front-end Interview Question

Date: _____ Page: _____

Q Explain the difference b/w cookies, session storage & local storage?

① **Local Storage**:- way to store data on the client's computer. allows the saving of key/value pairs in a web browser & it store data with no expiration date. It is accessed via JS & HTML5
→ data limit → 5 MB
→ never transfer to the server

② **Session Storage**:- Stores data only for a session.
→ data never transfered to server.
→ 5 - 10 MB limit
→ opening multiple tab/window with the same URL creates (sessionStorage) for each tab/window.

③ **Cookies**:- Stores data that has to be sent back to the server with XHR requests.
→ Its Expiration varies based on the type & the expiration duration can be set from either server-side/client-side.
→ 4 KB less than Storage
→ Cookies can be made secure by setting the httpOnly flag as true for that cookie.

Date: _____ Page: _____

Q → what does CORS stand for & what issue does it address?

CORS → Cross Origin - request Sharing

It helps to allow ~~limited~~ access of content from limited origins.

Q what does a doctype do?

It is an "information" to the browser about what document type to expect

→ It is NOT case sensitive.

Q what are data-* attributes good for?

Data attribute allow you to add your own info to tags in HTML.

mainly 2-part

① Attribute-Name: at least one char long, no capital letters & prefixed with 'data-'.

② Attribute value → Can be any string.

Q Describe the difference b/w <script>, <script async> & <script defer>.

① <Script> → HTML Parsing is blocked, the script is fetched & executed immediately, HTML parsing resume after the script is executed.

Date: Page:

② <script async> → The script will be fetched in parallel to HTML parsing & executed as soon as it is available.

③ <Script defer> → The script will be fetched in parallel to HTML parsing & executed when the page has finished parsing. If there are multiple of them, each deferred script is executed in the order they were encountered in the document

Q what is the difference b/w Canvas & SVG?

SVG is a language for describing 2D graphics in XML.

Canvas draws 2D graphics, on the fly (with a Java script).

CSS

Q what is CSS selector specificity & how does it work?

If there are two/more conflicting css rules that point to the same element, the browser follows some rules to determine which one is most specific & therefore wins out.

Inline Style > Internal Style > External Style

Inside External style sheet the specificity is:

ID Selector > Class selector or Attr. Selector > Type selector or pseudo-element

Date: _____ Page: _____

Q what is the difference b/w "resetting" and "normalizing" css?

CSS reset : Aim to remove all built-in browser styling. (Having No decoration at all)

Normalize CSS : aims to make built in browser styling consistent across browsers.

Q Describe float How they work?

Float is a CSS property for positioning. It place an element on the left / right side of its container, allowing text & inline element to wrap around it.
The element is removed from the normal flow of the page.

Q How would you approach fixing browser-specific styling issues?

⊛ After identifying the issues & the offending browser, use a seperate Style Sheet that only loads when that specific browser is being used. This technique requires Server-side redering though.

⊛ Use libraries like bootstrap that already handles these styling.

⊛ Use autoprefixer to automatically add vender prefixes to your code.

Date: _____ Page: ____

Q How to optimize your webpage for print?

- Create a stylesheet for print or use media queries.
- if using seperate stylesheet for print attach it at bottom of HTML.
- Make sure to put non-print style inside @media screen { --- }.

Q What are the advantages/disad. of using CSS preprocessors?

Advantages

i) CSS is made more maintainable
ii) Easy to write nested selectors.
iii) Can share theme file across different projects.
iv) Splitting your code into multiple file.

disadvantage:-

(i) Require tools for preprocessing.

Q Describe Pseudo-elements and discuss what they are used for.

A CSS Pseudo-element is a keyword added to a selector that lets you style a specific part of a selected element(s).

- use to decorate or adding element to the markup (Combined with content:--) without having to modify the markup

{ ::before ::after, ::first-line }
::first-letter,

Date: _____ Page: _____

Q Explain your understanding of the box model.

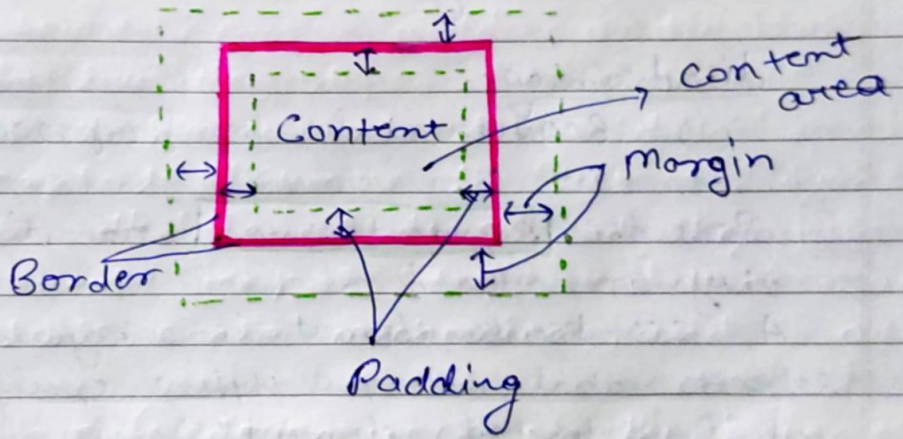
The CSS box model is responsible for calculating how much space a block-level element takes up.

→ also responsible for calculating box's dimensions.

→ The dimensions of a block element are calculated by width, height, padding, borders & margin(s)

→ If no height is specified, a block element will be as high as the content it contains, plus padding.

→ If no width is specified, a non-floated block element will expand to fit the width of its parent minus padding.



→ By default, elements have box-sizing: content-box and only the content size being accounted for.

→ box-sizing: border-box

height = content height + vertical padd + vertical box

width = content width + hori. padding + hori. borders.

Date: Page:

Q List values for display property?

none, block, inline-block, inline, table, flex, grid.

Q what is the difference b/w ~~inline~~ block & inline-block?

	BLOCK	Inline-block
Size:-	fills up the width of its parent container	Depends upon content.
position	Start on a new line	flows along with other content & allow other elements beside.
	Here also we are allowed to set width as well as height.	we are allowed to set width & height

Q what's the difference b/w a relative, fixed absolute & statically positioned elements?

① Static :- It is the default position; the flow of element on the webpage is normal. The top, right, bottom, left & z-index properties do not apply.

② Relative :- The ~~default position~~ The element's position is adjusted relative to itself, without changing layout.

Date: _____ Page: _____

(3) absolute :- The element is removed from the flow of the page and positioned at a specified position relative to its closest positioned ancestor if any, or otherwise relative to the intial containing block.

(4) fixed :- The element is removed from the flow of the page and positioned at a specified position relative to the viewport & doesn't move when scrolled.

(5) Sticky :- Sticky positioning is a hybrid of relative & fixed positioning. The element is treated as relative positioned until it crosses a specified threshold, at which point it is treated as fixed positioned.

Q How is responsive design different from adaptive design?

Both responsive and adaptive deg design attempt to optimize the user experience across different devices, adjusting for different viewport sizes, resolutions, usage contexts, control mechanisms, and so on.

Responsive design works on the Principle of flexiblity - a single fluid webpage that can look good on any device. Responsive webpages use media queries, flexible grids, & responsive images to create a good user experience that flexes & changes based on

a multitude of factors.

Adaptive design is more like the modern definition of progressive enhancement. Instead of one flexible design, adaptive design detects the device and other features, and then provides the appropriate feature and layout based on a predefined set of ~~view~~ viewport sizes and other characteristics.

Q Difference b/w px, em and rem as they relate to font sizing?

① px :- Pixels are the easiest measurement to use. But there is a disadvantage of using px. Let's say we used px throughout our website & we managed the media queries too. What if user changed the default font-size of device (or browser)? Your header's font-size (say 24px) will remain Same.

② em :- em is ~~Computed~~ Equal to the Computed font-size of that element's parent.

③ rem :- rem values are related to the root html element, not to the parent element

Date: _____ Page: _____

Q what is pseudo-classes? give few examples?

Pseudo class selector are CSS selector with a colon preceding them.

Eg:-

```
a:hover {
    /* hover is a pseudo class*/
}
```

They are immensely useful in variety of situations.

:visited, :hover, :active, :focus, :enable, :checked, :first child, :last-child etc.

JavaScript

Q Explain event delegation?

Event delegation is a technique involving adding event listeners to a parent element instead of adding them to the descendant elements. This listener will fire whenever the event is triggered on the descendant elements due to event bubbling up the DOM. The main benefit for this is Let's say if we add new child to some parent then we have to not remove & bind event listeners to child element when we bind event with parent.

Date: _____ Page: _____

Q Explain how This works in JS?

There's is no simple explanation for "this". It is one of the most confusing concepts in javascript to grab.
The following rules are applied:-

① If the "new" keyword is used when calling the function, "this" inside the function is a brand new object.

② If "apply, call or bind" are used to call/create a function, "this" inside the function is the object that is passed in as the argument.

③ If a function is called as a method, such as (obj.method()) – this is the object that the function is a property of.

(4) If a function is invoked as a free function invocation, meaning it was invoked without any of the conditions present above, (this) is the global object. In a browser, it is the window object. If in strict mode ('use strict'), this will be undefined instead of the global object.

(5) If multiple of the above rules apply, the rule that is higher wins and will set the 'this' value.

Date: _____ Page: _____

Q Explain how prototypal inheritance works?

All Javascript objects have a "prototype" property, that is a reference to another object. When a property is accessed on an object and if the property is not found on that object, the Javascript engine looks at the object's "prototype" and the "prototype's prototype and so on, until it finds the property on one of the prototype's or until it reaches the end of the prototype chain.

Q Explain why the following doesn't work as an IIFE: function foo(){}(); what needs to be changed to properly make it an IIFE?

IIFE stands for Immediately Invoked function Expressions. The Javascript parser reads function foo(){}(); as function foo(){} & (); where the former is a function declaration & the latter (a pair of brackets) is an attempt at calling a function but there is no name specified, hence it throws (Uncaught SyntaxError: Unexpected token).

Here are two ways to fix it that involves adding more brackets:

(i) (function foo(){})();

ii) (function foo() { } ()).

These function are not exposed in the global scope and you can even omit its ~~nam~~ name if you do not need to reference it self within the body.

Assigning the IIFE to a variable stores the function's return value, not the function definition it self.

```
var result = (function () {
    var name = "Barry";
    return name
}) ();

Console.log(result); // Barry
```

Q what's the difference b/w a variable that is null, undefined, or Undeclared? How would you go about checking for any of these states?

① Undeclared:- They are created when we assign to a value to an identifier that is not previously created using var, let or const. In strict mode, a ReferenceError will be thrown when you try to assign to an undeclared variable.

```
function foo() {
    x = 1;      // Throws a Reference Error in strict mood.
}
foo()
console.log(x) // 1
```

Date: _____ Page: _____

(2) Undefined:- A variable that is undefined is a variable that has been declared, but not assigned a value. It is of type undefined. To check for it, compare using the strict equality (===) operator. or typeof which will give the 'undefined' string.

```
var x;
Consol.log (x); // Undefined
Consol.log( x === undefined); // True
Consol.log (typeof x === 'undefined'); // True.
```

(3) NULL:- A variable that is null will have been explicitly assigned to the null value. It represent no value and is different from undefined in the sense that it has been explicitly assigned. To check for null, simply compare using the strict equality operator.

Q Difference between : function Person(){},
var person = Person(), &
var person = new Person();

My best guess at its intention is that it is asking about constructors in Javascript. function Person(){} is just a normal func" declration.

var person = Person() invokes the Person as a function, and not as a constructor.

Date: _____ Page: _____

Var person = new Person() creates an instance of the Person object using the new operator, which inherits from person.prototype

Q what's the difference b/w .call & .apply?

Both .call & .apply are used to invoke functions and the first parameter will be used as the value of 'this' within the function. However, .call takes in a comma-separated arguments as the next arguments while .apply takes in an array of arguments as the next argument

```
function add(a,b) {
    return a+b;
}
console.log(add.call(null,1,2)) //3
console.log(add.apply(null,[1,2])) //3
```

Q Explain function.Prototype.bind.

The bind() method creates a new function that, when called, has its this keyword set to the provided value, with a given sequence of arguments preceding any provided when the new func^{n} is called.

Date: _____ Page: _____

Q What is the difference b/w == & ===?

== is the abstract equality operator while === is the strict equality operator. The == operator will compare for equality after doing any necessary type conversion. The === operator will not do type conversion, so if two values are not the same type === will simply return false. when using ==, funky things can happen, ~~such as~~

Q Why it is called Ternary expression, what does the word "Ternary" indicate?

"Ternary" indicates three & a ternery expression accepts three operands, the test condition, the "then" expression & the "else" expression.

Q What is "use strict"; ? what are the advantages and disadvantages to using it?

"use strict" is a statement used to enable strict mode to entire scripts or individual functions. Strict mode is a way to opt in to a restricted variant of JS.

Advantages:-

→ Makes it impossible to accidentally create global variable.

→ Require that function parameters names be unique.

→ 'this' is undefined in the global context.

Disadvantages:-

→ Many missing features that some developers might be used to.
→ No more access to function.caller & function.arguments.

Q What is event loop? what is the difference b/w call stack & task queue?

→ As we know Javascript is single-threaded programming language.

The event loop is a single-threaded loop that monitors the call stack and checks is any work to be done in the task queue. If the call stack is empty and there are callback functions in the task queue, a function is dequeued & pushed onto the call stack to be executed.

Q Explain the difference on the usage of foo b/w function foo() {} and var foo = function() {}

function foo() {} is a function declration while the latter is a function expression. The key difference is that function expressions are not hoisted (they have the same hoisting behaviour as variables)

Date: _____ Page: _____

If you try to invoke a function expression before it is defined, you will get an Uncaught TypeError: xxx is not a function error.

Function Declration:-

```
foo();  // o/p → 'Foooo'
function foo() {
    console.log('Foooo');
}
```

Function Expression :-

```
foo();//o/p → Uncaught TypeError:
var foo = function() {
    console.log('Foooo');
}
```

Q what is a ~~class~~ closure, and how/why would you use one?

Closures are frequently used in JavaScript for object data privacy, in event handlers & callback functions,

A closure is the combination of a func bundled together (enclosed) with referen to its surrounding state (the lexical environment). In other words, a closure give you access to outer function's scope from a inner function. In JavaScript, closures are cre every time a function is created, at function creation tim