**What Is JavaScript Used for in Front-end Web Development?**

A front-end web developer’s role is to create the code and mark-up that is rendered by a web browser when you visit a site (read: they control what you see when you visit a webpage).

There are three main components when it comes to front-end development: HTML, CSS and JavaScript. Each are critical for making a webpage what it is. HTML is the structure and content of the site, CSS (Cascading Style Sheets) makes it look pretty, and, lastly, JavaScript is what powers its interactivity. Each work hand-in-hand when it comes to building websites, but the focus of this blog post is on JavaScript and how it’s used.

**INTERACTIVITY**

JavaScript is a very powerful tool that can do many things for a website. For one, it powers the site’s general interactivity. JavaScript makes it possible to build rich UI components such as image sliders, pop-ups, site navigation mega menus, form validations, tabs, accordions, and much more.

It can also do subtler operations as well. For instance, you may click a checkbox on a form, and, depending on the checkbox selected, it loads a pop-up that asks you another question. It gives the site additional functionality that isn’t otherwise achievable with HTML and CSS alone. JavaScript allows webpages to respond to user activity and dynamically update themselves, and all without requiring a page reload to change its appearance.

**AJAX**

We know what you’re probably thinking, but we’re not talking about the stuff you clean your sink with! AJAX stands for Asynchronous JavaScript and XML. It basically allows a webpage to interact with web server in the background, without a page reload.

Let’s take, for example, a listing of events on a webpage. By default, these events are sorted by month, most recent, and then categorized by type. When changing the ‘ALL’ filter to the ‘General’ one, an operation is triggered in the background, and the page gets partially reloaded with our newly queried items. The page doesn’t flicker as it reloads, and we stay put exactly where we are, which makes for a more pleasing web experience.

This can also be used on forms to send data to the server to be stored; again, to keep the page from reloading. AJAX allows a webpage to send and receive data from a web server without triggering a page refresh, allowing for a seamless user experience between the user and the web application.

**CROSS-BROWSER COMPATIBILITY AND STANDARDS COMPLIANCE**

There are quite a few web browsers out there: Firefox, Chrome, Safari, Opera, Internet Explorer (10, 11, Edge)… and they all run on different operating systems and devices. From time to time, each of these browsers has its own unique bugs and quirks. Nothing is perfect, unfortunately. While W3C compliance standards continue to improve across the board, there still comes a time when a front-end web developer needs to resolve issues by leveraging JavaScript.

For example, you may write a CSS styling rule that works perfectly in all browsers… except for Internet Explorer 10. The issue you’re then faced with is to rewrite the whole section — which you painstakingly made perfect — or target the one browser and write in a workaround for it. However, by using a JavaScript library such as Modernizr, you can help isolate code to that specific browser. By leveraging the script you’ve written, a few new style rules, you’ve resolved the issue.

**PLUGINS**

There are various plugins and that run off of JavaScript. If you’ve ever visited a site that features banner ads, has chat support, suggests content, has forms, or offers social sharing, there’s a good chance it’s powered by a 3rd party JavaScript plugin. Usually, these plugins have configurable options that need additional set-up to function properly. Understanding the configurable options of these plugins is essential. Plugins are generally intended to be easily dropped into a site with little modification.

**FRAMEWORKS**

A JavaScript framework can be a powerful tool you can use to help render the page. These are typically only used when there are complex dynamic interactions that need to occur.

One example of this is if you have a multi-step form-fill. In this case, the form fill process has certain steps that only occur based on previously entered information. Also, certain data gets populated for certain inputs as well as previous inputs. Doing this without a framework can be very difficult task to achieve. Things can get problematic, and this can happen fast.

Using a JavaScript framework helps resolve these issues so you can complete your wonderful form, and make your clients happy. While there are dozens, the most popular ones (as of this writing) are Google’s Angular, Facebook’s React and the open source Vue.js.

JavaScript is a very important tool for a front-end web developer. Without it, webpages wouldn’t have become the dynamic web applications they are today. There would be no image carousels. There would be no partial page reloads that keep your spot on the page. There would be ‘no’ on lots of things that we’ve become accustomed to. JavaScript is kind-of like a Swiss Army Knife in a sense, it can do all you need and can prepare for the things you don’t know you need yet.

**How JavaScript works??**

since there are two phases in code execution frst is globla context creation mean global memory create huti h smjhny kliy aesa keh skty hn es me varible ko values wgra assign huti hn jb control es me enter huta h aur ye global context stack me b store huta jb es glbl cntxt ki baari aati h, variable k respect se hm hoisting kr skty hn but let,const me ye huta h k hoisting huti h but differently mtlb k agr varble ki value set kr rakhi h but variable ko pehly access kr rahy hn tu phr ye uninitialized ka error dy ga but agr value set he ni ki hugi aur variable ko access krny ki Koshish kre gy tu ye alg error dy ga jo k undefined hugi, axha compilation jo h us me tokenization hu rahe h aur parsing hu rahe huti h wgra but es me just scope wgra ka knowledge rakhta h takeh execution se pehly error ko check kiya ja sky,aur execution phase me jo GEC huta h aur es me jo global memory wala phase huta h us me un variable ki value set ki jaati hn aur phr jb control line by line execute krta h tu wo phr glbl memory ko yaha se original values set krta h aur jb ye sara execute hu jata h tu phr ye GEC stack se delete hu jata h aur jo next GEC huta h uski execution start huti h.

Conclusion:- compilation phase me scope check huta h aur GEC m jo memory creation phase huta h us me varible ki type set ki jaati h aur jo line by line execute huta h us me un variable ko values assigned huti hn axha jb es me function ya classes wgra execute huny lagti hn tu en kliy phr usi GCE me ek child LCE(local context execution)bnta h jo same procedure mtlb local memory creation aur line by execution ko follow krega aur ye saary call stack me track huty rahy gy, ek aur chota sa difference wo ye h k Global GEC me object by default window huta h but child GEC me by default object ek array object huta h uski property hm index aur length ko use kr k access kr skty hn ,es local m ye huta h k agr varble ko value local execution m ni Mili tu wo eska jo parent huga us me dekhy ga agr waha b ni mili tu wo us k parent ko dekhy ga aur es trh dekhta rahega agr kahein b ni milti tu phr ye error throw krega.

**The new features introduced in ES6 version of JavaScript are:**

Arrow functions.

Multi-line Strings.

The destructuring assignment.

Enhanced object literals.

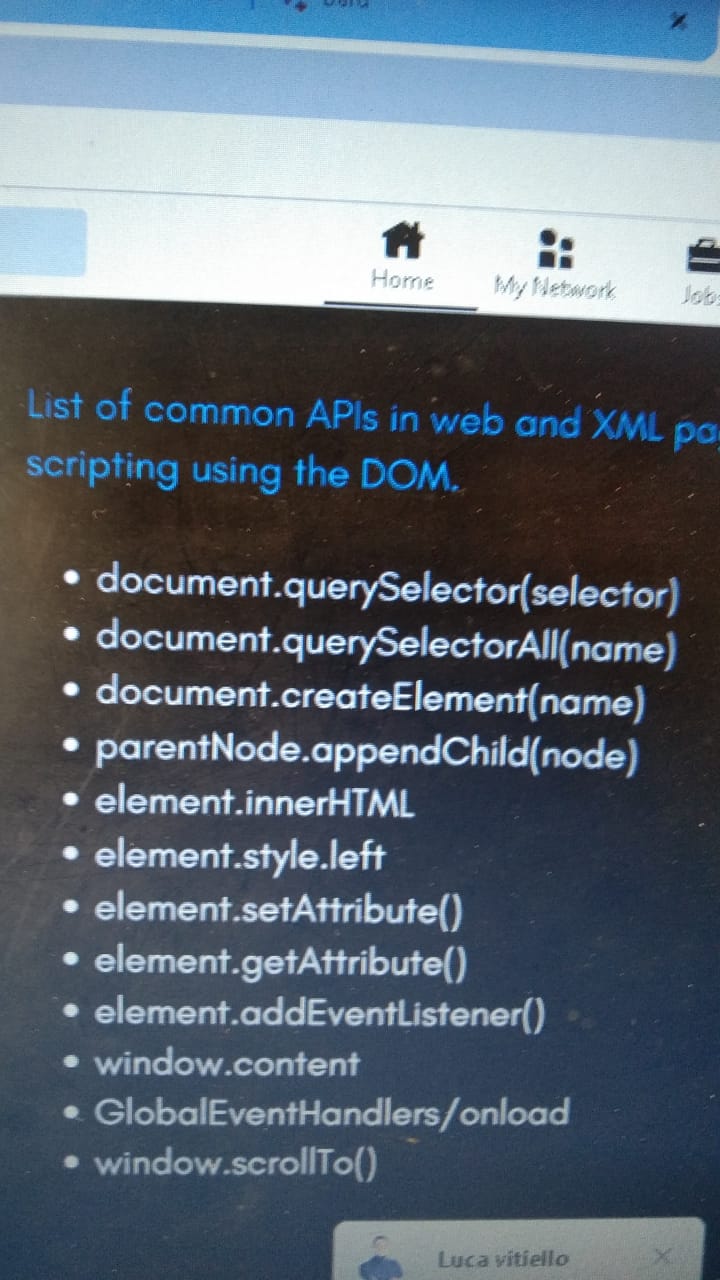
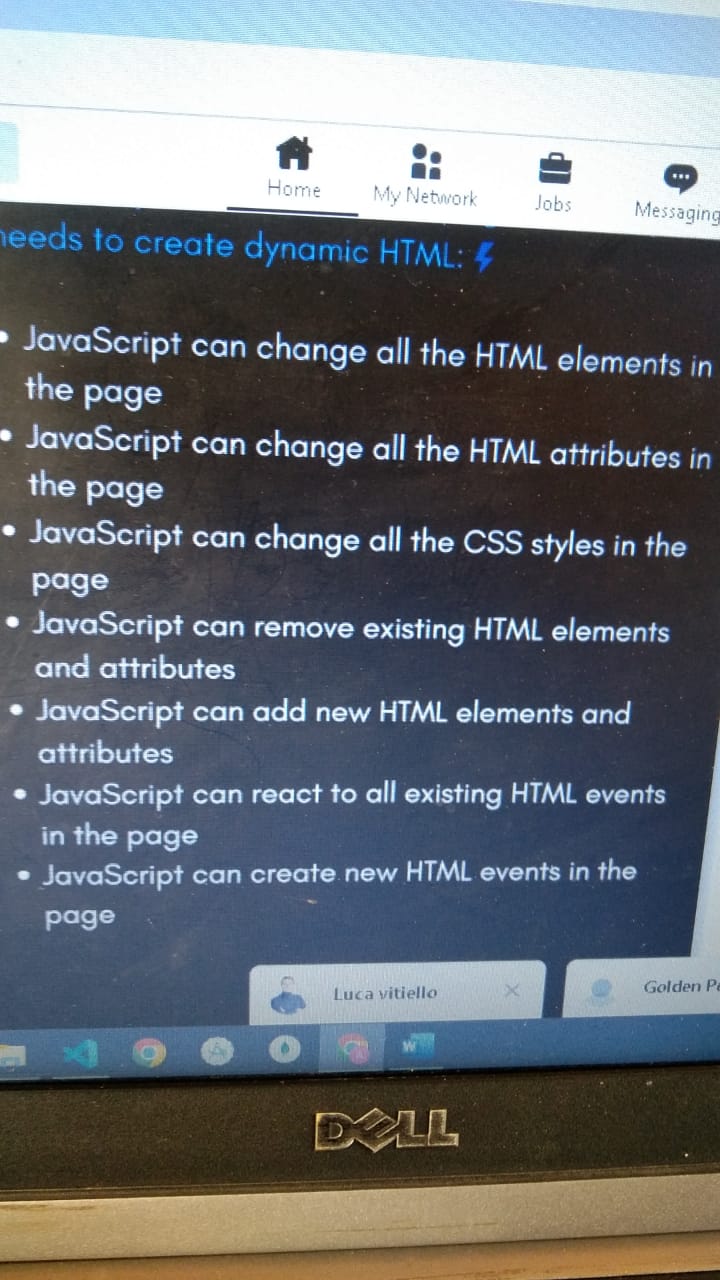
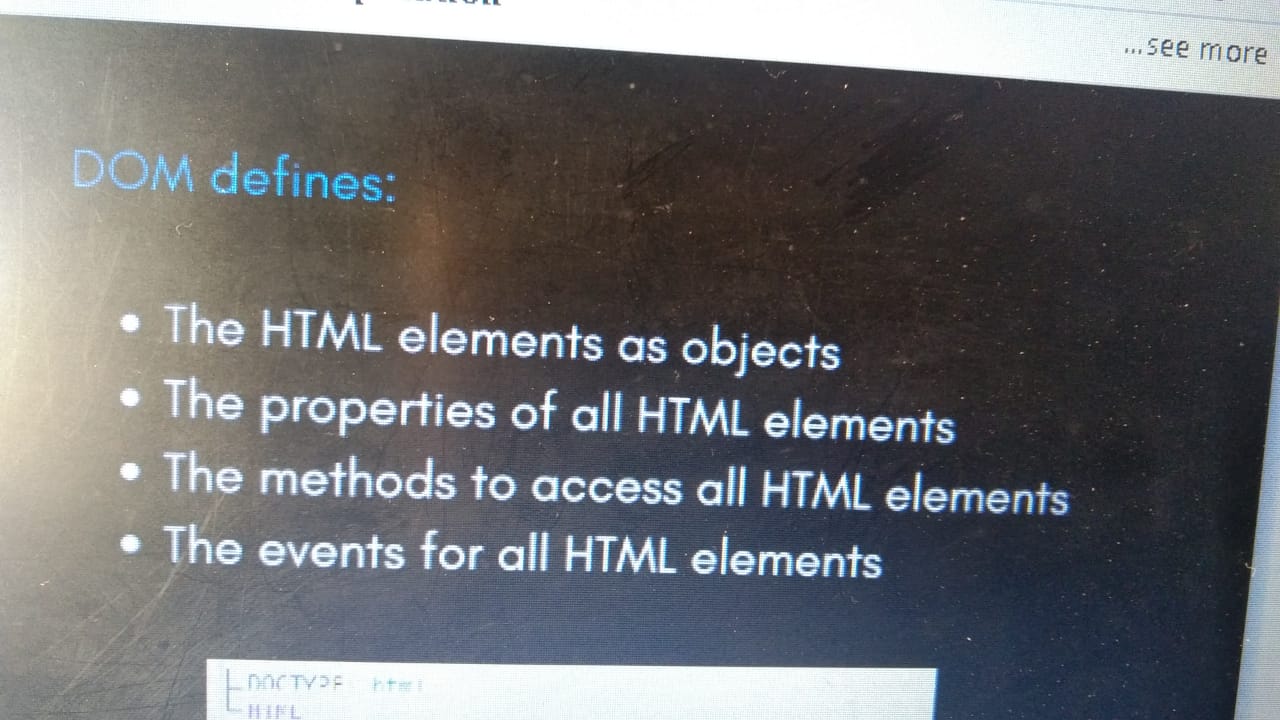
Promises

**Is javascript a statically typed or a dynamically typed language?**

JavaScript is a dynamically typed language. In a dynamically typed language, the type of a variable is checked during run-time in contrast to a statically typed language, where the type of a variable is checked during compile-time.

**Static Typing 🡪** have types Variables and cannot change type,

**Dynamic Typing🡪** Variables have no types and Variables can change type

**DOM:**

**DOM EVENTS:**

**Events mean to perform some action or clicks on the front end,we have different types of events like mouse hover,click,key press.**

**Ways to handle the events:**

**1-simple 🡪** mean we pass/define function either regular or arrow or other task in the event directly

**2-**in this way we can define the function outside and call it in the event

**3-**We can use the eventaddlistener method 🡪through this we can occur the event ,we can also use the loop to do the event. And this way is more efficient.

**How events works?**

**1-Event Captchurng:** As we have the web api which holds the events and javascript GC (global context),so when any event occur in the web api it passes the event in the call back queue, As stack empty it call those events one by one in the stack from the call back queue ,so in this way events are run by js engine.

**2-Bubbling: As we have seen above in Event Captchuring there is only one event occur at a time,**

**but in the bubbling all the events linked in the call back hell ,they all execute at once.**

**3-In Event Deligation: We can set the only one event for all elements instead of bubbling where for every element we set the event and so all the events call at a time.**

**promise,XHR, fetch,async await:**

promise is a future value,if promise is fulfill then value will resolve otherwise value will be rejected we can also resolve value by using then method and catch method for rejected, then always return promise that's why we can create chain by using then always return promise from then otherwise it will call itself and it's chain promise both simultaneously we can also remove curly braces from call back function and now no need to write the return,we can also return promise from function,it is fall in micro queue, promise is a object , promise is consumed by the browser.

2-Request,3ways to request the server xmlhttpreqesust,fetch=>ye b promise return krta h aur es m b hm promise ki trh chaining kr skty hn es me catch value ni leta until then me error throw krvaay ya network error jb tk na aay ye usually URL leta h but hm agr request post krni hu tu wo b es me kr skty hn by using curly braces aur phr us me jo b data huga pass kr dy gy aur ye promise ko .then kr k consume kiya ja raha h ,async wait ye b jb use krty hn ek promise he return krta h aur phr es me b hm fetch wgra k saath await use kr k jis function me es ko bnaaya huga usko call kr k us k saath chaining wgra kr skty hn aur ye tb hm use krty hn jb hmy data ko fetch ya load krna hu aur agr arrow function use kry gy tu us me paranthsis se pehly phr async lagy ga aur normal function me pehly lagay ga aur ye broser k backend m run huga

**Array aur Object:**

premitive m hm value ko chng ni kr skty hn but in reference m hm ek m change kre gy tu wo dosry m b chnge hu jaay ga bcz both pointer refere same address,unshift front se add krta h aur ahift front se remove krta h similalry push last pe add krta h aur pop last se remove krta h,const me hm value change ni kr skty variable ki but array with const me values add kr skty hn but const k saath new same name array ni create kr skty but ye let me create hu jata h,simple for loop hmy array return krta h with values but hm index k through b value access kr skty hn, for of loop hmy just values deta h aur for in loop hmy just index deta h, array 3 ways se clone huta h slice,concat, spread operator ye basically hmy jo iterable huti hn like strings,array etc unk elements return krta h, but in destructuring hm always esko end pe use krty hn,

object b ek reference type h aur es me b hm dot laga k key ya property ko add kr skty hn aur access b but agr ek property m 2words hu aur space hu tu phr usko access krny kliy hm bracket aur double quotes ka use kre gy,hm variable ko b key bna skty using square bracket esko computed key b kehty hn,obj destructuring m curly braces m key ko likhy gy aur us obj k name assign kr dy gy tu wo keys ab variables bn jaay gy,mested destructing me hm array symbol aur us me object symbol wgra ka use kr k destructing kr skty hn

**Everything About Functions:**

function expression jis me function ko variable me assig krty hn aur es time function ko name dy b skty hn aur ni b, function declaration ye wohe normal function h, let and const are block scope and var is functional scope , function accept function is called callback function and also function can return a function and when we have to use the function we have to call in which this function will be saved otherwise the function only Will be printed.

**Scope: Local and Global Scope for variables and functions**

Scope Chain refers to the situation when one variable, which may have a global, local, function, or block scope, is used by another variable, function, or block, which may also have a global, local, function, or block scope. This entire chain construction continues till the user decides to halt it in accordance with the need

**Array Methods: (es me slice method add aur delete kliy use huta by using index)**

foreach method ye callback function leta h aur loop ka kaam krta h aur hr iteration pe array ki value aur index return krta h ,map methd ye ba callback function leta h aur array ki value aur index return krta h but es me frq ye h k ye un value ka array return krta h,filter method ye b callback function leta h but ye always us callback function se boolean value accept krta h , reduce method me ye huta h k ye b callback function leta h aur jb frst time function call huga us time frstvalue aur us array ki second value le ga aur phr jb next time call huga tb jo pehly waly call ki return value hugi wo accumulator hugi aur array ki third value currentvalue bn jaay gi es me hm accumulator ki default value b pass kr skty hn as 3rd parameter in reduce and so on for all the values of array ,

sort method do sorting on string and change the array ,find method search the value of specific criteria,every method is similar to other but it return true when callback function also return the true for all values of array , some method work only for one specific value in array,fill method is used to insert the new values in existing values of array and it will change the array , slice method is used for selection of array elements

**Set and Map:**

set me unique values huti hn es ki iterate b kiya ja skta h using of loop ,map b object ki trh huti h but difence ye h k object me keys srf string ya symbol hu skti hn aur un pe of loop ni laga skty but in loop laga skty hn aur ye oprtr b ?. use krty hn jb hmy value ka pta na hu but hm ni chaaty k error aay us se bchny kliy ye use krty hn but map me hm koi b key use kr skty hn aur esko iterate b kr skty hn using of loop aur es me set method k through value add kr skty hn map me aur key ko get method k through access krty hn aur ye hmy array return krta h aur es array ko hm destructing b kr skty hn aur map m hm array bna kr phr us me array bna kr key value set kr skty hn

**code side db setup huta h aur db side pe db formation huta h,**

**code side--mongodb side**

**db setup--> db formation**

**model-->collection**

**schema-->documents**