

Understanding the Language Basics



Roland Guijt

INDEPENDENT SOFTWARE DEVELOPER AND TRAINER

@rolandguijt www.rmgsolutions.nl



Module Overview



TypeScript syntax

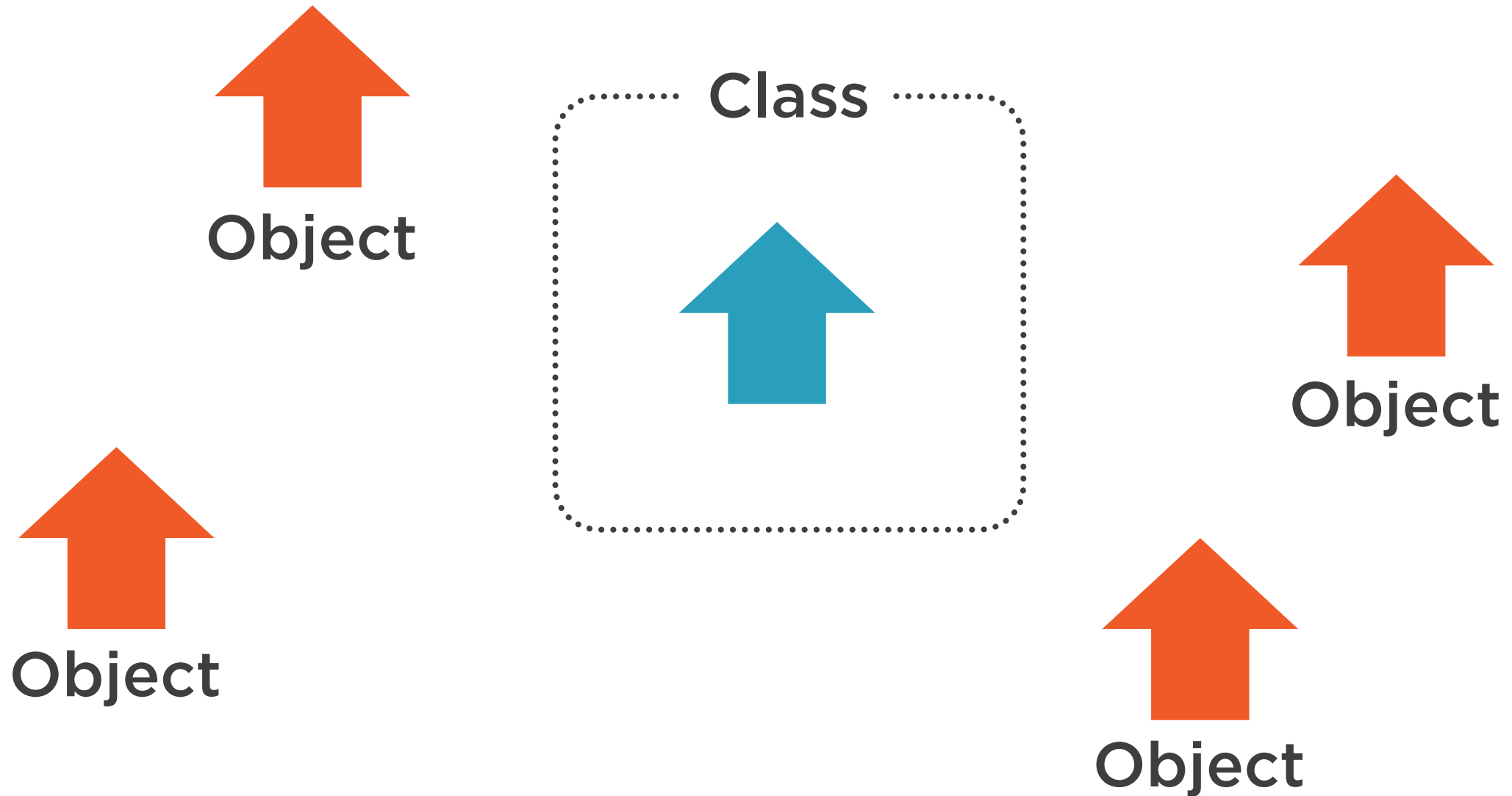
Built-in types

Custom types

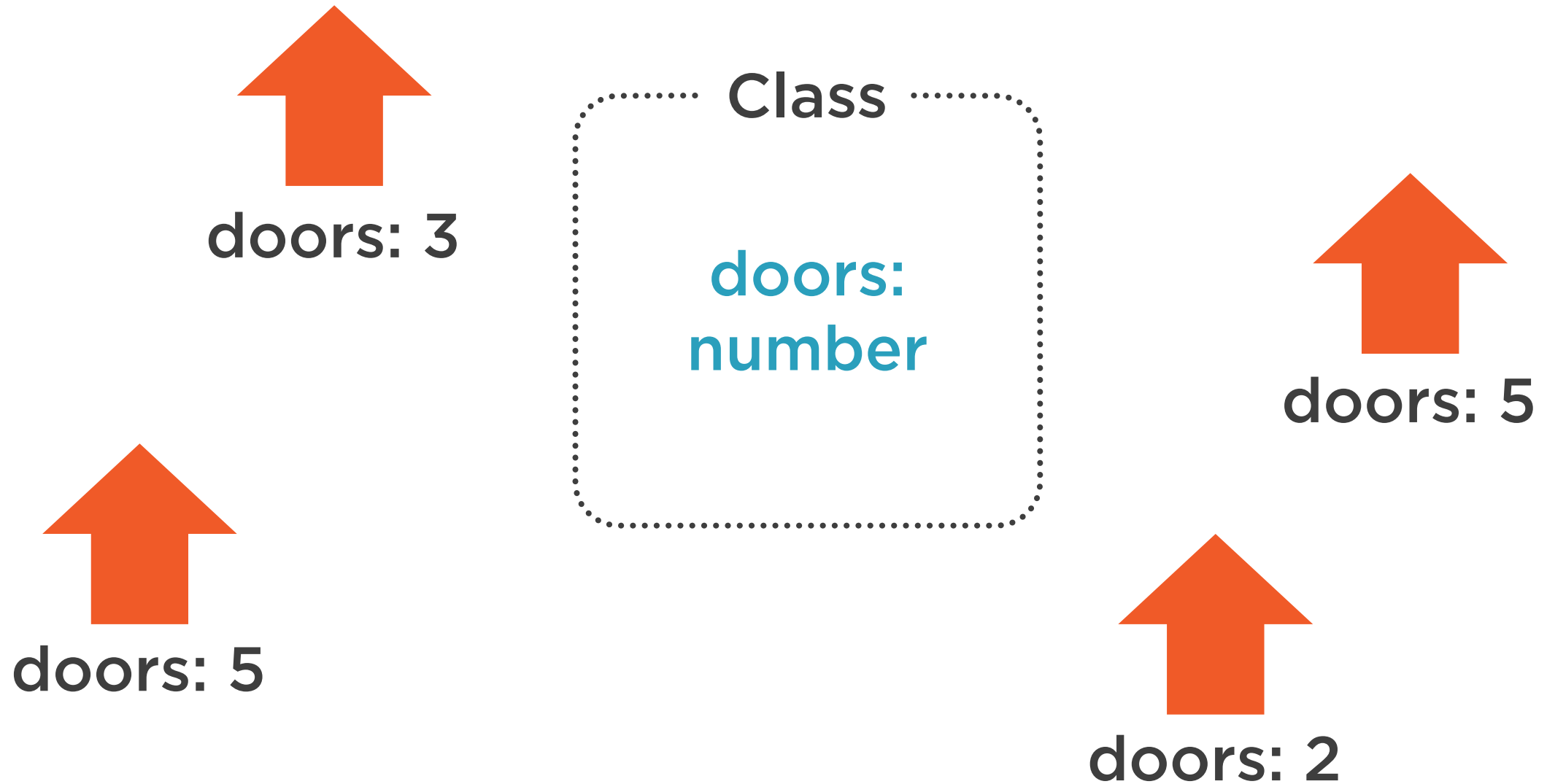
Working vending machine



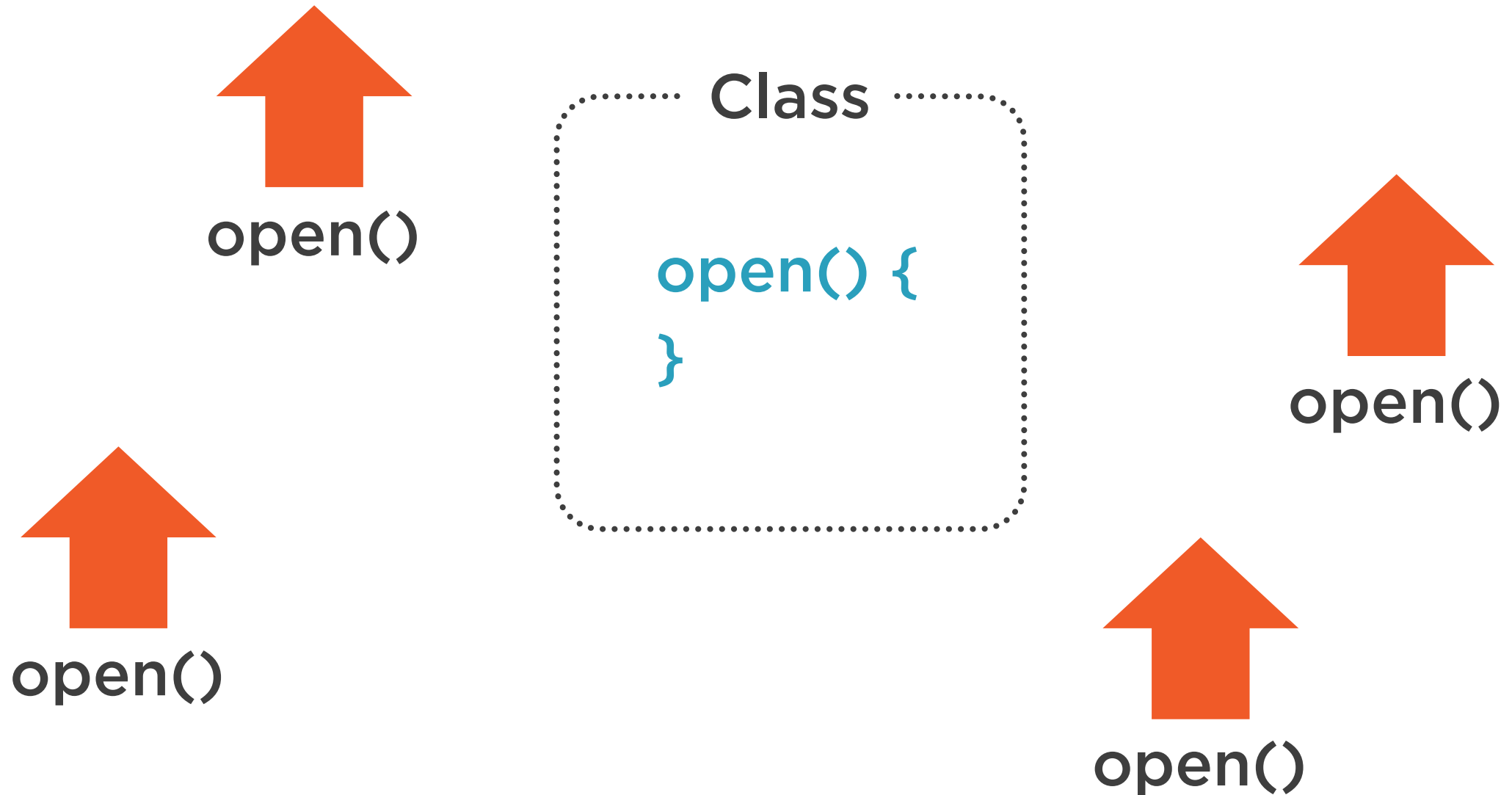
Classes and Objects



Classes and Objects



Classes and Objects



Data Types

Built-in

boolean

string

number

Custom

enum

array

interface

class



Any

```
var notSure: any = 4;
```

```
notSure = "maybe a string instead";
```

```
notSure = false;
```



Arrow Functions

```
class VendingMachine {  
  
    acceptCoin = (coin: Quarter): void => {  
  
        this.paid += coin.value;  
  
    }  
  
}
```



This

This in a non-arrow function will point to the object that is calling the function

In arrow functions **this** always refers to the object (of the class) the function is in



Arrow Functions

```
var acceptCoin = ( coin:Quarter ) => {  
  
    this.paid += coin.value;  
  
}
```

```
acceptCoin(new Quarter());
```



Cleaner Code

Use data binding

Knockout.js

There are other solutions



Role of Knockout.js

View (HTML)



Model View



Model View

Model (ts object)

Day = ~~True~~ False



Type Definitions

TypeScript contains typing for standard javascript objects


But it doesn't know about external libraries

Type definitions help out



Function Scope

```
acceptCoin = (coin: Quarter): void => {  
  var oldTotal = this.paid(); //2  
  if (oldTotal > 10) {  
    var much = true;  
  }  
  
  console.log(oldTotal); //2  
  
  console.log(much); //true  
}
```

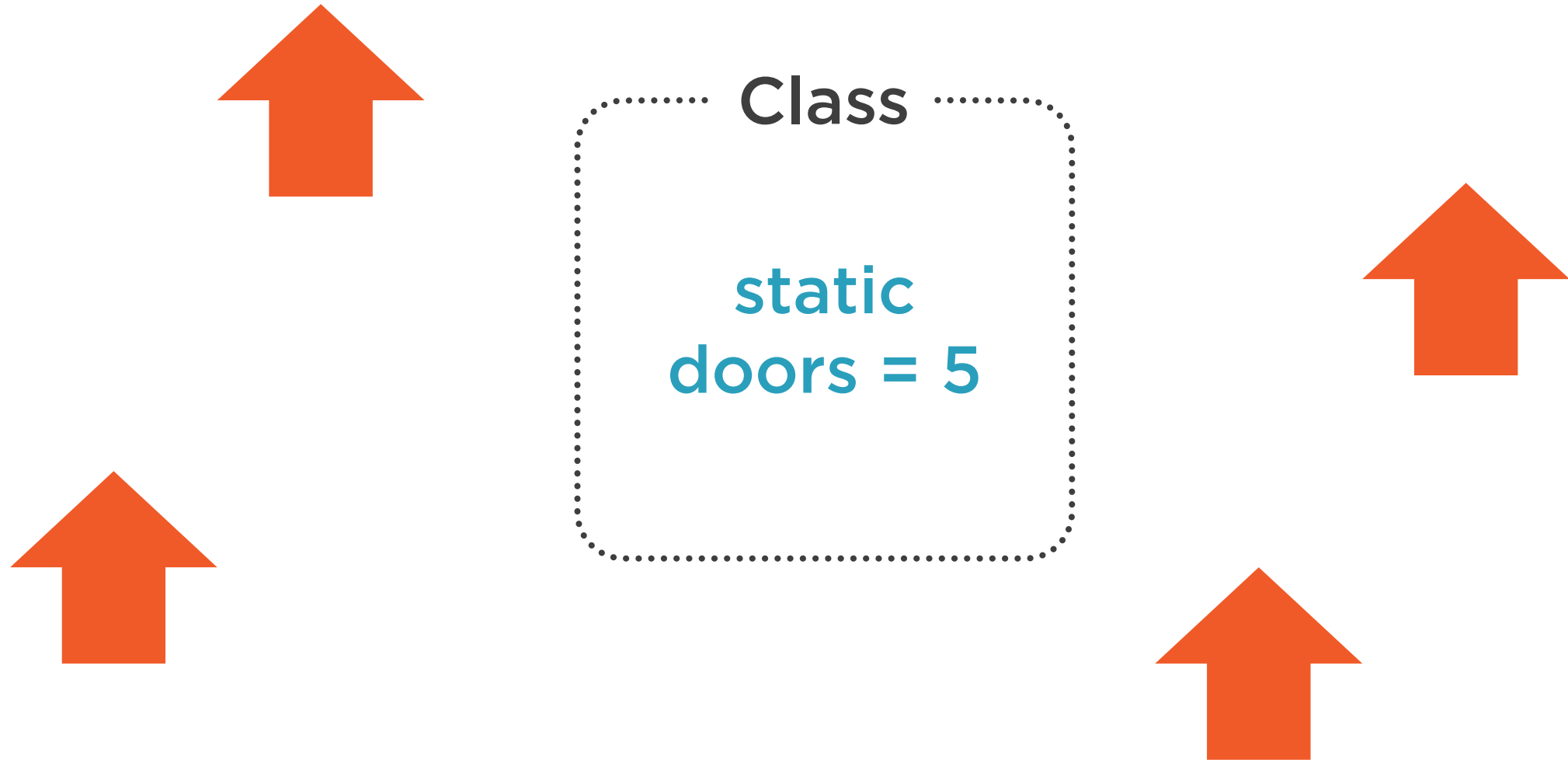


Block Scope

```
acceptCoin = (coin: Quarter): void => {  
  
    let oldTotal = this.paid(); //2  
  
    if (oldTotal > 10) {  
  
        let much = true;  
    }  
  
    console.log(oldTotal); //2  
  
    console.log(much); //undefined  
}
```



Statics



Enums

```
enum VendingMachineSize {  
    small = 6,  
    medium = 9,  
    large = 12  
}
```

..

```
vendingMachine.size = VendingMachineSize.medium;
```



Benefits of Enums

Constraint

Maintainability of Code

Readability of Code



Summary



Classes

Data Types

Typing

Properties

Functions

Type Definitions

Let

Arrays

Enums

