Learning Python

Curated RAG-Friendly Content

TeachMate AI Agent – Knowledge Base Upload

Prepared for: Retrieval■Augmented Generation (RAG) demos

Author: ChatGPT (OpenAI o3)

1. Introduction to Python

Python is a high■level, general■purpose programming language created by Guido van Rossum and first released in 1991.

Why Python?

- Simple, readable syntax

- Large standard library & vibrant ecosystem

- Cross■platform (Windows, macOS, Linux)

- Ideal for data science, web development, automation, and more

Python executes line by line (interpreted) but can be compiled to byte■code for performance.

2. Getting Started

1. Install Python 3.x from python.org or via package managers (e.g., Homebrew, apt).

2. Verify installation: $ python --version

3. Use an IDE or text editor (VS Code, PyCharm) OR the built■in REPL by typing $ python

4. Create virtual environments to isolate dependencies:

```
$ python -m venv venv   &&   source venv/bin/activate (Unix)

venv\Scripts\activate   (Windows)
```

3. Variables & Data Types

- Numbers: int, float, complex

- Boolean: True / False

- Strings: immutable sequences of Unicode characters

- Lists: ordered, mutable collections [1, 'a', 3]

- Tuples: ordered, immutable collections (1, 2)

- Sets: unordered, unique elements {1, 2, 3}

- Dicts: key■value pairs {'name': 'Ada', 'age': 30}

Python uses dynamic typing: variable types are determined at runtime.

4. Control Flow

Conditional statements:

```
if condition:
    ...
elif other_condition:
    ...
else:
    ...
```

Loops:

```
for item in iterable:
    ...
while condition:
    ...
```

Use 'break' to exit, 'continue' to skip to next iteration.

5. Functions & Modules

Define reusable blocks with def:

```
  def add(a, b=0):

      """Return the sum of a and b."""

      return a + b
```

Everything in Python is an object; functions are first■class.

Import modules:

```
  import math

  from pathlib import Path
```

Create your own modules (files ending in .py) and packages (folders with __init__.py).

6. Core Data Structures

List comprehensions:

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

Dictionary comprehensions:

```
  mapping = {c: ord(c) for c in 'abc'}
```

Built■in functions: len(), sorted(), enumerate(), zip(), map(), filter().

7. Object■Oriented Programming (OOP)

Python supports classes & multiple inheritance.

```
class Animal:
    def __init__(self, name):
        self.name = name
    def speak(self):
        raise NotImplementedError

class Dog(Animal):
    def speak(self):
        return 'Woof!'
```

Key concepts: encapsulation, inheritance, polymorphism, composition.

8. Popular Libraries

- NumPy & Pandas: data manipulation

- Matplotlib & Seaborn: visualization

- scikit■learn: machine learning

- FastAPI, Django, Flask: web frameworks

- LangChain, OpenAI, transformers: LLM & NLP

- Streamlit & Gradio: data apps & demos

9. Next Steps & Resources

• Official Docs: docs.python.org/3/

• Interactive Tutorials: realpython.com | learnpython.org

• Books: 'Automate the Boring Stuff with Python', 'Fluent Python'

• Community: /r/Python, Python Discord, local meetups

Project Ideas:

- Build a CLI todo app

- Create a Streamlit dashboard

- Automate file organization

- Train a small ML model with scikit■learn

Happy Coding!