INGENIERIA EN COMPUTACION

Materia: Programación Python

Alumno: Ayon Dimas Andres

Matricula: 366845

Profesor: Pedro Nunes Yepiz

Actividad: #6

Tema: Ciclos y funciones

Ensenada B.C 25 de septiembre del 2023

INTRODUCCION

Aquí podremos ver la importancia de lo que son los ciclos y las funciones. Podremos ver las funciones, pero ahora en Python en lugar de C, y estas funciones en Python no son muy distintas a las de C. Sirven para lo mismo, pero con una pequeña modificación a la hora de aplicar la sintaxis y también las usaremos, pero con ciclos y veremos cómo es que estas se aplican y para qué sirven estas funciones y a su vez también veremos para que sirven y en qué momento es el adecuado para usarlas en nuestros programas en C.

COMPETENCIA

Dominar las funciones en Python y aprender cuando se deben utilizar estas funciones y como aplicarlas a nuestros programas en Python, así mismo como seguir utilizando ciclos y condiciones, esto con el fin de poder hacer nuestros programas más eficientes, ya que al seguir utilizando estas sentencias y seguir aplicando esto a nuestros programas nos ayuda a seguir mejorando a la hora de implementar las condiciones, ciclos y funciones en Python.

FUNDAMENTOS

Una función en Python (y en cualquier otro lenguaje de programación) es un bloque de líneas de código o un conjunto de instrucciones cuya finalidad es realizar una tarea específica. Puede reutilizarse a voluntad para repetir dicha tarea.

Las funciones nos ayudan a que el código sea más fácil de leer y entender. Además, saber cómo generar funciones habla muy bien de la capacidad de abstracción de un programador, en tanto que es una de las habilidades técnicas más difíciles de consolidar en el desarrollo de software. Para que comiences en este camino, te daremos una guía básica de cómo definirlas.

Algunas consideraciones que debemos tener en cuenta al momento de crear y utilizar nuestras propias funciones en Python son:

- Siempre retornar un valor de respuesta.
- Definir valores por default para todos los parámetros de entrada que sea posible.
- Verificar que la cantidad de parámetros con que invocamos la función son los mismos que se definieron al momento de crear la función.
- Python es un lenguaje que utiliza la indentación para delimitar los bloques de código, por lo que tenemos que verificar que las líneas que queremos que pertenezcan a la función estén correctamente indentadas.
- Otra recomendación es utilizar solo variables locales y evitar el uso de variables globales para evitar comportamientos inesperados.

Python es un lenguaje interpretado: aunque las líneas de código de la definición de la función contengan algún tipo de error, el intérprete de Python lo señalará hasta el momento en que se invoque la función.

PROCEDIMIENTO

```
# ejemplo de definición de una función

def suma (a, b):

return a + b

result = suma (8,5)

# result = 13
```

También puedes definir valores predeterminados para los parámetros de entrada de tus funciones. Esto te permite indicar que dichos parámetros son opcionales, es decir, que pueden no ser informados al momento de invocar la función; sin embargo, como cuentan con parámetros por default previamente definidos, la función seguirá ejecutándose con normalidad.

```
def suma (a, b=8):

return a + b

result = suma (5)

# result = 13
```

Ejemplo de un programa en Python usando funciones y ciclos:

```
# Definición de una función para calcular el factorial de un número

def calcular_factorial(numero):

   factorial = 1 # Inicializamos el factorial en 1

   while numero > 1:

      factorial *= numero # Multiplicamos el factorial por el número actual

      numero -= 1 # Decrementamos el número en 1 en cada iteración
```

```
    return factorial # Devolvemos el resultado


# Función principal del programa
def main ():
    # Solicitar al usuario que ingrese un número
    numero = int (input ("Ingrese un número para calcular su factorial: "))


    # Verificar si el número es negativo
    if numero < 0:
        print ("No se puede calcular el factorial de un número negativo.")
    else:
        # Llamar a la función calcular_factorial y mostrar el resultado
        resultado = calcular_factorial(numero)
        print (f"El factorial de {numero} es {resultado}")


# Llamar a la función principal para ejecutar el programa
if __name__ == "__main__":
    main()
```

RESULTADOS Y CONCLUSIONES

Aquí pudimos ver la importancia de utilizar las funciones ya que un punto importante a destacar de las funciones es que nos sirven para ejecutar determinada arte del programa sin necesidad de que se ejecute todo lo que se encuentra dentro de él, así mismo como pudimos ver, realmente no hay mucha diferencia en cuanto a la sintaxis de las funciones, esto en comparación con las funciones en c, que son muy similares a las de Python. Pudimos ver la importancia que tienen también los ciclos a la hora de realizar una función, y a que es algo prácticamente fundamental a la hora de realizarlos, y esto nos ayuda a que nuestro programa está más definido en cuanto a la sintaxis ya

que se aprecia de una manera más clara y a su vez nos ayuda a que el programa sea más eficiente.