React Components - Theory Summary

Explain React Components

Components are the building blocks of any React application. They encapsulate logic and UI together, and can be reused across different parts of an application.

Components help divide the UI into independent, reusable pieces, and think about each piece in isolation.

Differences Between Components and JavaScript Functions

- React components return JSX (HTML-like syntax), JavaScript functions return values.
- React components can manage state and lifecycle (especially class components).
- React components can trigger UI re-renders; JavaScript functions cannot.

Types of Components

1. Class Components
2. Function Components

Both are used to define React components, but function components are more commonly used in modern React with Hooks.

Explain Class Component

A class component is a JavaScript ES6 class that extends React.Component and must contain a render() method. It can also have a constructor and lifecycle methods.

Example:

```
class MyComponent extends React.Component {
  render() {
    return <h1>Hello from Class Component</h1>;
  }
}
```

Explain Function Component

A function component is a simpler way to write components using a JavaScript function.

Example:

```
function MyComponent() {
  return <h1>Hello from Function Component</h1>;
}
```

Define Component Constructor

The constructor is a method in class components that is called before the component is mounted. It is used to initialize state or bind event handlers.

Example:

```
constructor(props) {
  super(props);
  this.state = { key: 'value' };
}
```

Define render() Function

The render() method is required in every class component. It returns the JSX that defines the UI of the component.

React calls render() whenever it needs to update the DOM.