2
print("El mayor es: ",mayor)
```

Calcular la utilidad que un trabajador recibe en el reparto anual de utilidades si esta se calcula como un porcentaje de su salario mensual que depende de su antigüedad en la empresa de acuerdo con la siguiente tabla:

Tiempo	Utilidad
Menos de 1 año	5 % del salario
1 año o más y menos de 2 años	7% del salario
2 años o más y menos de 5 años	10% del salario
5 años o más y menos de 10 años	15% del salario
10 años o más	20% del salario

```
# Inicializar variables
salario_Mes = 0.0
antiguedad = 0
utilidad = 0.0
# Entrada de datos
salario_Mes = float(input("De el salario mensual: "))
antiguedad= int(input("De la antiguedad de la persona: "))
#proceso
if antiguedad < 1:
    utilidad = 0.05* salario_Mes
elif antiguedad >= 1 and antiguedad <2:
    utilidad = 0.07*salario_Mes
elif antiguedad >= 2 and antiguedad <5:
    utilidad = 0.1* salario_Mes
elif antiguedad >=5 and antiguedad < 10:
    utilidad = 0.15*salario_Mes
else:
    utilidad= 0.2*salario_Mes
print("Utilidad: ", utilidad)
```

Otro forma:

```
# Inicializar variables
salario_Mes = 0.0
antiguedad = 0
utilidad = 0.0
# Entrada de datos
salario_Mes = float(input("De el salario mensual: "))
antiguedad = int(input("De la antiguedad de la persona: "))
#proceso
if antiguedad < 1:
    utilidad = 0.05*salario_Mes
elif antiguedad <2:
    utilidad = 0.07*salario_Mes
elif antiguedad <5:
    utilidad = 0.1* salario_Mes
elif antiguedad < 10:
    utilidad = 0.15*salario_Mes
else:
    utilidad = 0.2*salario_Mes
print("Utilidad: ", utilidad)
```

Estructuras if/else anidadas

Son estructuras condicionales if/else incluidas en otras estructuras condicionales.

Consideremos el problema de leer tres enteros e imprimirlos en forma ascendente y descendente

```
#Inicializar variables
n1=n2=n3=0 # Guarda los tres numeros
# Guarda la posicion  primero, segundo y tercero
p = s = t = 0
# Entrada de datos
n1 = int(input("De N1: "))
n2 = int(input("De N2: "))
n3 = int(input("De N3: "))
# Proceso
if n1 >n2 and n1>n3:
    p = n1
    if n2 > n3:
        s = n2
        t = n3
    else:
        s = n3
        t = n2
else:
    if n2 > n3:
        p = n2
        if n1 > n3:
            s = n1
            t = n3
        else:
            s = n3
            t = n1
    else:
        p = n3
        if n1 > n2:
            s = n1
            t = n2
        else:
            s = n2
            t = n1
print("Ascedente: ",p," ",s," ",t)
print("Descendente: ",t," ",s," ",p)
```