



Universidad Autónoma de Baja California

Facultad de Ingeniería, Arquitectura y Diseño

Ingeniero en Computación

Asignatura:

Programación Estructurada

Actividad 6:

Ciclos y Funciones

Brayan Arturo Rocha Meneses

Matricula:

371049

Ensenada Baja California 26 de Septiembre del 2023

Introducción:

Los ciclos y funciones son dos conceptos fundamentales en Python, así como en la programación en general. Juntos, permiten a los programadores escribir código más eficiente, modular y reutilizable.

Los ciclos, son estructuras de control que permiten repetir un bloque de código múltiples veces. En Python, los dos tipos de ciclos más comunes son el "for" y el "while".

"for": Este tipo de bucle se utiliza para iterar sobre una secuencia (como una lista, una tupla, un diccionario, etc.) o para ejecutar un bloque de código un número específico de veces.

"while": Este bucle se ejecuta mientras una condición especificada sea verdadera. Es útil cuando no se conoce de antemano cuántas veces se debe repetir el bloque de código.

Las funciones son bloques de código reutilizables que realizan una tarea específica. Las funciones se definen con la palabra clave "def" y pueden tener parámetros que permiten la entrada de datos.

Las funciones son útiles para dividir un programa en piezas más pequeñas y manejables, lo que facilita el mantenimiento y la depuración. También promueven la reutilización del código, ya que puedes llamar a una función en múltiples lugares de tu programa.

Competencia:

Utilizar las declaraciones del lenguaje interpretado, a través del análisis de sus conceptos, observación de ejemplos y la experimentación en un entorno interactivo de ejecución, para escribir sentencias de código que efectúen control de flujo o manejo de bifurcaciones y resulten en la ejecución de operaciones específicas, de manera creativa y ordenada.

Fundamentos:

¿Qué son los ciclos?

En programación, los ciclos son estructuras que permiten ejecutar un conjunto de instrucciones de manera repetitiva mientras se cumple una condición determinada. Los ciclos son una herramienta esencial en cualquier lenguaje de programación, incluyendo Python.

Introducción a los ciclos

Los ciclos permiten automatizar tareas repetitivas, lo que reduce la cantidad de código necesario para lograr un resultado y hace que el código sea más legible y fácil de mantener. En Python, existen dos tipos principales de ciclos: el ciclo "for" y el ciclo "while".

Tipos de ciclos en Python

El ciclo "for": este ciclo se utiliza para recorrer una secuencia de elementos, como una lista o una tupla, y ejecutar un conjunto de instrucciones para cada elemento. En Python, se utiliza la palabra clave "for" para iniciar el ciclo y se utiliza una variable para hacer referencia a cada elemento de la secuencia.

El ciclo "for" se utiliza cuando se conoce de antemano la cantidad de veces que se quiere repetir una acción, es decir, cuando se sabe la cantidad exacta de elementos que se van a recorrer en una lista o una secuencia. Este ciclo se compone de una variable de control que toma valores consecutivos de una lista, tupla, diccionario o cualquier otra estructura de datos iterable. Dentro del cuerpo del ciclo "for", se pueden ejecutar una serie de instrucciones que se repiten en cada iteración del ciclo.

El ciclo "while": este ciclo se utiliza para ejecutar un conjunto de instrucciones mientras se cumpla una condición determinada. En Python, se utiliza la palabra clave "while" para iniciar el ciclo y se especifica una condición que debe ser verdadera para que el ciclo continúe.

DEFINICION DE: Contador, Acumulador, Bandera, Elemento Iterable

Contador: una variable que se utiliza para llevar la cuenta del número de veces que se ha ejecutado un ciclo.

Acumulador: una variable que se utiliza para acumular los resultados de cada iteración de un ciclo.

Bandera: una variable booleana que se utiliza para controlar la ejecución de un ciclo.

Elemento iterable: una secuencia de elementos que se puede recorrer utilizando un ciclo "for".

PALABRAS CLAVES: continue, break, pass, else

En Python, los ciclos for y while pueden ser controlados con 4 palabras clave especiales: continue, break, pass, else.

continue: esta palabra clave se utiliza dentro de un ciclo para omitir la iteración actual y saltar a la siguiente iteración. En otras palabras, si se encuentra una determinada condición, la iteración actual se salta y el ciclo se continúa con la siguiente iteración. Por ejemplo, supongamos que se quiere imprimir los números del 1 al 10, excepto el 5. Podemos usar continue para omitir la iteración cuando el número es igual a 5:

break: esta palabra clave se utiliza dentro de un ciclo para salir del ciclo por completo. Si se encuentra una determinada condición, el ciclo se detiene y se ejecuta el código después del ciclo. Por ejemplo, supongamos que se quiere imprimir los números del 1 al 10, pero se quiere detener la impresión en el número 5. Podemos usar break para salir del ciclo cuando el número es igual a 5:

pass: esta palabra clave se utiliza para indicar que no se debe realizar ninguna operación. Es útil cuando se necesita una sentencia en un lugar donde la sintaxis de Python lo requiere, pero no se necesita ninguna operación. Por ejemplo, supongamos que se quiere crear una función que aún no está completamente implementada. Podemos usar pass como marcador de posición para indicar que no hay ninguna operación en la función:

else: esta palabra clave se utiliza junto con un ciclo para ejecutar un bloque de código después de que el ciclo ha terminado de iterar sobre todos los elementos de una secuencia. La sección else se ejecuta solo si el ciclo se completa sin haber encontrado una sentencia break. Por ejemplo, supongamos que se quiere buscar un número en una lista y se quiere imprimir un mensaje si el número no está en la lista:

EL CICLO for

El ciclo "for" es una estructura de control de flujo en Python que se utiliza para iterar sobre una secuencia de elementos, como una lista, tupla, diccionario o conjunto. La iteración es el proceso de recorrer cada elemento de una secuencia uno por uno, y se realiza para realizar una tarea repetitiva de manera automatizada.

El ciclo "for" es uno de los ciclos más utilizados en Python debido a su simplicidad y flexibilidad. Se utiliza para ejecutar un conjunto de instrucciones un número determinado de veces, una vez por

cada elemento de la secuencia. El ciclo "for" se compone de tres partes principales: la palabra clave "for", la variable que hace referencia a cada elemento de la secuencia y la secuencia que se va a iterar.

FUNCIONES EN PYTHON

Las funciones son bloques de código reutilizables que realizan una tarea específica en un programa. En Python, las funciones son objetos de primera clase, lo que significa que pueden ser asignadas a variables, pasadas como argumentos a otras funciones y devueltas como valores de funciones.

Las funciones son una parte fundamental de la programación en Python, ya que permiten una mayor modularidad y una mejor organización del código. Al dividir un programa en funciones más pequeñas y manejables, se puede facilitar el proceso de desarrollo y mantenimiento de código. Las funciones también permiten la reutilización de código, lo que reduce la duplicación de esfuerzos y mejora la eficiencia del programa.

En Python, las funciones pueden recibir argumentos opcionales (por defecto), argumentos posicionales y argumentos con nombre. También pueden devolver valores explícitamente utilizando la palabra clave return.

En resumen, las funciones en Python son bloques de código reutilizables que permiten modularizar el código, facilitar el desarrollo y mantenimiento del programa, reducir la duplicación de esfuerzos y mejorar la eficiencia del programa.

Para definir una función en Python, se utiliza la siguiente sintaxis: Las funciones se definen mediante la palabra clave def, seguida del nombre de la función y los parámetros que recibe (si los hay), y terminando con dos puntos (:). El cuerpo de la función se define a continuación, indentado hacia la derecha.

La sintaxis básica es la siguiente:

```
def nombre_de_la_funcion(parametros):
```

```
    # Cuerpo de la función
```

```
    return valor_de_retorno
```

Explicación de los elementos de la sintaxis:

`def`: es una palabra reservada en Python que indica que se está definiendo una función.

`nombre_de_la_funcion`: es el nombre que le das a la función. Este nombre se utiliza para llamar a la función desde otras partes del programa.

`parametros`: son los valores que se pasan a la función cuando se llama. Los parámetros son opcionales, por lo que una función también puede no recibir ningún parámetro.

`Código que realiza una tarea`: es el código que se ejecuta cuando se llama a la función. Este código puede incluir cualquier instrucción válida de Python, como operaciones matemáticas, estructuras de control de flujo, etc.

`return valor_de_retorno`: es la sentencia `return` que devuelve un valor opcional como resultado de la función. Si la función no devuelve ningún valor, se puede omitir la sentencia `return`.

Procedimiento:

FUNCIONES PARA SER LLAMADAS EN EL MENU

1.- FUNCIÓN QUE LEA n CANTIDAD DE NÚMEROS HASTA QUE EL USUARIO LO DESEE, DESPLEGAR LA SUMA DE LOS NÚMEROS, MEDIA Y VALOR DE LOS NÚMEROS MAYORES Y MENORES.

2.- FUNCIÓN QUE GENERE 15 NÚMEROS IMPARES ENTRE 10 Y 60 o MÁXIMO DE 25 NÚMEROS. DESPLEGAR LA MEDIA DE LOS PARES Y MEDIA DE IMPARES.

3.- FUNCIÓN QUE SIRVA PARA LEER UN RANGO DADO POR EL USUARIO. REPETIR ESTA ACCIÓN HASTA QUE EL USUARIO LO DESEE, DESPLEGAR CANTIDAD DE NUMEROS Y PROMEDIO DE LOS NUMEROS..

4.- FUNCIÓN QUE RECIBA COMO PARÁMETRO LOS VALORES PARA EL ÁREA DE UN TRIANGULO Y RETORNE SU RESULTADO

5.- FUNCION QUE SIRVA QUE SIRVA PARA EVALUAR EL PROMEDIO DE 3 CALIFICACIONES DADAS, SI EL USUARIO SU CALIFICACION ES APROBADA MANDAR MSGE QUE DIGA "felicidades avanzas al siguiente semestre", SI LA CALIFICACIONE ESTA REPROBADA EL MENSAGE "repetir materia " EL ALUMNO TENDRA UN MAXIMO DE CURSAR 3 VESES LA MATERIA, SI LA REPRUEBA 3 VESES MANDAR MSGE " lastima estas fuera de la UABC"

NOTA: VALIDAR EL PROGRAMA, DEBE MANDAR MSGES DE ERROR DE CALIFICACIONES Y NO PERMITIR LETRAS. (try: except:)

Conclusiones:

Las funciones son elementos muy importantes al programar con Python. No sólo nos permiten encapsular y reutilizar código de manera indefinida, si no que facilitan su lectura acortando la cantidad de líneas a leer y contando con nombres descriptivos para saber rápidamente que es lo que hacen.

En Python se recomienda encapsular en funciones el código que se repita constantemente y aquel que no sea fácil de descifrar o sea muy complejo. Es una buena práctica además, el definir tus funciones al principio del programa, y colocar comentarios que expliquen cómo funcionan, que valores reciben y que valores regresan.

Aunque se suele utilizar ciclos for cuando sabemos cuándo terminara un ciclo de ejecutarse y los ciclos while cuando no tenemos esta certeza, prácticamente cualquier operación puede realizarse con cualquiera de los dos ciclos.

Los ciclos nos ofrecen varias herramientas para utilizarlos, como break y continue, que te pueden servir para complementar tus ciclos y asegurarte de que siempre lleguen a un fin, sin embargo, si en tus ciclos dependes solamente de break para tener la certeza de que llegaran a un fin, deberías pensar de nuevo el planteamiento del ciclo. De igual forma ocurre con los ciclos anidados

Anexo:

https://github.com/BrayanArM98/Python/blob/main/RMBA_PY_ACT6.ipynb