

Part 1 - Introduction to Developing Python Programs with ChatGPT

Yahya Nazer

2026.06.18

Table of contents

Part 1 - Introduction to Developing Python Programs with ChatGPT	2
Purpose of This Part	2
Learning Objectives	3
Chapter 1 - The Standard Project Folder Structure	3
Objective of This Chapter	3
User Folder Structure	3
Folder Definitions	4
A-Data	4
B-Engine	4
C-Results	4
D-Documentation	4
ChatGPT Script for This Chapter	5
What ChatGPT Should Explain	5
Exercise	5
Chapter Checklist	5
What You Learned	6
Chapter 2 - Installing Python and VS Code	6
Objective of This Chapter	6
Installing Python	6
Installing VS Code	6
Starting VS Code	7
Step 1	7
Step 2	7
Step 3	7
Step 4	7
Step 5	7
ChatGPT Script for This Chapter	7
Exercise	8
What You Learned	8
Chapter 3 - Understanding ChatGPT Scripts	8
Objective of This Chapter	8
Poor Script Example	8
Better Script Example	8
Standard ChatGPT Script Template	9
Exercise	9
Chapter Checklist	10

What You Learned	10
Chapter 4 - Your First Python Program Generated by ChatGPT	10
Objective of This Chapter	10
ChatGPT Script	10
ChatGPT Generated Python Code	11
Save the Program	11
Run the Program in VS Code	11
Exercise	11
Chapter Checklist	11
What You Learned	12
Chapter 5 - Improving the ChatGPT Script	12
Objective of This Chapter	12
Improvement Request to ChatGPT	12
Improved ChatGPT Python Code	12
Run the Improved Program	13
What Changed?	13
Exercise	13
Chapter Checklist	14
What You Learned	14
Chapter 6 - Asking ChatGPT to Explain the Code	14
Objective of This Chapter	14
Explanation Script	14
Example Code to Explain	14
What ChatGPT Should Explain	14
Exercise	15
Chapter Checklist	15
What You Learned	15
Chapter 7 - Asking ChatGPT to Debug Errors	15
Objective of This Chapter	15
Debugging Script	15
Example Error	16
Corrected Code	16
Exercise	16
Chapter Checklist	16
What You Learned	16
Chapter 8 - Part 1 Review	16
The Main Workflow	16
Part 1 Skills Checklist	17
Summary	17
Looking Ahead	17

Part 1 - Introduction to Developing Python Programs with ChatGPT

Purpose of This Part

This course is different from a traditional Python course.

The goal is not to memorize Python syntax.

The goal is to learn how to work with ChatGPT to create Python programs.

In every chapter, you will learn this workflow:

```
I have an idea
  ↓
I write a clear ChatGPT script
  ↓
ChatGPT generates Python code
  ↓
I test the code in VS Code
  ↓
I improve the ChatGPT script
  ↓
ChatGPT improves the Python code
  ↓
I have a working program
```

By the end of this part, you will understand:

- How to organize your project folders
 - How to start VS Code
 - How to write a good ChatGPT script
 - How to review generated Python code
 - How to test programs
 - How to improve programs using ChatGPT
-

Learning Objectives

After completing this part, you will be able to:

- Create a project folder structure
 - Open a project in VS Code
 - Write a clear ChatGPT script
 - Generate Python code with ChatGPT
 - Save Python code in the correct folder
 - Run Python code in VS Code
 - Modify a ChatGPT script
 - Improve Python code using ChatGPT
-

Chapter 1 - The Standard Project Folder Structure

Objective of This Chapter

Before writing Python programs, you need a simple and repeatable folder structure.

This makes it easier to keep your data, Python programs, results, and documentation organized.

User Folder Structure

For every project in this course, use this structure:

Project-Name

A-Data

B-Engine

C-Results

D-Documentation

Folder Definitions

A-Data

This folder contains input files.

Examples:

```
Excel files
CSV files
Images
Documents
Text files
PDF files
```

B-Engine

This folder contains Python programs.

Examples:

```
Hello-World.py
File-Copy-Tool.py
Excel-Analyzer.py
Photo-Organizer.py
```

C-Results

This folder contains files created by your Python programs.

Examples:

```
Reports
Processed images
Generated Excel files
Generated DOCX files
Generated PDF files
Log files
```

D-Documentation

This folder contains notes and instructions.

Examples:

```
README files
User guides
QMD files
Project notes
```

ChatGPT Script for This Chapter

Copy this script into ChatGPT:

```
I am learning how to use ChatGPT to develop Python programs.
```

```
Please explain this folder structure in simple language:
```

```
Project-Name  
+- A-Data  
+- B-Engine  
+- C-Results  
+- D-Documentation
```

```
Explain:
```

- ```
1. What each folder is for.
2. Why this structure is useful.
3. Where Python files should go.
4. Where input files should go.
5. Where output files should go.
```

```
Make the explanation beginner friendly.
```

---

## What ChatGPT Should Explain

ChatGPT should explain that:

- A-Data stores input files.
  - B-Engine stores Python scripts.
  - C-Results stores outputs.
  - D-Documentation stores notes and user guides.
- 

## Exercise

Create this folder structure on your computer:

```
ChatGPT-Python-Course
 A-Data
 B-Engine
 C-Results
 D-Documentation
```

---

## Chapter Checklist

- Created main project folder
- Created A-Data
- Created B-Engine
- Created C-Results

- Created D-Documentation
  - Asked ChatGPT to explain the folder structure
- 

## What You Learned

You learned how to organize your Python projects before writing code.

---

## Chapter 2 - Installing Python and VS Code

### Objective of This Chapter

In this chapter, you will install the two main tools used in this course:

- Python
- VS Code

Python runs your programs.

VS Code is where you write and test your programs.

---

### Installing Python

Visit:

```
https://www.python.org/downloads/
```

Download and install Python.

After installation, open a terminal and test:

```
python --version
```

On some Mac computers, use:

```
python3 --version
```

Expected result:

```
Python 3.x.x
```

---

### Installing VS Code

Visit:

```
https://code.visualstudio.com/
```

Install VS Code.

Then install the Python extension inside VS Code.

---

## Starting VS Code

### Step 1

Open VS Code.

### Step 2

Select:

```
File
→ Open Folder
```

### Step 3

Select your project folder.

Example:

```
ChatGPT-Python-Course
```

### Step 4

Open the B-Engine folder.

### Step 5

Create a new Python file:

```
Hello-World.py
```

---

## ChatGPT Script for This Chapter

Copy this script into ChatGPT:

```
I am a beginner.

Please explain how to start a Python project in VS Code.

Assume I already created this folder structure:

Project-Name
+- A-Data
+- B-Engine
+- C-Results
+- D-Documentation

Explain:
1. How to open the main project folder in VS Code.
2. How to create a Python file in B-Engine.
3. How to open the VS Code terminal.
4. How to run a Python program.
5. What to do if python does not work but python3 works.

Make it simple and step-by-step.
```

## Exercise

In VS Code:

- Open your project folder
  - Open B-Engine
  - Create `Hello-World.py`
  - Open Terminal
  - Test Python version
- 

## What You Learned

You learned how to start a project in VS Code and prepare to run Python code.

---

# Chapter 3 - Understanding ChatGPT Scripts

## Objective of This Chapter

A ChatGPT script is the instruction you give ChatGPT.

A good script produces better Python code.

A poor script produces poor Python code.

In this course, the word “script” means:

The instruction you write to ChatGPT so ChatGPT can generate Python code for you.

---

## Poor Script Example

```
Write a Python program.
```

This is too vague.

Problems:

- No purpose
  - No folder structure
  - No requirements
  - No comments requested
  - No explanation requested
  - No testing instructions
- 

## Better Script Example

```
Write a Python program.
```

**Purpose:**

```
Ask the user for their name.
```

**Requirements:**

1. Use `input()`.
2. Print a welcome message.
3. Add comments explaining every important line.
4. Make it suitable for beginners.
5. Explain how to run it in VS Code.

This is better because it tells ChatGPT exactly what you need.

---

## Standard ChatGPT Script Template

Use this template throughout the course:

Write a Python program.

Project folder structure:

Top Folder

+ A-Data

+ B-Engine

+ C-Results

+ D-Documentation

Program purpose:

[Explain what the program should do.]

Requirements:

1. [Requirement 1]

2. [Requirement 2]

3. [Requirement 3]

Coding style:

1. Use beginner-friendly Python.

2. Add comments to explain the code.

3. Use UTF-8 when reading or writing files.

4. Use clear variable names.

5. Add error handling where useful.

Output:

[Explain what the program should create or display.]

Please also explain how to run the program in VS Code.

---

## Exercise

Write a ChatGPT script for a program that:

- Asks for your name
  - Asks for your city
  - Prints a short welcome message
-

## Chapter Checklist

- Understand what a ChatGPT script is
  - Understand why vague prompts are weak
  - Understand the standard script template
  - Wrote your first structured ChatGPT script
- 

## What You Learned

You learned that writing a clear ChatGPT script is the first step in developing Python programs with ChatGPT.

---

# Chapter 4 - Your First Python Program Generated by ChatGPT

## Objective of This Chapter

You will ask ChatGPT to generate your first Python program.

The program will:

- Ask for your name
  - Print a welcome message
- 

## ChatGPT Script

Copy this script into ChatGPT:

```
Write a Python program.
```

```
Project folder structure:
```

```
Top Folder
```

```
+-- A-Data
```

```
+-- B-Engine
```

```
+-- C-Results
```

```
+-- D-Documentation
```

```
Program purpose:
```

```
Ask the user for their name and display a welcome message.
```

```
Requirements:
```

```
1. Use input().
```

```
2. Use print().
```

```
3. Add comments explaining every important line.
```

```
4. Make it beginner friendly.
```

```
5. The file will be saved in B-Engine as Hello-World.py.
```

```
Output:
```

```
Display a welcome message using the user's name.
```

```
Please also explain how to run the program in VS Code.
```

---

## ChatGPT Generated Python Code

ChatGPT should generate code similar to this:

```
=====
Program Name:
Hello-World.py

Purpose:
Ask the user for their name and display a welcome message.
=====

Ask the user to enter their name.
name = input("Enter your name: ")

Display a welcome message using the name.
print(f"Hello {name}. Welcome to Python.")
```

---

## Save the Program

Save the code as:

```
B-Engine/Hello-World.py
```

---

## Run the Program in VS Code

Open Terminal in VS Code:

```
Terminal
→ New Terminal
```

Run:

```
python Hello-World.py
```

On Mac, if needed:

```
python3 Hello-World.py
```

Example output:

```
Enter your name: Yahya
Hello Yahya. Welcome to Python.
```

---

## Exercise

Run the program three times using different names.

---

## Chapter Checklist

- Copied the ChatGPT script
- Generated Python code

- Saved code in B-Engine
  - Ran code in VS Code
  - Tested the program
- 

## What You Learned

You learned how to go from a ChatGPT script to a working Python program.

---

## Chapter 5 - Improving the ChatGPT Script

### Objective of This Chapter

Most programs are improved in steps.

You rarely get the final version on the first try.

The workflow is:

```
Script to ChatGPT
 ↓
Python code
 ↓
Test
 ↓
Improve script
 ↓
Better Python code
```

---

### Improvement Request to ChatGPT

After your first program works, ask ChatGPT:

```
Please modify the program.
```

```
New requirements:
```

1. Ask for the user's name.
  2. Ask for the user's city.
  3. Print both values in a friendly message.
  4. Keep the code beginner friendly.
  5. Add comments explaining every important line.
- 

### Improved ChatGPT Python Code

ChatGPT should generate code similar to this:

```
=====
Program Name:
Hello-World.py
#
Purpose:
```

```
Ask the user for their name and city,
then display a friendly message.
=====

Ask the user for their name.
name = input("Enter your name: ")

Ask the user for their city.
city = input("Enter your city: ")

Display a friendly message using the name and city.
print(f"Hello {name}.")
print(f"It is nice to know that you live in {city}.")
```

---

## Run the Improved Program

Run:

```
python Hello-World.py
```

or:

```
python3 Hello-World.py
```

Example:

```
Enter your name: Yahya
Enter your city: Toronto
Hello Yahya.
It is nice to know that you live in Toronto.
```

---

## What Changed?

The improved program added:

- A second input
  - A second variable
  - More output text
  - Better comments
- 

## Exercise

Ask ChatGPT to modify the program again.

New requirements:

- Ask for favorite food
  - Ask for favorite color
  - Print a friendly sentence using all answers
-

## Chapter Checklist

- Improved the original ChatGPT script
  - Generated improved Python code
  - Tested the improved program
  - Compared old and new code
- 

## What You Learned

You learned that developing with ChatGPT is an interactive process.

---

## Chapter 6 - Asking ChatGPT to Explain the Code

### Objective of This Chapter

When ChatGPT generates code, you should not just copy it.

You should ask ChatGPT to explain it.

This helps you learn Python faster.

---

### Explanation Script

Copy this into ChatGPT after it generates code:

```
Please explain this Python code line by line.
```

```
Use beginner-friendly language.
```

```
For each line explain:
```

1. What the line does.
2. Why it is needed.
3. What would happen if it were removed.

```
Here is the code:
```

```
[paste the Python code here]
```

---

### Example Code to Explain

```
name = input("Enter your name: ")
print(f"Hello {name}. Welcome to Python.")
```

---

### What ChatGPT Should Explain

ChatGPT should explain:

- `input()` asks the user to type something.

- `name` stores what the user typed.
  - `print()` displays output.
  - `f"Hello {name}"` inserts the value of `name` into the message.
- 

## Exercise

Ask ChatGPT to explain your improved program with name, city, food, and color.

---

## Chapter Checklist

- Asked ChatGPT to explain code
  - Learned what `input()` does
  - Learned what `print()` does
  - Learned what a variable is
  - Learned what an f-string is
- 

## What You Learned

You learned how to use ChatGPT as a Python tutor, not just a code generator.

---

# Chapter 7 - Asking ChatGPT to Debug Errors

## Objective of This Chapter

Python errors are normal.

When an error happens, you can ask ChatGPT to help.

The important skill is to paste the full error message.

---

## Debugging Script

Use this script when your program fails:

```
My Python program has an error.
```

```
Please help me debug it.
```

```
Explain:
```

1. What the error means.
2. Which line caused the error.
3. How to fix it.
4. How to avoid it next time.

```
Here is my code:
```

```
[paste code here]
```

Here is the full error message:

[paste error message here]

---

## Example Error

Code:

```
name = input("Enter your name: ")
print(f"Hello {nam}.")
```

This may create an error because `nam` is misspelled.

---

## Corrected Code

```
name = input("Enter your name: ")
print(f"Hello {name}.")
```

---

## Exercise

Create a small mistake in your code and ask ChatGPT to debug it.

Examples:

- Misspell a variable name
  - Remove a quote
  - Remove a parenthesis
- 

## Chapter Checklist

- Understand that errors are normal
  - Know how to copy full error messages
  - Used the debugging script
  - Fixed a simple error
- 

## What You Learned

You learned how to use ChatGPT as a debugging assistant.

---

## Chapter 8 - Part 1 Review

### The Main Workflow

The most important lesson in Part 1 is this workflow:

```
Objective
 ↓
ChatGPT Script
 ↓
Generated Python Code
 ↓
Save in B-Engine
 ↓
Run in VS Code
 ↓
Test
 ↓
Improve Script
 ↓
Improve Code
```

---

## Part 1 Skills Checklist

By the end of this part, you should be able to:

- Create the course folder structure
  - Open a project in VS Code
  - Create a Python file in B-Engine
  - Write a structured ChatGPT script
  - Ask ChatGPT to generate Python code
  - Save the code correctly
  - Run the code
  - Ask ChatGPT to improve code
  - Ask ChatGPT to explain code
  - Ask ChatGPT to debug code
- 

## Summary

In Part 1, you learned the foundation of the entire course.

You learned that the course is not only about Python.

It is about using ChatGPT as a development partner.

The key skill is learning how to write a clear script to ChatGPT.

---

## Looking Ahead

In Part 2, you will learn how to use ChatGPT scripts to build beginner Python programs using:

- Variables
- Text
- Numbers
- User input
- Simple calculations
- Conditions

Each chapter will begin with the objective, then the ChatGPT script, then the generated Python code, then testing and improvement.