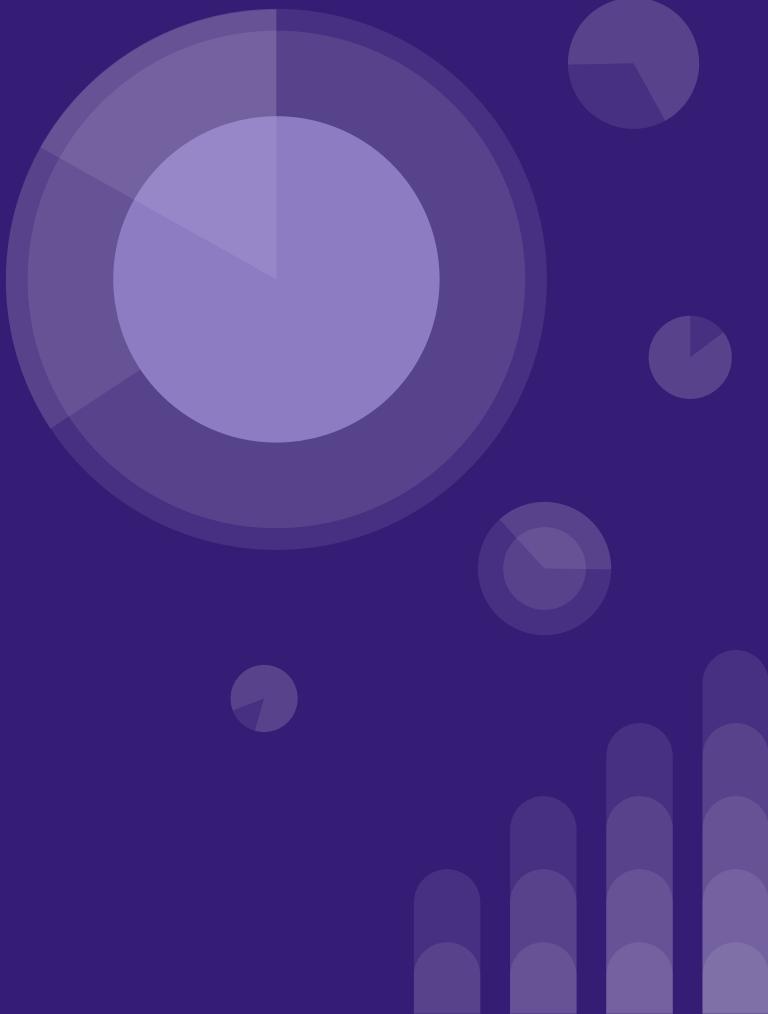


JS Calisthenics

Lorenzo Mignola





Is JS calisthenics?





Object Calisthenics

- Only one level of indentation per method
- Don't use the ELSE keyword
- Wrap all primitives and strings
- First class collections
- Only one dot per line
- No abbreviations
- Keep all entities small
- No classes with more than two instance variables
- No public getter/setters/properties



~~Object~~ Calisthenics

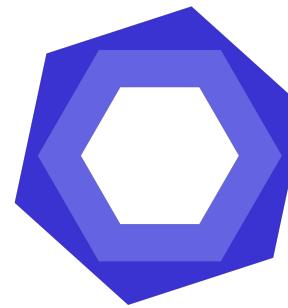
- Only one level of indentation per method
- Don't use the ELSE keyword
- Wrap all primitives and strings
- ~~First class collections~~
- Only one dot per line
- No abbreviations
- Keep all entities small
- ~~No classes with more than two instance variables~~
- No public getter/setters/properties
...and more



eslint-config-airbnb



+



<https://www.npmjs.com/package/eslint-config-airbnb> ↗



What is a linter?

- Show errors on IDE
- Ensure style



Only one level of indentation per method



No guide line



- max-depth ↗



Only one level of indentation per method

✗ Bad

```
1  function foo() {  
2      if (true) {  
3          if (true) {  
4              // ...  
5      }  
6  }  
7 }
```

✓ Good

```
1  function foo() {  
2      if (true && true) {  
3          // ..  
4      }  
5  }
```



Don't use the ELSE keyword



- 16.3 Blocks ↗



- no-else-return ↗



Don't use the ELSE keyword

✗Bad

```
1  function foo() {  
2      if (x) {  
3          return x;  
4      } else {  
5          return y;  
6      }  
7  }
```

✓Good

```
1  function foo() {  
2      if (x) {  
3          return x;  
4      }  
5      return y;  
6  }
```



Wrap all primitives and strings



- 5 Destructuring ↗



- prefer-destructuring ↗



Wrap all primitives and strings

✗ **Bad**

```
1 function getFullName(  
2   firstName,  
3   lastName  
4 ) {  
5   return `${firstName} ${lastName}`;  
6 }
```

✓ **Good**

```
1 function getFullName(  
2   firstName,  
3   lastName  
4 ) {  
5   return `${firstName} ${lastName}`;  
6 }
```



Only one dot per line



No style guide



- max-nested-callback ↗



Only one dot per line

✗Bad

```
1  foo1(function () {  
2    foo2(function () {  
3      // ...  
4    });  
5  });
```

✓Good

```
1  foo1(handleFoo1);  
2  function handleFoo1() {  
3    handleFoo2(handleFoo2);  
4  }  
5  function handleFoo2() {  
6    foo3();  
7  }
```



No abbreviations



- 23.1 Naming conventions ↗



- id-length ↗



No abbreviations

✗Bad

```
1  function q() {  
2      // ...  
3 }
```

✓Good

```
1  function query() {  
2      // ...  
3 }
```



Keep all entities small



- [19.13 Whitespace ↗](#)



- [indent ↗](#)
- [max-lines ↗](#)
- [max-len ↗](#)



Keep all entities small

- 2 spaces
- 300 lines
- 100 characters



No public getter/setter/properties



- [2 References ↗](#)



- [no-var ↗](#)



No public getter/setter/properties

✗Bad

```
1 if (true) {  
2   var count = 1;  
3 }  
4 console.log(count); // 1
```

✓Good

```
1 let count;  
2 if (true) {  
3   count = 1;  
4 }  
5 console.log(count); // 1
```

OK, but ...

Uncaught TypeError: myObject.foo is not a function



... use **TypeScript**

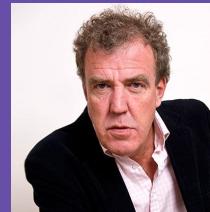




Accident rule

“Speed has never killed anyone. Suddenly becoming stationary, that's what gets you”

Jeremy Clarkson
TopGear Host



*“I don’t use the autocomplete
so I don’t need TypeScript”*

<https://www.youtube.com/watch?v=VHoyXNqjuGY>

Matteo Collina
Node.js TSC member



