Welcome to CoGrammar

WD Week 8 Tutorial

The session will start shortly...

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

CoGrammar

Full Stack Web Development Session Housekeeping

- 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**

Full Stack Web Development Session Housekeeping cont.

- 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**

Lecture Overview

- Recap of State Management
- Recap of Event Handling

CoGrammar

useState Hook

- state: Represents the current value of the state variable.
- setState: A function used to update the state variable and trigger re-rendering.

```
let [fullName, setFullName] = useState('Clark Kent');
```

CoGrammar

Example: Counter Component

```
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>
      <button onClick={() => setCount(count - 1)}>Decrement</button>
    </div>
  );
}

export default Counter;
```

CoGrammar

Prop drilling: Examples

```
import React from 'react';
import ParentComponent from './Parent';

function GrandParentComponent() {
  const userData = { name: 'John', age: 30 };

  return <ParentComponent userData={userData} />;
}

export default GrandParentComponent;
```

CoGrammar

Prop drilling: Examples

```
import React from 'react';
import ChildComponent from './Child';

function ParentComponent({ userData }) {
  return <ChildComponent userData={userData} />;
}

export default ParentComponent;
```

CoGrammar

Prop drilling: Examples

```
import React from 'react';
import User from './User';

function ChildComponent({ userData }) {
  return <User userData={userData} />;
}

export default ChildComponent;
```

CoGrammar

Prop drilling: Examples

```
import React from 'react';

function User({ userData }) {
  return (
    <div>
      <h2>User Details</h2>
      <p>Name: {userData.name}</p>
      <p>Age: {userData.age}</p>
    </div>
  );
}

export default User;
```

CoGrammar

Example: Using React Context API

```
import React, { createContext, useContext } from 'react';

const UserContext = createContext({});

export function useUserDate() {
  return useContext(UserContext);
}

export default UserContext;
```

CoGrammar

Example: Using React Context API

```
import React from 'react';
import ParentComponent from './Parent';
import UserContext from './UserContext';

function GrandParentComponent() {
  const userData = { name: 'John', age: 30 };

  return (
    <UserContext.Provider value={userData}>
      <ParentComponent />
    </UserContext.Provider>
  );
}

export default GrandParentComponent;
```

CoGrammar

Example: Using React Context API

```
import React from 'react';
import { useUserData } from './UserContext';

function User({ userData }) {
  let data = useUserData();
  return (
    <div>
      <h2>User Details</h2>
      <p>Name: {data.name}</p>
      <p>Age: {data.age}</p>
    </div>
  );
}

export default User;
```

CoGrammar

Adding event handlers

- To add an event handler, you will first define a function and then pass it as a prop to the appropriate JSX tag.

```
export default function Button() {
  function handleClick() {
    alert('You clicked me!');
  }

  return (
    <button onClick={handleClick}>
      Click me
    </button>
  );
}
```

CoGrammar

Adding event handlers

- Alternatively, you can define an event handler inline in the JSX:

```
<button onClick={function handleClick() {
  alert('You clicked me!');
}}>
```

- Or, more concisely, using an arrow function:

```
<button onClick={() => {
  alert('You clicked me!');
}}>
```

CoGrammar

Passing event handlers as props

- Often you'll want the parent component to specify a child's event handler.
- To do this, pass a prop the component receives from its parent as the event handler.

```
import React from 'react';
function SuperButton({ onClick }) {
  return <button onClick={onClick}>I am a Super Button!</button>;
}
```

```
export default function App() {
  return (
    <div>
      <SuperButton
        onClick={() => {
          alert('Haaaaayyyyyy!!!');
        }}
      />
```

CoGrammar

Event Object

```
import React from 'react';

const MyComponent = () => {
  const handleClick = (event) => {
    console.log('Button clicked!');
    console.log('Event:', event);
    console.log('Target Element:', event.target);
    console.log('Event Type:', event.type);
    console.log('Current Target:', event.currentTarget);
  };

  return <button onClick={handleClick}>Click Me</button>;
};
```

CoGrammar

onClick Event

- The onClick event is triggered when a user clicks on an element. It's widely used for handling button clicks, link clicks, or any other click interactions.

```
import React from 'react';

const MyComponent = () => {
  const handleClick = () => {
    console.log('Button clicked!');
  };

  return <button onClick={handleClick}>Click Me</button>;
};
```

CoGrammar

onChange Event

- The onChange event is triggered when the value of an input element changes. It's commonly used for handling user input in forms.

```
const MyComponent = () => {
  const [value, setValue] = useState('');

  const handleChange = (event) => {
    console.log('Input changed!');
    console.log('New value:', event.target.value);
    setValue(event.target.value);
  };

  return <input type="text" value={value} onChange={handleChange} />;
};
```

CoGrammar

onSubmit Event

- The onSubmit event is triggered when a form is submitted. It's essential for form validation and submission.

```
const MyComponent = () => {
  const handleSubmit = (event) => {
    console.log('Form submitted!');
  };

  return (
    <form onSubmit={handleSubmit}>
      <button type="submit">Submit</button>
    </form>
  );
};
```

CoGrammar

preventDefault

- The preventDefault method is called within event handlers to prevent the default behavior of an event. It's commonly used to prevent form submission or link navigation.

```
export default function App() {
  const handleClick = (e) => {
    e.preventDefault();
    console.log('Hayyyyyyy!!!');
  };

  return (
    <div>
      <a href="/items" onClick={handleClick}>
        Click me
      </a>
    </div>
  );
}
```

CoGrammar

Passing Arguments to Event Handlers

- You can pass additional arguments to event handlers using arrow functions.

```
export default function App() {
  const handleClick = (arg) => {
    console.log('Clicked with argument:', arg);
  };

  return <button onClick={() => handleClick('Hello')}>Click Me</button>;
}
```

CoGrammar

Event propagation

```
export default function Toolbar() {
  return (
    <div className="Toolbar" onClick={() => {
      alert('You clicked on the toolbar!');
    }}>
      <button onClick={(e) => {

          alert('Playing!');
    }}>
        Play Movie
      </button>
    </div>
  );
}
```

CoGrammar

Stopping propagation

- That event object also lets you stop the propagation.

```
export default function Toolbar() {
  return (
    <div className="Toolbar" onClick={() => {
      alert('You clicked on the toolbar!');
    }}>
      <button onClick={(e) => {
          e.stopPropagation();
          alert('Playing!');
    }}>
        Play Movie
      </button>
    </div>
  );
}
```

CoGrammar

Questions and Answers

CoGrammar

Thank you for attending

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar