

2023-COS30049-Computing Technology Innovation Project

Workshop Guide

Note: It's crucial to modify the distribution of time according to your specific requirements. You might also need to devise your own PowerPoint presentation in line with the guidelines of the workshop.

Workshop 01

Blockchain, Bitcoin, and Ethereum

Objective: By the end of this workshop, students should have an understanding of the basic concepts of blockchain, Bitcoin, and Ethereum, and be able to answer questions and engage in discussions on these topics.

Workshop Structure:

1. Understand the project requirements (35 mins) :

You will need to read the project description below depending on the project you choose. By reading the project description, you need to discuss it with your fellow team members and compile a list of requirements out of it for prototyping later.

Requirements lists are lists developed to ensure that related work is trackable and manageable, with the main elements including the preparation of related work items, completion dates and responsible persons.

PROJECT DESCRIPTION:

1.1 Smart contract audit system (For cyber security students)

The Smart Contract Audit System aims to assist users in identifying potential vulnerabilities in smart contract code. The platform should provide a responsive website that allows users to upload their smart contract code. By utilizing existing smart contract static analysis tools, the platform can rapidly analyze the contract code and identify

potential issues. Users can view the detection results on the website, enabling them to promptly identify and address any vulnerabilities in their contracts.

Core Functional Requirements:

1. Users can submit their smart contracts for auditing, and the system should accept smart contract files in common formats like Solidity (.sol).
2. The system will employ static analysis techniques to assess smart contracts for potential vulnerabilities and security risks.
3. The system should display comprehensive audit reports on a dedicated page, emphasizing the identified vulnerabilities and their respective categories.
4. The system needs to provide corresponding suggestions for each vulnerability category.
5. Users should be able to retrieve previous reports.

1.2 Decentralized trading system (For software engineering students)

The Decentralized Trading System aims to provide a secure and user-friendly platform for peer-to-peer trading of digital assets without the need for intermediaries. The system's primary goal is to enable seamless, trustless, and transparent transactions, enhancing the overall trading experience for users within the blockchain ecosystem.

Note: in this system, students don't need to implement the payment gateway.

Core Functional Requirements:

1. Users can view digital assets available for trading.
2. All the listed digital assets information should be stored in the database.
3. The system should provide a search and filter functionality for users to discover specific assets of interest.
4. The website will implement smart contracts to act as escrow during the trading process, and smart contracts should ensure that assets are held securely until the trade is completed or canceled.

5. Users should have access to a transaction history to view their past trades.

1.3 Blockchain Transaction Information Visualization System (For data analysis students)

The Blockchain Transaction Information Visualization System aims to provide an interactive and intuitive platform for visualizing and exploring blockchain transaction data. The system's primary goal is to present complex blockchain information in a user-friendly manner, enabling users to gain valuable insights into transaction patterns, network activity, and smart contract interactions.

Core Functional Requirements:

1. Users can input a wallet address in the search bar to retrieve basic information about the address. The website will display details such as the wallet's balance and other relevant data related to the address.
2. Upon searching for a wallet address, the system will display a directed graph, where each node represents a wallet address, and each edge represents a transaction between connected addresses.
3. Users can interact with the graph and explore the next/previous hop of connected addresses (e.g., clicking on nodes to explore further transaction paths)
4. The website will present relevant detailed transaction information in a tabular format
5. All transaction data needs to be stored in a graph database for efficient retrieval and visualization.

2. Design your prototype (40 mins) :

Based on the list of requirements discussed earlier, you will need to design a prototype system that represents the basic functionality of the selected project. After the group has discussed the design, you can discuss it with the client (tutor) to make sure that the requirements and the prototype meet the functionality of the system. The prototype can be delivered with the following methods:

- sketches on paper
- hand-drawn sketches on an iPad
- design tool
 - Axure <https://www.axure.com/>
 - Sketch <https://www.sketch.com/>
 - Figma <https://www.figma.com/>
 - Powerpoint
 - Or other prototype tools

You will have 25 minutes on prototype designing and 15 minutes to discuss with the client.

3. Config React locally (20 mins)

- For Mac users

1. Install node.js ENV first

brew install node

OR

download the ENV from node.js web

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

If your laptop need to install brew, then

Follow the instruction on <https://brew.sh/>

/bin/bash -c "\$(curl -fsSL <https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh>)"

check the version of node.js by

```
node -v
(base) ~ % node -v
v19.8.1
```

2. Create react app

```
npm i -g create-react-app
create-react-app hello-react
```

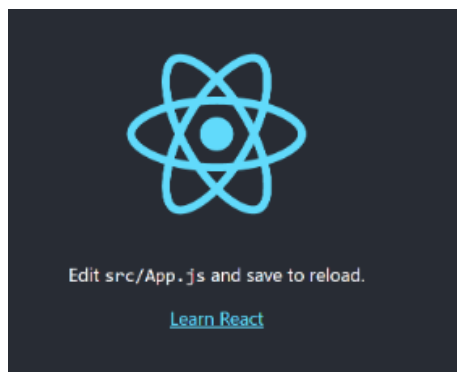
```
(base) ~ % npm i -g create-react-app
added 67 packages, and audited 68 packages in 14s

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

found 0 vulnerabilities
npm notice
npm notice New minor version of npm available! 7.5.3 -> 7.6.3
npm notice Changelog: https://github.com/npm/cli/releases/tag/v7.6.3
npm notice Run npm install -g npm@7.6.3 to update!
npm notice
```


3. Start your project

```
cd hello-react
npm start
```



- For Windows users

1. Download the node.js from
<https://nodejs.org/en/download>

node 

HOME | ABOUT | DOWNLOADS | DOCS | GET INVOLVED | SECURITY | CERTIFICATION | NEWS


Downloads

Latest LTS Version: 18.17.0 (includes npm 9.6.7)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.


LTS

Recommended For Most Users




Windows Installer

node-v18.17.0-x64.msi



macOS Installer

node-v18.17.0.pkg



Source Code

node-v18.17.0.tar.gz

Windows Installer (.msi)

Windows Binary (.zip)

macOS Installer (.pkg)

macOS Binary (.tar.gz)

Linux Binaries (x64)

Linux Binaries (ARM)

Source Code

32-bit	64-bit
32-bit	64-bit
64-bit / ARM64	
64-bit	ARM64
64-bit	
ARMv7	ARMv8
node-v18.17.0.tar.gz	

Additional Platforms

Docker Image

Linux on Power LE Systems

Linux on System z

AIX on Power Systems

Official Node.js Docker Image
64-bit
64-bit
64-bit

- Signed SHASUMS for release files ([How to verify](#))
- All download options

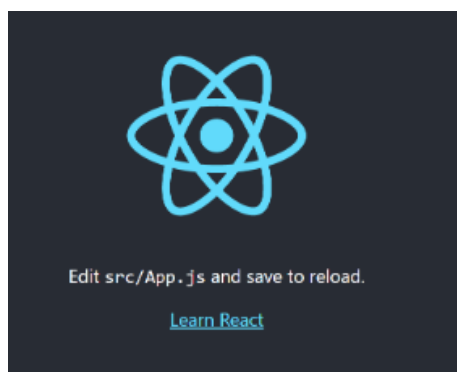
check the version of node.js by
node -v

2. Create react app

```
npm i -g create-react-app
create-react-app hello-react
```

3. Start your project

```
cd hello-react
npm start
```



Edit src/App.js and save to reload.

[Learn React](#)

4. HTML Tag and CSS in React (20 mins)

Online React playground

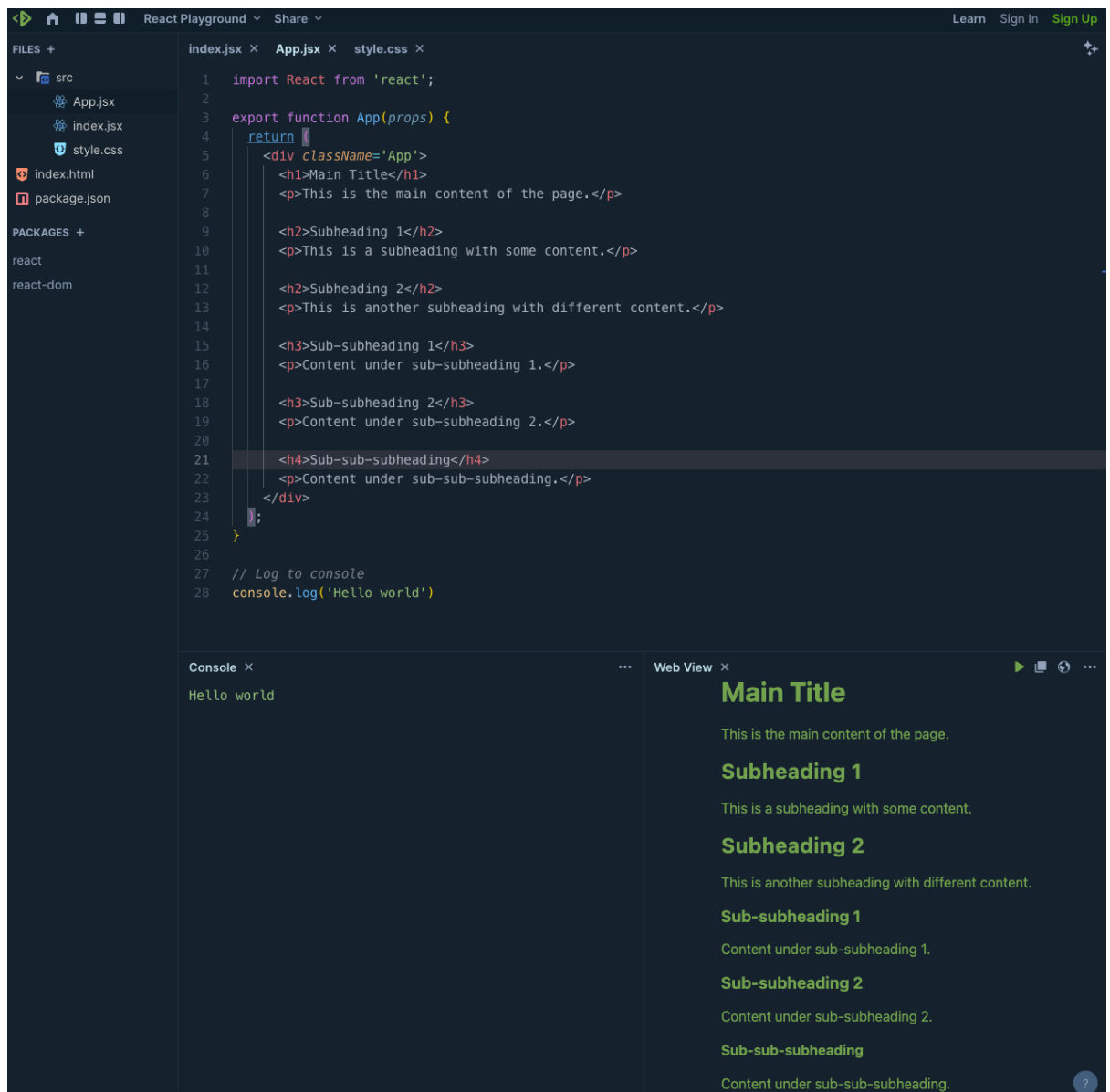
<https://playcode.io/react>

Since React.js is compatible with HTML tags, you can create web elements from HTML tags within the react framework. Here are some examples about:

1. Basic HTML tag
2. Comments inside React.js
3. Styling in React.js

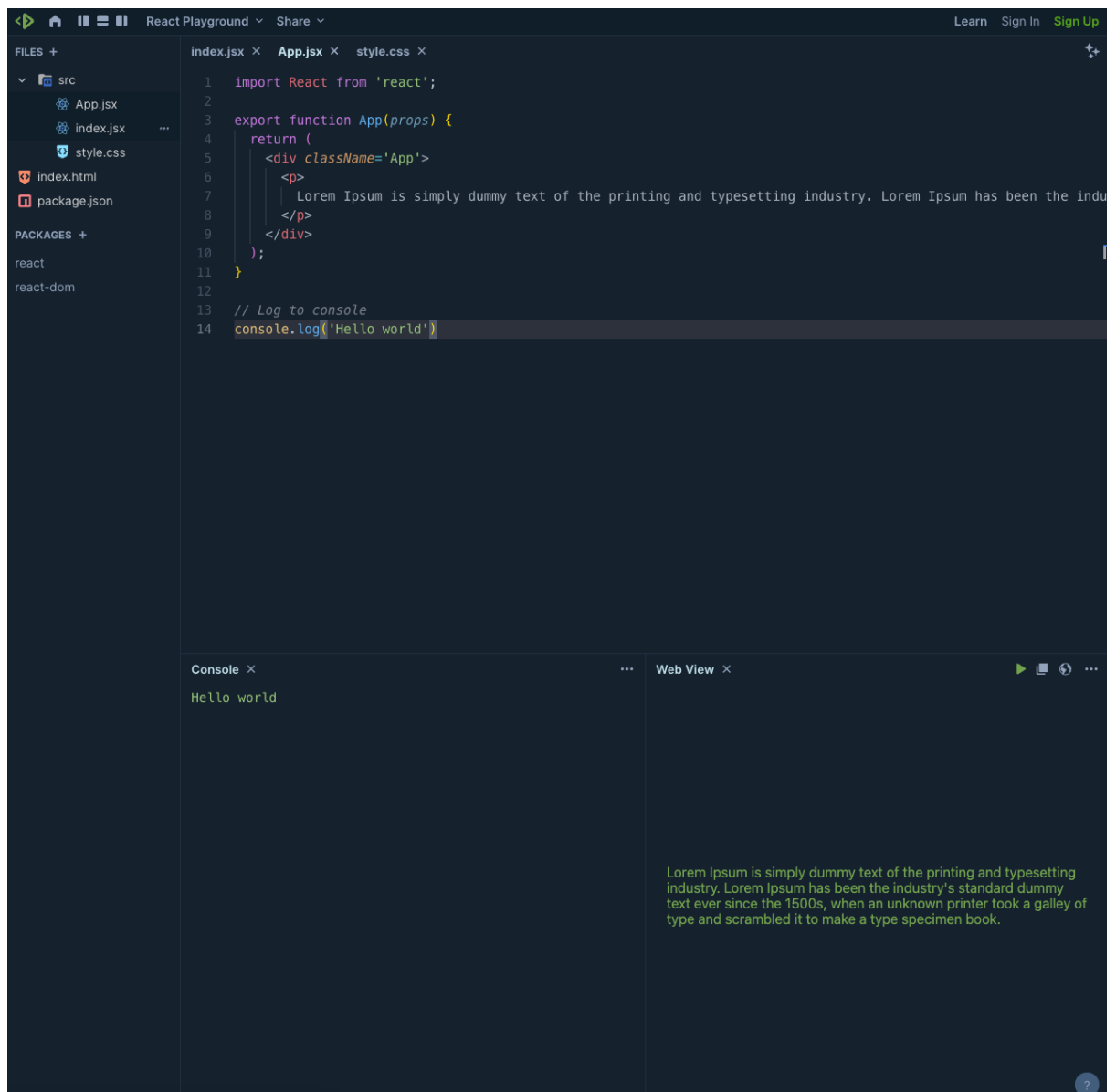
Heading Tag

- `<h1>`: The highest level heading on a webpage. It should be used for the main title of the page and is considered the most important heading on the page. Typically, there should be only one `<h1>` tag per page.
- `<h2>`: A second-level heading that is used to create subheadings under the main title. It represents a lower level of importance than `<h1>` but higher than subsequent heading tags.
- `<h3>` to `<h6>`: These are third to sixth-level headings, respectively. They are used to further divide and organize content under the main and subheadings, with `<h6>` being the least important.



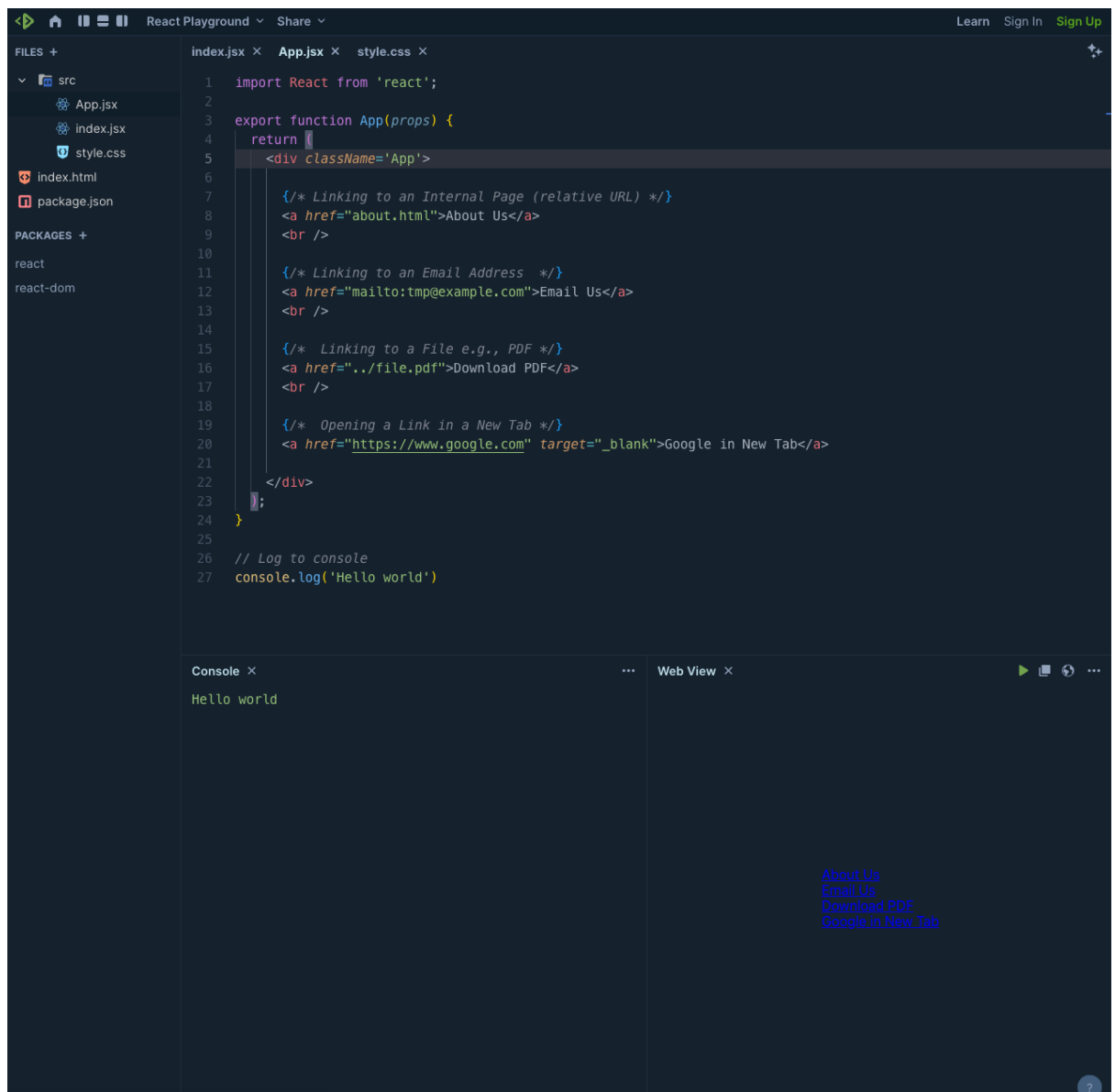
Paragraph Tag

The `<p>` element is a block-level element, which means it starts on a new line and creates a distinct block of content. When you use the `<p>` tag, the browser automatically adds vertical spacing before and after the paragraph, visually separating it from surrounding elements.



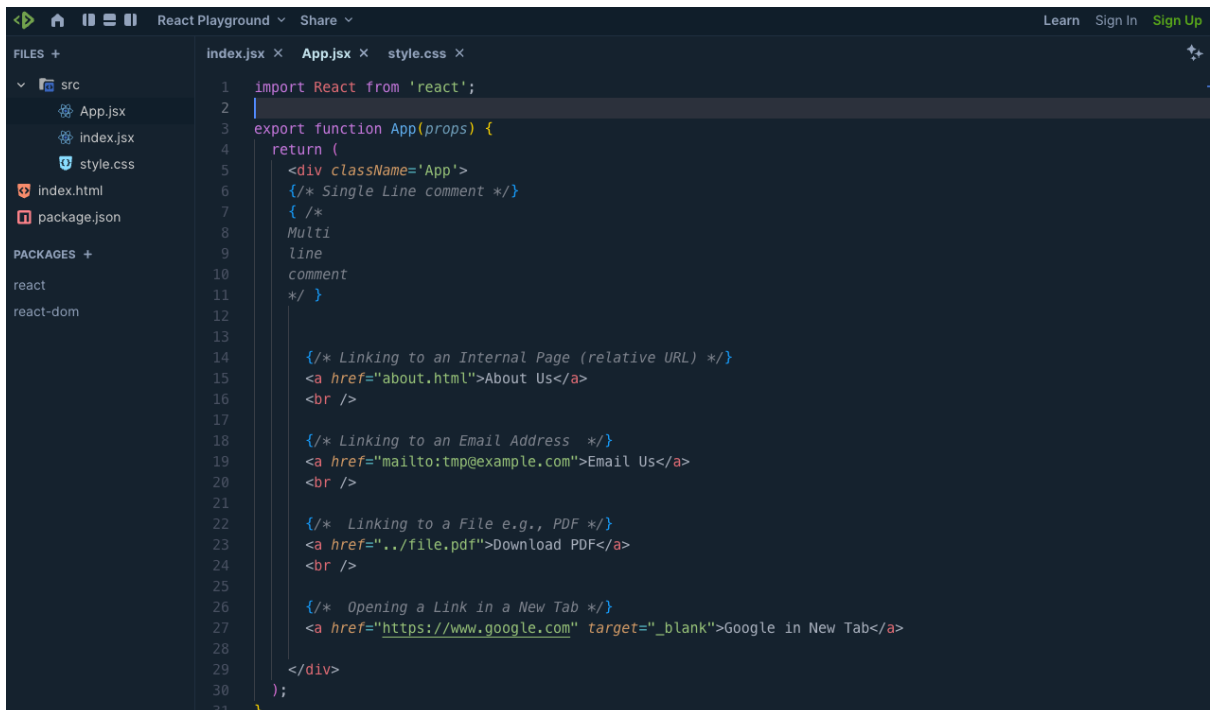
Link Tag

In HTML, links are created using the `<a>` (anchor) element, which stands for "anchor." The `<a>` element requires the `href` attribute, which specifies the destination URL or resource to which the link points.



Comments in React.js

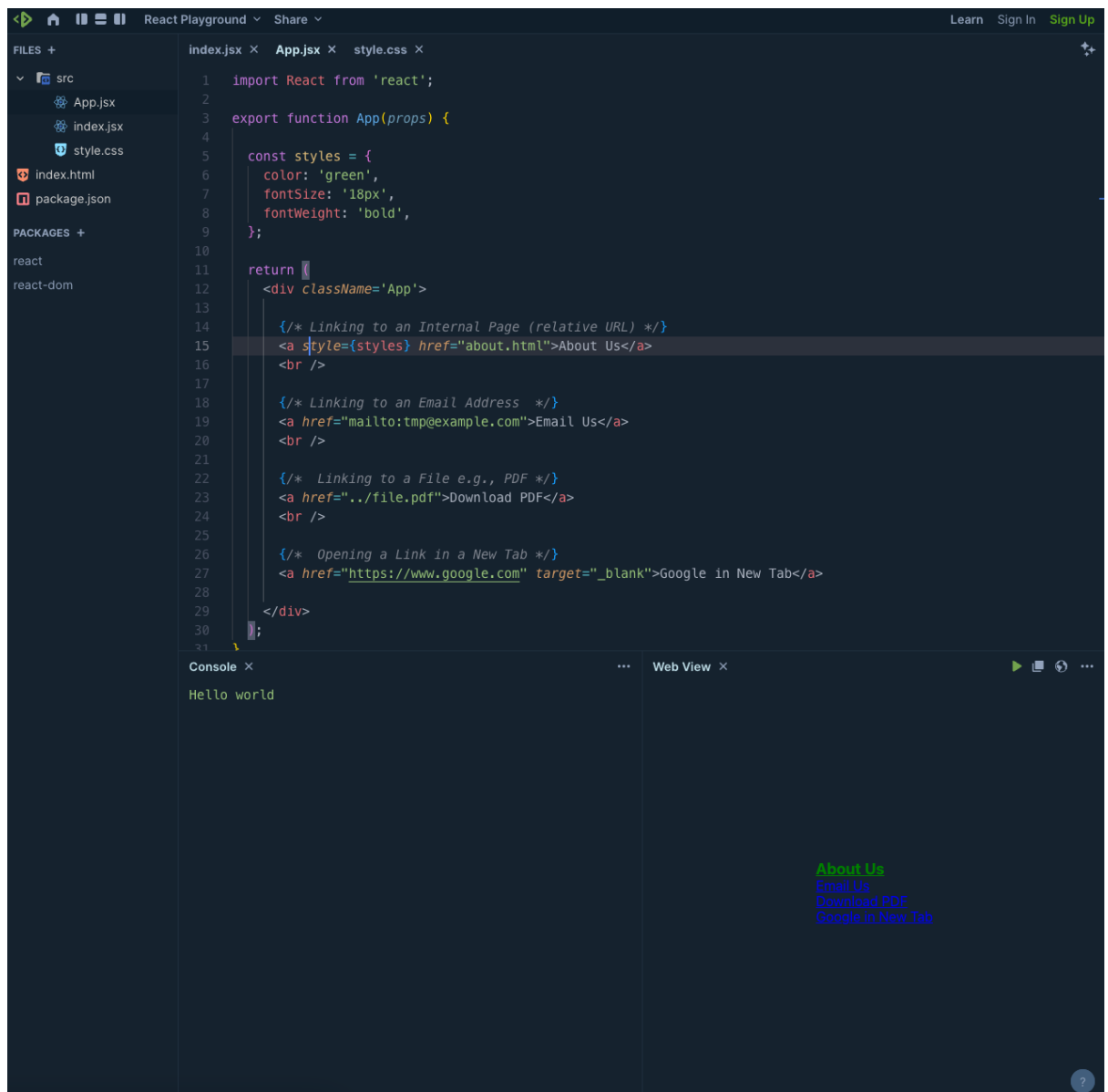
In React.js, you can use comment syntax to add comments. React will automatically ignore these comments, and they will not affect the rendering or functionality of the components.



```
1 import React from 'react';
2
3 export function App(props) {
4   return (
5     <div className='App'>
6       /* Single Line comment */
7       { /*
8        Multi
9        line
10        comment
11        */ }
12
13
14       /* Linking to an Internal Page (relative URL) */
15       <a href='about.html'>About Us</a>
16       <br />
17
18       /* Linking to an Email Address */
19       <a href='mailto:tmp@example.com'>Email Us</a>
20       <br />
21
22       /* Linking to a File e.g., PDF */
23       <a href='../file.pdf'>Download PDF</a>
24       <br />
25
26       /* Opening a Link in a New Tab */
27       <a href='https://www.google.com' target='_blank'>Google in New Tab</a>
28     </div>
29   );
30 }
```

Styling Your Elements

Inline Styling: You can add inline styles directly to JSX elements using the style attribute.



CSS Modules: CSS Modules is a technique that allows you to import and use CSS files directly in your components. It provides locally scoped CSS, avoiding class name conflicts.

React Playground

Learn Sign In Sign Up

FILES +

src

App.jsx

index.jsx

style.css

index.html

package.json

PACKAGES +

react

react-dom

index.jsx

App.jsx

style.css

```
1 body {
2   background: transparent; /* Make it white if you need */
3   padding: 0 24px;
4   margin: 0;
5   height: 100vh;
6   display: flex;
7   justify-content: center;
8   align-items: center;
9   font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans',
10 }
11
12 .App {
13   color: #72a24d;
14 }
15
16
17 .link{
18   color: #673137;
19 }
```

React Playground

Learn Sign In Sign Up

FILES +

src

App.jsx

index.jsx

style.css

index.html

package.json

PACKAGES +

react

react-dom

index.jsx

App.jsx

style.css

```
6   color: 'green',
7   fontSize: '18px',
8   fontWeight: 'bold',
9 };
10
11 return (
12   <div className='App'>
13     {
14       /* Linking to an Internal Page (relative URL) */
15       <a style={styles} href='about.html'>About Us</a>
16       <br />
17
18       {
19         /* Linking to an Email Address */
20         <a href='mailto:tmp@example.com'>Email Us</a>
21         <br />
22
23         {
24           /* Linking to a File e.g., PDF */
25           <a className='link' href='../file.pdf'>Download PDF</a>
26           <br />
27
28           {
29             /* Opening a Link in a New Tab */
30             <a href='https://www.google.com' target='_blank'>Google in New Tab</a>
31           }
32         }
33       }
34     </div>
35   );
36 }
37
38 // Log to console
39 console.log('Hello world')
```

Console

Hello world

Web View

About Us

Email Us

Download PDF

Google in New Tab

5. Reflection (5 mins)

Give students the opportunity to share what they learned, found interesting, or had difficulty understanding. Offer additional resources for them to learn more about React.js

Define Your Project Requirements: 11 Steps

<https://www.bairesdev.com/blog/define-your-project-requirements/>

React Playground

<https://playcode.io/react>

HTML Tag

https://www.w3schools.com/tags/tag_html.asp

React Documentation

<https://react.dev/learn/describing-the-ui>