

# DWA\_04.3 Knowledge Check\_DWA4

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

- [2.1](#) Use `const` for all of your references; avoid using `var`. eslint: `prefer-const`, `no-const-assign`

Using `const` ensures you cannot accidentally reassign variables you otherwise intended to be static values.

- [4.8](#) Use line breaks after opening array brackets and before closing array brackets, if an array has multiple lines

```
// bad
const arr = [
  [0, 1], [2, 3], [4, 5],
];

const objectInArray = [{
  id: 1,
}, {
  id: 2,
}];

const numberInArray = [
  1, 2,
];

// good
const arr = [[0, 1], [2, 3], [4, 5]];

const objectInArray = [
  {
    id: 1,
  },
  {
    id: 2,
  },
];

const numberInArray = [
  1,
  2,
];
```

Having it formatted this way makes it a lot easier to read since you can very clearly see where things open and close.

- [10.4](#) Only import from a path in one place. eslint: `no-duplicate-imports`

Why? Having multiple lines that import from the same path can make code harder to maintain.

```
// bad
import foo from 'foo';
// ... some other imports ... //
import { named1, named2 } from 'foo';

// good
import foo, { named1, named2 } from 'foo';

// good
import foo, {
  named1,
  named2,
} from 'foo';
```

Have one import line for the same path. It keeps things organised and succinct. The same when it comes to exporting.

- [12.1](#) Use dot notation when accessing properties. eslint: `dot-notation`

```
const luke = {
  jedi: true,
  age: 28,
};

// bad
const isJedi = luke['jedi'];

// good
const isJedi = luke.jedi;
```

This is a bonus one, I just think it's a lot faster to use dot notation than to add brackets and double ticks.

---

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

- [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';
```

It doesn't confuse me, I just don't like it. When I see single quotes I don't immediately associate them with strings and that can throw me off for a moment.

- [10.10](#) Do not include JavaScript filename extensions eslint: [import/extensions](#)

Why? Including extensions inhibits refactoring, and inappropriately hardcodes implementation details of the module you're importing in every consumer.

```
// bad
import foo from './foo.js';
import bar from './bar.jsx';
import baz from './baz/index.jsx';

// good
import foo from './foo';
import bar from './bar';
import baz from './baz';
```

I'm not sure why you wouldn't want to add the .js extension. I would think it keeps things tidy.

- [19.11](#) Do not add spaces inside brackets. eslint: `array-bracket-spacing`

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

// 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' };
```

Just the seemingly arbitrary designation of who gets spaces inside their brackets and who doesn't.

---