DWA_01.3 Knowledge Check_DWA1

---

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

It's important to manage complexity as it allows for the code to be more maintainable and accessible for not only yourself but for other potential developers that are modifying/working on the same code. The challenge however is not in making the code as complicated as possible but rather to keep it as simple as possible while doing complicated and complex things, managing the complexity makes that part a bit easier. What it boils down to is that the more complex the code gets the higher the developer frustration becomes and more bugs are unintentionally introduced.

---

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

Programming in and of itself is complex so the larger the scale of the programme the more complex it becomes, each new added part comes with its own little complexities. Adding more features or dimensions you weren't inherently accounting for, according to the client's request. Sometimes there is just not enough time to account for complexity in all aspects so some minor bugs seep through and more code would have to be created/changed in order to fix those problems.

---

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

Sticking to certain stylistic approaches, using comments instead of allowing for ambiguity, avoid redundancy, create more modular designs in the sense where the code gets broken down to different modules, for example scripts.js manages the the crux of the project and data.js retrieves and manipulates data which can be used in the scripts file instead of have everything in one big file.

---

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

Ultimately I think that the errors/bugs that were to arise wouldn't be that difficult to find or even get rid of but I think not starting to account for complexity in a small scale environment will ultimately open the door to bad habits of doing the same thing in larger scale environments.

---

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

Comments: It allows not only you to understand what is going on but more importantly for others as well, when everything is well commented the code explains itself and saves a heap of time in the long run.

Naming functions: Assigning the function to a variable with its own name, also having a descriptive name i.e. calcUserAge() by doing this one can immediately see what the function does/is supposed to do.

Keeping the code up to date: Try to use the most modern additions of JavaScript or the most uptodate accepted principles i.e. instead of using var use let and const. By doing this it increases the maintainability of the code for future coders as the readability is easier and the functionality is more efficient.

---

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

It wasn't necessarily a bug and it's not to say that I don't experience bugs but the thing that caused the most time wasted was misspelling the word length in a for loop and basically spending a good amount of time rewriting and changing code only to realize that the only problem was that I typed lenght instead of length

---