CoGrammar

Welcome to this session:

Task Walkthrough - React Elements and Components

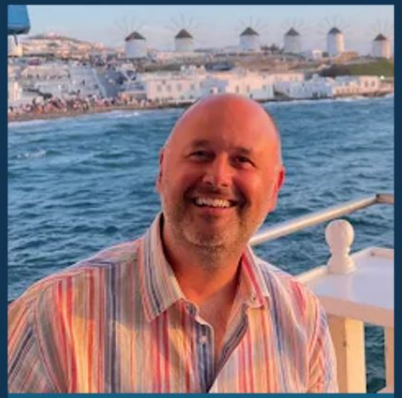
The session will start shortly...

Questions? Drop them in the chat.
We'll have dedicated moderators answering questions.

Safeguarding & Welfare

We are committed to all our students and staff feeling safe and happy; we want to make sure there is always someone you can turn to if you are worried about anything.

If you are feeling upset or unsafe, are worried about a friend, student or family member, or you feel like something isn't right, speak to our safeguarding team:



Ian Wyles
Designated Safeguarding Lead

Simone Botes

Nurhaan Snyman

Rafiq Manan

Ronald Munodawafa

Tevin Pitts

Scan to report a safeguarding concern

or email the Designated Safeguarding Lead:
Ian Wyles
safeguarding@hyperiondev.com

CoGrammar HyperionDev

Skills Bootcamp Cloud Web Development

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(Fundamental British Values: Mutual Respect and Tolerance)**
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Academic Sessions. You can submit these questions here: **Questions**

Skills Bootcamp Cloud Web Development

- For all **non-academic questions**, please submit a query: ***www.hyperiondev.com/support***
- **Report a safeguarding incident: *www.hyperiondev.com/safeguardreporting***
- We would love your feedback on lectures: ***Feedback on Lectures***
- If you are hearing impaired, please kindly use your computer's function through Google chrome to enable captions.

Learning Outcomes

- **Set up and initialize a React project** using Vite as the build tool and NPM for dependency management.
- **Understand and use JSX** to create dynamic, declarative components in React.
- **Create and style reusable React components** that display structured information.
- **Implement and display data** from JavaScript objects and arrays within JSX.

Lecture Overview

- ➔ Presentation of the Task
- ➔ Introduction to React
- ➔ Introduction to JSX
- ➔ React Components
- ➔ Task Walkthrough

React Task

Imagine you're designing the ultimate user profile for a new social media app called "ProfileVerse"—a space where users showcase their personality, activities, and style. 🌟 Your mission is to create a Dynamic User Profile Page using React and JSX that displays user details like their name, email, profile picture, and recent activities.

Your profile should display the user's information in an organized and visually appealing way, making use of lists, images, and headings. And don't forget to sprinkle in some CSS magic to give your page the personality it deserves!

What is React primarily used for?

A. Building server-side applications
B. Creating interactive user interfaces
C. Managing databases
D. Styling web pages

What does JSX allow developers to do in React?

A. Manage server-side state
B. Handle database queries
C. Write HTML directly in JavaScript
D. Style components

React Introduction

- ReactJS is a **declarative, efficient, and flexible JavaScript library** for building **reusable UI components**.
- It is an **open-source, component-based front end library** responsible only for the **view layer of the application**.
- A ReactJS application is made up of multiple **components**, each component responsible for outputting a **small, reusable piece of code**.
- The components are the heart of all React applications. These components can be **nested with other components** to allow complex applications to be built of simple building blocks.

Rendering in React

- React renders HTML to the web page by using a function called **createRoot()** and its method **render()**.
- The **createRoot()** function takes one argument, an HTML element. The purpose of the function is to define the HTML element where a React component should be displayed.
- The **render()** method is then called to define the React component that should be rendered.

Rendering in React

```
const rootElement = document.getElementById('root');
const root = createRoot(rootElement);

root.render(
  <StrictMode>
    <App />
  </StrictMode>
);
```

React JSX

- **JSX(JavaScript Extension), is a React extension** which allows writing JavaScript code that looks like HTML.
- In other words, JSX is an **HTML-like syntax** used by React that extends ECMAScript so that HTML-like syntax can co-exist with JavaScript/React code.
- JSX allows you to **write HTML/XML-like structures** (e.g., DOM-like tree structures) in the **same file where you write JavaScript code**, then preprocessor will transform these expressions into actual JavaScript code.

```
<div>
  <h1>Hello</h1>
```

Why use JSX?

- It is **faster** than regular JavaScript because it **performs optimisation** while translating the code to JavaScript.
- Instead of separating technologies by putting markup and logic in separate files, React uses components that **contain both technologies**.
- It is **type-safe**, and most of the errors can be found at compilation time.
- It makes it **easier to create templates**.

JSX Attributes

- JSX assign attributes with the HTML elements same as regular HTML.
- JSX uses **camelcase naming convention** for attributes rather than standard naming convention of HTML such as a class in HTML becomes className in JSX because the class is the reserved keyword in JavaScript.

```
<div>
  <h1 className="hello">The Final Countdown</h1>
</div>
```

JSX Attributes

- In JSX, we can specify attribute values in two ways:
  - **As String Literals**: We can specify the values of attributes in double quotes.

```
<h2 className="firstAttribute">Hello Zahir</h2>
```

  - **As Expressions**: We can specify the values of attributes as expressions using curly braces **{}**

```
<h2 className={varName}>Hello Zahir</h2>
```

JSX Comments

- JSX allows us to use comments that begin with /* and ends with */ and wrapping them in curly braces {}.

```
{/* This is a comment in JSX */}
```

JSX Styling

- To set inline styles, you need to use camelCase syntax.

```
export default function App() {
  let myStyle = {
    fontSize: 80,
    fontFamily: 'Courier',
    color: '#003300',
  };

  return (
    <div>
      <h2 style={myStyle}>Hello Zahir</h2>
    </div>
  );
}
```

React Components

- A Component is considered as the **core building blocks** of a React application.
- It makes the task of building UIs much easier.
- Each component exists in the **same space**, but they **work independently** from one another and **merge all in a parent component**, which will be the **final UI of your application**.
- All React components have their own structure, methods as well as APIs. They can be reusable as per your need.

Functional Components

```
import React from 'react';
function WelcomeMessage(props) {
  return <h1>Welcome to the , {props.name}</h1>;
}

export default WelcomeMessage;
```

Props

- Components can be passed props, which stands for properties.
- Props are like function arguments, and you send them into the component as attributes.

```
<Welcome name="Zahir"></Welcome>
```

CoGrammar

React Task

Imagine you're designing the ultimate user profile for a new social media app called "ProfileVerse"—a space where users showcase their personality, activities, and style. 🌟 Your mission is to create a Dynamic User Profile Page using React and JSX that displays user details like their name, email, profile picture, and recent activities.

Your profile should display the user's information in an organized and visually appealing way, making use of lists, images, and headings. And don't forget to sprinkle in some CSS magic to give your page the personality it deserves!

Which command initializes a React project using Vite?

A. npm install react
B. npm create vite@latest my-app -- --template react
C. npx create-react-app my-app
D. npm init react-app

In React, what is a component?

A. A function or class that returns a React element
B. A CSS file used for styling
C. A server-side API
D. A database schema

CoGrammar

Q & A SECTION

Please use this time to ask any questions relating to the topic, should you have any.

Thank you for attending

CoGrammar

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education