

Estructuras de control - Alternativa simple

- Al ejecutarse la instrucción **if** se evalúa la condición lógica.
 - Si la condición lógica es **True** se ejecutan de manera secuencial el bloque de instrucciones .
 - Si la condición es **False** no se ejecuta el bloque de instrucciones.
 - Una vez ejecutado el **if** (opción verdadera o falsa) se continúa la ejecución de forma secuencial por la siguiente instrucción (bloque de instrucción no indentado).

```
edad = int(input("Dime tu edad:"))  
if edad >= 18:  
    print("Eres mayor de edad")  
print("Programa terminado")
```

Estructuras de control - Alternativa doble

- Al ejecutarse la instrucción **if** se evalúa la condición lógica.
 - Si la condición lógica es **True** se ejecutan de manera secuencial el primer bloque de instrucciones.
 - Si la condición es **False** se ejecuta el segundo bloque de instrucción.
 - Una vez ejecutado el **if** (opción verdadera o falsa) se continúa la ejecución de forma secuencial por la siguiente instrucción (bloque de instrucción no indentado).

```
edad = int(input("Dime tu edad:"))
if edad >= 18:
    print("Eres mayor de edad")
else:
    print("Eres menor de edad")
print("Programa terminado")
```

Estructuras de control - Alternativa múltiple

- En este caso tenemos varias opciones. Vamos preguntando por cada una de las opciones y según el valor de la expresión ejecutamos un bloque o otro.

```
nota = int(input("Dime tu nota:"))
if nota >=1 and nota <= 4:
    print("Suspenso")
elif nota == 5:
    print("Suficiente")
elif nota == 6 or nota == 7:
    print("Bien")
elif nota == 8:
    print("Notable")
elif nota ==9 or nota == 10:
    print("Sobresaliente")
else:
    print("Nota incorrecta")
print("Programa terminado")
```

Estructura repetitiva - While

- La instrucción **while** ejecuta una secuencia de instrucciones mientras una condición sea verdadera.

```
while <condición>:  
    <instrucciones>
```

- Al ejecutarse esta instrucción, la condición es evaluada. Si la condición resulta verdadera, se ejecuta una vez la secuencia de instrucciones que forman el cuerpo del ciclo. Al finalizar la ejecución del cuerpo del ciclo se vuelve a evaluar la condición y, si es verdadera, la ejecución se repite. Estos pasos se repiten mientras la condición sea verdadera.
- Se puede dar la circunstancia que las instrucciones del bucle no se ejecuten nunca, si al evaluar por primera vez la condición resulta ser falsa.
- Si la condición siempre es verdadera, al ejecutar esta instrucción se produce un ciclo infinito. A fin de evitarlo, las instrucciones del cuerpo del ciclo deben contener alguna instrucción que modifique la o las variables involucradas en la condición, de modo que ésta sea falsificada en algún momento y así finalice la ejecución del ciclo.

ejemplo

- Crea un programa que pida al usuario una contraseña, de forma repetitiva mientras que no introduzca “asdasd”. Cuando finalmente escriba la contraseña correcta, se le dirá “Bienvenido” y terminará el programa.

```
secreto = "asdasd"
clave = input("Dime la clave:")
while clave != secreto:
    print("Clave incorrecta!!!")
    clave = input("Dime la clave:")
print("Bienvenido!!!")
print("Programa terminado")
```

Instrucción *break*

- Termina la ejecución del bucle, además no ejecuta el bloque de instrucciones.
- Ejemplo

```
secreto = "asdasd"
clave = input("Dime la clave:")
while clave != secreto:
    print("Clave incorrecta!!!")
    otra = input("¿Quieres introducir otra clave (S/N)?:")
    if otra.upper()=="N":
        break;
    clave = input("Dime la clave:")
if clave == secreto:
    print("Bienvenido!!!")
print("Programa terminado")
```

Instrucción *continue*

- Deja de ejecutar las restantes instrucciones del bucle y vuelve a iterar.
- Ejemplo: usamos la instrucción *continue* para mostrar los números pares del 1 al 10:

```
cont = 0
while cont < 10:
    cont = cont + 1
    if cont % 2 != 0:
        continue
    print(cont)
```

Instrucción *repetir – hasta que*

- En Python podemos simular el comportamiento de la estructura *repetir-hasta que* utilizando un bucle *while* con la instrucción *break*:

```
secreto = "asdasd"
while True:
    clave = input("Dime la clave:")
    if clave != secreto:
        print("Clave incorrecta!!!")
    if clave == secreto:
        break;
print("Bienvenido!!!")
print("Programa terminado")
```


Estructura repetitiva - for

- La estructura *for* nos permite recorrer los elementos de una secuencia (lista, tupla, cadena de caracteres,...).
- Vamos a usar *for* de forma similar a la instrucción *Para* de pseudocódigo, es decir, ejecutar una secuencia de instrucciones un número determinado de veces, desde un valor inicial, hasta un valor final y con un posible incremento.
 - Para ello vamos a usar el tipo de datos *range* que nos permite generar listas de números. Vamos a usar *for* para crear bucles de instrucciones donde sabemos a priori el número de iteraciones que hay que realizar.

ejemplos

- Escribir en pantalla del 1 al 10

```
for var in range(1,11):  
    print(var, " ",end="")
```

- Escribir en pantalla del 10 al 1

```
for var in range(10,0,-1):  
    print(var, " ",end="")
```

- Escribir los número pares desde el 2 al 10

```
for var in range(2,11,2):  
    print(var, " ",end="")
```

Variables- Contador

- Un **contador** es una variable entera que la utilizamos para contar cuando ocurre un suceso.

- Se inicializa a un valor inicial.

`cont = 0;`

- Se incrementa, cuando ocurre el suceso que estamos contando se le suma 1.

`cont = cont + 1;`

- Otra forma de incrementar el contador:

`cont += 1`

- Ejemplo. Introducir 5 números y contar los pares

```
cont = 0;
for var in range(1,6):
    num = int(input("Dime un número:"))
    if num % 2 == 0:
        cont = cont + 1
print("Has introducido ",cont," números pares.")
```

Variables- Acumulador

- Un **acumulador** es una variable numérica que permite ir acumulando operaciones. Permite ir haciendo operaciones parciales.
- Se inicializa a un valor inicial según la operación que se va a acumular:
 - a 0 si es una suma
 - a 1 si es un producto.
- Se acumula un valor intermedio.
`acum = acum + num;`

- Ejemplo. Introducir 5 números y sumar los pares

```
suma = 0;
for var in range(1,6):
    num = int(input("Dime un número:"))
    if num % 2 == 0:
        suma = suma + num
print("La suma de los números pares es ",suma)
```

Variables- Indicador

- Un **indicador** es una variable lógica, que usamos para recordar o indicar algún suceso.
- Se inicializa a un valor lógico que indica que el suceso no ha ocurrido.
 indicador = False
- Cuando ocurre el suceso que queremos recordar cambiamos su valor.
 indicador = True

- Ejemplo. Introducir 5 números e indicar si se ha introducido algún número par

```
indicador = False;
for var in range(1,6):
    num = int(input("Dime un número:"))
    if num % 2 == 0:
        indicador = True
if indicador:
    print("Has introducido algún número par")
else:
    print("No has introducido algún número par")
```