

# TEMARIO DEFINITIVO

De principiante a Back-end Developer Experto en  
Python.

# Nivel Principiante

## 1. Introducción a Python

- Historia de Python
- Instalación de Python
- Configuración del entorno de desarrollo (IDEs recomendados: PyCharm, VSCode)
- Uso de la terminal y scripts básicos

## 2. Sintaxis Básica

- Variables y tipos de datos (int, float, str, bool)
- Operadores aritméticos, lógicos y de comparación
- Estructuras de control (if, elif, else)
- Bucles (for, while)
- Funciones básicas
  - Definición de funciones, parámetros y valores de retorno
- Comentarios y documentación

## 3. Estructuras de Datos

- Listas
  - Creación, acceso, modificación y métodos (append, remove, sort, etc.)
- Tuplas
  - Creación, acceso e inmutabilidad
- Diccionarios
  - Creación, acceso, modificación y métodos (keys, values, items)
- Sets
  - Creación, operaciones de conjunto (union, intersection)

#### **4. Manejo de Archivos**

- Lectura y escritura de archivos
- Manejo de archivos CSV y JSON
- Uso de context managers (`with` statement)

#### **5. Introducción a la Programación Orientada a Objetos (POO)**

- Clases y objetos
  - Definición de clases, instanciación de objetos
- Atributos y métodos
- Encapsulamiento básico

# Nivel Intermedio

## 1. Estructuras de Datos Avanzadas

- List comprehensions
- Dic comprehensions
- Manejo avanzado de listas y diccionarios

## 2. Módulos y Paquetes

- Importación y uso de módulos estándar (math, datetime, os, etc.)
- Creación de módulos y paquetes propios
- Uso de `pip` para instalación de paquetes externos

## 3. Programación Orientada a Objetos Avanzada

- Herencia
- Polimorfismo
- Encapsulamiento (métodos y atributos privados y protegidos)
- Métodos especiales (dunder methods)
- Composición y agregación

## 4. Excepciones y Manejo de Errores

- Manejo básico de excepciones (`try`, `except`, `finally`)
- Creación de excepciones personalizadas

## 5. Funciones Avanzadas

- Decoradores
- Generadores (`yield` statement`)
- Closures y funciones anidadas
- Lambda functions

## 6. Manejo de Archivos Avanzado

- Serialización con ``pickle``
- Manejo avanzado de archivos (binarios, logs)

## 7. Manejo de Procesos y Hilos

- Módulo ``threading``
- Módulo ``multiprocessing``
- Programación concurrente y paralela

# Nivel Avanzado

## 1. Programación Funcional

- Map, filter y reduce
- Funciones de orden superior
- Inmutabilidad y expresiones lambda avanzadas

## 2. Metaprogramación

- Decoradores avanzados
- Metaclases
- Manipulación del bytecode

## 3. Optimización y Performance

- Profiling y benchmark (módulo `timeit`, `cProfile`)
- Optimización de código
- Uso de bibliotecas de alta performance (NumPy, Cython)

## 4. Desarrollo Web

- Frameworks (Django, Flask)
- REST APIs
- Autenticación y autorización
- Despliegue y escalabilidad

## 5. Bases de Datos

- Conexión y manejo de bases de datos SQL (SQLite, PostgreSQL, MySQL)
- Conexión y manejo de bases de datos NoSQL (MongoDB, Redis)
- ORMs (SQLAlchemy, Django ORM)

## 6. Testing

- Pruebas unitarias con `unittest` y `pytest`
- Pruebas de integración y funcionales
- Test-driven development (TDD)

## 7. Despliegue y DevOps

- Contenedores con Docker
- CI/CD con Jenkins, GitHub Actions
- Despliegue en servicios en la nube (AWS, Heroku, DigitalOcean)

# Nivel Experto

## 1. Deep Dive en Internals de Python

- Python internals (cómo funciona el interprete CPython)
- Gestión de memoria y optimización
- Creación y uso de extensiones en C/C++

## 2. Asincronía y Concurrency Avanzada

- Asyncio (corutinas, event loop)
- Programación asíncrona avanzada
- Manejo de I/O asíncrono

## 3. Big Data y Machine Learning

- Manejo de grandes volúmenes de datos con Python (Pandas, Dask)
- Introducción a Machine Learning (scikit-learn, TensorFlow, PyTorch)
- Procesamiento de datos en tiempo real

## 4. Seguridad en Python

- Principios de seguridad en aplicaciones
- Protección contra vulnerabilidades comunes (XSS, CSRF, SQL Injection)
- Criptografía y manejo seguro de datos

## 5. Arquitectura de Software

- Patrones de diseño
- Arquitecturas escalables y microservicios
- Domain-Driven Design (DDD)



## 6. Desarrollo de Software a Gran Escala

- Mantenimiento y evolución de grandes bases de código
- Refactorización avanzada
- Gestión de dependencias y monorepos

## Recursos Adicionales

### - Libros:

- "Automate the Boring Stuff with Python" de Al Sweigart
- "Fluent Python" de Luciano Ramalho
- "Effective Python" de Brett Slatkin
- "Python Cookbook" de David Beazley y Brian K. Jones

### - Cursos en línea:

- Coursera, Udemy, Pluralsight para cursos específicos de

### Python y desarrollo backend

- MOOCs de universidades (edX, Coursera)
- Comunidades y Foros:
  - Stack Overflow
  - Reddit (subreddits como r/learnpython, r/Python)
  - Grupos de Meetup para desarrolladores de Python
  - Contribuciones a proyectos de código abierto en GitHub

Este temario estructurado te llevará desde lo básico hasta lo más avanzado en Python, cubriendo todos los aspectos necesarios para convertirte en un experto en el lenguaje.