



Universidad Autónoma de Baja
California
Facultad de Ingeniería Arquitectura
y Diseño



Materia: *Lenguaje de Programación Python*

Alumno: FABIAN AGUIAR SERGIO

Carrera: Ingeniero en computación

Fecha: 10/03/2024

Matrícula: 374317

Maestro: Pedro Nuñez Yepiz

Grupo: 432

Práctica No. : #5

Tema - Unidad : *(try, for, range(), random())*

INTRODUCCIÓN

En esta actividad se llevará a cabo el uso de diferentes tipos de estructuras , esto con el fin de entender las estructuras y cómo se pueden usar en este tipo de lenguaje, se vio que estas estructuras no son tan complicados y cómo los puede manejar uno mismo, donde usamos el 'for' es una estructura fundamental que nos permite iterar sobre elementos en una secuencia y La función 'range' es una herramienta especialmente útil para generar secuencias numéricas que pueden ser utilizadas en conjunto con el bucle `for` para ejecutar un conjunto de instrucciones un número específico de veces o para iterar sobre una colección de datos El random no se queda atrás este es una herramienta poderosa que nos permite introducir aleatoriedad en nuestros programas, todos estos tipos de funciones y estructuras nos ayudan a entender mejor este tipo de lenguaje y a como saberlo utilizar.

COMPETENCIA

En esta práctica uno tiene que aprender a usar estas funciones y estructuras para que más adelante al momento de hacer códigos más complicados uno no tenga tanta dificultad de saber utilizar estas funciones o estructuras y entienda que estos ciclos tiene su finalidad y que no es tan complicado cuando uno va practicando con estas herramientas.

FUNDAMENTOS

introducción a las funciones en Python

Una función te permite definir un bloque de código reutilizable que se puede ejecutar muchas veces dentro de tu programa.

Las funciones te permiten crear soluciones más modulares y [DRY](#) para problemas complejos.

Si bien Python ya proporciona muchas funciones integradas como `print()` y `len()`, también puedes definir tus propias funciones para usar en tus proyectos.

Una de las grandes ventajas de usar funciones en tu código es que reduce el número total de líneas de código en tu proyecto.

Las funciones son bloques de código que se pueden reutilizar simplemente llamando a la función. Esto permite la reutilización de código simple y elegante sin volver a escribir explícitamente secciones de código. Esto hace que el código sea más legible, facilita la depuración y limita los errores de escritura.

Los bucles (loops) se usan cuando se necesita repetir un conjunto de instrucciones muchas veces.

PYTHON soporta dos tipos diferentes de bucles: el bucle for y el bucle while.

Los bucles for son la opción más sencilla cuando usted sabe cuántas veces necesita repetir el bucle.

Los bucles while (bucle fundamental) son las opciones más sencillas cuando necesita mantener la repetición de las instrucciones hasta que se satisface un criterio.

Los bucles while son similares a los bucles for el cual se usa para repetir un bloque de instrucciones dentro de nuestro programa, la gran diferencia con el FOR es la forma en que PYTHON decide cuántas veces repetir el bucle. Los bucles while continúan hasta que se satisface algún criterio.

C, S. (2021, 26 abril). *Ciclo While en Python*. Control Automático Educación.

<https://controlautomaticoeducacion.com/python-desde-cero/ciclo-while-en-python/>

Bustamante, S. J. (2021, 21 febrero). *Guía de funciones de Python con ejemplos*. freeCodeCamp.org.

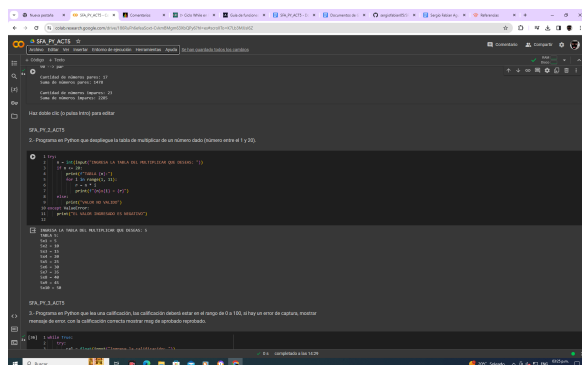
<https://www.freecodecamp.org/espanol/news/guia-de-funciones-de-python-con-ejemplos/>

PROCEDIMIENTO

En esta práctica se hicieron 10 programas para que uno sepa usar los diferentes tipos de funciones que nos propuso el profe como en algunos códigos se usó un **import random** esto para que nos dieran números al azar sin que uno los ponga, esto para que uno no se complique al imponer uno por uno se usaron ciclos **for** también para que nos dieran números o fuera más fácil la estructura de código, usamos diferentes tipos de ciclos funciones, para que uno mismo sepa estructuras y programar sin complicaciones.

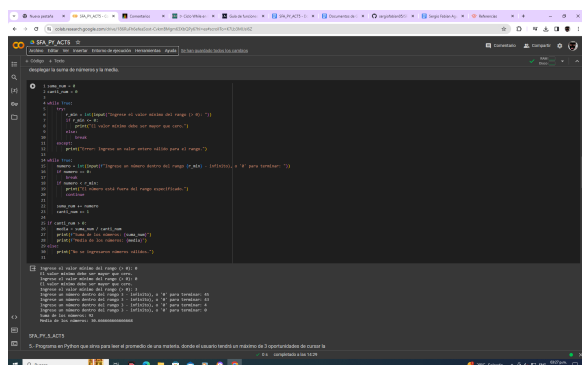
CONCLUSIÓN Y RESULTADOS

En el otro pdf se anexaron los códigos y resultados pero pondre unos aqui para poder entender mas estas funciones.



```
1 # Importamos el módulo random para generar números aleatorios
2 import random
3
4 # Función para generar un número aleatorio entre 1 y 20
5 def generar_numero():
6     return random.randint(1, 20)
7
8 # Función para imprimir la tabla de multiplicar de un número
9 def imprimir_tabla(numero):
10    for i in range(1, 11):
11        print(f"{numero} x {i} = {numero * i}")
12
13 # Programa principal
14 if __name__ == "__main__":
15     numero = generar_numero()
16     print(f"Se generó el número: {numero}")
17     imprimir_tabla(numero)
```

En este código se mira que uno le daba un número aleatorio del 1 al 20 y este mismo te soltaba la tabla de multiplicar, tiene agregado un try y except ya que si el usuario ingresa una letra en vez de número le avise que no se puede ejecutar ya que no agrego un numero.



```
1 # Importamos el módulo random para generar números aleatorios
2 import random
3
4 # Función para generar un número aleatorio entre min y max
5 def generar_numero(min, max):
6     return random.randint(min, max)
7
8 # Función para verificar si un número está dentro de un rango
9 def verificar_rango(numero, min, max):
10    return min <= numero <= max
11
12 # Programa principal
13 if __name__ == "__main__":
14     # Pedir al usuario los valores de rango
15     min = int(input("Ingrese el valor mínimo del rango (entre 0 y 100): "))
16     max = int(input("Ingrese el valor máximo del rango (entre 0 y 100): "))
17
18     # Generar 5 números aleatorios dentro del rango
19     for i in range(5):
20         numero = generar_numero(min, max)
21         print(f"Número generado: {numero}")
22
23         # Verificar si el número está dentro del rango
24         if verificar_rango(numero, min, max):
25             print(f"El número {numero} está dentro del rango [{min}, {max}].")
26         else:
27             print(f"El número {numero} no está dentro del rango [{min}, {max}].")
```

en este código pedía que uno proporciona un rango máximo y mínimo que agregaran números y te marcaran si estaba dentro del rango o no, lo mismo que el otro se usó try y except igual para que el usuario no agregara letras y pusiera números

En conclusión en esta práctica se quiso entender cómo funciona cada tipo de ciclo o funciones ya que esto será una gran ayuda para códigos que hagamos mas adelante y que uno vaya agarrando el ritmo y sepa diferencial como se usa cada uno y cómo puede hacer que su código sea más reducido y optimizada y entendible para el usuario, se obtuvo lo esperado y los resultados son los que se querían obtener y con esto poder hacer códigos ya sean faciles o sean complicados para uno mismo.

GIT HUB

https://github.com/sergiofabian05/SFA_PY_ACT5.py.git