DWA_04.3 Knowledge Check_DWA4

---

1. Select three rules from the Airbnb Style Guide that you find **useful** and explain why.

4.2 Use Array#push instead of direct assignment to add items to an array.

This style guide helps in terms of the readability of the code, rendering the developer more control, and simplified and shortened code syntax than what the direct assignment of items in the array would have rendered, which would be more prone to errors.

5.1 Use object destructuring when accessing and using multiple properties of an object. eslint: `prefer-destructuring`

Destructuring saves you from creating temporary references for those properties, and from repetitive access of the object. Repeating object access creates more repetitive code, requires more reading, and creates more opportunities for mistakes. Destructuring objects also provides a single site of definition of the object structure that is used in the block, rather than requiring reading the entire block to determine what is used.

6.3 When programmatically building up strings, use template strings instead of concatenation. eslint: `prefer-template` `template-curly-spacing`

This gives the developer more control without much effort as compared to when concatenation is used instead of the template literal. Concatenation requires precise considerations of spacing, line breaks, and use of punctuation. These are achieved organically with template strings, reducing the possible frequency of making mistakes as opposed to the counterpart method.

---

2. Select three rules from the Airbnb Style Guide that you find **confusing** and explain why.

7.12 Never mutate parameters. eslint: `no-param-reassign`

Why? Manipulating objects passed in as parameters can cause unwanted variable side effects in the original caller.
But what if a method is used to add a property in the object passed to it as a parameter each time a certain condition is met? The question suggests that there might be cases where it is imperative to mutate the object, what should then be done in that case? Also, is there any tolerance in peril?

13.6 Avoid using unary increments and decrements (`++`, `--`). eslint `no-plusplus`

Why? Per the eslint documentation, unary increment and decrement statements are subject to automatic semicolon insertion and can cause silent errors with incrementing or decrementing values within an application. It is also more expressive to mutate your values with statements like `num += 1` instead of `num++` or `num ++`. Disallowing unary increment and decrement statements also prevents you from pre-incrementing/pre-decrementing values unintentionally which can also cause unexpected behavior in your programs.

What is meant by being subjective to automatic semicolon insertion?

17.2 Don't use selection operators in place of control statements.

I do not understand why? Eg. !arrived && callPassenger(); discouraged.
