

Lectura Sesión Introducción al Lenguaje Python y Sentencias Básicas

Python es un lenguaje de programación. Pero... ¿qué es un lenguaje de programación?

 **por Kibernet Capacitación S.A.**

¿Qué es Python? (Reseña del lenguaje)

Es una forma de hablarle a la computadora para que haga lo que tú quieres:

- Mostrar mensajes en pantalla
- Hacer cálculos
- Guardar datos
- Automatizar tareas repetitivas
- Crear páginas web, apps, juegos, análisis de datos y mucho más

Python fue creado por Guido van Rossum a finales de los años 80 y su primera versión se lanzó en 1991.

Aunque muchos creen que su nombre viene de una serpiente, en realidad fue inspirado por el programa de comedia británico "Monty Python's Flying Circus", que su autor admiraba.

Van Rossum eligió ese nombre para reflejar un lenguaje divertido, accesible y no intimidante.

El logo con la serpiente fue adoptado después, pero no tiene relación con el origen del nombre.

La filosofía de Python es hacer que programar sea simple, legible y agradable para todos.

Empresas que usan Python

Es un lenguaje usado por:



Google

Utiliza Python en muchos de sus servicios y aplicaciones internas



Netflix

Emplea Python para su plataforma de streaming y análisis de datos



Facebook

Usa Python para gestionar su infraestructura y servicios



NASA

Aplica Python en cálculos científicos y control de misiones



¿Cuál es el propósito de Python?

Python fue diseñado con un propósito claro:

"Hacer que programar sea fácil, legible y accesible para todos."

Eso significa que:

- Puedes leer un programa en Python casi como si leyeras una receta de cocina.
- No necesitas saber cosas súper técnicas para empezar.
- Sirve para muchos usos distintos (es multipropósito).



Ejemplos del mundo real:

- Un robot en una fábrica podría estar programado en Python.
- Un bot que te responde en WhatsApp, puede usar Python.
- Un programa que analiza datos del clima, también puede estar hecho en Python.

Principales características del lenguaje Python

1 Fácil de aprender

No necesitas poner llaves {}, ni punto y coma ;, ni cosas complicadas para que funcione.

```
print("Hola mundo") # Esto ya imprime algo. Así de simple.
```

2 Código limpio y legible

Usa espacios y sangrías en lugar de símbolos raros. Esto obliga a escribir código ordenado, como una lista clara de instrucciones.

```
if edad > 18:
    print("Puedes votar")
    ....
```

3 Tiene muchas librerías

Las librerías son como cajas de herramientas que otros ya crearon para ti. ¿Quieres hacer gráficos? Hay una librería. ¿Quieres conectarte a una base de datos? También hay una. ¿Quieres hacer inteligencia artificial? Hay muchas.

Ejemplos: pandas, numpy, matplotlib, tensorflow, etc.



4 Funciona en todos los sistemas

Puedes usar Python en: Windows, Mac, Linux, Incluso en la nube (sin instalar nada)

5 Tiene una comunidad enorme

Hay millones de personas que ya usan Python, y si tienes dudas, probablemente ya alguien preguntó lo mismo y hay una respuesta en internet.

Versiones de Python (¿Y cuál debo usar?)

Python tiene muchas versiones.

Las más importantes son:

Python 2

- Es una versión antigua.
- Ya no se usa ni se actualiza.
- Mejor evitarla.

Python 3

- Es la versión actual.
- Se sigue actualizando.
- Es la que debes usar siempre.

La versión más usada es Python 3.10 o superior.

¿Dónde descargarlo? Desde el sitio oficial: <https://www.python.org/downloads/>

IMPORTANTE: En Windows, cuando instales Python, marca la casilla que dice "Add Python to PATH". Así podrás usarlo desde cualquier parte del sistema.

Entornos de trabajo y herramientas para escribir código en Python

Un entorno de trabajo es el lugar donde vas a escribir y probar tu código. Como cuando escribes una carta en Word, o haces cálculos en Excel.

Aquí te explico los más comunes:

Anaconda

Es como un "todo en uno". Incluye Python, Jupyter Notebooks, librerías científicas y más.

Ideal para:

- Ciencia de datos
- Machine learning
- Cursos donde se usen gráficos y análisis

Descargar desde:

<https://www.anaconda.com/products/distribution>

Jupyter Notebooks

Es como un cuaderno digital interactivo donde puedes escribir código + texto explicativo.

Ideal para:

- Aprender paso a paso
- Mostrar resultados justo debajo del código
- Visualizar datos

Puedes usarlo desde Anaconda o directamente online en Colab (más abajo).

Más entornos de desarrollo para Python

Google Colab

Es un Jupyter Notebook pero en internet. No necesitas instalar nada.

Ideal para:

- Principiantes que no quieren instalar nada
- Clases en línea
- Personas que trabajan desde distintos computadores

Entra gratis desde:
<https://colab.research.google.com/>

Visual Studio Code (VS Code)

Es un editor de código liviano y muy potente. Sirve para muchos lenguajes, no solo Python.

Ideal para:

- Proyectos más grandes
- Aprender a programar de forma más profesional

Necesitas instalar Python y una extensión para que funcione.

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

Spyder

Es un entorno incluido en Anaconda. Es como un "Word para Python".

Ideal para:

- Personas que vienen de usar MATLAB o R
- Proyectos científicos o numéricos

Viene en Anaconda:
<https://www.anaconda.com/products/distribution>

¿Qué herramienta usar para empezar?

Tu situación	Herramienta recomendada
No quieres instalar nada	Google Colab → https://colab.research.google.com
Quieres tener todo listo con un clic	Anaconda → https://www.anaconda.com/products/distribution + Jupyter Notebook → https://jupyter.org
Quieres escribir código como un pro	Visual Studio Code (VS Code) → https://code.visualstudio.com
Quieres algo visual como Excel/Word	Spyder → https://www.spyder-ide.org o Jupyter Notebook → https://jupyter.org

¿Cómo saber si ya tienes Python?

Abre una terminal o consola y escribe:

```
python --version
```

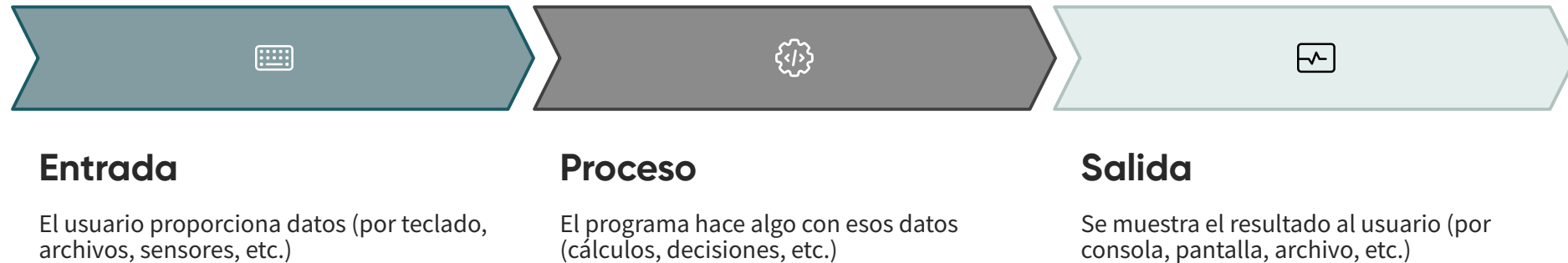
Si ves algo como Python 3.10.11, ¡estás listo para comenzar!

Sentencias Básicas del Lenguaje Python

¿Qué es programar?

Programar es dar instrucciones a una computadora para que realice tareas. Estas instrucciones se escriben en un lenguaje que la máquina pueda entender, como Python.

Proceso básico de un programa:



Variables y Tipos de Datos Fundamentales

¿Qué es una variable?

Una variable es un espacio en la memoria que sirve para guardar información que usaremos en el programa. Es como una caja con nombre donde puedes guardar datos.

```
nombre = "Sofía"
edad = 25
```

¿Qué tipos de datos existen?

Python maneja distintos tipos de datos según lo que necesitas guardar:

Tipo de dato	Ejemplo	Uso común
Entero (int)	30	Contar, edades, cantidades
Decimal (float)	3.14	Medidas, porcentajes, precios
Cadena (str)	"Hola"	Nombres, mensajes, palabras
Booleano (bool)	True o False	Para decisiones lógicas (sí o no)
Complejo (complex)	2 + 3j	Matemática avanzada, ingeniería

```
# Ejemplos de cada tipo
entero = 10
decimal = 3.14
texto = "Hola mundo"
es_estudiante = True
numero_complejo = 2 + 5j
```

Operaciones en Python

Conversiones de Tipo

¿Qué es una conversión de tipo?

Es el proceso de transformar un dato de un tipo a otro. Por ejemplo, transformar un número a texto o viceversa.

¿Por qué es útil?

- Para mostrar datos de forma legible.
- Para realizar operaciones cuando los tipos no coinciden.
- Para leer entradas de usuario, que siempre vienen como texto.

```
# De string a entero
edad_texto = "30"
edad = int(edad_texto)

# De número a texto
numero = 10
texto = str(numero)

# De número a decimal
entero = 5
decimal = float(entero)
```

Operadores y Expresiones Aritméticas

¿Qué son los operadores?

Son símbolos que permiten realizar cálculos matemáticos.

- + (suma)
- - (resta)
- * (multiplicación)
- / (división)
- // (división entera)
- % (módulo o resto)
- ** (potencia)

```
a = 10
b = 3

print(a + b) # Suma → 13
print(a - b) # Resta → 7
print(a * b) # Multiplicación → 30
print(a / b) # División → 3.333...
print(a % b) # Módulo (resto) → 1
print(a ** b) # Potencia → 1000
print(a // b) # División entera → 3
```

Control de Flujo en Python

Operadores y Expresiones Lógicas

¿Qué es una expresión lógica?

Una expresión lógica evalúa condiciones que pueden ser verdaderas o falsas (True o False).

```
x = 5
y = 10

print(x > y)    # False
print(x < y)    # True
print(x == 5)   # True
```

Operadores lógicos:

- and: las dos condiciones deben ser verdaderas
- or: al menos una debe ser verdadera
- not: niega la condición

```
print(x > 3 and y < 15) # True
print(x < 3 or y > 9)   # True
print(not(x == 5))      # False
```

Sentencias Condicionales (if, elif, else)

¿Qué es el control de flujo?

Es la capacidad del programa de tomar decisiones y ejecutar distintos caminos según condiciones.

¿Qué es una bifurcación?

Es cuando el código elige entre varias opciones según una condición.

```
edad = 20

if edad < 18:
    print("Eres menor de edad")
elif edad == 18:
    print("Tienes justo 18 años")
else:
    print("Eres mayor de edad")
```

Solo se ejecuta el bloque que cumple la condición.

Entrada y Salida en Python

Impresión en Consola

¿Para qué sirve print()?

Permite mostrar información en la pantalla.

```
nombre = "Miguel"
print("Hola", nombre)
```

Entrada de Datos en Consola

¿Cómo recibe datos el programa?

Con la función input() puedes leer datos desde el teclado.

```
nombre = input("¿Cómo te llamas? ")
edad = int(input("¿Qué edad tienes? "))

print("Hola", nombre, ", tienes", edad, "años.")
```

Todo lo que entra por input() es texto, por eso se convierte con int() o float() si quieres números.

Creación y Ejecución de Scripts en Python

¿Qué es un script?

Es un archivo de texto (.py) que contiene código Python que se puede ejecutar.

¿Cómo lo creo y ejecuto?

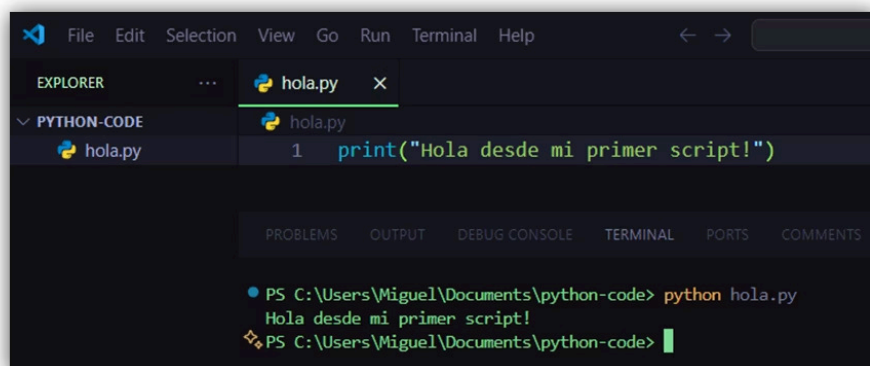
1. Abre un editor (VS Code, por ejemplo).
2. Crea un archivo llamado hola.py.
3. Escribe el siguiente código:

```
print("Hola desde mi primer script!")
```

Abre una terminal y ejecuta:

```
python hola.py
```

El script ejecuta todo lo que está escrito, línea por línea.



Actividad práctica: Tu primer programa interactivo en Python

Objetivo:

Crear un programa en Python que solicite al usuario su nombre, edad y país, y luego imprima un mensaje personalizado utilizando los datos ingresados.

Paso a paso:

Paso 1: Crear el archivo

- Abre tu editor de código favorito (VS Code o Google Colab).
- Crea un archivo llamado: `mi_primer_programa.py`

Paso 2: Escribir el código

Escribe el siguiente código en el archivo:

```
print("¡Bienvenido al programa interactivo!")
nombre = input("¿Cómo te llamas? ")
edad = input("¿Cuántos años tienes? ")
pais = input("¿De qué país eres? ")

print("\nResumen de tu información:")
print(f"Nombre: {nombre}")
print(f"Edad: {edad}")
print(f"País: {pais}")

print(f"\nHola {nombre}, tienes {edad} años y vienes de {pais}. ¡Genial!")
```

Paso 3: Ejecutar el programa

- 1. Abre la terminal o consola.
- 2. Ejecuta el archivo:

```
python mi_primer_programa.py
```

- 3. Ingresa tus datos cuando el programa lo solicite.

Paso 4: Reflexiona

- ¿Qué sucede si ingresas un número en el campo de nombre?
- ¿Y si dejas un campo vacío?
- ¿Qué podrías agregarle al programa para que sea más interactivo?

Recomendación:

Modifica el programa para que también pregunte por tu comida favorita y la imprima al final. ¡Experimenta y diviértete!