IE 4727 Web Application Design

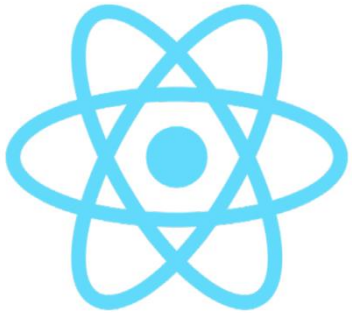
React.js

Lecturer: Dr. Hu Xiao

ReactJS

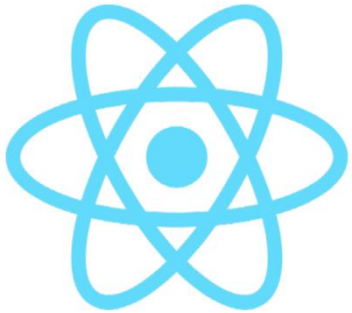
1. **What is React ?**
2. Why React ?
3. Fundamental of React
4. Create React Project

What is React ?

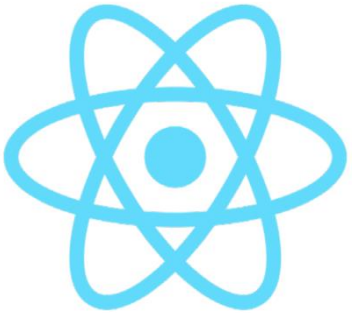


JavaScript Framework

Client-side library

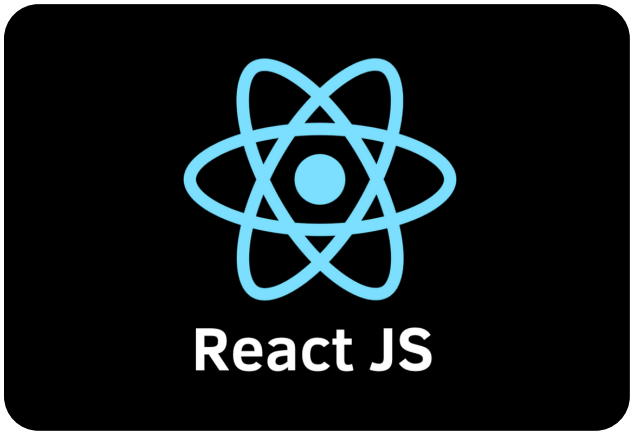


Used for front end web development



Famous for implementing a Virtual DOM

Why React ?



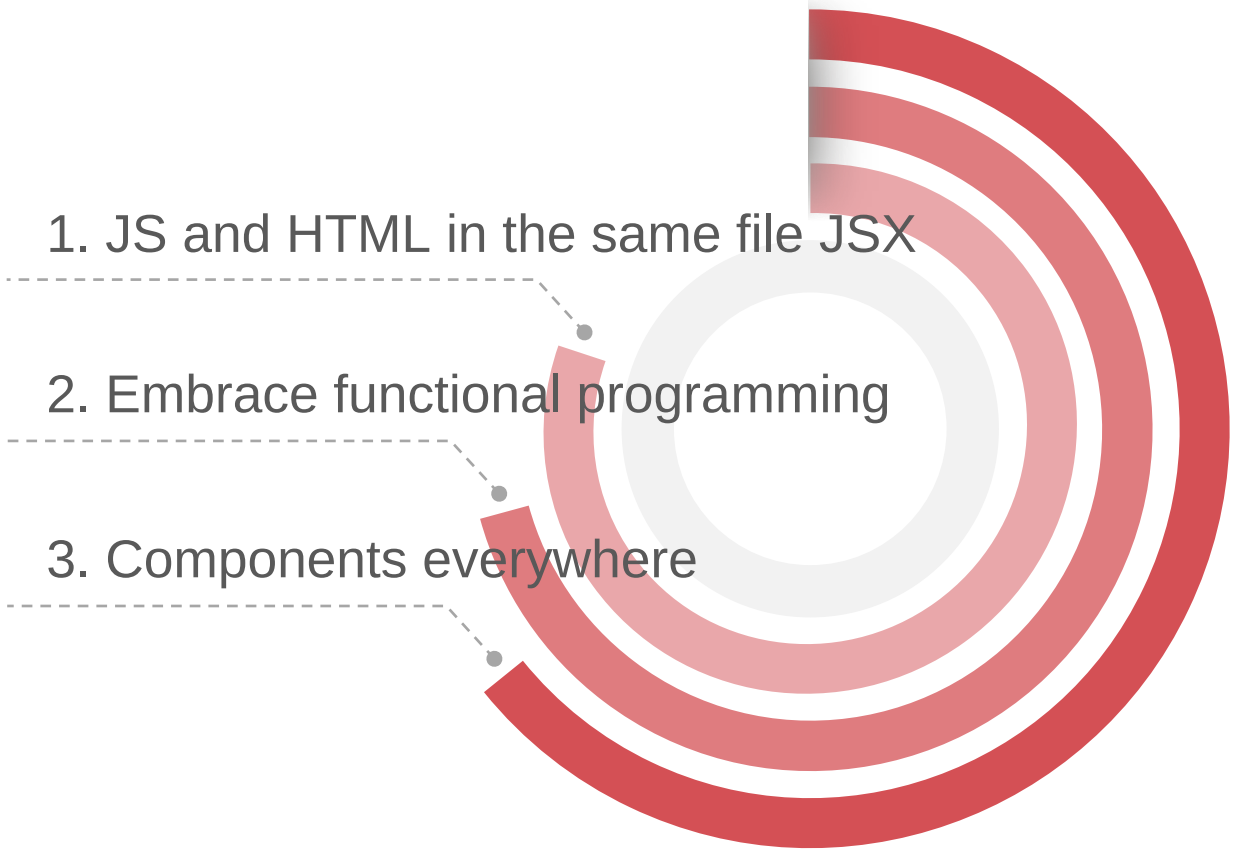
React is Flexible: React uses a component-based architecture, where UIs are broken down into reusable pieces. This makes it easier to manage and maintain complex UIs.

Efficient Updating: React uses a virtual DOM, which is a lightweight copy of the real DOM. When the state of a component changes, React compares the virtual DOM with the real DOM and only updates the parts that have changed. This can lead to better performance compared to manually manipulating the DOM.

JSX: React uses JSX, a syntax extension that allows you to write HTML-like code directly in your JavaScript. This can make your code more readable and maintainable.

Community Support: React has a large and active community of developers, which means there are plenty of resources and tutorials available to help you learn and solve problems.

Fundamentals of React



JS and HTML in the same file

Traditional approach

React approach

JSX: The React Programming Language

Functional Programmming

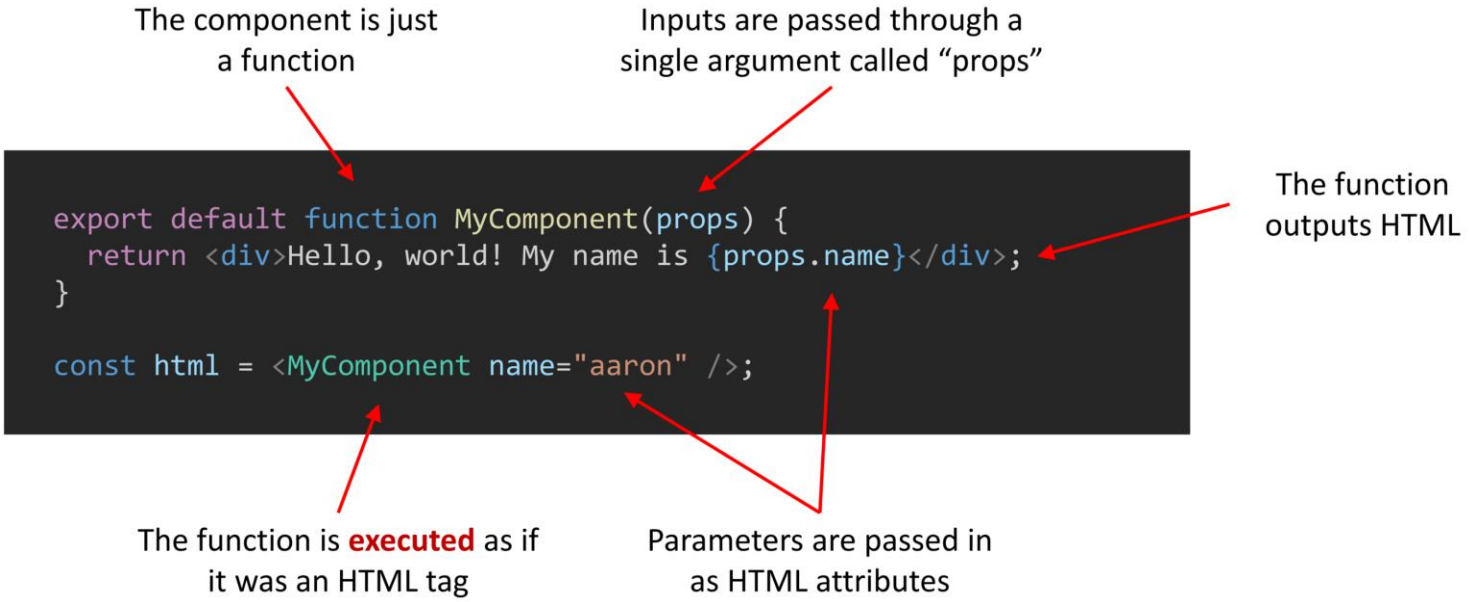
Functional programming is a programming paradigm that treats computation as the evaluation of mathematical functions and avoids changing-state and mutable data. In functional programming, functions are first-class citizens, meaning they can be assigned to variables, passed as arguments, and returned from other functions.

```
let add = function() {
  console.log('Now adding numbers');
  const five = 3 + 2;
};
```

```
function performTask(task) {
  task();
  console.log('Task performed!');
}

performTask(add);
```

```
function foo() {
  return function() {
    console.log('What gets printed?');
  };
}

foo
foo();
foo()();
```

Anatomy of a React component



Everything is a component

Component rendering: When a component function executes, we say it "renders"

Hooks: Special functions that allow developers to hook into state and lifecycle of React components.

State: One or more data values associated with a React component instance.

Lifecycle: The events associated with a React component instance (create, render, destroy, etc).

Built-in Hooks: useState

First React Hook: useState

```jsx
import React, { useState } from 'react';

function Counter() {
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>Count: {count}</p>
      <button onClick={() => setCount(count + 1)}>Increment</button>
    </div>
  );
}
```

state

Function to update the state

A component will only re-render when: 1. A value inside props changes Or 2. A useState setter is called

Download and run React App

Node.js

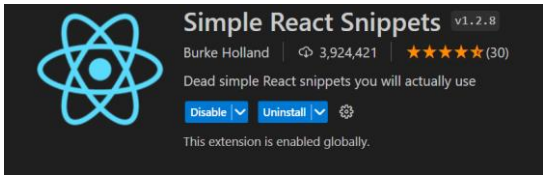
1. Download Node.js from **https://nodejs.org/en/download and add Node.js** to PC environment variables

- Node.js is a JavaScriot runtime built on Chrome's V8 JavaScript engine.
- It allows you to run JavaScript code outside of a web browser.

Node.js is commonly used in the development and deployment process of React applications

VS Code

2. Download and Install React Extension in VS Code



Vite

3. Install vite: Within the VS Code terminal, enter **npm create vite@latest**, and hit enter.

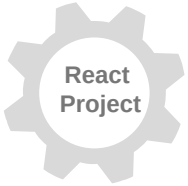
- Vite is a build tool that is designed to make development of modern web projects particularly those using React and Vue.js

React Project

4. Create new React project in VS Code: **npm create vite@latest <project name> -- -- template react**

```
PS C:\Users\65852\Desktop\FirstReact> npm create vite@latest firstReactApp -- --template react
√ Package name: ... firstreactapp

Scaffolding project in C:\Users\65852\Desktop\FirstReact\firstReactApp...

Done. Now run:

  cd firstReactApp
  npm install
  npm run dev
```

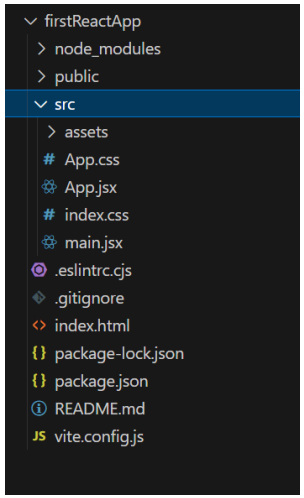
Download and run React App



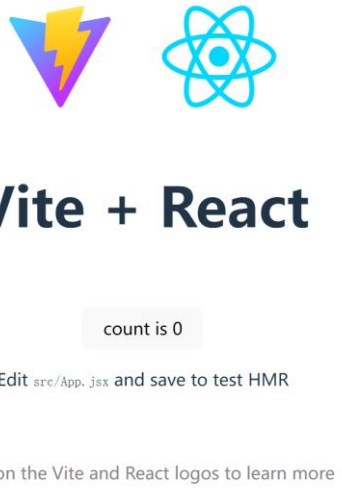
4. Create new React project in VS Code: **npm create vite@latest <project name> -- -- template react**

```
PS C:\Users\65852\Desktop\FirstReact> npm create vite@latest firstReactApp -- --template react
√ Package name: ... firstreactapp

Scaffolding project in C:\Users\65852\Desktop\FirstReact\firstReactApp...

Done. Now run:

  cd firstReactApp
  npm install
  npm run dev
```

5. Type **cd firstReactApp** -> **npm i** -> **npm run dev**




```
VITE v5.2.10  ready in 454 ms

➜  Local:   http://localhost:5173/
➜  Network: use --host to expose
➜  press h + enter to show help
```



Case Study 03: Form Validation Using React.js

01 FormValidationExample.jsx

Create form contents in .jsx file

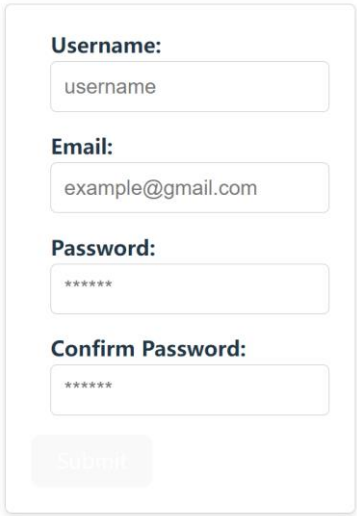
```jsx
import { useState } from 'react'

import './App.css'


const FormValidationExample = () => {

  const [formData, setFormData] = useState({
    username: '',
    email: '',
    password: '',
    confirmPassword: ''
  })
```

App.css

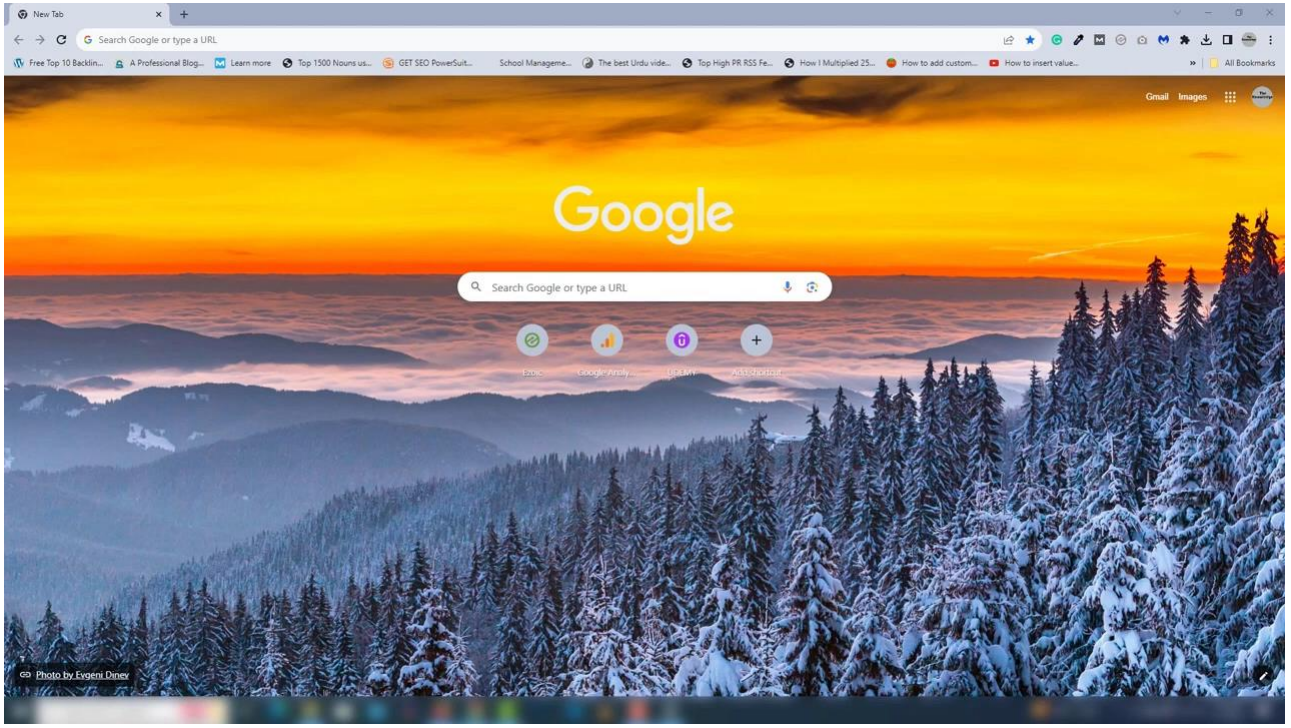
Add sheet styles (.css file) on the form

```css

/* Styles for the form container */
* {
  box-sizing: border-box;
  margin-left: auto;margin-right: auto;

}
form {
  max-width: 400px;
  margin-left: 0px;
```

02 Vite+React on Browser



Installation of Node.js



Thanks

Lecturer: Dr. Hu Xiao

Email: xiao.hu@ntu.edu.sg