DWA_01.3 Knowledge Check_DWA1

1. Why is it important to manage complexity in Software?

Managing complexity helps a developer maintain a clear understanding of the various processes involved in creating a piece of software. This allows anyone involved in the project to work on it with greatest efficacy.

2. What are the factors that create complexity in Software?

Code is complex by nature. Therefore, if a developer is not proactive in attempting to mitigate some of that complexity, problems can begin to develop very quickly.
This can occur either from the very start of a project or as a result of constant updates being made to the code without the necessary steps being taken to manage the growing complexity.

3. What are ways in which complexity can be managed in JavaScript?

The following is a list of good practices that can be implemented in any code:

1. Make sure variable names are distinct enough to not get confused with one another
2. Variable names should be aligned with their function: isSomething for booleans, getSomething for retrieving things, etc.
3. In areas where the code could potentially not execute as intended, throw error messages to catch potential problems
4. Keep the code modular by implementing OOP and/or functional programming. This will help in keeping the code specific, readable and reusable.
5. Utilise abstraction to once again keep the code clean, readable and specific.

6. Document the various aspects of the code that might not be intuitive.

Note that it is not entirely necessary to apply all of these things to every part of the code. A developer should use their knowledge and discretion to decide when this is applicable and when it isn’t.

---

4. Are there implications of not managing complexity on a small scale?

As scale is relative, it honestly depends. I think it is always valuable to mitigate as many potential issues as possible, as long as it’s necessary. Once again, the developer should make that call.

---

5. List a couple of codified style guide rules, and explain them in detail.

1. Use “const” when declaring every variable unless you know it will need to be updated at a later stage. This will ensure that your variables aren’t accidentally changed.
2. Utilise the strict equality operator “===” wherever possible to avoid type coercion and other unexpected issues.

---

6. To date, what bug has taken you the longest to fix - why did it take so long?

I think the longest one was making a specific element draggable on a webpage. The reason this took a long time is because I was working on code that I hadn’t written and therefore had to first understand what was being attempted before I could write my solution.
