



Unidad No.2 Lenguajes Altamente Dinámicos/débilmente Tipados

TI2011 – Introducción a la Programación

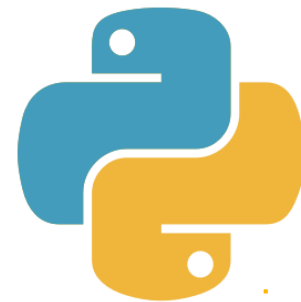
For Loop



FOR Loop

Otro tipo de ciclo disponible en Python proviene de la observación de que a veces es más importante **contar los "giros o vueltas" del ciclo** que verificar las condiciones.



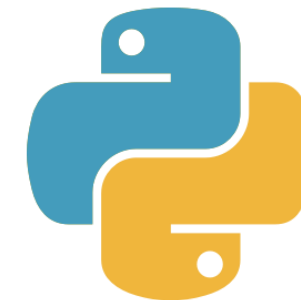


En un ejemplo de ciclo que debe ejecutarse exactamente diez veces. Al utilizar el ciclo **while** para hacerlo, puede tener este aspecto:

```
i = 0
while i < 10:
    # hacer_algo()
    i += 1
```

El ciclo for está diseñado para realizar tareas más complicadas, puede "explorar" grandes colecciones de datos elemento por elemento.

```
for i in range (10):
    #hacer algo()
    pass
```



Palabra Reservada, Inicia el ciclo, No hay condición después de eso

Cualquier variable después de la palabra reservada *for* es la **variable de control** del ciclo; cuenta los giros del ciclo y lo hace automáticamente.

La palabra reservada *in* introduce un elemento de sintaxis que describe el rango de valores posibles que se asignan a la variable de control.

```
for i in range (10) :  
    #hacer algo()  
pass
```

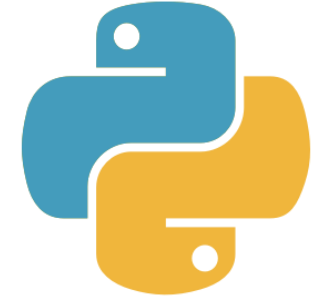
la palabra clave *pass* dentro del cuerpo del ciclo - no hace nada en absoluto; es una **instrucción vacía**

La función `range()` (una función muy especial) es responsable de generar todos los valores deseados de la variable de control; como ejemplo, la función creará (incluso podemos decir que alimentará el ciclo con) valores subsiguientes del siguiente conjunto: 0, 1, 2 .. 9;

nota: la función `range()` comienza su trabajo desde 0 y lo finaliza un paso (un número entero) antes del valor de su argumento.

¿Cuál sería la salida de valores para el siguiente código?

```
for i in range (2, 8):  
    print("El valor de i es actualmente", i)
```



La función **range()** también puede aceptar tres argumentos.

El tercer argumento es un **incremento**: esto significa que es un valor agregado para controlar la variable en cada giro del ciclo (el valor predeterminado del incremento **es 1**).

```
for i in range(2, 8, 3):  
    print("El valor de i es actualmente", i)
```

Nota: el conjunto generado por range() debe ordenarse en un orden ascendente. No hay forma de forzar el range() para crear un conjunto en una forma diferente. Esto significa que el segundo argumento de range() debe ser mayor que el primero.

Actividad:

1. Utilizando ciclo for, escriba un programa que genere 20 números correlativos desde un rango definido por el usuario.
2. Utilizando ciclo for, escriba un programa que genere 20 números correlativos desde un rango definido por el usuario pero con un incremento de 3.
1. Utilizando ciclo for, escriba un programa que muestre la serie de exponentes correspondiente a un byte (2^n). El algoritmo mostrará el resultado de la suma de todos los exponentes.
2. Utilizando ciclo for, escriba un programa que ingresado el nombre de una persona entregue como respuesta las letras del nombre tabuladas de forma horizontal.
3. Utilizando ciclo for, escriba un programa que ingresado el nombre de una persona, entregue las letras del nombre ordenadas de manera vertical (línea a línea). Además el algoritmo entregará el número de letras del nombre, en caso de que el nombre sea compuesto ej. Juan Carlos, no se contará el espacio como letra por tanto el total de letras del nombre será 10.
4. Calcular e imprimir la tabla de multiplicar, del 1 al 20 para un valor x ingresado por un usuario, debe realizar el algoritmo mediante un ciclo FOR.
5. Sumar los números impares que existen entre 10 y 60.

