What is JavaScript?

JavaScript is the most popular programming language in the world. JavaScript is a client-side scripting language that can be inserted into HTML pages and is understood by web browsers. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed.

Who is the developer of JavaScript?

Netscape is the developer of JavaScript. Read full history on WikiPedia.

What is the difference between JavaScript and Jscript?

Both are almost similar. JavaScript is developed by Netscape and Jscript was developed by Microsoft.

How are JavaScript and ECMA Script related?

ECMAScript is nothing but another name for JavaScript. Precisely, ECMAScript is the formal name of JavaScript, when XML elements have to be accessed.

How can JavaScript codes be hidden from old browsers that don't support JavaScript?

For hiding JavaScript codes from old browsers add "" without the quotes in the code just before the tag.

Old browsers will now treat this JavaScript code as a long HTML comment. While, a browser that supports JavaScript, will take the "" as one-line comments.

What is DOM in JavaScript?

DOM stands for **Document Object Model** and is responsible for how various objects in a document interact with each other. DOM is required for developing web pages, which includes objects like paragraph, links, etc. These objects can be operated to include actions like add or delete. DOM is also required to add extra capabilities to a web page. On top of that, the use of API gives an advantage over other existing models.

Is JavaScript case sensitive?

Yes, JavaScript is case sensitive. For example, a function parseInt is not same as the function Parseint.

What are deferred scripts in JavaScript?

By default, the parsing of the HTML code, during page loading, is paused until the script has not stopped executing. It means, if the server is slow or the script is particularly heavy, then the webpage is displayed with a delay. While using Deferred, scripts delays execution of the script till the time HTML parser is running. This reduces the loading time of web pages and they get displayed faster.

What is a namespace in JavaScript?

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.

What are the various functional components in JavaScript?

The different functional components in JavaScript are-

First-class functions: Functions in JavaScript are utilized as first class objects. This usually means that these functions can be passed as arguments to other functions, returned as values from other functions, assigned to variables or can also be stored in data structures.

Nested functions: The functions, which are defined inside other functions, are called Nested functions. They are called ‘everytime’ the main function is invoked.

What is scope in JavaScript?

JavaScript has two scopes: global and local.

A variable that is declared outside a function definition is a **global variable**, and its value is accessible and modifiable throughout your program. A variable that is declared inside a function definition is **local variable**. It is created and destroyed every time the function is executed, and it cannot be accessed by any code outside the function.

The var keyword is used to declare a local variable or object. If the var keyword is omitted, an automatic global variable is declared.

Example:

```
// Declaring a global variable
globalVar = "I am a global variable and declared outside of the function";

// Declaring a local variable.
function myFunc(){
    var localVar = "I am inside function and local variable";
}
// Declaring an automatic global variable.
function myFunc(){
    autoGlobalVar = "I am an automatic global variable";
}
```

The problems that are faced by using global variables are the clash of variable names of local and global scope. Also, it is difficult to debug and test the code that relies on global variables.

Read more on:

- JavaScript Scope
- Variable Scope

Does JavaScript has concept level scope?

No. JavaScript does not have concept level scope. The variable declared inside the function has scope inside the function.

Which keyword is used to print the text in the screen?

document.write("your text") is used to print the text in screen.

What are JavaScript data types?

The JavaScript types are:

- Number
- String
- Boolean
- Function
- Object
- Null
- Undefined

What are the boolean operators used in JavaScript?

- The ‘And’ Operator (&&), ‘Or’ Operator (||) and the ‘Not’ Operator
- (!) can be used in JavaScript.
- *Operators are without the parenthesis.

What is the difference in between null and undefined in JavaScript?

In JavaScript, undefined means a variable has been declared but has not yet been assigned a value. For example:

```
var TestVar;
alert(TestVar); //shows undefined
alert(typeof TestVar); //shows undefined
```

null is an assignment value. It can be assigned to a variable as a representation of no value:

```
var TestVar = null;
alert(TestVar); //shows null
alert(typeof TestVar); //shows object
```

From the preceding examples, it is clear that undefined and null are two distinct types: undefined is a type itself (undefined) while null is an object.

Which symbol is used for comments in JavaScript?

Double slash "//" used for single line comments. Slash proceeding a star symbol and closed at the end with stat proceeding with slash used for multi-line comments..

Example

```
//I am a single line comment
/*
I am inside
a multi-line comment.
*/
```

What is the use of Void(0) in JavaScript?

- Void(0) is used to prevent the page from refreshing and parameter "zero" is passed while calling.
- Void(0) is used to call another method without refreshing the page.

Difference between == and === operator?

Literally, both are comparison operators and behaves same. (==) checks only for equality in value whereas (===) performs a stricter equality test.

A strict comparison operator (===) return true only if the operands are of the same type and value. There is no type conversion done, and the types must be the same to be considered equal.

The most commonly used equality operator (==) converts the operands to the same Type before making the comparison.

What would be the result of 3+2+"7"?

Result is 57. As 3 and 2 are integers, they will be added numerically. And since 7 is a string, its concatenation will be done.

How to change the style/class of an element?

This can be achieved by the following way:

```
document.getElementById(“elementID”).style.fontSize = “20px";
//or
document.getElementById(“elementID”).className = “myclass”;
```

Explain window.onload and onDocumentReady?

- The **window.onload()** method will not run until all the information on the page is loaded. This leads to a substantial delay before any code is executed.
- **onDocumentReady** loads the code just after the DOM is loaded. This allows early manipulation of the code.

What are all the looping structures in JavaScript?

Following are looping structures in JavaScript:

- For
- While
- do-while loops
- For each

What is break and continue statements?

- Break statement exits from the current loop.
- Continue statement continues with next statement of the loop.

What are all the types of Pop up boxes available in JavaScript?

- Alert
- Confirm
- Prompt

What is the difference between an alert box and a confirmation box?

- An alert box displays only one button which is the OK button.
- A Confirmation box displays two buttons namely OK and cancel.

What is a prompt box?

A prompt box is a box which allows the user to enter input by providing a text box. Label and box will be provided to enter the text or number.

What is the use of isNaN function?

The **isNan()** function returns true if the argument is not a number otherwise it is false. For one thing, although NaN means “not a number”, its type is, believe it or not, Number.

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 is used to point at the current object in the code. For instance: If the code is presently at an object created by the help of the ‘new’ keyword, then ‘this’ keyword will point to the object being created.

Explain the working of timers in JavaScript and it's drawbacks?

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 the difference between ViewState and SessionState?

ViewState is specific to a page in a session. **SessionState** is specific to user specific data that can be accessed across all pages in the web application.

What is called Variable typing in Javascript?

Variable typing is used to assign a number to a variable and the same variable can be assigned to a string.

Example

```
var i = 10;  i = "string";
```

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);
```

How to append a value to an array?

Add new value to an existing array:

```
var arr = [];
//or var arr = new Array();
arr.push(value);
arr[arr.length] = value;
```

What is the use of Push method in JavaScript?

The push method is used to add or append one or more elements to the end of an Array. Using this method, we can append multiple elements by passing multiple arguments

What is unshift method in JavaScript?

Unshift method is like push method which works at the beginning of the array. This method is used to prepend one or more elements to the beginning of the array.

What does the following statement declares?

```
var myArray = [[[]]];
```

It declares a three dimensional array.

What do you 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.

How are event handlers in JavaScript?

Events are the actions that result from activities, such as clicking a link or filling a form, by the user. An event handler is required to manage proper execution of all these events. Event handlers are an extra attribute of the object. This attribute includes event's name and the action taken if the event takes place.

What is the use of delete operator?

The functionality of delete operator is used to delete all variables and objects in a program but it cannot delete variables declared with VAR keyword.

What is the data type of variables of in JavaScript?

All variables in the JavaScript are object data types.

What are escape characters?

Escape characters (Backslash) is used when working with special characters like single quotes, double quotes, apostrophes and ampersands. Place backslash before the characters to make it display.

Example:

```
document.write "I m a "good" boy" document.write "I m a \"good\" boy"
```

What are the two basic groups of dataypes in JavaScript?

- Primitive types are number and Boolean data types.
- Reference types are more complex types like strings and dates.

How to create an objects in JavaScript?

Objects can be created as:

```
var obj = new Object();
//or
var obj = {};
```

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.

What are the different types of errors in JavaScript?

There are three types of errors:

- **Load time errors:** Errors which come up when loading a web page like improper syntax errors are known as Load time errors and it generates the errors dynamically.

- **Run time errors:** Errors that come due to misuse of the command inside the HTML language.
- **Logical Errors:** These are the errors that occur due to the bad logic performed on a function which is having different operation.

Which keywords are used to handle exceptions?

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

Example:

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

What is the use of blur function?

Blur() function fires when you removes the focus from the specified object.

How to find operating system in the client machine using JavaScript?

The 'Navigator.appversion' is used to find the name of the operating system in the client machine.

How are object properties assigned?

Properties are assigned to objects in the following way -

Example:

```
var obj = {};
obj["first_name"] = 'your first name';
//or
obj.first_name = 'your first name';
```

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.

Example:

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

What is the significance, and what are the benefits, of including 'use strict' at the beginning of a JavaScript source file?

The short and most important answer here is that use strict is a way to voluntarily enforce stricter parsing and error handling on your JavaScript code at runtime. Code errors that would otherwise have been ignored or would have failed silently will now generate errors or throw exceptions. In general, it is a good practice.

Some of the key benefits of strict mode include:

- Makes debugging easier. Code errors that would otherwise have been ignored or would have failed silently will now generate errors or throw exceptions, alerting you sooner to problems in your code and directing you more quickly to their source.
- Prevents accidental globals. Without strict mode, assigning a value to an undeclared variable automatically creates a global variable with that name. This is one of the most common errors in JavaScript. In strict mode, attempting to do so throws an error.
- Eliminates this coercion. Without strict mode, a reference to a this value of null or undefined is automatically coerced to the global. This can cause many headfakes and pull-out-your-hair kind of bugs. In strict mode, referencing a a this value of null or undefined throws an error.
- Disallows duplicate property names or parameter values. Strict mode throws an error when it detects a duplicate named property in an object (e.g., var object = {foo: "bar", foo: "baz"};) or a duplicate named argument for a function (e.g., function foo(val1, val2, val1){}), thereby catching what is almost certainly a bug in your code that you might otherwise have wasted lots of time tracking down.
- Makes eval() safer. There are some differences in the way eval() behaves in strict mode and in non-strict mode. Most significantly, in strict mode, variables and functions declared inside of an eval() statement are not created in the containing scope (they are created in the containing scope in non-strict mode, which can also be a common source of problems).
- Throws error on invalid usage of delete. The delete operator (used to remove properties from objects) cannot be used on non-configurable properties of the object. Non-strict code will fail silently when an attempt is made to delete a non-configurable property, whereas strict mode will throw an error in such a case.

What is the significance of, and reason for, wrapping the entire content of a JavaScript source file in a function block?

This is an increasingly common practice, employed by many popular JavaScript libraries (jQuery, Node.js, etc.). This technique creates a closure around the entire contents of the file which, perhaps most importantly, creates a private namespace and thereby helps avoid potential name clashes between different JavaScript modules and libraries.

Another feature of this technique is to allow for an easily referenceable (presumably shorter) alias for a global variable. This is often used, for example, in jQuery plugins. jQuery allows you to disable the $ reference to the jQuery namespace, using jQuery.noConflict(). If this has been done, your code can still use $ employing this closure technique, as follows:

```
(function($) { /* jQuery plugin code referencing $ */ } )(jQuery);
```

What is the way to get the status of a CheckBox?

The status can be acquired as follows -

```
document.getElementById('checkbox1').checked
```

If the CheckBox will be checked, this alert will return TRUE.

How will you explain closures in JavaScript? When are they used?

A closure is an inner function that has access to the outer (enclosing) function's variables—scope chain. The closure has three scope chains: it has access to its own scope (variables defined between its curly brackets), it has access to the outer function's variables, and it has access to the global variables.

Simply accessing variables outside of your immediate lexical scope creates a closure. Closure is a locally declared variable related to a function which stays in memory when the function has returned.

Example:

```
function foo(x) {
  var tmp = 3;

  return function (y) {
    alert(x + y + (++tmp)); // will also alert 16
  }
}

var bar = foo(2); // bar is now a closure.
bar(10);
```

There are exclusive article on:

- [Mozilla Developer Network (MDN)]
- [StackOverflow]
- [Understanding JavaScript Closure]

When, why, and how would you use a closure in JavaScript? Provide examples.

Probably the most famous example of the use case for closures is the following code snippet:

```
for (var i = 0; i < 5; i++) {
  setTimeout(function() { console.log(i); }, i * 1000 );
}
```

As expected, the above code will log 5 values to the console, one every 1,000 milliseconds (i.e., one every second). However, it will*not* display the values 0, 1, 2, 3, and 4 as might be expected; rather, it will display 5, 5, 5, 5, and 5.

The reason for this is that each function executed within the loop will be executed *after* the entire loop has completed and *all* will therefore reference the *last* value stored in `i`, which was 5.

Closures can be used to prevent this problem by creating a unique scope for each iteration, storing each unique value of the variable within its scope, as follows:

```
var createFunction = function(i) {
  return function() { console.log(i); };
};
for (var i = 0; i < 5; i++) {
  setTimeout(createFunction(i), i * 1000 );
}
```

This will produce the presumably desired result of logging 0, 1, 2, 3, and 4 to the console.

Closures can also be used to resolve issues with the `this` keyword, which is unique to each scope. For example:

```
// Using a closure to access inner and outer object instances simultaneously.
var outerObj = {
  myName: "outer",
  outerFunction: function() {
    // Provide a reference to outerObj through innerFunction's closure
    var self = this;
    var innerObj = {
      myName: "inner",
      innerFunction: function() {
        console.log( self.myName, this.myName ); // "outer inner"
    }
  };
  innerObj.innerFunction();
  console.log( this.myName ); // "outer"
  }
};
outerObj.outerFunction();
```

Explain the for-in loop?

The for-in loop is used to loop through the properties of an object.

The syntax for the for-in loop is :

```
for (variable name in object){
  //statement or block to execute
}
```

In each repetition, one property from the object is associated to the variable name, and the loop is continued till all the properties of the object are depleted.

Describe the properties of an anonymous function in JavaScript?

A function that is declared without any named identifier is known as an anonymous function. In general, an anonymous function is inaccessible after its declaration.

Anonymous function declaration -

```
var anon = function() {
  alert('I am anonymous');
};
anon();
```

What is Event Bubbling?

JavaScript allows DOM elements to be nested inside each other. In such a case, if the handler of the child is clicked, the handler of parent will also work as if it were clicked too.

What is the difference between .call() and .apply()?

The function .call() and .apply() are very similar in their usage except a little difference.

- .call() is used when the number of the function’s arguments are known to the programmer, as they have to be mentioned as arguments in the call statement.
- .apply() is used when the number is not known. The function .apply() expects the argument to be an array.

The basic difference between .call() and .apply() is in the way arguments are passed to the function. Their usage can be illustrated by the given example.

```
var someObject = {
    myProperty : 'Foo',
    myMethod : function(prefix, p ostfix) {
       alert(prefix + this.myProperty + postfix);
    }
};

someObje ct.myMethod('');
//alerts '' var someOtherObject = { myProperty : 'Bar' };

someObject.myMethod.call(someOtherObject, '');
// alerts '';

someObject.myMethod.apply(someOtherObject, ['']);
// alerts ''
```

How can a particular frame be targeted, from a hyperlink, in JavaScript?

This can be done by including the name of the required frame in the hyperlink using the ‘target’ attribute.

Write the point of difference between web-garden and a web-farm?

Both web-garden and web-farm are web hosting systems. The only difference is that web-garden is a setup that includes many processors in a single server while web-farm is a larger setup that uses more than one server.

How are object properties assigned?

Assigning properties to objects is done in the same way as a value is assigned to a variable. For example, a form object's action value is assigned as 'submit' in the following manner - Document.form.action="submit"

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 the method for reading and writing a file in JavaScript?

This can be done by Using JavaScript extensions (runs from JavaScript Editor), example for opening of a file -

```
fh = fopen(getScriptPath(), 0);
```

Define unescape() and escape() functions?

The escape () function is responsible for coding a string so as to make the transfer of the information from one computer to the other, across a network.

Example:

```
document.write(escape(“Hello? How are you!”));
//Output: Hello%3F%20How%20are%20you%21;
```

The unescape() function is very important as it decodes the coded string.

Example:

```
document.write(unescape(“Hello%3F%20How%20are%20you%21”));
//Output: Hello? How are you!
```

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.

Example:

```
var uri="my test.asp?name=ståle&car=saab"; document.write(encodeURI (uri)+ "");
document.write(decodeURI(uri));
//Output - my%20test.asp?name=st%C3%A5le&car=saab
//my test.asp?name=ståle&car=saab
```

Why it is not advised to use innerHTML in JavaScript?

innerHTML content is refreshed every time and thus is slower. There is no scope for validation in innerHTML and, therefore, it is easier to insert rouge code in the document and, thus, make the web page unstable.