What is DSC NUS?

DSC NUS is made up of people from diverse backgrounds.

We come together to push our mission and impact communities. #TechforGood

Developer Student Clubs

Who are **we**

Upcoming Events

DSC NUS Hack for Good

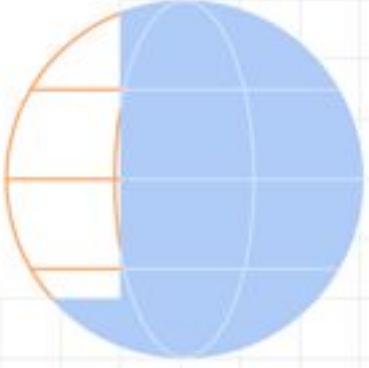
(more announcement soon!)

React FUNdamentals

Developer Student Clubs
National University of Singapore

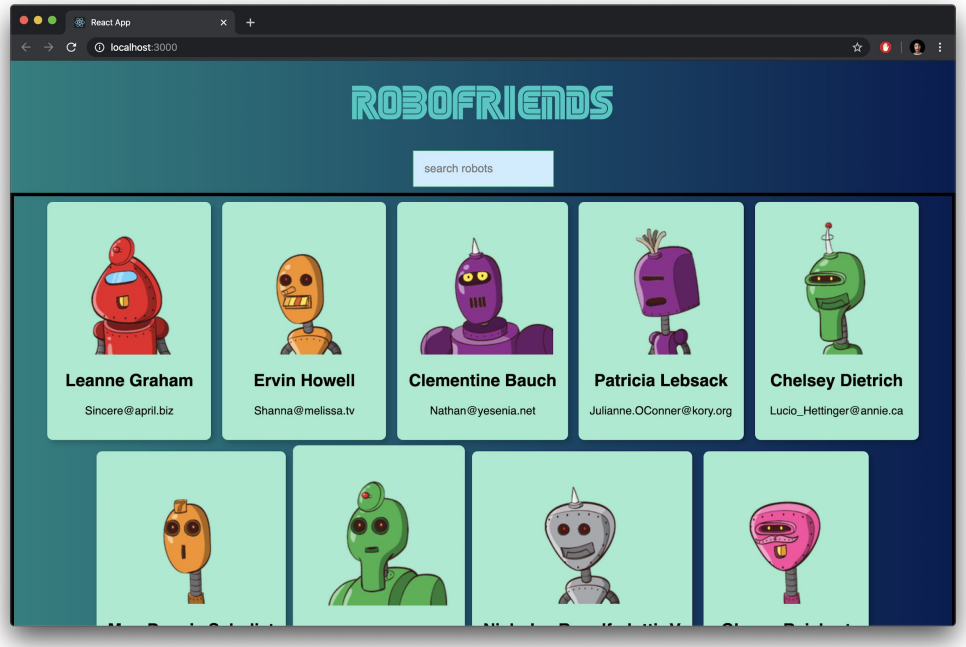
Speakers:

- Kyler
- En Hao
- Andre
- Jun Hup



Brief Overview + Setup

What we will be building today



Workshop pre-installation

- You will need a **code editor.** We recommend **VSCode**
- Download it from: https://code.visualstudio.com/

Workshop pre-installation

- You will need a **Github** account
- (If you are using Windows) Ensure you have downloaded **Git Bash**
  - Download it from: https://git-scm.com/downloads

Workshop pre-installation

- You will need **NodeJS** installed
- Download it from: https://nodejs.org/en/download/current/

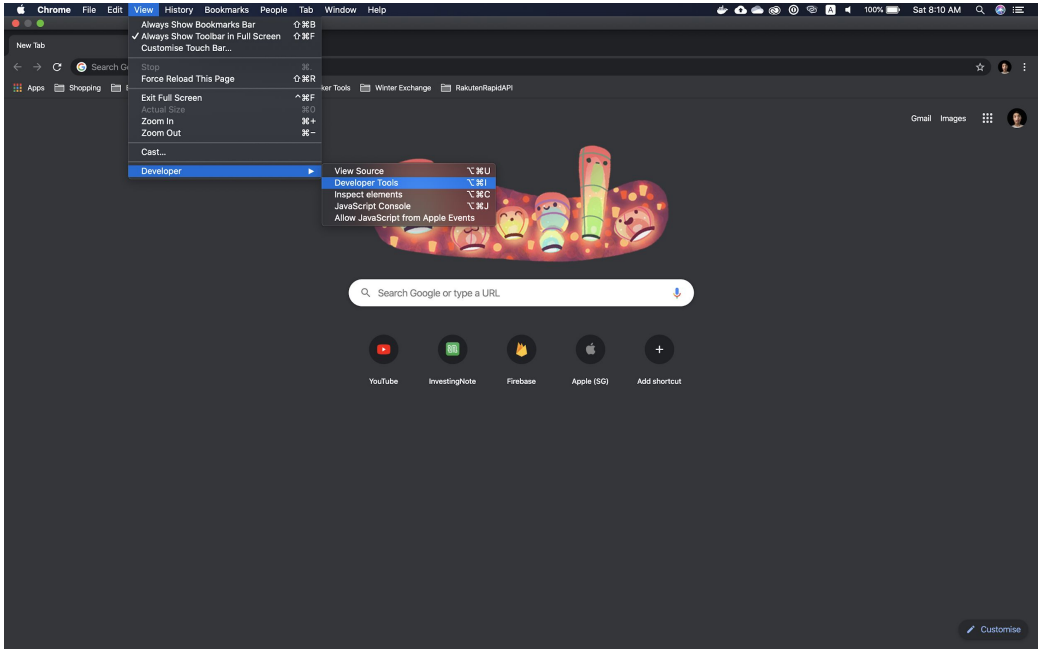
Using the Node Package Manager (NPM)

- You can run a website on your machine without a web host
- Use your machine as a host (localhost)
- Used mainly for development
- Related commands (to be run with NodeJS installed on device)
  - `node install`
  - `node run start`

JavaScript

Developer Student Clubs

We will be using the Developer Console



What and why is JavaScript

- JavaScript is ***different*** from Java
- JavaScript is a ***loosely typed scripting*** language (does not need variable to be defined)
- Interactive websites needs to have underlying logic
- JavaScript brings logical flows to websites and many development frameworks
- TLDR; if you want to be a web developer, JavaScript is important

JavaScript Overview

- Create variables: `let`
- Create immutable values: `const`
- Writing functions: `function`
- Conditionals: `if (condition)`
- For-loops: `for (let i = 0; i < n; i++)`
- While-loops: `while (i < n)`
- Display results on screen: `console.log(result)`

Common Commands

- Displays a message: `alert()`
- Asks for user input: `prompt()`
- Popup but with cancel button: `confirm()`

Variables and Constants

Variables

- `let pi = 3.142` `// A variable `pi` that stores value 3.142`
- `console.log(pi)` `// 3.142 is shown in the console screen`
- `pi = 'pie'` `// pi now stores ‘pie’ instead`

Constants

- `const pi = 3.142` `// A variable `pi` that stores value 3.142`
- `pi = 'pie'` `// Uncaught TypeError: Assignment to constant variable.`

Functions and Logic

```
function myFunction(p1, p2) {
     if (p1 % 2 === 0) {
        return p1 * p2;
    } else {
        for (let i = 0; i < p2; i++) {
           p1 = p1 - 1;
         }
      return p1;
    }
} // The function returns p1 * p2 if p1 is even. If p1 is odd, returns p1 - p2.
```

Arrays and Dictionaries

Arrays

- `let array = []` `// This creates a new array`
- `array.push(item1)` `// Adds item1 to the end of the array`
- `array.unshift(item2)` `// Adds item2 to the start of the array`
- `console.log(array)` `// [item1, item2]`
- Think of array as a list of variables.

Arrays and Dictionaries

Dictionaries

- `let dict = {}` `// This creates a new dictionary`
- `dict['hi'] = 'bye'` `// Map the value of 'hi' to 'bye'`
- `dict['username'] = password`
- `console.log(dict['hi'])` `// 'bye'`
- Useful when you only what to associate a value with another value
- For example, every username of an user to his or her password

JavaScript Exercise 1

Developer Student Clubs

https://tinyurl.com/NUSReact

1. Prompt the user for a new password using `prompt()`
2. (User enters an arbitrary input)
3. Store the input into an array
4. Repeat steps 1 to 2 for 2 more times
5. If all three inputs are equal, thank the user.

   Otherwise, show any error message!

Map Functions

- Takes an array of values and apply a transformation to it.
- For example, you have an array of Person

```
class Person {
    constructor(name, age) {
        this.name = name;
        this.age = age;
    }
}
```

- And you only want an array of each of the person's name
- How can you do this?

Map Functions

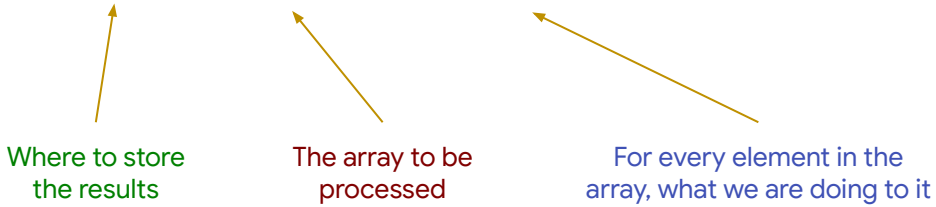
- Perfect time to use Map :')
- What we have

```
let people = [new Person('name1', 20), new Person('name2', 30), ...];
```

- What we want
- `let names = ['name1', 'name2', ...];`
- What we should do here
- `let names = people.map(x => x.name);`

Map Functions

- Breaking down the syntax
- `let names = people.map(x => x.name);`



- Imagine you have a basket of apples, and another empty basket. You want to slice the apples before putting it in the empty basket.

Different Syntax Same Meaning

```
let names = people.map(x => x.name);


let names = people.map(function (x) {
     return x.name;
});
```

Filter Functions

- Takes an array of values and filter out those that **fails a test**
- For example, I have a basket of apples again, I want to filter out the bad apples.
- Essentially, **our test** which is our **filter** is to:

  “only keep apples that are good”
- After translating into code, it can look like this:

```
(apple) => apple.condition == condition.good
```

- The filter function is similar to that of map

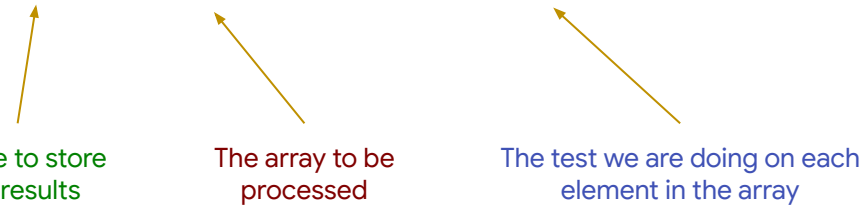
Filter Functions

- After translating into code, the test can look like this:

```
(apple) => apple.condition == condition.good
```

- Syntax for Filter: `[Array].filter([test])`
- If the basket of apples is an array named `apples`

```
goodapples = apples.filter(apple => apple.condition == condition.good)
```



Note that only elements that **pass** the test are **kept**.

JavaScript Exercise 2

Developer Student Clubs

https://tinyurl.com/NUSReact

1. You have a list of Persons.
2. You want to collect personal information.
3. Prohibited by law to collect personal information of those below 13 years old.
4. Filter out the people below 13 years old, and show the name of the parents of those remain.

You are given

```
class Person {
    constructor(name, age, parentName) {
        this.name = name;
        this.age = age;
        this.parentName = parentName;
    }
}

let people = [];
people.push(new Person('Kyler', 18, 'Bob'));
people.push(new Person('Melodies', 12, 'Bob'));
people.push(new Person('Evan', 15, 'Bob'));
people.push(new Person('Andre', 50, 'Bob'));
people.push(new Person('Enhao', 29, 'Bob'));
people.push(new Person('Junhup', 54, 'Bob'));
```

JavaScript: Section Summary

- Variables and Constants
- Conditionals
- Arrays and Dictionaries
- Map and Filter

JavaScript: Additional Materials

- To hone strengthen your JavaScript fundamentals
  - 10 Days of JavaScript by HackerRank
- To master JavaScript
  - JavaScript Reference by MDN

React

What is React?

- React is a JavaScript Library used for website building
- It is created by Facebook
- Written in JSX
- Uses Component
- Able to manage States

Benefits of React

- Uses a Virtual DOM (Document Object Model)
  - Virtual DOM
    - Able to update a specific portion of the webpage when needed
    - Much Faster
  - Real DOM
    - Requires to refresh the entire DOM
    - Much Slower
- Codes Reusable
  - Components (E.g. Checkbox) can be reuse to increase productivity
- React Native
  - Able to use similar codes for its Mobile App counterpart

One Way DataFlow

Project

- An web application that display different temperatures across Singapore
- Includes Information:
  - Place
  - Temperature
  - Image (If Temp > 28, show Sun, else show Cloud)
- Getting the dataset through the API from https://data.gov.sg/dataset/realtime-weather-readings
- Sample
  - https://enhao25.github.io/weatherapp/

Hello World

Starting a React Project

- Open terminal (mac) / command prompt (search for cmd) / git bash
- Create a new react project
  - npx create-react-app <name of application>
  - E.g. npx create-react-app weatherapp
- Change Directory to the newly created folder
  - cd weatherapp
- Start the project
  - npm start
- Install other libraries as and when needed
  - npm install tachyons
  - Insert this line to index.js -> import “tachyons”;
  - https://tachyons.io

<h1>Hello World</h1>

- Open the newly created file (weatherapp) via your IDE (Editor - E.g. VSCode)
- The main script file is the index.js located in “weatherapp/src/index.js”
  - Include hello world in index.js
  - ReactDom.render(<h1>Hello World</h1>, document.getElementbyId(‘root’))

Standard Function Declaration (Function)

```
import React from 'react'; // To run JSX, must always import React

function <name of function>() {
    return(
        <div> // There must only be 1 main element for us to return
            Body Content
        </div>
     )
}
export default <name>
```

Another way to declare Function (Arrow Function)

```
import React from 'react'; // To run JSX, must always import React

const <name> = () => {
    return(
        <div> // There must only be 1 main element for us to return
            Body Content
        </div>
     )
}
export default <name>
```

Another way of declaration (ES6 Class)

```
import React, { Component } from 'react'; // To run JSX, must always import React

class <name> extends Component {
    render(){
        return(
            <div> // There must only be 1 main element for us to return
                // Body Content
            </div>
        )
    }
}
export default <name>
```

Hello World as a Component

- Create a new file “Hello.js”
- Change <App /> to <Hello />
- Body:

```
<div className='f1 tc’>
        <h1>Hello World</h1>
        <p>Welcome to React</p>
    </div>
```

Properties (Props)

- Passing values to be passed from one components to another
- Top: `greeting ={“Hello World”}`
- Bottom: `{this.props.greeting}` (Class) / `{props.greeting}` (Function / Arrow Function)

Weatherapp 1 (Stateless)

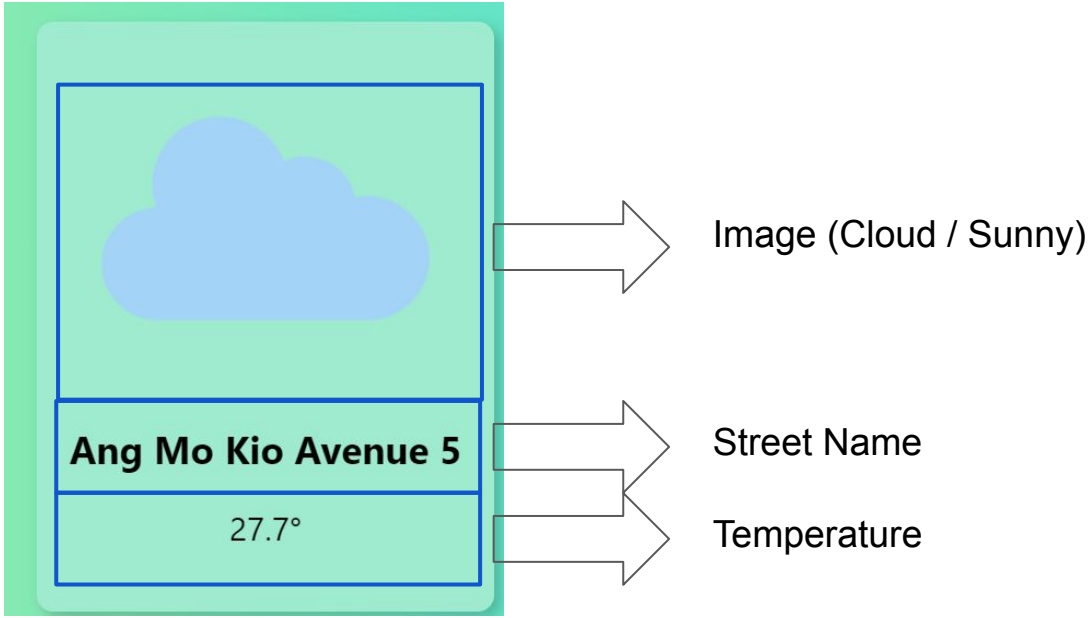
Development Planning

- What Libraries should we include? (E.g. Tachyons /Bootstrap)
- What are the Components needed?
  - Index (Main)
  - SearchBox
  - CardList
  - Card

Card Component

Card Component

https://drive.google.com/open?id=1U207BRo7S05LlIByY3pDaNv4KxN-qmRz



Card Component

```
import React from 'react';
import cloud from './assets/cloud.png'

const Card = () => {
    return (
        <div className="tc bg-light-green dib br3 pa3 ma2 grow">
            <img src={cloud}/>
            <div>
                <h2><name></h2>
                <p><temperature></p>
            </div>
        </div>
    )
}
export default Card;
```

Using Multiple Cards with Datas

- Create a weather.js file
- Copy and paste the CardList.json file in the google drive
- Using the Array of values:
  - import {weather} from "./weather";
  - Input the values of the database as props:
    - E.g. `<Card id={weather[0].id} name={weather[0].name} value={weather[0].value} />`
  - Use the props in Card.js
    - E.g. `{props.name}` / `{props.value}`

If statement with Card Component

How to write if / conditional state:

- {condition ? true : false }

Card Component:

```
{props.value < 28
            ? <img src={cloud} height="200" width="200" alt="Cloud Pic"/>
            : <img src={sun} height="200" width="200" alt="Sun Pic"/>}
```

CardList Component

```
const cardsArray = weather.map((user, i) => {
        return <Card key={i} id={weather[i].id} name={weather[i].name}
value={weather[i].value} />
    })
```

App Component (Parent Component)

```
import React from 'react';
import CardList from "./CardList";
import SearchBox from './SearchBox';

const App = () => {
    return (
        <div className="tc">
            <h1>Weather App</h1>
            <SearchBox />
            <CardList />
        </div>
    )
}

export default App;
```

SearchBox Component

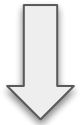
```
<div className="pa2">
            <input
                className="pa3 ba b--green bg-lightest-blue"
                type="search"
                placeholder="Search Place" />
</div>
```

Break

WeatherApp Part 2 (Stateful)

How does the searchbox component knows how to filter the card lists by name?

By making the app “smart”



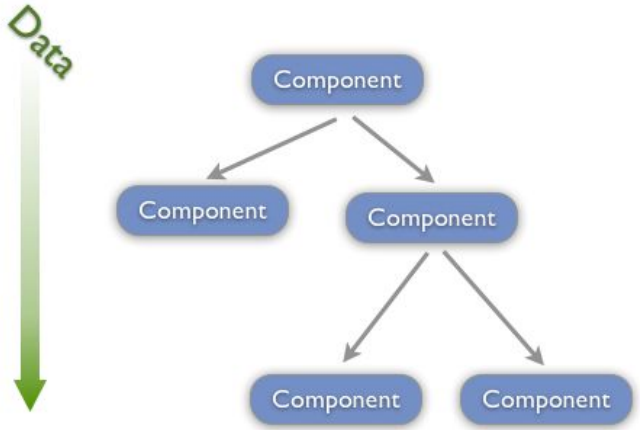
This is done through converting app.js component from stateless to stateful

State

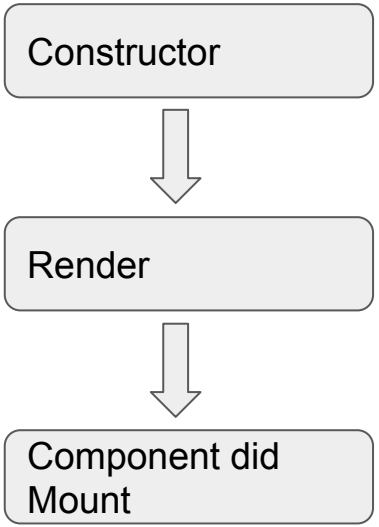
- State represents the data stored in a component
- The data is stored in the object of this.state
- This data can be changed via method call: setState()
- This data can be passed on to stateless component as props

State

- One way data flow
  - This means that data should only be known to itself and components that it renders



Lifecycle of a stateful component



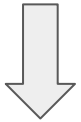
- constructor()
  - Initialize this.state
  - Initialize props received
- render()
  - Displays components
  - Will re-render if there is a change in state
- componentDidMount()
  - Called immediately after component is rendered
  - Where asynchronous calls are made

Class component declaration

```
import React, { Component } from 'react'; // To run JSX, must always import React

class <name> extends Component {
    render(){
        return(
            <div> // There must only be 1 main element for us to return
                // Body Content
            </div>
        )
    }
}
export default <name>
```

Now we want to enter names in the search box and filter the weather



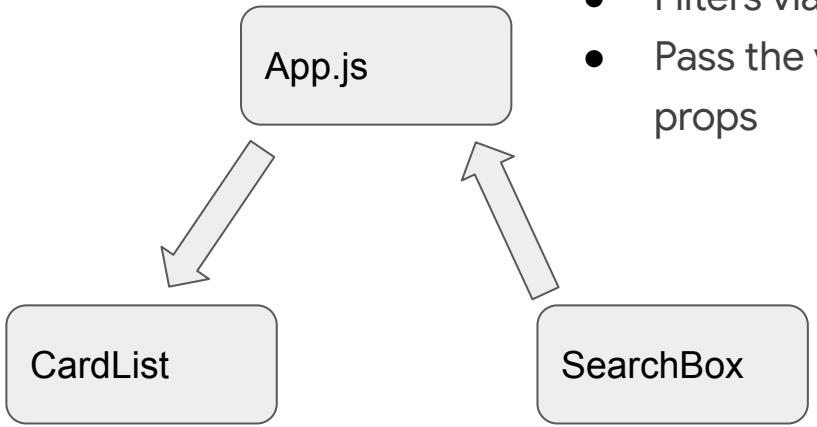
Event handling!

Event handling

- Change the state of the search value
- This will re-render the card lists to display cards with names of the following search value
- The value is usually retrived using event.target.value
- For this project we are going to use onChange event
- https://reactjs.org/docs/events.html

Stateful app component

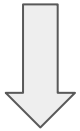
- Stores the data on the searchbox
- Filters via the search value
- Pass the value down to the CardList as props



- onSearchChange changes the value of the data stored in App.js

- Received data of weather filtered by the search value

Now we want real data



We can do this by fetching 3rd-party data with asynchronous api calls

Api Calls

- We use fetch() to call data from other sources
- This is done asynchronously because it takes time for ask for and receive information
- To ask for information, this is known as sending a request to a server
- To receive the information, this is known as receive a response from a server
- The data that we will be using will be https://api.data.gov.sg/v1/environment/air-temperature

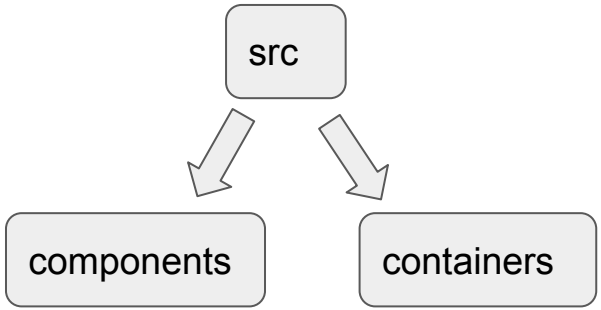
Api Calls

- The code below combines the metadata object with the items object
- We want to get the name of the location as well as the temperature of the location

```
const updatedWeather = [];
for (let i = 0; i < weather.metadata.stations.length; i++) {
        updatedWeather.push({ ...weather.metadata.stations[i], ...weather.items[0].readings[i] })
}
```

React conventions pt1

- A way in which something is usually done
- Categorising your files into components and containers folders
- Components for stateless components
- Containers for stateful components
- Better clarity for others who look at your code



React conventions pt2

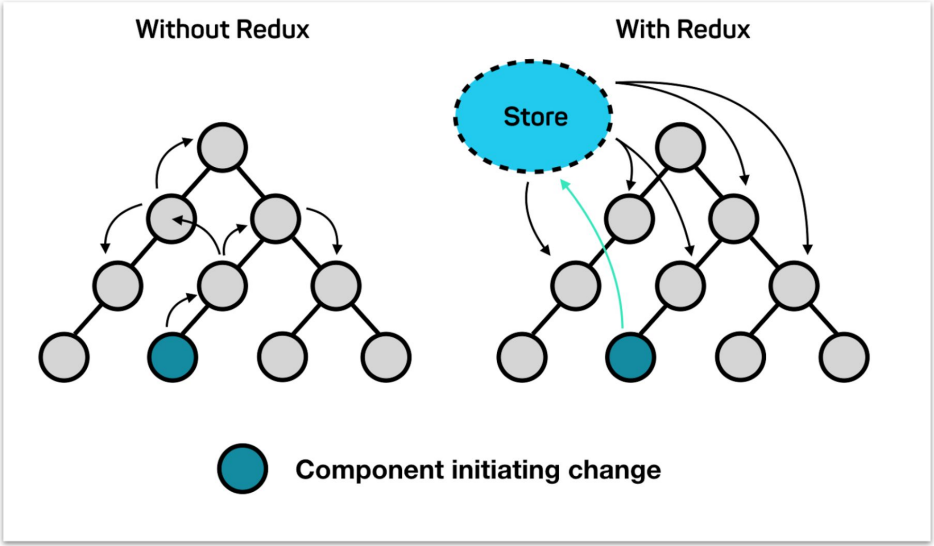
- A way in which something is usually done
- Object destructuring
- Its cumbersome to always type this.state.XXX
- Instead, write:
- const { weather, searchfield } = this.state;

For further reading

- Examples of projects done in react
  - https://reactjs.org/community/examples.html
- We covered mostly the section on main concepts
  - You can read the section on advance guide, hooks and testing
  - https://reactjs.org/docs/getting-started.html

For further reading / For advanced users

- What is there are multiple stateful components with multiple states?
- We can have a global and centralized state where components can access them
- This is done through Redux
- https://redux.js.org/



For further reading / For advanced users

- The app we did is only 1 page.
- What is I want multiple pages?
- What if I want to render a page only if user is logged in?
- We can use react-router
- https://reacttraining.com/react-router/web/guides/quick-start

https://tinyurl.com/NUSReact2

Deployment

Github Pages

Introduction

- Static site hosting service
- Project, user & organization
- <user>.github.io / <organization>.github.io

Github Pages

Benefits

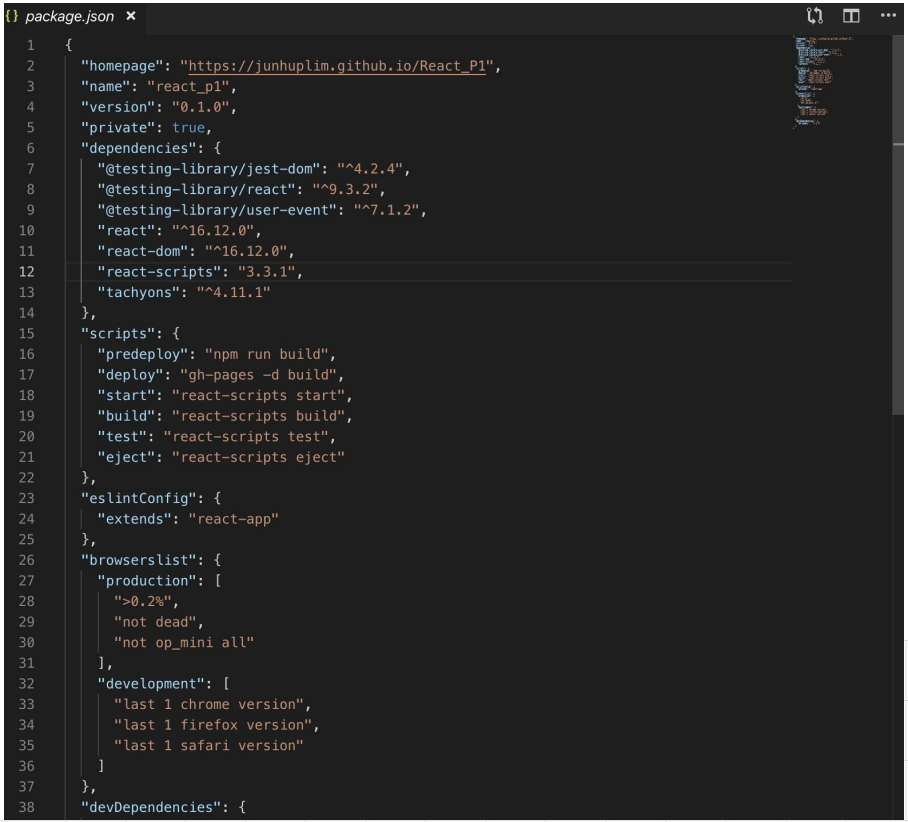
- Easy setup
- Seamless collaboration using Git and Github
- Free hosting with >95% uptime
- Live updating with normal Github workflow

Deployment to Github Pages

1) In the package.json file
   a) Add line 2 above “name”
   b) Add lines 16 & 17 under “scripts”

*line 2, change to your name & your repo name.

“https://{username}.github.io/{repo_name}”



Deployment to Github Pages

1. Go to https://github.com/
2. Create a new repository
3. Upload project files into the git repository using drag & drop
    a. Git init
    b. Git remote add origin <Insert github repo link>
    c. Npm run deploy
4. Your website is now published!

Summary

Summary

- React is an open-source JavaScript library for building user interfaces, built around components.
- Stateless (dumb)
  - Data passed down by parents via props
- Stateful (smart)
  - Keeps track of changing data
- One of the most popular libraries especially for JavaScript developers
- Web hosting services allows for free website deployment with limitations (E.g. Github Pages / Netlify / Heroku)

Developer Student Clubs
National University of Singapore

Please helps us to do the feedback

Thank You!

https://tinyurl.com/w3ufxeg

http://t.me/dscnus

https://www.linkedin.com/company/dscnus/

@dscnus