How to start frontend

# Essential frontend skills

### Learn Fundamentals

* HTML( for structuring the website or content of website)
* CSS( add beauty of website)
* JS (add functionality to website)

### Framework vs library

Framework force our application into structure while library not.

* React
* Angular
* Vue

### Version control system

To track our project history and work with other.

GIT

### CSS preprocessor

These are the advance css which has more features but not spported by browsers so we compile these advance css through preprocessor and it return us old css which is spported in browers.

* SASS
* LESS
* Stylus

### JS preprocessor

Just like css there are also js preprocessor that convert mordren js in old js such as

* TypeScript
* CoffeeScript

# How web work

There is a server beyond any website and the server contain all the source code of the website when to access the website user go to the browser and in the search bar type the address of the server the usually the address of the server is in simple language like we can understand like alibaba.com or Amazon etc but beyond this there is some IP addresses of every website and over browser is link with DNS domain name system which is a huge dictionary where all where the name of the website in user readable form is link with the IP addresses ipv4 or IPv6 this process is called requesting when over request is goes to the server then reserve gives response to us in by sharing their source code the source code includes HTML CSS and JS of the website.

How to create a web page using semantic tags

Semantic tags are HTML tags that describe the content they contain. They are important for search engine optimization (SEO) and web accessibility. Here are the steps to create a web page using semantic tags:

1. Plan your content: Before you start coding, plan your content. Decide what information you want to include on your web page and organize it into sections.
2. Choose your semantic tags: Choose the semantic tags that best describe the content of each section. For example, use <header> for the page header, <nav> for the navigation menu, <main> for the main content, <section> for each content section, <article> for articles or blog posts, <aside> for sidebars, and <footer> for the page footer.
3. Write your HTML code: Once you have decided on your semantic tags, write your HTML code. Start with the basic structure of an HTML document, including the <!DOCTYPE> declaration, <html> tag, <head> tag, and <body> tag. Then, add your semantic tags to structure your content.
4. Add content: Add your content to each section. Use headings (<h1>, <h2>, etc.) to structure your content and make it easier to read. Use lists (<ul> and <ol>) for bullet points or numbered lists, and use <p> tags for paragraphs of text.
5. Add images and multimedia: Use <img> tags to add images to your web page. Use <video> and <audio> tags to add multimedia content.
6. Test and validate your code: Use a web browser to test your web page and make sure it looks and functions as expected. Use an HTML validator to check your code for errors and make sure it meets web standards.

By following these steps, you can create a well-structured and accessible web page using semantic tags.