

Here's a detailed week-by-week breakdown of a comprehensive Python course, covering fundamental to advanced topics:

Week 1: Introduction to Python

1. Getting Started

- Introduction to Python
- Installing Python and setting up the development environment
- Running Python scripts

2. Basic Syntax

- Python syntax and conventions
- Comments and docstrings

3. Basic Data Types

- Numbers, strings, and booleans
- Type conversion

4. Practice: Simple Python scripts, basic arithmetic operations, and string manipulations

Week 2: Control Flow and Functions

5. Control Flow

- if, elif, and else statements
- while and for loops
- break, continue, and pass statements

6. Functions

- Defining and calling functions
- Function arguments and return values
- Variable scope

7. Practice: Writing functions, implementing control flow in scripts

Week 3: Data Structures

8. Lists

- Creating and accessing lists
- List methods and comprehensions

9. Tuples

- Creating and accessing tuples

- Tuple operations

10. Sets

- Creating and using sets
- Set operations

11. Dictionaries

- Creating and accessing dictionaries
- Dictionary methods and comprehensions

12. Practice: Working with different data structures, manipulating data

Week 4: Modules and Packages

13. Modules

- Importing modules
- Creating and using custom modules

14. Packages

- Creating and using packages
- Installing third-party packages using pip

15. Practice: Creating and using modules and packages, installing and using third-party packages

Week 5: File Handling

16. Reading and Writing Files

- Opening and closing files
- Reading from and writing to files

17. File Methods

- Using file methods for various operations

18. Practice: Reading from and writing to different types of files, working with file methods

Week 6: Error Handling and Exceptions

19. Exceptions

- Understanding exceptions
- Handling exceptions with try, except, else, and finally blocks

20. Custom Exceptions

- Raising and creating custom exceptions

21. Practice: Implementing error handling in scripts, creating custom exceptions

Week 7: Object-Oriented Programming (OOP)

22. Classes and Objects

- Defining classes and creating objects
- Instance variables and methods

23. Inheritance

- Implementing inheritance
- Method overriding

24. Encapsulation and Polymorphism

- Understanding encapsulation
- Implementing polymorphism

25. Practice: Creating and using classes, implementing inheritance and polymorphism

Week 8: Advanced OOP Concepts

26. Special Methods

- Understanding and using special methods
- Operator overloading

27. Class and Static Methods

- Defining and using class methods
- Defining and using static methods

28. Practice: Implementing advanced OOP concepts in classes

Week 9: Working with Libraries

29. Standard Libraries

- Overview of Python standard libraries
- Using libraries like datetime, math, random, etc.

30. Third-Party Libraries

- Installing and using popular third-party libraries (e.g., NumPy, pandas)

31. Practice: Using standard and third-party libraries in projects

Week 10: Data Analysis and Visualization

32. NumPy and pandas

- Introduction to NumPy and pandas
- Working with arrays and dataframes

33. Matplotlib and Seaborn

- Creating visualizations with Matplotlib
- Advanced visualizations with Seaborn

34. Practice: Analyzing and visualizing data with NumPy, pandas, Matplotlib, and Seaborn

Week 11: Web Development with Flask

35. Introduction to Flask

- Setting up Flask
- Creating a simple Flask application

36. Routes and Templates

- Defining routes
- Using templates for dynamic content

37. Forms and User Input

- Handling forms and user input

38. Practice: Building a simple web application with Flask

Week 12: Web Scraping

39. Introduction to Web Scraping

- Understanding web scraping
- Setting up BeautifulSoup and requests

40. Scraping Data

- Extracting data from web pages
- Navigating HTML with BeautifulSoup

41. Practice: Scraping data from websites, extracting and processing information

Week 13: Working with APIs

42. Understanding APIs

- What are APIs and how they work
- Making API requests with requests

43. Processing API Responses

- Handling JSON data
- Working with popular APIs (e.g., OpenWeather, Twitter)

44. Practice: Making API requests, processing and using API data

Week 14: Testing and Debugging

45. Unit Testing

- Introduction to unit testing
- Writing tests with unittest and pytest

46. Debugging Techniques

- Common debugging techniques
- Using debugging tools

47. Practice: Writing unit tests, debugging Python scripts

Week 15: Final Project and Review

48. Final Project

- Planning and developing a comprehensive project
- Implementing learned concepts

49. Review of Key Concepts

- Recap of important topics
- Common pitfalls and best practices

50. Project Presentation

- Presenting and documenting the final project