

Python Programming: A Comprehensive Guide

Generated by AI

August 1, 2025

Contents

1	Lesson 1: Introduction to Python	2
2	Lesson 2: Variables and Data Types	2
3	Lesson 3: Control Structures	2
4	Lesson 4: Functions	3
5	Lesson 5: Lists and Dictionaries	3
6	Lesson 6: File Handling	4
7	Lesson 7: Error Handling	4
8	Lesson 8: Object-Oriented Programming (OOP)	5
9	Lesson 9: Libraries and Modules	5
10	Lesson 10: Advanced Topics	5

1 Lesson 1: Introduction to Python

Objective

Understand the basics of Python and set up your environment.

Topics Covered

- What is Python and why it is widely used.
 - Installing Python (e.g., from python.org or Anaconda).
 - Using IDEs like VS Code, PyCharm, or Jupyter Notebook.
 - Running your first Python script:

```
1 print("Hello, world!")
```

Exercise

Write a Python script that prints your name.

2 Lesson 2: Variables and Data Types

Objective

Learn about variables and basic data types in Python.

Topics Covered

- Declaring variables:

```
1 x = 10 y = "Python" z = 3.14
```

- Common data types: integers, floats, strings, booleans.
- Type conversion using `int()`, `float()`, `str()`.

Exercise

Create variables for your age, name, and favorite number, then print them.

3 Lesson 3: Control Structures

Objective

Learn how to use conditional statements and loops.

Topics Covered

- If-else statements:

```
1 x = 10 if x > 5:
2     print("x is greater than 5")
3 else:
4     print("x is less than or equal to 5")
```

- Loops: for and while. Example:

```
1 for i in range(5):
2     print(i)
```

Exercise

Write a program that prints all even numbers from 1 to 20.

4 Lesson 4: Functions

Objective

Learn how to define and use functions in Python.

Topics Covered

- Defining functions:

```
1 def greet(name):
2     return f"Hello, {name}!"
```

- Function arguments and return values.
- Default arguments and keyword arguments.

Exercise

Write a function that calculates the factorial of a number.

5 Lesson 5: Lists and Dictionaries

Objective

Learn about lists and dictionaries in Python.

Topics Covered

- Lists:

```
1 fruits = ["apple", "banana", "cherry"] print(fruits[0]) # Output
2 :
apple
```

- List operations: append, remove, slicing.
- Dictionaries:

```
1 person = {"name": "Alice", "age": 25} print(person["name"]) #  
2 Output: Alice
```

Exercise

Create a list of your favorite movies and a dictionary of your personal details.

6 Lesson 6: File Handling

Objective

Learn how to read from and write to files.

Topics Covered

- Reading from a file:

```
1 with open("file.txt", "r") as file:  
2     content = file.read()  
3     print(content)
```

- Writing to a file:

```
1 with open("file.txt", "w") as file:  
2     file.write("Hello, Python!")
```

Exercise

Write a program that reads a file and counts the number of lines.

7 Lesson 7: Error Handling

Objective

Learn how to handle errors and exceptions.

Topics Covered

- Try-except blocks:

```
1 try:  
2     result = 10 / 0  
3 except ZeroDivisionError:  
4     print("Cannot divide by zero!")
```

- Raising exceptions with raise.

Exercise

Write a program that asks the user for input and handles invalid input gracefully.

8 Lesson 8: Object-Oriented Programming (OOP)

Objective

Learn the basics of OOP in Python.

Topics Covered

- Classes and objects:

```
1 class Dog:
2     def __init__(self, name):
3         self.name = name
4
5     def bark(self):
6         print(f"{self.name} says woof!")
```

- Inheritance and polymorphism.

Exercise

Create a class `Car` with attributes like `model` and `year`.

9 Lesson 9: Libraries and Modules

Objective

Learn how to use Python libraries and modules.

Topics Covered

- Importing modules:

```
1 import math print(math.sqrt(16)) # Output: 4.0
```

- Popular libraries: NumPy, Pandas, Matplotlib.

Exercise

Use the `random` module to generate a random number between 1 and 100.

10 Lesson 10: Advanced Topics

Objective

Explore advanced Python features.

Topics Covered

- List comprehensions:

```
1 squares = [x**2 for x in range(10)]
```

- Lambda functions:

```
1 add = lambda x, y: x + y print(add(2, 3)) # Output: 5
```

- Decorators and generators.

Exercise

Write a list comprehension to create a list of the first 10 cubes.