

# Guide 4 Loops

April 7, 2019

## Guía 4: Ciclos

Ingeniería en Estadística, Universidad de Valparaíso

Profesor: Eduardo Jorquera - eduardo.jorquera@postgrado.uv.cl

Los ciclos son muy importantes para la programación estadística, y así básicos para la programación en general. Son usados para construir una tarea repetitiva una y otra vez. La tarea es repetida hasta que una condición se mantenga como cierta, después de lo cual el ciclo termine y si la salida ocurre sin un break, entonces la parte del else se ejecuta. La sintaxis de los ciclos es como sigue:

```
In [ ]: while (declaración):  
    ...  
    Aquí van las instrucciones  
    ...  
else:  
    ...  
    Aquí hay más instrucciones  
    ...
```

La tabulación (ó indentación) al inicio de la línea indica que allí están las instrucciones correspondientes al ciclo. Por esta razón, es de suma importancia que las uses cuidadosamente. También, la parte del else es un agregado que sólo Python tiene.

Ahora te mostraré un simple ejemplo de cómo puedes usarlo:

```
In [2]: s=1  
        while (s<10):  
            print(s, "\n")  
            s=s+1  
        else:  
            print("No hay mas eses uwu")
```

1

2

3

4

5

6

7

8

9

No hay mas eses uwu

## 1 Actividad:

Haga un programa que muestre un número (ingresado por el usuario) y su factorial usando un ciclo while para la multiplicación:

In [ ]:

## 2 Actividad:

Haga un programa que muestre un número (ingresado por el usuario) y su potencia usando un ciclo while para la multiplicación del número por sí mismo:

In [ ]:

Para eso son los ciclos while. Pero si queremos obviar la creación de una variable que recorra cada número, podemos usar los ciclos for, donde una variable recorre todo un “dominio” de valores, como en el siguiente ejemplo.

Qué elementos contiene range? Recuerdas cómo acceder individualmente a cada uno de ellos?

```
In [11]: for i in range(5):  
         print(i)
```

0

1

2

3

4

En general, los ciclos for sirven no sólo para trabajar con números, sino que para cualquier colección de elementos; mira cómo puede servir con una lista de frutas:

```
In [12]: fruits = ['plátano', 'manzana', 'mango']  
         for index in range(len(fruits)):  
             print('fruta actual :', fruits[index])
```

```
fruta actual : plátano
fruta actual : manzana
fruta actual : mango
```

También puedo poner un ciclo for dentro de otro ciclo for, por ejemplo si quiero mostrar una matriz, puedo hacer lo siguiente:

```
In [26]: nums=range(1,10)
         k=0
         for i in range(3):
             for j in range(3):
                 print(nums[k], end=" ")
                 k=k+1
             print()
```

```
1 2 3
4 5 6
7 8 9
```

### 3 Tarea

En los siguientes ejercicios use los ciclos for.

1. Escriba un programa para generar los siguientes patrones en Python, donde la cantidad de filas las indica el usuario:

```
In [ ]: *
        * *
        * * *
        * * * *
```

```
In [ ]: 1
        2 2
        3 3 3
        4 4 4 4
```

```
In [ ]: 2
        2 3
        2 3 4
        2 3 4 5
        2 3 4 5 6
```

```
In [ ]: 1
        2 3
        4 5 6
        7 8 9 10
```

2. Pregunte al usuario por un número entero y encuentre qué número menor o igual a este es primo usando un ciclo `for`.
3. A la pregunta anterior, indique cuáles son los factores de tal número.
4. Pregunte al usuario para que ingrese dos números y encuentre el mínimo común múltiplo (MCM) entre ellos. Por ejemplo, si los números ingresados son 30 y 20, el MCM entre ellos es 60.
5. Pregunte al usuario por dos números y halle el factor común más grande. Por ejemplo, si los números son 30 y 20, el factor común más grande es 10.
6. Halle la media y la varianza de una lista de  $n$  números ingresados por el usuario.
7. Pregunte al usuario por los valores de  $a$  y  $b$  y encuentre  $a^{ba}$ .