1. What is !important?
2. The not important rule in css is used to add more impotance to a property / value than normal. In fact, if you use the !important rule, it will override all previous styling rules for that specific property on that element!

2. What is the box model?

1. In CSS, the term "box model" is used when talking about design and layout.The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content.  
   Content - The content of the box, where text and images appear

Padding - Clears an area around the content. The padding is transparent

Border - A border that goes around the padding and content

Margin - Clears an area outside the border. The margin is transparent  
  
The total width of an element should be calculated like this:  
  
Total element width = width + left padding + right padding + left border + right border + left margin + right margin  
  
The total height of an element should be calculated like this:  
  
Total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

3. What is the difference between position absolute, relative and fixed?  
 A. The position propety sets how an element is positioned in a document . The top, left, bottom, right determines the final location of the positioned element  
**Relative :** The element is positioned according to the normal flow of the document, and then offset *relative to itself* based on the values of top, right, bottom, and left. The offset does not affect the position of any other elements; thus, the space given for the element in the page layout is the same as if position were static.  
  
**Absolute :** The element is removed from the normal document flow, and no space is created for the element in the page layout. It is positioned relative to its closest positioned ancestor, if any; otherwise, it is placed relative to the initial [containing block](https://developer.mozilla.org/en-US/docs/Web/CSS/Containing_block). Its final position is determined by the values of top, right, bottom, and left.

This value creates a new [stacking context](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Positioning/Understanding_z_index/The_stacking_context) when the value of z-index is not auto. The margins of absolutely positioned boxes do not [collapse](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Box_Model/Mastering_margin_collapsing) with other margins.  
  
**Fixed** : The element is removed from the normal document flow, and no space is created for the element in the page layout. It is positioned relative to the initial [containing block](https://developer.mozilla.org/en-US/docs/Web/CSS/Containing_block) established by the [viewport](https://developer.mozilla.org/en-US/docs/Glossary/Viewport).Its final position is determined by the values of top, right, bottom, and left.

This value always creates a new [stacking context](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Positioning/Understanding_z_index/The_stacking_context). In printed documents, the element is placed in the same position on *every page*.

4. What is CSS positioning?  
A. The position property specifies the type of positioning method used for an element (static, relative, fixed, absolute or sticky).Elements are then positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first. They also work differently depending on the position value.  
  
**Static :** HTML elements are positioned static by default.Static positioned elements are not affected by the top, bottom, left, and right properties.An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page.  
  
**Relative :** The element is positioned according to the normal flow of the document, and then offset *relative to itself* based on the values of top, right, bottom, and left. The offset does not affect the position of any other elements; thus, the space given for the element in the page layout is the same as if position were static.  
  
**Absolute :** The element is removed from the normal document flow, and no space is created for the element in the page layout. It is positioned relative to its closest positioned ancestor, if any; otherwise, it is placed relative to the initial [containing block](https://developer.mozilla.org/en-US/docs/Web/CSS/Containing_block). Its final position is determined by the values of top, right, bottom, and left.

This value creates a new [stacking context](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Positioning/Understanding_z_index/The_stacking_context) when the value of z-index is not auto. The margins of absolutely positioned boxes do not [collapse](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Box_Model/Mastering_margin_collapsing) with other margins.  
  
**Fixed** : The element is removed from the normal document flow, and no space is created for the element in the page layout. It is positioned relative to the initial [containing block](https://developer.mozilla.org/en-US/docs/Web/CSS/Containing_block) established by the [viewport](https://developer.mozilla.org/en-US/docs/Glossary/Viewport).Its final position is determined by the values of top, right, bottom, and left.

5. the difference between display none and display hidden  
A. display:none means that the tag in question will not appear on the page at all (although you can still interact with it through the dom). There will be no space allocated for it between the other tags.  
Display :hidden means that unlike display:none, the tag is not visible, but space is allocated for it on the page. The tag is rendered, it just isn't seen on the page.

6. Difference between flex and grid? Where to use it?  
A. [Flexbox](https://www.geeksforgeeks.org/introduction-to-css-flexbox/)**:** The CSS Flexbox offers a one-dimensional layout. It is helpful in allocating and aligning the space among items in a container (made of grids). It works with all kinds of display devices and screen sizes.To get started you have to define a container element as a grid with **display: flex;**[Grid](https://www.geeksforgeeks.org/css-grid-property/):CSS Grid Layout, is a two-dimensional grid-based layout system with rows and columns, making it easier to design web pages without having to use floats and positioning. Like tables, grid layout allow us to align elements into columns and rows.

To get started you have to define a container element as a grid with **display: grid**, set the column and row sizes with grid-template-columns and grid-template-rows, and then place its child elements into the grid with grid-column and grid-row.

7. What is BEM?  
A. BEM (which stands for Block-Element-Modifier) is a naming convention standard for CSS class names. It has fairly wide adoption and is immensely useful in writing CSS that is easier to read, understand, and scale.  
There are three main parts of BEM.  
**Block** which holds everything (elements) inside and acts as a scope.  
**Element** which acts as a specific part of the component.  
**Modifier** which adds additional styles to a specific element(s).

8. What is the z-index? What is the stacking context?  
A. The **stacking order** describes the order in which HTML elements are positioned. By default, HTML elements are positioned in the following order:

* Root element(<html>)
* Non-positioned elements in the order they’re defined(elements with no position described i.e. static)
* Positioned elements in the order they are defined(elements with  
  position other than static)

**z-index:** In order to change the stacking order, we can use **z-index**. Element with higher z-index is placed on top of the element with lower z-index. Let us use the same. An important thing to note is that in order to use z-index, elements should be positioned**.  
  
9. Explain briefly what happens when you hit a url? explain DNS  
Lookup  
A.**[**URL**](https://www.geeksforgeeks.org/url-full-form/) **stands for Uniform Resource Locator. URL is the address of the website which you can find in the address bar of your web browser. It is a reference to a resource on the internet, be it images, hypertext pages, audio/video files, etc  
DNS is short for Domain Name System. Like a phonebook, DNS maintains and maps the name of the website, i.e. URL, and particular IP address it links to. Every URL on the internet has a unique IP address which is of the computer which hosts the server of the website requested.  
Steps for what happens when we enter a URL :  
- Browser checks cache for DNS entry to find the corresponding** [**IP address**](https://www.geeksforgeeks.org/introduction-of-classful-ip-addressing/) **of website.  
It looks for following cache. If not found in one, then continues checking to the next until found.  
Browser Cache  
Operating Systems Cache  
Router Cache  
ISP Cache  
- If not found in cache, ISP’s (Internet Service Provider) DNS server initiates a DNS query to find IP address of server that hosts the domain name.  
The requests are sent using small data packets that contain information content of request and IP address it is destined for.  
- Browser initiates a** [**TCP (Transfer Control Protocol)**](https://www.geeksforgeeks.org/tcp-and-udp-in-transport-layer/) **connection with the server using synchronize(SYN) and acknowledge(ACK) messages.  
- Browser sends an** [**HTTP**](https://www.geeksforgeeks.org/http-non-persistent-persistent-connection/) **request to the web server. GET or POST request.  
- Server on the host computer handles that request and sends back a response. It assembles a response in some format like JSON,** [**XML**](https://www.geeksforgeeks.org/xml-basics/) **and HTML.  
- Server sends out an HTTP response along with the status of response.  
- Browser displays** [**HTML**](https://www.geeksforgeeks.org/html-tutorials/) **content**

**10. What is a URLs full form? Explain what a url is and the four parts it consists of The protocol in use The hostname of the server The location of the file The arguments to the file.  
A.** [**URL**](https://www.geeksforgeeks.org/url-full-form/) **stands for Uniform Resource Locator.URL is the address of the website which you can find in the address bar of your web browser. It is a reference to a resource on the internet, be it images, hypertext pages, audio/video files, etc  
- The first part of the URL is the *scheme*, which indicates the protocol that the browser must use to request the resource (a protocol is a set method for exchanging or transferring data around a computer network). Usually for websites the protocol is HTTPS or HTTP (its unsecured version). Ex : https://**[**www.google.com**](http://www.google.com) **- Host Name : The unique reference that represent a webpage . Example :** [**www.google.com**](http://www.google.com) **- Port name : Usually not visible in URLS. Always following a colon : port 80 is the default port of web server.** [**www.google.com**](http://www.google.com)**:80  
- Path : A path refers to a file or location on the web server .   
Example : https://**[**www.google.com**](http://www.google.com)**:80//path/to/myfile.html**11.What is HTTP protocol?  
A. HTTP is a [protocol](https://developer.mozilla.org/en-US/docs/Glossary/Protocol) for fetching resources such as HTML documents. It is the foundation of any data exchange on the Web and it is a client-server protocol, which means requests are initiated by the recipient, usually the Web browser. A complete document is reconstructed from the different sub-documents fetched, for instance, text, layout description, images, videos, scripts, and more.Clients and servers communicate by exchanging individual messages (as opposed to a stream of data). The messages sent by the client, usually a Web browser, are called *requests* and the messages sent by the server as an answer are called *responses*.  
  
12. What is TCP Protocol?  
A. TCP stands for **Transmission Control Protocol**. It is a transport layer protocol that facilitates the transmission of packets from source to destination. It is a connection-oriented protocol that means it establishes the connection prior to the communication that occurs between the computing devices in a network. This protocol is used with an [IP](https://www.javatpoint.com/ip-full-form) protocol, so together, they are referred to as a [TCP/IP](https://www.javatpoint.com/tcp-ip-full-form).  
The main functionality of the TCP is to take the data from the application layer. Then it divides the data into a several packets, provides numbering to these packets, and finally transmits these packets to the destination. The TCP, on the other side, will reassemble the packets and transmits them to the application layer. As we know that TCP is a connection-oriented protocol, so the connection will remain established until the communication is not completed between the sender and the receiver.

13.Explain all the different HTTP methods?  
A. **GET :** The GET method requests a representation of the specified resource. Requests using GET should only retrieve data.  
HEAD : The HEAD method asks for a response identical to a GET request, but without the response body.  
POST : The POST method submits an entity to the specified resource, often causing a change in state or side effects on the server.  
PUT : The PUT method replaces all current representations of the target resource with the request payload.  
DELETE : The DELETE method deletes the specified target resource.  
PATCHT : he PATCH method applies partial modifications to a resource.  
[CONNECT](https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/CONNECT) : The CONNECT method establishes a tunnel to the server identified by the target resource.  
[OPTIONS](https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/OPTIONS) : The OPTIONS method describes the communication options for the target resource.  
[TRACE](https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/TRACE) : The TRACE method performs a message loop-back test along the path to the target resource.

14. What are HTTP headers?  
A. The **HTTP headers** are used to pass additional information between the  
clients and the server through the **request** and **response** header. All the headers are case-insensitive, headers fields are separated by colon, key-value pairs in clear-text string format. The end of the header section denoted by an empty field header. There are a few header fields that can contain the comments. And a few headers can contain quality(q) key-value pairs that separated by an equal sign. **There are four kinds of headers context-wise:**   
**General Header:** This type of headers applied on Request and Response headers both but with out affecting the database body.  
**Request Header:** This type of headers contains information about the fetched request by the client.  
**Response Header:** This type of headers contains the location of the source that has been requested by the client.  
**Entity Header:** This type of headers contains the information about the body of the resources like MIME type, Content-length.

15. What are some HTTP response codes? what does it mean? 2xx, 3xx, 4xx, 5xx  
A. HTTP response status codes indicate whether a specific [HTTP](https://developer.mozilla.org/en-US/docs/Web/HTTP) request has been successfully completed. Responses are grouped in five classes: 1xx,2xx, 3xx, 4xx, 5xx  
1xx - Information responses - the request was received, continuing process  
2xx - Successful responses - the request was successfully received, understood, and accepted  
3xx - Redirection message - further action needs to be taken in order to complete the request  
4xx - Client error response - the request contains bad syntax or cannot be fulfilled  
5xx - Server error response - the server failed to fulfil an apparently valid request

16. What does cache control on http response mean?  
A. The **Cache-Control header** is a general header, that specifies the caching policies of server responses as well as client requests. Basically, it gives information about the manner in which a particular resource is cached, location of the cached resource, and its maximum age attained before getting expired i.e. time to live.  
**Supported Browsers:** The browsers are compatible with **HTTP Cache-Control header** are listed below:  
Google Chrome, Edge, Firefox, Internet Explorer, Opera,Safari.

17. What is polling?  
A.Polling is a mechanism used by the push technology whereby a request is send by the client to the server at regular intervals. In return the server updates the status of connected client . Between two polling request the server stores the updates sent to the client in its memory until the next polling request is received.

18. What is long polling?  
A.**HTTP Long Polling** is a technique used to push information to a client as soon as possible on the server. As a result, the server does not have to wait for the client to send a request.  
In Long Polling, the server does not close the connection once it receives a request from the client. Instead, the server responds only if any new message is available or if a timeout threshold is reached.  
Once the client receives a response, it immediately sends a new request to the server to have a new pending connection to send data to the client, and the operation is repeated. With this approach, the server emulates a Realtime Server Push feature.

19. What are web sockets?  
A. **WebSocket:** WebSocket is bidirectional, a full-duplex protocol that is used in the same scenario of client-server communication, unlike HTTP it starts from **ws://** or **wss://**. It is a stateful protocol, which means the connection between client and server will keep alive until it is terminated by either party (client or server). After closing the connection by either of the client and server, the connection is terminated from both ends.   
Uses :- Real-time web application, Gaming application, Chat application:  
20. How is web sockets different from HTTP?  
A. WebSocket is a bidirectional communication protocol that can send the data from the client to the server or from the server to the client by reusing the established connection channel. The connection is kept alive until terminated by either the client or the server. **Whereas** The HTTP protocol is a unidirectional protocol that works on top of TCP protocol which is a connection-oriented transport layer protocol, we can create the connection by using HTTP request methods after getting the response HTTP connection get closed.  
- Almost all the real-time applications like (trading, monitoring, notification) services use WebSocket to receive the data on a single communication channel. **Whereas** Simple RESTful application uses HTTP protocol which is stateless.  
- All the frequently updated applications used WebSocket because it is faster than HTTP Connection. **Whereas** When we do not want to retain a connection for a particular amount of time or reuse the connection for transmitting data; An HTTP connection is slower than WebSockets.

21. What is HTTPS?  
A. Hypertext transfer protocol secure (HTTPS) is the secure version of [HTTP](https://www.cloudflare.com/learning/ddos/glossary/hypertext-transfer-protocol-http/), which is the primary protocol used to send data between a web browser and a website. HTTPS is encrypted in order to increase security of data transfer. This is particularly important when users transmit sensitive data, such as by logging into a bank account, email service, or health insurance provider.

22. What is Cross Origin Resource Sharing? ( CORS ) Why do we need it?  
A. Cross-Origin Resource Sharing ([CORS](https://developer.mozilla.org/en-US/docs/Glossary/CORS)) is an [HTTP](https://developer.mozilla.org/en-US/docs/Glossary/HTTP)-header based mechanism that allows a server to indicate any [origins](https://developer.mozilla.org/en-US/docs/Glossary/Origin) (domain, scheme, or port) other than its own from 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 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.

23. What does Access-Control-Allow-Origin header do ?  
A. The Access-Control-Allow-Origin header is included in the response from one website to a request originating from another website, and identifies the permitted origin of the request. A web browser compares the Access-Control-Allow-Origin with the requesting website's origin and permits access to the response if they match.  
**Syntax:**Access-Control-Allow-Origin: \* | <origin> | null  
**Directives:** Access-Control-Allow-Origin accepts there types of directives mentioned above and described below:  
**\*:** This directive tells the browsers to allow requesting code from any origin to access the resource. Used as a wildcard.  
**<origin>:** This directive defines any single origin.  
**null:** This directive defines null that should not be used due to any origin can create a hostile document with a “null” Origin. The “null” value for the **ACAO(Access-Control-Allow-Origin)** header should therefore, be avoided.”

24. What is clickjacking? How do you fix it?  
A. Clickjacking is an attack that fools users into thinking they are clicking on one thing when they are actually clicking on another. Its other name, user interface (UI) redressing, better describes what is going on. Users think they are using a web page’s normal UI, but in fact there is a hidden UI in control; in other words, the UI has been redressed. When users click something they think is safe, the hidden UI performs a different action.  
A better approach to prevent clickjacking attacks is to ask the browser to block any attempt to load your website within an iframe. You can do it by sending the X-Frame-Options HTTP header.

25. What are some performance metrics for your website?  
A. Page Speed. Page speed is one of the most important metrics to be aware of, and it can have a far greater impact than you may even realize. …  
Time to Title. …  
Time to Start Render. …  
Time to Interact. …  
DNS Lookup Speed. …  
Bounce Rate. …  
Requests Per Second. …  
Error Rate.

26. What does the following term mean?  
Time to first byte,  
Page load time  
A. Time to First Byte (TTFB) refers to the time between the browser requesting a page and when it receives the first byte of information from the server. This time includes [DNS](https://developer.mozilla.org/en-US/docs/Glossary/DNS) lookup and establishing the connection using a [TCP](https://developer.mozilla.org/en-US/docs/Glossary/TCP) handshake and [SSL](https://developer.mozilla.org/en-US/docs/Glossary/SSL) handshake if the request is made over [https](https://developer.mozilla.org/en-US/docs/Glossary/https).  
TTFB is the time it takes between the start of the request and the start of the response, in milliseconds:  
TTFB = responseStart - navigationStart  
- In its simplest terms, page load time is **the average amount of time it takes for a page to show up on your screen**. It's calculated from initiation (when you click on a page link or type in a Web address) to completion (when the page is fully loaded in the browser).

27. What do CDN or Content Delivery Networks do? When do you need a CDN?  
A. A CDN (Content Delivery Network) is a group of servers spread out over many locations. These servers store duplicate copies of data so that servers can fulfill data requests based on which servers are closest to the respective end-users. CDNs make for fast service less affected by high traffic.  
You should seriously consider a CDN if:  
- Your site is growing at an exponential rate – A regular server is going to have a tough time handling a spike in traffic, unless you prepare for it with a CDN.  
- Large files and static content are dragging your site down – Items like large images, scripts, CSS and other elements can prevent your site from loading quickly. This is particularly the case for online magazines and sites that use tons of media.  
- Your audience is starting to expand to a worldwide crowd – If you notice that many site visitors are coming from the other side of the world, or even just in a different country, it may be time to look into a CDN.

28. What is the difference between Client Side Renderring and Server Side Renderring?  
A. In SSR, when the user makes a request to the webpage, the server prepares the HTML page by fetching the required data from the database and sends to the user's machine over the internet. Then the browser presents all the requested actions on the user UI. All these processes of fetching data from the database to creating an HTML page and sending it to the client are done in mere milliseconds.

Client-side rendering means that a website’s JavaScript is rendered in your browser, rather than on the website’s server. So now, instead of getting all the content from the HTML doc, only the required HTML with the JS files will be rendered. The rendering time for the first upload is a bit slow. However, the next page loads will be very fast as we don't have to wait for every page render. Moreover, there is no need to reload the entire UI after every call to the server. The client-side framework manages to update UI with changed data by re-rendering only that particular DOM element.

30. What is Progressive Renderring  
A . Progressive Server-Side Rendering (PSSR) is based on the concept of HTML streaming. PSSR breaks the pages into meaningful components using code splitting. Those pieces of the page are controlled by a separate script, and now we have the opportunity to hydrate them independently based on some priority that we’ve determined earlier. ( or ) Progressive Rendering is the technique of sequentially rendering portions of a webpage in the server and streaming it to the client in parts without waiting for the whole page to rendered.

31.What is the difference between Preloading and Prefetching resources?  
A. <link rel="preload"> tells the browser to download and cache a resource (like a script or a stylesheet) as soon as possible. It’s helpful when you need that resource a few seconds after loading the page, and you want to speed it up.  
The browser doesn’t do anything with the resource after downloading it. Scripts aren’t executed, stylesheets aren’t applied. It’s just cached – so that when something else needs it, it’s available immediately.

<link rel="prefetch"> asks the browser to download and cache a resource (like, a script or a stylesheet) in the background. The download happens with a low priority, so it doesn’t interfere with more important resources. It’s helpful when you know you’ll need that resource on a subsequent page, and you want to cache it ahead of time.

The browser doesn’t do anything with the resource after downloading it. Scripts aren’t executed, stylesheets aren’t applied. It’s just cached – so that when something else needs it, it’s available immediately.

32.What are service workers?  
A.**Service Worker:**A service worker is a script that runs independently in the browser background. On the user side, it can intercept its network requests and decide what to load (*fetch*).  
Service workers mainly serve features like background sync, push notifications and they are commonly used for’offline first’ applications, giving the developers the opportunity to take complete control over the user experience.  
Before it’s time there has been API called *AppCache*, which has been trying to serve the offline experience feature. However, there have been numerous problems in the interface of the *AppCache API* and Service Workers are here, going over them.

Video Link : <https://drive.google.com/file/d/1m23eJ-0do27D4bdB9BczNvZjSJTkQlAO/view?usp=sharing>  
<https://drive.google.com/file/d/1AiyPg0rHda5NpVa9JL6GKvwYoc_A8oTV/view?usp=sharing>