# Let's dive into React.

## React Topics

1. JSX
2. Components
3. State
4. Props
5. Lifecycle Methods
6. Hooks
7. Context API
8. React Router
9. Redux

## JSX Basics

JSX is a syntax extension for JavaScript that allows you to write HTML-like code in your JavaScript files.

***Example***

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

***Key Concepts***

1. JSX tags
2. Attributes
3. Children
4. Expressions

***Problems***

1. Write a JSX element that displays "Hello, World!" in an h1 tag.
2. Create a JSX element with a div tag and two child elements: an h2 and a p.
3. Use JSX expressions to display the result of 2 + 2 in a p tag.

## Functional Components

Functional components are simple, reusable functions that return JSX.

***Example***

```
function Hello() {
  return <h1>Hello, World!</h1>;
}
```

***Key Concepts***
1. Function signature
2. Return statement
3. Props
4. State (using hooks)

***Problems***
1. Create a functional component that displays "Goodbye, World!".
2. Write a functional component that takes a name prop and displays "Hello, {name}!".
3. Use the useState hook to create a counter component.

## Class Components
Class components are more complex, reusable classes that extend React.Component.

***Example***
```
class Counter extends React.Component {
  state = { count: 0 };

  render() {
    return (
      <div>
        <p>Count: {this.state.count}</p>
        <button onClick={() => this.setState({ count: this.state.count + 1 })}>
          Increment
        </button>
      </div>
    );
  }
}
```

***Key Concepts***
1. Class definition
2. State
3. Lifecycle methods (e.g., componentDidMount())
4. Event handling
5. Binding

***Problems***
1. Create a class component that displays a clock with the current time.
2. Write a class component that takes a name prop and displays "Hello, {name}!".
3. Use the componentDidMount() lifecycle method to fetch data from an API.

## State Basics
State is an object that stores data that changes over time.

***Example***
```
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>
  );
}
```

***Key Concepts***
1. useState() hook
2. State object
3. Update functions
4. Initial state

***Problems***
1. Create a state variable to store a user's name.
2. Write a function to update the state variable.
3. Use state to display a list of todo items.

***Solutions***
*Problem 1 Solution*

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

function UserName() {
  const [name, setName] = useState('');

  return (
    <div>
      <p>Name: {name}</p>
      <input type="text" value={name} onChange={(e) =>
setName(e.target.value)} />
    </div>
  );
}
```

*Problem 2 Solution*

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

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

  const increment = () => {
    setCount(count + 1);
  };

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

*Problem 3 Solution*

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

function TodoList() {
  const [todos, setTodos] = useState(['Buy milk', 'Walk dog']);

  return (
    <ul>
      {todos.map((todo, index) => (
        <li key={index}>{todo}</li>
      ))}
    </ul>
  );
}
```

**Props Basics**

Props (short for "properties") are how components communicate with each other.

***Example***

```
import React from 'react';

function Hello({ name }) {
  return <h1>Hello, {name}!</h1>;
}

function App() {
  return <Hello name="Alice" />;
}
```

***Key Concepts***

1. Props object
2. Prop types
3. Default props
4. Destructuring props

***Problems***

1. Create a component that takes a title prop and displays it in an h1.
2. Write a component that takes a list of items prop and displays them as li.
3. Use propTypes to validate prop types.

***Solutions***

*Problem 1 Solution*

```
import React from 'react';

function Title({ title }) {
  return <h1>{title}</h1>;
}
```

*Problem 2 Solution*

```
import React from 'react';

function ItemList({ items }) {
  return (
    <ul>
      {items.map((item, index) => (
        <li key={index}>{item}</li>
      ))}
    </ul>
  );
}
```

*Problem 3 Solution*

```
import React from 'react';
import PropTypes from 'prop-types';

function Title({ title }) {
  return <h1>{title}</h1>;
}

Title.propTypes = {
  title: PropTypes.string.isRequired,
};
```