# Object Calisthenics rules

## How does it help?

# Only one level of indentation per method

**WHY?**

- Reduce complexity
- Easier to understand
- Easier to read

**HOW?**

- Make it simpler
- Split up the method

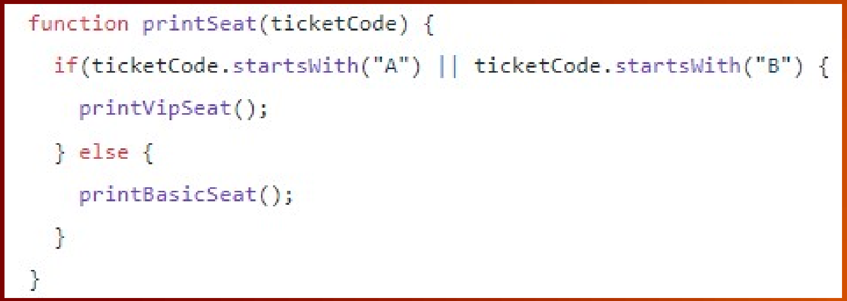
# Don’t use the ELSE keyword

## WHY?

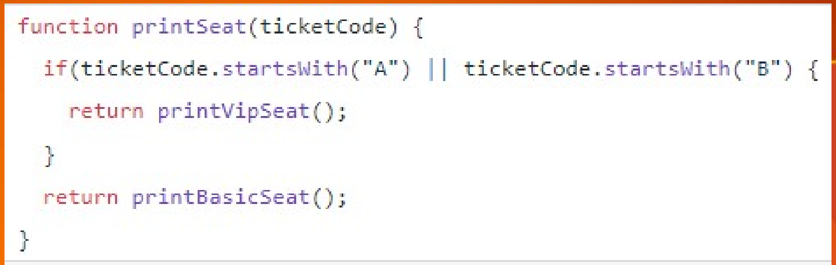
- Reduce complexity
- Reduce duplication

## HOW?

- Early return
- Polimorphism



```
function printSeat(ticketCode) {
  if(ticketCode.startsWith("A") || ticketCode.startsWith("B") {
    printVipSeat();
  } else {
    printBasicSeat();
  }
}
```

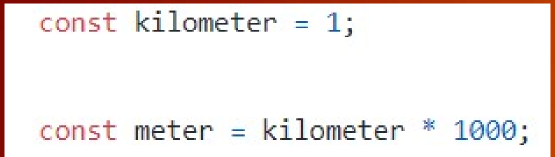


```
function printSeat(ticketCode) {
  if(ticketCode.startsWith("A") || ticketCode.startsWith("B") {
    return printVipSeat();
  }
  return printBasicSeat();
}
```

# Wrap all primitives and strings

## WHY?

- Represent the real world

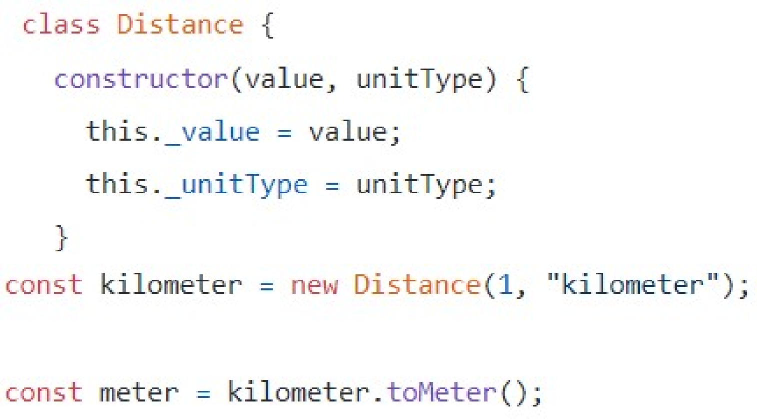


```
const kilometer = 1;


const meter = kilometer * 1000;
```

## HOW?

- Wrap them in a class



```
 class Distance {
   constructor(value, unitType) {
     this._value = value;
     this._unitType = unitType;
   }
const kilometer = new Distance(1, "kilometer");


const meter = kilometer.toMeter();
```

# First class collections (wrap all collections)

**WHY?**

- Represent the real world
- Behaviors at the right place

**HOW?**

- Wrap them in a class

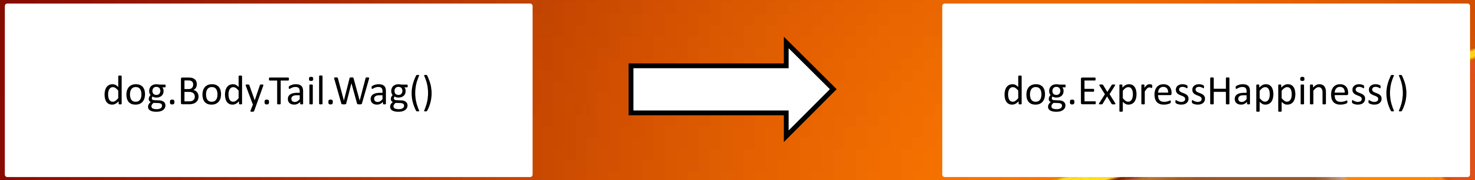
# Only one dot per line

**WHY?**

- More readable
- Less knowledge to enable a behavior

**HOW?**

- Wrap them in a class



dog.Body.Tail.Wag()

dog.ExpressHappiness()

# No abbreviations

**WHY?**

- Less confusing
- More understandable

**HOW?**

- Meaningful names

# Keep all entities small

**WHY?**

- Focused on one thing
  - Easier to reuse
- Easier to understand

**HOW?**

- Use the Object Calisthenics rules

# No classes with more than two instance variables

## WHY?

- Cohesion won’t decrease
- Separate the responsibilities
- Simpler classes

## HOW?

- Wrap them in a class

# No public getters/setters/properties

## WHY?

- Cohesion won’t decrease
  - Reduce duplication

## HOW?

- Meaningful methods
  - Tell, don’t ask

# Why should we follow the rules?

- Simpler code
- Easier to read and understand
- Easier to maintain
- Easier to test

“Always code as if the guy who ends up maintaining your code will be a violent psychopath who knows where you live”

— John Woods

# References

- https://medium.com/@davidsen/clean-code-object-calisthenics-f6f4dec07c8b

# Questions?

# Thank you for your attention!

Ildikó Takács