

## Dag 4

### Higher-order functions

Executing a function is called invoking, calling or applying it (aanroepen).

Arrow functions (ES6) wordt vaak gebruikt bij higher-order functions, omdat deze op één regel kan worden geschreven en dus makkelijk is te lezen.

**Higher-order functions:** een functie wordt als argument gebruikt in een andere functie.

Callback function: functie als argument die binnen een andere functie wordt aangeroepen, bijv. `addEventListener (function())`. Hierbij is `function()` de callback function

MDN web docs: filter, map, reduce: hebben altijd een andere functie nodig om te vullen

Vaak lopen higher-order functions al van zichzelf over de array én maken vaak ook een nieuwe array aan

Gebruik je vaak bij het opschonen van data

Werken alleen op arrays

Hoe hoger je gaat in ES versies, hoe minder browsers deze ondersteunen

**Objects:** array on steroids

```
const object = {  
  key1: 'value',  
  key2: 12,  
  key3: [1, 2, 3]  
}  
  
const object = {  
  "key1": 'value',  
  key2: 12,  
  key3: [{  
    key: 'value',  
    key: 'value'}]  
}
```

```
}
```

Mutability = overschrijfbaarheid

const is immutable

- ➔ Object kan niet overschreven worden (immutable)
- ➔ Properties kunnen wel overschreven worden (mutable)
- ➔ Met `object.freeze` worden properties immutable

**Methods:** een functie binnen een object

```
console.log(name.length)
```

length = method

this refereert naar iets binnen het huidige object

method is niet per se een HO function

HO function is standaard megegeven in JS

check de video's van backend voor de volgende les