

# React Components



## Introduction to Components

The React application contains a root component in which other subcomponents are included; for example - in a single page, the application is subdivided into three parts - Header, Main Content, and Footer. So, a single App Component has three subcomponents - Header Component, Main Content Component, and Footer Component.



## Reusable Code

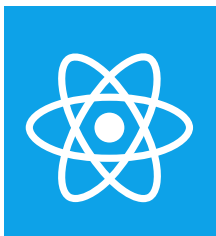
Components also make it possible to write reusable code.



### For Example:

You could have the component for an article on React; the same component can be reused for articles on Angular or Vue by simply passing the right data into the article component for enterprise applications; the ability to reuse code is a huge plus point

React



Angular



Vue



How does a component translate a code in our application?  
The component code is usually placed in a JavaScript file.

For Example:

### **App component is placed in App.js**

the component is the code inside a .js file, but what does that code look like? That depends on the type of the component.

Components describe a part of the user interface and the building blocks of any react application. They are reusable and can be nested inside other components. so it is crucial to get a good understanding of the two types of the components

## Functional Components

Functional components are just JavaScript functions; they can optionally receive an object of properties which is referred to as props, and return HTML, which describes the UI.

Now, the HTML is something known as JSX but for the sake of understanding from a beginner's point of view, let's just call it HTML.

So a functional component is a JavaScript function that accepts an input of properties and returns HTML that describes the UI.

## Class Components

Class components are ES6 classes similar to a functional component a class component also can optionally receive props as input and return HTML; apart from the props, a class component can also maintain a private internal state; in simpler words, it can maintain some information which is private to that component and use that information to describe the user interface.

## Introduction to props

Props Short for properties is the optional input that your component can accept; it also allows the component to be dynamic.

props are just an object that contains the attributes and their values which have been passed from the parent component

props are immutable, or in simpler words, their value cannot be changed in our example

## Conditional Rendering of Components

When building react applications, you may often need to show or hide some HTML based On a certain condition

Conditional rendering in React works the same way conditions work in JavaScript  
We have four different approaches, and we will take a detailed look at each one of them

1. if/else
2. Element variables
3. Ternary conditional operator
4. Short circuit operator

## How to merge components into one page

In React, we can only render one element. Even if we have multiple elements to render, there can only be a single root element.

If we want to render two or more elements, we have to wrap them in another element or component.

Commonly, the element used for this is a `<div>` tag.

## What will we be doing in class?

We will be creating an application:

## Upcoming Class Teaser:

- Class-based component lifecycle methods
- Introduction to state ( functional components )
- Handling events

