



Python: Estructuras Secuenciales

Centro de Servicios y Gestión Empresarial
SENA Regional Antioquia



www.sena.edu.co

Conceptualización

¿Qué es Python?

Python es un lenguaje de programación popular. Fue creado por Guido van Rossum y lanzado en 1991. Se utiliza para:

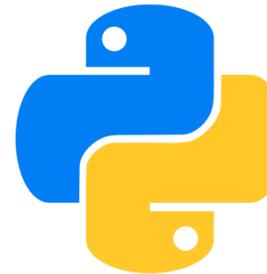
- Desarrollo web (lado del servidor)
- Desarrollo de software
- Ciencia de datos, matemáticas
- Secuencias de comandos del sistema.



¿Qué puede hacer Python?

- Python se puede usar en un servidor para crear aplicaciones web.
- Python se puede usar junto con el software para crear flujos de trabajo.
- Python puede conectarse a sistemas de bases de datos. También puede leer y modificar archivos.
- Python se puede usar para manejar big data y realizar operaciones matemáticas complejas.
- Python se puede utilizar para la creación rápida de prototipos o para el desarrollo de software listo para la producción.

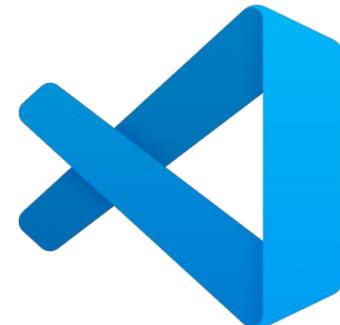
¿Cómo utilizar Python?



PYTHON

Python 3.13.2

<https://www.python.org/>

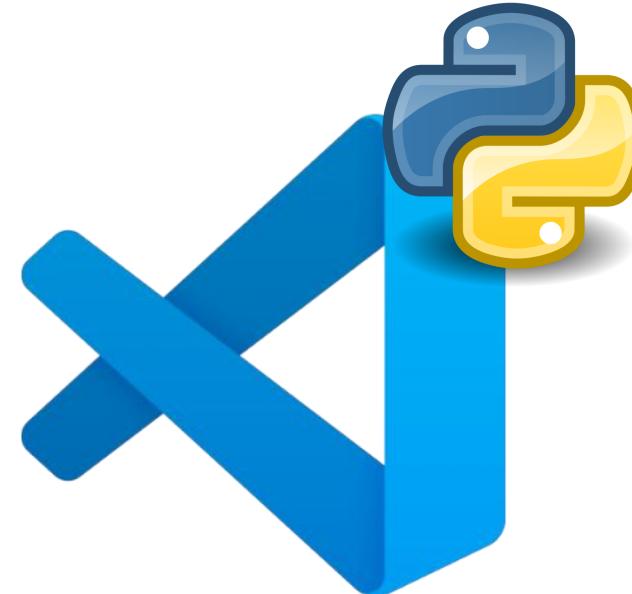


Visual Studio Code

<https://code.visualstudio.com/>

Extensiones

- Python
- Error Lens
- **Code Runner**
 - Archivo – Preferencias – Configuración – Run in Terminal



Gestión de entornos de desarrollo en Python



Entornos de desarrollo en Python



Los entornos virtuales permiten aislar dependencias de proyectos en Python, evitando conflictos entre versiones de librerías.

1. Crear un entorno virtual

Ejecuta este comando en la terminal o línea de comandos:

```
python -m venv mi_entorno
```

💡 Esto crea una carpeta llamada **mi_entorno** que contiene el entorno virtual.

Entornos de desarrollo en Python



2. Activar el entorno virtual

Use los siguientes comandos en Windows (CMD o PowerShell)

```
mi_entorno\Scripts\activate
```

💡 Cuando el entorno está activo, el nombre del entorno aparece en el prompt de la terminal, por ejemplo:

```
(mi_entorno) usuario@pc:~
```

Entornos de desarrollo en Python



3. Desactivar el entorno virtual

Para salir del entorno, simplemente ejecuta:

```
deactivate
```

💡 Esto nos devuelve al entorno global de Python.

4. Eliminar un entorno virtual

Si desea borrar un entorno, simplemente elimina la carpeta correspondiente:

```
rmdir /s /q mi_entorno
```

Entornos de desarrollo en Python



5. Instalar paquetes en el entorno virtual

Con el entorno activado, usa **pip** para instalar paquetes sin afectar el sistema global:

```
pip install nombre_del_paquete
```

💡 Puedes ver los paquetes instalados con:

```
pip list
```

Comentarios



Los comentarios en Python se usan para explicar el código y facilitar su comprensión. No afectan la ejecución del programa, ya que Python los ignora.

1. Comentarios de una línea (#)

Para escribir un comentario en una sola línea, usa el símbolo #:

```
# Esto es un comentario en una línea
print("Hola, mundo") # Comentario al final de una línea de código
```

Comentarios



2. Comentarios de varias líneas (Docstrings """ """" o ''' ''')

Si es requerido escribir un comentario que abarque varias líneas, usa triple comillas (""" o '''):

```
"""
Este es un comentario de varias líneas.
Se usa para documentar partes del código.
Python lo ignora en la ejecución.
"""

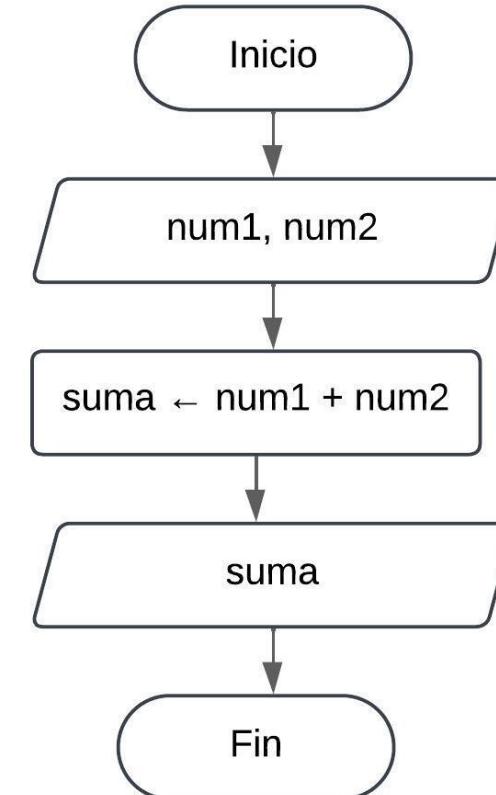
print("Hola, mundo")
```

Estructuras Secuenciales

Estructuras Secuenciales

Una estructura secuencial es el tipo más básico de flujo de control en un programa.

En este tipo de estructura, las instrucciones se ejecutan una tras otra en el mismo orden en que están escritas.



Variables



Las variables en Python se usan para almacenar datos y pueden cambiar de valor. No necesitan declaración previa y se crean al asignarles un valor.

Ejemplo:

```
# Ejemplo de Variables
nombre = "Juan"
edad = 25
pi = 3.1416
estudiante = True
```

Constantes



Las constantes en Python no existen como tal, pero por convención, se escriben en mayúsculas para indicar que su valor no debe cambiar.

Ejemplo:

```
# Ejemplo de Constantes
PI = 3.1416
GRAVEDAD = 9.81
```

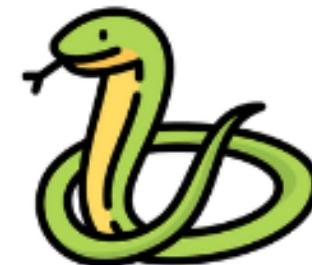
Convenciones de nomenclatura



camelCase



kebab-case



snake_case



PascalCase

Convenciones de nomenclatura



En Python se aplica la siguiente convención de nomenclatura:

- 🚀 Snake case para **variables** y **funciones**:
nombre_variable
- 🚀 Pascal case para **clases**: NombreClase
- 🚀 Mayúsculas para **constants**: CONSTANTE
- 🚀 Evitar **palabras reservadas** (def, class, if, etc.).

Tipos de Datos



Python tiene varios tipos de datos básicos. Para saber el tipo de una variable se usa **type()**:

```
# Ejemplo de Tipos de Datos
entero = 10          # int
print(type(entero))  # <class 'int'>

flotante = 3.14       # float
print(type(flotante)) # <class 'float'>

cadena = "Hola"       # str
print(type(cadena))  # <class 'str'>

booleano = True        # bool
print(type(booleano)) # <class 'bool'>
```

Conversión de tipos de datos



```
# Convertir de entero a flotante
num = 10
num_flotante = float(num) # 10.0

# Convertir de flotante a entero
num_entero = int(num_flotante) # 10

# Convertir de número a cadena
num_cadena = str(num) # "10"

# Convertir de cadena a número
cadena_numero = "15"
numero = int(cadena_numero) # 15

# Convertir de booleano a entero
verdadero = int(True) # 1
falso = int(False) # 0
```

Python permite la conversión entre tipos de datos:

Operadores aritméticos

Operador	Descripción	Ejemplo
+	Suma	$5 + 3 \rightarrow 8$
-	Resta	$10 - 4 \rightarrow 6$
*	Multiplicación	$6 * 7 \rightarrow 42$
/	División	$10 / 2 \rightarrow 5.0$ (Siempre da flotante)
//	División entera	$10 // 3 \rightarrow 3$ (Descarta decimales)
%	Módulo (resto)	$10 \% 3 \rightarrow 1$
**	Potencia	$2 ** 3 \rightarrow 8$

Operadores aritméticos



```
# Ejemplo de Operandos Aritméticos
numero_uno = 10
numero_dos = 5

print("Suma: ", numero_uno + numero_dos)
print("Resta: ", numero_uno - numero_dos)
print("Multiplicación: ", numero_uno * numero_dos)
print("División: ", numero_uno / numero_dos)
print("División Entera: ", numero_uno // numero_dos)
print("Módulo: ", numero_uno % numero_dos)
print("Potencia: ", numero_uno ** numero_dos)
```

Entradas y Salidas



El usuario puede ingresar datos mediante **input()**, pero el valor recibido es siempre una cadena (str) y la función **print()** permite mostrar información en la consola.

```
# Ejemplo de Input y Output
nombre = input("¿Cuál es tu nombre? ")
edad = int(input("¿Cuántos años tienes? ")) # Convertir a entero

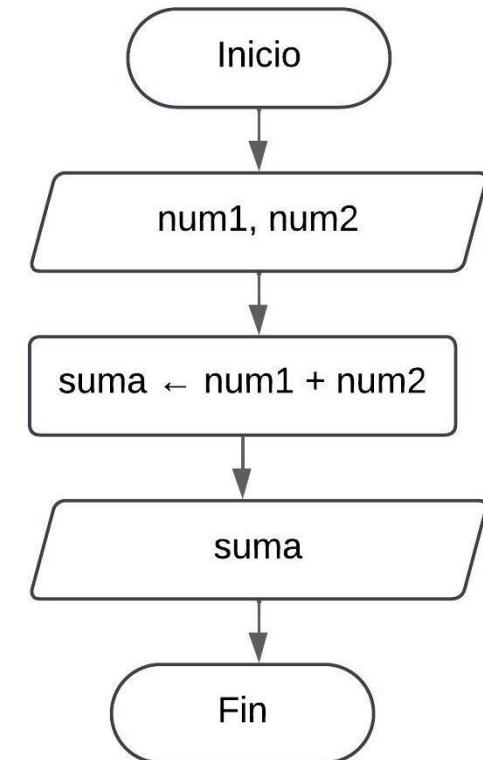
print("Hola", nombre, ", tienes", edad, "años.") # Concatenación
print("Hola " + nombre + ", tienes " + str(edad) + " años.") # Concatenación
print("Hola {}, tienes {} años.".format(nombre, edad)) # Formateo
print(f"Hola {nombre}, tienes {edad} años.") # Formateo
```

Ejemplo Integrador

Algoritmo

Ejemplo: Generar un algoritmo que permita sumar dos números

```
Inicio
    Definir num1, num2, suma Como Entero
    Escribir "Ingrese el primer número"
    Leer num1
    Escribir "Ingrese el segundo número"
    Leer num2
    suma ← num1 + num2
    Escribir "La suma es:", suma
Fin
```



Entrada	num1	num2	Suma
Caso 1	5	3	8
Caso 2	10	-2	8
Caso 3	0	0	0

Python



```
# Estructuras Secuenciales  
# Autor: Alvaro Perez N  
# Fecha: 18/02/2025  
# Versión: 1.0.0
```

```
"""
```

Ejercicio: Generar un algoritmo que permita sumar dos números enteros y muestre el resultado en pantalla.

```
"""
```

```
# Ingreso de datos  
print("Suma de dos números enteros")  
numero_uno = int(input("Ingrese el primer número: "))  
numero_dos = int(input("Ingrese el segundo número: "))  
# Proceso  
suma = numero_uno + numero_dos  
# Salida  
print(f"La suma de los números {numero_uno} y {numero_dos} es: {suma}")
```

Ejercicios de Practica

- 🚀 Un restaurante desea calcular el total a pagar por un cliente después de aplicar el IVA (19%) y una propina del 10%. El usuario ingresará el precio de su consumo, y el programa mostrará el total final.
- 🚀 Una compañía de telefonía cobra las llamadas en función de su duración en minutos. El costo por minuto es \$0.50 USD y se requiere calcular el total a pagar por una llamada dada su duración en minutos.
- 🚀 Un profesor universitario desea calcular el promedio de calificaciones de un estudiantes. Teniendo en cuenta que en semestre se realizaron 05 actividades calificables.



G R A C I A S

Presentó: Alvaro Pérez Niño

Instructor Técnico

Correo: aperezn@sena.edu.co

<http://centrodesserviciosygestionempresarial.blogspot.com/>

Línea de atención al ciudadano: 01 8000 910270

Línea de atención al empresario: 01 8000 910682



www.sena.edu.co