# DWA_01.3 Knowledge Check_DWA1

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

With complicated code, it’s easy to overlook simple errors that may render your code completely useless. Additionally, your code needs to be readable/understandable by other developers, so should be neat and organised.

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

Essentially coding involves people and people are complicated things. We all speak different languages, and we all have our own thought processes. Programmers all have their own solutions and while we can all get to understand each other, computers can’t make the logic leaps we can to “just understand the concept”. A computer can only understand the exact code we give it, and not what we think it should mean.

The complexity we deal with is created through all of our different languages and also our struggles to sometimes write exactly what we mean to.

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

Through using standardised coding styles when working on shared codebases or even just your own personal projects. This can include things like documenting with JSDocs, coming up with a variable naming scheme that involves more detailed variable names, and organising your blocks of code more efficiently.

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

Yes. If you do not learn to manage complexity at the small scale, you’ll never learn to do it at a larger scale. Additionally, all the small parts of a codebase add up and the effect of mismanagement and any errors it creates increases exponentially.

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

Global constants created in upper case snake. This helps set them apart and is a universal way of showing other developers that they are global constants.

No unused variables. This helps cut back on any unnecessary code.

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

The day/night theme in IWA19. I just couldn’t figure out the logic behind it even with the provided “baseline”. Eventually I realised that I was looking at the idea all wrong and assuming I needed to change something that seems impossible to change. I overcomplicated the solution and went down rabbit holes I didn’t need to because I didn’t think the simple solution was the “right” one.