# React Fundamentals

Tessema Mengistu (Ph.D.)
Department of Computer Science
Virginia Tech
Mengistu@vt.edu

# Outline

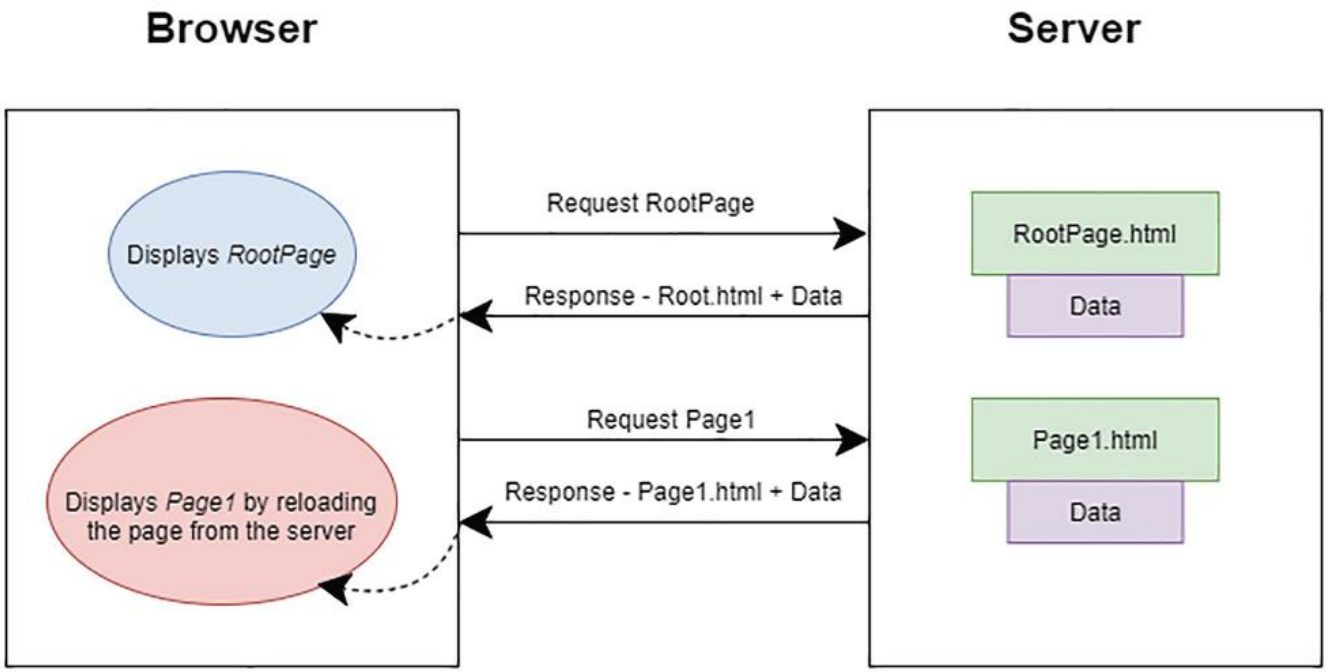
- Introduction
- React Basics
- React Components
- Props and State
- React Router

# Introduction

- Front-end development
  - Look and feel
    - HTML and CSS
  - Interactive behavior
    - JavaScript
- Front-end Architecture
  - Single Page Architecture (SPA)
  - Multi-Page Architecture (MPA)

# Introduction

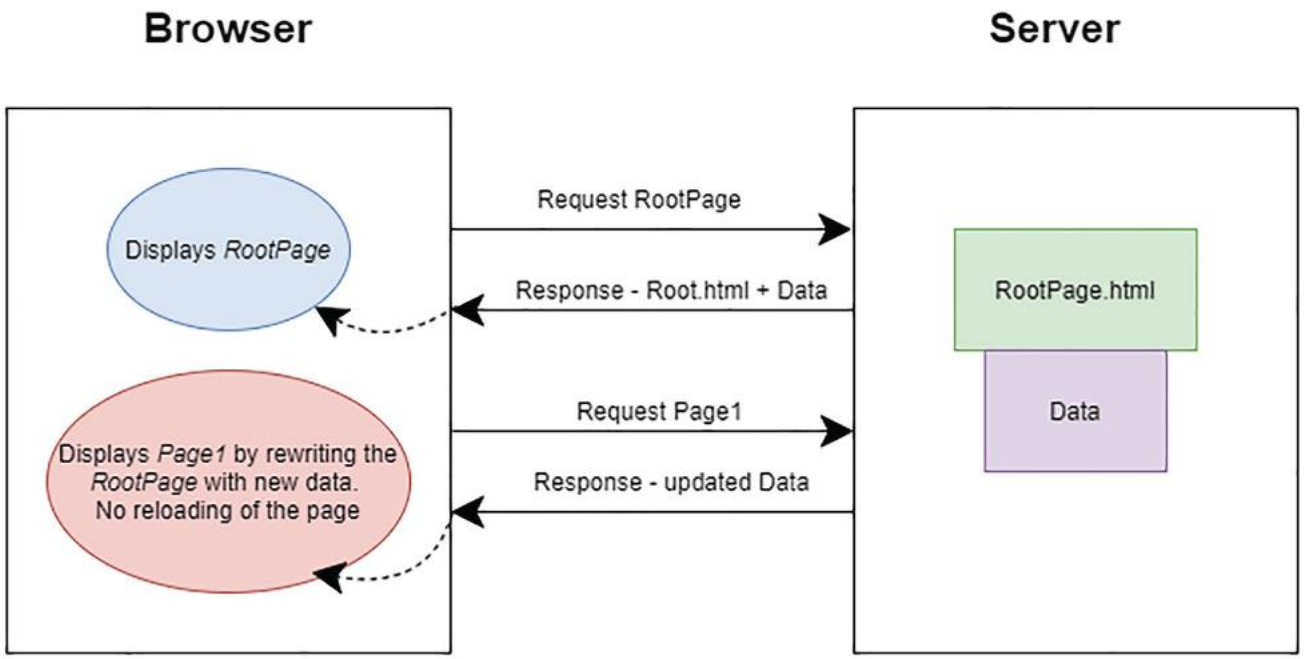
- In a Multi-page application (MPA), the browser gets multiple pages from the server




Source: Narayn, Hari. *Just React: Learn React the React Way*. Berkeley, CA: Apress L. P, 2022. Web

# Introduction

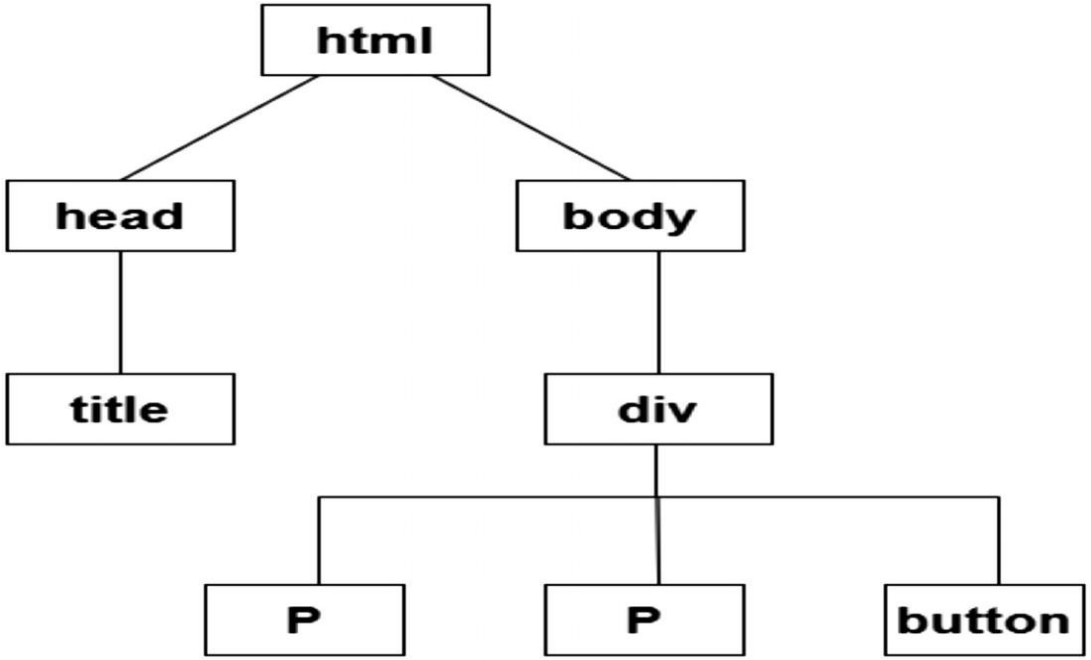
- In a single-page application (SPA), the browser gets only one page from the server




Source: Narayn, Hari. *Just React: Learn React the React Way*. Berkeley, CA: Apress L. P, 2022. Web

# Introduction

- DOM – Document Object Model
  - Tree structure that the browser builds as it parses HTML



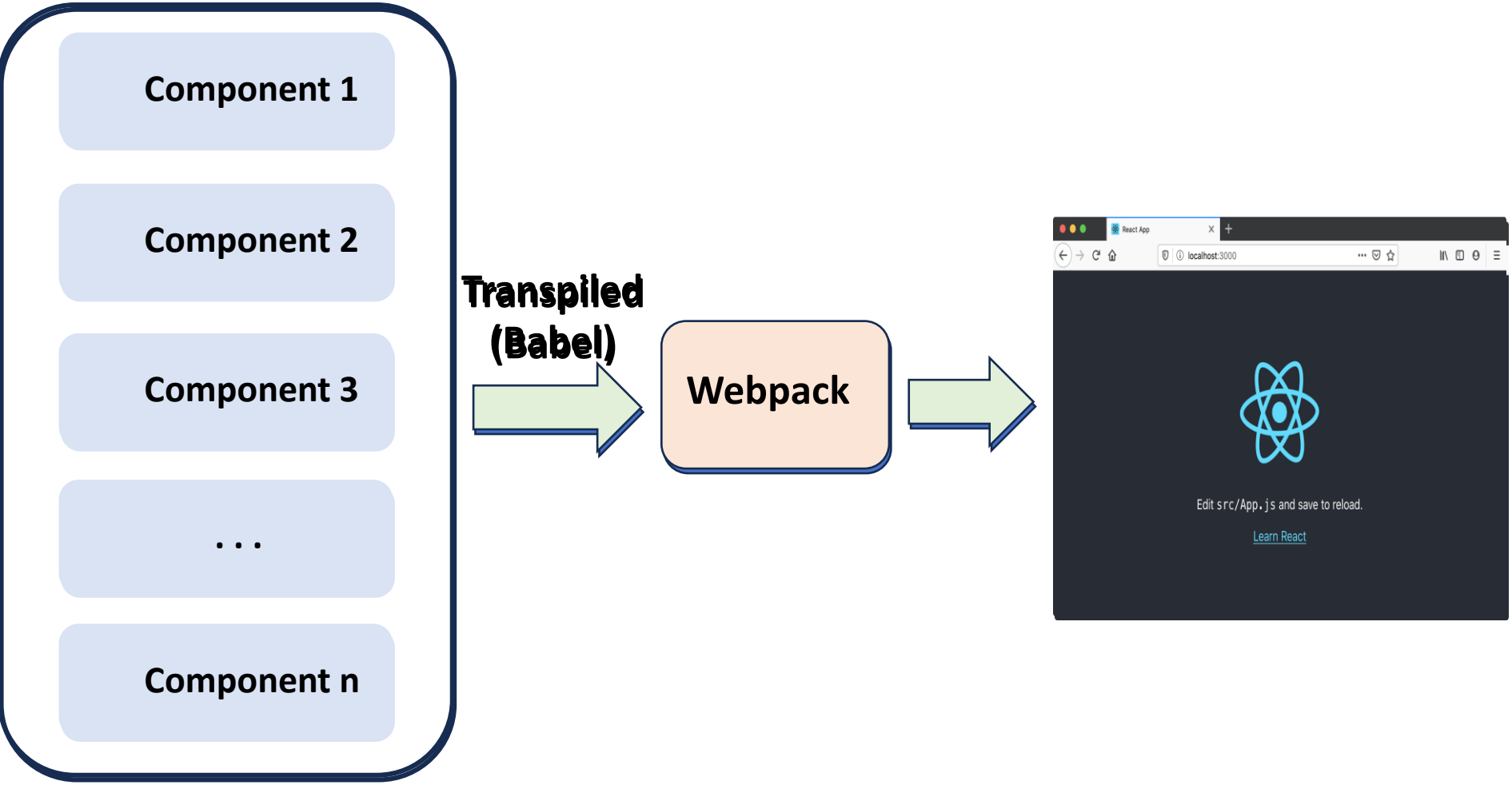
# Introduction

- There are many libraries and frameworks available that are built on top of JavaScript
  - React, Angular, Vue, jQuery, etc.
- A framework comes with a structure and more features

# React Basics

- React is a JavaScript library
  - Created by Facebook
- React focuses on the user interface (UI)
  - Reusable user interface (UI) components
- Light weight and very responsive
  - Uses Virtual – DOM
- Very popular
  - Facebook, Instagram, PayPal, Netflix, Airbnb, etc.

# React Basics



# React Basics

- In the DOM, HTML elements are represented as a tree of objects
- React constructs a second tree of elements as per the components
  - Virtual DOM tree
  - The Virtual DOM tree is a blueprint that exists in memory but is never rendered
- React creates two copies of the DOM in the memory

# React Basics

- Diffing
  - When there is a change  in the DOM
    - React changes one copy of the Virtual DOM
    - Compares that copy with the other copy to determine what has changed.
  - This process is called diffing
- React batches these changes and applies them to the real DOM in one shot
  - React updates the real DOM only after the diffing process
    - Reduces direct manipulation and helps in improved performance

# React Basics

- Tools necessary to create a React based UI
  - HTML, CSS
  - React library
  - Transpiler
    - Babel
  - Bundler
    - Webpack, Vite
  - Node.js
  - Optional
    - Typescript, Eslint, etc.

# React Basics

- *create-react-app* command
  - A CLI command that setup the environment with all the necessary tools to create a react based application
    - Webpack, Babel, react, react-dom, etc.
  - Syntax
    - *create-react-app app-name*

# React Components

- Components
  - Building blocks of any React app
  - A JavaScript class or function
    - takes an input optionally
    - returns a React element
      - Describes how a section of the UI should appear
  - Enable reusability and modularity
  - Component name must start with a capital letter*

*https://legacy.reactjs.org/docs/jsx-in-depth.html#user-defined-components-must-be-capitalized

# React Components

- JSX
  - JavaScript XML
  - Describes how the UI looks like
  - Appears to be HTML, but it is a syntax coating on top of JavaScript
  - **transpiled** into JavaScript to be rendered by browsers using a tool such as Babel

# React Components

```
    <div className = "example">
        <h1> This is example JSX!</h1>
    </div>
);
```

# React Components

```
1  function Example()
2  {
3      return (
4          <div className = "example">
5              <h1> This is example JSX!</h1>
6          </div>
7      );
8  }
9  export default Example;
```

# Props and State

- Props
  - Optional parameter that is passed into a React component
  - Allow components to communicate
  - Read-only
  - Disadvantage
    - Prop drilling problem

# Props and State

- State
  - JavaScript object that represents information about the component's current situation
    - Example: Error or loading
  - A change to a component state causes the component to refresh – re-rendering
  - Defined using a useState function (hook) from React
  - Syntax :

```
const [state, setState] = useState(initialState);
```

  - `useState` hook:
    - Remember values internally when the component re-renders
    - Tell React to re-render the component when the value changes - setState

# Props and State

- React Hooks
  - Special functions that allow developers to hook into state and life cycle of React components
  - JavaScript functions that accept arguments and return specific values
  - Should be imported before use
  - Example:
    - useState
    - useEffect
    - useReducer
    - useParam
    - . . .

# React Routers

- React Router
  - A routing library for React apps that gives navigational capability
  - Responsible for selecting what to show in the app for a requested path
  - Is part of a package called `react-router-dom`
    - Should be installed
  - Different kinds of routers
    - BrowserRouter
    - HashRouter
    - …

# Resources

- React Docs - reactjs.org/docs/getting-started.html
- MDN - developer.mozilla.org
- DevDocs - devdocs.io