

**UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL**

**FACULTAD DE ECONOMÍA Y EMPRESAS**

**CONTABILIDAD Y AUDITORÍA**

**AUDITORÍA DE SISTEMA Y BASES DE DATOS**

**DESARROLLO DE EJERCICIOS BASICOS DE PROGRAMACION EN**

**PYTHON**

**DOCENTE:**

**ING FABIAN DELGADO LOOR**

**INTEGRANTES**

❖ **VILLACIS CASTILLO LISBETH ALELI**

❖ **ANTHONY GREGORY ARELLANO ALVARADO**

❖ **DEL VALLE MONTALVAN ERIKA STEFANIA**

# Ejercicios Básicos de Programación en Python

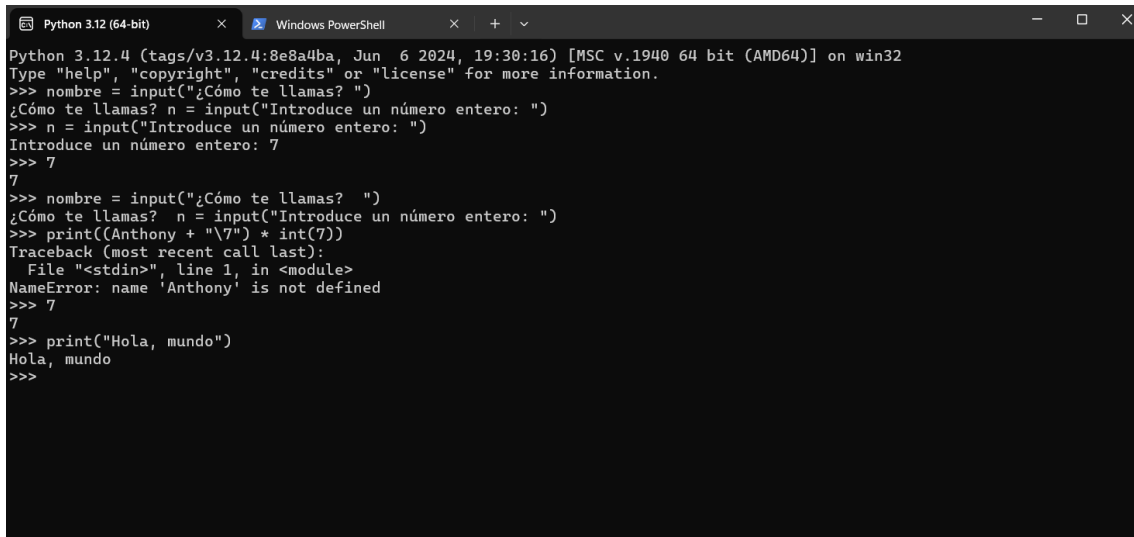
## 1. Programación básica de ingreso de información

Con la finalidad de entender los códigos de programación, a continuación se abordará un ejercicio básico y simple:

Código de programación:

```
print("Hola, mundo")
```

Resultado del código ejecutado en Python:



```
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> nombre = input("¿Cómo te llamas? ")
¿Cómo te llamas? n = input("Introduce un número entero: ")
>>> n = input("Introduce un número entero: ")
Introduce un número entero: 7
>>> 7
7
>>> nombre = input("¿Cómo te llamas? ")
¿Cómo te llamas? n = input("Introduce un número entero: ")
>>> print((Anthony + "\7") * int(7))
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'Anthony' is not defined
>>> 7
7
>>> print("Hola, mundo")
Hola, mundo
>>>
```

## 2. Creación de listas y elementos de programación

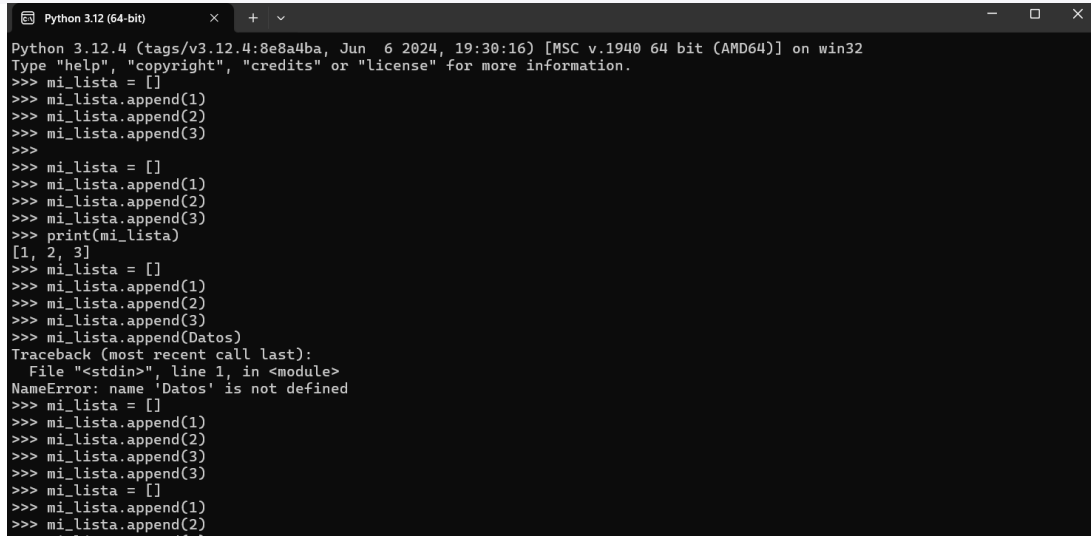
Con la finalidad de aprender a implementar listas, y agregar información a estas, a continuación se ejecutará un código de programación que las permitan estructurar y diseñar listas de datos.

Código de programación:

```
mi_lista = []
mi_lista.append(1)
mi_lista.append(2)
```

```
mi_lista.append(3)
print(mi_lista)
```

Resultado del código ejecutado en Python:



```
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> mi_lista = []
>>> mi_lista.append(1)
>>> mi_lista.append(2)
>>> mi_lista.append(3)
>>>
>>> mi_lista = []
>>> mi_lista.append(1)
>>> mi_lista.append(2)
>>> mi_lista.append(3)
>>> print(mi_lista)
[1, 2, 3]
>>> mi_lista = []
>>> mi_lista.append(1)
>>> mi_lista.append(2)
>>> mi_lista.append(3)
>>> mi_lista.append(Datos)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'Datos' is not defined
>>> mi_lista = []
>>> mi_lista.append(1)
>>> mi_lista.append(2)
>>> mi_lista.append(3)
>>> mi_lista.append(3)
>>> mi_lista = []
>>> mi_lista.append(1)
>>> mi_lista.append(2)
```

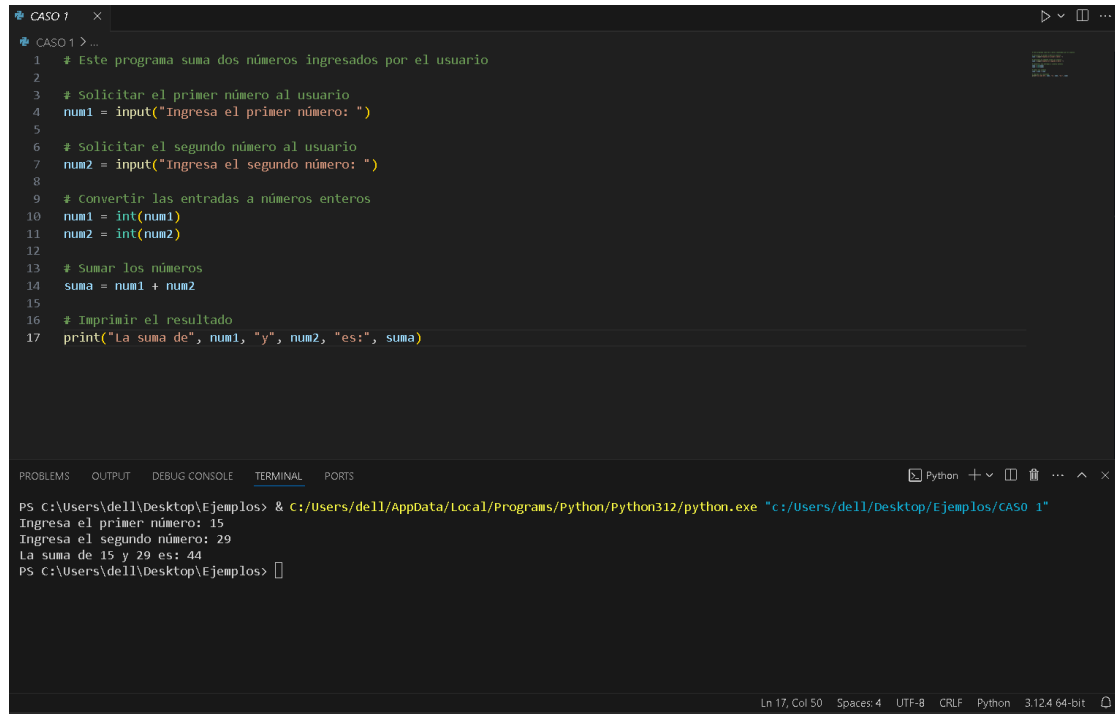
### 3. Programación de suma de números enteros

La programación de suma de números enteros es un ejercicio que permitirá entender cómo recibir entradas del usuario y hacer operaciones básicas, el código para su programación es el siguiente:

```
# Este programa suma dos números ingresados por el usuario
# Solicitar el primer número al usuario
num1 = input("Ingresa el primer número: ")
# Solicitar el segundo número al usuario
num2 = input("Ingresa el segundo número: ")
# Convertir las entradas a números enteros
num1 = int(num1)
num2 = int(num2)
# Sumar los números
suma = num1 + num2
```

```
# Imprimir el resultado
print("La suma de", num1, "y", num2, "es:", suma)
```

### **Resultado del código ejecutado en Visual Studio Code – Python:**



The screenshot shows the Visual Studio Code editor with a file named 'CASO 1'. The code is a Python script that prompts the user for two numbers, converts them to integers, and prints their sum. The terminal at the bottom shows the command to run the script and the resulting output.

```
CASO 1 > ...
1 # Este programa suma dos números ingresados por el usuario
2
3 # Solicitar el primer número al usuario
4 num1 = input("Ingresa el primer número: ")
5
6 # Solicitar el segundo número al usuario
7 num2 = input("Ingresa el segundo número: ")
8
9 # Convertir las entradas a números enteros
10 num1 = int(num1)
11 num2 = int(num2)
12
13 # Sumar los números
14 suma = num1 + num2
15
16 # Imprimir el resultado
17 print("La suma de", num1, "y", num2, "es:", suma)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + - ... ^ x

```
PS C:\Users\dell\Desktop\Ejemplos> & C:/Users/dell/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/dell/Desktop/Ejemplos/CASO 1"
Ingresa el primer número: 15
Ingresa el segundo número: 29
La suma de 15 y 29 es: 44
PS C:\Users\dell\Desktop\Ejemplos>
```

Ln 17, Col 50 Spaces: 4 UTF-8 CRLF Python 3.12.4 64-bit

## **4. Programación de un pequeño juego de adivinanzas**

El código de este juegos nos permiten comprender mejor la variedad de conceptos fundamentales de programación en Python de una manera práctica y aplicada, lo que ayuda entender mejor cómo construir y estructurar programas, así como a interactuar con el usuario y manejar el flujo del programa. El código designada para su programación es el siguiente:

```
import random
# Este programa es un juego de adivinanza de números
def juego_adivinanza():
    numero_secreto = random.randint(1, 100)
```

```

        intentos = 0
    print("Estoy pensando en un número entre 1 y 100.")
    while True:
        intento = int(input("Adivina el número: "))
        intentos += 1
        if intento < numero_secreto:
            print("Muy bajo. Intenta de nuevo.")
        elif intento > numero_secreto:
            print("Muy alto. Intenta de nuevo.")
        else:
            print(f"¡Felicidades! Adivinaste el número en {intentos} intentos.")
            break
    juego_adivinanza()

```

### Resultado del código ejecutado en Visual Studio Code – Python:

```

* CASO 1  adivinanza.py x
adivinanza.py >...
1  import random
2
3  # Este programa es un juego de adivinanza de números
4
5  def juego_adivinanza():
6      numero_secreto = random.randint(1, 100)
7      intentos = 0
8      print("Estoy pensando en un número entre 1 y 100.")
9
10     while True:
11         intento = int(input("Adivina el número: "))
12         intentos += 1
13
14         if intento < numero_secreto:
15             print("Muy bajo. Intenta de nuevo.")
16         elif intento > numero_secreto:
17             print("Muy alto. Intenta de nuevo.")
18         else:
19             print(f"¡Felicidades! Adivinaste el número en {intentos} intentos.")
20             break
21
22     juego_adivinanza()
23
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
Python
Muy alto. Intenta de nuevo.
Adivina el número: 44
Muy alto. Intenta de nuevo.
Adivina el número: 40
Muy alto. Intenta de nuevo.
Adivina el número: 38
Muy alto. Intenta de nuevo.
Adivina el número: 34
Muy bajo. Intenta de nuevo.
Adivina el número: 35
¡Felicidades! Adivinaste el número en 9 intentos.
PS C:\Users\dell\Desktop\Ejemplos>
Ln 4, Col 1  Spaces: 4  UTF-8  CRLF  Python  3.12.4 64-bit

```