Certainly! Let's explore some key features of React along with code examples for each:

1. JSX (JavaScript XML)

JSX is a syntax extension for JavaScript that allows you to write HTML-like code within JavaScript. It simplifies the process of creating React elements and makes the code more readable.

Code Example 1: JSX Syntax
```jsx
import React from 'react';

const App = () => {
  return (
    <div>
      <h1>Hello, JSX!</h1>
      <p>This is JSX syntax in action.</p>
    </div>
  );
};

export default App;
```

In this example, JSX is used to define a React component (`App`). HTML-like tags such as `<div>`, `<h1>`, and `<p>` are used to describe the UI structure.

Code Example 2: JSX Expressions
```jsx
import React from 'react';

const name = 'John Doe';
const greet = <p>Hello, {name}!</p>;

const App = () => {
  return (
    <div>
      {greet}
    </div>
  );
};

export default App;
```

Here, JSX allows embedding JavaScript expressions within curly braces `{}`. In this case, the `name` variable is inserted into the JSX expression to dynamically generate a greeting message.

2. Component-Based Architecture

React promotes a component-based architecture where UIs are built from reusable and composable components. This helps in encapsulating the UI logic and makes the codebase easier to maintain and scale.

Code Example 1: Functional Component
```jsx
import React from 'react';

const Hello = () => {
  return <h1>Hello, React!</h1>;
};

export default Hello;
```

In this example, a simple functional component named `Hello` is defined. It returns JSX that renders a greeting message.

Code Example 2: Class Component
```jsx
import React, { Component } from 'react';

class Greeting extends Component {
  render() {
    return <h1>Welcome, {this.props.name}!</h1>;
  }
}

export default Greeting;
```

Here, a class component named `Greeting` is defined. It extends the `Component` class from React and implements a `render()` method to return JSX. This component accepts a `name` prop to customize the greeting message.

These examples illustrate some of the key features of React, including JSX for defining UI components and a component-based architecture for building reusable and maintainable UIs.