

Programas que involucran estructuras de decisión.

Loops

Son utilizados en casos Cuando tu necesitas repetir un set de instrucciones una y otra vez, hasta que una cierta condición es cumplida. En Python existen dos tipo de loops:

- For Loop
- While Loop

Ejemplo 1

Escribe un programa que imprima del número 1 al 10.

Variable temporal.

Palabra reservada **in** se utiliza para decir el intervalo de iteración.

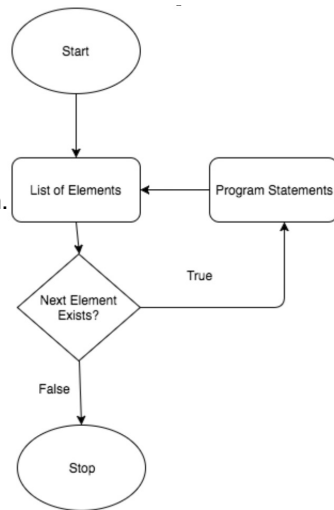
La palabra reservada **for** se utiliza para declarar un loop.

Genera una **secuencia de números** de un intervalo definido por el usuario.

Este **espacio vacío**, se denomina indentation. Esto indica que las instrucciones pertenecen a este bloque de código

Los **dos puntos** le indica a Python que las siguientes instrucciones pertenecen a esta función.

```
for i in range(1,10):  
    print(i, end=' ')
```



```
for i in range(1,11):  
    print(i)
```

Ejercicio 1

Escribir un programa que pida al usuario una palabra y la muestre por pantalla 10 veces.

```
word = raw_input("Escribe una palabra: ")  
  
for variable in range(1,11):  
    print(str(variable) + " " + word)
```

```
word = raw_input("Escribe una palabra: ")  
word1 = raw_input("Escribe una segunda palabra: ")  
  
for variable in range(1,11):  
    print(str(variable) + " " + word)  
  
for variable in range(1,6):  
    print(str(variable) + " " + word1)
```

Ejercicio 2

Escribir un programa que pregunte al usuario su edad y muestre por pantalla todos los años que ha cumplido (desde 1 hasta su edad).

```
edad = int(raw_input("Escriba su edad: "))  
  
print("La cuenta de su edad seria: ")  
for x in range(1, edad + 1):  
    print(str(x) + " años")
```

Ejercicio 3

Escribir un programa que pida al usuario un número entero positivo y muestre por pantalla todos los números impares desde 1 hasta ese número separados por comas.

```
numero = int(raw_input("Escriba un número entero positivo: "))

out = ""
for y in range(1, numero + 1, 2):
    out += str(y) + ", " # out = out + str(y) + ", "

print(out)
```

Ejercicio 4

Escribir un programa que pida al usuario un número entero positivo y muestre por pantalla la cuenta atrás desde ese número hasta cero separados por comas.

```
numero = int(raw_input("Escriba un número entero positivo: "))

out = ""
for y in range(numero, -1, -1):
    out += str(y) + ", " # out = out + str(y) + ", "

print(out)
```

Ejercicio 5

Escribir un programa que pregunte al usuario una cantidad a invertir, el interés anual y el número de años, y muestre por pantalla el capital obtenido en la inversión cada año que dura la inversión.

```
cantidad = float(raw_input("Ingrese la cantidad a invertir: "))
interes = float(raw_input("Ingrese el interés anual: "))
anos = int(raw_input("Cuántos años: "))

for w in range(1, anos+1):
    cantidad *= 1 + interes / 100
    #cantidad = cantidad * 1 + interes / 100
    print("Capital tras %s años: %s" % (str(w), str(round(cantidad, 2))))
```

Ejercicio 6

Escribir un programa que pida al usuario un número entero y muestre por pantalla un triángulo rectángulo como el de más abajo, de altura el número introducido.

```
*
**
***
****
*****
```

```
altura = int(raw_input("Introduce la altura del triángulo: "))

# for h in range(1, altura + 1):
#     print("*" * h)

out=""
for j in range(1, altura+1):
    out += "*"
    print(out)
```

Ejercicio 7

Escribir un programa que muestre por pantalla la tabla de multiplicar del 1 al 10.

```
for i in range(1, 11):
    out = ""
    for j in range(1, 11):
        out += str(i*j) + " "
    print(out)
```

```
for i in range(1, 11):
    out = ""
    for j in range(1, 11):
        out += str(i*j) + "\t"
    print(out)
```

Ejercicio 8

Escribir un programa que pida al usuario un número entero y muestre por pantalla un triángulo rectángulo como el de más abajo.

```
1
3 1
5 3 1
7 5 3 1
9 7 5 3 1
```

```
numero = int(raw_input("Escriba un numero entero positivo: "))

for x in range(1, numero + 1, 2):
    out = ""
    for y in range(x, 0, -2):
        out += str(y) + " " # out = out + str(y) + " "
    print(out)
```

Ejercicio 9

Escribir un programa que almacene la cadena de caracteres contraseña en una variable, pregunte al usuario por la contraseña hasta que introduzca la contraseña correcta.

```
pwd = "ClaseG102.1"

pwd_usr = ""

💡
while pwd != pwd_usr: # == igual, != desigual
    pwd_usr = raw_input("Introduzca su contraseña: ")

print("Contraseña Correcta")
```

Ejercicio 10

Escribir un programa que pida al usuario un número entero y muestre por pantalla si es un número primo o no.

```
num = int(raw_input("Introduce un numero positivo mayor que 2: "))

i = 2
while num % i != 0:
    i += 1

if i == num:
    print("Es primo.")
else:
    print("No es primo.")
```

```
num = int(raw_input("Introduce un numero positivo mayor que 2: "))

for i in range(2, num):
    if num % i == 0:
        break

if (i+1) == num:
    print("Es primo.")
else:
    print("No es primo.")
```

Ejercicio 11

Escribir un programa que pida al usuario una palabra y luego muestre por pantalla una a una las letras de la palabra introducida empezando por la última.

```
>>> word = "Hola"
>>> len(word)
4
>>> range(len(word)-1, -1, -1)
[3, 2, 1, 0]
```

```
word = raw_input("Introduzca una palabra: ")

# [ 0, 1, 2, 3]
# ["Hola"]

for i in range(len(word)-1, -1, -1):
    print(word[i])
```

Ejercicio 12

Escribir un programa en el que se pregunte al usuario por una frase y una letra, y muestre por pantalla el número de veces que aparece la letra en la frase.

```
frase = raw_input("Introduce una frase: ").lower()
letra = raw_input("Introduce una letra: ").lower()

contador = 0

for x in frase:
    if x == letra:
        contador += 1 # contador = contador + 1

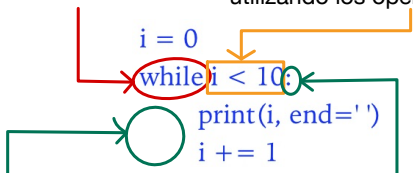
if contador == 0:
    print("Error: \n \t La frase contiene 0 apariciones")
    # \n -> Salto de línea
    # \t -> Tab.
else:
    print("La letra \"%s\" aparece %i veces en la frase" %(letra, contador))
    # print("La letra '%s' aparece %i veces en la frase" %(letra, contador))
    # %s -> string
    # %i -> int
```

Ejercicio 13

Escribir un programa que muestre el eco de todo lo que el usuario introduzca hasta que el usuario escriba “salir” que terminará.

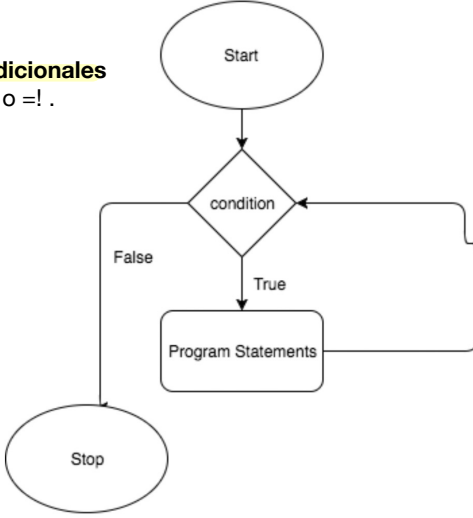
La palabra reservada while se utiliza para declarar un loop.

Es posible escribir diferentes **condicionales** utilizando los operadores <, >, == o != .



Este **espacio vacío**, se denomina indentation. Esto indica que las instrucciones pertenecen a este bloque de código

Los **dos puntos** le indica a Python que las siguientes instrucciones pertenecen a esta función.



```
while True:
    frase = raw_input("Introduce algo: ")
    if frase.lower() == "salir":
        break
    print(frase)
```