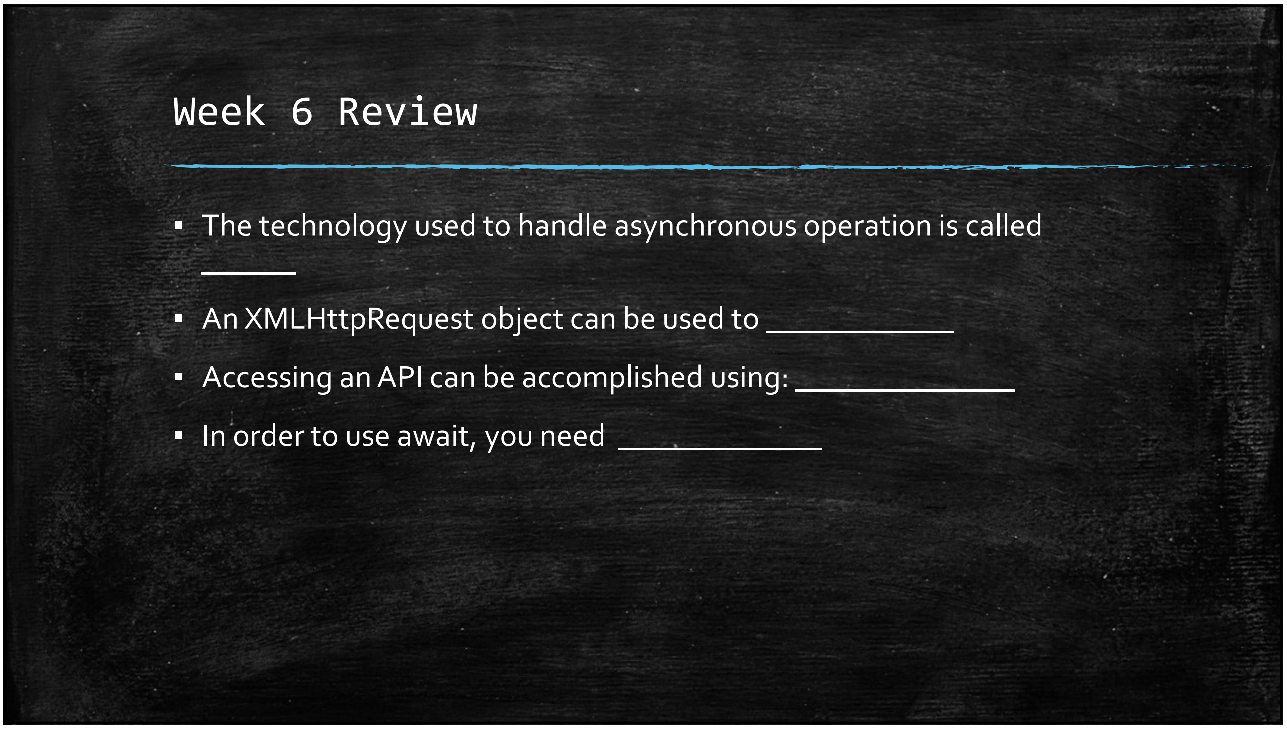
Week 7

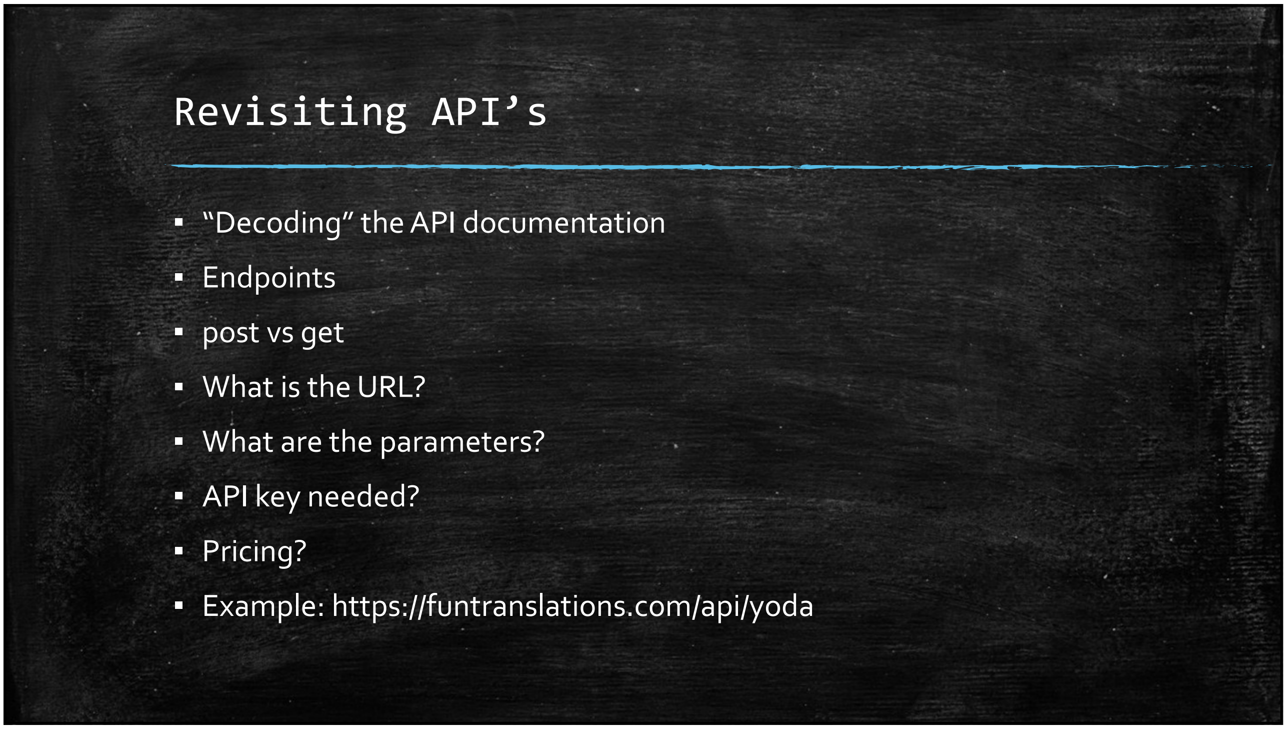
184



Week 6 Review

- The technology used to handle asynchronous operation is called ______
- An XMLHttpRequest object can be used to ____________
- Accessing an API can be accomplished using: ______________
- In order to use await, you need _____________

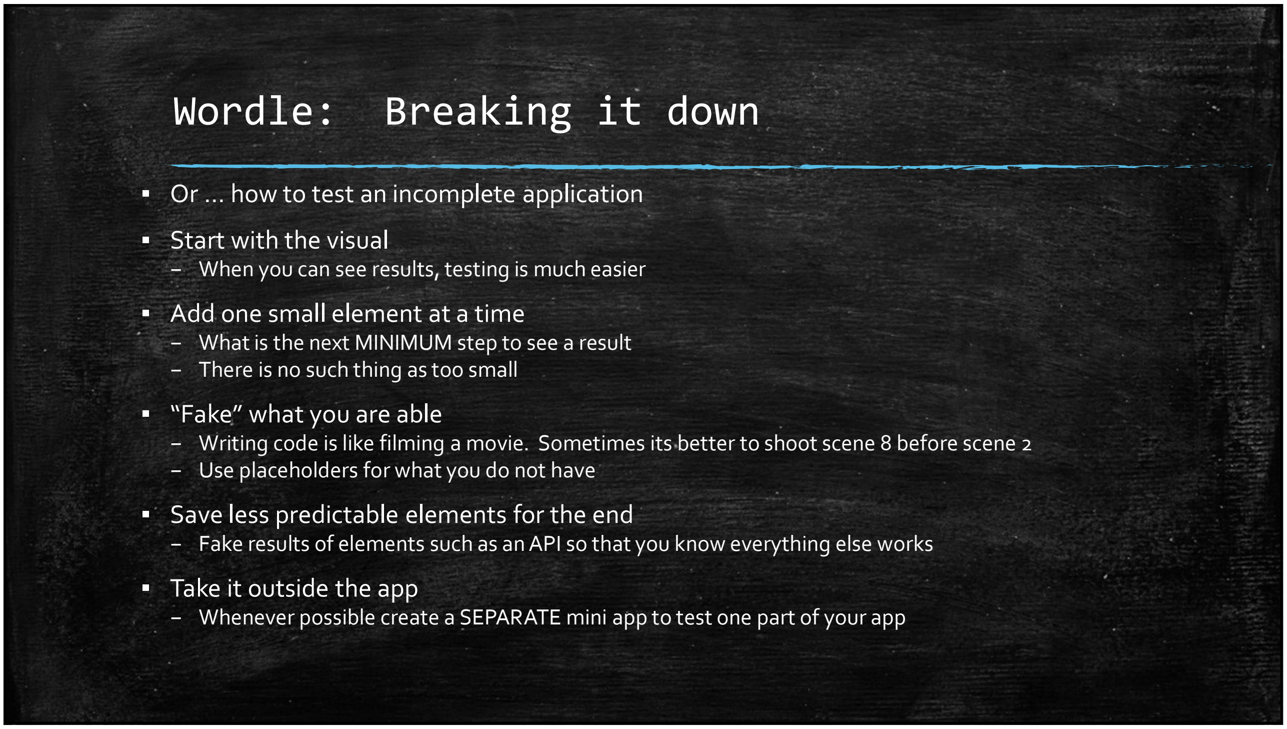
185



Revisiting API’s

- “Decoding” the API documentation
- Endpoints
- post vs get
- What is the URL?
- What are the parameters?
- API key needed?
- Pricing?
- Example: https://funtranslations.com/api/yoda

186



Wordle: Breaking it down

- Or … how to test an incomplete application
- Start with the visual
  - When you can see results, testing is much easier
- Add one small element at a time
  - What is the next MINIMUM step to see a result
  - There is no such thing as too small
- “Fake” what you are able
  - Writing code is like filming a movie. Sometimes its better to shoot scene 8 before scene 2
  - Use placeholders for what you do not have
- Save less predictable elements for the end
  - Fake results of elements such as an API so that you know everything else works
- Take it outside the app
  - Whenever possible create a SEPARATE mini app to test one part of your app

187

Functional Programming

- From Wikipedia:
  In computer science, functional programming is a programming paradigm where programs are constructed by applying and composing functions. It is a *declarative* programming paradigm in which function definitions are trees of expressions that map values to other values, rather than a sequence of *imperative* statements which update the running state of the program.
- This works because of how functions behave in JavaScript:
  - They can be assigned to a variable
  - They can be passed as a parameter
  - They can be returned from a function
  - They can be array or object values

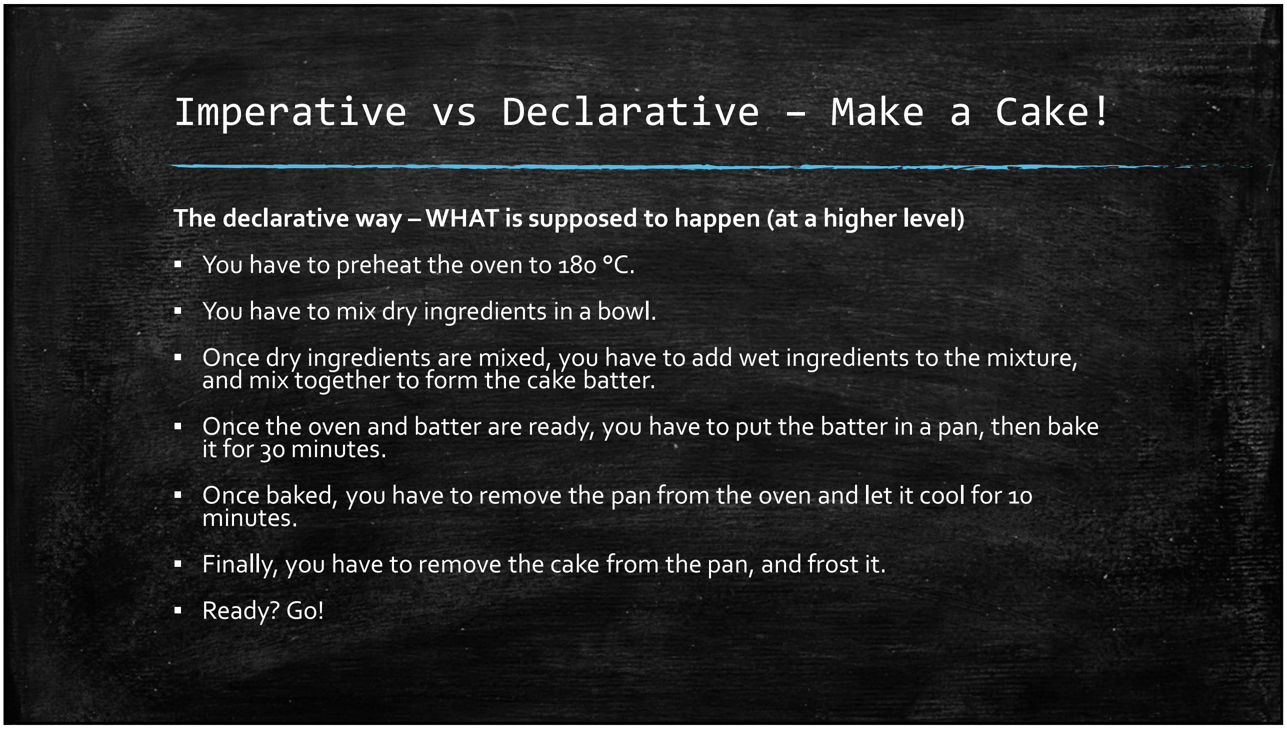
188

Imperative vs Declarative – Make a Cake!

The imperative way- HOW is it supposed to happen

- First, turn on the oven to preheat it at 180°C.
- Next, add flour, sugar, cocoa powder, baking soda and salt to a large bowl, then stir the mixture with a paddle.
- Then, add milk, vegetable oil, eggs and vanilla extract to the mixture, and mix together on medium speed until well combined.
- Distribute the cake batter evenly in a large cake pan, then bake it for approx. 30 minutes.
- Remove the pan from the oven with a pot holder, let it cool for 10 minutes.
- Finally, remove the cake from the pan with the tapping method, and frost it evenly with chocolate frosting.

189

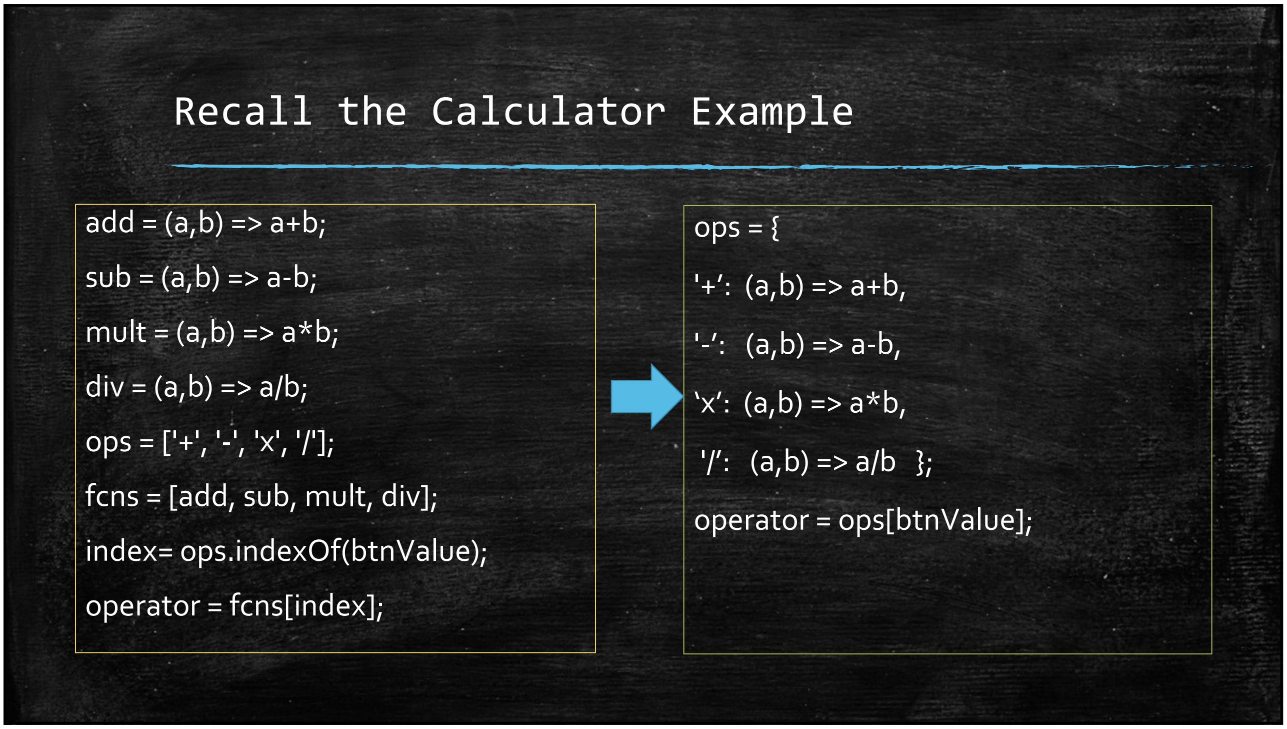


Imperative vs Declarative – Make a Cake!

The declarative way – WHAT is supposed to happen (at a higher level)

- You have to preheat the oven to 180 °C.
- You have to mix dry ingredients in a bowl.
- Once dry ingredients are mixed, you have to add wet ingredients to the mixture, and mix together to form the cake batter.
- Once the oven and batter are ready, you have to put the batter in a pan, then bake it for 30 minutes.
- Once baked, you have to remove the pan from the oven and let it cool for 10 minutes.
- Finally, you have to remove the cake from the pan, and frost it.
- Ready? Go!

190

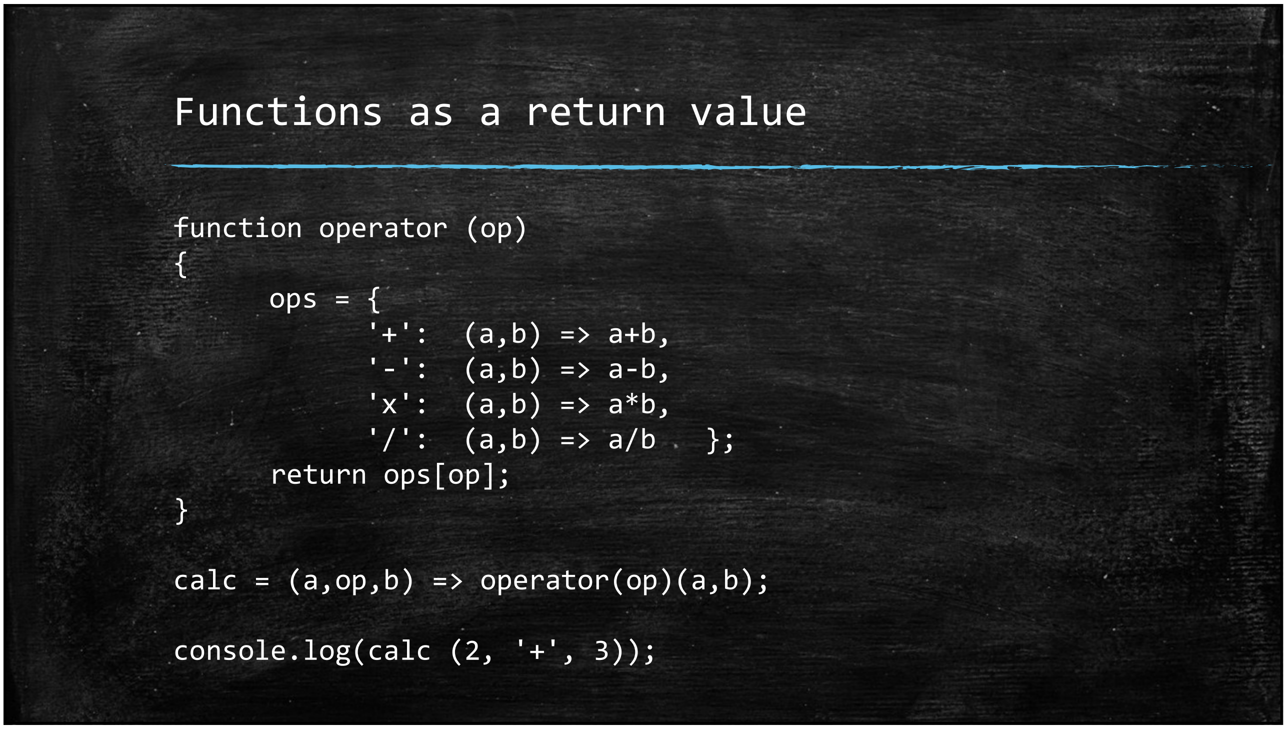


Recall the Calculator Example

```
add = (a,b) => a+b;
sub = (a,b) => a-b;
mult = (a,b) => a*b;
div = (a,b) => a/b;
ops = ['+', '-', 'x', '/'];
fcns = [add, sub, mult, div];
index= ops.indexOf(btnValue);
operator = fcns[index];
```

```
ops = {
'+’:  (a,b) => a+b,
'-’:  (a,b) => a-b,
‘x’:  (a,b) => a*b,
 '/’:  (a,b) => a/b   };
operator = ops[btnValue];
```

191



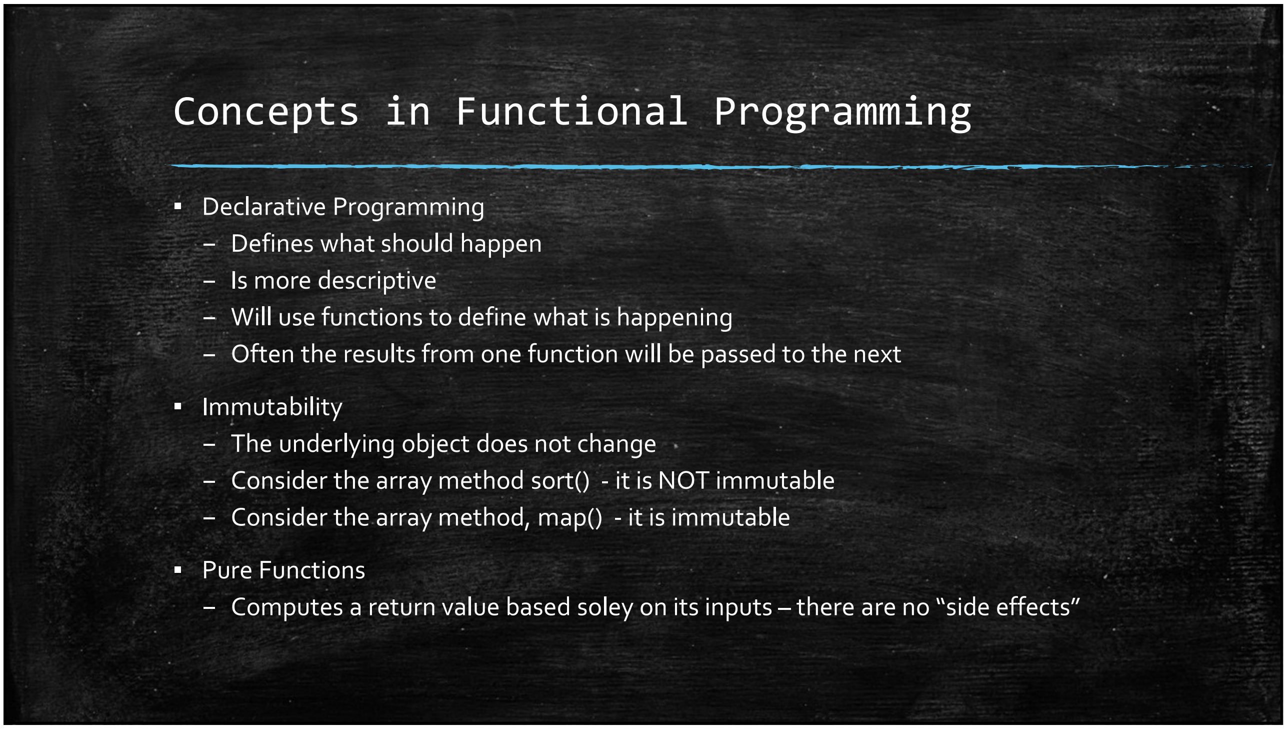
Functions as a return value

```
function operator (op)
{
      ops = {
            '+':  (a,b) => a+b,
            '-':  (a,b) => a-b,
            'x':  (a,b) => a*b,
            '/':  (a,b) => a/b   };
      return ops[op];
}

calc = (a,op,b) => operator(op)(a,b);

console.log(calc (2, '+', 3));
```

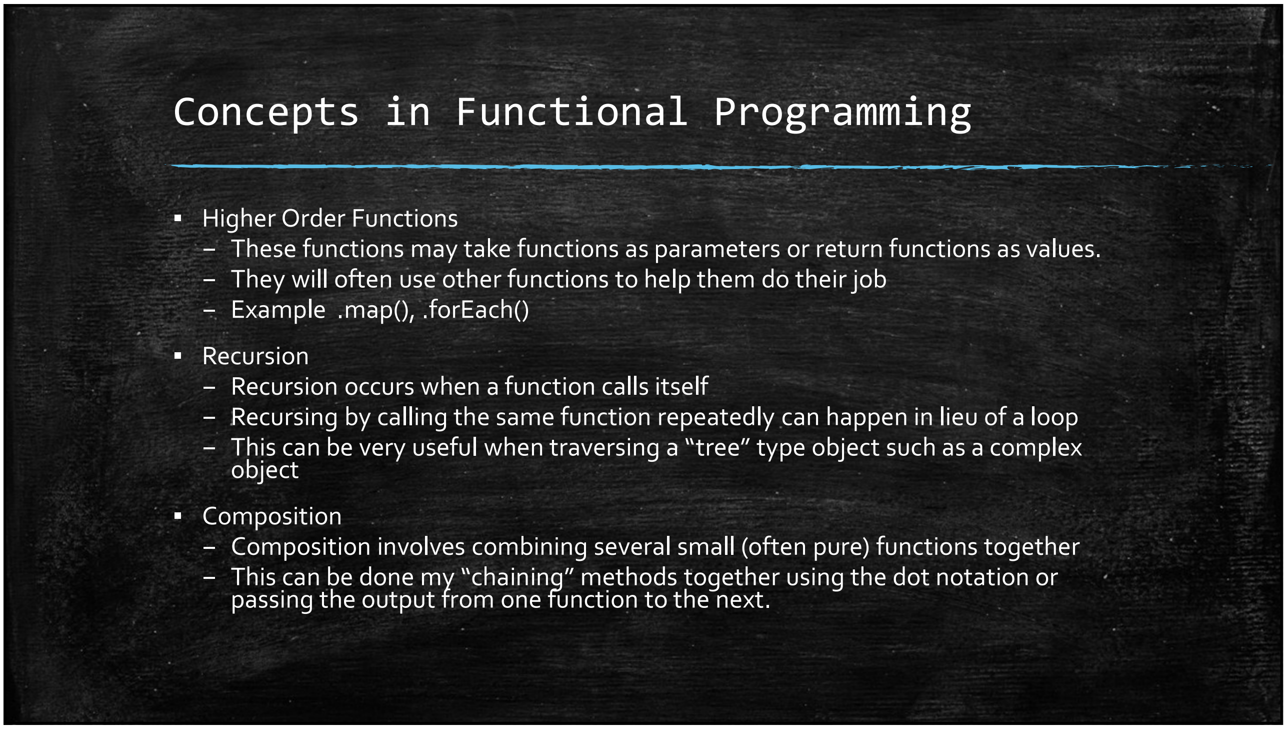
192



Concepts in Functional Programming

- Declarative Programming
  - Defines what should happen
  - Is more descriptive
  - Will use functions to define what is happening
  - Often the results from one function will be passed to the next
- Immutability
  - The underlying object does not change
  - Consider the array method sort() - it is NOT immutable
  - Consider the array method, map() - it is immutable
- Pure Functions
  - Computes a return value based soley on its inputs – there are no "side effects"

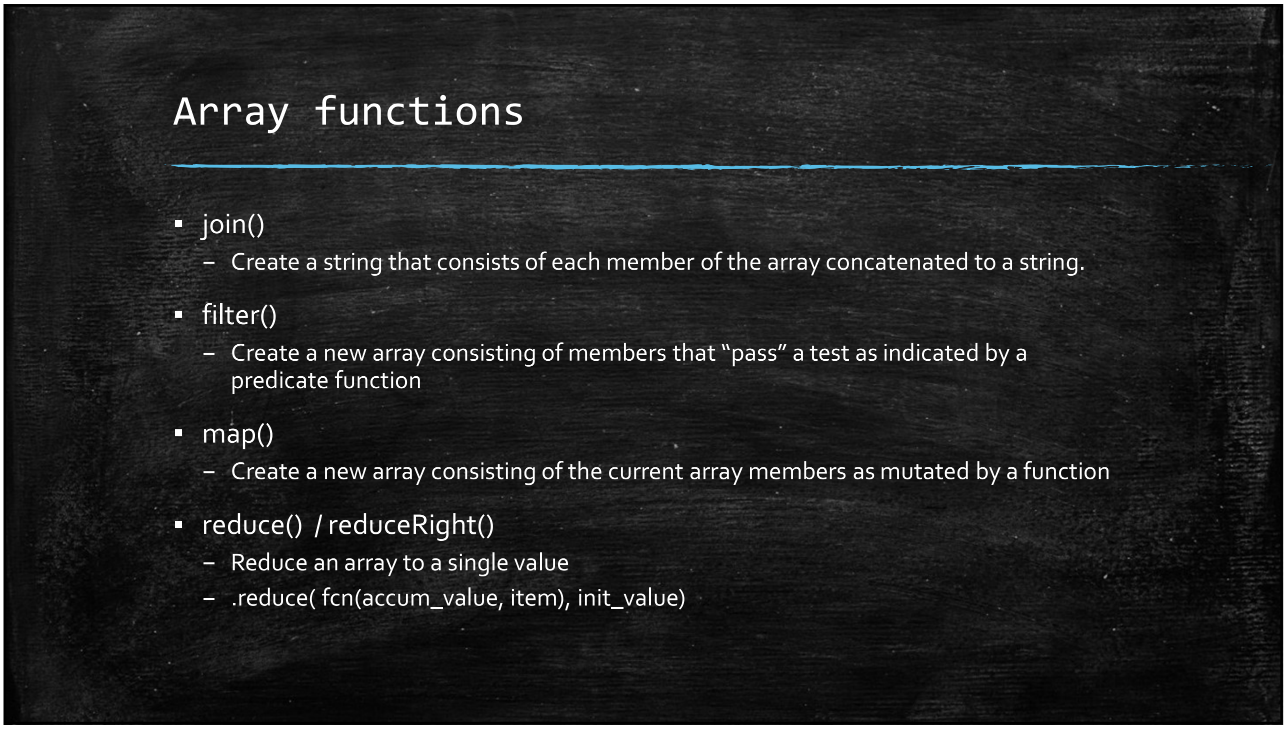
193



Concepts in Functional Programming

- Higher Order Functions
  - These functions may take functions as parameters or return functions as values.
  - They will often use other functions to help them do their job
  - Example .map(), .forEach()
- Recursion
  - Recursion occurs when a function calls itself
  - Recursing by calling the same function repeatedly can happen in lieu of a loop
  - This can be very useful when traversing a "tree" type object such as a complex object
- Composition
  - Composition involves combining several small (often pure) functions together
  - This can be done my "chaining" methods together using the dot notation or passing the output from one function to the next.

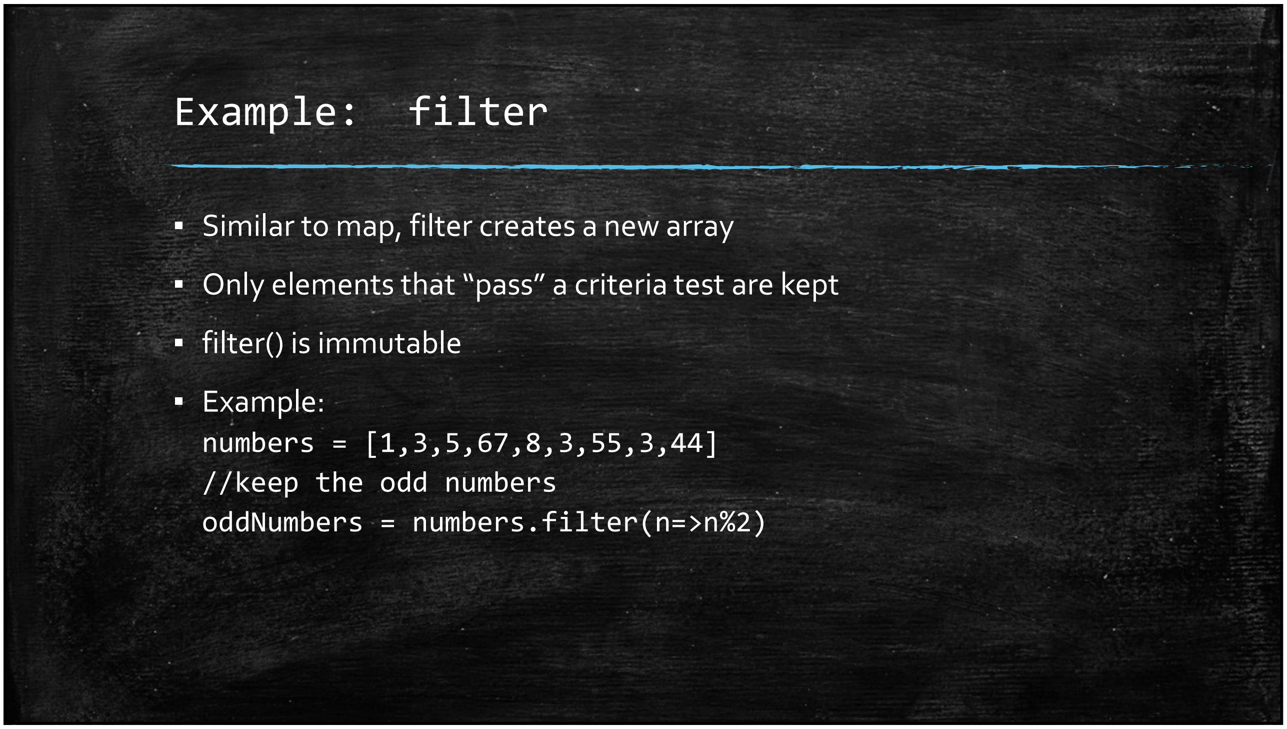
194



Array functions

- join()
  - Create a string that consists of each member of the array concatenated to a string.
- filter()
  - Create a new array consisting of members that "pass" a test as indicated by a predicate function
- map()
  - Create a new array consisting of the current array members as mutated by a function
- reduce() / reduceRight()
  - Reduce an array to a single value
  - .reduce(fcn(accum_value, item), init_value)

195

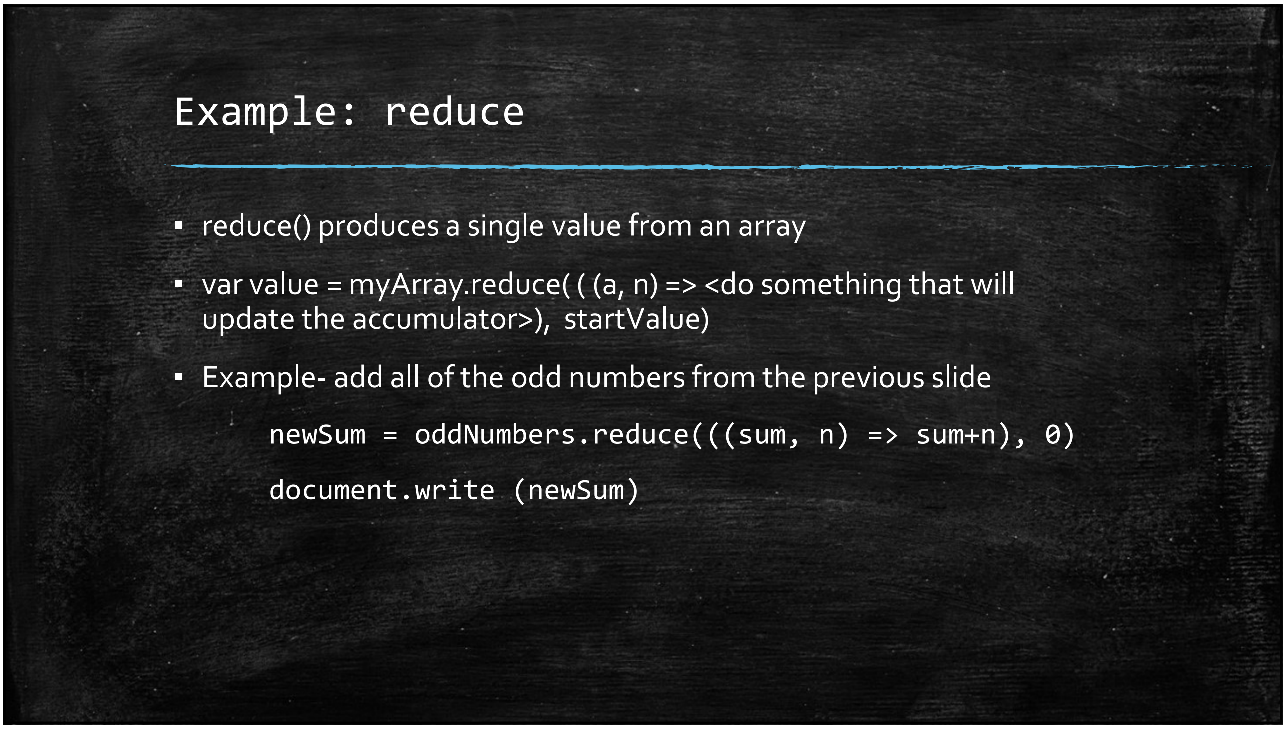


Example: filter

- Similar to map, filter creates a new array
- Only elements that “pass” a criteria test are kept
- filter() is immutable
- Example:

```
numbers = [1,3,5,67,8,3,55,3,44]
//keep the odd numbers
oddNumbers = numbers.filter(n=>n%2)
```

196



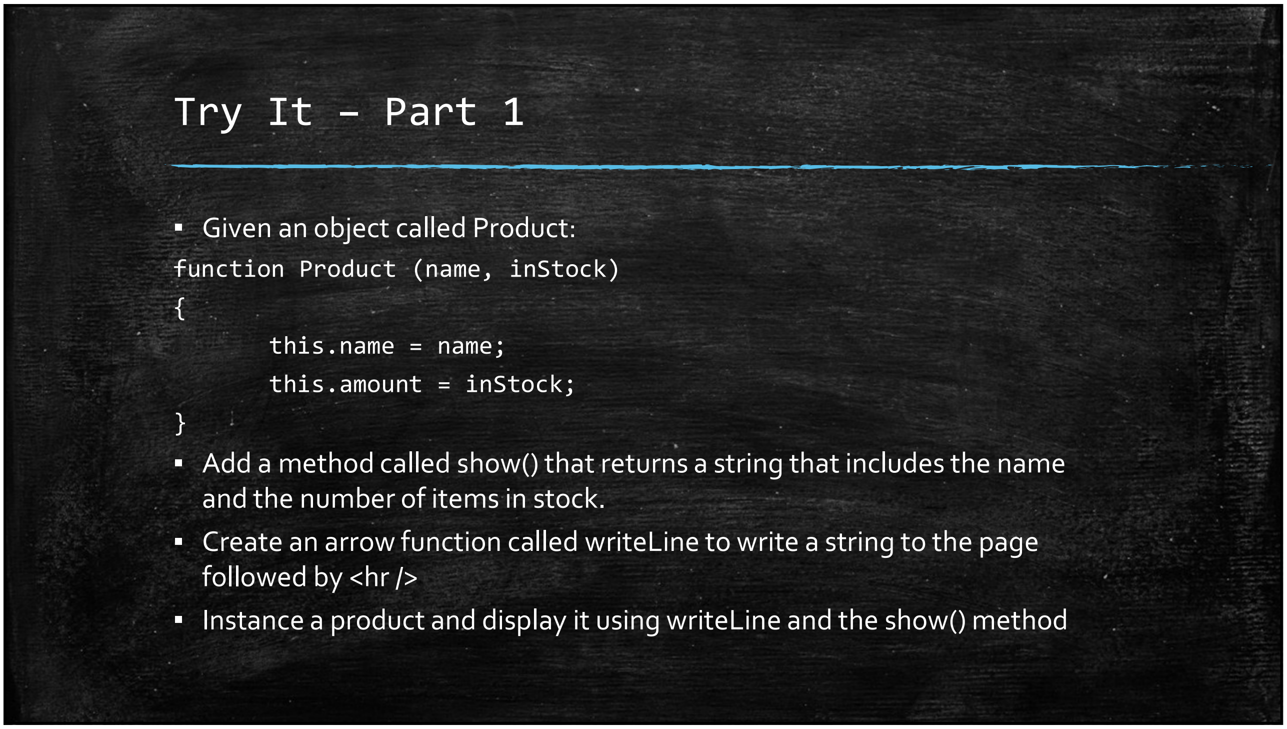
Example: reduce

- reduce() produces a single value from an array
- var value = myArray.reduce(((a, n) => <do something that will update the accumulator>), startValue)
- Example- add all of the odd numbers from the previous slide

```
    newSum = oddNumbers.reduce(((sum, n) => sum+n), 0)

    document.write (newSum)
```

197



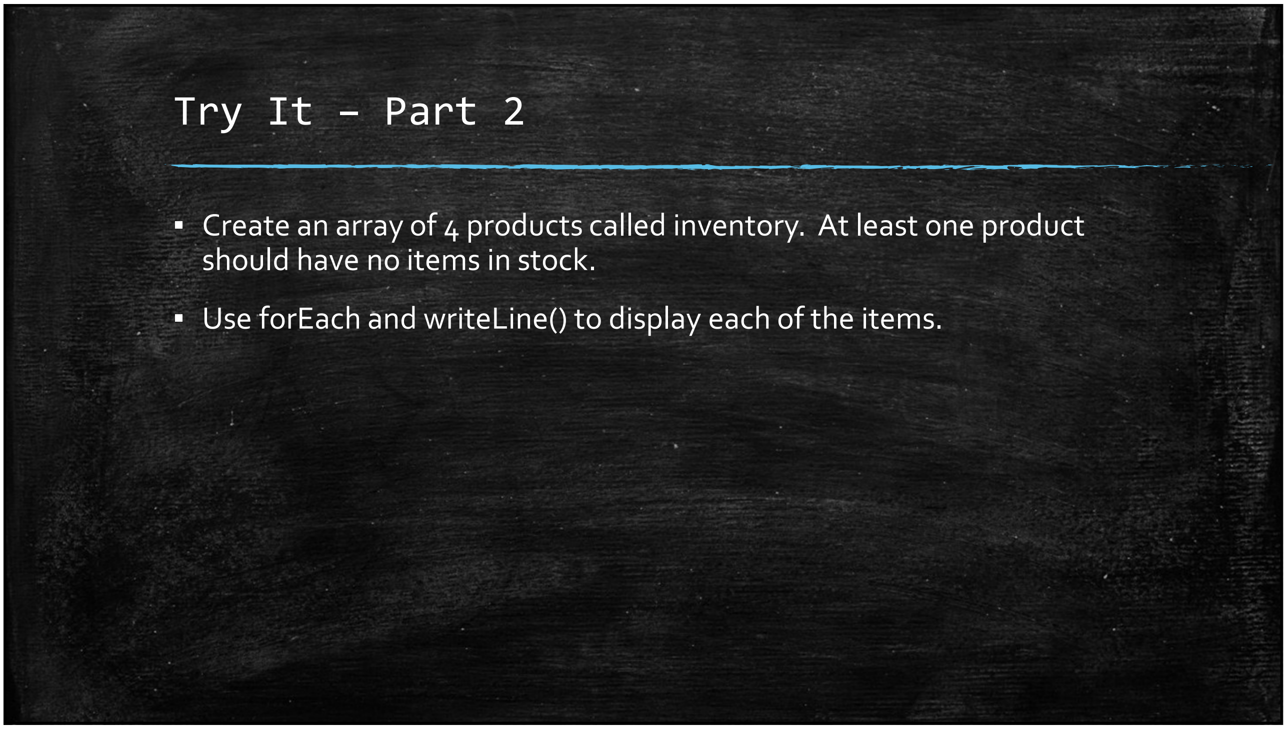
Try It – Part 1

- Given an object called Product:

```
function Product (name, inStock)
{
        this.name = name;
        this.amount = inStock;
}
```

- Add a method called show() that returns a string that includes the name and the number of items in stock.
- Create an arrow function called writeLine to write a string to the page followed by <hr />
- Instance a product and display it using writeLine and the show() method

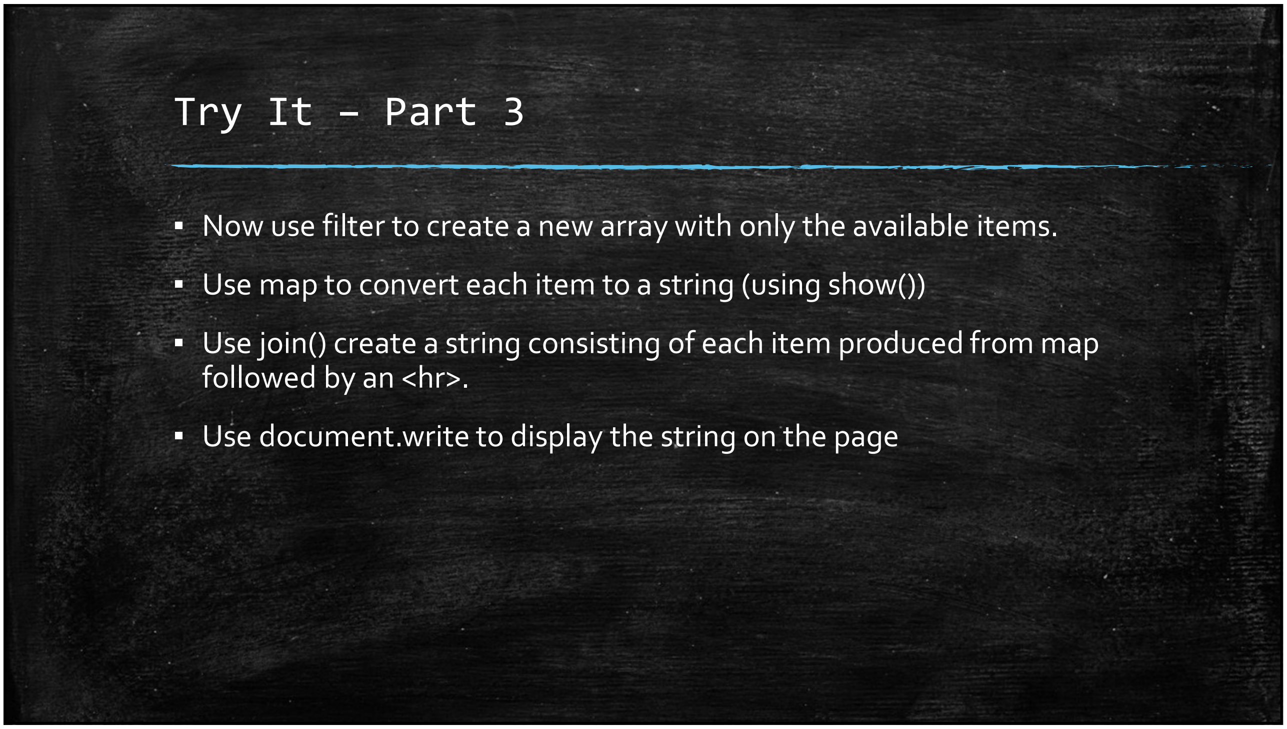
198



Try It – Part 2

- Create an array of 4 products called inventory. At least one product should have no items in stock.
- Use forEach and writeLine() to display each of the items.

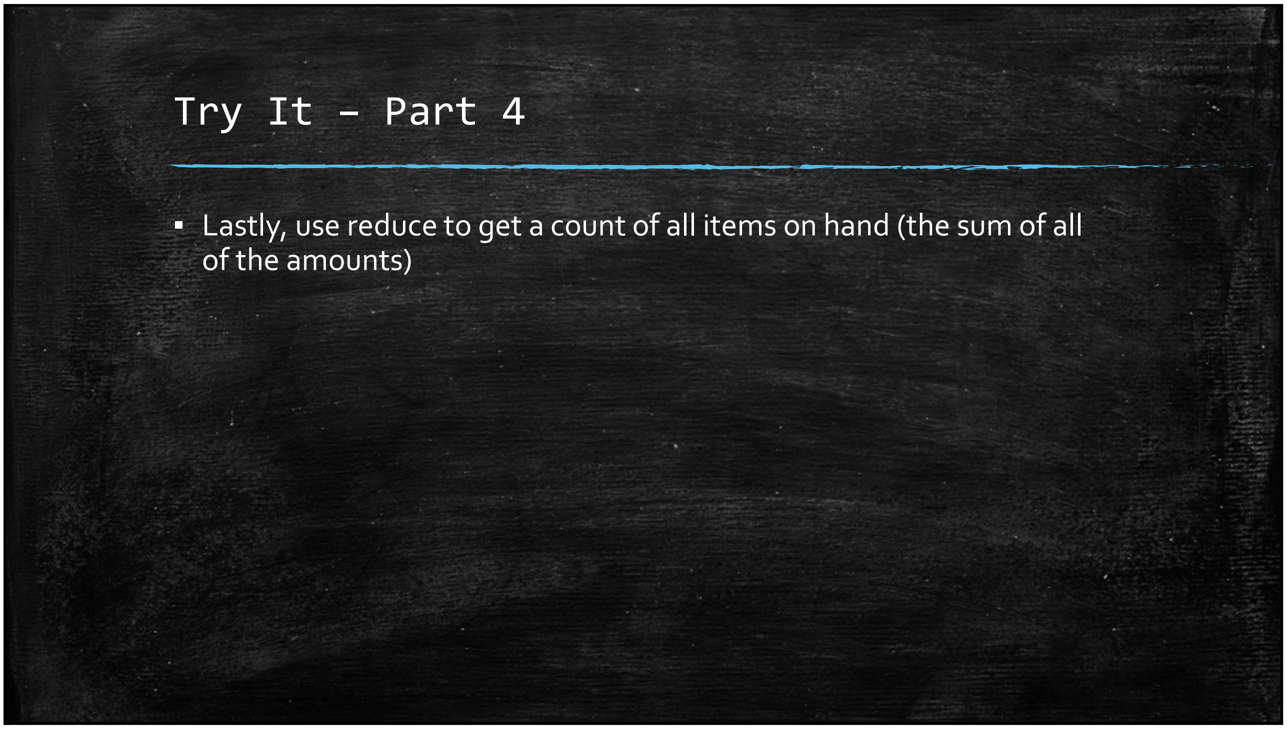
199



Try It – Part 3

- Now use filter to create a new array with only the available items.
- Use map to convert each item to a string (using show())
- Use join() create a string consisting of each item produced from map followed by an <hr>.
- Use document.write to display the string on the page

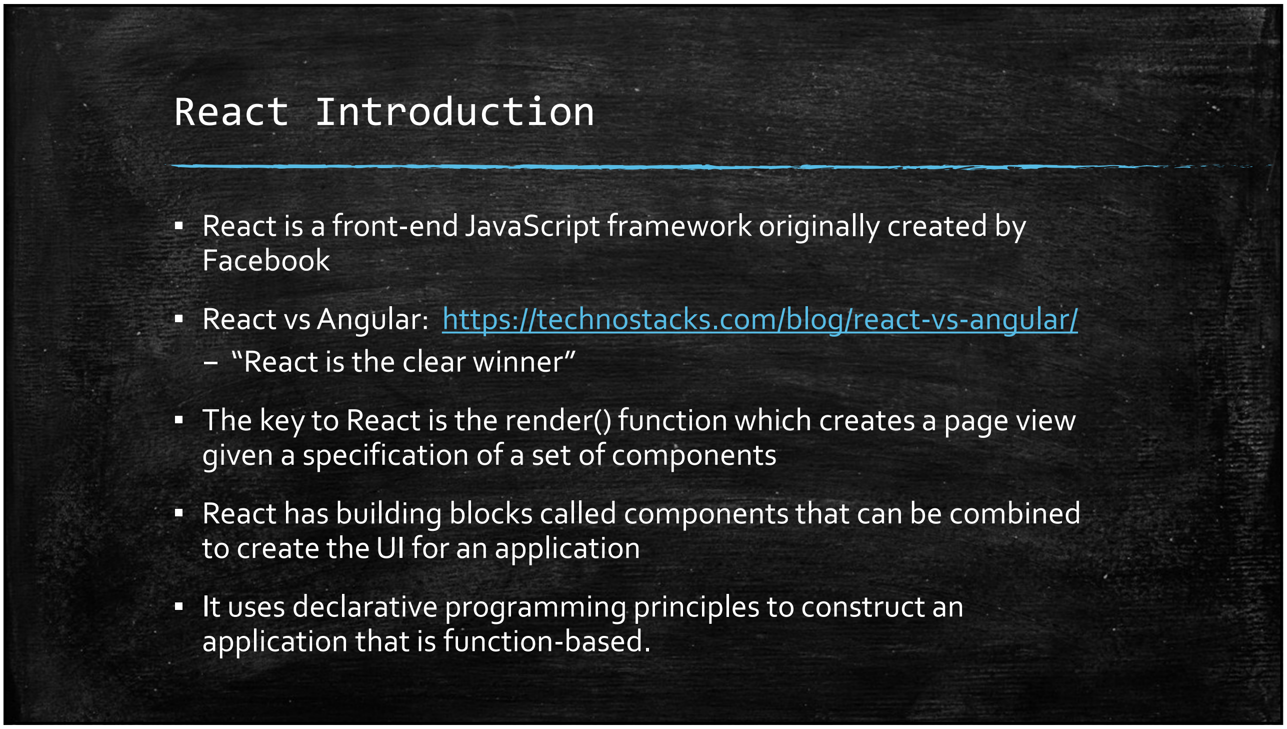
200



Try It – Part 4

- Lastly, use reduce to get a count of all items on hand (the sum of all of the amounts)

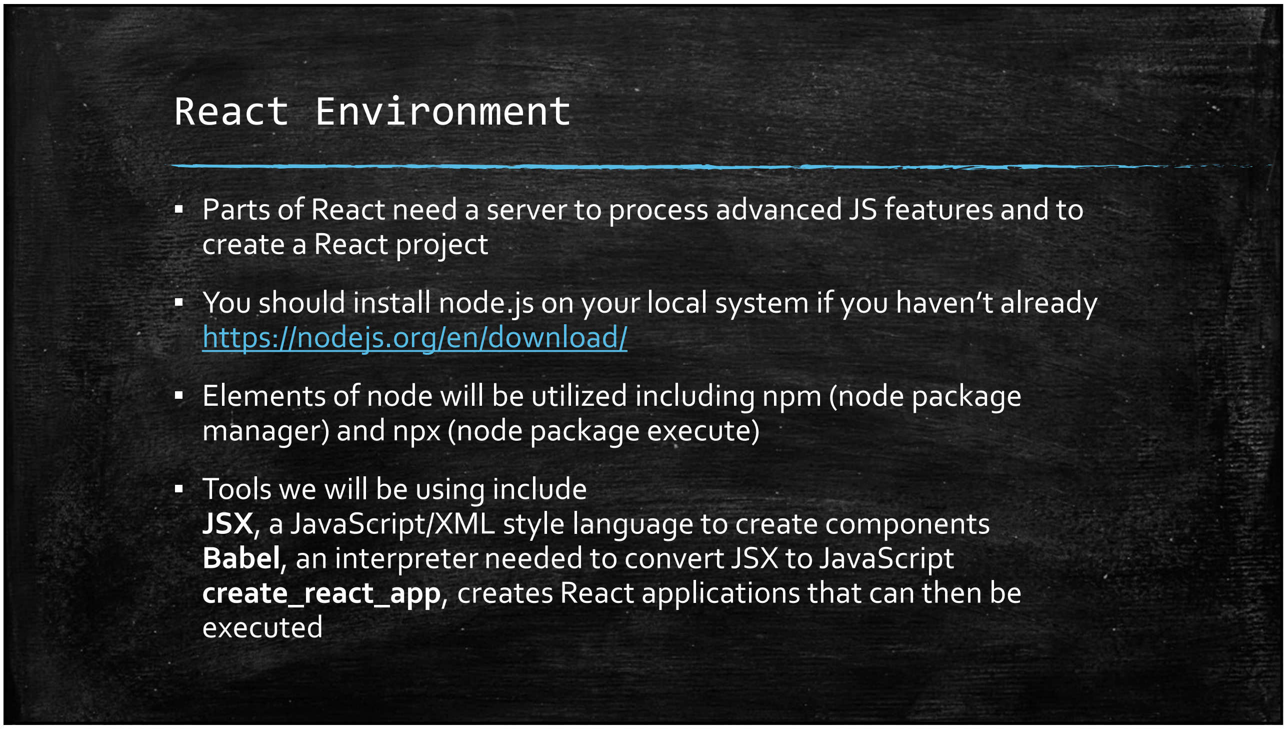
201



React Introduction

- React is a front-end JavaScript framework originally created by Facebook
- React vs Angular: https://technostacks.com/blog/react-vs-angular/
  - "React is the clear winner"
- The key to React is the render() function which creates a page view given a specification of a set of components
- React has building blocks called components that can be combined to create the UI for an application
- It uses declarative programming principles to construct an application that is function-based.

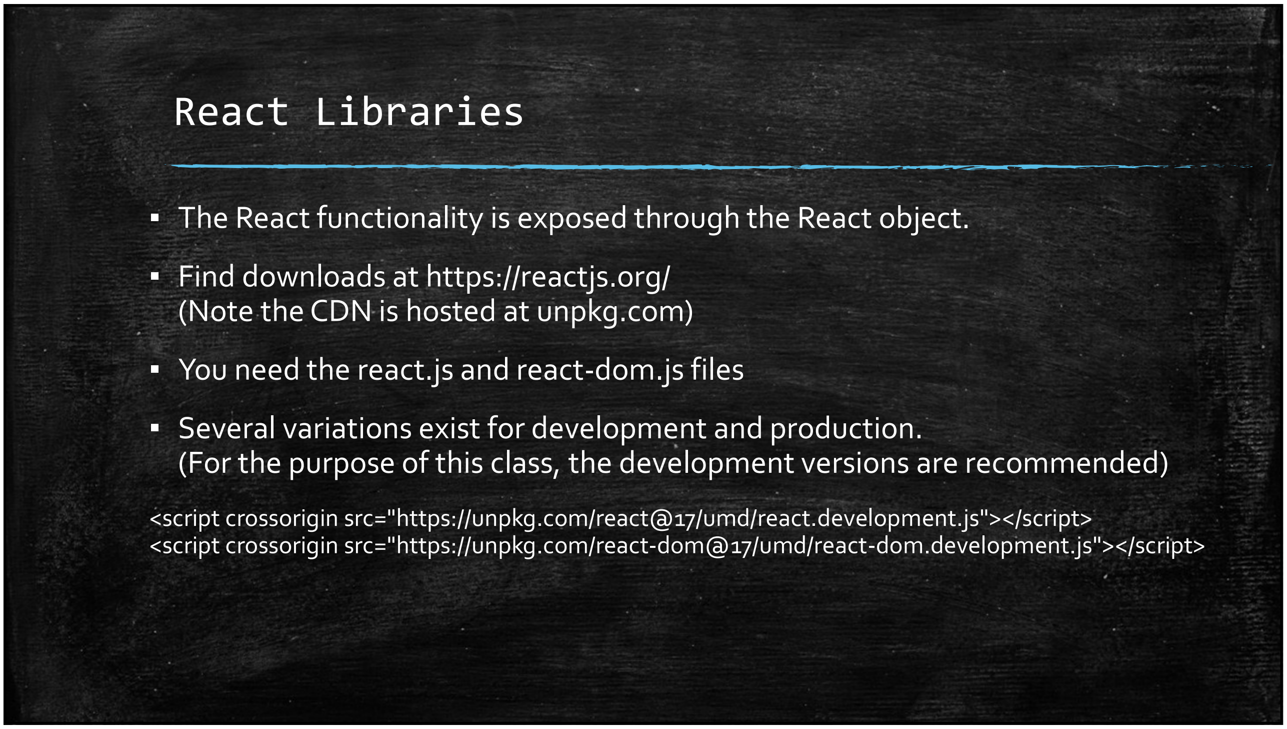
202



React Environment

- Parts of React need a server to process advanced JS features and to create a React project
- You should install node.js on your local system if you haven't already https://nodejs.org/en/download/
- Elements of node will be utilized including npm (node package manager) and npx (node package execute)
- Tools we will be using include
  JSX, a JavaScript/XML style language to create components
  Babel, an interpreter needed to convert JSX to JavaScript
  create_react_app, creates React applications that can then be executed

203



React Libraries

- The React functionality is exposed through the React object.
- Find downloads at https://reactjs.org/
  (Note the CDN is hosted at unpkg.com)
- You need the react.js and react-dom.js files
- Several variations exist for development and production.
  (For the purpose of this class, the development versions are recommended)

```
<script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
<script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
```