

Trick or TreAt Python

Univ. Sergio Alejandro Paucara Saca

Se empleara el uso y comparacion de los ciclos de programacion While y For y con ello algunos ventajas de tener un codigo limpio.

"Cualquier idiota puede escribir código que compila, pero sólo un buen programador puede hacer código que otros entiendan", Martin Fowler.

1. Swith - Banderas o flags Booleanos

Un booleano es un vaiable que puede tomar dos valores (0 o 1) o (True o False). Muchas veces tenemos que validar o verificar muchas condiciones en nuestros programas y es cuando hacemos uso de un booleano:

Ejemplo. Verificar si el año es bisiesto.

Usaremos varios **if** para cumplir nuestra condiciones .

```
1 #Verificaremos si el anio es bisiesto
2 a = int(input())
3 flag = False
4 if a % 4 == 0:
5     if a % 100 == 0:
6         if a % 400 == 0:
7             flag = True
8
9 if flag == True:
10     print('Es bisiesto')
11 else:
12     print('No es bisiesto')
```

bool_1.py

Ejemplo. Verificar si el año es bisiesto.

Usaremos solo un **if** para cumplir nuestra condiciones .

```
1 #Verificaremos si el anio es bisiesto
2 a = int(input())
3 flag = False
4 if a % 4 == 0 and (a % 100 != 0 or a % 400 == 0):
5     flag = True
6
7 if flag == True:
8     print('Es bisiesto')
9 else:
10     print('No es bisiesto')
```

bool.py

Este es un ejemplo pequeño de como usar un booleano, solo se tuvo que verificar tres condiciones, sin embargo hay programas donde tendremos que verificar muchas mas cosas y es donde nos ayuda el uso de booleanos.

2. Bucle For

For, nos permite manejar cuantas veces queremos repetir un proceso desde un limite hasta otro limite, sin embargo en Python nos permite iterar sobre algun item o items de listas o cadenas, al trabajar solo con numeros como en otros lenguajes de programacion

necesitaremos ayuda de la funcion `range()`.

Funcion `range()`: La funcion range nos dara una secuencia de numeros por defecto empezara en el 0, y su incremento sera de 1.

La funcion `range`, puede recibir hasta tres parametros que son **inicio**, **fin**, e **incrementador**.

`range(inicio, fin, incrementador)`

- Cuando le enviamos solo un parametro a nuestro `for` mediante la funcion `range(x)` por defecto itera todo los datos desde el numero 0 hasta un numero menor del parametro enviado, por ejemplo:

Entrada:

4

Salida:

0

1

2

3

```
1 for i in range(n):  
2     print(i)
```

unParametro.py

- Cuando enviamos dos parametro a la funcion `range(x,y)`, al enviar dos parametros, el primer parametro sera nuestro **inicio** y el segundo parametro sera el **fin**, sin embargo no llegara hasta el **fin**, sino un numero antes, por ejemplo:

Entrada:

2

7

Salida:

2

3

4

5

6

```
1 for i in range(x,y):  
2     print(i)
```

dosParametros.py

- Cuando enviamos un tercer parametro, la forma en que trabaja sera similar a la anterior, sin embargo ahora el tercer parametro sera nuestro incrementador, por ejemplo:

Entrada:

2

10

1

Entrada:	6
2	7
10	8
2	9

Salida:
2

Entrada:	4
2	6
10	8
3	

Salida:	Salida:
2	2
3	5
4	8
5	

En los anteriores ejemplo vimos como recurrir un **for** en rango con un incrementador, y ahora veremos como funciona con un decrementador, osea un numero negativo. Como vamos a decrementar entonces tenemos que empezar en un numero mayor y nuestro fin sera un numero menor al mayor.

Entrada:	Salida:
10	10
2	9
-1	8
	7
	6
	5
	4
	3

Entrada:	Salida:
10	10
2	8
-2	6
	4

Entrada:	Salida:
10	10
2	7
-3	4

```

1 #Donde x = inicio , y = fin , z = incremetador o decrementar
2 for i in range(x, y, z):
3     print(i)

```

tresParametro.py

3. Bucle While

Un bucle While, al igual que un bucle For realiza iteracion, mientras se cumpla una condicion, es inevitable que esta condicion se rompa en algun momento, ya que se puede tener un While infinito.

STRING

```
1      # 0   1   2   3
2      # h   o   l   a
3      # -4  -3 -2  -1
4  cad = "hola"
5  print(cad[0])
6  print(cad[-1])
```

Listing 1: Crear un matriz de N x M llenada de 0's

4. Lectura Datos

```
1 # Leer datos hasta fin de archivo
2 import sys
3 for k in sys.stdin:
4     print(k)          # Donde k es el dato que se lee hasta fin de archivo
5     k = k.strip()     # Se recomienda hacer un strip() a 'k' ya que es una
                        # cadena con salto de linea
```

Listing 2: Leer datos hasta fin de archivo

```
1 # Leer un dato
2 n = input()          # Para cadenas
3 print(n)
4 n = int(input())     # Para numeros
5 print(n)
```

Listing 3: Leer un dato

```
1 # Leer datos con separacion : -
2 x, y, z = map(int, input().split(":"))
3 print(x, y, z)
```

Listing 4: Leer datos con separacion : ó -

```
1 # Leer varios datos en un linea
2 a, b, c = map(int, input().split())
3 print(a, b, c)
```

Listing 5: Leer varios datos en un linea

```
1 # Leer N datos o una lista de cadenas
2 V = input().split()
3 print(V)
```

Listing 6: Leer N datos o una lista de cadenas

```
1 # Leer N numeros o una lista de numeros
2 # Opcion 1
3 num = list(map(int, input().split()))
4 print(num)
5 # Opcion 2
6 num2 = [int(i) for i in input().split()]
7 print(num2)
```

Listing 7: Leer N numeros o una lista de numeros

```

1 # Leer una matriz de cadenas de N x M, cuando los datos estan separados
2 # a b c d
3 # e f g h
4 # i j k l
5
6 n, m = map(int, input().split())
7 mat = []
8 for i in range(n):
9     mat.append(input().split())
10 # Imprimir la matrix
11 for i in range(n):
12     for j in range(m):
13         print(mat[i][j], end = ' ')
14     print('')

```

Listing 8: Leer una matriz de cadenas de N x M cuando los datos estan separados

```

1 # Leer una matriz de cadenas de N x M, cuando las datos estan juntos
2 # abcd
3 # efgh
4 # ijkl
5 n, m = map(int, input().split())
6 mat = []
7 for i in range(n):
8     mat.append(list(input()))
9 # Imprimir la matrix
10 for i in range(n):
11     for j in range(m):
12         print(mat[i][j], end = ' ')
13     print('')

```

Listing 9: Leer una matriz de cadenas de N x M cuando las datos estan juntos

```

1 # Leer una matriz de numeros N x M
2 n, m = map(int, input().split())
3 mat = []
4 for i in range(n):
5     mat.append(list(map(int, input().split())))

```

Listing 10: Leer una matriz de numeros N x M

```

1 # Imprimir la matrix
2 for i in range(n):
3     for j in range(m):
4         print(mat[i][j], end = ' ')
5     print()

```

Listing 11: Imprimir la matriz

```

1 # Crear un matriz de N x M llenada de 0's
2 n, m = map(int, input().split())
3 mat = [[0 for _ in range(m)] for _ in range(n)]
4 print(mat)

```

Listing 12: Crear un matriz de N x M llenada de 0's