

DWA_04.3 Knowledge Check_DWA4

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

- **NAMING** - "Use camelCase for variables, functions, and object properties": This rule suggests using camelCase for naming variables, functions, and object properties. CamelCase involves starting each word with an uppercase letter except the first, without spaces or underscores.
- **DECLARATION** - "Always use const or let to declare variables": According to this rule, the Airbnb JavaScript Style Guide recommends using `const` or `let` keywords to declare variables instead of `var`. This is because `const` and `let` have block scope, which helps prevent unintended variable reassignments and improves code clarity. The use of `const` indicates that the variable is not intended to be reassigned, while `let` indicates that the variable can be reassigned if necessary.
- **QUOTES** - "Avoid backticks for single-line strings": The Airbnb style guide suggests avoiding the use of backticks (```) for single-line strings that don't contain template literals or string interpolation.

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

- (4.3) - The code initializes an empty array `itemsCopy`, creates a variable `len` to store the length of the `items` array, and uses a traditional `for` loop to iterate through the `items` array. Inside the loop, it assigns each item from the original array to the corresponding index of the new array.
- (5.2) - The code manually assigns each element of the `arr` array to separate variables (`first` and `second`). It uses array indexing (`arr[0]` and `arr[1]`) to access and assign the first and second elements of the array. This approach works but

requires multiple lines of code and can be less readable, especially when dealing with larger arrays or when extracting multiple elements.

- (7.4) - The code conditionally declares a function `test` inside an `if` statement. While this code may work, it can lead to unexpected behavior due to hoisting. In JavaScript, function declarations are hoisted to the top of their scope during the compilation phase. So, regardless of whether the condition (`currentUser`) is true or false, the function declaration is hoisted, and `test` will be accessible throughout the scope.

4.3 Use array spreads `...` to copy arrays.

```
// bad
const len = items.length;
const itemsCopy = [];
let i;

for (i = 0; i < len; i += 1) {
  itemsCopy[i] = items[i];
}

// good
const itemsCopy = [...items];
```

5.2 Use array destructuring. eslint: `prefer-destructuring`

```
const arr = [1, 2, 3, 4];

// bad
const first = arr[0];
const second = arr[1];

// good
const [first, second] = arr;
```

7.4 Note: ECMA-262 defines a `block` as a list of statements. A function declaration is not a statement.

```
// bad
if (currentUser) {
  function test() {
    console.log('Nope.');
  }
}

// good
let test;
if (currentUser) {
  test = () => {
    console.log('Yup.');
  };
}
```
