**CORS:**Cross origin resource sharing. Cross origin request 🡪 a request for a resource(like an image) outside the origin.   
ORIGIN 🡪 combination of URI scheme, host name and port number. It prevents a malicious script on one page from obtaining access to sensitive data on another web page through that page’s DOM.  
CORS 🡪 It is a mechanism that enables web browsers or other web clients to cross-origin requests. It's a browser mechanism that allows web pages in one domain to have controlled access to resources in other domains. A CORS policy was implemented to circumvent(finds a way) restrictions caused by same-origin policies.

1. User loads the page from ‘A.com’
2. While loading the initial page from ‘a.com’ an ajax request is made to ‘B.com’
3. Request from ‘A.com’ to ‘B.com’ is a cross origin request. Hence browser will first perform preflight request.
4. If preflight request is successful, B sends the “Access-control-allow-origin:HTTP://A.com” header, then the original request will be sent.
5. If preflight request is unsuccessful, B sends the “Access-control-allow-origin:No”, then the browser will throw an error.

Ajax(Asynchronous JavaScript and XML) helps in

* Updating a web page without reloading the page
* Request data from server - after the page has loaded
* Receive data from server - after the page has loaded
* Send data to server - in the background

Preflight request: It is a cors request that checks to see if cors protocol is understood and a server is aware using specific methods and headers. It is an option request, using 3 http request headers: ACR method, ACR headers and Origin header.

HTTP 🡪 used for transferring data over a network. Website content , API calls uses http protocol.

* **HTTP vs HTTPS:**  
   HTTPS is HTTP(Hypertext Transfer Protocol) with (public key(2 keys))encryption and verification. HTTPS uses TLS(SSL) to encrypt normal HTTP requests and responses, and to digitally sign those requests and response. HTTP Works at Application Layer and HTTPS works at Transport Layer

<https://www.cloudflare.com/en-gb/learning/ssl/why-is-http-not-secure/>

http – port number = 80  
https – port number = 443

**Multi Threading:**Thread is an independent part or unit of a process that is being executed. Whenever multiple threads execute in a process at the same time, we call this as “multithreading”.   
ADV: Response time of application is improved since requests from one thread don’t block requests from other threads. If one thread encounters an exception, it will not affect the other threads.

Connection leak: Connection leaks occur when some database requests or transactions are not closed properly or are not committed, causing the connections to be abandoned and closed permanently.

**Promises:** It is a proxy for a value not necessarily known when the promise is created.  
A promise object contains both the producing code and calls to the consuming code

let myPromise = new Promise(function(myResolve, myReject) {  
 //Producing code “may take some time”  
 myResolve(); //when successful  
 myReject(); //when error  
 });  
 //Consuming code “Must wait for a fulfilled promise”  
 myPromise.then(  
 function(value){ /\*code if successful\*/ },  
 function(error){ /\*code if some error\*/ }  
 );

When producing code obtains the result, it should call one of the 2 callbacks: S🡪myResolve, E🡪 myReje  
A javaScript Promise object can be pending – result is undefined, fulfilled – result is a value, rejected – result is an error object.

**API: Application programming interface**  
 These are set of rules and protocols that define how software programs or devices can communicate with each other. 3 most common APIs are REST, RPC and SOAP.

REST(Representational state transfer):  
 Request for data is sent to REST API, usually in http. Can return messages in a variety of formats: HTML, XML, JSON(javascript object notation), plain text

SOAP(Simple object access protocol):  
 First designed so that applications built with different languages and on different platforms could communicate. Because it is a protocol, it imposes built-in rules that increases its complexity, which can lead to longer page load times.

**Temporal Dead Zone:** let and const declarations are block scoped. Var doesn’t have such a restriction. Unlike var, which can be accessed before its declaration, you cannot access the let or const variables until they are initialized with some value. TDZ is time from beginning of the execution of a block in which let or const is declared until these variables are initialized. If anyone tries to access those variables during that zone, JS will always throw a reference error  
 console.log(varNumber); // undefined  
console.log(letNumber); // Throws the reference error letNumber is not defined  
var varNumber = 3;  
let letNumber = 4;

| **MVC** | **MVP** |
| --- | --- |
| MVC suggests splitting the code into three components. As soon as the developer creates a class or file for an application, he or she must categorize it into one of three layers: Model, View, and Controller. | This is an architectural pattern that helps compensate for some of the shortcomings of MVC. It is composed of three components i.e., Model, View, and Presenter. |
| The controller serves as a bridge between the view and model layers and therefore provides the application's user interface. As soon as the Model changes, the Controller updates the View. | The presenter pulls data from the model and applies the UI  (user interface) logic to determine what to show. In response to the user's input notification, it manages the state of the View and takes appropriate actions. |
| Controllers and views have a many-to-one relationship since one Controller can select different Views depending on the operation required. | Presenter and View have a one-to-one relationship since the Presenter class manages only one View at a time. |
| Support for unit testing is limited. | The unit testing process is highly supported. |

A cookie is generally a small data that is sent from website and stored on the user’s machine by a web browser that was used to access the website. Used to remember the information for later use and also to record the browsing activity on a website.

Closures:  
 Global variables can be made local with closures. It is a function having access to the parent scope, even after the parent function.  
 const add = (function () {  
   let counter = 0;  
   return function () {counter += 1; return counter}  
 })();

Document: comes under the windows object and can also be considered as its property.  
Window: in JS is a global object that holds the structure like variables, functions, location etc

Undefined: variable has been declared, but a value has not been assigned to that variable.  
Undeclared: variables that are not declared or that do not exist in program or applications.  
Null: is actually an assignment value that we can assign to any variable that is meant to contain no value.