



# Object Calisthenics

## 3 Simple Code Transformations

Simon Hodel, 17.08.2020

# Rule 1: One Level of Indentation per Method

- How do we fix this?

```
1  function getStatus(healthPoint) {
2    if (healthPoint <= 0) {
3      // One Level Indentation is fine
4      return 'DEAD';
5    }
6    if (healthPoint > 0) {
7      if (healthPoint < 50) {
8        // Two Level Indentation found. it has violated the rule
9        return 'LOW HEALTH';
10     }
11   }
12   return 'ALIVE';
13 }
```

# Rule 1: One Level of Indentation per Method

- Merging lines 6 and 7 to one if statement

```
1  function getStatus(healthPoint) {
2    if (healthPoint <= 0) {
3      // One Level Indentation is fine
4      return 'DEAD';
5    }
6    if (healthPoint > 0 && healthPoint < 50) {
7      // become One Level Indentation
8      return 'LOW HEALTH';
9    }
10   return 'ALIVE';
11 }
```

# Rule 3: Wrap All Primitives And Strings

- Violation because of standalone primitives

```
1  const kilometer = 1;  
2  
3  const meter = kilometer * 1000;
```

calisthenics-wrap-primitives-before.js hosted with ❤️ by GitHub

[view raw](#)

# Rule 3: Wrap All Primitives And Strings

- Create a Distance class to wrap value

```
1  class Distance {
2    constructor(value, unitType) {
3      this._value = value;
4      this._unitType = unitType;
5    }
6
7    toMeter() {
8      // logic to convert current unitType to meter
9    }
10 }
11
12 //implementation
13 const kilometer = new Distance(1, "kilometer");
14
15 const meter = kilometer.toMeter();
```

# Rule 9: No Getters / Setters / Properties

- Violation: Getters and setters used

```
1  class Wallet {
2    constructor() {
3      this._balance = 0;
4    }
5
6    setBalance(balance) {
7      this._balance = balance;
8    }
9
10   getBalance() {
11     return this._balance;
12   }
13 }
```

```
15  //implementation on main
16
17  const wallet = new Wallet();
18  wallet.setBalance(5000); // add initial balance
19
20  wallet.setBalance(wallet.getBalance() + 2000); // want
21
22  console.log(`Last Balance: ${wallet.getBalance()}`);
```

calisthenics-no-setter-getter-before.js hosted with ❤️ by GitHub

# Rule 9: No Getters / Setters / Properties

- Meaningful methods & encapsulation

```
1 class Wallet {
2   constructor(balance) {
3     this._balance = balance;
4   }
5
6   increase(money) {
7     this._balance += money;
8   }
9
10  toString() {
11    return `Last Balance: ${wallet.getBalance()}`;
12  }
13 }
```

```
15 //implementation on main
16
17 const wallet = new Wallet(5000);
18
19 wallet.increase(2000);
20
21 console.log(wallet.toString());
```

calisthenics-no-setter-getter-after.js hosted with ❤️ by GitHub



# Credits

- Code snippets from <https://medium.com/@davidsen/clean-code-object-calisthenics-f6f4dec07c8b>