Functions 2

Javascript

NIFR 2023

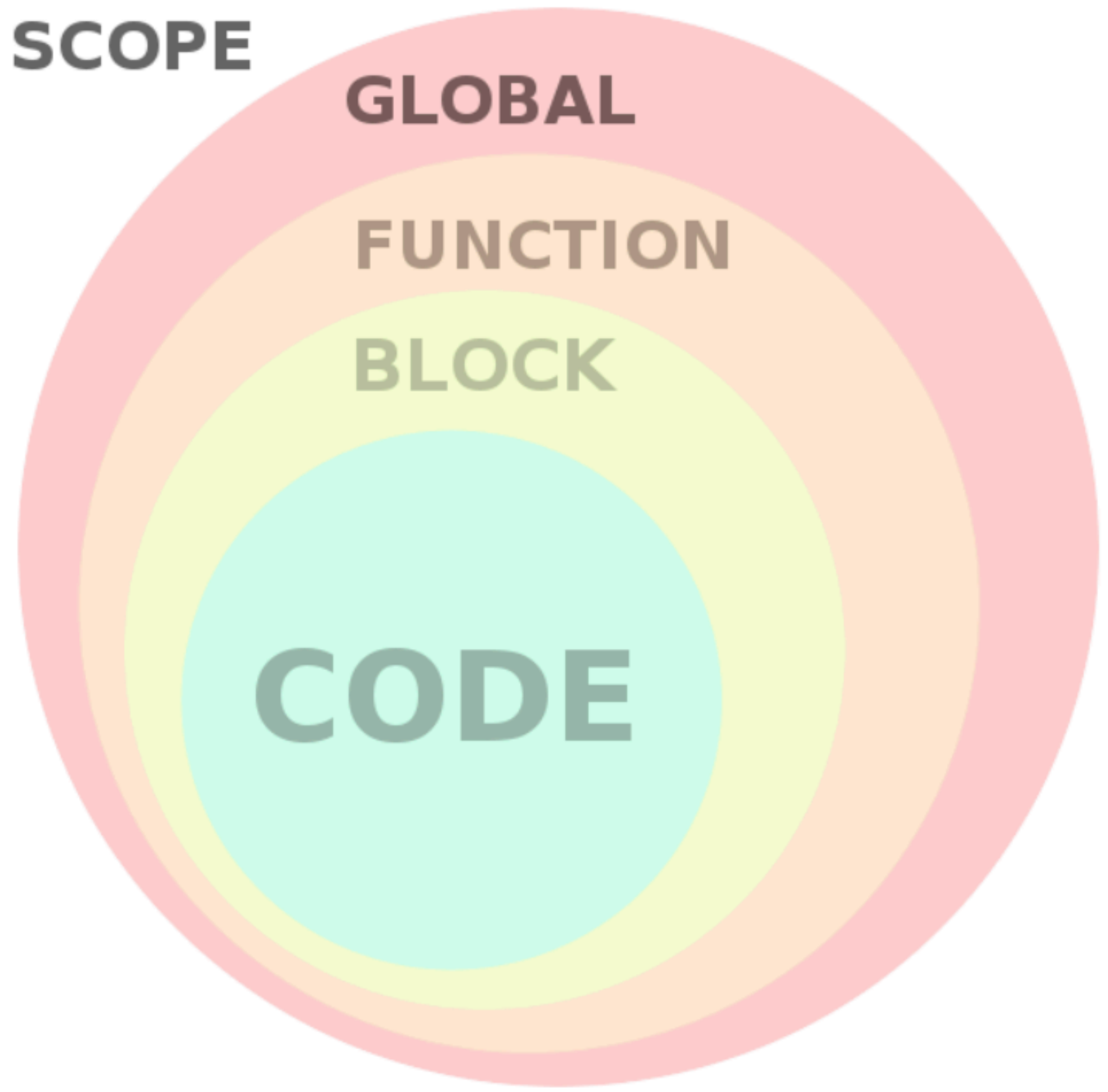
Husk Test - næste fredag

https://github.com/nicklasdean/ita-23-1-sem-code

Recap

Javascript functions

- Function signature
- Function scope
- Return values



Name

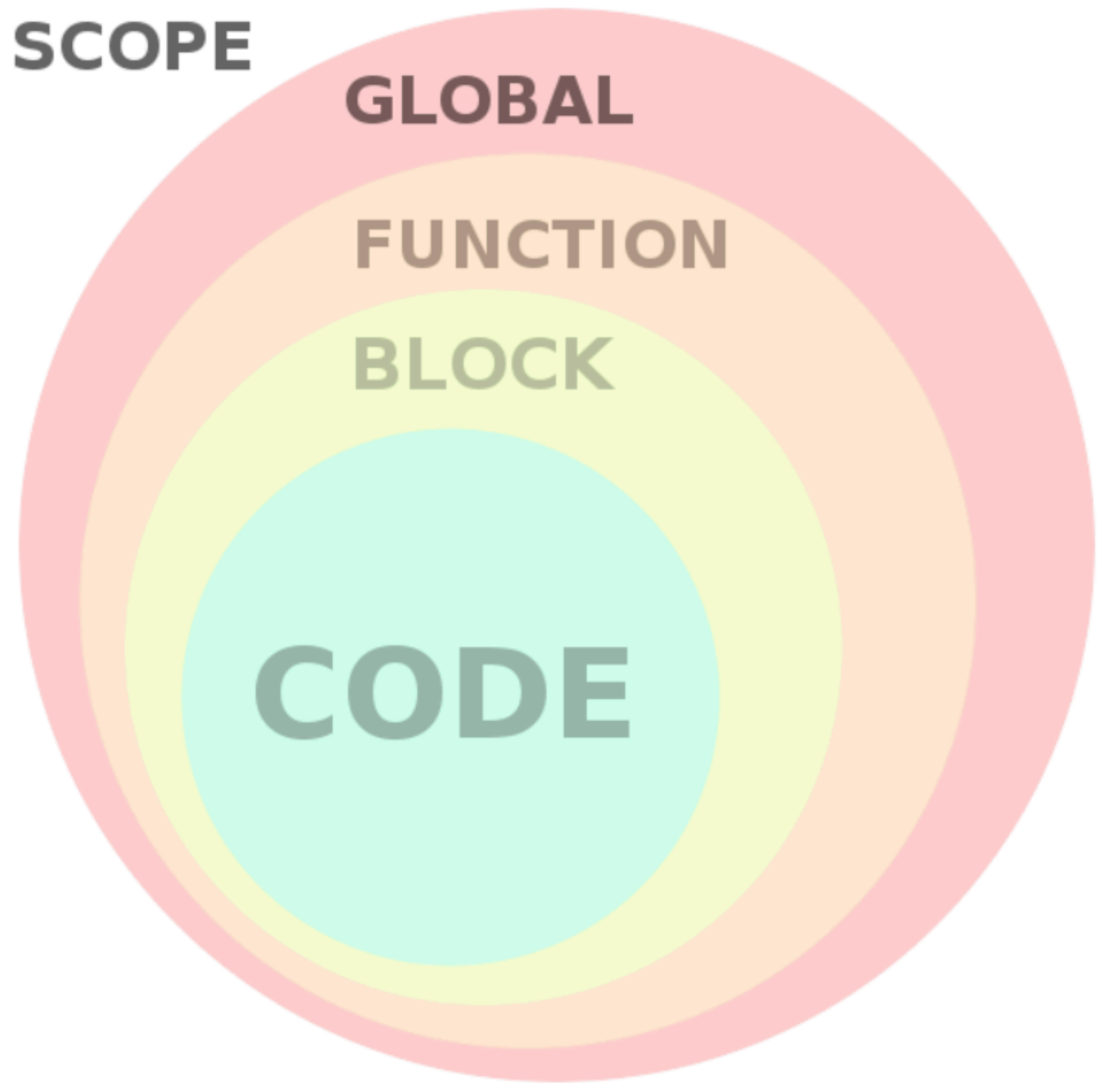
Parameters

```
function sumTwoNumbers(a,b){
    let result = a + b;
    console.log(result);
}
```

Scope

Function scope

- Global variables can be reached within local functions
- Function scope cannot be reached within the global scope
- It will make sense with time



```
const applicationName = "calculator";
function sumTwoNumbers(a,b){
    let result = a + b;
    console.log(applicationName + " just printed " + result);
}
```

Agenda

Javascript Functions Advanced

- Arrow functions
- Functions as arguments
- ForEach
- Filter/Map

Function syntax

Overview

- Functions can be declared in different ways
- The three following examples are expressing almost the same
- The arrow function syntax was introduced the latest in the ECMASCRIPT 2015 aka ES6 standard

```
//Function declaration
function sumTwoNumbers(a,b){
    let result = a + b;
    console.log(result);
}

//Function expression
const sumTwoNumbers = function(){
    let result = a + b;
    console.log(result);
}

//Arrow function
const sumTwoNumbers = (a,b) => {
    let result = a + b;
    console.log(result);
}
```

Function declarations

Javascript functions

- Function declarations lives in the global scope
- Function declarations are *hoisted* - they can be accessed and used before they are declared

```
//Function declaration
function sumTwoNumbers(a,b){
    let result = a + b;
    console.log(result);
}
```

```
//Function expression
const sumTwoNumbers = function(){
    let result = a + b;
    console.log(result);
}

//Arrow function
const sumTwoNumbers = (a,b) => {
    let result = a + b;
    console.log(result);
}
```

Function expressions & arrow functions

Javascript functions

- Function declaration lives in the most local scope defined
- Function declarations are not *hoisted* - they cannot be accessed and used before they are declared
- A variable can be assigned to the function
- Implicit return statement in arrow function

```
//Function declaration
function sumTwoNumbers(a,b){
    let result = a + b;
    console.log(result);
}
```

```
//Function expression
const sumTwoNumbers = function(){
    let result = a + b;
    console.log(result);
}

//Arrow function
const sumTwoNumbers = (a,b) => {
    let result = a + b;
    console.log(result);
}
```

Function declarations & arrow functions

Javascript functions

- Similar syntax
- Parameters
- Name
- Body

```
//Function declaration
function sumTwoNumbers(a,b){
   let result = a + b;
   console.log(result);
}

//Arrow function
const sumTwoNumbers = (a,b) => {
   let result = a + b;
   console.log(result);
}
```

Arrow functions

Javascript functions

Example

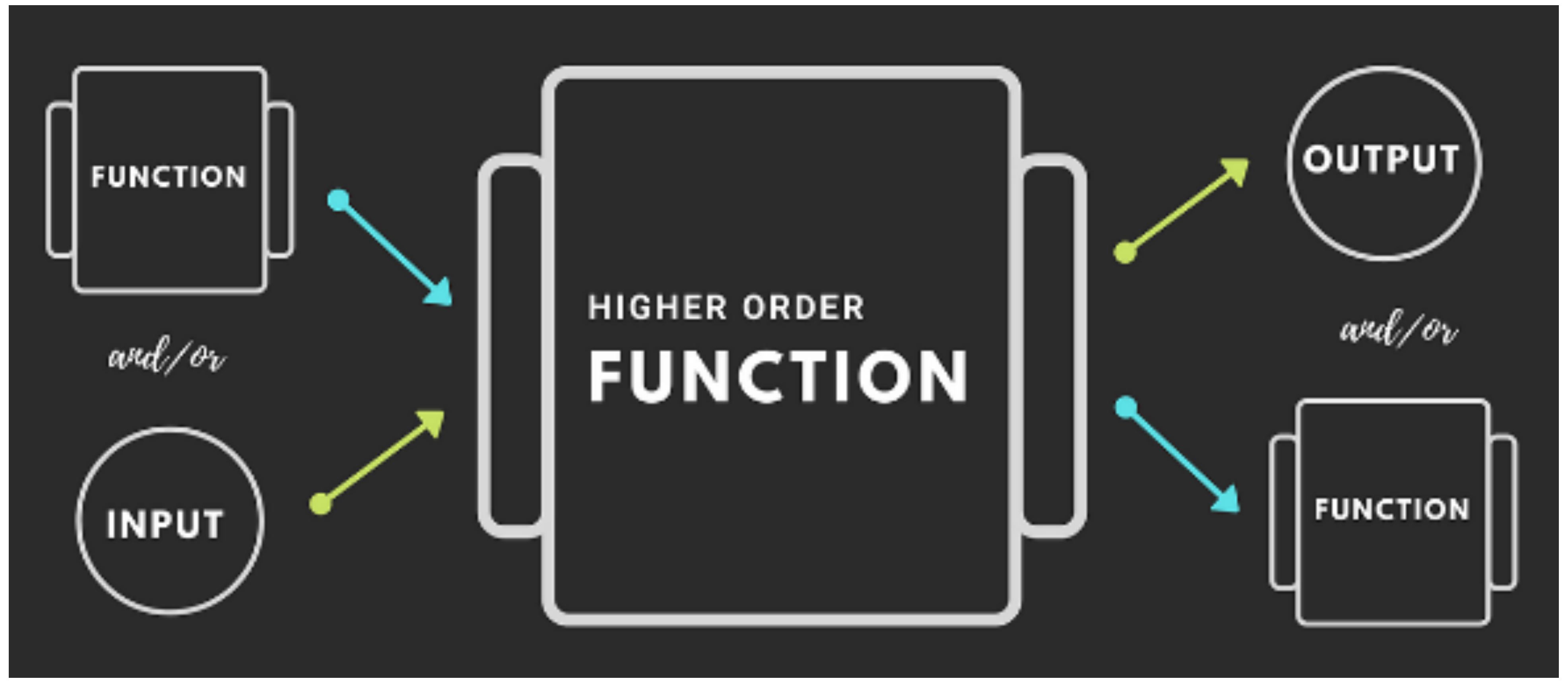
https://github.com/nicklasdean/ita-23-1-sem-code/blob/main/javascript-functions/advanced/arrow-functions.js

Exercises A

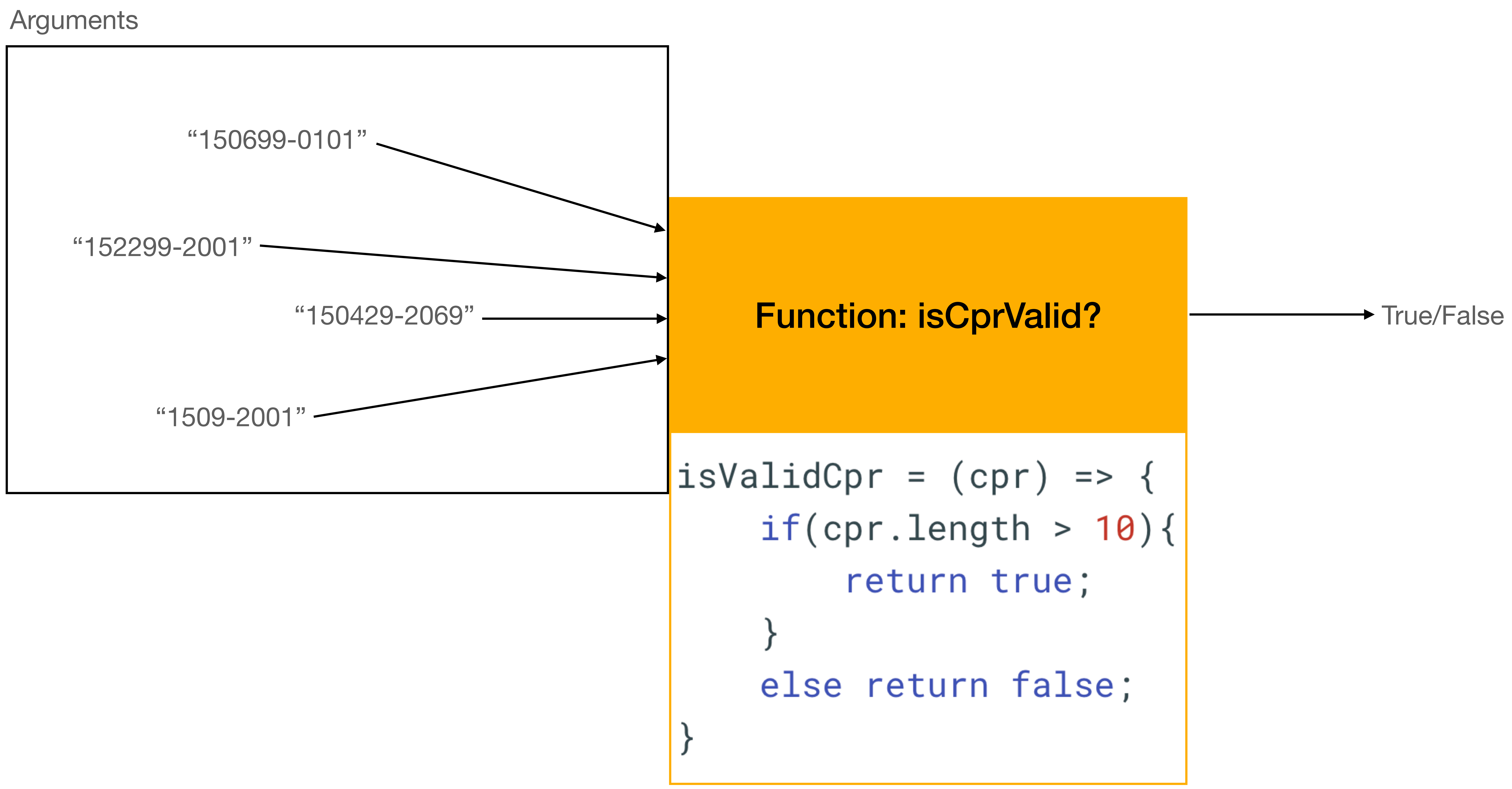
Functions as parameters/arguments

Callback functions

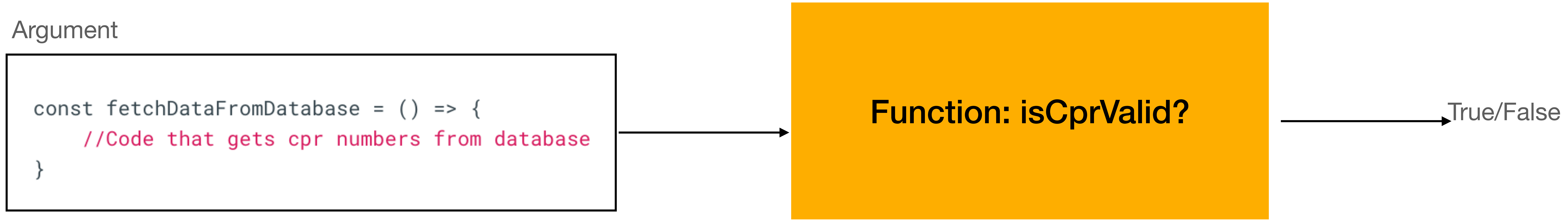
- In Javascript functions are said to be "first class citizens"
- Can be utilised the same as values or objects e.g. a number or a string
- Can be used as parameters/ arguments



A string as an argument



```
isValidCpr = (cpr) => {
    if(cpr.length > 10){
        return true;
    }
    else return false;
}
```



```
const fetchDataFromDatabase = () => {
    //Code that gets cpr numbers from database
}
```

The Foreach Loop

Functions as argument

Example

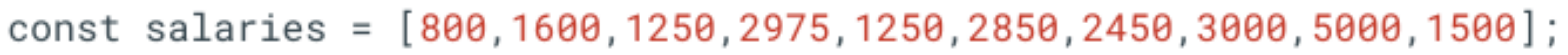
https://github.com/nicklasdean/ita-23-1-sem-code/blob/main/javascript-functions/advanced/foreach-example.js

Javascript array Filter

Filter

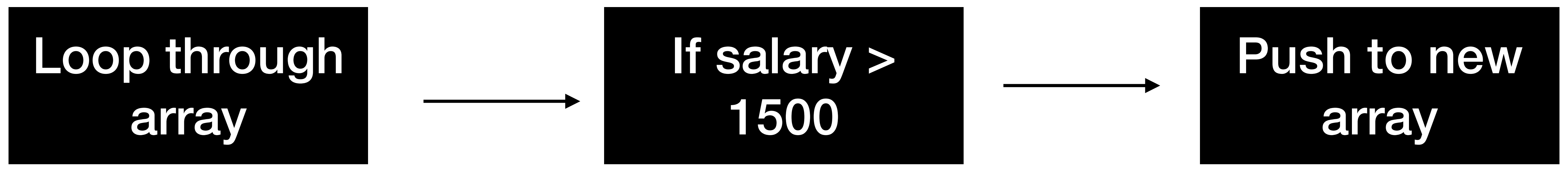
Functions as argument

Task: Find all salaries that are higher than 1500



```
const salaries = [800,1600,1250,2975,1250,2850,2450,3000,5000,1500];
```

Simple strategy



Filter

Functions as argument

- Use a built in function in the array
- The function is able to filter an return a new array
- The function has to know how to filter the data
- We will provide a function as argument to communicate how the data should be filtered

Example

https://github.com/nicklasdean/ita-23-1-sem-code/blob/main/javascript-functions/advanced/filter-example.js

How should the array be filtered?
Defined by a function

```
const salaries = [800,1600,1250,2975,1250,2850,2450,3000,5000,1500];
```

False True False True False True True True True True

```
if (value > 1500){
    return true
}
else{
    return false
}
```

Filter depends on a function that either returns true or false

Exercises B