DWA_04.3 Knowledge Check_DWA4

---

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

Arrow Function Parentheses (arrow-parens):
I find this rule useful because it improves code consistency, readability and makes the code short.

javascript
Copy code
```
// Bad
const foo = x => x * 2;

// Good
const foo = (x) => x * 2;
```

Object Shorthand (object-shorthand):
Helps to write cleaner and more concise code by eliminating redundancy. It makes it easier to read and understand the object's properties.

Copy code
```
const x = 10;

// Bad
const obj = {
  x: x,
  y: 20,
};

// Good
const obj = {
  x,
  y: 20,
};
```

No Unused Variables (no-unused-vars):
Helps in maintaining code cleanliness and preventing unnecessary clutter. It ensures that variables are only declared when they are actually needed, promoting better code quality.

Copy code
```
// Bad
const x = 10;
const y = 20;

// Good
const x = 10;
console.log(x);
```

---

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

No-else-return (no-else-return):
This rule can be confusing because it implies that having an else clause after a return statement is always undesirable.

Copy code
```
// Bad
function foo(condition) {
  if (condition) {
    return x;
  } else {
    return y;
  }
}

// Good
function foo(condition) {
  if (condition) {
```

```
    return x;
  }
  return y;
}
```

Disallow Reassignment of Function Parameters (no-param-reassign)
This rule can be confusing because it goes against the expectation that function parameters can be treated as regular variables within the function body.

Copy code

```
// Bad
function myFunction(param) {
  param = 'new value';
}
```

No Multi Spaces (no-multi-spaces):
While the intention of this rule is to enforce consistent spacing, it can be confusing because it doesn't differentiate between spaces used for alignment and those used for indentation

javascript
Copy code

```
// Bad
const x  = 10;
const y =  20;

// Good
const x = 10;
const y = 20;
```

.
