



Universidad Autónoma de Baja California

Facultad de Ingeniería, Arquitectura y Diseño

Ingeniero en Computación

Asignatura:

Lenguaje de Programación Python

Actividad 5:

(try, for, range(), random())

Brayan Arturo Rocha Meneses

Matricula:

371049

Ensenada Baja California 17 de Septiembre del 2023

Introducción:

En Python, las palabras clave `try`, `for`, `range()`, y el módulo `random` son elementos fundamentales que desempeñan roles clave en el desarrollo de programas y scripts. Cada uno de estos elementos tiene un propósito específico y se utiliza en diferentes contextos para facilitar la escritura de código más eficiente y versátil.

Competencia:

Utilizar los operadores de comparación, a través de la observación de casos y la experimentación, para escribir sentencias de código que efectúen operaciones de comparación entre valores de variables y entreguen un resultado de tipo booleano, de manera ordenada y determinada.

Fundamentos:

Al programar en Python algunas veces podemos anticipar errores de ejecución, incluso en un programa sintáctica y lógicamente correcto, pueden llegar a haber errores causados por entrada de datos inválidos o inconsistencias predecibles.

En Python, puedes usar los bloques `try` y `except` para manejar estos errores como excepciones.

El bloque `try` es el bloque con las sentencias que quieres ejecutar. Sin embargo, podrían llegar a haber errores de ejecución y el bloque se dejará de ejecutarse.

El bloque `except` se ejecutará cuando el bloque `try` falle debido a un error. Este bloque contiene sentencias que generalmente nos dan un contexto de lo que salió mal en el bloque `try`.

Siempre deberías de mencionar el tipo de error que se espera, como una excepción dentro del bloque `except` dentro de como lo muestra el ejemplo anterior.

Podrías usar `except` sin especificar el . Pero no es una práctica recomendable, ya que no estarás al tanto de los tipos de errores que puedan ocurrir.

Cuando se ejecute el código dentro del bloque `try`, existe la posibilidad de que ocurran diferentes errores

Procedimiento:

`try` y `except` Estas palabras clave se utilizan para manejar excepciones en Python. Cuando se ejecuta un bloque de código dentro de un `try`, el programa monitorea si se producen errores. Si ocurre una excepción, se puede capturar y manejar utilizando un bloque `except`. Esto permite que un programa continúe ejecutándose incluso cuando se encuentra con errores, lo que lo hace más robusto y menos propenso a bloquearse inesperadamente.

El bucle ``for`` es una estructura de control que se utiliza para iterar sobre una secuencia de elementos, como listas, tuplas, cadenas de caracteres o rangos numéricos. Permite ejecutar un bloque de código repetidamente para cada elemento de la secuencia. Es una herramienta poderosa para automatizar tareas que requieren la repetición de acciones.

La función ``range()`` genera una secuencia de números enteros en un rango específico. Es comúnmente utilizada junto con bucles ``for`` para controlar el número de iteraciones. Puedes especificar el inicio, el final y el paso de la secuencia, lo que la hace extremadamente flexible.

El módulo ``random`` proporciona funciones para generar números aleatorios en Python. Esto es útil en una amplia variedad de aplicaciones, como juegos, simulaciones y estadísticas. Puedes utilizar funciones como ``random.randint()`` para generar números enteros aleatorios en un rango dado o ``random.choice()`` para seleccionar elementos aleatorios de una secuencia.

1.- Programa en Python que genere 40 números aleatorios entre el 0 y 200, desplegar los números y la leyenda de cada número si es par o impar , la cantidad de los números pares e impares así como la suma de los números pares o impares.

2.- Programa en Python que despliegue la tabla de multiplicar de un número dado (número entre el 1 y 20).

Tabla del 5

$$5 * 1 = 5$$

$$5 * 2 = 10$$

$$5*10=50$$

3.- Programa en Python que lea una calificación, las calificación deberá estar en el rango de 0 a 100, si hay un error de captura, mostrar mensaje de error. Con la calificación correcta mostrar msg de aprobado reprobado.

4.- Programa en Python que lea n cantidad de números enteros dentro de un rango dado (> 0) , el programa deberá terminar cuando el usuario introduzca el número cero. Desplegar la suma de números y la media.

5.- Programa en Python que sirva para leer el promedio de una materia. donde el usuario tendrá un máximo de 3 oportunidades de cursar la materia, si el promedio es aprobado, felicitarlo y continuar el siguiente semestre, si promedio es reprobado deberá salir mensaje de repetir materia o es baja académica si ha reprobado 3 veces.

PARTE 2:

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 y validar un número dentro de un rango dado por el usuario. repetir esta acción hasta que el usuario lo desee, desplegar cantidad de números y promedio de los números..

4.- función que reciba como parámetro los valores para el área de un triángulo y retorne su resultado

5.- función que sirva para validar un número dentro de un rango dado.

Conclusiones:

El manejo de excepciones es una parte esencial de la programación en Python y contribuye en gran medida a la robustez y estabilidad de tus programas. Al utilizar las estructuras try-except, puedes anticipar y gestionar errores de manera efectiva

Anexo:

<https://drive.google.com/file/d/1Z4aEpE8F4HjFz54fiqETge3WEOPS8e9s/view?usp=sharing>