

DWA_04.3 Knowledge Check_DWA4

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

Important to know the difference between rest or spread operator.

[JavaScript Rest vs Spread Operator – What's the Difference? \(freecodecamp.org\)](https://www.freecodecamp.org/javascript/rest-vs-spread-operator)

7.6 Never use `arguments`, opt to use rest syntax `...` instead. eslint: `prefer-rest-params`

Why? `...` is explicit about which arguments you want pulled. Plus, rest arguments are a real Array, and not merely Array-like like `arguments`.

```
// bad
function concatenateAll() {
  const args = Array.prototype.slice.call(arguments);
  return args.join('');
}
```

```
// good
function concatenateAll(...args) {
  return args.join('');
}
```

I need to know what a default parameter is. And that its bad to mutate it later than in the function argument before.

7.7 Use default parameter syntax rather than mutating function arguments.

```
// really bad
function handleThings(opts) {
  // No! We shouldn't mutate function arguments.
  // Double bad: if opts is falsy it'll be set to an object
  which may
  // be what you want but it can introduce subtle bugs.
  opts = opts || {};
  // ...
}
```

```

// still bad
function handleThings(opts) {
  if (opts === void 0) {
    opts = {};
  }
  // ...
}

// good
function handleThings(opts = {}) {
  // ...
  • }

```

7.14 Prefer the use of the spread operator `...` to call variadic functions. eslint:

`prefer-spread`

Why? It's cleaner, you don't need to supply a context, and you can not easily compose new with `apply`.

```

// bad

const x = [1, 2, 3, 4, 5];

console.log.apply(console, x);

```

```

// good

const x = [1, 2, 3, 4, 5];

console.log(...x);

```

```

// bad

new (Function.prototype.bind.apply(Date, [null, 2016, 8, 5]));

```

```
// good
```

- `new Date(...[2016, 8, 5]);`

This is more efficient way of copying arrays.

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];`

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

Not sure why default parameter is what it is or that name should be last.
Answer. Opts is assigned in default parameters and wont change.

7.9 Always put default parameters last.

// bad

```
function handleThings(opts = {}, name) {
```

```
    // ...
```

```
}
```

// good

```
function handleThings(name, opts = {}) {
```

```
    // ...
```

```
}
```

3.4 Use property value shorthand. eslint: `object-shorthand` jscs:

`requireEnhancedObjectLiterals`

Why? It is shorter to write and descriptive.

```
const lukeSkywalker = 'Luke Skywalker';
```

// bad

```
const obj = {
```

```
    lukeSkywalker: lukeSkywalker,
```

```
};
```

// good

```
const obj = {
```

```
lukeSkywalker,
```

- `};`

7.8 Avoid side effects with default parameters.

Why? They are confusing to reason about.

```
var b = 1;
```

```
// bad
```

```
function count(a = b++) {  
  console.log(a);  
}
```

```
count(); // 1
```

```
count(); // 2
```

```
count(3); // 3
```

- `count(); // 3`

5.2 Use array destructuring. `eslint: prefer-destructuring` `jscs:`

`requireArrayDestructuring`

```
const arr = [1, 2, 3, 4];
```

```
// bad
```

```
const first = arr[0];
```

```
const second = arr[1];
```

```
// good
```

- `const [first, second] = arr;`
-