1. What is Emmet?

Set of plugins for text editors that allow for high speed coding and editing in HTML,XML etc.

Syntax:

Elements – you can use elements names like div or p to generate html tags. Emmet doesn’t have a predefined set of available tag names, you can write any word and transform it into a tag div , foo etc.

Child: >

div>ul>li –

<div>

    <ul>

        <li></li>

    </ul>

   </div>

Sibling – + operator

Div+p+bq

<div></div>

   <p></p>

   <blockquote></blockquote>

Climb up- ^

With > operator you’re descending down the generated tree and positions of all sibling elements will be resolved against the most deepest element:

With ^ operator, you can climb one level up the tree and change context where following elements should appear:

div+div>p>span+em^bq

results into->

<div></div>

<div>

<p><span></span><em></em></p>

<blockquote></blockquote>

</div>

Multiplication

With \* operator you can define how many times element should be outputted.

ul>li\*5

...outputs to

<ul>

<li></li>

<li></li>

<li></li>

<li></li>

<li></li>

</ul>

Grouping – ()

Parenthesises are used by Emmets’ power users for grouping subtrees in complex abbreviations:

div>(header>ul>li\*2>a)+footer>p

...expands to

<div>

<header>

<ul>

<li><a href=""></a></li>

<li><a href=""></a></li>

</ul>

</header>

<footer>

<p></p>

</footer>

</div>

Attribute operator

div#header+div.page+div#footer.class1.class2.class3

...will output

<div id="header"></div>

<div class="page"></div>

<div id="footer" class="class1 class2 class3"></div>

**Custom attributes**

You can use [attr] notation (as in CSS) to add custom attributes to your element:

td[title="Hello world!" colspan=3]

...outputs

<td title="Hello world!" colspan="3"></td>

### Item numbering: $

With multiplication \* operator you can repeat elements, but with $ you can number them. Place $ operator inside element’s name, attribute’s name or attribute’s value to output current number of repeated element:

ul>li.item$\*5

...outputs to

<ul>

<li class="item1"></li>

<li class="item2"></li>

<li class="item3"></li>

<li class="item4"></li>

<li class="item5"></li>

</ul>

## Text: {}

You can use curly braces to add text to element:

a{Click me}

...will produce

<a href="">Click me</a>

Don’t use space in b/w because space is stop symbol where emmet stops abbreviation parsing.

1. Diff b/w library and framework?

The technical difference b/w framework and library lies in a term inversion of control.

When you use a library, you oversee the flow of the application. You care choosing when and where to call the library. When you use a framework, the framework oversees the flow. It provides some places for you to plug in your code, but it calls the code you plugged in as needed.

1. What is CDN? Why do we use it?

A content delivery network is a geographically distributed group of servers that caches content close to end users. A CDN allows for the quick transfer of assets needed for loading internet content, including HTML pages, including HTML pages etc.

Denial of service attack.

1. Why is react known as react?

It was developed by Facebook, and the name “React” was chosen because it is meant to help developers build user interfaces that are fast and responsive, or “reactive.” The library was designed to “react” to changes in data. When data in a React application changes, the components that depend on that data are automatically updated, which allows for efficient and seamless updates to the user interface. The name “React” reflects this reactive nature of the library. The idea behind React is to build reusable components that can be rendered on the front-end, rather than writing a new piece of code every time you need to display something on the screen. This makes it easier and more efficient to build complex user interfaces.

1. What is cross-origin in the script tag?

Origin – web content’s origin is defined by the scheme (protocol), hostname (domain), and port of the URL used to access it. Two objects have the same origin only when the scheme, hostname, and port all match.

Some operations are restricted to same origin content, and this restriction can be lifted using CORS.

CORS – cross origin resource sharing is an HTTP-header based mechanism that allows a server to indicate any origins other than its own form which a browser should permit loading resources. CORS also relies on a mechanism by which browsers make a “preflight” request to the server hosting the across origin resources, in order to check that the server hosting the cross-origin resource, in order to check that the server will permit the actual request. In that preflight, the browser sends headers that indicate the HTTP method and headers that will be used in the actual request.

The cross-origin attribute, valid on the audio, image, link and script elements, provides support for CORS, defining how the element handles cross-origin requests thereby enabling the configuration of the CORS requests for the element’s fetched data. Depending on the element, the attribute can be a CORS settings attribute.

The integrity attribute allows a browser to check the fetched script to ensure that the code is never loaded if the source has been manipulated.

1. Diff b/w react and React DOM?

React DOM is the glue b/w React and the DOM. Often, you will only use it one it for one single thing mounting with ReactDOM.render(). Another useful feature of ReactDOM is ReactDOM.findDOMNode() which you can use to gain direct access to a DOM element. If your app is isomorphic, you would also use ReactDOM.readerToString() in your back-and code.

For everything else, there’s React. You use React to define and create your elements, for lifecycle hooks, etc. i.e. the guts of a React application.

The reason React and React DOM were split into two libraries was due to the arrival of React native. React contains functionality utilized only in web apps.

This paves the way to writing components that can be shared between the web version of React and React Native. We don’t expect all the code in an app to be shared, but we want to able to be able to share the components that do behave the same across platforms.

1. Diff b/w production and development build in ReactJS.

The development build is used as the name suggests – for development reasons. You have source maps, debugging and often hot reloading ability in those builds.

The production build, on the other hand, runs in production mode which means this is code running on your client’s machine. The production build runs uglify and builds your source file into one or multiple minimized files. It also extracts CSS and images and of course any other sources you are loading with Webpack. There is also no hot reloading included. Source Maps might be included as separate files depending on your webpack devtool settings.