

## Emmet

Emmet is a plugin for many popular text-editors which greatly improves HTML & CSS workflow

↳ Emmet is a bunch of different shortcuts and snippets where we type a small amount of code & it'll generate a bunch of code for us by just typing a small amount of information.

✱ Emmet allow us to type shortcuts that are then expanded into full pieces of code.

## CORS

Cross-Origin Resource Sharing or CORS is a mechanism that allows browsers to request data from 3rd party URLs (or origins) & is a common pain point for web developers.

If we need to allow certain URLs then we need to set the "access control allow origin". header and its value needs to match the origin header on the request.

What is crossorigin in script tag?

↳ The crossorigin attribute does let's us control whether or not we want to send identifying information from the browser off to some other source. (browser details)

The crossorigin attribute has two possible values - anonymous & use-credentials



crossorigin = "anonymous"

Request uses ~~CORS~~ CORS headers and credentials flag is set to 'same-origin'. There is no exchange of user credential via cookies, client-side SSL certificates or HTTP authentication, unless destination is the same origin.

crossorigin = "use-credentials"

Request uses CORS headers, credentials flag is set to 'include' & user credentials are always included.

By default CORS is not used at all.

Why React Known as React?

↳ React was developed for applications that have constantly changing data. Since React is front-end framework or 'view' in MVC, this means that as the user clicks around & changes the app's data, the view should 'react' or change with those events. User events being mouse click, typing, submitting a form etc.

Create React Element

```
<div id="root"></div>
```

```
<script>
```

```
const heading1 = React.createElement('h1', {}, 'Hello World!');
```

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

```
root.render(heading1);
```

```
</script>
```

```
console.log(heading1);  
{---}
```

# React element is nothing but an object.



What is difference between React and ReactDOM?

- ↳ React is a JavaScript library for building user interface and ReactDOM is JavaScript library that allow React to interact with DOM.
- ↳ whenever we use component, classes, element, etc. we are using React. React package holds the react source for components, state, props and all the code that is react.
- ↳ whenever we use method like `render()` or `findDOMNode()` we're using React-DOM. "react-dom" package glue b/w React & the DOM.

Why separate them?

- ↳ React-DOM binds the idea of React to a web browser and we have have react native which we use for mobile app development.

What is difference between `react-development.js` and `react-production.js` file via CDN?

- ↳ The very basic diff is that Production Build has ugly, minified version of your javascript code, so this makes rendering of file on end user's browser very quick and performance enhancing. There no hot reloading included. Source Maps might be included as separate files depending on <sup>your</sup> webpack config.
- ↳ whereas, development build is used for development reason. We have Source Maps, debugging & often times hot reloading ability in those builds.



React is a JavaScript library for building user interfaces.

### Library v/s Framework

A library is a collection of functions developers can call at will and still control the flow of the software application. The operations performed to be performed by libraries are precisely and specifically defined.

A framework is a structure in which the application defines the content of the operation by asking the developer to fill in the spaces. The framework defines the concept with the functionality itself is defined by the developer with the end-user in mind.

A framework inverts the control of the program. It tells the developer what they need. A library doesn't. The programmer calls the library where and when they need it.