**Important Links:**

https://www.youtube.com/watch?v=057Rs6CgJnY advanced-javascript -interview-question

Link1 Link2 Link3 Link4 Link5 Link6 Link7 Link8

**What is JavaScript?**

JavaScript is a client-side as well as server-side scripting language that can be inserted into HTML pages and is understood by web browsers. JavaScript is also an Object based Programming language.

**What are JavaScript Data Types?**

Following are the JavaScript Data types:

- Number
- String
- Boolean
- Object
- Undefined

**What is the use of isNaN function?**

isNan function returns true if the argument is not a number otherwise it is false.

**What is the data type of variables of in JavaScript?**

All variables in the JavaScript are object data types.

**What is negative infinity?**

Negative Infinity is a number in JavaScript which can be derived by dividing negative number by zero.

**What is 'this' keyword in JavaScript?**

'This' keyword refers to the object from where it was called.

**Explain the working of timers in JavaScript? Also elucidate the drawbacks of using the timer, if any?**

Timers are used to execute a piece of code at a set time or also to repeat the code in a given interval of time. This is done by using the functions **setTimeout, setInterval** and **clearInterval**.

The **setTimeout (function, delay)** function is used to start a timer that calls a particular function after the mentioned delay. The **setInterval (function, delay)** function is used to repeatedly execute the given function in the mentioned delay and only halts when cancelled. The **clearInterval (id)**function instructs the timer to stop.

Timers are operated within a single thread, and thus events might queue up, waiting to be executed.

**What is === operator?**

=== is called as strict equality operator which returns true when the two operands are having the same value without any type conversion.

**What are all the looping structures in JavaScript?**

Following are looping structures in Javascript:

- For
- While
- do-while loops

**What is called Variable typing in Javascript?**

Javascript is dynamically typed. This means that in Javascript variables can be reassigned to values of any type, and thus that you don't ever need to explicitly denote the type of variables or the return type of functions. Just declare a value with the datatype "var". JavaScript automatically adjusts the datatype while assigning or reassigning the variable.

**How can you convert the string of any base to integer in JavaScript?**

The parseInt() function is used to convert numbers between different bases. parseInt() takes the string to be converted as its first parameter, and the second parameter is the base of the given string.

In order to convert 4F (of base 16) to integer, the code used will be -

```
parseInt ("4F", 16);
```

**Explain the difference between "==" and "==="?**

"==" checks only for equality in value whereas "===" is a stricter equality test and returns false if either the value or the type of the two variables are different.

**What would be the result of 3+2+"7" & “7”+3+2 & 3+“7”+2?**

Since 3 and 2 are integers, they will be added numerically. And since 7 is a string, its concatenation will be done. So, the result would be 57. Similarly, 732 and 372 for other 2.

**What do mean by NULL in Javascript?**

The NULL value is used to represent no value or no object. It implies no object or null string, no valid boolean value, no number and no array object.

**What is the function of delete operator?**

The delete keyword is used to delete the property as well as its value.

Example

```
var student= {age:20, batch:"ABC"};
delete student.age;
```

**What is an undefined value in JavaScript?**

Undefined value means the

- Variable used in the code doesn't exist
- Variable is not assigned to any value
- Property doesn't exist

What is the use of Void(0)**?**

The void **operator** evaluates the given *expression* and then returns undefined.

What are escape characters**?**

Escapes or unescapes a JavaScript string removing traces of offending characters that could prevent interpretation.

What are JavaScript Cookies**?**

Cookies are the small test files stored in a computer and it gets created when the user visits the websites to store information that they need. Example could be User Name details and shopping cart information from the previous visits.

What is push(), pop(), shift() and unshift() method in JavaScript**?**

The push() method can append one or more elements to the end of an array. The pop() method pulls the last element off of the given array and returns it. The unshift() method is like the push() method, only it works at the beginning of the array. The unshift() method can prepend one or more elements to the beginning of an array. The shift() method is like the pop() method, only it works at the beginning of the array. The shift() method pulls the first element off of the given array and returns it.

What is scope**?**

Scope refers to the current context of your code. Scopes can
be globally or locally defined.

**What is break and continue statements?**

Break statement exits from the current loop.

Continue statement continues with next statement of the loop.

**What is the use of type of operator?**

'Typeof' is an operator which is used to return a string description of the type of a variable.

## How JS handle exceptions?

Try… Catch---finally is used to handle exceptions in the JavaScript

```
Try{
        Code
}
Catch(exp){
        Code to throw an exception
}
Finally{
        Code runs either it finishes successfully or after catch
}
```

## What is the 'Strict' mode in JavaScript and how can it be enabled?

Strict Mode adds certain compulsions to JavaScript. Under the strict mode, JavaScript shows errors for a piece of codes, which did not show an error before, but might be problematic and potentially unsafe. Strict mode also solves some mistakes that hamper the JavaScript engines to work efficiently.

Strict mode can be enabled by adding the string literal "use strict" above the file. This can be illustrated by the given example:

```
function myfunction() {
    "use strict";
    var v = "This is a strict mode function";
}
```

**Explain window.onload and onDocumentReady?**

**window.onload** - Code in this method get executed when DOM tree is ready. But fact is code in this method executed when DOM tree is ready and All the external resource are load like Images, Flash vidoe or quicktime video. Loading of the external resources deplay execution of the actual script when page get displayed.
**onDocumentReady** - Code in this method get executed once DOM tree loading is done. That means it’s not wait for the external resource get loaded. And run the javascript code which is in function.

## Describe the function in JavaScript?

In JavaScript, functions are first-class objects, because they can have properties and methods just like any other object.

**Define event bubbling and event propagation?**

The concept of ***event bubbling*** was introduced to deal with situations where a single event, such as a mouse click, may be handled by two or more event handlers defined at different levels of the *Document Object Model* (DOM) hierarchy. If this is the case, the event bubbling process starts by executing the event handler defined for individual elements at the lowest level (e.g. individual hyperlinks, buttons, table cells etc.). From there, the event *bubbles up* to the containing elements (e.g. a table or a form with its own event handler), then up to even higher-level elements (e.g. the BODY element of the page). Finally, the event ends up being handled at the highest level in the DOM hierarchy, the document element itself (provided that your document has its own event handler).

The term ***event propagation*** is often used as a synonym of *event bubbling*. However, strictly speaking, event propagation is a wider term: it includes not only event bubbling but also event capturing. ***Event capturing*** is the opposite of bubbling (events are handled at higher levels first, then *sink* down to individual elements at lower levels).

**What are the decodeURI() and encodeURI()?**

EncodeURl() is used to convert URL into their hex coding. And DecodeURI() is used to convert the encoded URL back to normal.

```
<script>
        var uri="my test.asp?name=ståle&car=saab";

        document.write(encodeURI(uri)+ "<br>");

        document.write(decodeURI(uri));
</script>
```

Output -

my%20test.asp?name=st%C3%A5le&car=saab

my test.asp?name=ståle&car=saab

What is namespacing in JavaScript and how is it used**?**

Namespacing is used for grouping the desired functions, variables etc. under a unique name. It is a name that has been attached to the desired functions, objects and properties. This improves modularity in the coding and enables code reuse.

## Is JavaScript a case-sensitive language?

Yes, JavaScript is a **case sensitive** language. The language keywords, variables, function names, and any other identifiers must always be typed with a consistent capitalization of letters.

## What is the purpose of 'This' operator in JavaScript?

The JavaScript **this** keyword refers to the object it belongs to. This has different values depending on where it is used. In a method, this refers to the owner object and in a function, this refers to the global object.

## What is Callback?

A **callback** is a plain JavaScript function passed to some method as an argument or option. It is a function that is to be **executed** after another function has finished executing, hence the name '**call back**'. In JavaScript, functions are objects. Because of this, functions can take functions as arguments, and can be returned by other functions.

## What is Closure? Give an example.

**Closures** are created whenever a variable that is defined outside the **current scope** is accessed from within some inner scope. It gives you access to an outer function's scope from an inner function. In JavaScript, closures are created every time a function is created. To use a closure, simply define a function inside another function and expose it.

## How does TypeOf Operator work?

The **typeof** operator is used to get the data type of its operand. The operand can be either a **literal**or a **data structure** such as a variable, a function, or an object. It is a **unary** operator that is placed before its single operand, which can be of any type. Its value is a string indicating the data type of the operand.

## List out the different ways an HTML element can be accessed in a JavaScript code.

Here are the list of ways an HTML element can be accessed in a Javascript code:
(i) **getElementById('idname'):** Gets an element by its ID name

(ii) **getElementsByClass('classname'):** Gets all the elements that have the given classname.
(iii) **getElementsByTagName('tagname'):** Gets all the elements that have the given tag name.
(iv) **querySelector():** This function takes css style selector and returns the first selected element.

## What are the ways to define a variable in JavaScript?

The three possible ways of defining a variable in JavaScript are:

- **Var –** The JavaScript variables statement is used to declare a variable and, optionally, we can initialize the value of that variable. Example: var a =10; Variable declarations are processed before the execution of the code.
- **Const –** The idea of const functions is not allow them to modify the object on which they are called. When a function is declared as const, it can be called on any type of object.
- **Let –** It is a signal that the variable may be reassigned, such as a counter in a loop, or a value swap in an algorithm. It also signals that the variable will be used only in the block it's defined in.

## What is the difference between Local storage & Session storage?

**Local Storage** – The data is not sent back to the server for every HTTP request (HTML, images, JavaScript, CSS, etc) – reducing the amount of traffic between client and server. It will stay until it is manually cleared through settings or program.

**Session Storage** – It is similar to local storage; the only difference is while data stored in local storage has no expiration time, data stored in session storage gets cleared when the page session ends. Session Storage will leave when the browser is closed.

## What is the difference between null & undefined?

Undefined means a variable has been **declared** but has not yet been **assigned** a value. On the other hand, null is an assignment value. It can be assigned to a variable as a representation of no value. Also, undefined and null are two distinct types: undefined is a type itself (undefined) while null is an object.

## What is the difference between undeclared & undefined?

Undeclared variables are those that do not **exist** in a program and are not declared. If the program tries to read the value of an undeclared variable, then a **runtime error** is encountered. Undefined variables are those that are declared in the program but have not been given any value. If the program tries to read the value of an undefined variable, an undefined value is returned.