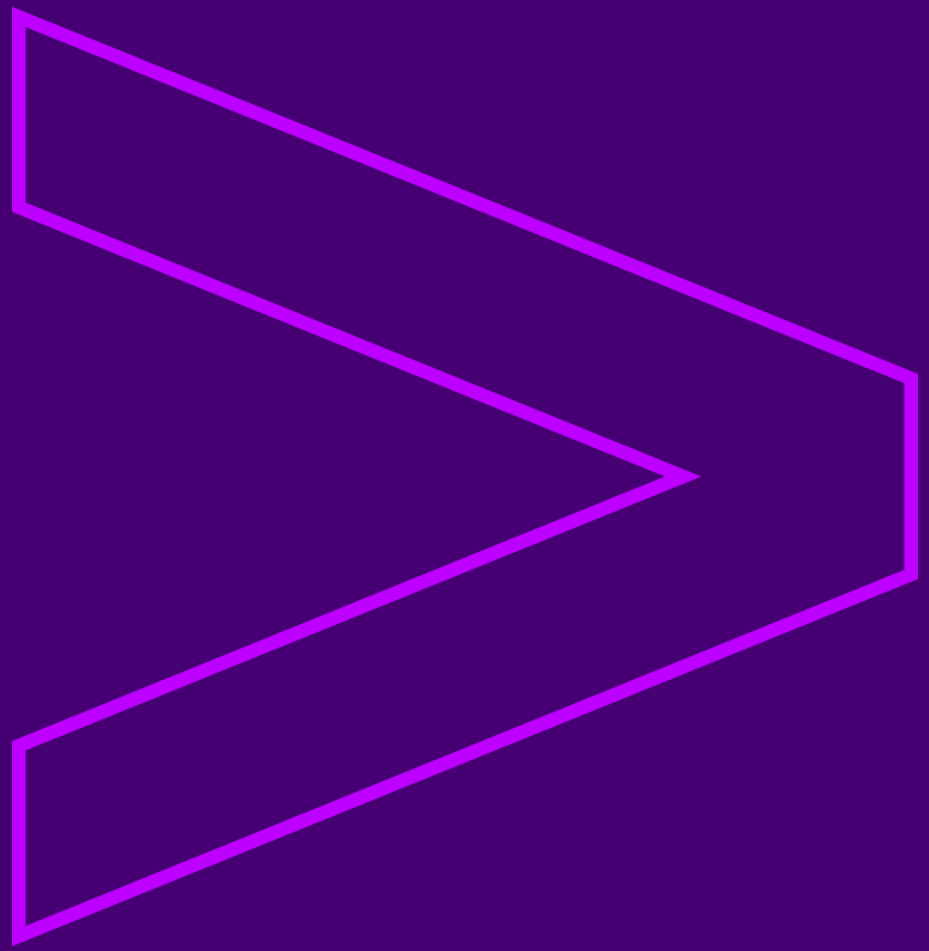
MODULE 1 :
REACTJS
INTRODUCTION TO REACT

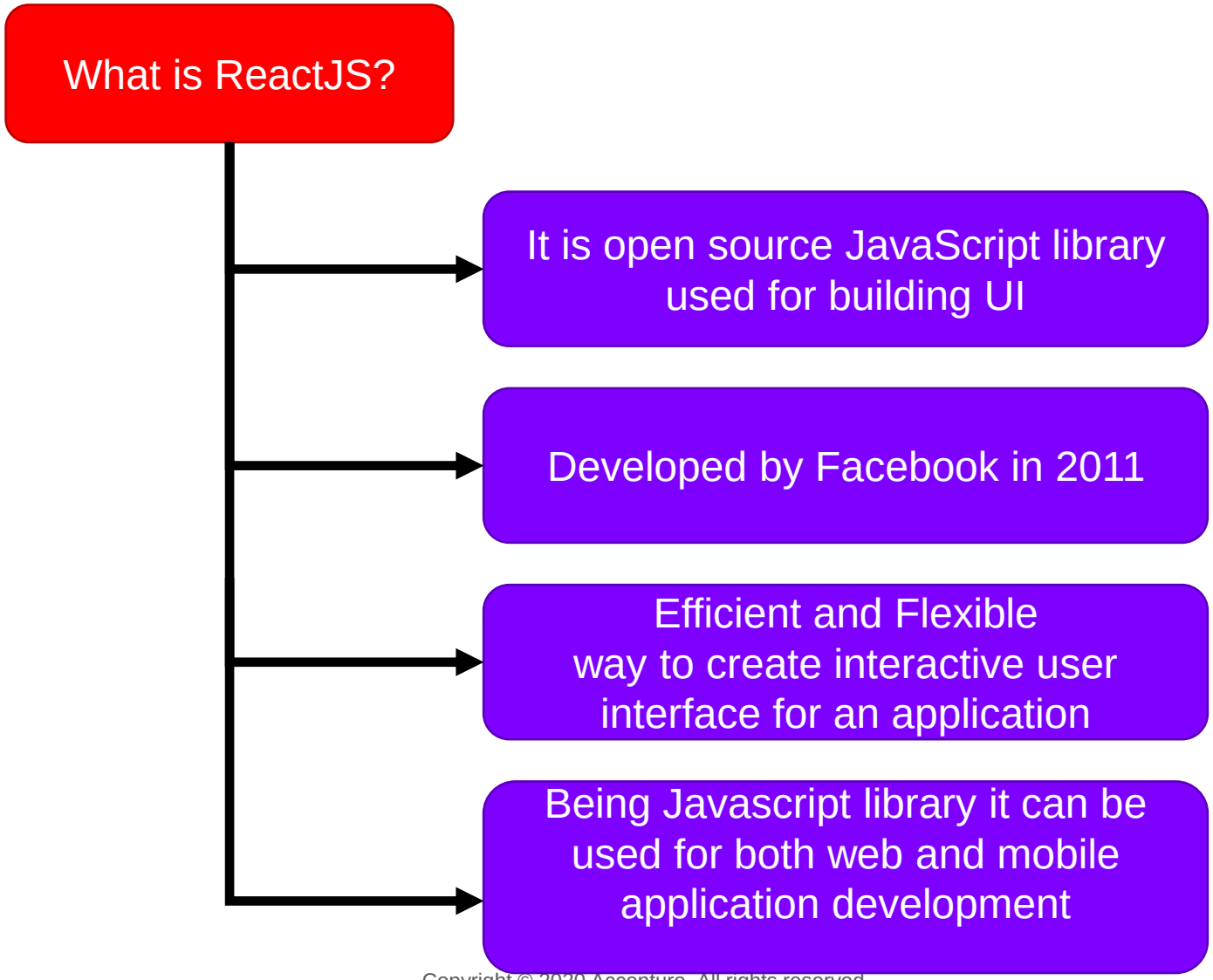


accenture

AGENDA

- Introduction to ReactJS
- Features of ReactJS
- Create React app
- Understand project structure

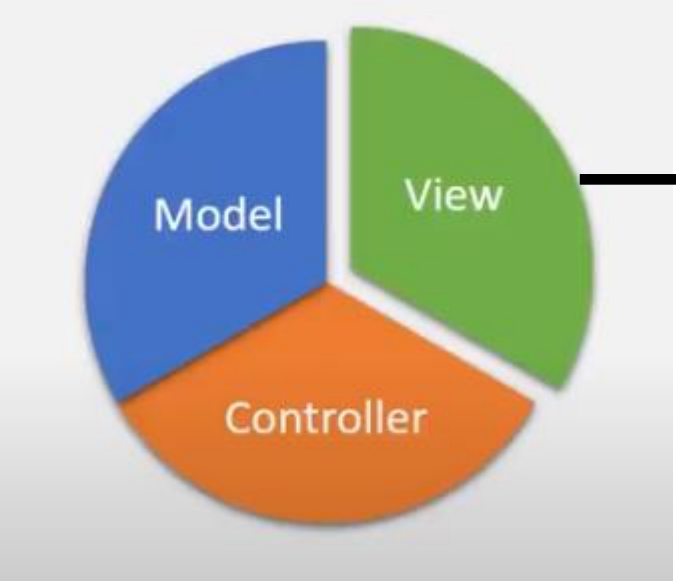
INTRODUCTION TO REACTJS



accenture

INTRODUCTION TO REACTJS

MVC Pattern



- In a Model-View-Controller architecture, React is the View using this one can design UI/View for an applications
- React also responsible for how application looks and feels.

WHY REACTJS

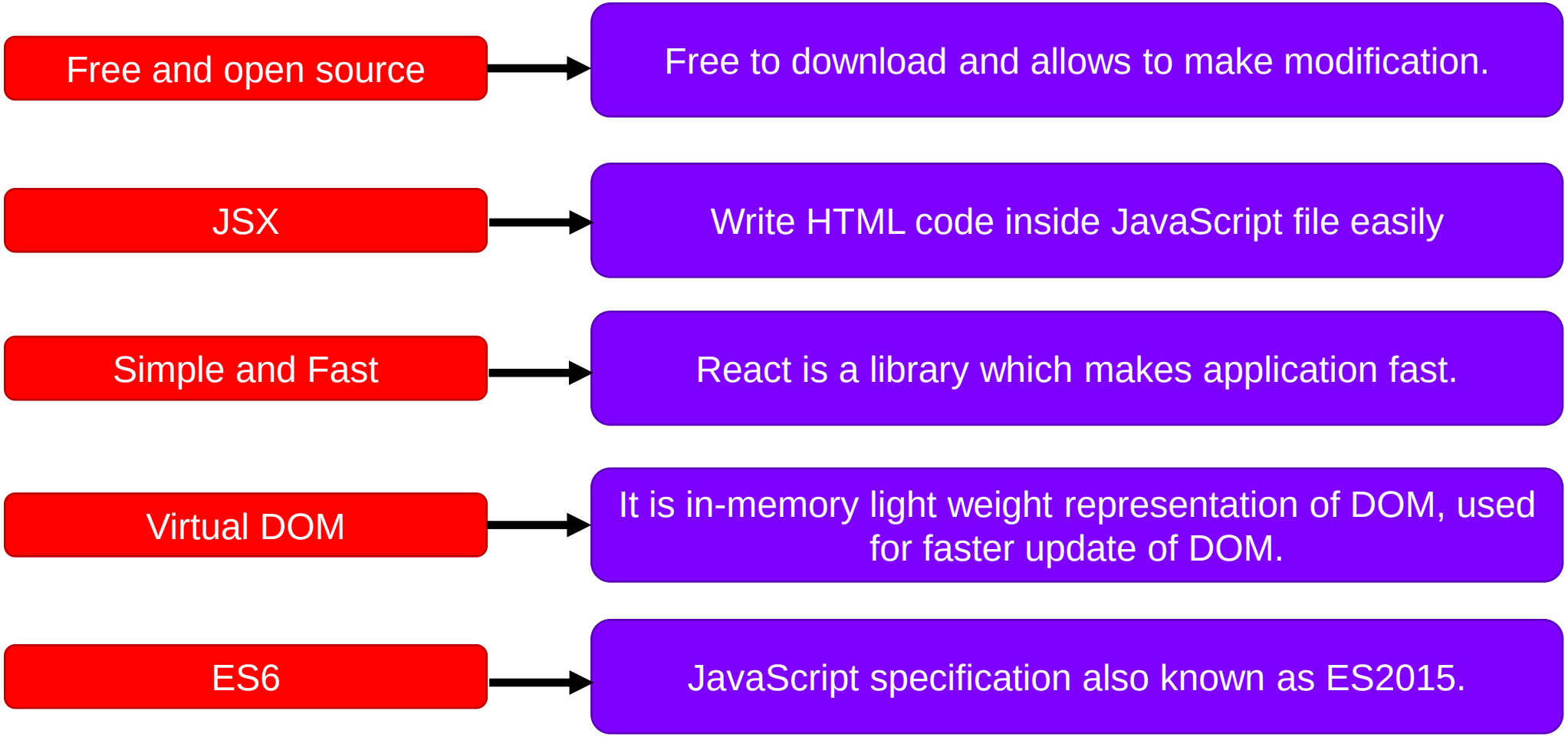
- Using react we can develop dynamic web applications in an easy way.
- React also provides performance enhancements.
- React provides reusable components using which development time can be reduced.
- React supports one way data flow which makes react easy to test and maintain.
- React also provides dedicated tools for debugging.
- React is lightweight library.

accenture

COMPANIES USING REACTJS

React is popular and used by following

REACTJS FEATURES



VIRTUAL DOM

When the HTML code(UI) is processed by the browser, the browser creates DOM with respect to the HTML code

DOM provides the way elements are organized and give a way to access each element

Any change in the UI, the browser will reconstruct the entire DOM once again. This is time consuming process

React uses virtual DOM concept which is efficient way to update the DOM. This makes react application faster

accenture

VIRTUAL DOM

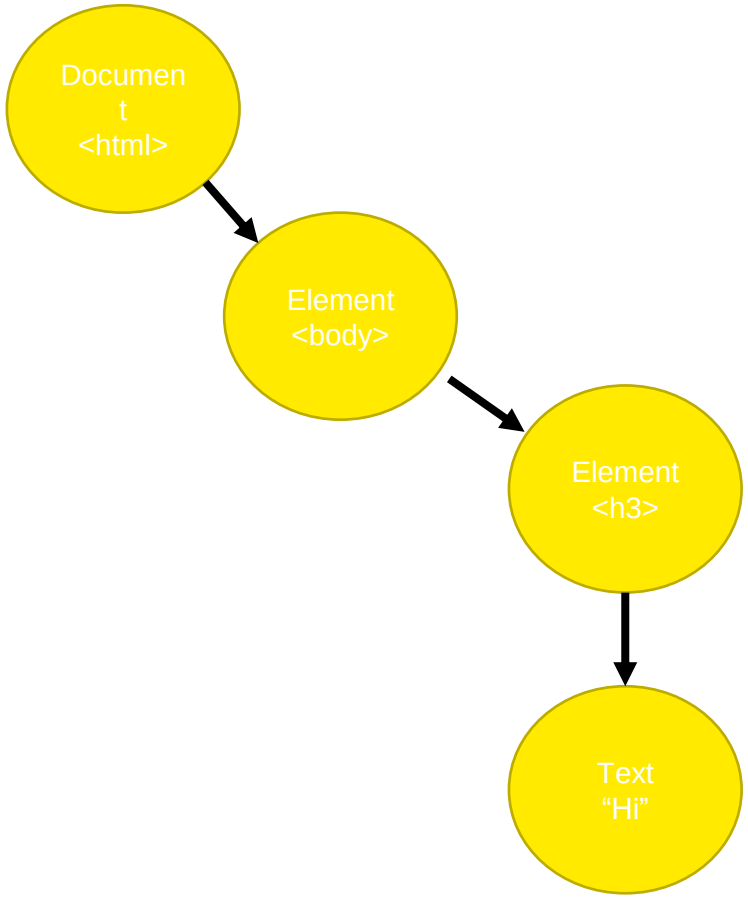
Index.html

```
<html>
   <body>
          <h3>Hi</h3>
    </body>
</html>
```

Every individual element will have corresponding DOM object

So when the above code is executed the DOM will look like this

DOM



VIRTUAL DOM

Index.html

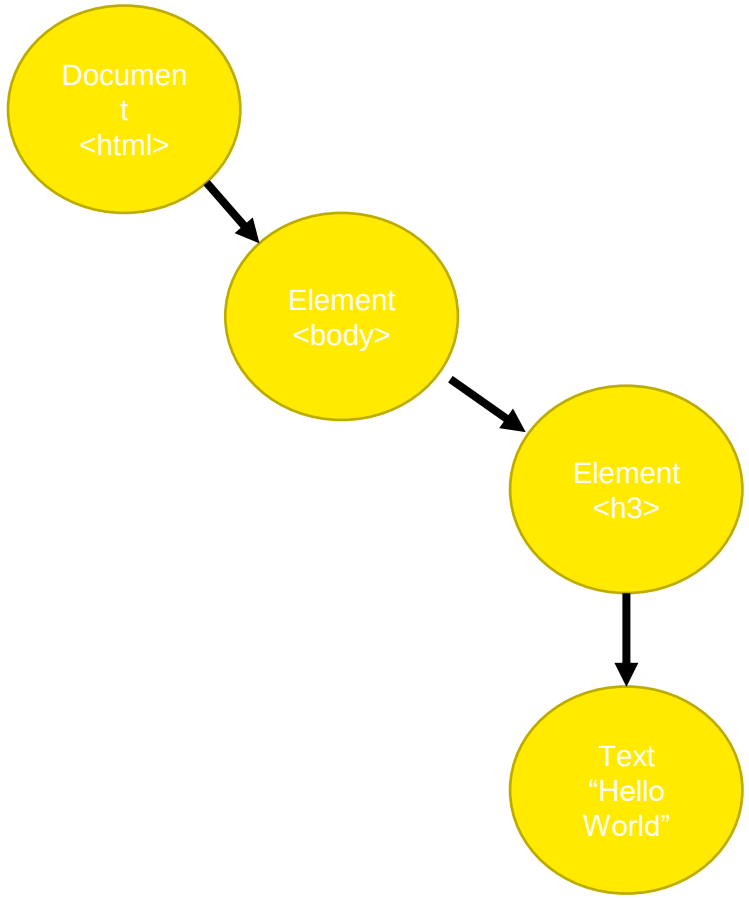
```
<html>
   <body>
          <h3>Hi</h3>
    </body>
</html>
```

If the code changes

Index.html

```
<html>
   <body>
          <h3>HelloWorld</h3>
    </body>
</html>
```

DOM



accenture

VIRTUAL DOM

Virtual DOM is exact copy of real DOM. Which is created by react

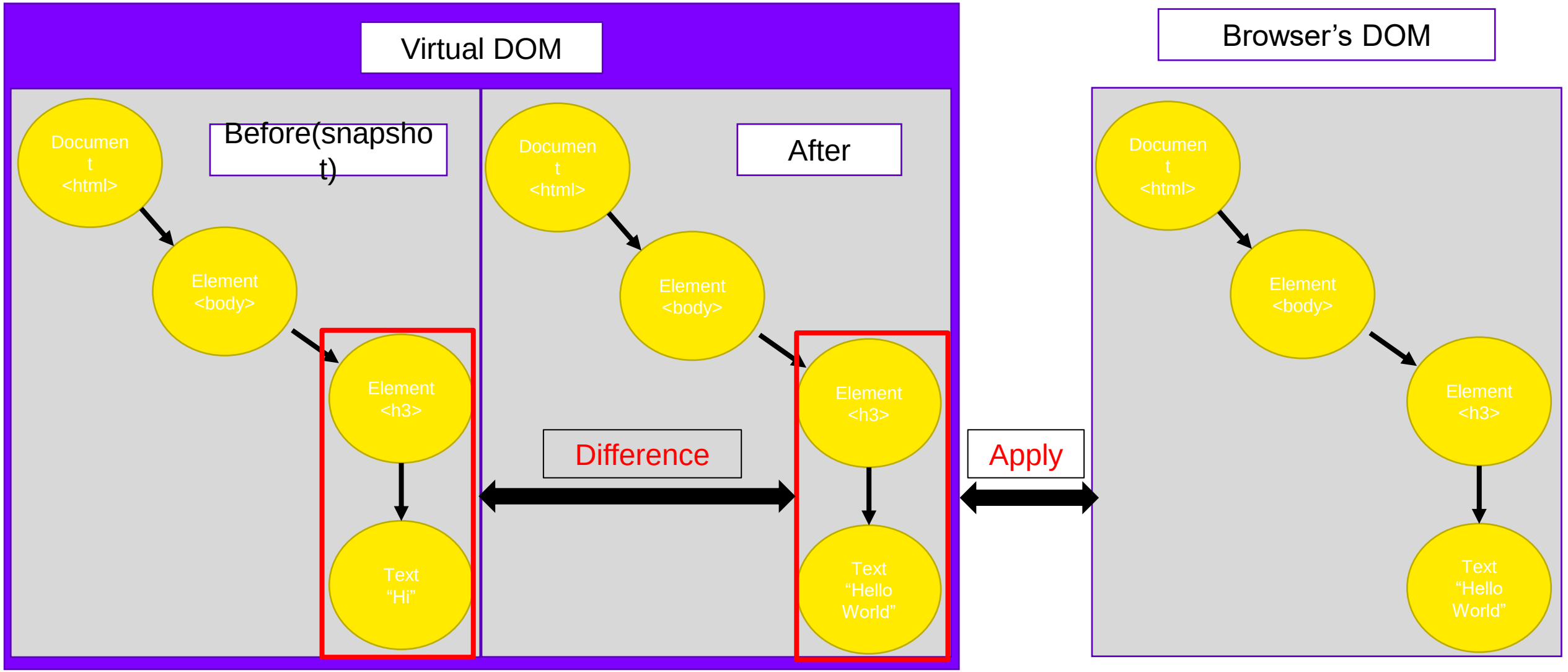
Virtual DOM uses its own algorithm to find the difference between earlier snapshot which it has taken before updating with the latest DOM. This process is known as “Diffing”

If there are any changes identified then those changes will be applied to browser’s actual DOM and only those objects on the real DOM are modified

This speeds up DOM manipulation

accenture

VIRTUAL DOM

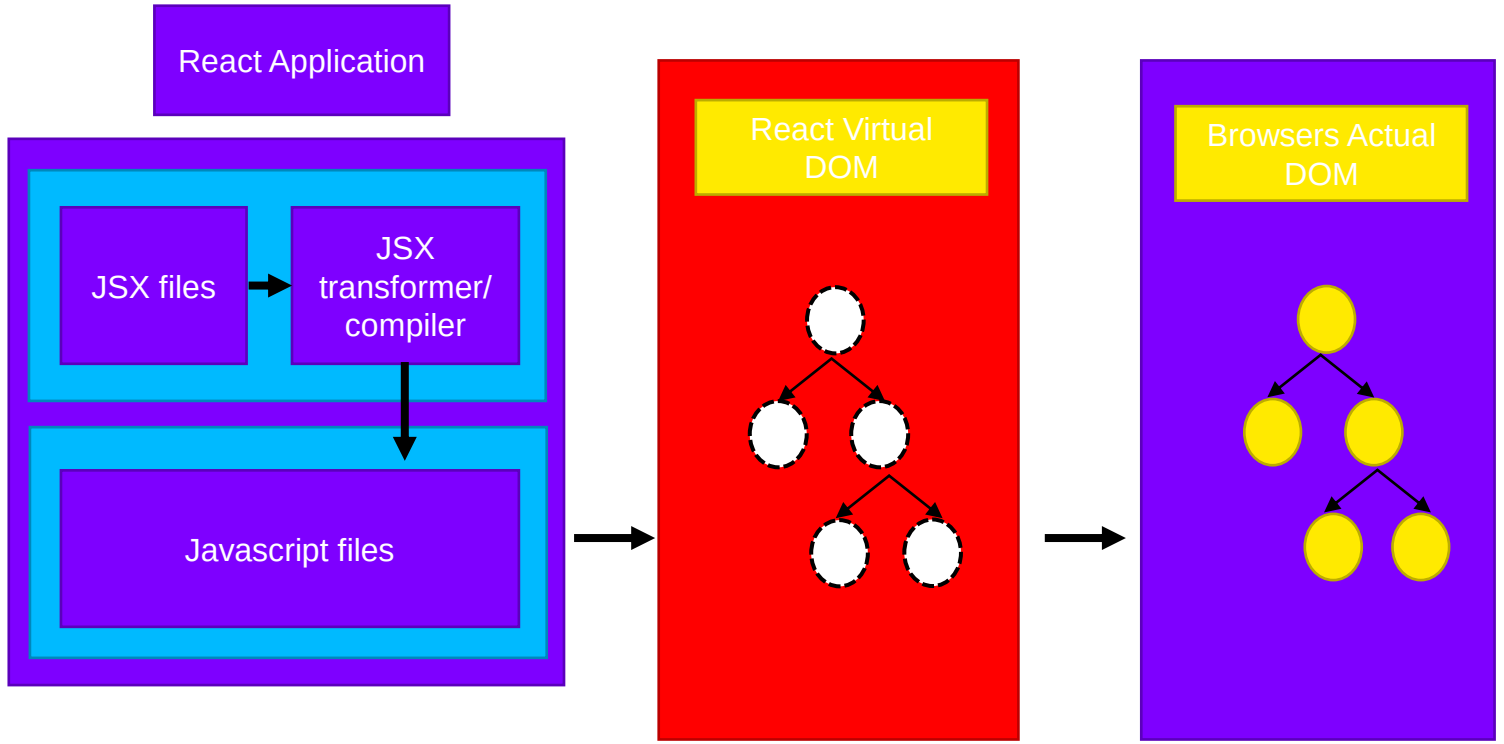


REACTJS WORKFLOW

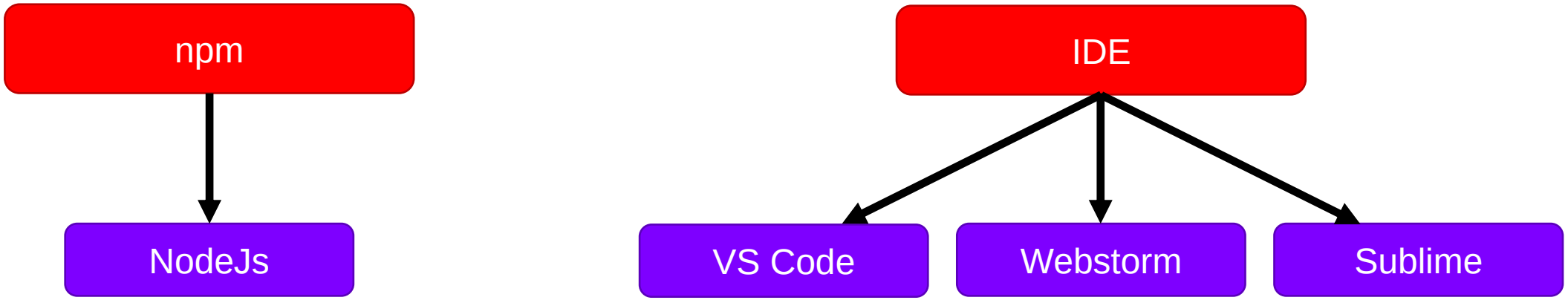
React Application will consist of javascript files, it can also include JSX code which will be converted by JSX transformer/compiler

These files will be given to react's virtual DOM for synchronization

Finally the browser's DOM gets updated and display the output



SOFTWARE REQUIREMENTS



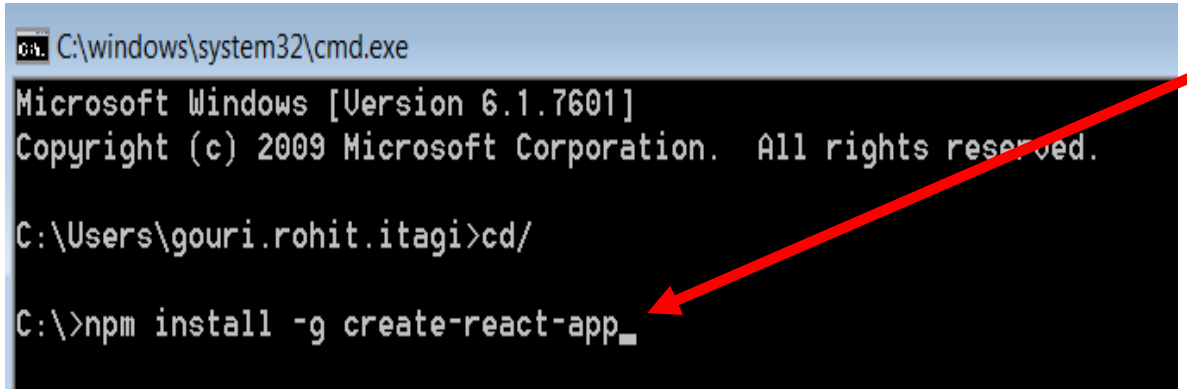
Note :

NodeJs version must be
>= v6

CREATE FIRST REACT APP

Step 1

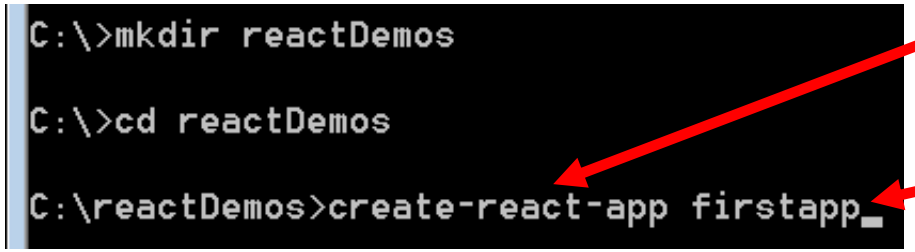
Open command prompt and go to C: drive and Install create-react-app globally



Npm package to create react app

Step 2

Create new directory in C: drive, go to that specific directory and create application using “create-react-app”



This command is used to create react application

Name of the application

CREATE FIRST REACT APP

Below screen will display after the successful react application creation.

```
Success! Created firstapp at C:\reactDemos\firstapp
Inside that directory, you can run several commands:

  yarn start
    Starts the development server.

  yarn build
    Bundles the app into static files for production.

  yarn test
    Starts the test runner.

  yarn 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 firstapp
  yarn start

Happy hacking!

C:\reactDemos>
```

Step 3

To execute the application execute “npm start”

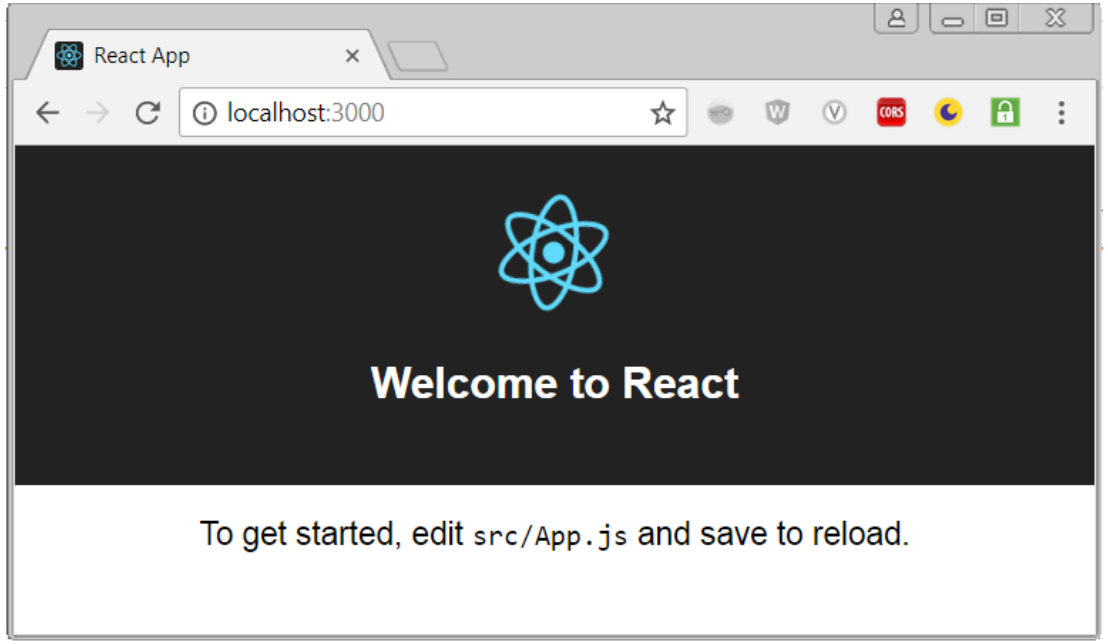
```
C:\reactDemos\firstapp>npm start
```

The application will run on localhost with port number 3000

```
npm
Compiled successfully!

You can now view firstapp in the browser.

  Local:            http://localhost:3000/
  On Your Network:  http://10.116.207.215:3000/

Note that the development build is not optimized.
To create a production build, use yarn build.
```

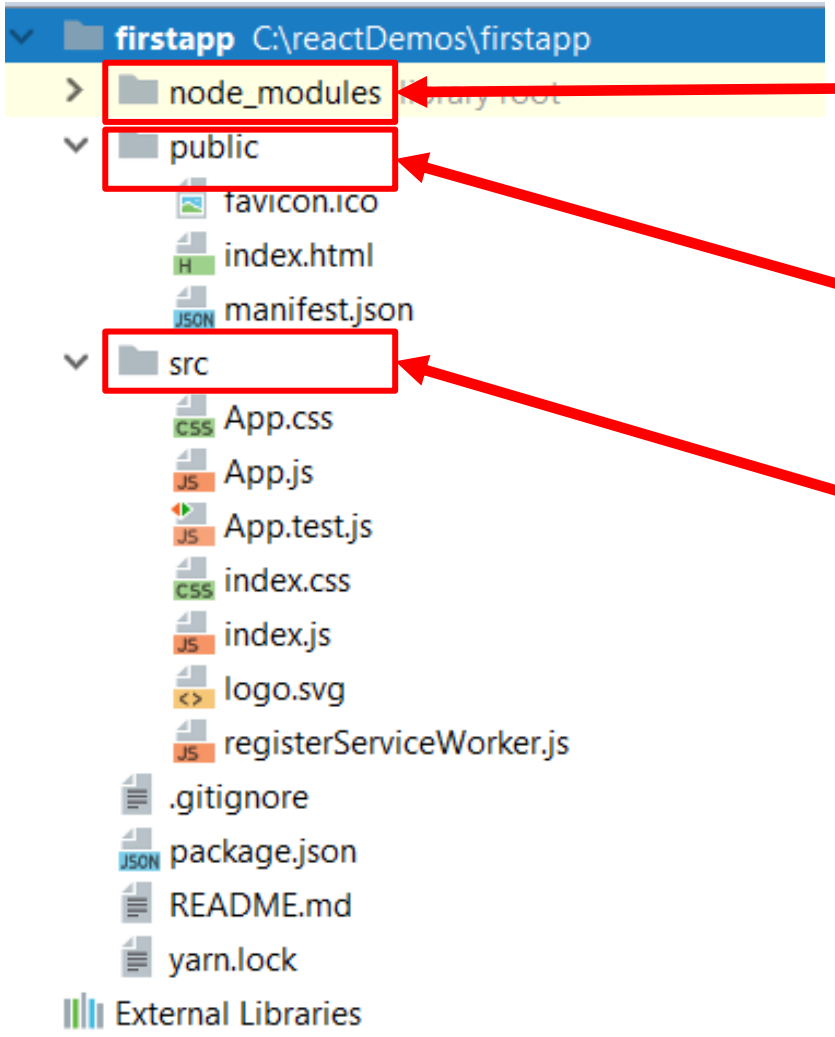
CREATE FIRST REACT APP

Output



PROJECT STRUCTURE

Open the react application “firstapp” using IDE.



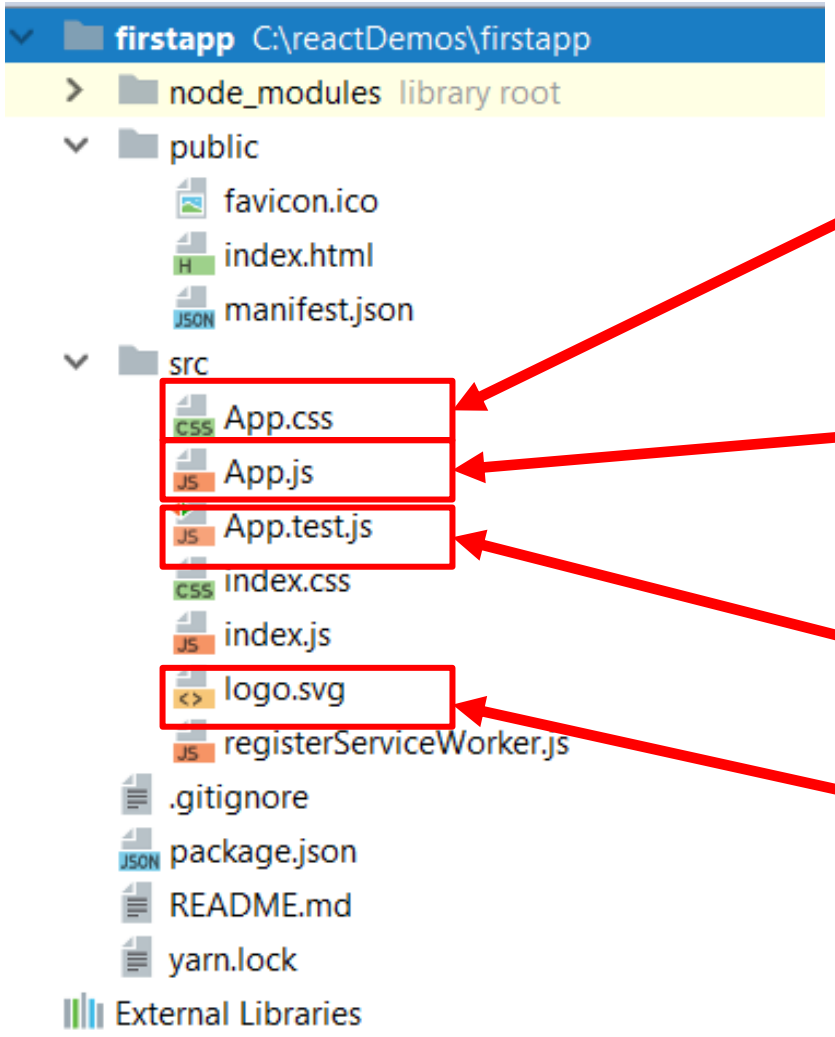
Holds all dependent npm modules

Contains index.html file

Javascript, style sheets and testing files are placed here

PROJECT STRUCTURE

Open the react application “firstapp” using IDE.



This file consists of styles required for app.js file.

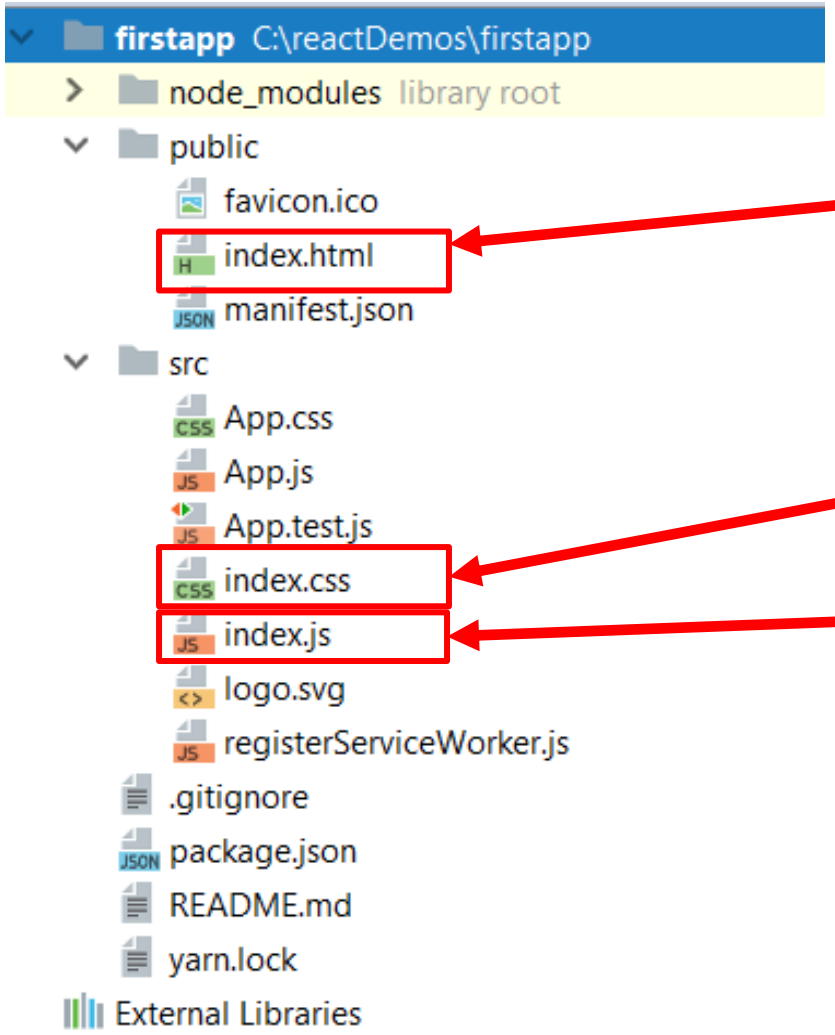
This file contain the UI for application. This can also be referred as component

This file will contain testing related code.

This is image file.

PROJECT STRUCTURE

Open the react application “firstapp” using IDE.



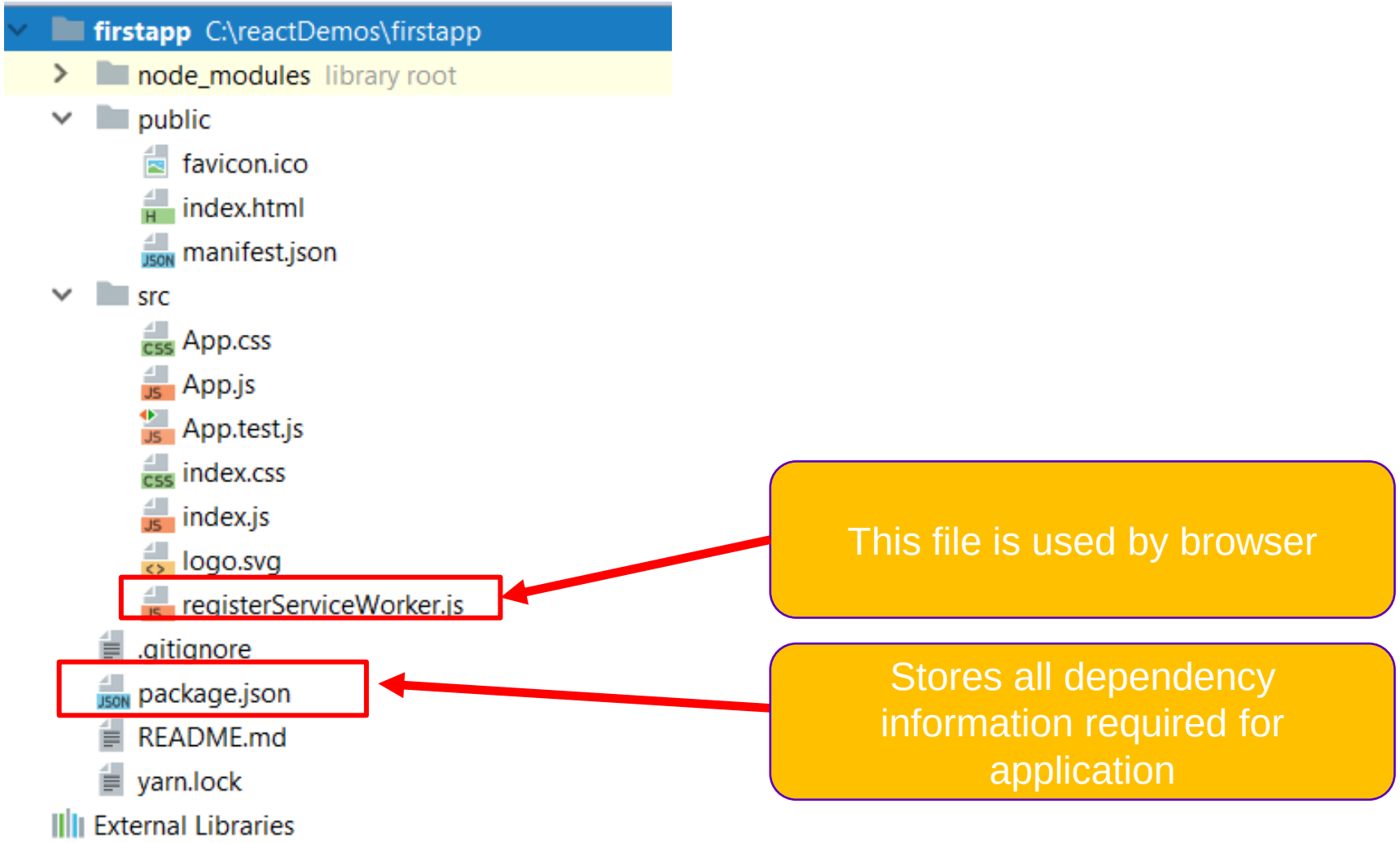
Startup html file which consist of “root” element

This file consists of styles required for index.js file.

This file will bootstrap App.js and renders(display) the content using index.html

PROJECT STRUCTURE

Open the react application “firstapp” using IDE.



PROJECT STRUCTURE

App.js.

```
import React, { Component } from 'react';
import logo from './logo.svg';
import './App.css';

class App extends Component {
  render() {
    return (
      <div className="App">
        <header className="App-header">
          <img src={logo} className="App-logo" alt="logo" />
          <h1 className="App-title">Welcome to React</h1>
        </header>
        <p className="App-intro">
          To get started, edit <code>src/App.js</code> and save to reload.
        </p>
      </div>
    );
  }
}

export default App;
```

In ReactJS component will be created as javascript file.

App.js is a JavaScript file, and it uses keywords like “class”, ”extends” and “exports”

These new features are supported by ECMA Script 2015 or ES6

accenture

TRANSPILATION

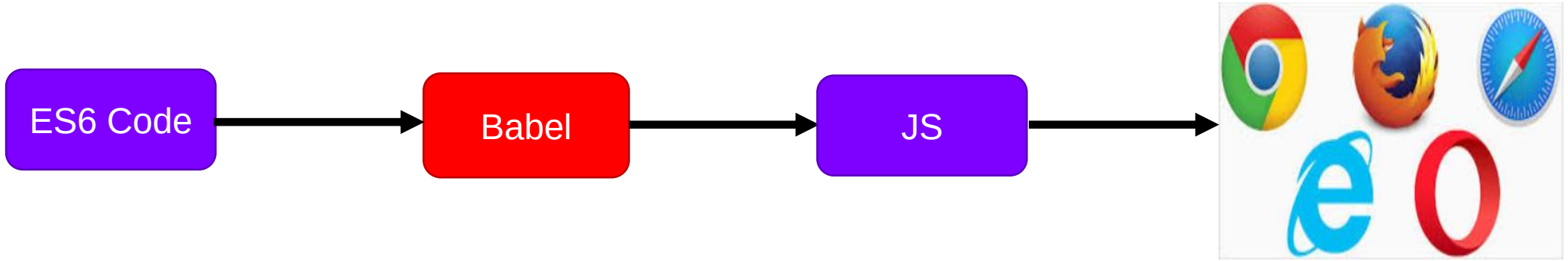
Browser maynot understand ES6 code, however it can understand plain vanilla javascript code.

The process which converts ES6 code to plain javascript code is known as “Transpilation”

The compiler which would perform transpilation is known “Transpiler”

Babel

Is a compiler for react application which can perform transpilation.



PROJECT STRUCTURE

App.js.

```
import React, { Component } from 'react';
import logo from './logo.svg';
import './App.css';

class App extends Component {
  render() {
    return (
      <div className="App">
        <header className="App-header">
          <img src={logo} className="App-logo" alt="logo" />
          <h1 className="App-title">Welcome to React</h1>
        </header>
        <p className="App-intro">
          To get started, edit <code>src/App.js</code> and save to reload.
        </p>
      </div>
    );
  }
}

export default App;
```

JSX

HTML code written inside Javascript file.

JSX

Stands for “Javascript XML”

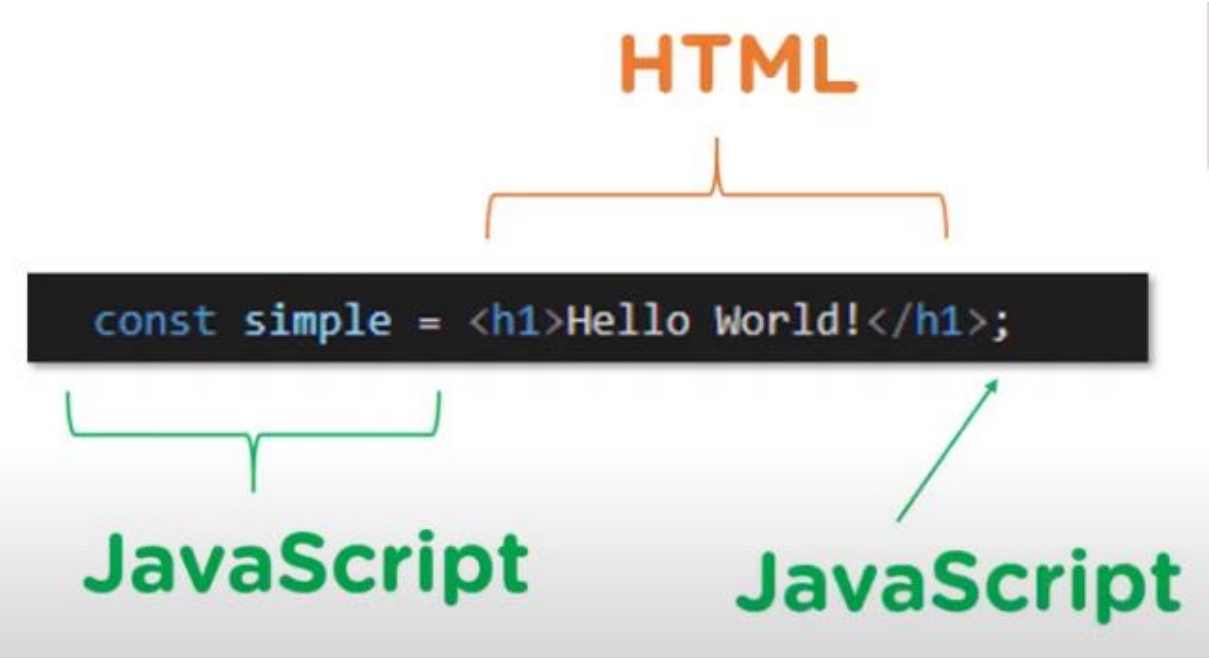
It is an extension to javascript using XML/HTML

It is combination of Javascript and HTML

Allows to combine expressions, calculations inside markup

```
<h3>Total = {2+2}</h3>
```

JSX EXAMPLE



- In the example shown const simple is javascript code.
- <h1>Hello World!</h1> is a HTML code
- Semicolon is optional still it is a part of javascript code

Hence JSX is combination of Javascript and HTML

UNDERSTAND HOW CODE WORKS

App.js

```
import React, { Component } from 'react';
import logo from './logo.svg';
import './App.css';

class App extends Component {
  render() {
    return (
      <div className="App">
        <header className="App-header">
          <img src={logo} className="App-logo" alt="logo" />
          <h1 className="App-title">Welcome to React</h1>
        </header>
        <p className="App-intro">
          To get started, edit <code>src/App.js</code> and save to reload.
        </p>
      </div>
    );
  }
}

export default App;
```

index.html

```
<div id="root">


</div>
```

index.js

```
ReactDOM.render(<App />, document.getElementById('root'));
```

App.js will have user interface which we want to display

Index.html will have section with the id=root

Index.js is responsible to capture the UI from App.js and renders(display) this UI in div section of index.html

accenture

MODULE SUMMARY

- What is ReactJS
- Features of ReactJS
- Workflow of ReactJS
- Project structure
- Why Babel
- What is JSX
- Understand how code works

Questions

THANK YOU