

academia



Módulo 1: Fundamentos de Python

Objetivo

Entender la sintaxis y los conceptos básicos de Python para resolver problemas sencillos y escribir scripts básicos.

1. Instalación y Configuración

- Instalación de Python y configuraciones iniciales.
- Uso de editores de código y entornos de desarrollo: Visual Studio Code, Jupyter Notebook.

2. Sintaxis Básica y Estructura de Código

- Introducción a variables y tipos de datos (números, cadenas, booleanos).
- Operadores (aritméticos, de comparación, lógicos).
- Funciones de entrada y salida (print, input).

3. Estructuras de Control de Flujo

- Condicionales (`if`, `else`, `elif`).
- Bucles (`for`, `while`).
- Introducción a listas y bucles anidados.

4. Funciones y Modularidad

- Definición y uso de funciones (`def`).
- Parámetros y argumentos, valores de retorno.
- Importación de módulos y librerías básicas.

5. Ejercicios Prácticos

- Cálculo de números primos, factorial, y Fibonacci.
 - Creación de un convertidor de unidades (por ejemplo, de millas a kilómetros).
-

Módulo 2: Estructuras de Datos

Objetivo

Aprender las estructuras de datos fundamentales de Python para almacenar y manipular datos de forma eficiente.

1. Listas

- Operaciones básicas en listas (agregar, quitar, buscar elementos).

- List comprehensions y manipulación de listas.
 - 2. **Tuplas y Conjuntos**
 - Uso de tuplas para datos inmutables.
 - Conjuntos y operaciones de conjuntos (unión, intersección, diferencia).
 - 3. **Diccionarios**
 - Creación y manipulación de diccionarios.
 - Iteración y comprensión de diccionarios.
 - 4. **Manipulación de Cadenas**
 - Métodos de cadenas y formato de texto.
 - Conversión entre tipos de datos.
 - 5. **Ejercicios Prácticos**
 - Creación de un sistema de almacenamiento de contactos con diccionarios.
 - Listas de compras y manipulación de listas.
-

Módulo 3: Programación Orientada a Objetos (OOP)

Objetivo

Dominar los conceptos básicos de la programación orientada a objetos para estructurar y organizar el código.

1. **Introducción a Objetos y Clases**
 - Clases, objetos y métodos.
 - Atributos de clase y de instancia.
 2. **Principios de OOP**
 - Encapsulamiento, herencia y polimorfismo.
 - Uso de `super()` para herencia.
 3. **Métodos Especiales**
 - `__init__` (constructor) y otros métodos especiales (`__str__`, `__repr__`).
 - Operadores sobrecargados y métodos personalizados.
 4. **Ejercicios Prácticos**
 - Creación de una clase básica para representar personas o vehículos.
 - Sistema básico de inventario o biblioteca usando clases.
-

Módulo 4: Manejo de Errores y Excepciones

Objetivo

Aprender a manejar errores de forma adecuada y a depurar código.

1. **Errores Comunes en Python**

- Errores de sintaxis, errores de tiempo de ejecución y errores lógicos.
 - 2. **Manejo de Excepciones**
 - Bloques `try`, `except`, `else`, `finally`.
 - Creación de excepciones personalizadas.
 - 3. **Depuración y Buenas Prácticas**
 - Uso de mensajes de error y depuración básica.
 - Herramientas de depuración como `pdb`.
 - 4. **Ejercicios Prácticos**
 - Escritura de funciones con manejo de errores (por ejemplo, división por cero).
 - Simulación de un cajero automático con manejo de excepciones.
-

Módulo 5: Introducción a Bibliotecas para Data Science

Objetivo

Familiarizarse con las principales bibliotecas de Python utilizadas en Data Science.

1. **Introducción a Numpy**
 - Creación y manipulación de arrays.
 - Operaciones aritméticas en arrays.
 2. **Introducción a Pandas**
 - Series y DataFrames: creación, selección y filtrado de datos.
 - Operaciones básicas con DataFrames (agrupar, ordenar, describir datos).
 3. **Visualización Básica con Matplotlib**
 - Gráficos básicos (líneas, barras, histogramas).
 - Personalización de gráficos (etiquetas, colores, títulos).
 4. **Ejercicios Prácticos**
 - Análisis básico de un dataset sencillo (por ejemplo, de calificaciones).
 - Creación de gráficos simples para visualizar datos.
-

Módulo 6: Proyectos Básicos de Programación

Objetivo

Aplicar los conocimientos adquiridos en proyectos prácticos para consolidar el aprendizaje.

1. **Proyecto 1: Calculadora de Gastos**
 - Crear una calculadora que registre y analice gastos mensuales.
 - Uso de estructuras de datos y manejo de errores.
2. **Proyecto 2: Juego Adivina el Número**

- Programar un juego en el que el usuario intenta adivinar un número generado aleatoriamente.
 - Uso de bucles, condicionales y manejo de errores.
3. **Proyecto 3: Sistema de Gestión de Contactos**
- Crear un programa que permita almacenar, buscar y eliminar contactos.
 - Uso de diccionarios y funciones.
4. **Proyecto 4: Análisis de un Dataset Simple**
- Análisis exploratorio de un dataset público (por ejemplo, datos de películas o jugadores de fútbol).
 - Uso de `pandas` y `matplotlib` para análisis y visualización de datos básicos.