- **Which AI prompting strategies were most effective**

In *Intentional Prompting Strategies*, I would say that '*asking the AI to explain its code before accepting it'* was the most powerful technique.

For example, in one error we discovered a mismatch between the left side and the right side of the code. The left side only tried to unpack two variables, but the right side was designed to return three values. My AI suggested fixing the left side so that the program would return three elements: `umbrella_needed`, `rain_expectation`, and `details`.

However, I realised that I did not need `rain_expectation` — two elements were enough. Because I had asked the AI to explain what was happening, I understood the situation and was able to request a different fix: adjusting the right side to return only two elements.

This was very important for keeping control of the design in human hands, not in AI's hands.

- **How you improved upon the AI's initial solutions**

Identifying Errors and Refining
AI's first code version sometimes had mismatched variables or incomplete handling of user input. I pointed this out (e.g., the umbrella/rain expectation mismatch) and asked me to fix it.

c.f.,The umbrella/rain expectation mismatch is;
A user asks "Should I take an umbrella tomorrow?" and expects a yes/no answer, while the program gave only "Rain probability: 40%", without turning that into umbrella advice.

Clarifying User Interaction
I prompted my AI to make the console output clearer so I could actually see and test the program. This gave me visibility.

- **What you learned about effectively collaborating with AI tools**

I learned that seeing the console is a fundamental step, even though the Intentional Prompting Strategies (IPS) are powerful.

The IPS strategies helped me greatly, especially breaking down complex problems into smaller steps and asking the AI to explain its code before accepting it.

However, before I could apply these strategies, I needed to see the console output. For example, when an error occurred, I had to observe the "diagnosis" in the console. Without seeing the output, it was difficult to understand what was happening. As a result, I could neither break down the problem nor fully grasp the AI's explanation.

Seeing is therefore the foundation of all effective collaboration with AI.

(354 words)