

DWA_04.3 Knowledge Check_DWA4

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

Variables - Always use const and let so not to populate global variables. It is useful to follow this guideline to make code readable but to also ensure you follow.

```
// bad - unnecessary function call
```

```
function checkName(hasName) {
```

```
    const name = getName();
```

```
    if (hasName === 'test') {
```

```
        return false;
```

```
    }
```

```
    if (name === 'test') {
```

```
        this.setName('');
```

```
        return false;
```

```
    }
```

```
    return name;
```

```
}
```

```
// good
```

```
function checkName(hasName) {
```

```
    if (hasName === 'test') {
```

```
        return false;
```

```
    }
```

```
    const name = getName();
```

```
    if (name === 'test') {
```

```
        this.setName('');
```

```
        return false;
```

```
    }
```

```
    return name;
```

```
}
```

Controlled statements - for the sake of readability, keeping operators at the beginning of a line helps. Ensures understanding of logic used.

```

// bad
if (foo === 123
    && bar === 'abc') {
    thing1();
}

// bad
if (
    foo === 123 &&
    bar === 'abc'
) {
    thing1();
}

// good
if (
    foo === 123
    && bar === 'abc'
) {
    thing1();
}

// good
if (
    (foo === 123 || bar === 'abc')
    && doesItLookGoodWhenItBecomesThatLong()
    && isThisReallyHappening()
) {
    thing1();
}

```

Comments - useful because it explains the need for comments to provide context to the code but also concepts such as FIXME and TODO - which helps devs to decide whether something needs revisiting or a solution that needs implementation.

```

// bad
// make() returns a new element
// based on the passed in tag name
//
// @param {String} tag
// @return {Element} element
function make(tag) {

    // ...

    return element;
}

// good
/**
 * make() returns a new element
 * based on the passed-in tag name
 */
function make(tag) {

    // ...

    return element;
}

```

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

- 15.7 Avoid unneeded ternary statements. eslint: `no-unneeded-`

```
// bad
const foo = a ? a : b;
const bar = c ? true : false;
const baz = c ? false : true;
const quux = a !== null ? a : b;

// good
const foo = a || b;
const bar = !!c;
const baz = !c;
const quux = a ?? b;
```

-
- 22.5 Note: Be careful when using bitshift operations. Numbers are represented as [64-bit values](#), but bitshift operations always return a 32-bit integer ([source](#)). Bitshift can lead to unexpected behavior for integer values larger than 32 bits. [Discussion](#). Largest signed 32-bit Int is 2,147,483,647:

```
2147483647 >> 0; // => 2147483647
2147483648 >> 0; // => -2147483648
2147483649 >> 0; // => -2147483647
```