

DWA_04.3 Knowledge Check_DWA4

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

5.1) Use object destructuring when accessing and using multiple properties of an object.

- Destructuring saves you from creating temporary references for those properties, and saves you from repetitive code.

(13.6) Avoid using unary increments and decrements (++ , - -)

Unary increment and decrement statements are subject to automatic semicolon insertion and can cause silent errors with values within an application

Instead of (num++) use (num += 1)

(18.4) Prefixing your comments with FIXME or TODO helps other developers quickly understand if you're pointing out a problem that needs to be revisited, or if you're suggesting a solution to the problem that needs to be implemented.

This is good practice as it allows developers to keep track of the work they are doing instead of making mental notes on what needs to be fixed.

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

(4.3) Use array spreads... to copy arrays

// bad

Confused about the for...loop

```
const len = items.length;
```

```
const itemsCopy = [];
```

```
let i;
```

```
for (i = 0; i < len; i += 1) {
```

```
}
```

```
itemsCopy[i] = items[i];
```

// good

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

(9.1) Always use class. Avoid manipulating prototype directly.

// bad

```
function Queue(contents = []) {
```

```
  this.queue=[...contents];
```

```
}
```

```
Queue.prototype.pop = function(){ const value = this.queue[0];
```

```
this.queue.splice(0, 1); return value;
```

```
};
```

// good

```
class Queue {
```

```
  constructor(contents = []) {
```

```
  }
```

```
  pop() {
```

```
  }
```

```
}
```

```
this.queue=[...contents];
```

```
const value = this.queue[0]; this.queue.splice(0, 1);
```

```
return value;
```
