

Ejemplo de bucles simples:

Secuencia de numeros uno debajo de otro

In []:

```
cont = 0
while cont < 10:
    print(cont)
    cont = cont+1
```

Observe como funciona: Todo bucle con while tiene 3 elementos a considerar: a) el contador que empieza la secuencia, la condición como parte del while y la sentencia que termina la condición.

Ahora veamos el mismo bucle pero con una secuencia de numeros en una misma linea:

In []:

```
cont=0
while cont < 10:
    print(cont, end= ' ')
    cont= cont+1
```

Observe como funciona: Todo bucle con while tiene 3 elementos a considerar: a) el contador que empieza la secuencia, la condición como parte del while y la sentencia que termina la condición.

Ahora veamos el mismo bucle pero con una secuencia de numeros en una misma linea:

In []:

```
cont=0
while cont < 10:
    print(cont, end= ' ')
    cont= cont+1
```

Ahora prueba generando distintas secuencias como por ejemplo: a) La secuencia de numeros pares desde 2 a 10 b) La secuencia de numeros impares desde 1 hasta un número impar menor o igual al indicado por el usuario.

In []:

```
#solución a)
cont = 2
while cont <= 10:
    print(cont, end=' ')
    cont = cont+2
```

Observa que la sentencia `cont = cont + 2` genera el incremento de la variable contador y a la vez sirve para que se imprima el valor segun el pedido de la secuencia.

Ahora veamos un ejemplo para calcular valores acumulados es decir realizar sumas o productos.

a) Calcular la suma de una serie de numeros desde 1 hasta N. El valor de N lo indica el usuario

In []:

```
n = int(input('ingrese el valor de n:'))
cont = 1
suma = 0
while cont <= n:
    suma = suma + cont
    cont = cont+1
print(suma)
```

Observe como se va incrementando el valor de suma conforme va aumentando la variable cont. Mientras el incremento de la variable cont es de 1 en 1. La variable suma almacena el valor anterior + el nuevo valor de cont. Eso genera un valor acumulado.

suma| con 0 | 1 1 | 2 3 | 3 6 | 4 10 | 5

Ahora modifiquemos el algoritmo anterior para generar distintas sumatorias como las siguientes:

- a) suma de cuadrados
- b) suma de cubos
- c) suma de inversos de un numero $1/n$ donde n va de 1 hasta N. Un valor ingresado por el usuario
- d) suma de raices cuadradas.

In [1]:

```
#solucion a)
n = int(input('ingrese el valor de n:'))
cont = 1
suma = 0
while cont <= n:
    suma = suma + cont**2
    cont = cont+1
print(suma)
```

ingrese el valor de n:10
385

In []:

```
#solucion b)

n = int(input('ingrese el valor de n:'))
cont = 1
suma = 0
while cont <= n:
    suma = suma + cont**3
    cont = cont+1
```

```
print (suma)
```

In []:

```
#solucion c)

n = int(input('ingrese el valor de n:'))
cont = 1
suma = 0
while cont <= n:
    suma = suma + 1/cont
    cont = cont+1
print (suma)
```

Modifique el programa anterior para que devuelva el numero con una precisión de 3 dígitos. use round(valor, 3) en la ultima sentencia print(round(suma, 3))

In []:

```
#solucion d)
import math
n = int(input('ingrese el valor de n:'))
cont = 1
suma = 0
while cont <= n:
    suma = suma + math.sqrt(cont)
    cont = cont+1
print (round (suma,4))
```

Ahora imprimamos una matriz de simbolos

In []:

```
n = int(input('ingrese n:'))
cont1 = 1
while cont1 <= n:
    cont2 = 1
    while cont2 <= n:
        print('@', end= ' ')
        cont2 = cont2+1
    print()
    cont1 = cont1+1
```

Ahora modifiquemos la tabla para que imprima una secuencia de numeros de 1 hasta nxn

In [2]:

```
n = int(input())

cont3 = 1
cont1 = 1
```

```
while cont1 <= n:
    cont2 = 1
    while cont2 <= n:
        print(cont3, end=' ')
        cont3 = cont3 + 1
        cont2 = cont2+1
    print()
    cont1 = cont1+1
```

10

1 2 3 4 5 6 7 8 9 10

11 12 13 14 15 16 17 18 19 20

21 22 23 24 25 26 27 28 29 30

31 32 33 34 35 36 37 38 39 40

41 42 43 44 45 46 47 48 49 50

51 52 53 54 55 56 57 58 59 60

61 62 63 64 65 66 67 68 69 70

71 72 73 74 75 76 77 78 79 80