

DWA_04.3 Knowledge Check_DWA4

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

1. I find this rule to be useful because it shortens the code needed to create a deep copy of an array or object.

- 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];
```

2. I find this rule to be useful because it makes the code easier to read when you can just jump to where there are mutable variables and where there are constants.

- 13.3 Group all your `const` s and then group all your `let` s.

Why? This is helpful when later on you might need to assign a variable depending on one of the previously assigned variables.

```
// bad
let i, len, dragonball,
    items = getItems(),
    goSportsTeam = true;

// bad
let i;
const items = getItems();
let dragonball;
const goSportsTeam = true;
let len;

// good
const goSportsTeam = true;
const items = getItems();
let dragonball;
let i;
let length;
```

3. I find this rule to be useful because , the more blank lines in your code, then the larger the file and a person has to scroll way more than they would if blank lines are overused.

- 19.8 Do not pad your blocks with blank lines. eslint: `padded-blocks`

```
// bad
function bar() {

  console.log(foo);

}

// bad
if (baz) {

  console.log(quux);
} else {
  console.log(foo);

}

// bad
class Foo {
```

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

1. Why? I find that it doesn't affect the readability for me either way so I find this to be an unnecessary thing to enforce.

☰ README.md

// good
const foo = [1, 2, 3];
console.log(foo[0]);

• 19.12 Add spaces inside curly braces. eslint: `object-curly-spacing`

// bad
const foo = {clark: 'kent'};

// good
const foo = { clark: 'kent' };

• 19.13 Avoid having lines of code that are longer than 100 characters (including whitespace). Note: per [above](#), long strings are exempt from this rule, and should not be broken up. eslint: `max-len`

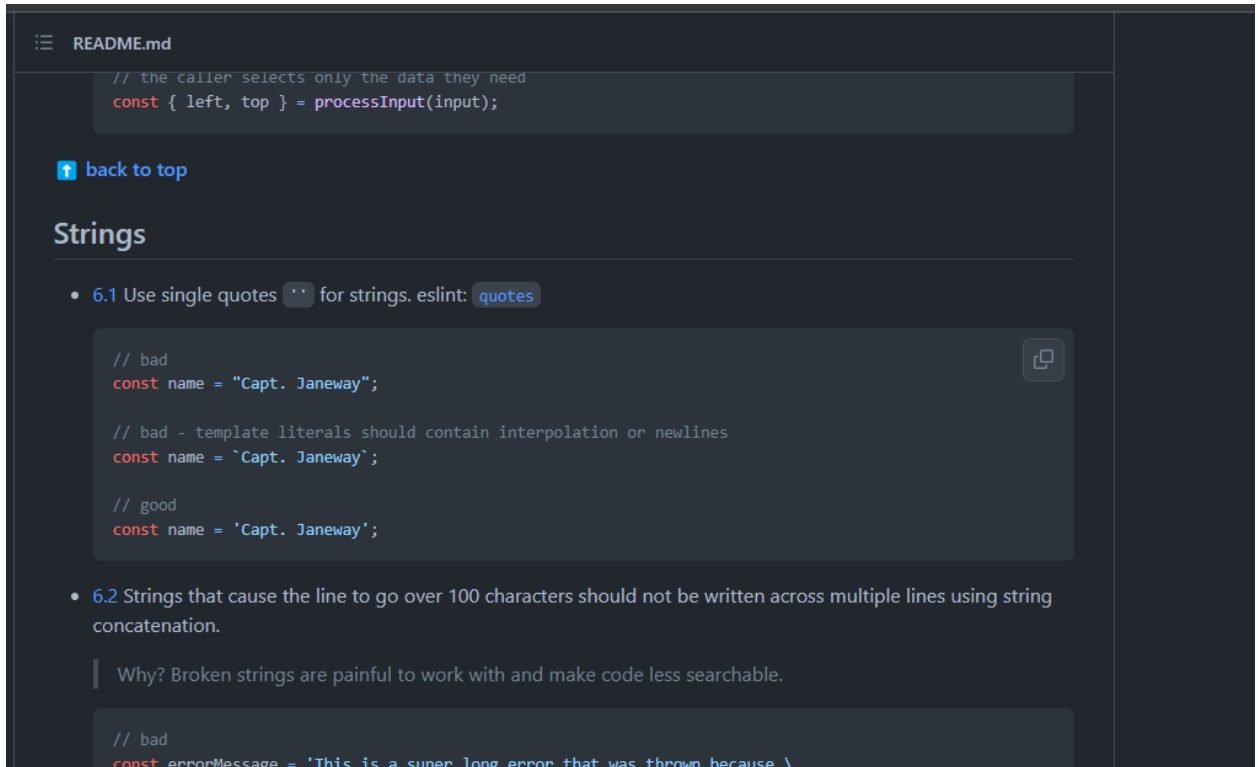
Why? This ensures readability and maintainability.

// bad
const foo = jsonData && jsonData.foo && jsonData.foo.bar && jsonData.foo.bar.baz && jsonData.foo.bar.baz.

// bad
\$.ajax({ method: 'POST', url: 'https://airbnb.com/', data: { name: 'John' } }).done(() => console.log('Cc

// good
const foo = jsonData

2. Why are single quotes preferred over double quotes for strings?



The screenshot shows a README.md file with the following content:

```
// the caller selects only the data they need
const { left, top } = processInput(input);
```

[back to top](#)

Strings

- 6.1 Use single quotes `'` for strings. eslint: `quotes`

```
// bad
const name = "Capt. Janeway";

// bad - template literals should contain interpolation or newlines
const name = `Capt. Janeway`;

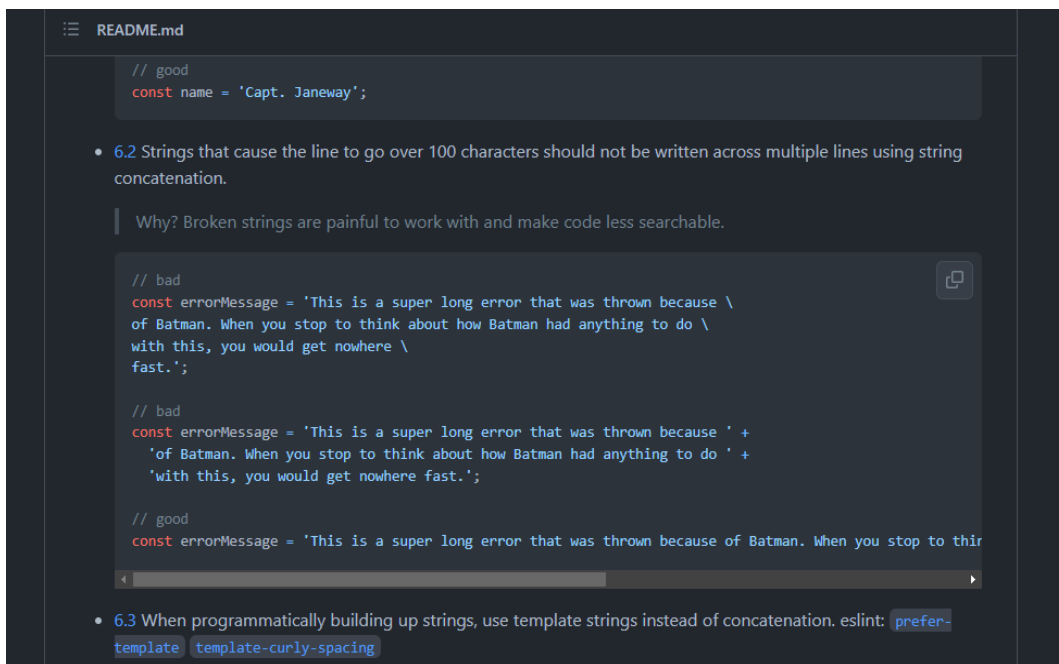
// good
const name = 'Capt. Janeway';
```

- 6.2 Strings that cause the line to go over 100 characters should not be written across multiple lines using string concatenation.

Why? Broken strings are painful to work with and make code less searchable.

```
// bad
const errorMessage = 'This is a super long error that was thrown because \
```

3. Doesn't this method make the code hard to read because the line will go out of the viewpoint then you have to side scroll to get to the end of the statement? Shouldn't template literals be used to increase readability?



The screenshot shows a README.md file with the following content:

```
// good
const name = 'Capt. Janeway';
```

- 6.2 Strings that cause the line to go over 100 characters should not be written across multiple lines using string concatenation.

Why? Broken strings are painful to work with and make code less searchable.

```
// bad
const errorMessage = 'This is a super long error that was thrown because \
of Batman. When you stop to think about how Batman had anything to do \
with this, you would get nowhere \
fast.';

// bad
const errorMessage = 'This is a super long error that was thrown because ' +
  'of Batman. When you stop to think about how Batman had anything to do ' +
  'with this, you would get nowhere fast.';

// good
const errorMessage = 'This is a super long error that was thrown because of Batman. When you stop to thir
```

A horizontal scrollbar is visible at the bottom of the code block, indicating that the line is too long to fit in the viewport.

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

