



Universidad Autónoma de Baja California



Asignatura: Programación con Python

Alumno: Rodríguez Gómez Alberto

Docente: Parra Galaviz Ray Brunett

Grupo: 372

Semestre: 2018-2

Estructuras de Control e Iteración

En lenguajes de programación, las estructuras de control permiten modificar el flujo de ejecución de las instrucciones de un programa. En Python las estructuras de control de flujo condicionales, se definen mediante el uso de tres palabras claves reservadas, del lenguaje: if (si), elif (sino, si) y else (sino).

If: Condición de tipo booleano, y cuando ésta se cumple se ejecuta el bloque de código dentro de ella.

Elif: abrevia el if else if, si esta se llega a cumplir, se ejecuta el bloque que se encuentra dentro de esta, similar al if.

Else: Condición es de tipo booleano, esta va justo después de un if o elif. Al correr el código primero pasa por las condiciones if / elif, si estas no se llegan a cumplir entonces se ejecuta el bloque de código dentro del else.

Estructuras de Iteración

A diferencia de las estructuras de control condicionales, las iterativas (también llamadas cíclicas o bucles), nos permiten ejecutar un mismo código, de manera repetida, mientras se cumpla una condición.

En Python se dispone de dos estructuras cíclicas:

- El bucle while
- El bucle for

Bucle while

Este bucle, se encarga de ejecutar una misma acción "mientras que" una determinada condición se cumpla. Es decir, que la acción se estará realizando repetidamente hasta que llegue el momento en que la condición ya no se cumpla, entonces se detendrá el ciclo.

Bucle for

Este bucle tiene la siguiente estructura:

Comienzo del bucle

for i in lista_elementos:

accion1

accion2

 ...

accionN

Fin del bucle

i representa, en cada iteración, un elemento en lista_elementos. Este objeto (lista_elementos) puede ser una secuencia de números, una lista (un tipo de datos compuesto en Python, que veremos más adelante), o cualquier “cosa” sobre la que se pueda iterar. Por último, accion1, accion2, hasta accionN, representan una serie de operaciones o declaraciones que por lo general afectan el elemento i actual.

En Python, range() es una función que devuelve una lista de números enteros. Acepta como argumentos el principio de la lista, el fin, y el incremento entre un elemento y el siguiente (en ese orden). También existe la posibilidad de omitir uno o dos de los mismos.

- **range(n)** devuelve una lista de enteros comenzando por 0 y finalizando en $n - 1$.
- **range(comienzo, fin)** devuelve la lista de enteros comprendidos entre comienzo y fin, sin incluir a este último.
- **range(comienzo, fin, paso)** igual que el caso anterior, únicamente que entre comienzo y el próximo elemento existirá una diferencia de paso, y así sucesivamente con el próximo elemento.

El range se utiliza cuando se desea emplear el for pero no se tiene una lista o tupla la cual imprimir o utilizar como cantidad de iteraciones.

Generalmente, se prefiere utilizar un bucle for cuando se conoce la cantidad de elementos sobre los que se necesita iterar. Es decir, sabemos de antemano cuántas veces se desea realizar tal o cual operación. Por otra parte, el bucle while se usa cuando no conocemos ese dato. En ese caso, la iteración se lleva a cabo mientras la condición asociada evalúe a True.

Bibliografía

<https://blog.carreralinux.com.ar/2017/05/uso-del-bucle-for-en-python/>

https://librosweb.es/libro/python/capitulo_2/estructuras_de_control_de_flujo.html

https://www.fing.edu.uy/inco/cursos/fpr/wiki/index.php/Estructuras_de_Control_en_Python