

DWA_04.3 Knowledge Check_DWA4

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

- [16.1](#) Use braces with all multiline blocks. eslint: [nonblock-statement-body-position](#)

```
// bad
if (test)
  return false;

// good
if (test) return false;

// good
if (test) {
  return false;
}

// bad
function foo() { return false; }
```

Using braces makes code easier and quicker to read which personally helps me understand the code faster

- [3.6](#) Only quote properties that are invalid identifiers. eslint: [quote-props](#)

Why? In general we consider it subjectively easier to read. It improves syntax highlighting, and is also more easily optimized by many JS engines.

```
// bad
const bad = {
  'foo': 3,
  'bar': 4,
  'data-blah': 5,
};

// good
const good = {
  foo: 3,
  bar: 4,
  'data-blah': 5,
};
```

This makes code easier to understand and to use

- [23.1](#) Avoid single letter names. Be descriptive with your naming. eslint: `id-length`

```
// bad
function q() {
  // ...
}

// good
function query() {
  // ...
}
```

By using descriptive names code is a lot of easier to understand compared to single letter names

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

- [3.1](#) Use the literal syntax for object creation. eslint: `no-new-object`

```
// bad
const item = new Object();

// good
const item = {};
```

As a learner i find it easier to currently understand the first version as it describes what it does with the text

- [3.3 Use object method shorthand.](#) eslint: `object-shorthand`

```
// bad
const atom = {
  value: 1,

  addValue: function (value) {
    return atom.value + value;
  },
};

// good
const atom = {
  value: 1,

  addValue(value) {
    return atom.value + value;
  },
};
```

Once again as a learner the longer method is easier to understand

- [7.2 Wrap immediately invoked function expressions in parentheses.](#) eslint: `wrap-iife`

Why? An immediately invoked function expression is a single unit - wrapping both it, and its invocation parens, in parens, cleanly expresses this. Note that in a world with modules everywhere, you almost never need an IIFE.

```
// immediately-invoked function expression (IIFE)
(function () {
  console.log('Welcome to the Internet. Please follow me.');
```

```
}());
```

I personally find it cleaner to have less unnecessary parenthesis
