

# React

React is an open-source Javascript library for building user interfaces, primarily used for single-page applications (SPAs) and dynamic web applications. It was developed by Facebook and is maintained by a community of developers.

React responds to changes in state and updates the UI components. If a component changes (props or state change) then React will re-render just that component and its children instead of the whole screen. This is known as being 'reactive'.

## How React ties into HTML

index.html => main.js => router => screens

## Components

Reusable building blocks defined as functions.

- 100% Atomic Design and dry.

```
JavaScript
function App() {
  return (
    <div>
      <h1>Hello World!</h1>
    </div>
  )
}
```

## JSX

Extension for Javascript that lets you write HTML-like code.

Looks like HTML but is really JS

- Can embed JS expressions directly with curly braces: {}

JavaScript

```
function App() {  
  const title = 'Howdy!'  
  return (  
    <div>  
      <h1>{title}</h1>  
    </div>  
  )  
}
```

JavaScript

```
function App() {  
  const title = 'Howdy!'  
  const color = 'red'  
  return (  
    <div>  
      <h1 style={{ color: color }}>{title}</h1>  
    </div>  
  )  
}
```

- Can add components as JSX elements (HTML-like)

JavaScript

```
function Title() {  
  const title = 'Howdy!'  
  const color = 'red'  
  return (  
    <h1 style={{ color: color }}>{title}</h1>  
  )  
}
```

```
function App() {  
  return (  
    <div>  
      <Title />  
    </div>  
  )  
}
```

```
}
```

- className instead of class (class is a reserved word in JS)

JavaScript

```
<h1
  className={'bg-primary title-height'}
  style={{ color: color }}
>
  {title}
</h1>
```

- Event handling - onClick and other attributes available on JSX elements

JavaScript

```
const changeColor = () => {
  let colors = ['red', 'blue', 'green', 'pink']
  let randomColor = colors[Math.floor(Math.random() * colors.length)]
  console.log('RANDOM COLOR: ', randomColor)
  return randomColor
}

function Title() {
  const title = 'Howdy!'
  const color = 'red'
  return (
    <h1
      className={'bg-primary title-height'}
      onClick={() => changeColor()}
      style={{ color: color }}
    >
      {title}
    </h1>
  )
}
```

# Components Props

Components are Javascript functions - can pass props between them as attributes in the JSX elements.

JavaScript

```
function Title({ color }) {
  const title = 'Howdy!'
  return (
    <h1
      className={'bg-primary title-height'}
      onClick={() => changeColor()}
      style={{ color: color }}
    >
      {title}
    </h1>
  )
}

function App() {
  return (
    <div>
      <Title color='pink' />
    </div>
  )
}
```

# Component State

JavaScript

```
import { useState } from "react"

const changeColor = () => {
  let colors = ['red', 'blue', 'green', 'pink']
  let randomColor = colors[Math.floor(Math.random() * colors.length)]
  return randomColor
}

function Title({ color }) {
  const title = 'Howdy!'
  return (
    <h1
```

```

        className={'bg-primary title-height'}
        style={{ color: color }}
      >
        {title}
      </h1>
    )
  }

function App() {
  const [color, setColor] = useState('pink')
  return (
    <div onClick={() => setColor(changeColor())}>
      <Title color={color} />
    </div>
  )
}

```

## Component Changes On Load

JavaScript

```

import { useEffect, useState } from 'react'

const changeColor = () => {
  let colors = ['red', 'blue', 'green', 'pink']
  let randomColor = colors[Math.floor(Math.random() * colors.length)]
  return randomColor
}

function Title({ color }) {
  const title = 'Howdy!'
  return (
    <h1 className={'bg-primary title-height'} style={{ color: color }}>
      {title}
    </h1>
  )
}

function App() {
  const [color, setColor] = useState('')

```

```

useEffect(() => {
  setColor(changeColor())
}, [])

return (
  <div onClick={() => setColor(changeColor())}>
    <Title color={color} />
  </div>
)
}

```

## Routing & Navigation

[React-Router library](#)

Create paths and assign the component that will be loaded.

```

JavaScript
const router = createBrowserRouter([
  {
    path: '/',
    element: <App />,
    errorElement: <ErrorPage />
  },
  {
    path: '/about',
    element: <About />
  },
])

```

Reference those paths in links.

```

JavaScript
import { Link } from "react-router-dom"

function App() {
  return (
    <Link to='/About'>About</Link>
  )
}

```

