DWA_01.3 Knowledge Check_DWA1

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

It is important because it allows you to be able break down your code into smaller bits and also be able to fine tune any part without ruining the whole code.And also to make it readable for other programmers working on or with your code.

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

Size of codebase
Interdependencies and Coupling
Poorly Structured Code
Changing Requirements
Inconsistent or Incomplete Documentation

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

Reduce Nested Code
Consistent Coding Style
Proper Naming and Documentation
Modularization

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

Neglecting it while code is small leads to bugs accumulating and making it harder to debug resulting in wasted time used to go through the code line by line when all of this could have been avoided by reviewing the code on a smaller scale before integrating it into the whole project .

---

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

Prefer const Over let and var:

This rule encourages developers to use `const` for declaring variables unless the variable's value needs to be reassigned. It promotes immutability and helps prevent accidental reassignments, making the code more predictable and easier to reason about.

```
// Good: Using const for variables that don't need reassignment
const pi = 3.14;
const apiKey = "abc123";

// Bad: Using let when const can be used
let itemCount = 10;

// Good: Using let when the variable's value needs to change
let counter = 0;
counter += 1;
```

Indentation and Spacing:

Use consistent indentation (usually 2 or 4 spaces) to improve code readability.
It makes the code structure more organized and easier to follow. It helps developers quickly identify code blocks and nested structures. Proper spacing around operators and other elements improves the visual presentation and makes the code more readable.

Use Arrow Functions for Short Functions:

This rule suggests using arrow functions for concise function expressions, especially when the function has a short body. Arrow functions have a more concise syntax, lexical `this` binding, and avoid the need for explicit `return` statements for single expressions.

```
// Good: Using arrow function for short functions
const double = (x) => x * 2;
const greet = (name) => `Hello, ${name}!`;

// Bad: Using regular function for short functions
function multiply(a, b) {
  return a * b;
```

6. To date, what bug has taken you the longest to fix - why did it take so long?
Accessing specific data from an array within an array to assign the value to a newly created variable ,I learnt how to destructure and also use loops accompanied with `dataset` to be able to use that specific property or value within the object of arrays