# DWA_01.3 Knowledge Check_DWA1

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

Managing complexity in software development is important first and foremost, to avoid potential bugs, improving functionality and increasing the overall quality of the code/software. It also makes it easier for other developers to read the code and be able to comprehend it.

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

The main factors that contribute towards complexity in software stem from human error -for example, using strings instead of numbers in a calculation or vice versa; using the wrong form of measurement or using the incorrect data values.

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

Keeping code as concise and simple as possible is one of the most obvious ways to manage complexity in Javascript. Avoiding unnecessary code and removing code that serves no purpose is extremely helpful. Furthermore, it can be helpful to instead of writing an overwhelming amount of code in one file, break it down into smaller sections of code before piecing everything together to build software.

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

Not managing complexity can prove to be catastrophic, therefore it is best to manage it on a small scale. Moreover, it's easier for developers to manage smaller sections at a time as opposed to trying to manage complexity in large quantities of code. Thus it is always in the best interest to manage complexity on a smaller scale.

---

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

-Airbnb is currently the most commonly used style guide for developers. Although it is considered to be quite a strict style guide, the main purpose of Airbnb Javascript style guide is to help developers maintain a consistent coding style. It's efficient, easy to read, and predictable. Some of the main cores of Airbnb include the use of const and let to assign variables accordingly (const to assign a variable that cannot be reassigned and let to assign variables that might be reassigned). More core rules include the usage of arrow functions, import/export modules, iterators, properties, blocks and comments -all to improve readability and ensure that code is to the point.
-Javascript Standard Style Guide has code fixer that allows users to format code automatically and detects bugs easily. It allows developers to write code without having to maintain multiple style configuration files and is therefore very simple. Some of the most important rules of Javascript Standard Style Guide includes using two spaces for indentation, using single quotes for strings, using curly braces for multi-line statements and avoiding multiple blank lines in between lines of code.
-Google Javascript Style Guide consists of two main rules -the first focusing on style rules (aesthetics and formatting) and the second being language rules (coding standards and conventions). Some of the main cores within these rules include using spaces instead of tabs, concatenating longer strings and using uppercase letters separated by underscores to name constants. Single quotes are also used instead of double quotes. Furthermore, horizontal alignment is generally discouraged and the preferred type of loop is the “for…of” loop as opposed to “for..in”.

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

Something I'm still struggling to debug is the final Capstone project for the Interactive Web App course. The problem I'm experiencing is that the app is failing to load any data -be it items in the library, the search bar functionality and being able to change the app's theme from light to dark. It's extremely frustrating because while most of the code I debugged as part of the assignment seems to be correct in terms of syntax, correct use of functions as well as using the import/export modules correctly, the data just won't load. It took me a week to debug the code that was provided and I was so sure I was on the right track, but then further down the line while reviewing my code, something in my mind clicked and I realized that I was completely off, thus forcing me to completely rewrite most of the code I debugged. This is both rewarding and frustrating -on one hand I was really excited that things were starting to make more sense and that I knew exactly how I had to correct my mistakes, time was limited and so there was a lot of pressure. To this day I haven't quite figured out exactly why the data won't load but after speaking to a peer after handing in, she told me she had the same problem and a few solutions she came up with to solve it, which ultimately led to her fixing the problem she had.