# React Basics

React is a JavaScript library for building fast, interactive user interfaces. It follows a component-based architecture and uses a declarative approach to describe the UI.

## 1. Structure of a React App

A typical React application has the following structure:
- index.html (inside /public): The single HTML file where the app is mounted.
- index.js (inside /src): The entry point for the JavaScript code, rendering the root component.
- App.js: The main application component.
- Components folder: Contains reusable UI components.

## 2. Components

Components are the building blocks of a React application. They can be:
- Functional Components: Simple functions that return JSX.
- Class Components: ES6 classes that extend React.Component (less common now).

## 3. Props

Props (short for properties) allow data to be passed from a parent component to a child component.
They are read-only and help make components reusable.

Example:

```
function Welcome(props) {
return Hello, {props.name};
}
```

## 4. State and useState

State is data that can change over time and trigger re-renders when updated.
In functional components, the useState hook is used to manage state.

Example:

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

function Counter() {
const [count, setCount] = useState(0);
return (
Count: {count}
setCount(count + 1)}>Increase

);
}
```

Understanding React's basic concepts—structure, components, props, and state—is essential to start building interactive and maintainable user interfaces.