

# DWA\_04.3 Knowledge Check\_DWA4

---

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

Rules:

**1.2. When you access a complex type (objects, arrays, functions) you are working or a reference to its value.**

Multiple can point to the SAME data. If the data is changed, it will appear so irrespective of the reference (variable) that is used to access it.

I usually have bug due to the lack of understanding of this rule.

**2.3 The let and const keywords are block-scoped while var is function scoped.**

Variables defined inside a {} block, using let and const, cannot be accessed outside of the code block. Variables defined using var will be accessible outside of the {} code block. This can lead to unintended behaviour in the code and naming collisions. This adds more justification to avoid using the var keyword.

**3.5 Group object property values that have used the shorthand syntax at the beginning** of the object declaration. This makes it easier to tell which properties have used the shorthand and makes the code more readable.

---

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

Rules:

**3.7. Do not call Object.prototype methods directly, such as hasOwnProperty, propertyIsEnumerable, and isPrototypeOf.**

I currently don't have a firm understanding of object prototypes and the reasoning for the rules (These methods may be shadowed by properties on the object in question) is even more confusing.

### 6.1 Use single quotes '' for strings.

If JavaScript treats double quote and single quote strings the same, why is it bad to use double quote strings? No reasoning for this rule is provided in the documentation.

### 8.2 If a function has side effects, use the braces and the return statement.

```
let bool = false;

// bad
foo(() => bool = true);

// good
foo(() => {
  bool = true;
});
```

In the screenshot above, I'm assuming setting bool to true is a side effect. If it is a side effect, why does the "good" foo function not have a return statement?

---