

DWA_04.3 Knowledge Check_DWA4

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

- **Maintaining Consistent Formatting:** Ensuring uniformity in formatting is essential to achieve a visually coherent and polished document or codebase. This principle emphasizes adopting a consistent style for indentation, spacing, naming conventions, and other formatting elements.
 - **Clarity through Descriptive Naming:** Selecting relevant and descriptive names for variables, functions, [classes], and other components greatly enhances the readability of the code.
 - **Embracing DRY Principle (Don't Repeat Yourself):** Encouraging the avoidance of duplicating code by abstracting common functionalities into reusable components or functions.
-

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

- 4.4 To convert an iterable object to an array, use spreads `...` instead of `Array.from`

```
const foo = document.querySelectorAll('.foo');

// good
const nodes = Array.from(foo);

// best
const nodes = [...foo];
```

- 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';
```

- 8.5 Avoid confusing arrow function syntax (`=>`) with comparison operators (`<=`, `>=`). eslint: `no-confusing-arrow`

```
// bad
const itemHeight = (item) => item.height <= 256 ? item.largeSize : item.smallSize;

// bad
const itemHeight = (item) => item.height >= 256 ? item.largeSize : item.smallSize;

// good
const itemHeight = (item) => (item.height <= 256 ? item.largeSize : item.smallSize);

// good
const itemHeight = (item) => {
  const { height, largeSize, smallSize } = item;
  return height <= 256 ? largeSize : smallSize;
};
```
