**Data developer using ChatGPT as an assistant: here 5 effective prompt templates with concrete use cases** for coding, including tips for getting the best responses.

## 1️. ****Code Generation / Boilerplate Creation****

**Use Case:** You need to quickly generate Python scripts, functions, or boilerplate code for a data pipeline.

**Prompt Template:**

"Generate a Python function that [describe functionality].

Include proper docstrings, type hints, and error handling.

Example input: [example]. Example output: [example]."

**Example:**

Generate a Python function that reads a CSV file, filters rows where "age" > 30, and returns the result as a Pandas DataFrame.

Include docstrings and error handling.

Example input: data.csv

Example output: DataFrame with filtered rows.

**Why it works:** Provides **context, examples, and constraints**, so ChatGPT produces usable code on the first try.

## 2. ****Debugging / Code Fixing****

**Use Case:** You have a snippet of code throwing errors or producing wrong outputs.

**Prompt Template:**

"I have the following Python code: [paste code].

It gives me [describe error or wrong output].

Explain the error and provide a corrected version of the code."

**Example:**

I have the following code:

```python

import pandas as pd

df = pd.read\_csv('data.csv')

df['age'] = df['age'].astype(int)

It gives me a ValueError because some values in 'age' are missing.  
Explain the error and provide a corrected version of the code that handles missing values.

**Why it works: ChatGPT analyzes the **code + error context** , producing an **explanation + fix**.**

---

**3. Code Optimization / Refactoring**

**Use Case: You want to improve performance, readability, or efficiency.**

**Prompt Template:**

"Refactor the following Python code to be more efficient and readable.  
Explain the changes you made:  
[insert code here]"

**Example:**

Refactor the following Python code to improve efficiency:

result = []

for i in range(len(my\_list)):

if my\_list[i] % 2 == 0:

result.append(my\_list[i])

Explain your changes.

**Why it works:** Encourages ChatGPT to **explain improvements,** teaching best practices along the way.

--------------------------------------------------------------------------------

**4. Code Optimization / Refactoring**

**Use Case:** You need code for analyzing datasets or generating plots.

**Prompt Template:**

"I have a dataset with columns: [list columns].  
Write Python code using Pandas and Matplotlib/Seaborn to analyze the data and visualize [specific insight].  
Include comments explaining each step."

**Example:**

I have a dataset with columns: 'date', 'sales', 'region'.  
Write Python code to visualize monthly sales trends by region using Seaborn.  
Include comments explaining each step.

**Why it works:** Gives ChatGPT clear domain + visualization goal, producing ready-to-run analysis code.

**5. Learning/ Documentation Assistant**

**Use Case:** You want to understand unfamiliar libraries, APIs, or complex functions.

**Prompt Template:**

"Explain what [function/library/method] does in Python.  
Give examples of usage, common pitfalls, and best practices.  
Keep it concise but informative."

**Example:**

Explain what the Pandas 'groupby' method does.  
Provide examples for grouping sales data by region and calculating the mean.  
Include common mistakes to avoid.

**Why it works:** Converts documentation into \*\*readable, example-driven explanations\*\*, saving time.

---

**Pro Tip:** For coding prompts, always provide:

- **Context** (what you are working on)

**- Input/output examples**

**- Constraints or preferences** (language version, libraries, performance)

- Activate the **ADA package** in Settings of ChatGPT Plus

This makes ChatGPT act like a **smart coding assistant,** not just a generic code generator.