



Fundamentos de programación en Python

Carrera de Administración de Redes y Comunicaciones

■ Alianzas Estratégicas ■





Google Classroom

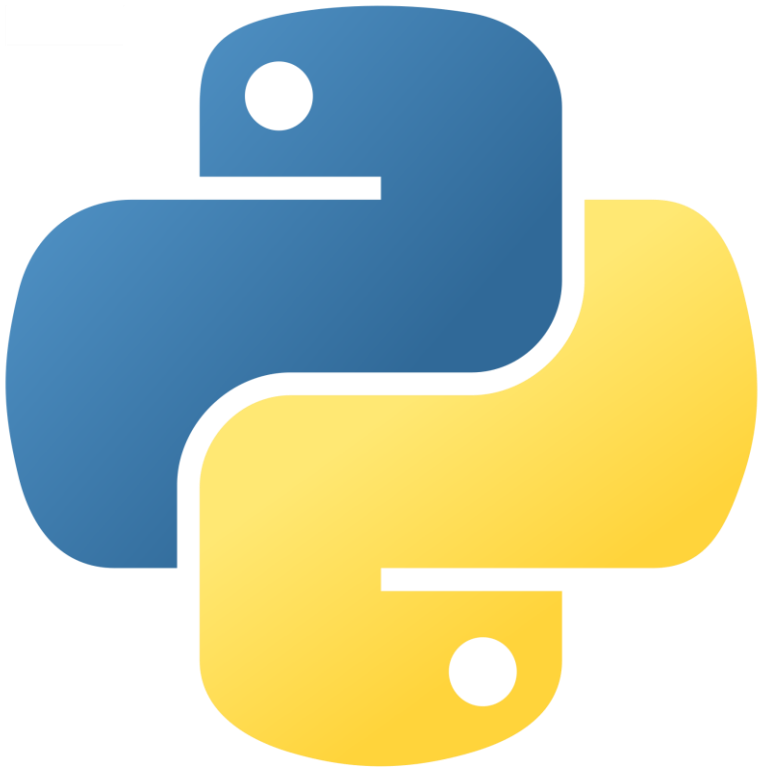


Q10
Académico

Fundamen
programación en Python.



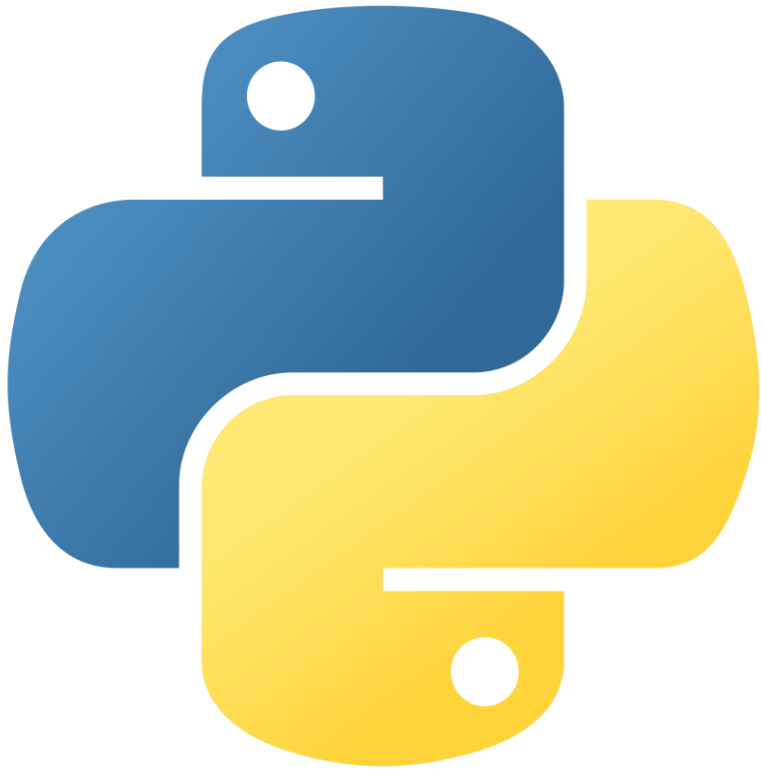
Sobre el curso



El curso está orientado a la enseñanza de fundamentos de Programación en el lenguaje Python v 3.0., bajo la modalidad definida por la Academia de Programming de Cisco. Esta modalidad consiste en el desarrollo de la currícula PCAP dividida en 8 módulos.

Durante el curso se desarrollarán actividades de revisión del contenido académico y laboratorios prácticos utilizando el emulador de Python que nos provee el mismo Cisco (SandBox) o de algún otro IDE para escritorio, así como el desarrollo de exámenes y actividades orientadas al entendimiento de estos conceptos de acuerdo con los temas que considera Cisco en este curso.

Semana 1



- Introducción a Python
- Instalación
- Códigos de Python
- Manejo de Variables
- Uso de operadores aritméticos
- Estructuras Secuenciales
- Evaluación Continua 1

Introducción a Python



Python es un lenguaje de programación de alto nivel, interpretado y de propósito general. Es conocido por su sintaxis clara y legible, lo que lo hace ideal para principiantes, pero también es poderoso y versátil, lo que lo convierte en una opción entre programadores experimentados.

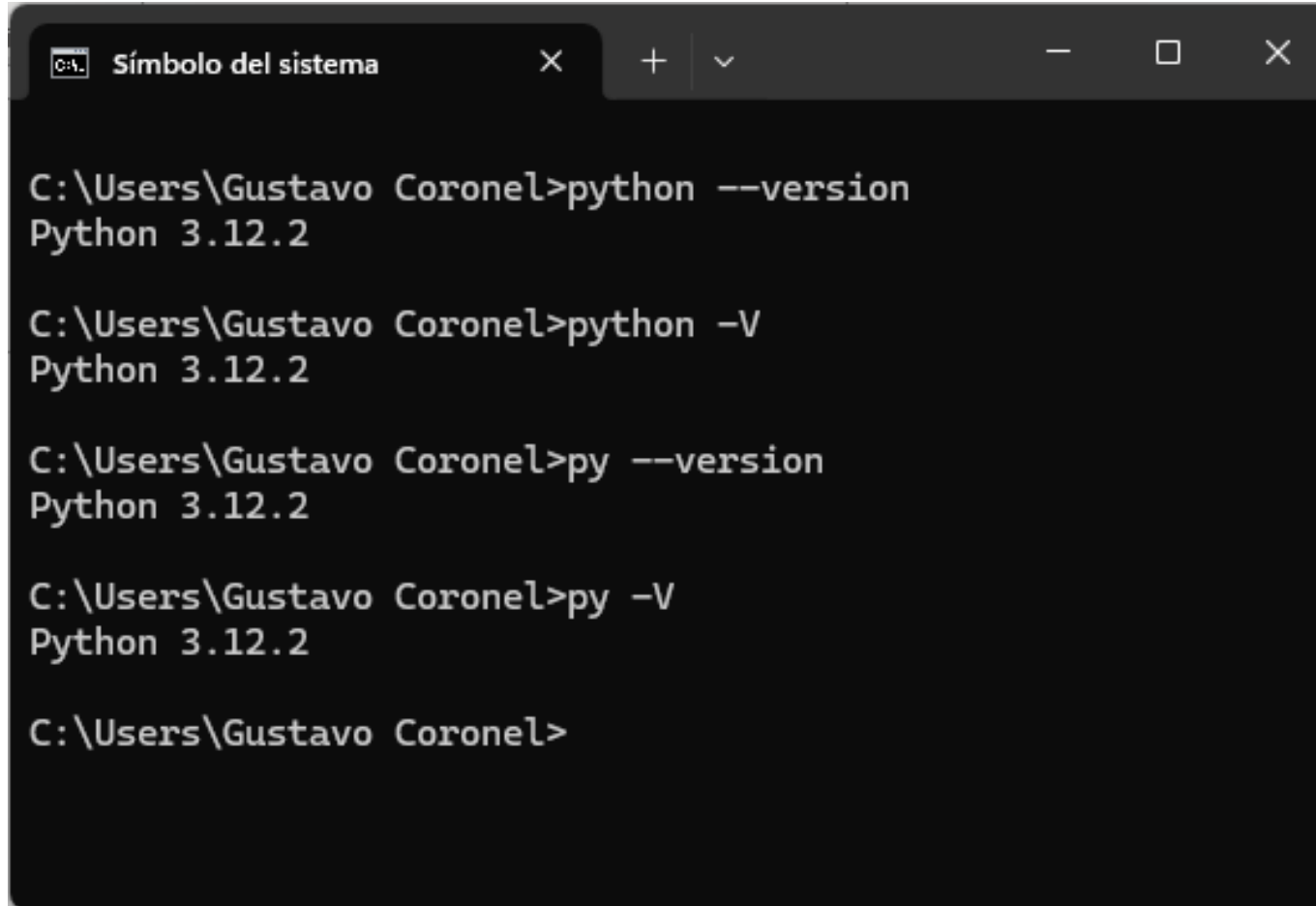
Python es ampliamente utilizado en una variedad de campos, como desarrollo web, ciencia de datos, inteligencia artificial, entre otros.

Una de las razones de su popularidad es su amplia gama de bibliotecas y frameworks, que permiten a los desarrolladores realizar una variedad de tareas sin necesidad de escribir mucho código desde cero.

Instalación



Instalación



```
C:\Users\Gustavo Coronel>python --version
Python 3.12.2

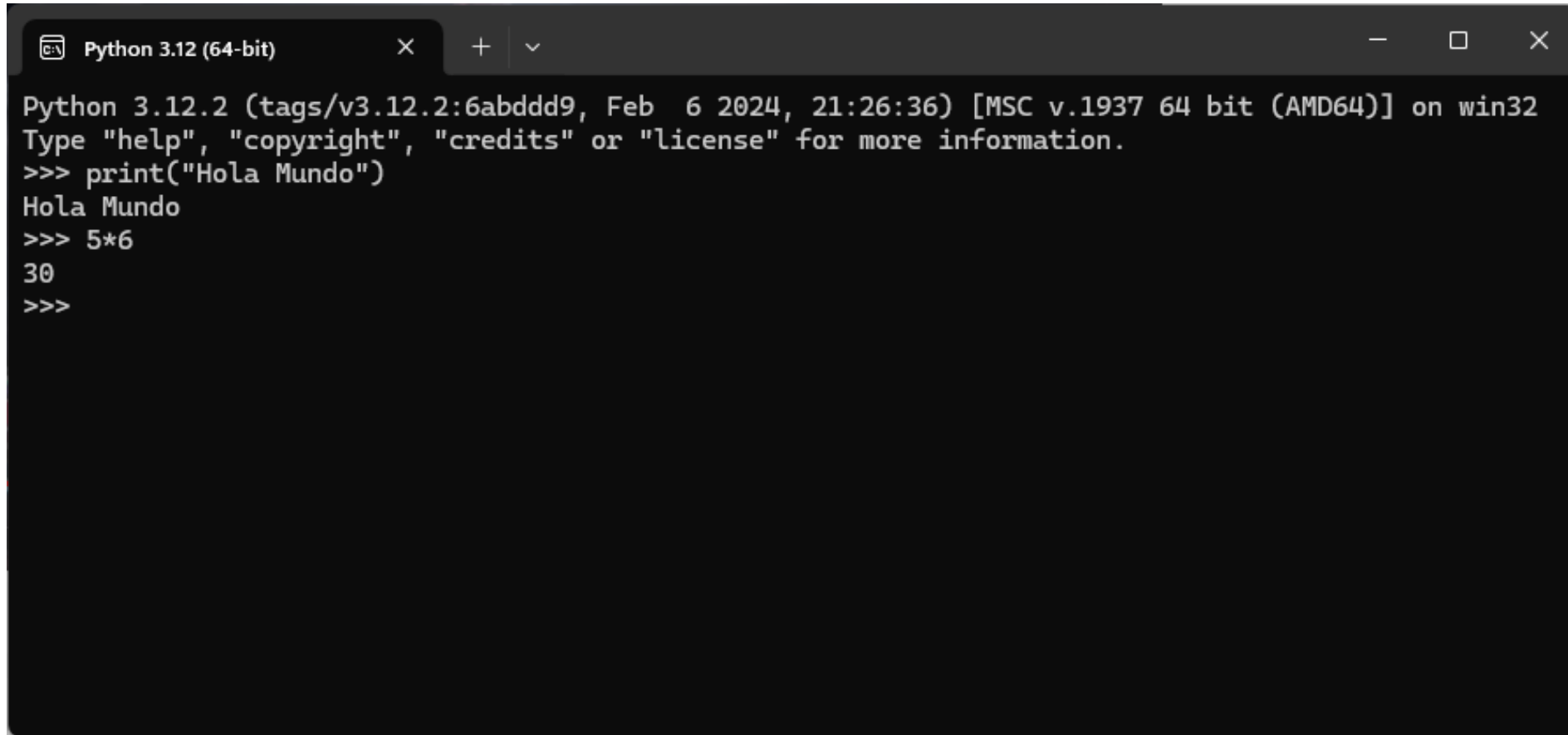
C:\Users\Gustavo Coronel>python -V
Python 3.12.2

C:\Users\Gustavo Coronel>py --version
Python 3.12.2

C:\Users\Gustavo Coronel>py -V
Python 3.12.2

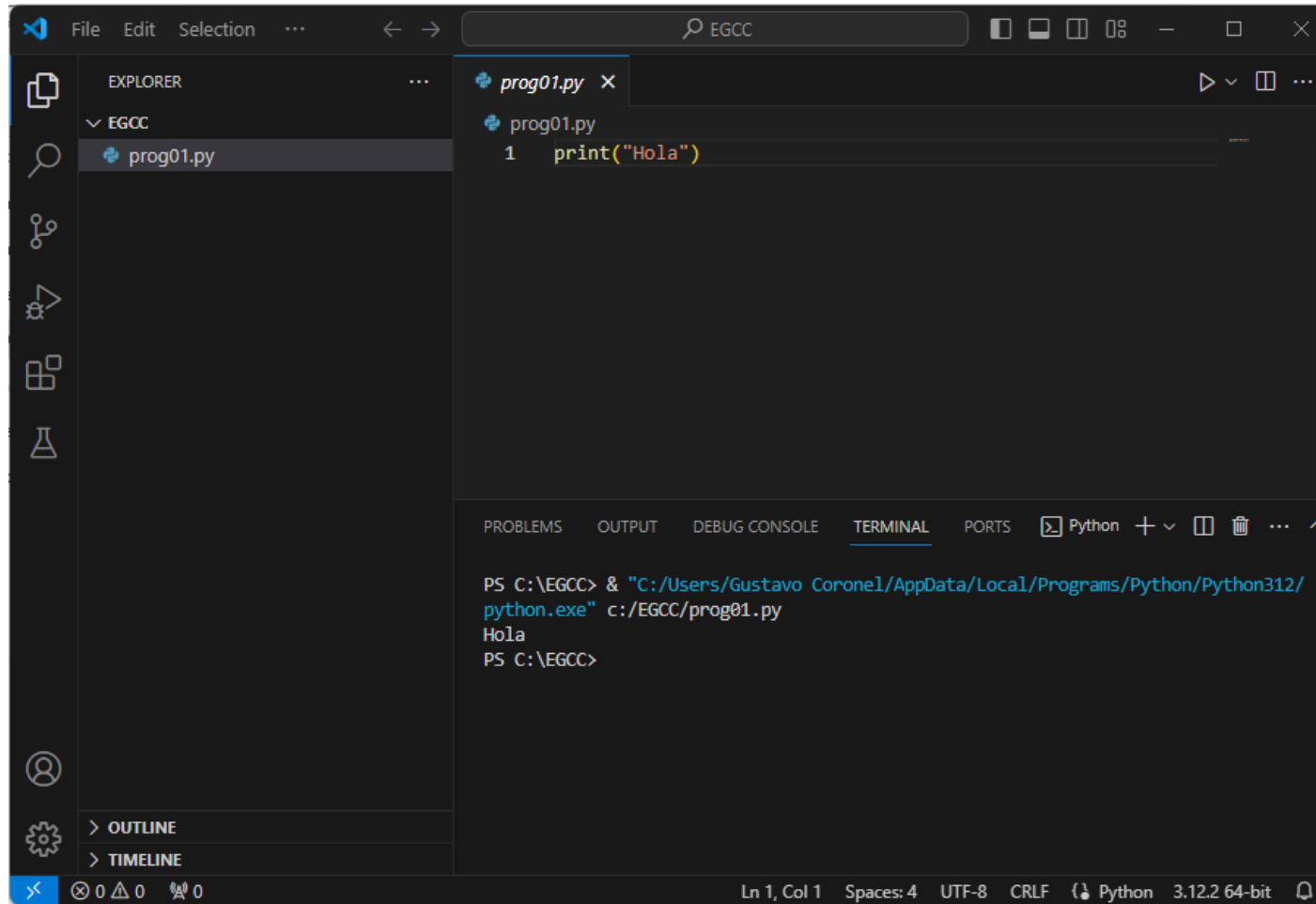
C:\Users\Gustavo Coronel>
```


Instalación



```
Python 3.12 (64-bit) × + ▾  
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb 6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license" for more information.  
>>> print("Hola Mundo")  
Hola Mundo  
>>> 5*6  
30  
>>>
```

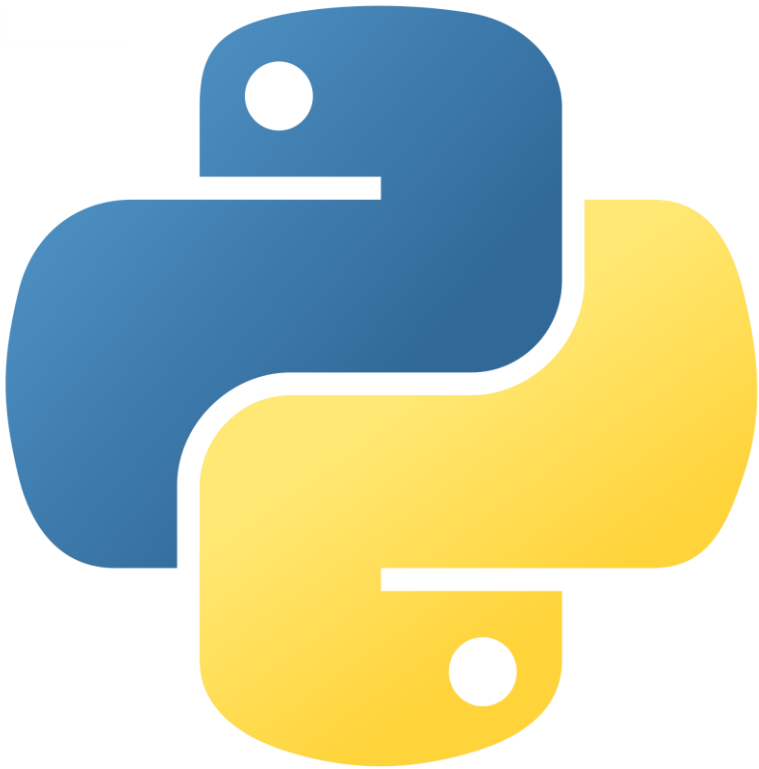
Instalación



The screenshot shows the Visual Studio Code interface with a dark theme. The Explorer panel on the left shows a folder named 'EGCC' containing a file 'prog01.py'. The main editor area shows the content of 'prog01.py', which is a single line of Python code: `print("Hola")`. The TERMINAL panel at the bottom shows the command prompt output: `PS C:\EGCC> & "C:/Users/Gustavo Corone1/AppData/Local/Programs/Python/Python312/python.exe" c:/EGCC/prog01.py`, followed by the output `Hola` and the prompt `PS C:\EGCC>`. The status bar at the bottom indicates the file is at line 1, column 1, with 4 spaces, using UTF-8 encoding, CRLF line endings, and is a Python 3.12.2 64-bit file.

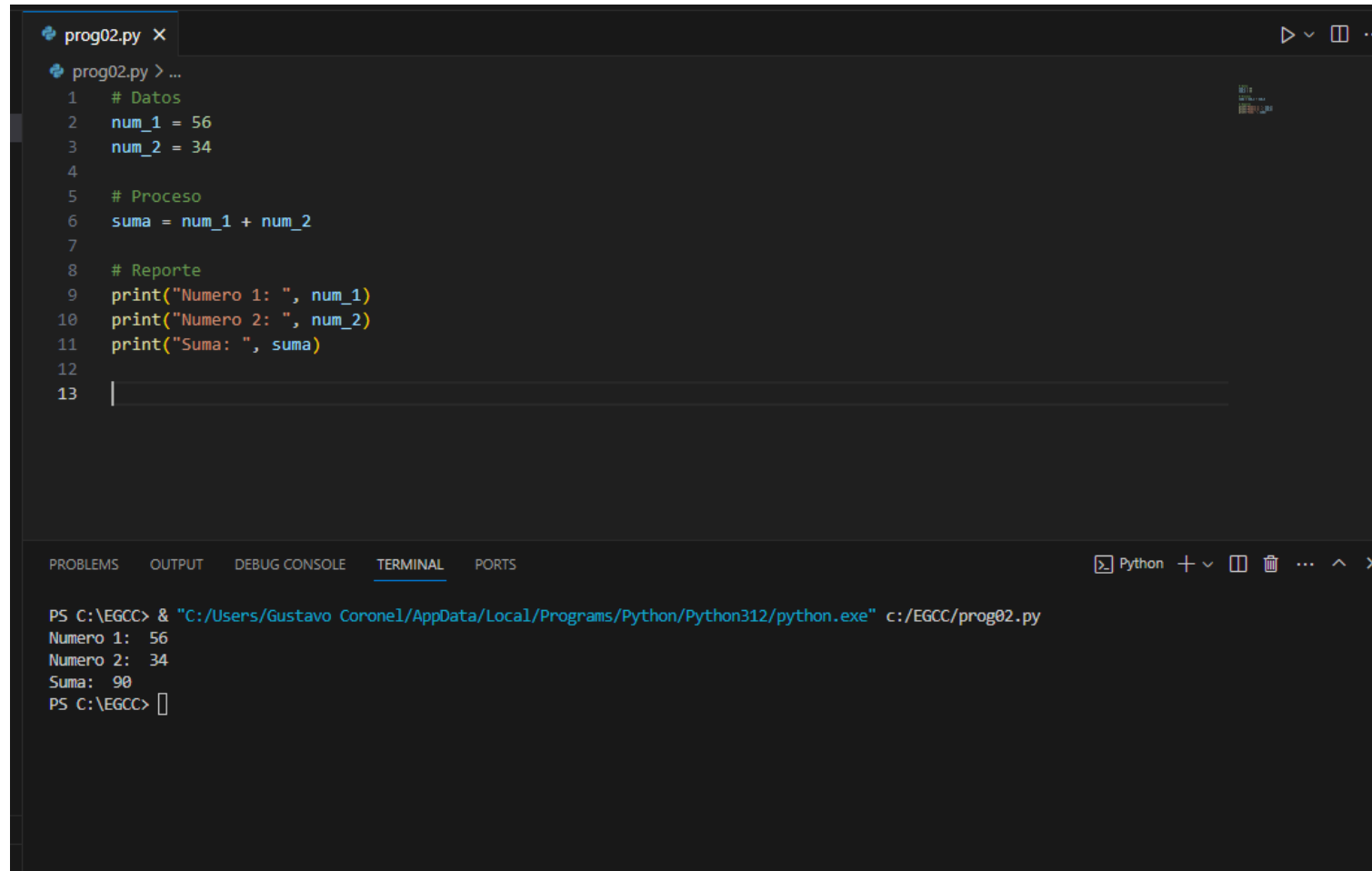
```
File Edit Selection ... < > EGCC
EXPLORER
  EGCC
    prog01.py
prog01.py
1  print("Hola")
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + - Python 3.12.2 64-bit
PS C:\EGCC> & "C:/Users/Gustavo Corone1/AppData/Local/Programs/Python/Python312/
python.exe" c:/EGCC/prog01.py
Hola
PS C:\EGCC>
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python 3.12.2 64-bit
```

Codificación en Python



- Utilizar la nomenclatura de **snake case** para nombrar variables.
- Utilizar palabras en mayúsculas para las constantes.

Codificación en Python



```
prog02.py X
prog02.py > ...
1  # Datos
2  num_1 = 56
3  num_2 = 34
4
5  # Proceso
6  suma = num_1 + num_2
7
8  # Reporte
9  print("Numero 1: ", num_1)
10 print("Numero 2: ", num_2)
11 print("Suma: ", suma)
12
13 |
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + - - - ^ X

```
PS C:\EGCC> & "C:/Users/Gustavo Corone1/AppData/Local/Programs/Python/Python312/python.exe" c:/EGCC/prog02.py
Numero 1: 56
Numero 2: 34
Suma: 90
PS C:\EGCC> |
```



Manejo de variables

Declaración de variables: En Python, no es necesario declarar explícitamente el tipo de una variable antes de usarla. Simplemente asigna un valor a una variable y Python determinará automáticamente el tipo de datos.

```
x = 5 # x es un entero
nombre = "Juan" # nombre es una cadena de texto
precio = 10.99 # precio es un número de punto flotante

print("x es de tipo: ", type(x))
print("nombre es de tipo: ", type(nombre))
print("precio es de tipo: ", type(precio))
```

Manejo de variables

Nombres de variables: Los nombres de variables en Python pueden contener letras, números y guiones bajos, pero deben comenzar con una letra o un guion bajo.

Además, los nombres de variables distinguen entre mayúsculas y minúsculas.

```
# Datos
print("Ingresa un numero:")
numero1 = int(input())
numero2 = int(input("Ingresa otro numero:"))

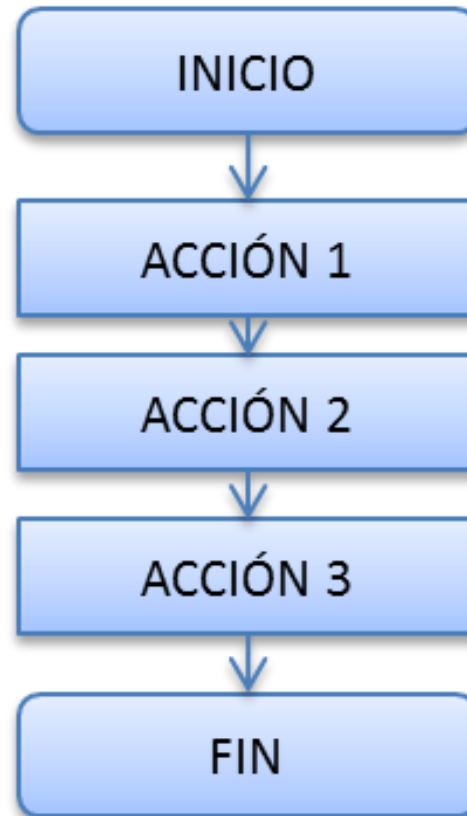
# Proceso
producto = numero1 * numero2

# Reporte
print("Producto: ", producto)
```


Operadores aritméticos

Operador	Descripción	Ejemplo	Resultado
+	Suma	$c = 3 + 5$	$c = 8$
-	Resta	$c = 4 - 2$	$c = 2$
-	Negación	$c = -7$	$c = -7$
*	Multiplicación	$c = 3 * 6$	$c = 18$
**	Potenciación	$c = 2 ** 3$	$c = 8$
/	División	$c = 7.5 / 2$	$c = 3.75$
//	División entera	$c = 7.5 // 2$	$c = 3.0$
%	Módulo	$c = 8 \% 3$	$c = 2$

Estructuras secuenciales



Estructuras secuenciales



GRACIAS
TOTALES