Front-end

lecture IV/VI

SCHEDULE

Lecture IV (today)

I. HomeWork

   A. Higher order functions

   B. Looping

II. Show Progress

   A. Show progress humans! (x amount of humans)

Homework

IV/VI

IV Higher order functions

- Functional programming (assuming):
  - Less bugs (easier to reason about)
  - Less time (reusable code)
  - More structure

IV: WHAT ARE HOF?!

SYNOPSIS

- ❖ Functions are values
  - ➢ We can assign an anonymous function to a variable 🤯
  - ➢ So we can store functions inside of variables.. we can then pass these functions as arguments.. to other functions!!!! L33tH4x0rrsz 👯‍♀️ - this concept is called: Higher order functions

SYNOPSIS

🤨

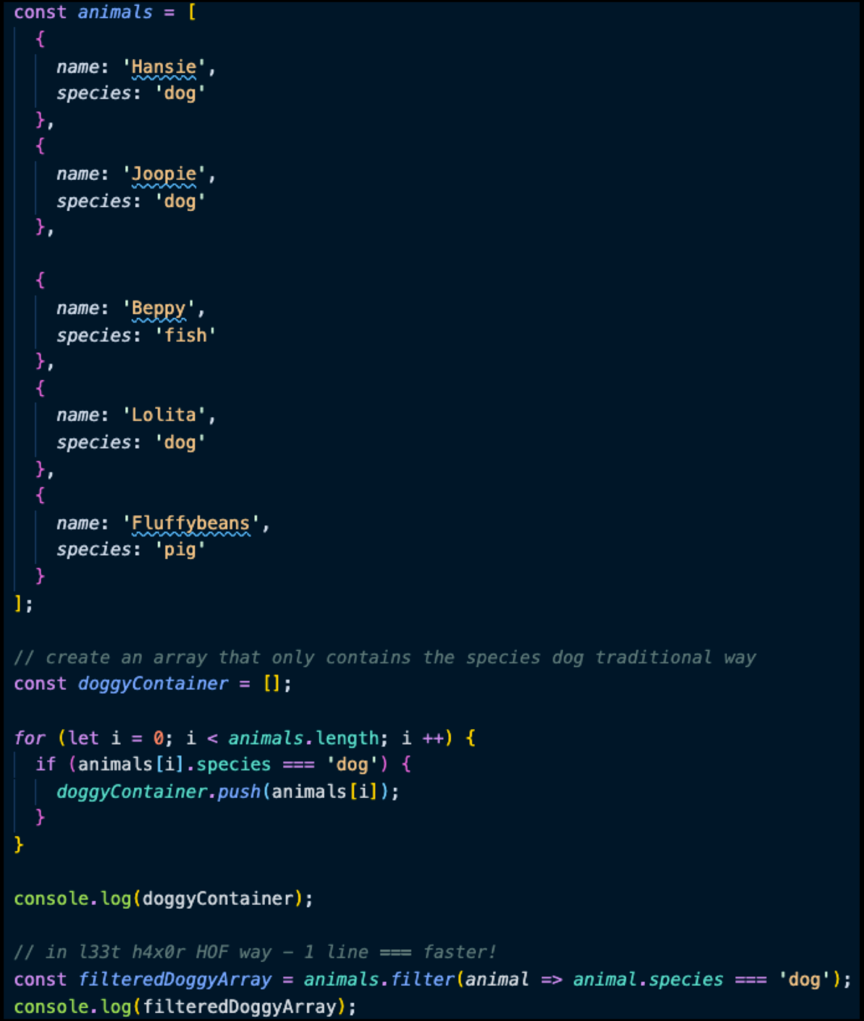
Cool and all.. but what are higher order functions good for?

IV/ HOF

SYNOPSIS

- Composition
  - Compose small functions into big functions..

Show me examples human!



```
const animals = [
  {
    name: 'Hansie',
    species: 'dog'
  },
  {
    name: 'Joopie',
    species: 'dog'
  },

  {
    name: 'Beppy',
    species: 'fish'
  },
  {
    name: 'Lolita',
    species: 'dog'
  },
  {
    name: 'Fluffybeans',
    species: 'pig'
  }
];

// create an array that only contains the species dog traditional way
const doggyContainer = [];

for (let i = 0; i < animals.length; i ++) {
  if (animals[i].species === 'dog') {
    doggyContainer.push(animals[i]);
  }
}

console.log(doggyContainer);

// in l33t h4x0r HOF way - 1 line === faster!
const filteredDoggyArray = animals.filter(animal => animal.species === 'dog');
console.log(filteredDoggyArray);
```

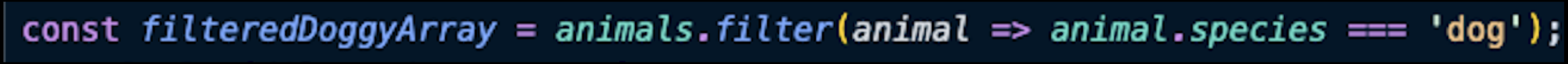
SYNOPSIS

- Write less logic === less code
- Compose small function into another function
- Readability - understandability
- HOF ftw.

SYNOPSIS

So of what does a filter(); exist?

SYNOPSIS



```
const filteredDoggyArray = animals.filter(animal => animal.species === 'dog');
```

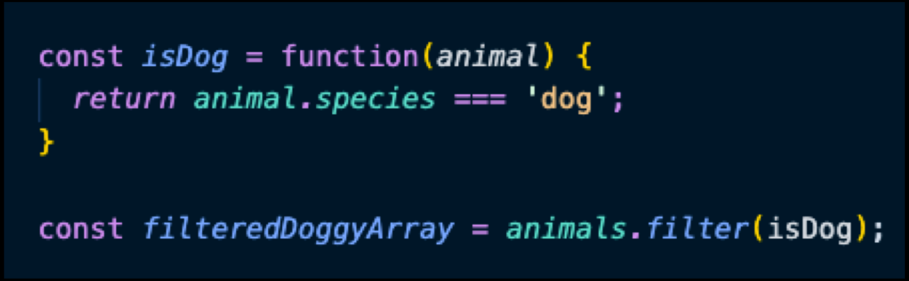
Filter takes in a function as argument - which is referred to as a callback function.

Filter loops through each item in the array - which it will check the condition on - which validates in true or false.

SYNOPSIS

Filter allows us the separation of concerns

SYNOPSIS



```
const isDog = function(animal) {
  return animal.species === 'dog';
}

const filteredDoggyArray = animals.filter(isDog);
```

The callback function is now stored into its own variable - namely; isDog.

SYNOPSIS

Does this mean I should always and forever use HOF and never again a for loop!?

No.

You should know it’s out there and how to use it. Whether or not to use, depends on the use case.

IV: Practise

https://www.youtube.com/watch?v=eGDBR2L5kzI

IV: Looping

SYNOPSIS

- In JS we have several ways to loop over Arrays and Objects. Popular Array looping methods are;
    - map 🗺️
        - map over an array and return specific data entries
    - filter 🐶
        - filter out data entries relevant to the case
    - reduce 🧙‍♂️
        - reduce to a sum or a tally

IV: Map filter reduce

- Rock the map(); filter(); and reduce(); exercises on codepen;

  https://codepen.io/turiGuiliano/pen/RwoBwmG

#IMPORTANT!: EXAM COMING UP

Exam: a1. (13-03)

Deadline a1. Tuesday 09-03 - Submit a link to your FED WiKi page on DLO

SYNOPSIS

Homework

- **A1: Hoisting**
- **A1: Closures**

- Continue working on FE feature
- Map / filter / reduce exercises ☒

Please document your findings in the wiki.

EXIT;

SEE YOU Next();