**Bucles 'WHILE' - 'FOR' en Python: Introducción a la computación**

Los **bucles** en el **lenguaje de programación** pueden ejecutar líneas de código tantas veces como se necesite o tantas veces como se desee[.](https://www.tusclasesparticulares.com/blog/que-lenguaje-programacion-aprender-primero)

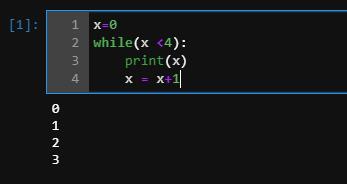
### **Bucle 'While'**

Ejecuta una sección de código en base a una **condición**, al igual que el control “**IF ELSE**”. Sin embargo, ejecutará el código mientras la condición se mantenga. El **código** dentro del dominio del **WHILE** se ejecutará siempre y cuando **la condición se mantenga**.

**Estructura del WHILE:**

1. while <CONDICIÓN> :
2. <CÓDIGO A EJECUTAR>

**Ejemplo:**



**Entender las líneas de código:**

1. Se crea la **variable** X con valor 0
2. **Inicia el bucle WHILE** verificando la condición x<4. El valor de entrada es 0, por lo que ejecuta el código dentro del bucle.
3. Mediante el comando **print(x)** muestra en pantalla el valor de x
4. **Aumenta en una unidad al valor de X**, esto se repetirá en cada iteración. Terminado de ejecutar esta línea, el bucle while reiniciará en la línea 3, esto hasta que el valor de x supere a 4.

### **Bucle 'FOR'**

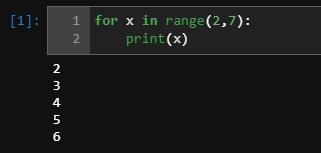
Del mismo modo que el 'WHILE' loop, el **FOR** se usa para r**epetir bloques de código**, pero a diferencia del 'WHILE' loop donde se dependía de una**condición (TRUE, FALSE)**, el bucle FOR finaliza cuando ya no tiene más elementos que iterar.

**Estructura FOR:**

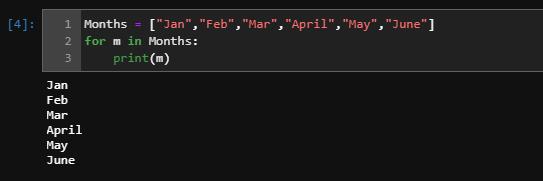
1. FOR <elemento> IN <elementos a iterar>:
2. <CÓDIGO A EJECUTAR>

**Ejemplo:**

Estas dos**líneas de código** iteran los números declarados entre 2 y 7, tomando el 2 pero no el 7.



En términos generales el bucle FOR puede iterar cualquier tipo de objeto que se encuentre **indexado**. Como por ejemplo una lista, un diccionario, un numpy array etc.



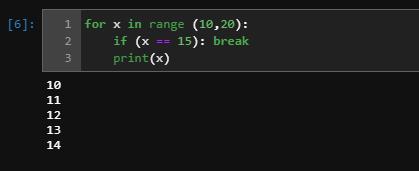
**Entender las líneas de código:**

1. **Creamos la lista Months**, la cual contiene los strings de los meses del año de enero hasta junio.
2. **Inicia el FOR**, el cual define la variable “m” cambiará de valor en cada iteración, tomando los valores que tenga el objeto Months.
3. Imprimirá en pantalla el **valor que tome “m”** en cada iteración.

### **'Break' en los bules FOR o WHILE**

Para **interrumpir o romper un bucle FOR o WHILE** podemos usar el **comando BREAK**. El comando BREAK solo puede ser usado en los bucles para interrumpirlos e impedir que sigan avanzando.

**Ejemplo**:



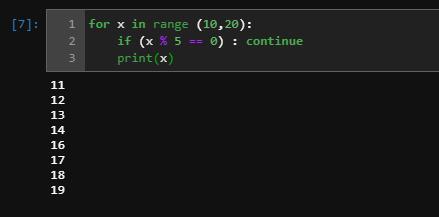
**Entender las líneas de código:**

1. **Inicia el FOR** creando la variable X, el cual tomará los valores entre 10 y 20, exceptuando el 20.
2. Se consulta si el valor de x es 15. **En caso sea verdadero, se romperá** el FOR y no se continurá con su ejecución, impidiendo que se ejecute la línea 3.
3. **En caso de que FOR no se interrumpiera** en la línea 2, se imprimirá el valor actual de x, luego reiniciará en la línea 2.

### **'Continue' en los bucles FOR**

El **comando 'continue'** termina la iteración actual y pasa a la siguiente, esto no quiere decir que el FOR se detiene o rompé, simplemente interrumpe la iteración actual y pasa a la siguiente. Este comando suele ser implementado cuando se requiere **evitar ciertos cálculos** o la creación de ciertos objetos.

**Ejemplo**:



En este ejemplo tenemos declarado un **bucle FOR** que itera para cada valor entre 10 y 19, pero de estos solo se iterarán aquellos que no sean divisibles por 5. Por lo que del rango entre 10 y 20, solo hay 3 números múltiplos de 5, (10, 15, 30). Por lo tanto, el código exceptuará al 10,15,20.

**Entender las líneas de código:**

1. **Se inicia el FOR**, para iterar entre el rango de numero entre 10 y 20, sin incluir al 20.
2. Se consulta si el número es divisible entre 5, **en caso lo sea, se ejecutará el comando “continue”.**
3. Mediante el **código print(x)** se muestra en la pantalla el valor actual de x.

[[](https://www.tusclasesparticulares.cl/blog/todo-necesario-aprender-programacion)](https://www.tusclasesparticulares.cl/blog/todo-necesario-aprender-programacion" \t "_blank)

[Artículo relacionado](https://www.tusclasesparticulares.cl/blog/todo-necesario-aprender-programacion" \t "_blank)

**[Todo lo necesario para aprender programación](https://www.tusclasesparticulares.cl/blog/todo-necesario-aprender-programacion" \t "_blank)**

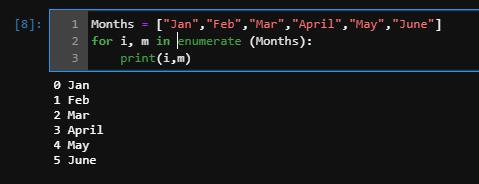
[Hoy en día todas las profesiones dependen tarde o temprano de un programador. Todos los negocios tienen o deben tener algo de pres...](https://www.tusclasesparticulares.cl/blog/todo-necesario-aprender-programacion" \t "_blank)

### **'Enumerate ()' en el bucle FOR**

**Python** tiene varias funciones integradas que podemos utilizar directamente, una de ellas es '**ENUMERATE**', usada para dar un numero de índice a cada elemento de un objeto iterable, de forma ordenada. Mayormente usada en los bucles FOR.

**Ejemplo**:

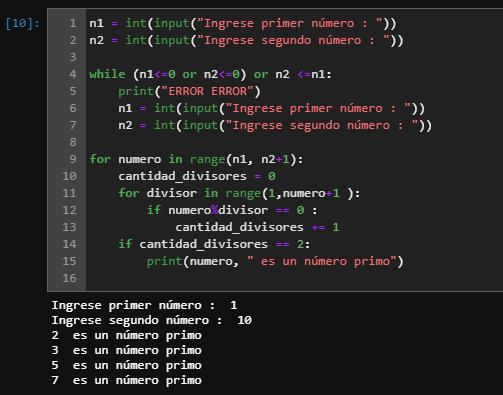
En el siguiente ejemplo enumeraremos cada mes (**string**) que se encuentre **dentro de la lista Month**s, para ello usaremos la **función enumerate().**



Con esto lograremos crear un**print()** que nos mostrará el índice o posición de cada mes dentro de la lista y a su vez al mes al que corresponde esa posición.

### **Caso examen – Introducción a la computación**

Escriba un **scritp en Python** que lea dos números enteros [>0], el primer número debe ser menor que el segundo número, si esto no se cumple debe volver a ingresar los números. Luego el programa debe mostrar todos los números primos comprendidos entre estos números (ambos inclusive).



**Entender las líneas de código:**

1. El programa solicita el **primer valor de entrada n1**
2. El programa solicita el **segundo valor de entrada n2**
3. **Inicia el proceso iterativo de verificación**, donde n1 y n2 tienen que ser mayores a cero y además n2>n1, por lo tanto se colocan los declaraciones inversas para que se repita hasta que se cumpla lo que queremos.
4. **Dentro del while**,**imprime error** en la pantalla.
5. Dentro del while, **solicita el primer valor n1**
6. Dentro del while, **solicita el segundo valor n2**
7. **Inicia el proceso de iteración entre los números n1 y n2**, incluido el n2. Para determinar quiénes son número primo.
8. **Inicializa el contador de divisores**, el cual se reinicia para cada numero.
9. **Inicia una segunda iteración anidada** la cual sirve para buscar divisores para cada número comprendido en la lista de la línea 9.
10. **Para cada posible divisor se consulta si existe el residuo al dividir al número analizado**, de ser el caso se procede a aumentar la cantidad de divisores en la línea 13.
11. Acabando la iteración con todos los posibles divisores, **se consulta si es un número primo**, de ser verdadero, se imprime en pantalla si el número es primo.
12. **BUCLE FOR LINEA 9 del código**: REPETIR PARA CADA NÚMERO de la lista de número entre n1 y n2

### **Conclusiones**

En esta corta explicación pudimos observar las formas más comunes de utilizar las **herramientas FOR y WHILE**, con las que podemos realizar tareas repetitivas. Esto es el **principio de la programación** ya que las computadoras saben hacer una cosa muy bien y esto es hacer siempre lo mismo de la forma exacta las veces que se requiera.

Con ejemplos sencillos no es posible[visualizar la magnitud de su alcance en el**desarrollo de software**](https://www.tusclasesparticulares.cl/blog/porque-importante-aprender-informatica), pero mientras más se interiorice más interesantes se convertirán, éxitos en tu viaje en este **mundo de la programación**.