

# Frontend Development

# What is Frontend?

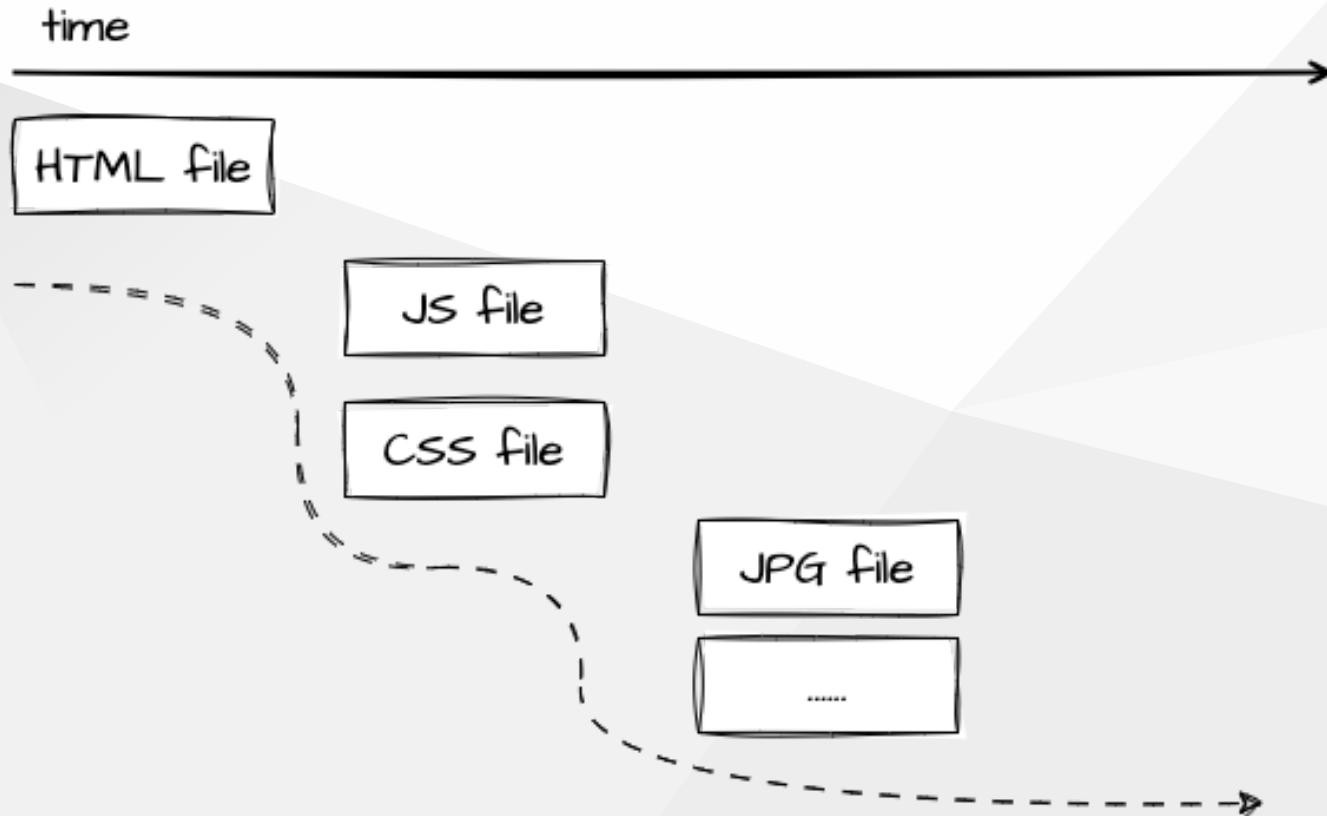
- Frontend is the part of the website that users can see and interact with.
- Frontend is also called **client-side**.
- Frontend is built with HTML, CSS, and JavaScript.

# A bit history of Frontend

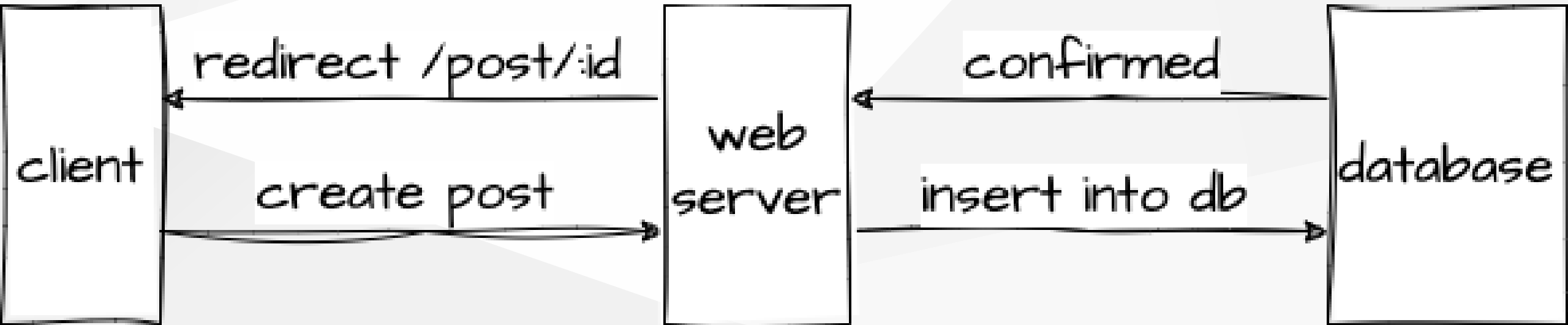
- In the early days of the web, websites were built with HTML only. Then, CSS was introduced to style the HTML. Later, JavaScript was introduced to make the website interactive.
- In 2006, jQuery was introduced to make JavaScript easier to use, but lack of data handling.
- From 2010 to 2011, AngularJS, Backbone, Knockout, and Ember were introduced.
- In 2013, React was introduced. Vue started in 2014.
- In 2015, Redux was introduced.
- In 2016, Angular 2 was introduced.

# Web Application 101

- Websites vs Web Applications



# Web Application 101 (cont'd)



# Web Application 101 (cont'd)

- Single Page Application (SPA)
- Get one HTML file and the rest is JavaScript

```
<!DOCTYPE html>
<html>
  <head>
    <title>My First SPA</title>
  </head>
  <body>
    <div id="root"></div>
    <script src="bundle.js"></script>
  </body>
</html>
```

# React

- React is a JavaScript library for building user interfaces.
- React is developed and maintained by Meta (Facebook).
- React is used to build single page applications (SPA), to build mobile applications (React Native), and to build desktop applications (Electron).
- React is component-based. A component is a piece of the user interface.
- React uses JSX, a syntax extension to JavaScript, to write HTML in JavaScript.

# History of React

- React was created by Jordan Walke, a software engineer at Facebook.
- React was first deployed on Facebook's newsfeed in 2011 and on Instagram in 2012.
- React was open-sourced at JSConf US in May 2013. RoR and Python start to support React right after.
- In 2015, React Native was introduced. GraphQL at React.js Conf.
- In 2016, React Fiber was introduced with React 16.
- React 18 was released in 2022 with concurrent rendering, SSR, and Suspense.



# Why React?

- component-based, declarative
- learn once, write anywhere
- virtual DOM, fast rendering
- large community
- over 11 million websites use React, which is roughly 4% of all websites
- Google, Facebook, Netflix, Instagram, Airbnb, Uber, Discord, Reddit, Slack, WhatsApp ...
- [https://2022.stateofjs.com/en-US/libraries/front-end-frameworks/#front end frameworks experience ranking](https://2022.stateofjs.com/en-US/libraries/front-end-frameworks/#front%20end%20frameworks%20experience%20ranking)

# Hello World in React

```
import React from 'react';  
import ReactDOM from 'react-dom';  
  
const App = () => {  
  return <h1>Hello World</h1>;  
};  
  
ReactDOM.render(<App />, document.getElementById('root'));
```

# JSX

- JSX is a syntax extension to JavaScript.
- JSX is not a requirement for using React.
- JSX makes it easier to write and add HTML in React.
- JSX is compiled to JavaScript.
- JSX is closer to JavaScript than HTML.

# JSX (cont'd)

```
const element = <h1>Hello World</h1>;
```

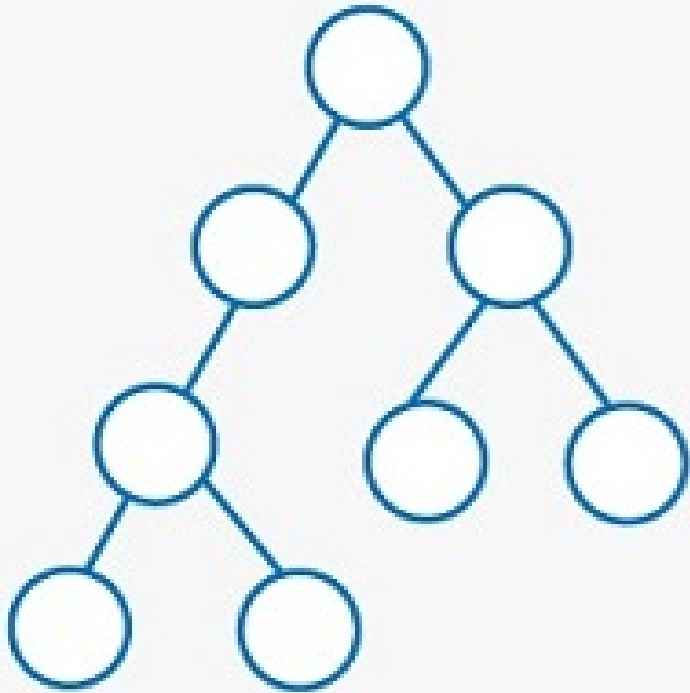
```
const element = React.createElement('h1', null, 'Hello World');
```

# Virtual DOM

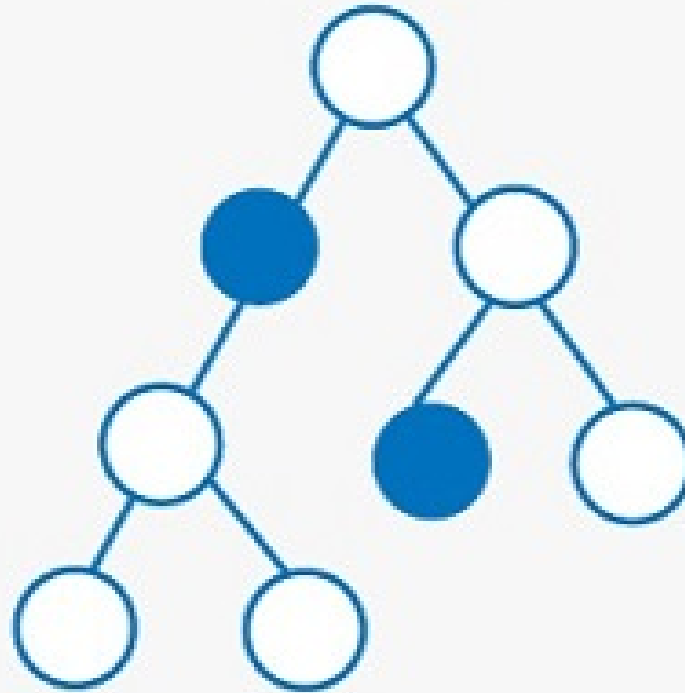
- Virtual DOM is a JavaScript object that represents the DOM, a copy of the DOM, lightweight, and fast.
- Diff strategy
  - Level by level comparison:  $O(n)$
  - based on type
  - based on key in list
  - based on position

# Virtual DOM (cont'd)

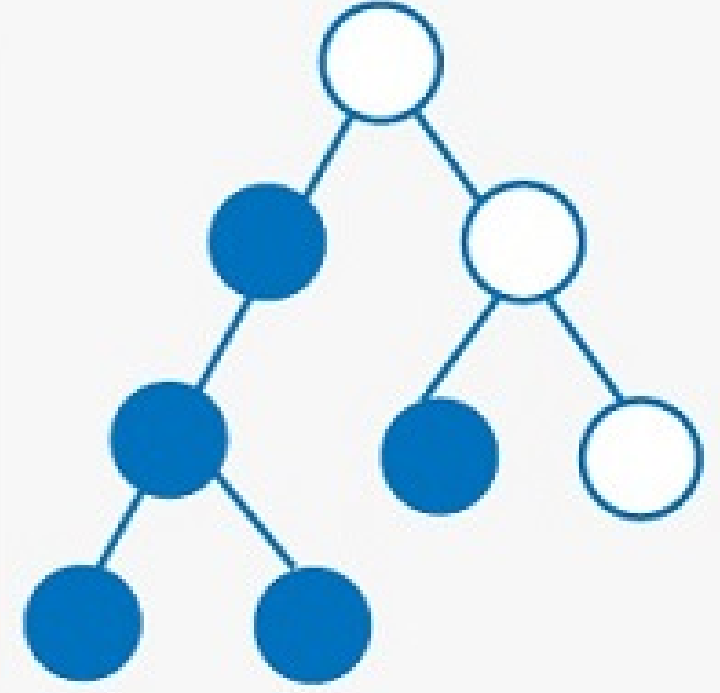
**Virtual DOM**



**New Virtual DOM**



**Affected nodes**



# Setup up a React Project

- Scaffold:
  - [Create React App\(CRA\)](#)
  - [Vite](#)
  - [Next.js](#)
  - [Gatsby](#)
- Manual setup:
  - [Webpack](#)
  - [Babel](#)
  - [ESLint](#)

# Create React App (CRA)

- Create React App is a tool to create React applications.

```
npx create-react-app <app_name>  
cd <app_name>  
npm start
```



# Vite

- Vite is a tool to build modern web applications.

```
npm create vite@latest <app_name> --template react  
cd <app_name>  
npm install  
npm run dev
```

# Project Structure

```
|— public
|   |— favicon.ico
|   |— index.html
|   |— manifest.json
|— src
|   |— App.css
|   |— App.js
|   |— App.test.js
|   |— index.css
|   |— index.js
|— package.json
```

# npm Scripts

```
{  
  "scripts": {  
    "start": "react-scripts start",  
    "build": "react-scripts build",  
    "test": "react-scripts test"  
  }  
}
```

# VSC Extensions

- ESLint
- Prettier
- ES7 React/Redux/GraphQL/React-Native snippets

# React Developer Tools

- React Developer Tools is a Chrome extension to inspect React components.
- <https://chrome.google.com/webstore/detail/react-developer-tools/fmkadmapgofadopljbjfkapdkoienihi>

# React JSX

- JSX must have one root element.
- JSX must be closed.
- `.js` or `.jsx` extension.

```
const element = (  
  <div>  
    <label htmlFor="name">Name</label>  
    <input id="name" type="text" />  
  </div>  
);
```

# React JSX (cont'd)

React Code

Browser

JSX

HTML

htmlFor

for



# React JSX (cont'd)

```
function getTitle(title) {  
  return title;  
}  
function App() {  
  return (  
    <div>  
      <h1>Hello {getTitle('React')}</h1>  
      <label htmlFor="search">Search: </label>  
      <input id="search" type="text" />  
    </div>  
  );  
}
```



# React JSX (cont'd)

- [Synthetic Event](#)
- [Inline Styles](#)

```
const element = (  
  <div>  
    <h1 style={{ marginBottm: '20px' }}>Hello World</h1>  
    <label htmlFor="search">Search: </label>  
    <input id="search" type="text" onChange=(e => console.log(e.target.value)) />  
  </div>  
)
```

# Components

- Components are the building blocks of React applications.
- Reusable and composable.
- **Class components** vs. Function components (later).
- Stateful components vs. Stateless components.
- Representational components vs. Container components.

# Class Components

```
class App extends React.Component {  
  constructor(props) {  
    super(props);  
    this.state = {  
      name: 'React'  
    };  
  }  
  render() {  
    return (  
      <div>  
        <h1>Hello {this.state.name}</h1>  
      </div>  
    );  
  }  
}
```

# State

- State is an object that holds some information that may change over the lifetime of the component.
- State is private to the component.
- **Immutable: State should be updated only by `setState()` method.**
- React re-renders the component when the state changes.
- `setState()` *is asynchronous.*

# State (cont'd)

- `setState(newState, callback)`
- `setState((prevState, props) => newState, callback)`
- don't mix two ways above

# Event Handling

event-handling.jsx

- `bind`
- arrow function