

What is react

=>React is a javascript library for building user interfaces

=>React is a front end javascript library developed by facebook ,it is an open source javascript framework

=>It is used to build web and mobile applications

=>React js is a client side javascript library

React.js Alternatives

1.Angular:complete component based UI framework,packed with features.Uses Typescript ,can be overkill for smaller projects.

2.React.js: Learn and focused component based UI library.Certain features(e.g.routing)are added via community packages.

3.Vue.js:complete component based UI framework,includes most core features.A bit less popular than React and Angular

EXAMPLE

```
import React from 'react';
```

```
import ReactDOM from 'react-dom';
```

```
var name = "Learner";
```

```
var element = <h1>Hello, { name }.Welcome to TupleScale.</h1>;
```

```
ReactDOM.render(  
  element,  
  document.getElementById("root")  
)
```

OUTPUT:

Hello,Learner.Welcome to TupleScale.

What is JSX

JSX stands for JavaScript XML.

JSX allows us to write HTML in React.

JSX makes it easier to write and add HTML in React.

JSX is an extension to javascript. It is a template script where you will have the power of using HTML and Javascript together.

Here is a simple example of a JSX code.

```
const h1tag = "<h1>Hello</h1>";
```

EX:1

```
import React from 'react';
import ReactDOM from 'react-dom/client';

const myElement = <h1>I Learn JSX!</h1>;

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(myElement);
```

OUTPUT

I Learn JSX!

EX:2

```
import React from 'react';
import ReactDOM from 'react-dom/client';

const myElement = React.createElement('h1', {}, 'I Learn JSX!');

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(myElement);
```

expressions{}

EX:3

```
import React from 'react';
import ReactDOM from 'react-dom/client';

const myElement = <h1>React is {5 + 5} times better with JSX</h1>;

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(myElement);
```

OUTPUT

React is 10 times better with JSX

Conditions

EX:4

```
import React from 'react';
import ReactDOM from 'react-dom/client';

const x = 5;

let text = "Good morning";

if (x < 10) {
  text = "Hello";
}

const myElement = <h1>{text}</h1>;

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(myElement);
```

OUTPUT

Hello

Ternary expression

EX:5

```
import React from 'react';
import ReactDOM from 'react-dom/client';

const x = 5;

const myElement = <h1>{(x) < 10 ? "Hello" : "Good Morning"}</h1>;
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(myElement);
```

OUTPUT

Hello

Rendering Elements

The ReactDOM.render() function takes two arguments, HTML code and an HTML element.

The purpose of the function is to display the specified HTML code inside the specified HTML element.

EX:1

```
import React from 'react';
import ReactDOM from 'react-dom/client';

ReactDOM.render(<p>Hello</p>, document.getElementById('root'));
```

OUTPUT

Hello

EX:2

```
import React from 'react';
import ReactDOM from 'react-dom/client';
```

```
const myelement = (
```

```
  <table>
```

```
    <tr>
```

```
      <th>Name</th>
```

```
    </tr>
```

```
    <tr>
```

```
      <td>sai</td>
```

```
    </tr>
```

```
    <tr>
```

```
      <td>soni</td>
```

```
    </tr>
```

```
</table>
```

```
);
```

```
ReactDOM.render(myelement, document.getElementById('root'));
```

OUTPUT

Name

sai

soni

Components:

Components are independent and reusable bits of code. They serve the same purpose as JavaScript functions, but work in isolation and return HTML.

Components come in two types, Class components and Function components,

Class Component

A class component must include the extends `React.Component` statement. This statement creates an inheritance to `React.Component`, and gives your component access to `React.Component`'s functions.

The component also requires a `render()` method, this method returns HTML.

Functional component

A Function component also returns HTML, and behaves much the same way as a Class component, but Function components can be written using much less code, are easier to understand