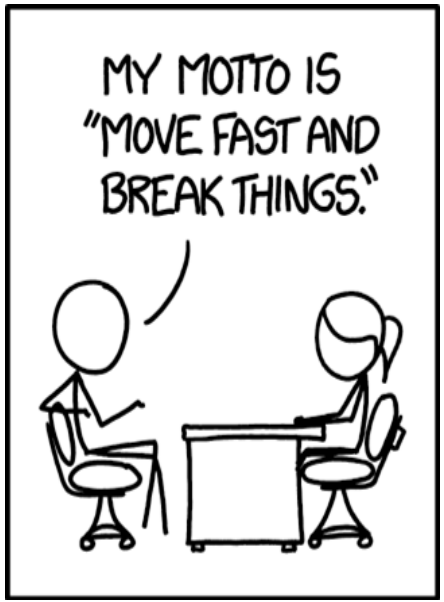
MISSION READY

DARE TO
DEVELOP

Introduction to React

Ewan Zhang



JOBS I'VE BEEN FIRED FROM

FEDEX DRIVER
CRANE OPERATOR
SURGEON
AIR TRAFFIC CONTROLLER
PHARMACIST
MUSEUM CURATOR
WAITER
DOG WALKER
OIL TANKER CAPTAIN
VIOLINIST
MARS ROVER DRIVER
MASSAGE THERAPIST

https://xkcd.com/1428/

What is React?

React

A JavaScript library for building user interfaces

https://legacy.reactjs.org/

What is library?

Libraries are typica… …egrated into larger software projects.

They contain mo… …s, or other reusable components tha… …onality.

Developers can i… …d… …ibrary into their codebase and utilize its featu… …hance their applications.

Node.js and React.js are the two most common web technologies used by Professional Developers and those learning to code.

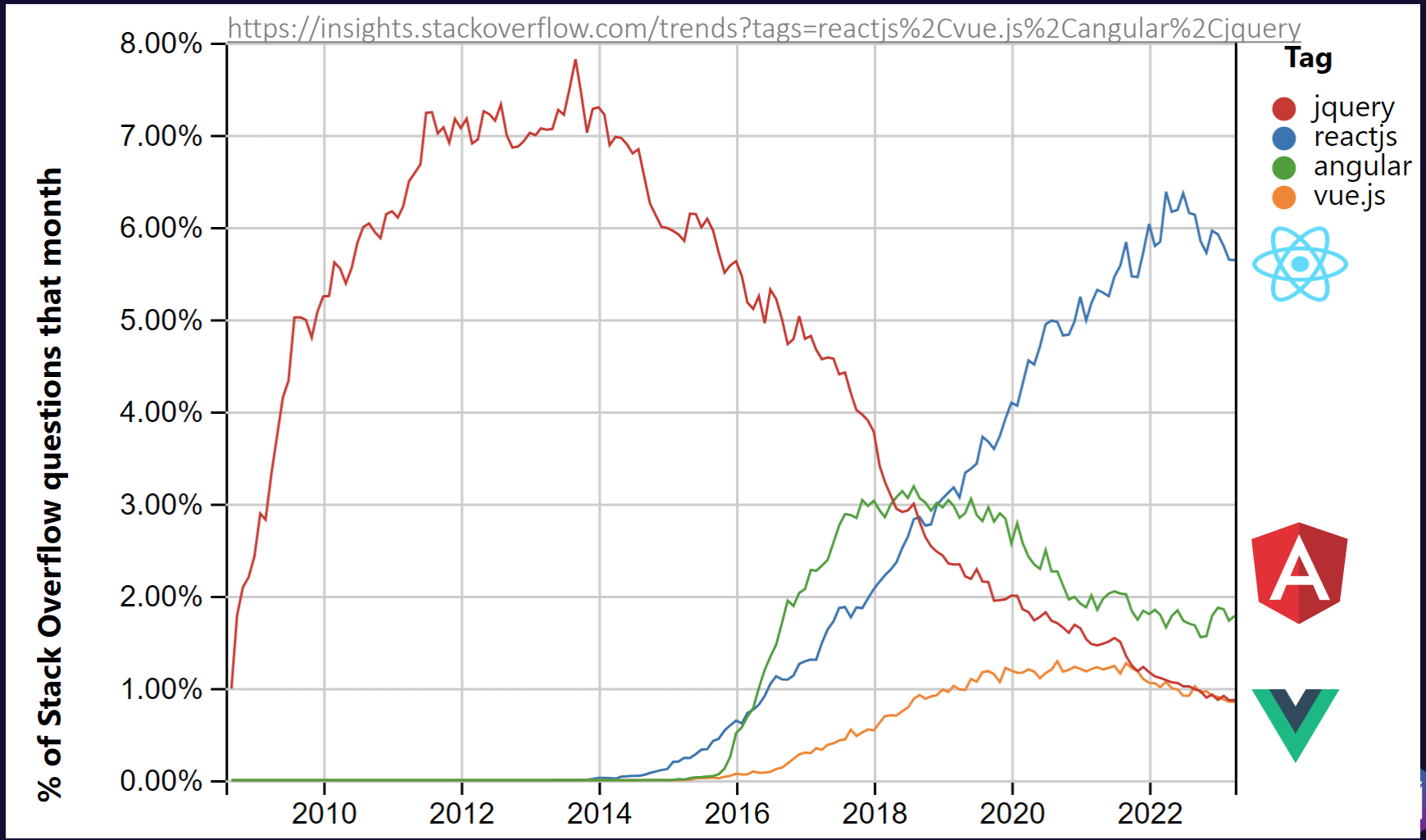
https://survey.stackoverflow.co/2022/#most-popular-technologies-webframe

Modern JavaScript libraries/frameworks

FaceBook 2013

Google Team 2016

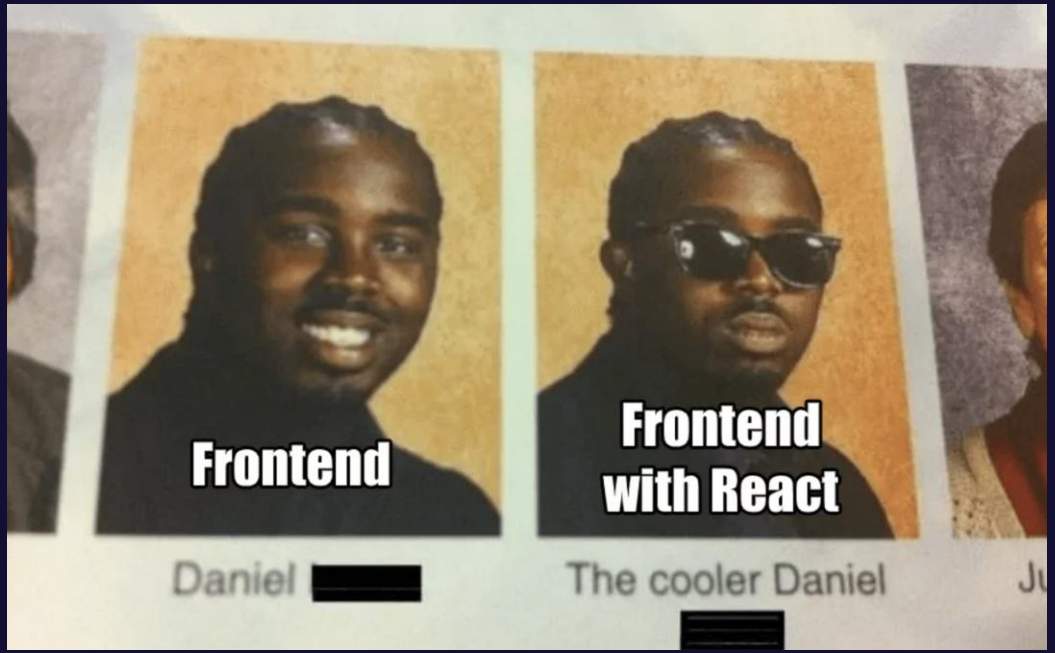
Evan You 2014


https://insights.stackoverflow.com/trends?tags=reactjs%2Cvue.js%2Cangular%2Cjquery
% of Stack Overflow questions that month
8.00%
7.00%
6.00%
5.00%
4.00%
3.00%
2.00%
1.00%
0.00%
2010
2012
2014
2016
2018
2020
2022
Tag
jquery
reactjs
angular
vue.js

Advantages of using a framework

- Reusable UI components.
  - Easier to build and maintain a complex website.
- Declarative paradigm - the developer describes what the UI should be like.
  - Better structure and maintainable programs
- Better tools and testing capability
- Higher performance with lesser effort

Do you know any website build by React?



Who uses React?

COMPANIES

7829 companies reportedly use **React** in their tech stacks, including **Airbnb**, **Uber**, and **Facebook**.


Airbnb

Uber


Facebook

Netflix


Instagram


medium.com


Pinterest

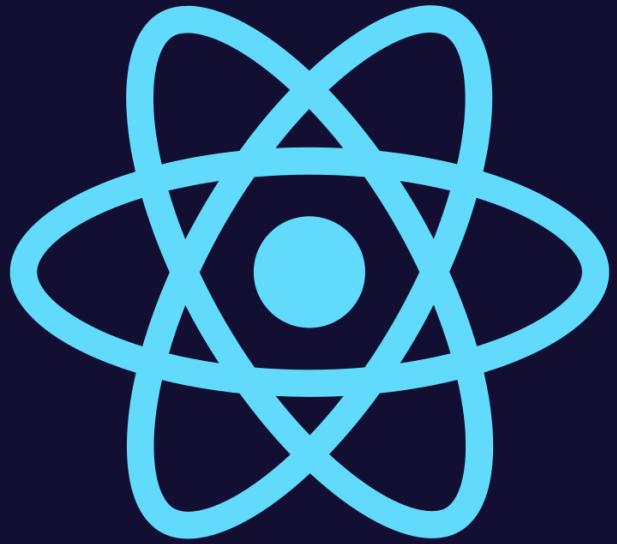

Twitter


reddit

Is react easy to learn?

What is React ?

- React is a declarative, **efficient**, and **flexible** JavaScript library for building user interfaces.
- It lets you compose ***complex UIs*** from small and isolated pieces of code called “components”.
- Created by Jordan Walke, a Software Engineer at Facebook
  - Open-sourced at JSConf 2013.



https://reactjs.org/tutorial/tutorial.html#what-is-react

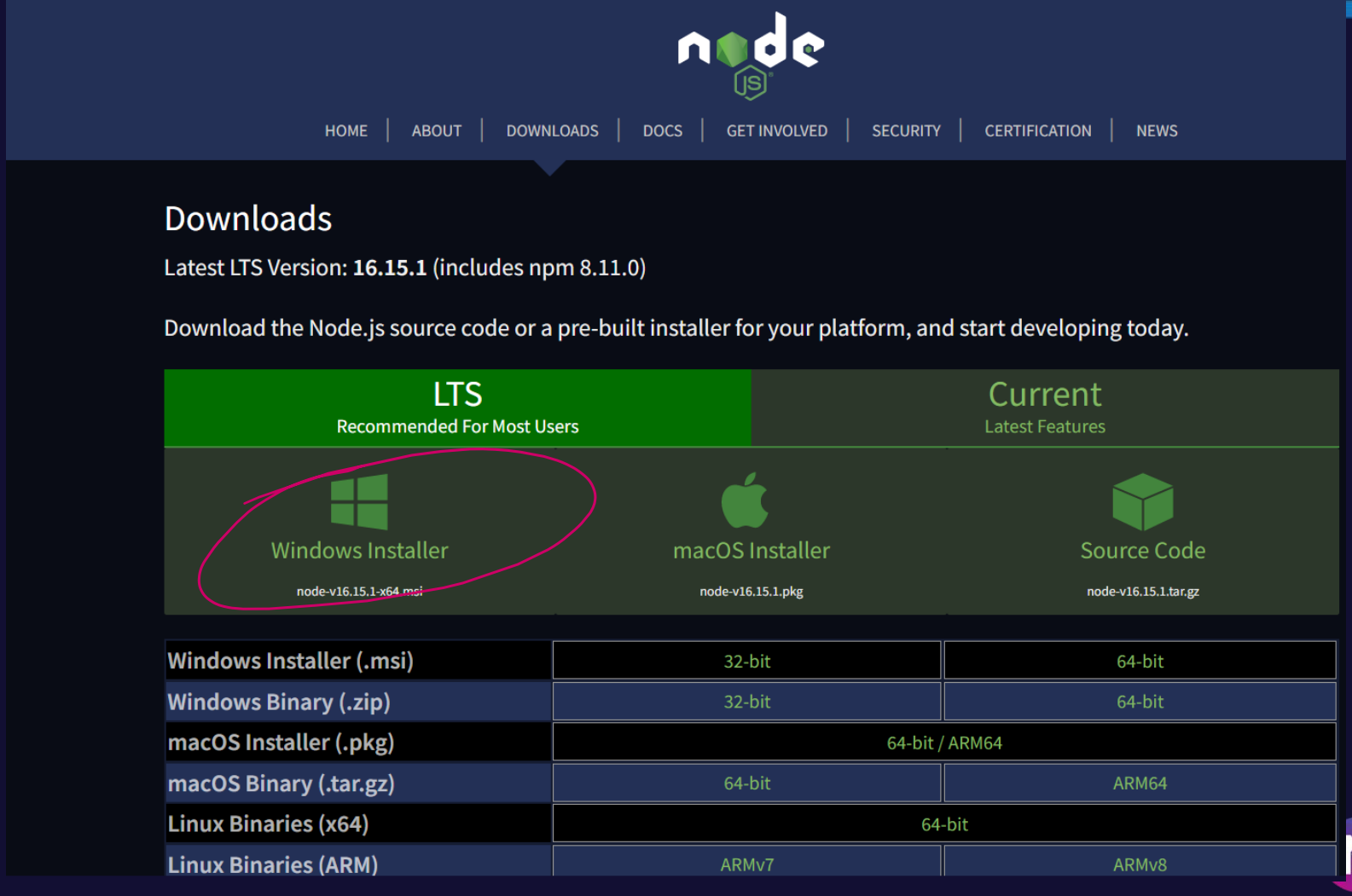
Some important React concepts

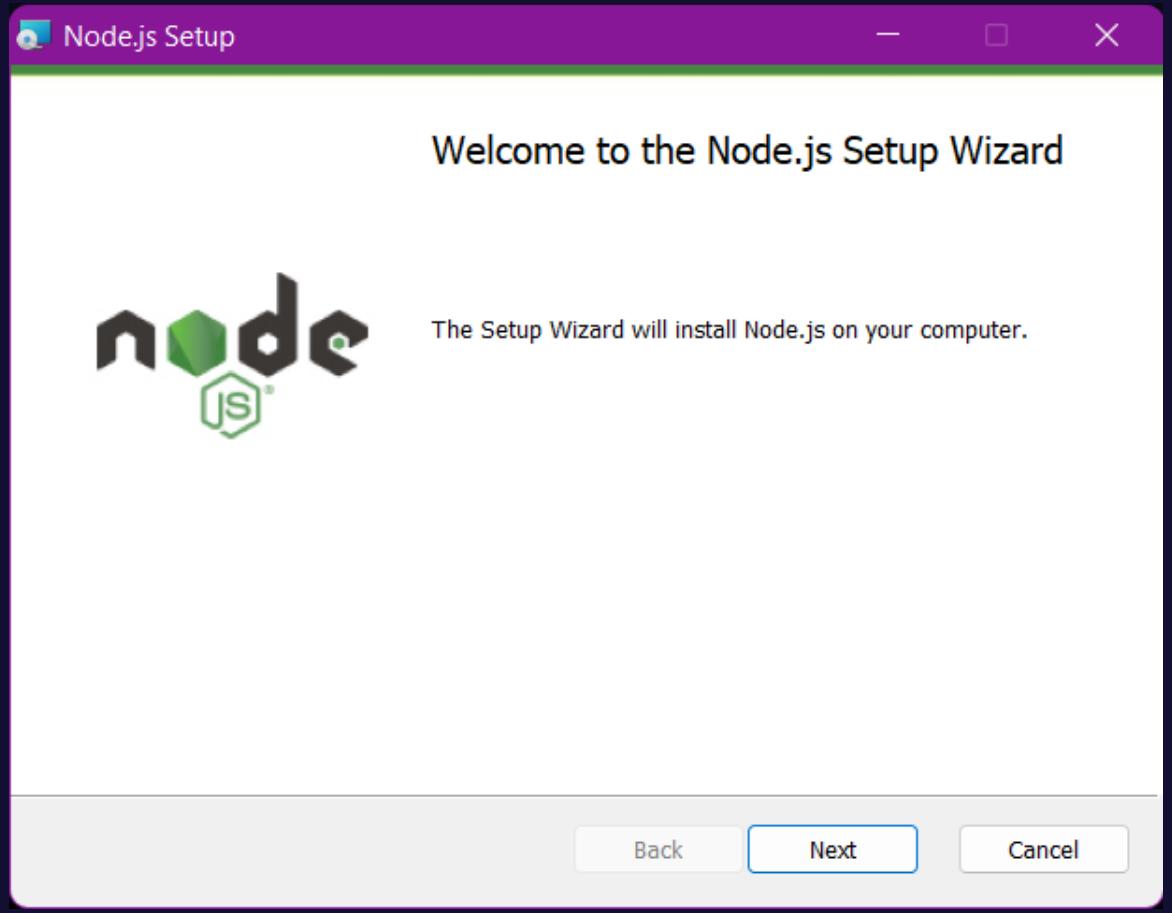
- Rendering elements
- Components and their types
- Component tree and decomposability
- Life cycle events
- JSX
- Props, Event
- Mutability and Immutability
- State and updating state
- Functions vs Classes for components
- Hooks

Environment Setup

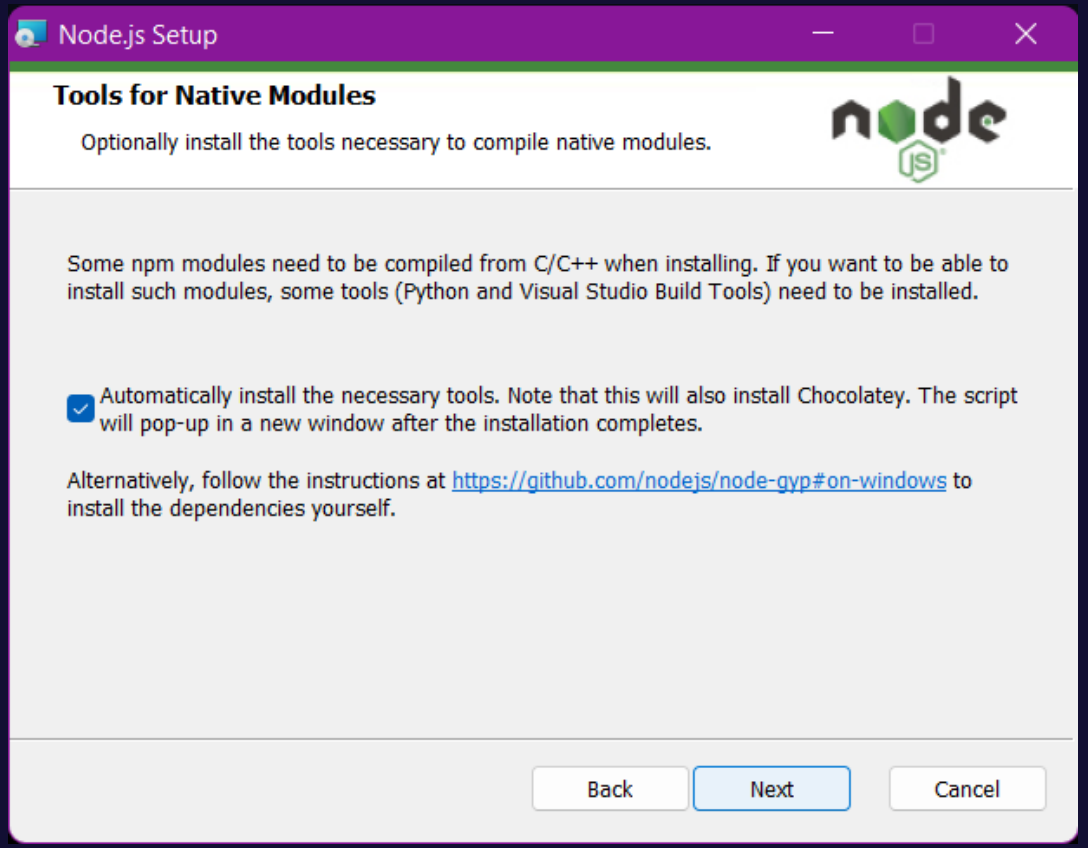
- Node.js
  - npm - Node Package Manager, an online repository of packages + a command line utility
  - npx - A node tool execute npm packages

https://nodejs.org/en/download/

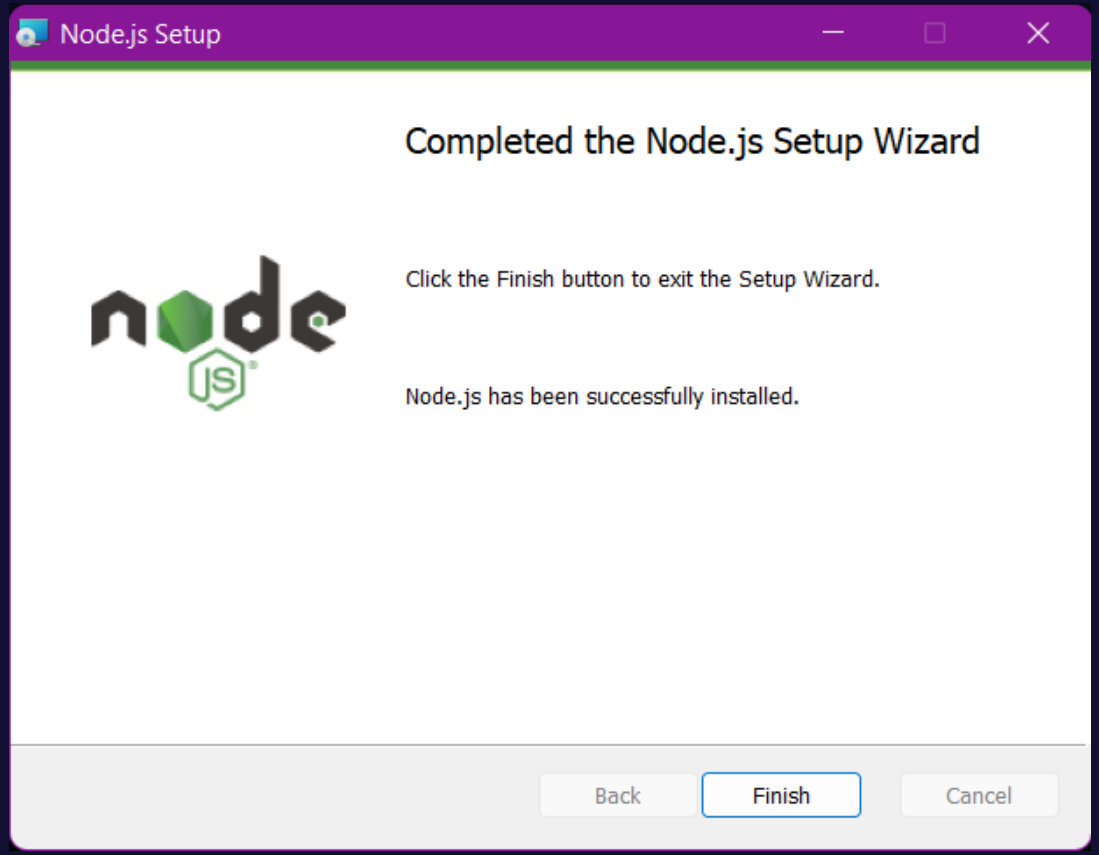

HOME
ABOUT
DOWNLOADS
DOCS
GET INVOLVED
SECURITY
CERTIFICATION
NEWS
Downloads
Latest LTS Version: 16.15.1 (includes npm 8.11.0)
Download the Node.js source code or a pre-built installer for your platform, and start developing today.
LTS
Recommended For Most Users
Current
Latest Features
Windows Installer
node-v16.15.1-x64.msi
macOS Installer
node-v16.15.1.pkg
Source Code
node-v16.15.1.tar.gz
Windows Installer (.msi) 32-bit 64-bit
Windows Binary (.zip) 32-bit 64-bit
macOS Installer (.pkg) 64-bit / ARM64
macOS Binary (.tar.gz) 64-bit ARM64
Linux Binaries (x64) 64-bit
Linux Binaries (ARM) ARMv7 ARMv8

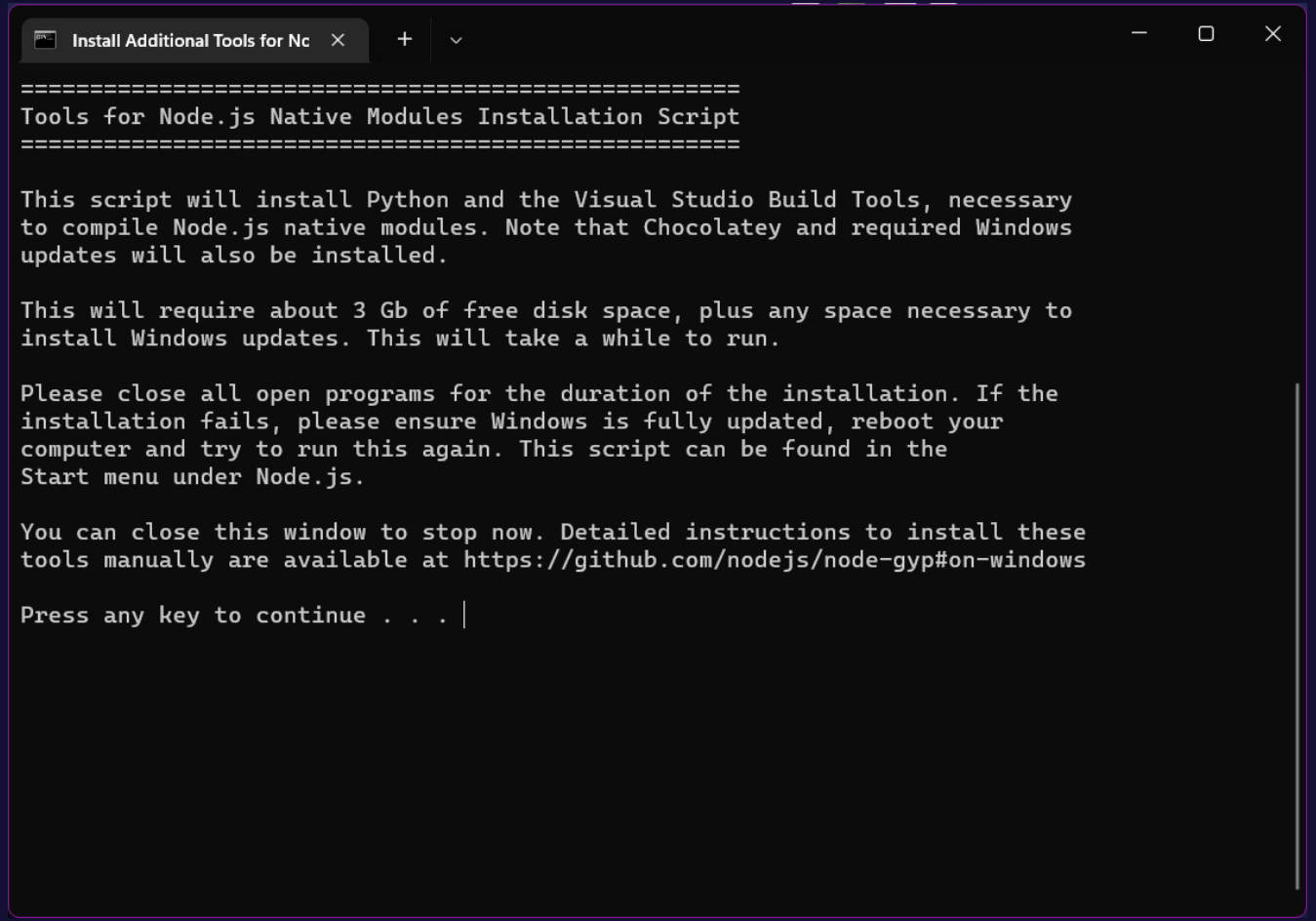


- Keep the default settings



- Tick the checkbox.


Node.js Setup
Completed the Node.js Setup Wizard
Click the Finish button to exit the Setup Wizard.
Node.js has been successfully installed.
Back
Finish
Cancel



- Press any key


+
npm


Flooring
Storage
We Guarantee the Lowest prices everyday. If you find a lower Price on a stocked item, we will beat it by 10%.
LOWEST PRICES
Special Orders Desk has an even wider range of products and information for you to choose from.
SPECIAL ORDERS
Garden
SELF CHECKOUT
Point, Kitchen & Bathroom, Cleaning, Window Coverings, Flooring & Electrical

```
node -v
    18.15.0
npm -v
    9.5.0
```

- Restart VS Code
- Verify node installation by running these commands in the VS Code terminal

Environment Setup (optional)

- React Dev Tools
  - Allows you to inspect the React component hierarchies in the
  - Installation
    - Firefox - https://addons.mozilla.org/en-US/firefox/addon/react-devtools/
    - Chrome - https://chrome.google.com/webstore/detail/react-developer-tools/fmkadmapgofadopljbjfkapdkoienihi?hl=en

Let’s get started – Exercise 1

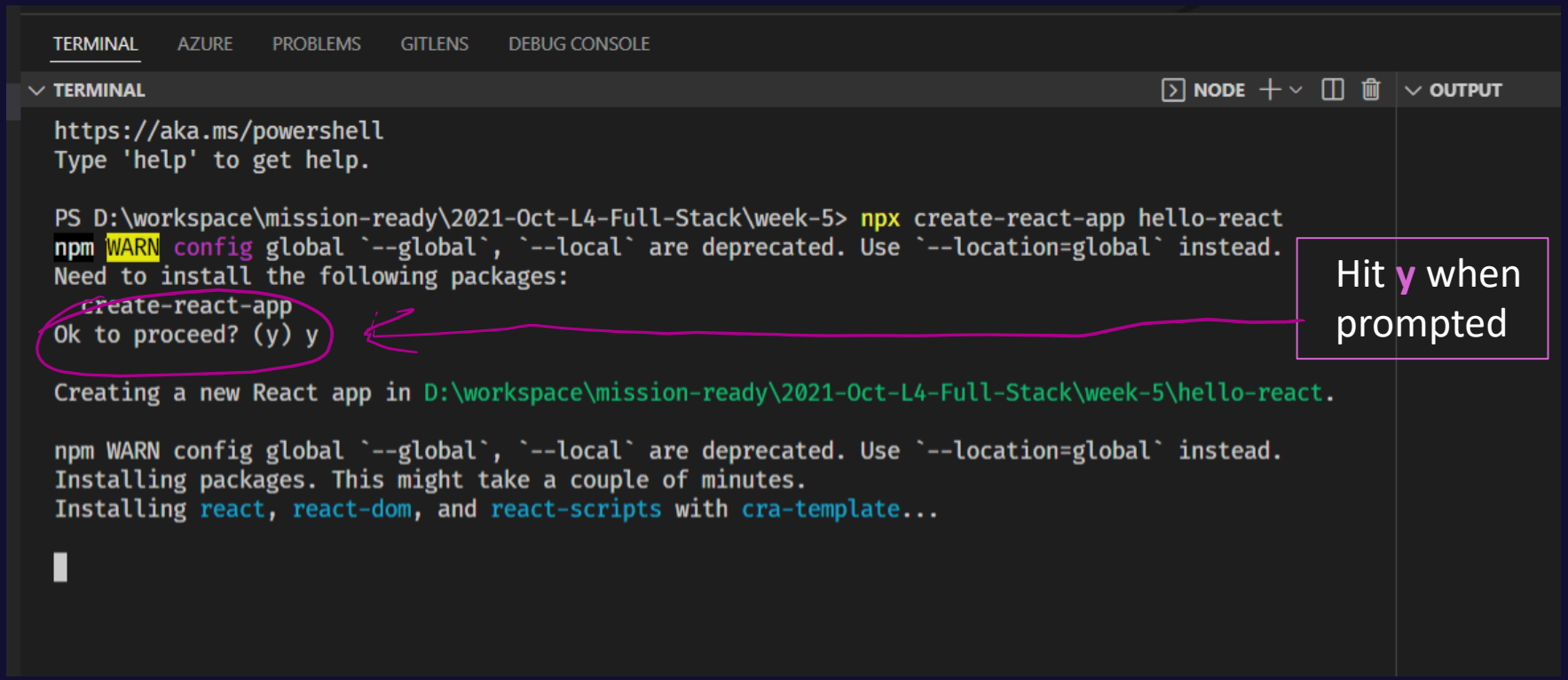
1. Create a new folder for week-5. Move into the folder in the terminal

```
cd week-5
```

2. Run the following commands to create a hello world in react.

```
npx create-react-app hello-react
cd hello-react
npm start
```

3. Share a screenshot when you’ve got the browser up.


TERMINAL
AZURE
PROBLEMS
GITLENS
DEBUG CONSOLE
TERMINAL
NODE
OUTPUT
https://aka.ms/powershell
Type 'help' to get help.
PS D:\workspace\mission-ready\2021-Oct-L4-Full-Stack\week-5> npx create-react-app hello-react
npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.
Need to install the following packages:
create-react-app
Ok to proceed? (y) y
Creating a new React app in D:\workspace\mission-ready\2021-Oct-L4-Full-Stack\week-5\hello-react.
npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.
Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...
Hit y when prompted

```
removed 1 package, and audited 1419 packages in 3s

186 packages are looking for funding
  run `npm fund` for details

6 high severity vulnerabilities
To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.

Success! Created hello-react at D:\workspace\mission-ready\2021-Oct-L4-Full-Stack\week-5\hello-react
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

  cd hello-react
  npm start

Happy hacking!
PS D:\workspace\mission-ready\2021-Oct-L4-Full-Stack\week-5>
```

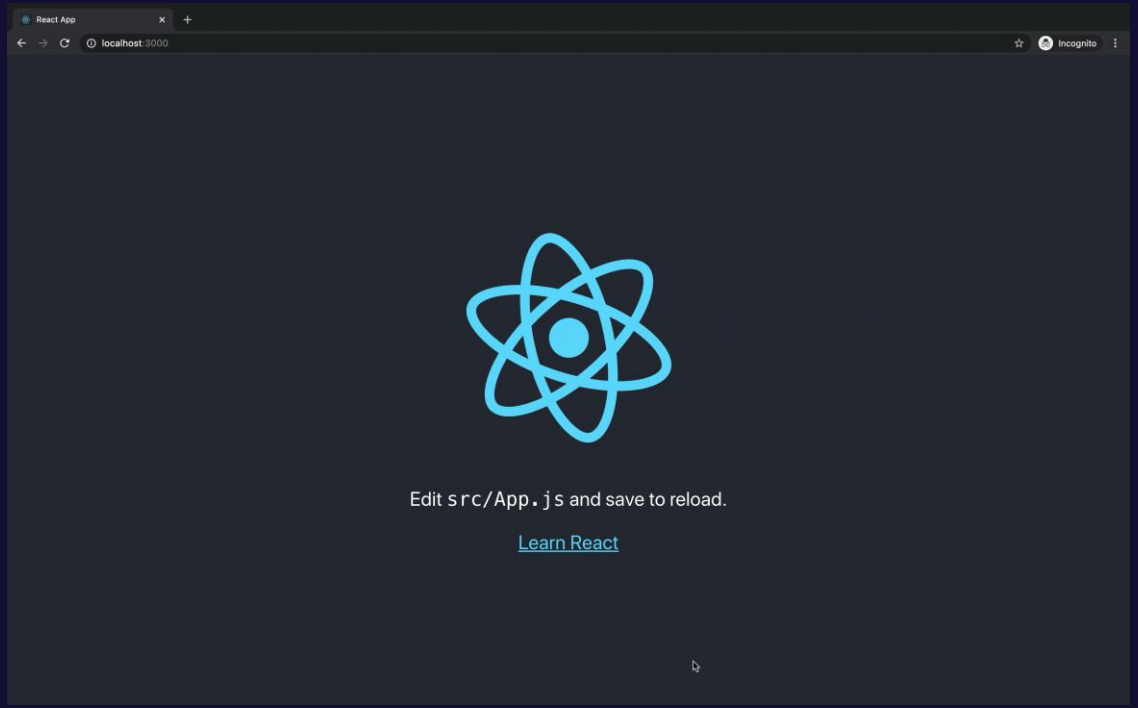
Run the commands
`cd hello-react`
`npm start`

So, what just happened

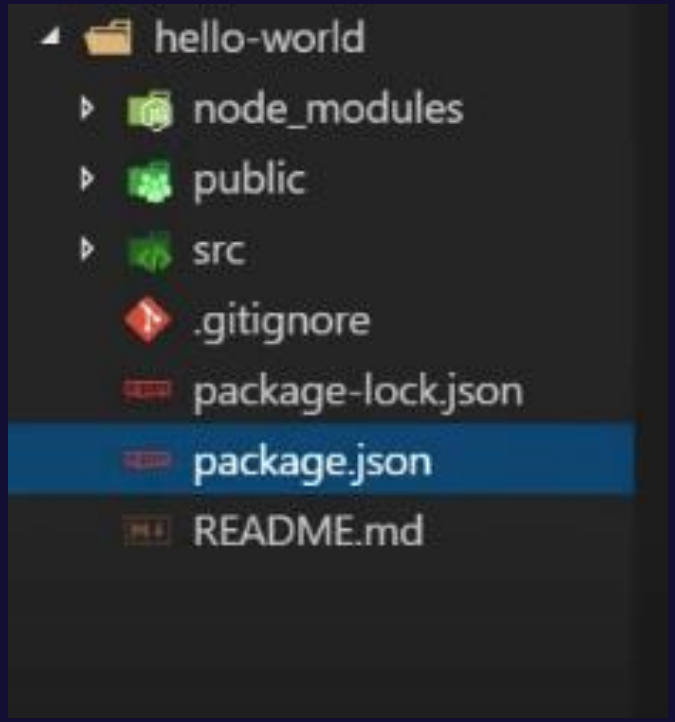
- Create React App is a comfortable environment for learning React, and is the best way to start building a new single-page application in React.
- Create React App doesn't handle backend logic or databases; it just ***creates a frontend*** build pipeline, so you can use it with any backend you want.

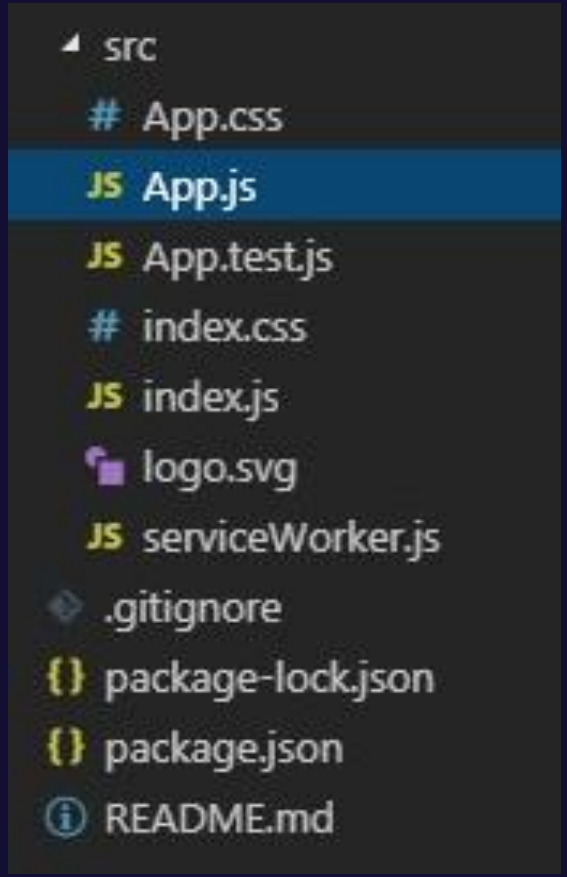
Create React App

- An officially supported way to create single-page React applications.
- A modern build setup with no configuration.



A whole lot of files !





Lets inspect those

package.json

- Specifies all packages / code to include.
- Part of npm/

index.html

- `<div id="root">`

index.js

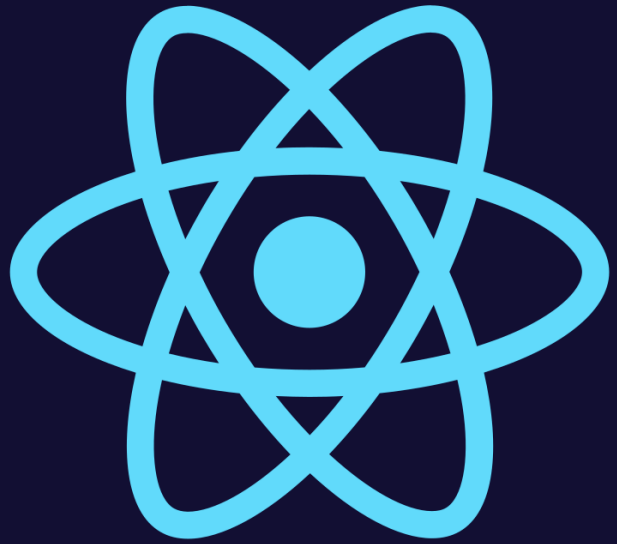
- `document.getElementById("root")`
- `<App />`

App.js

- The App component is the main component in React which acts as a container for all other components.

What is React ?

- React is a declarative, **efficient**, and **flexible** JavaScript library for building user interfaces.
- It lets you compose ***complex UIs*** from small and isolated pieces of code called “components”.
- Created by Jordan Walke, a Software Engineer at Facebook
  - Open-sourced at JSConf 2013.



https://reactjs.org/tutorial/tutorial.html#what-is-react

just HTML5

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>some title</title>
  </head>

  <body>
    <div>hello world</div>
  </body>
</html>
```

with Javascript

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>some title</title>
  </head>

  <body>
    <script>
      const helloDiv = document.createElement("div");
      helloDiv.textContent = "hello world";
      document.body.appendChild(helloDiv);
    </script>
  </body>
</html>
```

just React

```
<body>
  <script>
    const reactElement = React.createElement("div", {
      children: "Hello World",
    });
    ReactDOM.render(reactElement, document.body);
  </script>
</body>
```

JSX - a syntax extension to JavaScript

- Helps describe what the UI should look like.
- JSX is neither a string nor HTML, and after compilation produces React “elements”.
- So that we could write

```
const reactElement = <div> Hello World </div>;
```

instead of

```
const reactElement = React.createElement("div", {
    children: "Hello World",
});
```

https://babeljs.io/en/repl#

JSX produces React “elements”

JSX

- JSX has an HTML-like syntax (but NOT exactly HTML, e.g. class vs className)
- You must always return one tag wrapping all JSX code.
  - Can be any tag e.g. `<div>` or even `<>`
- JSX likes all tags to be closed `<br>` becomes `<br/>` or `<br></br>`
- Same way to comment per any JavaScript using `//` or `/* */`
- We can add JavaScript expressions using { }

https://reactjs.org/docs/fragments.html

Major differences

- To specify a CSS class, use the className attribute. This applies to all regular DOM and SVG elements like , <a>, and others.
- The style attribute accepts a JavaScript object with camelCased properties rather than a CSS string.
- All HTML/standard DOM attributes are supported as of React 16.

https://reactjs.org/docs/dom-elements.html

Expressions in JSX

```
const name = 'John Doe';
const element = <h1>Hello, {name}</h1>;
```

- Some JavaScript expression categories include:
  - Arithmetic: evaluates to a number, for example 3.14159.
  - String: evaluates to a character string, for example, "Fred" or "234".
  - Logical: evaluates to true or false.
  - Primary expressions: Basic keywords and general expressions in JavaScript.
  - Left-hand-side expressions: Left values are the destination of an assignment.

Exercise 2

1. From exercise – 1 , delete the following files,
   - App.css
   - App.test.js
   - Logo.svg
2. Remove the two imports in the App.js
3. Add two const variables with two string values before the App function in **App.js**

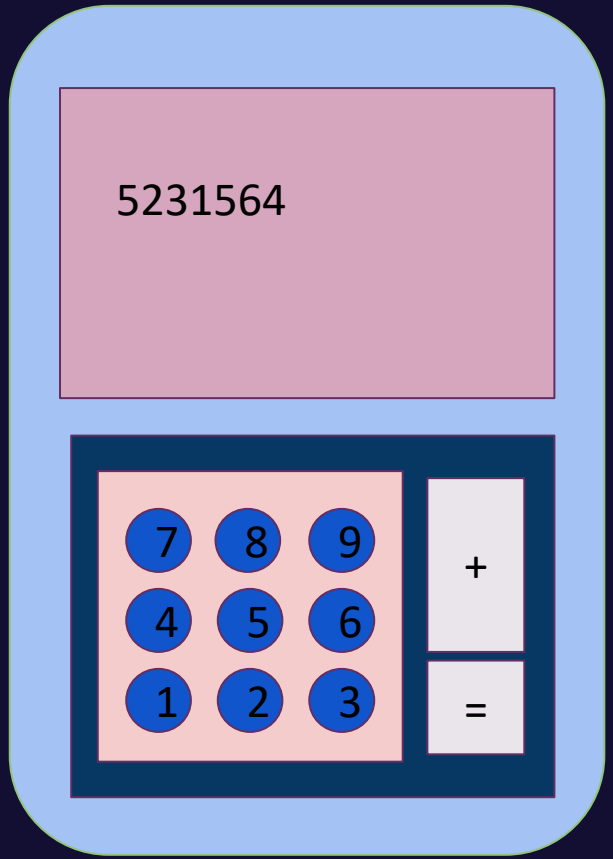
```
const message1 = "Hey you";
const message2 = "What's going on";
```

4. Remove the contents of the return statement and add the following.

```
<>
 <h1>{message1}</h1>
 <p>{message2 + "??"}</p>
</>
```

Components

- Components let you **split** the **UI** into independent, reusable pieces, and think about each piece in isolation.
- Components can refer to other components in their output. This lets us use the same component abstraction for any level of detail



x

7 8 9
4 5 6
1 2 3

---

7 8 9
4 5 6
1 2 3

+

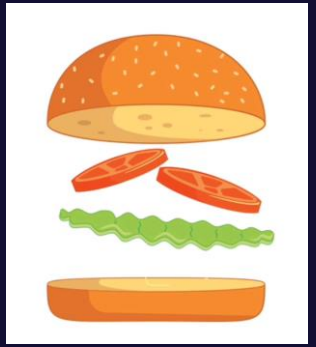
=

7 8 9
4 5 6
1 2 3

+

=

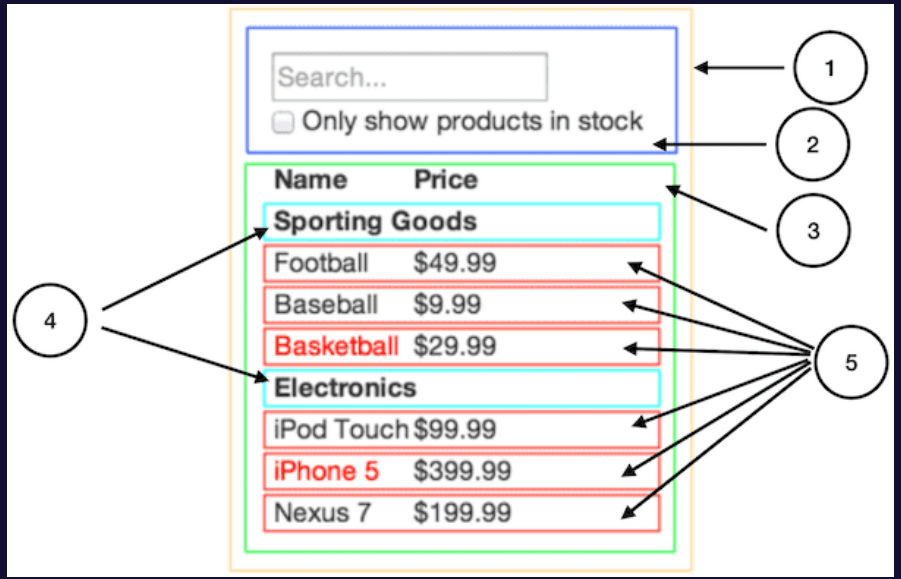
Components



Why do we need components?

Components

- Components are independent and reusable bits of code.
  - They serve the same purpose as JavaScript functions but work in isolation and return HTML.
- They accept arbitrary inputs (called “props”) and return React elements ***describing what should appear on the screen***.



```
function Header() {
  return (
    <div>
      <h1>Hello World</h1>
    </div>
  );
}

export default Header;
```

===

```
export default function Header() {
  return (
    <div>
      <h1>Hello World</h1>
    </div>
  );
}
```

Exercise 3 – Let's create a component

1. Create a Footer.js file inside the src folder
2. Add the following contents

```
function Footer() {
  return <div>This is a footer</div>;
}
export default Footer;
```

3. Import it into the App.js file, then add the component inside the return statement

```
import Footer from "./Footer";

function App() {
  return (
    <>
      <p>This is some HTML before the footer</p>
      <Footer></Footer>
    </>
  );
}
export default App;
```

4. Run `npm start` to check your changes

Components in React

Three main ways for expressing components in React:

1. Class components
2. ***Functional components***
3. Functional components with Hooks

Functional Component

```
function Welcome(props) {
  return <h1>Hello, {props.name}</h1>;
}
```

Class Component

```
class Welcome extends React.Component {
  render() {
    return <h1>Hello, {this.props.name}</h1>;
  }
}
```

Fun learning react

- Weather apps
- Calculators
- To do apps
- https://github.com/Asabeneh/30-Days-Of-React

MISSION READY

www.missionreadyhq.com

DARE TO DEVELOP

Thank you

Ewan Zhang