LỘ TRÌNH HỌC PYTHON TỪ CƠ BẢN ĐẾN NÂNG CAO

PYTHON FROM BASIC TO ADVANCED

THÔNG TIN KHÓA HỌC

Thông tin	Chi tiết
Tên khóa học	Python từ Cơ Bản đến Nâng Cao
Yêu cầu	Kiến thức cơ bản về máy tính, Logic tư duy
Tổng số buổi	36 buổi
Thời lượng	2.5 giờ/buổi
Tổng thời gian	90 giờ
Lộ trình	12 tuần (3 buổi/tuần)
Python Version	Python 3.10+

MỤC TIÊU

- Nắm vững Python fundamentals và syntax
- Lập trình hướng đối tượng với Python
- Hiểu và sử dụng thành thạo Python's standard library
- Viết clean, Pythonic code
- Handle errors và exceptions properly
- Testing và debugging
- Working với files, databases, APIs
- Ready cho Web Development, Data Science, Automation

CẤU TRÚC KHÓA HỌC - 36 BUỔI

Module	Tên Module	Buổi
1	Python Basics - Cơ Bản Python	6
2	Data Structures - Cấu Trúc Dữ Liệu	4
3	Functions & Modules - Hàm và Modules	4
4	OOP - Lập Trình Hướng Đối Tượng	5
5	Advanced Python - Python Nâng Cao	5

Module	Tên Module	Buổi
6	File & Database - File và Cơ Sở Dữ Liệu	3
7	Testing & Debugging	3
8	Standard Library & Tools	3
9	Web & APIs	3

MODULE 1: PYTHON BASICS (6 BUỔI)

Buổi 1: Introduction & Setup

- Python là gì: History, Use cases
- Python 2 vs Python 3
- Cài đặt Python: Windows, Mac, Linux
- Python REPL (Interactive Shell)
- IDEs & Editors: PyCharm, VS Code, Jupyter Notebook
- Virtual environments: venv, virtualenv
- pip: Package manager
- First program: Hello World
- Running Python scripts
- Python style guide: PEP 8

Buổi 2: Variables & Data Types

- Variables: Declaration, Assignment
- Dynamic typing
- Naming conventions
- Basic data types:
  - Numbers: int, float, complex
  - Strings: str
  - Booleans: bool
  - None type
- Type conversion: int(), float(), str(), bool()
- Basic operators:
  - Arithmetic: +, -, *, /, //, %, **
  - Comparison: ==, !=, <, >, <=, >=
  - Logical: and, or, not
  - Assignment: =, +=, -=, *=, /=
- Operator precedence
- input() function: User input

Buổi 3: Strings

- String literals: Single quotes, Double quotes, Triple quotes
- String indexing: Positive, Negative
- String slicing: [start🔚step]

- String methods:
    - Case: upper(), lower(), capitalize(), title()
    - Search: find(), index(), count()
    - Check: startswith(), endswith(), isdigit(), isalpha()
    - Modify: replace(), strip(), split(), join()
- String formatting:
    - Old style: % formatting
    - format() method
    - f-strings (Python 3.6+)
- String concatenation
- Raw strings: r""
- Escape characters

Buổi 4: Control Flow

- Conditional statements:
    - if statement
    - if-else statement
    - if-elif-else chain
    - Nested if
    - Ternary operator: x if condition else y
- Boolean context: Truthy and Falsy values
- Loops:
    - while loop
    - for loop
    - range() function
    - Iterating over sequences
- Loop control:
    - break: Exit loop
    - continue: Skip iteration
    - pass: Placeholder
    - else clause in loops
- Nested loops

Buổi 5: Basic I/O & Operators

- print() function:
    - Multiple arguments
    - sep parameter
    - end parameter
    - file parameter
- input() function: Reading user input
- Type conversion with input
- Identity operators: is, is not
- Membership operators: in, not in
- Bitwise operators: &, |, ^, ~, <<, >>
- Walrus operator := (Python 3.8+)

- Chaining comparisons: 1 < x < 10

Buổi 6: Practice & Mini Projects

- Project 1: Calculator
- Project 2: Number guessing game
- Project 3: Password strength checker
- Project 4: Simple text-based menu
- Project 5: Temperature converter
- Code review và best practices
- Common beginner mistakes

MODULE 2: DATA STRUCTURES (4 BUỔI)

Buổi 7: Lists

- List basics: Creation, Indexing, Slicing
- List methods:
  - Adding: append(), insert(), extend()
  - Removing: remove(), pop(), clear()
  - Sorting: sort(), reverse()
  - Searching: index(), count()
- List operations: +, *, in
- List comprehensions:
  - Basic syntax
  - With conditions
  - Nested comprehensions
- Copying lists: Shallow vs Deep copy
- Unpacking lists
- Nested lists: 2D lists, Matrices

Buổi 8: Tuples & Sets

- Tuples:
  - Immutable sequences
  - Creation: (), tuple()
  - Packing and unpacking
  - Single element tuples
  - Tuple methods: count(), index()
  - Named tuples
  - When to use tuples vs lists
- Sets:
  - Unordered collections
  - Creation: {}, set()
  - Set operations: union, intersection, difference
  - Set methods: add(), remove(), discard()
  - Frozen sets

    - Set comprehensions

Buổi 9: Dictionaries

- Dictionary basics: Key-value pairs
- Creating dictionaries
- Accessing values: [], get()
- Adding and updating items
- Removing items: pop(), popitem(), clear()
- Dictionary methods:
    - keys(), values(), items()
    - update()
    - setdefault()
- Dictionary comprehensions
- Nested dictionaries
- Iterating through dictionaries
- Dictionary unpacking
- defaultdict và Counter (collections module)
- OrderedDict

Buổi 10: Advanced Collections

- Collections module:
    - namedtuple: Tuples with named fields
    - deque: Double-ended queue
    - Counter: Count hashable objects
    - defaultdict: Dict with default values
    - OrderedDict: Remember insertion order
    - ChainMap: Combine dictionaries
- Itertools module (preview):
    - product, permutations, combinations
    - count, cycle, repeat
    - chain, zip_longest
- Array module: Efficient numeric arrays
- Choosing the right data structure

MODULE 3: FUNCTIONS & MODULES (4 BUỔI)

Buổi 11: Functions Basics

- Defining functions: def keyword
- Function parameters:
    - Positional parameters
    - Default parameters
    - Keyword arguments
    - Arbitrary arguments: *args
    - Arbitrary keyword arguments: **kwargs

    - Positional-only and keyword-only parameters
- Return values:
    - return statement
    - Multiple return values
    - None as default return
- Function scope:
    - Local scope
    - Global scope
    - global keyword
    - nonlocal keyword
- Docstrings: Documenting functions

Buổi 12: Advanced Functions

- Lambda functions: Anonymous functions
- map() function
- filter() function
- reduce() function (functools)
- Higher-order functions
- Closures:
    - Nested functions
    - Accessing outer scope
- Decorators:
    - Function decorators
    - @syntax
    - Wrapping functions
    - Decorators with arguments
    - Multiple decorators
    - functools.wraps
- Recursion:
    - Base case and recursive case
    - Recursion vs Iteration
    - Tail recursion

Buổi 13: Modules & Packages

- Modules:
    - Creating modules
    - Importing modules: import, from...import
    - Module search path
    - **name** == "**main**"
    - Reloading modules
- Packages:
    - Creating packages
    - **init**.py file
    - Package imports
    - Relative imports

    - Namespace packages
- Standard library overview
- Third-party packages: PyPI
- requirements.txt
- Installing packages: pip install

Buổi 14: Useful Built-in Functions

- Type functions: type(), isinstance()
- Conversion functions: int(), str(), list(), dict()
- Math functions: abs(), pow(), round(), min(), max(), sum()
- Sequence functions: len(), sorted(), reversed(), enumerate(), zip()
- all() and any()
- eval() and exec() (và risks)
- dir() and help()
- id() and hash()
- getattr(), setattr(), hasattr()
- callable()

MODULE 4: OOP (5 BUỔI)

Buổi 15: Classes & Objects

- OOP concepts: Classes, Objects, Attributes, Methods
- Defining classes: class keyword
- **init** method: Constructor
- Instance variables vs Class variables
- self parameter
- Instance methods
- Creating objects
- Accessing attributes và methods
- **str** and **repr** methods
- **dict** attribute
- Class documentation

Buổi 16: Encapsulation & Properties

- Encapsulation concept
- Public vs Private:
    - Convention: _single_leading_underscore
    - Name mangling: __double_leading_underscore
- Getters and setters
- @property decorator:
    - Property getter
    - Property setter
    - Property deleter
- Read-only attributes

- Computed properties
- Validation in setters

Buổi 17: Inheritance

- Inheritance basics: Parent class, Child class
- Single inheritance
- Method overriding
- super() function
- Multiple inheritance:
  - MRO (Method Resolution Order)
  - Diamond problem
  - super() in multiple inheritance
- Multilevel inheritance
- isinstance() and issubclass()
- Abstract base classes (abc module)

Buổi 18: Polymorphism & Magic Methods

- Polymorphism concept
- Duck typing
- Method overloading (not direct support)
- Magic methods (Dunder methods):
  - **init**, **str**, **repr**
  - Comparison: **eq**, **lt**, **gt**, etc.
  - Arithmetic: **add**, **sub**, **mul**, etc.
  - Container: **len**, **getitem**, **setitem**
  - Callable: **call**
  - Context managers: **enter**, **exit**
  - Attribute access: **getattr**, **setattr**
- Operator overloading
- Iterator protocol: **iter**, **next**

Buổi 19: Advanced OOP

- Class methods: @classmethod
- Static methods: @staticmethod
- Class methods vs Static methods vs Instance methods
- Composition vs Inheritance
- Mixins
- Data classes: @dataclass (Python 3.7+)
- Slots: **slots**
- Metaclasses (overview)
- Design patterns (overview):
  - Singleton
  - Factory
  - Observer
  - Strategy

MODULE 5: ADVANCED PYTHON (5 BUỔI)

Buổi 20: Iterators & Generators

- Iteration protocol: **iter**, **next**
- Creating custom iterators
- StopIteration exception
- iter() function
- Generators:
    - Generator functions: yield keyword
    - Generator expressions
    - yield vs return
    - Generator advantages: Memory efficiency
- Generator methods: send(), throw(), close()
- Chaining generators
- itertools module:
    - Infinite iterators: count, cycle, repeat
    - Combinatorics: product, permutations, combinations
    - Other: chain, islice, tee

Buổi 21: Decorators & Context Managers

- Function decorators (review)
- Class decorators
- Decorator patterns:
    - Timing decorator
    - Logging decorator
    - Caching/Memoization
    - Authorization decorator
- functools module:
    - wraps
    - lru_cache
    - partial
- Context managers:
    - with statement
    - **enter** and **exit**
    - Creating context managers
    - contextlib module: contextmanager decorator
    - Multiple context managers
- Use cases: File handling, Database connections, Locks

Buổi 22: Exception Handling

- Errors vs Exceptions
- Exception hierarchy
- try-except block
- Catching specific exceptions

- Multiple except blocks
- except Exception vs bare except
- else clause in try-except
- finally clause
- Raising exceptions: raise keyword
- Custom exceptions:
  - Inheriting from Exception
  - Exception with arguments
- Exception chaining
- assert statement
- Best practices:
  - Specific exceptions
  - Don't catch all
  - Logging exceptions
  - Clean up resources

Buổi 23: Comprehensions & Functional Programming

- List comprehensions (review)
- Dictionary comprehensions
- Set comprehensions
- Generator expressions
- Nested comprehensions
- Conditional expressions trong comprehensions
- Functional programming concepts:
  - First-class functions
  - Pure functions
  - Immutability
- map(), filter(), reduce()
- Lambda expressions (review)
- functools và operator modules
- Partial functions
- Currying

Buổi 24: Type Hints & Documentation

- Type hints (Python 3.5+):
  - Basic types: int, str, float, bool
  - Collections: List, Dict, Tuple, Set
  - Optional types: Optional, Union
  - Callable types
  - Type aliases
  - Generic types
- typing module
- mypy: Static type checker
- Docstrings:
  - PEP 257

    - Google style
    - NumPy style
    - reStructuredText
- Sphinx: Documentation generator
- Type hints best practices

MODULE 6: FILE & DATABASE (3 BUỔI)

Buổi 25: File Handling

- Opening files: open() function
- File modes: r, w, a, r+, w+, a+, b
- Reading files:
    - read(): Entire file
    - readline(): Single line
    - readlines(): List of lines
    - Iterating over file object
- Writing files:
    - write()
    - writelines()
- File positions: tell(), seek()
- with statement: Automatic close
- Binary files
- File and directory operations:
    - os module: path operations
    - shutil module: File operations
    - pathlib module: Modern path handling
- Working with CSV: csv module
- Working with JSON: json module
- Working with XML: xml module

Buổi 26: Working with Data Formats

- JSON:
    - json.dumps(), json.loads()
    - json.dump(), json.load()
    - Custom JSON encoders
- CSV:
    - Reading CSV: csv.reader, csv.DictReader
    - Writing CSV: csv.writer, csv.DictWriter
    - Dialects
- YAML (with PyYAML)
- XML (with xml.etree.ElementTree)
- ConfigParser: INI files
- pickle: Python object serialization
- Regular expressions:
    - re module

  - Pattern matching
  - Search, match, findall
  - Groups và capturing
  - Substitution

Buổi 27: Database Integration

- Database concepts
- SQLite:
  - sqlite3 module
  - Connecting to database
  - Creating tables
  - CRUD operations
  - Executing queries
  - Parameterized queries
  - Transactions
  - Error handling
- Database best practices:
  - Connection pooling
  - Prepared statements
  - SQL injection prevention
- ORM preview:
  - SQLAlchemy (overview)
  - Django ORM (overview)
- NoSQL databases (overview):
  - MongoDB with PyMongo
  - Redis with redis-py

MODULE 7: TESTING & DEBUGGING (3 BUỔI)

Buổi 28: Unit Testing

- Testing importance
- unittest module:
  - Test cases: unittest.TestCase
  - Test methods: test_*
  - Assertions: assertEqual, assertTrue, assertRaises, etc.
  - setUp and tearDown
  - Test suites
  - Test discovery
- pytest framework:
  - Installation
  - Writing tests
  - Fixtures
  - Parametrized tests
  - Markers
  - Plugins

- Test-Driven Development (TDD)
- Code coverage: coverage.py
- Mocking: unittest.mock

Buổi 29: Debugging

- Debugging strategies
- print() debugging
- Logging:
    - logging module
    - Logging levels: DEBUG, INFO, WARNING, ERROR, CRITICAL
    - Loggers, handlers, formatters
    - Configuration
    - Logging to files
    - Rotating logs
- Python debugger (pdb):
    - Setting breakpoints
    - Stepping through code
    - Inspecting variables
    - pdb commands
- IDE debugging tools
- Profiling:
    - cProfile
    - timeit module
    - line_profiler
    - memory_profiler

Buổi 30: Code Quality

- Code style:
    - PEP 8
    - Naming conventions
    - Code organization
- Linting tools:
    - pylint
    - flake8
    - pycodestyle
- Code formatters:
    - black
    - autopep8
- Type checking: mypy
- Code complexity: mccabe
- Documentation:
    - Docstrings
    - Comments
    - README files
- Version control: Git basics

- Code review best practices

MODULE 8: STANDARD LIBRARY & TOOLS (3 BUỔI)

Buổi 31: Important Standard Library Modules

- datetime module:
  - date, time, datetime objects
  - timedelta
  - Formatting và parsing
  - Timezone handling
- collections module (detailed):
  - Counter, defaultdict, OrderedDict
  - deque, namedtuple, ChainMap
- random module:
  - random(), randint(), choice()
  - shuffle(), sample()
  - seed()
- math module:
  - Mathematical functions
  - Constants: pi, e
- statistics module
- sys module:
  - sys.argv: Command-line arguments
  - sys.path, sys.modules
  - sys.exit()
- os module:
  - Environment variables
  - Process management
  - Path operations

Buổi 32: Working with Time & Concurrency

- time module:
  - time(), sleep()
  - Performance timing
- Threading:
  - threading module
  - Creating threads
  - Thread synchronization: Lock, RLock
  - Thread communication: Queue
  - Thread pool: ThreadPoolExecutor
- Multiprocessing:
  - multiprocessing module
  - Process creation
  - Process communication: Queue, Pipe
  - Process pool: ProcessPoolExecutor

- Asyncio (introduction):
    - Async/await syntax
    - Coroutines
    - Event loop
    - asyncio.gather()
- Global Interpreter Lock (GIL)

Buổi 33: Command Line & System

- argparse module:
    - Defining arguments
    - Positional arguments
    - Optional arguments
    - Argument types
    - Help messages
- subprocess module:
    - Running external commands
    - Capturing output
    - Error handling
- Environment variables: os.environ
- Platform module: System information
- pathlib module (detailed):
    - Path objects
    - Path operations
    - File operations
- Building CLI tools
- Creating executables: PyInstaller
- Virtual environments management

MODULE 9: WEB & APIs (3 BUỔI)

Buổi 34: HTTP & REST APIs

- HTTP basics: Methods, Status codes, Headers
- requests library:
    - GET requests
    - POST requests
    - PUT, DELETE requests
    - Headers, Parameters
    - JSON data
    - Authentication
    - Sessions
    - Timeouts
    - Error handling
- Working with APIs:
    - REST APIs
    - API keys

    - Rate limiting
    - Pagination
- urllib module (standard library)
- http.client module

Buổi 35: Web Scraping

- Web scraping basics
- HTML structure
- BeautifulSoup:
    - Installation
    - Parsing HTML
    - Finding elements
    - Navigating DOM
    - Extracting data
    - CSS selectors
- Handling JavaScript: Selenium (overview)
- scrapy framework (overview)
- Ethics và legality
- robots.txt
- Best practices:
    - Respecting rate limits
    - User agents
    - Error handling

Buổi 36: Web Frameworks Overview & Final Project

- Web frameworks overview:
    - **Flask:**
        - Lightweight
        - Routes
        - Templates
        - Forms
    - **Django:**
        - Full-featured
        - MTV architecture
        - ORM
        - Admin panel
    - **FastAPI:**
        - Modern
        - Async support
        - Type hints
        - Auto documentation
- REST API creation (simple example)
- Templates: Jinja2
- Forms handling
- Database integration

- Authentication basics
- Deployment basics
- **Final Project:** Choose one:
  - CLI Tool (File organizer, Task manager)
  - Web Scraper (News aggregator)
  - REST API (Todo API, Blog API)
  - Automation Script (Email sender, Report generator)
  - Data Analysis Tool (CSV analyzer)

PYTHON DEVELOPER ROADMAP

Core Skills (Buổi 1-24):

- Python syntax và fundamentals
- Data structures
- OOP
- Functional programming
- Exception handling
- File I/O

Advanced Skills (Buổi 25-36):

- Database integration
- Testing và debugging
- Standard library mastery
- Web và APIs
- Concurrency

Next Steps Based on Career Path:

1. Web Development:

- Deep dive into Flask/Django/FastAPI
- Frontend basics: HTML, CSS, JavaScript
- Databases: PostgreSQL, MongoDB
- Deployment: Docker, AWS, Heroku
- REST APIs, GraphQL

2. Data Science:

- NumPy: Numerical computing
- Pandas: Data manipulation
- Matplotlib/Seaborn: Visualization
- Scikit-learn: Machine learning
- Jupyter notebooks
- Statistics và math

3. Automation & DevOps:

- Advanced scripting
- CI/CD: Jenkins, GitHub Actions
- Configuration management: Ansible
- Cloud platforms: AWS, Azure, GCP
- Monitoring tools
- Docker và Kubernetes

4. Machine Learning/AI:

- Mathematics: Linear algebra, Calculus, Statistics
- TensorFlow/PyTorch
- Deep Learning
- NLP
- Computer Vision
- MLOps

5. Game Development:

- Pygame
- Game logic
- Graphics và animation
- Sound handling

LỘ TRÌNH HỌC TẬP 12 TUẦN

Tuần 1-2: Python Basics (6)
Tuần 3-4: Data Structures (4), Functions & Modules (2)
Tuần 5: Functions & Modules (2), OOP (1)
Tuần 6-7: OOP (4), Advanced Python (2)
Tuần 8: Advanced Python (3)
Tuần 9: File & Database (3)
Tuần 10: Testing & Debugging (3)
Tuần 11: Standard Library & Tools (3)
Tuần 12: Web & APIs (3)
