

Módulo 1.2: Introducción Específica a Python

Prof. Carlos Cedeño

Bienvenidos al Módulo 1.2. En el módulo anterior, exploramos los conceptos fundamentales de la programación y los algoritmos. Ahora, centraremos nuestra atención específicamente en Python, el lenguaje que utilizaremos a lo largo de este curso. Este módulo te proporcionará una visión general de Python, su sintaxis, las herramientas que puedes usar para programar en él y cómo se ejecuta el código.

¿Qué es Python?

Python es un **lenguaje de programación de alto nivel, interpretado y de propósito general.**

- **Alto Nivel:** Se refiere a que su sintaxis está diseñada para ser relativamente cercana al lenguaje humano (particularmente al inglés), lo que facilita su lectura y escritura en comparación con lenguajes de bajo nivel, que están más próximos al lenguaje de la máquina.
- **Interpretado:** Como vimos en el Módulo 1.1, esto significa que el código Python se ejecuta línea por línea por un programa llamado "intérprete". Esto facilita la depuración y permite una mayor portabilidad entre diferentes sistemas operativos.
- **Propósito General:** Python no está diseñado para un único tipo de tarea. Se utiliza en una amplia variedad de aplicaciones, incluyendo desarrollo web, análisis de datos, inteligencia artificial, computación científica, automatización de tareas, desarrollo de videojuegos y mucho más.

Una de las características más apreciadas de Python es su **filosofía de diseño**, que enfatiza la **legibilidad del código** y la **simplicidad**.

La Filosofía de Python: Legibilidad y Simplicidad

Python fue creado por Guido van Rossum a finales de los años 80 y principios de los 90. Su diseño se centra en la idea de que el código se lee mucho más a menudo de lo que se escribe. Por lo tanto, la claridad es primordial.

Indentación Significativa (¡Muy Importante!)

A diferencia de muchos otros lenguajes de programación que usan llaves `{ }` o palabras clave como `begin` y `end` para definir bloques de código (grupos de instrucciones), Python utiliza la **indentación** (el espacio en blanco al inicio de una línea).

```
1 print "Starting loop"
2 for x in [1,2,3,4,5]
3     if x%2 == 0
4         x += 3
5         print x
6     print "One execution of the loop!"
7 print "end of loop"
```

- **¿Qué es un bloque de código?** Es un conjunto de sentencias que forman una unidad lógica, como el cuerpo de una función, un bucle o una estructura condicional (conceptos que veremos más adelante).


```
# Function
def sum(a, b):
    print("sum two numbers")
    resultado = a + b
    return resultado

print(sum(1, 2))

#If statement
numero = 5
if numero == 5:
    print("Es 5")
else:
    print("No es 5")
```

]} Bloque de código

} Bloque de código

} Bloque de código

Indentación

¿Cómo funciona en Python?

Las líneas de código que están indentadas al mismo nivel pertenecen al mismo bloque. Un aumento en la indentación significa el inicio de un nuevo bloque anidado, y una disminución en la indentación significa el final del bloque actual.

Ejemplo conceptual (no necesitas entender el código aún, solo observa la estructura):

```
# Esto es un comentario, se ignora al ejecutar
si condicion_es_verdadera:
    # Este bloque de código se ejecuta si la condición es verdadera
    # Observa la indentación (espacios al inicio)
    hacer_algo()
    hacer_otra_cosa()
# La siguiente línea ya no está indentada, así que está fuera del bloque anterior
continuar_con_el_programa()
```

La consistencia en la indentación (usualmente 4 espacios por nivel) es crucial; de lo contrario, Python generará un error. Esta característica obliga a escribir código visualmente estructurado y más fácil de leer.

Sensibilidad a Mayúsculas y Minúsculas (Case Sensitivity)

Python distingue entre mayúsculas y minúsculas. Esto significa que una variable llamada `nombre` es diferente de `Nombre` o `NOMBRE`.

```
variable = 10  
Variable = 15  
print(variable)  
print(Variable)
```

Salida:

```
10  
15
```

Comentarios: Notas para Humanos

Python permite incluir comentarios en el código. Los comentarios son notas o explicaciones que el intérprete ignora por completo, pero son muy útiles para los programadores (y para ti mismo en el futuro) para entender qué hace una sección particular del código.

- Un comentario de una sola línea comienza con el símbolo de almohadilla `#`.

```
# Esto es un comentario. Explica lo que hace la siguiente línea.  
edad = 30 # También puedes poner comentarios al final de una línea de código.
```

Herramientas para Escribir Python: Editores de Código e IDEs

Para escribir y trabajar con código Python, necesitarás un software especializado. Principalmente, se utilizan editores de código o Entornos de Desarrollo Integrado (IDEs).

¿Qué es un Editor de Código o un IDE?

- Un **editor de código** es un programa de texto diseñado específicamente para escribir código. Suelen incluir características como resaltado de sintaxis (colorea diferentes partes del código para facilitar la lectura), autocompletado básico y manejo de archivos.
- Un **IDE (Entorno de Desarrollo Integrado)** es una suite de software más completa que generalmente incluye un editor de código, herramientas de construcción, un depurador (para encontrar y corregir errores) y otras funcionalidades integradas para facilitar el ciclo de desarrollo de software.

Opciones Populares y Recomendadas para Principiantes:

Existen muchas opciones, pero aquí te presentamos algunas de las más populares y adecuadas para quienes comienzan con Python:

1. Visual Studio Code (VS Code):

- **Tipo:** Editor de código muy potente y extensible.
- **Características:** Gratuito, ligero, altamente personalizable con miles de extensiones (incluyendo una excelente extensión para Python de Microsoft). Amplia comunidad y soporte. Multiplataforma (Windows, macOS, Linux).
- **Ideal para:** Casi cualquier tipo de desarrollo en Python, desde scripts simples hasta proyectos grandes. Muy popular en la industria.

2. PyCharm (Community Edition):

- **Tipo:** IDE específico para Python.
- **Características:** La Community Edition es gratuita. Ofrece una integración muy profunda con Python, excelente autocompletado, herramientas de depuración y análisis de código potentes. Multiplataforma.
- **Ideal para:** Proyectos de Python de cualquier tamaño. Puede ser un poco más pesado que VS Code, pero sus características específicas para Python son muy valoradas.

3. Entornos Interactivos: Jupyter Notebooks y Google Colaboratory (Colab):

- **Tipo:** Entornos basados en web que permiten crear y compartir documentos que contienen código vivo, ecuaciones, visualizaciones y texto narrativo.
- **Características:**
 - **Jupyter Notebooks:** Se ejecutan localmente o en servidores. Muy popular en ciencia de datos, análisis y educación. Permiten ejecutar código en celdas individuales y ver los resultados inmediatamente.
 - **Google Colab:** Una versión de Jupyter Notebooks que se ejecuta completamente en la nube de Google, sin necesidad de instalación local. Ofrece acceso gratuito a GPUs para tareas más intensivas. Permite la colaboración en tiempo real.
- **Ideal para:** Experimentación, análisis de datos, aprendizaje interactivo y colaboración.

¿Cómo se Ejecuta el Código Python? 

Una vez que has escrito tu código Python, ya sea en un archivo o directamente en la consola, es crucial entender cómo "cobra vida". La ejecución del código Python siempre involucra al **intérprete de Python**.

1. El Intérprete de Python: El Traductor Esencial

Como mencionamos, Python es un lenguaje interpretado. Esto significa que el código que escribes no es entendido directamente por el hardware de tu computadora. En su lugar, un programa especial llamado **intérprete de Python** actúa como intermediario.

- **¿Qué es?** El intérprete es un programa que debes tener instalado en tu sistema (o que es provisto por entornos en la nube como Google Colab).
- **¿Qué hace?** Lee tu código Python instrucción por instrucción (o línea por línea, en general), lo analiza para asegurarse de que la sintaxis sea correcta, y luego lo traduce a un conjunto de instrucciones de más bajo nivel que la computadora puede finalmente procesar y ejecutar.

2. Ejecutando Archivos de Script (.py)

Esta es la forma más común de ejecutar programas completos en Python:


- **Escribir y Guardar:** Escribes tu código en un editor de texto o IDE y lo guardas en un archivo con la extensión `.py` (por ejemplo, `calculadora.py`, `saludo_personalizado.py`).
- **Invocar al Intérprete:** Abres una terminal o línea de comandos en tu computadora, navegas hasta el directorio donde guardaste el archivo, y le pides al intérprete de Python que ejecute tu script con un comando como:

```
python nombre_del_archivo.py
```

(En algunos sistemas, especialmente si tienes múltiples versiones de Python, puede que necesites usar `python3` en lugar de `python`).

4. Ejecución desde Editores de Código e IDEs

Cuando utilizas un editor de código avanzado (como VS Code) o un IDE (como PyCharm o Thonny):

- **Facilidades Integradas:** Estas herramientas suelen ofrecer botones de "Ejecutar" (a menudo un triángulo ) o atajos de teclado (como F5) que te permiten ejecutar tu script actual con un solo clic.
- **Funcionamiento Interno:** Es importante entender que estos botones y atajos son simplemente una forma conveniente de decirle al IDE que, "**detrás de escena**", **invoque al intérprete de Python** con el archivo que tienes abierto. El IDE gestiona el comando por ti (similar a lo que harías manualmente en la terminal) y a menudo muestra la salida en una ventana o panel integrado dentro del propio IDE.

No es Magia: El IDE no es Python; es una herramienta que te ayuda a escribir y a *gestionar la ejecución* de tu código Python a través del intérprete que tengas configurado.

En Resumen: ¿Qué Sucede Cuando "Corres" Código Python?

De forma muy simplificada:

1. Tú escribes el código fuente en Python.
2. Le pides al intérprete de Python (directamente vía terminal o indirectamente a través de un IDE) que procese ese código.
3. El intérprete analiza tu código, lo traduce a un formato que la máquina puede entender y ejecutar (pasando por un estado intermedio llamado bytecode).
4. La computadora realiza las acciones que tus instrucciones especificaron.

El Zen de Python (Opcional, pero inspirador) 📜

Python tiene una especie de "poema" o conjunto de principios guía conocido como "El Zen de Python". Puedes verlo en la consola interactiva de Python escribiendo `import this` y presionando Enter. Resume muchas de las filosofías de diseño del lenguaje