DWA_01.3 Knowledge Check_DWA1

---

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

It is extremely important to manage complexity as a software developer for a number of reasons:
1: When we start working with huge files containing thousands of lines of code, and we haven't written our code in an eloquent way, it is much harder to find and fix bugs. This also of course applies to smaller projects, but the stakes are not as high as:
2: the possibility of bringing an entire company to its knees because of one erroneous line of code. It is important to keep the complexity of our code under control because if we don't we fall into disaster.
3:A third reason is that we must remember that *we write code for other humans to read.* Readability is extremely important in the management of code complexity because our code will often be passed around to multiples other people who work on specific things in one big file.

"Any fool can write code that a computer can understand. Good programmers write code that humans can understand."

— Martin Fowler

---

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

- vague or ambiguous variable names
- related code is NOT grouped together
- no checks to prevent code from running
- not using comments to provide additional information
- The environment of programming is constantly evolving, which means that there are always new requirements to keep up with - it's a moving target

---

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

- Using JSDocs for explanations that someone else can access quickly instead of scrolling through 1000s of lines of code.
- Using a code style consistently
- Regular comments for explanations and clarity as to what the code is intended to do (document types and shapes) What are the expected shapes/values that can be in that object? What kinds of values are ALLOWED to be in that object? What does this function return?
- This makes code more predictable and clear.
- Modular and flexible code: making code easy to reuse. This also relates to abstraction, and what is mentioned in my answer to Q2: one should always group related code together. That is what makes it easy to reuse somewhere else. Another way one could achieve this would be by storing related code inside an object or function. This also achieves abstraction, because it makes the code much more concise and less messy and difficult to read.
- Use functional or object-oriented programming.
- Abstraction: reducing a large piece of code to make it more concise and manageable. “The key to building large software is to never build large software”. This means building little pieces of software and composing with these pieces. Also, DON’T over-explain! This also adds to complexity.

---

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

Yes. Programming is complex. Even if you are only working on a small to-do list app, if you wrote your code in a really complex way, the implication of that would be that you spend hours trying to debug one tiny piece of code, whereas if you had written eloquent code in the first place, that would never have happened. So yes, even in small projects, there are implications, such as time being wasted.

---

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

- Variable-related style rules: Don't separate your variables with commas. Keep all your consts together and all your lets together. Ensure that your variables look *visually* different at a glance. Make your variable names descriptive, not random!
- Have every property in an object begin on a new line.
- When nesting properties/objects/functions, indent accordingly:the deeper something is nested, the more indentations it should have.
- Use brackets if an if statement is more than a single line.
- Always write global constants using upper snake case.

---

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

There was an issue in the book-connect project where for some reason the books just wouldn't display when I clicked the show more button. Turns out I had two variables named books and book, and so because they were so similar, I accidentally used the wrong one in one of my functions.
