DWA_01.3 Knowledge Check_DWA1

---

1. Why is it important to manage complexity in Software?
There are multiple reasons:

1. To ensure that devs including yourself can read the code and modify it effectively.
2. To ensure that errors can be clearly seen at a glance and traced to the source of the problem.

---

2. What are the factors that create complexity in Software?
Industry or problem related complexity: Depending on what the problem is that the code has to solve the complexity can increase due to the restraints or formatting the field the code is used in has or the complexity of what the code needs to output or manage.

Size and Collaboration: The size of the team working on the code can also increase complexity as multiple teams or members work on one program passing modules to each other. If the answer to the problem requires a large code base the code may also need to be scattered to increase readability which requires a bigger understanding of the whole code base and how each module may fit in.

The code itself may become complex due to it being recycled code or code which is acquired from third parties.

The inputs required for the software to run can also increase complexity if there are multiple inputs from either sensors or user inputs the code will have to manage a wider combination of data.

---

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

Separating code into smaller functional modules that are well documented can increase a coder's ability to isolate problems.

Documenting and using expressive simple language can help give each piece of code a unique identity which allows a dev or user to read what the program needs in order to function and the output they can expect from it.

Following a company or industry wide styling guide such as the AirBNB style or a company's internal formatting guide can help increase readability as the standard creates expectations which can be managed. Should an expectation fail to result in the required outcome the dev/user can identify what part of the software is associated with the outcome.

---

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

Yes the larger the software gets the more complex it will get and the more variables and inputs it will receive. If you manage the small scale code in a way that makes it identifiable it will be easier to understand if there is an overlap or shortcoming once the software scales or is outsourced.

At the end of the day a lot of code is open source or publicized which means that any developer should be able to understand what you create as the program may be recycled or upscaled into projects that are more complex than its original function. Thus the code needs to be universally readable to ensure each aspect of the code can be understood to its base implications.

---

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

Naming conventions: The use of explicit and simplistic names for different variables, objects and functions. Some examples include naming global variables in Upper Snake Case or using a descriptive name for functions such as handlerSubmitLoginForm. This allows us to identify what the code is at a glance as well as the expected outcome the code might have. It can also help us understand what data is saved under the name or what format it might be in.

Comma placement and semicolon capping: The use of commas or semicolons after a function or item. Always end your item or statement declaration on the same line so as to ensure that a user can determine the item is exactly as it is written in line. This is a simple rule that most languages adhere to to increase the readability and decrease confusion between when a statement should end or that the full name regardless of how abstract ends at this point.

For example: A large function which spans multiple lines and has multiple nested code elements should end with }; so a reader can identify that this function is closed at that point and any other code which follows is not directly related to the function.

The other case may be an object which has abstract items listed inside. By ensuring that at the end of each item a comma is placed on the same line the reader is made aware that that item and its value terminates there and that the following item starts on the next line. However readers often expect an item to exist on its own line and just as in english not to run on therefore an item that starts with a comma (eg: ,param1) will be assumed as a typo and may be corrected without appending the comma to the previous item.

---

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

My hardest bug to solve has been due to an array being created by a function without being printed visually and then modified to produce an extracted result. The reason it took so long was because I could not see what the original array looked like and make sure that the values which were printed were the ones we needed and in the right order. I had to solve the problem by extracting that specific array generator and printing it to identify that the array was being printed incorrectly.
