DWA_01.3 Knowledge Check_DWA1

---

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

Complexity can make the software hard to understand and fix for other developers,
It increases the probability of bugs that will be harder to find because the code is hard to read and understand. Increased complexity can also make it hard to add new functionality to the program.

---

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

Technical debts from leaving hard coded and reused code unfixed. It will all collect over time and become a lot of technical debt.
Scaling
Size of the program.
Complexity of the program functions.

---

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

Abstraction, documentation, group by functionality, consistent styling and syntax - descriptive variables by using a style guide, modulation - global constants and naming. Try to make sure people don't have to go through all the code to understand a single function.

---

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

Will create bugs that will be harder to identify. Code is hard to edit and learn for new developers on the projects.

---

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

1. Commas should have a space after them.
   This increases code readability for lists, arrays etc.
2. For multi-line statements, use curly braces.
   It may lead to code bugs and is used to increase the readability of multi-line code blocks.
3. Multi-block empty lines are not allowed.
   While white space can be useful for separating logical sections of code, but excess white space can take up more screen space and result in more scrolling when reading through the code.
4. Add space between key-value pairs in objects.
   This is to enforce consistent spacing in object literals, to increase readability.
5. No duplicate keys in object liberals.
   This can cause bugs and misunderstandings when reading the code and when the code is executed.

---

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

For the IWA capstone project, when I created the function to change the dark/light theme by iterating through the day and night option then changing the theme accordingly. I put the if statement that checks what option has been selected inside the for of loop iterating through the options. This method caused the theme to change to day no matter what option I chose and not being able to change it back. I realised that every time the loop runs, it ends on the day option and only executes that condition so the dark theme is never executed, no matter which option you select. All I had to do was move the if statement out of the for loop.
