React Components - Summary

## Explain React components

React components are the building blocks of a React application. They encapsulate logic, structure (HTML), and styling (CSS) into reusable pieces. Components can manage their own state and can be composed to build complex user interfaces.

## Identify the differences between components and JavaScript functions

While both React components and JavaScript functions can contain logic, React components specifically return JSX (a syntax extension that looks like HTML) and are used to render UI elements. Components can maintain internal state and lifecycle methods (in class components), while regular JavaScript functions cannot.

## Identify the types of components

There are two main types of React components:  
1. Class Components  
2. Function Components

## Explain class component

Class components are ES6 classes that extend from React.Component. They must include a render() method and can hold local state and lifecycle methods. They are useful when more functionality like state or lifecycle control is needed.

## Explain function component

Function components are simple JavaScript functions that accept props as arguments and return JSX. With the introduction of Hooks (like useState and useEffect), function components can now handle state and side effects, making them as powerful as class components.

## Define component constructor

In class components, the constructor is a special function used for initializing state and binding event handlers. It is called before the component is mounted. The constructor typically calls super(props) to access 'this.props' inside the constructor.

## Define render() function

The render() function is a required method in class components. It returns the JSX to be rendered to the DOM. It must be a pure function, meaning it does not modify component state or interact with the browser directly.