

Curso Completo de Python - Básico a Experto

MÓDULO 1: FUNDAMENTOS DE PROGRAMACIÓN (8 sesiones)

Sesión 1: Sintaxis Básica y Operaciones

Duración: 2 horas

- Variables y asignación
- Tipos de datos básicos (int, float, str, bool)
- Operadores aritméticos, de comparación y lógicos
- Función `print()` y `input()`
- Comentarios y documentación básica

Ejercicios prácticos:

- Calculadora básica
- Conversor de unidades
- Programa de saludo personalizado

Sesión 2: Estructuras de Control - Condicionales

Duración: 2 horas

- Declaración `if`, `elif`, `else`
- Operadores de comparación múltiple
- Condicionales anidados
- Operador ternario

Ejercicios prácticos:

- Sistema de calificaciones
- Calculadora de descuentos
- Juego de adivinanza de números

Sesión 3: Estructuras de Control - Bucles

Duración: 2 horas

- Bucle `for` con `range()`
- Bucle `while`
- Instrucciones `break` y `continue`

- Bucles anidados

Ejercicios prácticos:

- Tabla de multiplicar
- Suma de números pares/impares
- Patrón de asteriscos

Sesión 4: Listas y Manipulación Básica

Duración: 2 horas

- Creación y acceso a listas
- Métodos básicos: `append()`, `remove()`, `pop()`, `insert()`
- Slicing de listas
- Iteración sobre listas

Ejercicios prácticos:

- Lista de compras interactiva
- Ordenamiento manual de números
- Eliminar duplicados

Sesión 5: Tuplas y Conjuntos

Duración: 2 horas

- Tuplas: creación, acceso, inmutabilidad
- Conjuntos: operaciones básicas, unión, intersección
- Conversiones entre tipos de colecciones
- Cuándo usar cada tipo

Ejercicios prácticos:

- Coordenadas geográficas con tuplas
- Análisis de palabras únicas en texto
- Comparación de listas de estudiantes

Sesión 6: Diccionarios

Duración: 2 horas

- Creación y acceso a diccionarios
- Métodos: `keys()`, `values()`, `items()`

- Diccionarios anidados
- Iteración sobre diccionarios

Ejercicios prácticos:

- Agenda de contactos
- Contador de palabras
- Sistema de inventario básico

Sesión 7: Strings y Manipulación de Texto

Duración: 2 horas

- Métodos de string: upper(), lower(), strip(), split(), join()
- Formateo de strings (f-strings, format())
- Expresiones regulares básicas
- Validación de datos de entrada

Ejercicios prácticos:

- Validador de email básico
- Contador de caracteres y palabras
- Generador de contraseñas

Sesión 8: Funciones Básicas

Duración: 2 horas

- Definición y llamada de funciones
- Parámetros y argumentos
- Valores de retorno
- Alcance de variables (scope local/global)

Ejercicios prácticos:

- Calculadora modular
- Validadores de datos
- Generador de reportes simples

MÓDULO 2: PROGRAMACIÓN INTERMEDIA (10 sesiones)

Sesión 9: Funciones Avanzadas

Duración: 2.5 horas

- Argumentos por defecto y argumentos variables (*args, **kwargs)
- Funciones lambda
- Funciones como objetos
- Decoradores básicos

Ejercicios prácticos:

- Sistema de logging personalizado
- Calculadora con funciones especializadas
- Medidor de tiempo de ejecución

Sesión 10: Comprensiones de Listas y Generadores

Duración: 2.5 horas

- List comprehensions
- Dictionary comprehensions
- Set comprehensions
- Expresiones generadoras
- Función `yield`

Ejercicios prácticos:

- Procesador de datos CSV
- Generador de números primos
- Filtrado masivo de datos

Sesión 11: Manejo de Archivos

Duración: 2.5 horas

- Apertura y cierre de archivos
- Modos de apertura (r, w, a, r+)
- Context managers (`with`)
- Procesamiento de archivos CSV y JSON
- Manejo de rutas con `pathlib`

Ejercicios prácticos:

- Sistema de logs
- Procesador de datos de ventas
- Conversor de formatos de archivo

Sesión 12: Manejo de Errores y Excepciones

Duración: 2.5 horas

- Tipos de errores comunes
- Bloques try/except/else/finally
- Creación de excepciones personalizadas
- Debugging básico
- Validación robusta de datos

Ejercicios prácticos:

- Validador robusto de formularios
- Calculadora a prueba de errores
- Sistema de backup automático

Sesión 13: Módulos y Paquetes

Duración: 2.5 horas

- Importación de módulos
- Creación de módulos propios
- Estructura de paquetes
- Variable `__name__` y `__main__`
- Módulos de la biblioteca estándar útiles

Ejercicios prácticos:

- Biblioteca de utilidades matemáticas
- Sistema modular de gestión de estudiantes
- Kit de herramientas de texto

Sesión 14: Fechas, Tiempo y Expresiones Regulares

Duración: 2.5 horas

- Módulo datetime
- Formateo y parsing de fechas
- Cálculos con fechas
- Expresiones regulares con módulo re
- Patrones comunes de validación

Ejercicios prácticos:

- Calculadora de edad y días
- Validador de formatos de datos
- Extractor de información de texto

Sesión 15: Estructuras de Datos Avanzadas

Duración: 2.5 horas

- Collections: Counter, defaultdict, namedtuple
- Deque y su uso
- Heapq para colas de prioridad
- Implementación de pilas y colas

Ejercicios prácticos:

- Sistema de análisis estadístico
- Simulador de cola de atención
- Organizador de tareas por prioridad

Sesión 16: Algoritmos de Ordenamiento y Búsqueda

Duración: 2.5 horas

- Algoritmos de ordenamiento (burbuja, inserción, selección)
- Búsqueda lineal y binaria
- Recursión básica
- Análisis de complejidad temporal básico

Ejercicios prácticos:

- Comparador de algoritmos de ordenamiento
- Motor de búsqueda básico
- Resolver problemas con recursión

Sesión 17: Introducción a la Programación Orientada a Objetos

Duración: 3 horas

- Conceptos básicos: clases y objetos
- Atributos y métodos
- Constructor `__init__`
- Métodos especiales básicos (`__str__`, `__repr__`)

Ejercicios prácticos:

- Clase Persona y Estudiante
- Sistema de cuentas bancarias
- Biblioteca de libros

Sesión 18: POO Avanzada

Duración: 3 horas

- Herencia y polimorfismo
- Encapsulación y atributos privados
- Métodos de clase y estáticos
- Propiedades con decorador @property

Ejercicios prácticos:

- Jerarquía de vehículos
- Sistema de empleados con herencia
- Simulador de formas geométricas

MÓDULO 3: PROGRAMACIÓN AVANZADA (12 sesiones)

Sesión 19: Patrones de Diseño

Duración: 3 horas

- Singleton
- Factory Method
- Observer
- Strategy Pattern
- Decorator Pattern aplicado

Ejercicios prácticos:

- Sistema de configuración global
- Fábrica de objetos
- Sistema de notificaciones

Sesión 20: Programación Funcional

Duración: 3 horas

- Funciones de orden superior

- map(), filter(), reduce()
- Functools: partial, lru_cache
- Inmutabilidad y efectos secundarios
- Closures avanzados

Ejercicios prácticos:

- Pipeline de procesamiento de datos
- Sistema de cache inteligente
- Transformador de datos funcional

Sesión 21: Iteradores y Generadores Avanzados

Duración: 3 horas

- Protocolo de iterador
- Creación de iteradores personalizados
- Generadores infinitos
- yield from
- Itertools para combinaciones complejas

Ejercicios prácticos:

- Generador de secuencias matemáticas
- Iterator para archivos grandes
- Pipeline de datos streaming

Sesión 22: Context Managers y Descriptores

Duración: 3 horas

- Protocolo de context manager
- Creación de context managers personalizados
- Descriptores y su protocolo
- Metaclasses básicas

Ejercirios prácticos:

- Manager de conexiones de base de datos
- Sistema de validación de atributos
- Creador automático de propiedades

Sesión 23: Concurrencia - Threading

Duración: 3 horas

- Conceptos de concurrencia vs paralelismo
- Módulo threading
- Locks, RLocks, y Semaphores
- Queue thread-safe
- Problemas comunes (race conditions, deadlocks)

Ejercicios prácticos:

- Descargador concurrente de archivos
- Simulador de productor-consumidor
- Monitor de sistema en tiempo real

Sesión 24: Concurrencia - Multiprocessing

Duración: 3 horas

- Diferencias entre threads y processes
- Módulo multiprocessing
- Pool de procesos
- Comunicación entre procesos
- Shared memory

Ejercicios prácticos:

- Procesador paralelo de imágenes
- Calculadora distribuida
- Sistema de análisis de logs masivos

Sesión 25: Programación Asíncrona - Asyncio

Duración: 3.5 horas

- Conceptos de programación asíncrona
- async/await
- Event loop
- Coroutines y tasks
- Aiohttp para requests asíncronos

Ejercicios prácticos:

- Web scraper asíncrono
- Chat server básico
- Monitor de APIs múltiples

Sesión 26: Testing y Debugging Avanzado

Duración: 3.5 horas

- Unittest framework
- Pytest básico y avanzado
- Mocking y fixtures
- Coverage testing
- Debugging con pdb
- Profiling de rendimiento

Ejercicios prácticos:

- Suite completa de tests para proyecto
- Tests de integración
- Benchmarking de algoritmos

Sesión 27: Metaclasses y Programación Metaprogramática

Duración: 3.5 horas

- Qué son las metaclasses
- Creación dinámica de clases
- `__new__` vs `__init__`
- Decoradores de clase
- Reflection y introspección

Ejercicios prácticos:

- ORM básico
- Sistema de validación automática
- Framework de serialización

Sesión 28: Optimización y Performance

Duración: 3.5 horas

- Profiling con cProfile y line_profiler

- Optimización de algoritmos
- Uso eficiente de memoria
- NumPy para cálculos numéricos
- Cython básico

Ejercicios prácticos:

- Optimizador de consultas
- Procesador de big data
- Algoritmos optimizados para ML

Sesión 29: Integración con C/C++

Duración: 3.5 horas

- ctypes para librerías C
- CFFI (C Foreign Function Interface)
- Extensiones con Cython
- Pybind11 básico
- Casos de uso para integración

Ejercicios prácticos:

- Wrapper para librería C
- Acelerador numérico con Cython
- Interface con sistema operativo

Sesión 30: Arquitectura de Software Avanzada

Duración: 3.5 horas

- Principios SOLID
- Clean Architecture
- Dependency Injection
- Event-driven architecture
- Microservicios con Python

Ejercicios prácticos:

- Refactoring de aplicación monolítica
- Sistema de eventos
- API microservice

MÓDULO 4: ESPECIALIZACIÓN Y APLICACIONES (8 sesiones)

Sesión 31: Desarrollo Web con Flask/FastAPI

Duración: 4 horas

- Fundamentos de desarrollo web
- Routes y templates
- Manejo de formularios
- APIs REST
- Autenticación básica

Ejercicios prácticos:

- API de gestión de tareas
- Web app con autenticación
- Microservicio de datos

Sesión 32: Bases de Datos y ORM

Duración: 4 horas

- SQLAlchemy Core y ORM
- Migraciones de base de datos
- Queries complejas
- Optimización de consultas
- Bases de datos NoSQL (MongoDB)

Ejercicios prácticos:

- Sistema completo de gestión
- Análisis de datos empresariales
- API con base de datos

Sesión 33: Data Science y Análisis

Duración: 4 horas

- Pandas para manipulación de datos
- NumPy para cálculos numéricos
- Matplotlib y Seaborn para visualización
- Análisis estadístico básico

- Jupyter Notebooks

Ejercicios prácticos:

- Análisis de dataset real
- Dashboard de métricas
- Predicción simple con sklearn

Sesión 34: Machine Learning Básico

Duración: 4 horas

- Scikit-learn básico
- Regresión lineal y logística
- Clustering (K-means)
- Validación cruzada
- Evaluación de modelos

Ejercicios prácticos:

- Predictor de precios
- Clasificador de texto
- Sistema de recomendaciones básico

Sesión 35: Automatización y Scripting

Duración: 4 horas

- Automatización de tareas del SO
- Web scraping con BeautifulSoup y Scrapy
- Automatización de correos
- Bots para redes sociales
- Scheduled tasks

Ejercicios prácticos:

- Bot de scraping de noticias
- Automatizador de reportes
- Monitor de precios

Sesión 36: DevOps y Deployment

Duración: 4 horas

- Docker básico para Python
- CI/CD con GitHub Actions
- Deployment en cloud (Heroku, AWS)
- Monitoring y logging
- Configuration management

Ejercicios prácticos:

- Containerizar aplicación Python
- Pipeline de deployment automático
- Monitor de aplicación en producción

Sesión 37: Security y Best Practices

Duración: 4 horas

- Seguridad en aplicaciones Python
- Validación y sanitización
- Criptografía básica
- OWASP Top 10 para Python
- Code review y static analysis

Ejercicios prácticos:

- Auditoría de seguridad
- Sistema de autenticación seguro
- Encriptador de datos

Sesión 38: Proyecto Final Integrado

Duración: 6 horas

- Planificación de proyecto complejo
- Integración de todos los conceptos
- Documentación profesional
- Testing comprehensivo
- Presentación de proyecto

Proyecto Final: Sistema completo que integre:

- Backend con API REST
- Base de datos con ORM

- Frontend básico
- Tests automatizados
- Deployment en cloud
- Documentación completa

METODOLOGÍA DE EVALUACIÓN

Evaluación Continua (60%)

- Ejercicios prácticos semanales
- Quizzes de conceptos clave
- Participación en clase
- Proyectos modulares

Proyectos Intermedios (25%)

- Proyecto Módulo 1: Aplicación de consola completa
- Proyecto Módulo 2: Sistema POO con archivos
- Proyecto Módulo 3: Aplicación concurrente

Proyecto Final (15%)

- Sistema completo integrando múltiples tecnologías
- Documentación técnica
- Presentación oral
- Code review peer-to-peer

RECURSOS COMPLEMENTARIOS

Herramientas Recomendadas

- IDE: PyCharm Community/VS Code
- Control de versiones: Git
- Testing: pytest
- Linting: flake8, black
- Documentación: Sphinx

Bibliografía

- "Python Crash Course" - Eric Matthes
- "Fluent Python" - Luciano Ramalho
- "Effective Python" - Brett Slatkin

- "Architecture Patterns with Python" - Harry Percival

Plataformas de Práctica

- LeetCode para algoritmos
- HackerRank para challenges
- Kaggle para data science
- GitHub para proyectos open source

CRONOGRAMA SUGERIDO

Duración Total: 6 meses

- **Mes 1-2:** Módulo 1 (Fundamentos)
- **Mes 3-4:** Módulo 2 (Intermedio)
- **Mes 5:** Módulo 3 (Avanzado)
- **Mes 6:** Módulo 4 (Especialización)

Modalidad: 3 sesiones por semana, 2-4 horas por sesión

Total de horas lectivas: 120 horas **Horas de práctica estimadas:** 200 horas