- For a web browser to read a JSX file, the file needs to be transformed into a regular JavaScript object. For this, we use Babel
- **State** : The state is a built-in React object that is used to contain data or information about the component. The state in a component can change over time, and whenever it changes, the component re-renders.
- **Props**: Props are short for Properties. It is a React built-in object that stores the value of attributes of a tag (that works similarly to HTML attributes). Props provide a way to pass data from one component to another component. Props are passed to the component in the same way as arguments are passed in a function.
  Props can hold any type of data, including Strings, numbers, arrays, objects, functions or even other react components.

```
// ParentComponent.js
import React from 'react';
import ChildComponent from './ChildComponent';

function ParentComponent() {
 return (
  <div>
   <ChildComponent name="John" age={30} />
  </div>
 );
export default ParentComponent;
// ChildComponent.js//inside the component, we can access the passed props through the 'props' obj.
import React from 'react';
function ChildComponent(props) {
 return (
  <div>
   <p>Name: {props.name}</p>
   <p>Age: {props.age}</p>
  </div>
 );
}
export default ChildComponent;
```

- **What are the differences between state and props?**
  State- Holds information about the components, mutable, can be changed
  Prop- Allows to pass data from one component to other components as an argument, immutable, cannot be changed
- **JSX-** SX stands for JavaScript XML. It's a syntax extension for JavaScript that allows developers to write HTML-like code within JavaScript files
  JSX is not a separate language; it's a syntax extension that gets transformed into regular JavaScript code by tools like Babel during the build process. Ultimately, JSX expressions are compiled into React.createElement() function calls, which create React elements in the virtual DOM

- **Bundler**: A bundler is a tool used in web development to combine multiple files, such as JavaScript, CSS, and images, into a single file or set of files for efficient delivery to the browser.
- **React:**
  React is an open-source JavaScript library for building user interfaces, particularly for single-page applications where the user interface needs to be dynamic and responsive. It was developed and is maintained by Facebook. React allows developers to create reusable UI components, making it easier to manage and update complex user interfaces. It employs a virtual DOM (Document Object Model) to optimize rendering performance by efficiently updating only the parts of the page that need to change. React is commonly used in conjunction with other libraries and frameworks to build modern, interactive web applications.
- **Library:** You call a library when you need its specific functionality. You have control over your code, and you decide when to use the library.
- **Framework:** The framework controls the flow of the application. You use a framework to build your application, and it calls your code when needed. The framework dictates the structure and flow of your program.

  In essence, with a library, you are in control, and you call the shots. With a framework, the control is inverted - the framework is in control, and you fill in the blanks.
- **React DOM vs react native:**
  React is a core fundamental library. React dom is react's implementation on web
  React DOM: Used for building web applications that run in browsers.
  React Native: Used for building mobile applications (iOS and Android) using React principles.
- **npx:**
  npx is a package runner tool that comes with Node.js. It is used to execute packages directly from the npm registry. Instead of globally installing a package, you can use npx to run it without installing it permanently. This is particularly useful for running tools or scripts for one-time use or for packages that you don't want to install globally.
- **Creating react app:**
  npx create-react-app 01basics- here is npx is package executor & create-react-app is utility i.e., software through which we can create our project. After running this command delete all files from src folder except index.js, app.js
- **Package.json:** In a package.json file, the "scripts" section is used to define various commands that can be executed using npm or yarn. Two common scripts are:

  "start": This script is typically used to launch your application. For example, if it's a web application, it might start a development server. -→ npm run start
  "build": This script is often used to create a production-ready version of your application. After running it, it gives the separate build folder containing all static files(html,css, js). -→ npm run build
- **Hooks:** hooks in React are special functions that allow functional components to do things that were previously possible only in class components.
- **reconciliation:** the algorithm react uses to differentiate one tree with another to determine which parts need to be changed
- **Fiber:** A reimplementation of React's core algorithm for rendering components. It allows for better control over the timing and priority of updates, making React more efficient and responsive.

- **Virtual DOM:** A technique in React where changes to the user interface are first made to a virtual representation of the DOM (Document Object Model) before being applied to the actual DOM. This helps optimize performance by reducing direct manipulation of the real DOM.
- **Axios**: Axios is a popular JavaScript library used for making HTTP requests from web browsers and Node.js applications. It is commonly used in front-end applications, particularly with frameworks like React or Vue.js, to interact with APIs and fetch data from servers.
- **React Lifecycle Methods:**

  Mounting Phase: These methods are called when an instance of a component is being created and inserted into the DOM.
  - constructor(): Initializes state and binds event handlers.
  - render(): Returns the JSX representing the component's UI.
  - componentDidMount(): Invoked after the component is mounted and rendered to the DOM. Used for initialization tasks and side effects.

  Updating Phase: These methods are called when a component is being re-rendered as a result of changes to props or state.
  - componentDidUpdate(): Invoked after the component is updated and re-rendered.

  Unmounting Phase:
  - componentWillUnmount(): Invoked immediately before a component is unmounted and removed from the DOM. Used for cleanup tasks and releasing resources.

Differences between Stateful and Stateless Components in React:

- **Stateful Components:** Also known as class components, stateful components manage their own state using the this.state object and can update it using this.setState(). They also contain lifecycle methods.
- **Stateless Components**: Also known as functional components, stateless components are pure functions that receive props as input and return JSX to describe the UI. They don't have their own state or lifecycle methods. With the introduction of React Hooks, functional components can now manage state and lifecycle methods using hooks like useState and useEffect.