Chapter 01 - Inception

Theory -

● What is Emmet?

Most text editors out there allow to store and re-use commonly used code chunks, called “snippets”.

Emmet is **a free add-on for your text editor that allows you to type shortcuts that** are then expanded into full pieces of code.

Support for [Emmet](https://emmet.io/) snippets and expansion is built right into Visual Studio Code, **no extension required**.

More information regarding Emmet with respect to VS Code IDE can be found on

<https://code.visualstudio.com/docs/editor/emmet>

● Difference between a Library and Framework?

Both frameworks and libraries are code written by someone else that is used to help solve common problems.

A library is like you already have your own code, but you need a bit of help with some features of that library, which can make your code faster and scalable. You don’t feel like making your everything from scratch due to time crunch. Library allows you to pick and choose different things to go in your code. You are in control.

A framework, on the other hand, is like building a code. You have a set of architecture and design with limited choices. Ultimately, the framework is in control. And it will let you know when and where you can provide your input.

The technical difference between a framework and library lies in a term called inversion of control.

More details about Inversion of Control can be found on this video by Akshay Saini, who beautifully elaborated Callback Hell & Inversion of Control-

<https://www.youtube.com/watch?v=yEKtJGha3yM>

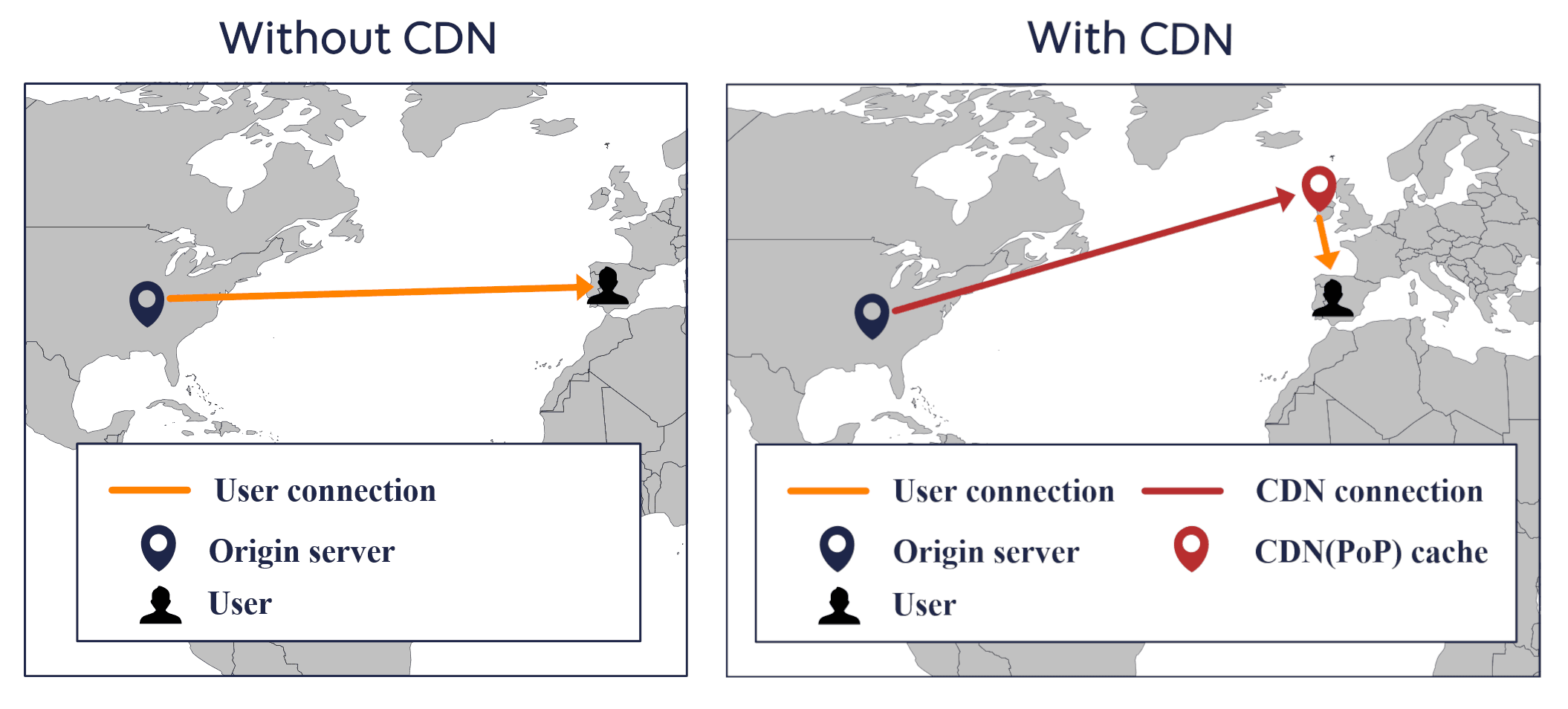
In short, A framework inverts the control of the program. It tells the developer what they need.

A library doesn’t. The programmer calls the library where and when *they* need it.

● What is CDN? Why do we use it?

CDN stands for Content Delivery Network. The Origin server is the server where original web files are placed. In the absence of CDN, if the user sends request directly to the origin server, the client will have to wait for the request to be fulfilled depending upon the geo location of the client and server. Also, website’s work load will increase. CDN also protects from online attacks by masking the original server making primary data source practically invisible.

With a CDN, static and dynamic content, large file downloads, etc **can be cached (saved for some time)** in multiple locations at once, on servers all over the world. These locations are known as PoPs (Points of Presence), sometimes also as edge locations and they are **strategically located in populated areas** with the most demand.



Why do we use it?

1. CDN is key to a fast and responsive site
2. It will [lower your cost for data transfer](https://www.stormit.cloud/blog/aws-data-transfer-pricing-how-to-reduce-costs/) from your origin server because the most desired content will be cached in PoPs and your origin server doesn’t have to serve as many requests as it would normally do.
3. Protect not only your content but also your infrastructure behind it.
4. Protects your origin server from traffic spikes and it is easily and automatically scalable to your needs.
5. Data that should be accessible to your users at any time are replicated on multiple servers, so if your origin is unavailable, PoPs can easily manage that for you.

● Why is React known as React?

React is a JavaScript library for building user interfaces.

React is named “React” because:

* It "reacts" quickly to changes without reloading the whole page.
* It uses the virtual DOM to efficiently update parts of a webpage.
* It’s built around components that "react" and update.

● What is crossorigin in script tag?

The **<script> crossorigin Attribute** is used for loading an external script into their domain from a third party server or another domain with the support of HTTP **CORS Request.** This attribute is used to protect sensitive information from the third party when fetching out the results.

Cross-Origin Resource Sharing(CORS) is a tool that allows web pages requesting for resources from another domain that is outside of their own domain. It is also used for managing cross-origin requests which is a request for a resource from an outside domain.

● What is difference between React and ReactDOM

We need to include 2 libraries- React and ReactDOM in order to work with React. React library is responsible for creating views and ReactDOM library is responsible to actually render UI in the browser.

For e.g.-

When we create a new React element, we need React liabrary.

Syntax: React.createElement(type, {}, [...children]);

const pElement = React.createElement(‘p', null, 'React.js paragraph’);

For creating react element, we are passing 3 arguments. As the example suggests, type indicates the type of element, here its <p> tag. Null is given as no props need to be assigned to this paragraph. Here, we can replace it with any className if we need to apply any CSS to it. Third argument is the text you need to insert in tag <p>.

Now when react element is created, we need to show it on UI. And to do this, we need ReactDOM.

For this, we need to createRoot and render the same.

e.g.-

const root= ReactDOM.createRoot(document.getElementById(“root”));

(Assuming we have <div> element in body with id “root”.)

root.render(pElement);

As the example specifies, ReactDOM has methods createRoot and render, using these the <p> tag will be rendered on browser.

● What is difference between react.development.js and react.production.js files via CDN?

The react.development.js file is meant for development purposes, while the react.production.js file is meant for production purposes.

React.development.js file contains development-specific warning messages that could help developers detect and fix errors. This file is larger in size as compared to react.production.js, as it includes additional code related to development environments.

On the other hand, react.production.js is optimized for performance, hence, is faster and smaller in size as compared to react.development.js. This file removes development-specific warnings and uses a minified code that enhances the performance of react components and applications.

When these files are delivered via CDN (Content Delivery Network), it means that they are hosted and distributed through a global network of servers closest to your user’s location. This reduces the loading time of your website, making it fast, efficient and easily accessible to users worldwide.

In summary, the difference between react.development.js and react.production.js files via CDN lies in their functionality. While react.development.js is meant for development purposes, react.production.js is optimized for performance in production environments, and delivering them through CDN means faster loading times.

● What is async and defer? - see my Youtube video ;)

<script> tag can be combined with async or defer or no attribute.

In normal script tag without any attribute, html parsing starts, as soon as it encounters script, its fetched and executed and then again html parsing goes on.

E.g.- <script src=” ”> </script>

In case of async attribute, html parsing and script tag is fetched simultaneously and as soon as script is fetched from the network and executed then only HTML parsing continues.

E.g.- <script async src=” ”> </script>

In case of defer attribute, html parsing and script tag is fetched simultaneously, it waits for HTML parsing to complete and then script is executed.

E.g.- <script defer src=” ”> </script>