

DWA_04.3 Knowledge Check_DWA4

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

Quotes

- Always use double quotes (") for JSX attributes, but single quotes (') for all other JS. eslint: `jsx-quotes`

Why? Regular HTML attributes also typically use double quotes instead of single, so JSX attributes mirror this convention.

```
// bad
<Foo bar='bar' />

// good
<Foo bar="bar" />

// bad
<Foo style= />

// good
<Foo style= />
```

- I find this useful as it gives consistency throughout the document, given the outline of when to use which quotation character.

Tags

- Always self-close tags that have no children. eslint: `react/self-closing-comp`

```
// bad
<Foo variant="stuff"></Foo>

// good
<Foo variant="stuff" />
```

- This is great as it affirms that my tag is correct since at times when closing a tag without any children nested inside makes it seem like its incorrect or something is missing.

- 13.3 Group all your `const` s and then group all your `let` s.

Why? This is helpful when later on you might need to assign a variable depending on one of the previously assigned variables.

```
// bad
let i, len, dragonball,
    items = getItem(),
    goSportsTeam = true;

// bad
let i;
const items = getItem();
let dragonball;
const goSportsTeam = true;
let len;

// good
const goSportsTeam = true;
const items = getItem();
let dragonball;
let i;
let length;
```

- This is useful in terms of easily finding a specific variable when need to make changes at a later stage.

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

- 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.');
```

- This might be confusing as additional brackets affect readability and may be unnecessary, since all functions technically run immediately when they invoked.

- 4.3 Use array spreads `...` to copy arrays.

```
// bad
const len = items.length;
const itemsCopy = [];
let i;

for (i = 0; i < len; i += 1) {
  itemsCopy[i] = items[i];
}

// good
const itemsCopy = [...items];
```

- Why is using the fully written code not better if it allows the reader to better understand the transfer of the content of one array to another? Using array spreads are less detailed.

- 14.2 Anonymous function expressions hoist their variable name, but not the function assignment.

```
function example() {  
  console.log(anonymous); // => undefined  
  
  anonymous(); // => TypeError anonymous is not a function  
  
  var anonymous = function () {  
    console.log('anonymous function expression');  
  };  
}
```

- How is this allowed? The result of the function defined below returns “is not a function” when it’s called above, does it mean it's only used to define a point?
-