**1. Difference between window.onload and onDocumentReady?**

The onload event does not fire until every last piece of the page is loaded, this includes css and images, which means there’s a huge delay before any code is executed.  
That isnt what we want. We just want to wait until the DOM is loaded and is able to be manipulated. onDocumentReady allows the programmer to do that.

**2. What is the difference between == and === ?**

The == checks for value equality, but === checks for both type and value.

**3. What does “1″+2+4 evaluate to? What about 5 + 4 + “3″?**

Since 1 is a string, everything is a string, so the result is 124. In the second case, its 93.

**4. What is the difference between undefined value and null value?**

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.  
Unassigned variables are initialized by JavaScript with a default value of undefined. JavaScript never sets a value to null. That must be done programmatically.

**5. How do you change the style/class on any element?**

document.getElementById(“myText”).style.fontSize = “20″;  
-or-  
document.getElementById(“myText”).className = “anyclass”;

**6. What are Javascript closures?When would you use them?**

Two one sentence summaries:

\* a closure is the local variables for a function – kept alive after the function has returned, or  
\* a closure is a stack-frame which is not deallocated when the function returns.

A closure takes place when a function creates an environment that binds local variables to it in such a way that they are kept alive after the function has returned. A closure is a special kind of object that combines two things: a function, and any local variables that were in-scope at the time that the closure was created.

The following code returns a reference to a function:

function sayHello2(name) {  
var text = ‘Hello ‘ + name; // local variable  
var sayAlert = function() { alert(text); }  
return sayAlert;  
}

Closures reduce the need to pass state around the application. The inner function has access to the variables in the outer function so there is no need to store the information somewhere that the inner function can get it.

This is important when the inner function will be called after the outer function has exited. The most common example of this is when the inner function is being used to handle an event. In this case you get no control over the arguments that are passed to the function so using a closure to keep track of state can be very convenient.

**7. What is unobtrusive javascript? How to add behavior to an element using javascript?**

Unobtrusive Javascript refers to the argument that the purpose of markup is to describe a document’s structure, not its programmatic behavior and that combining the two negatively impacts a site’s maintainability. Inline event handlers are harder to use and maintain, when one needs to set several events on a single element or when one is using event delegation.

|  |  |
| --- | --- |
| 1 | <input type="text" name="date" /> |

Say an input field with the name “date” had to be validated at runtime:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | document.getElementsByName("date")[0].                     addEventListener("change", validateDate, false);    function validateDate(){  // Do something when the content of the 'input' element with the name 'date' is changed.  } |

Although there are some browser inconsistencies with the above code, so programmers usually go with a javascript library such as JQuery or YUI to attach behavior to an element like above.

**8.  What is Javascript namespacing? How and where is it used?**

Using global variables in Javascript is evil and a bad practice. That being said, namespacing is used to bundle up all your functionality using a unique name. In JavaScript, a namespace is really just an object that you’ve attached all further methods, properties and objects. It promotes modularity and code reuse in the application.

**9.  What datatypes are supported in Javascript?**  
Number, String, Undefined, null, Boolean

10. What is the difference between innerHTML and append() in JavaScript?

InnerHTML is not standard, and its a String. The DOM is not, and although innerHTML is faster and less verbose, its better to use the DOM methods like appendChild(), firstChild.nodeValue, etc to alter innerHTML content.

**How do JavaScript timers work? What is a drawback of JavaScript timers?**

Timers allow you to execute code at a set time or repeatedly using an interval. This is accomplished with the setTimeout, setInterval, and clearInterval functions. The setTimeout(function, delay) function initiates a timer that calls a specific function after the delay; it returns an id value that can be used to access it later. The setInterval(function, delay) function is similar to the setTimeout function except that it executes repeatedly on the delay and only stops when cancelled. The clearInterval(id) function is used to stop a timer. Timers can be tricky to use since they operate within a single thread, thus events queue up waiting to execute.

**1).What is JavaScript?**  
Ans:-JavaScript is a scripting language most often used for client-side web development.  
  
  
**2).Difference between JavaScript and Jscript?**  
Ans:-Both JavaScript and Jscript are almost similar. Java script was developed by Netscape. Microsoft reverse engineered Javascript and called it JScript  
  
  
**3).How do we add JavaScript onto a web page?**  
Ans:-  
There are serveral way for adding javascript on a web page but there are two way with is commonly userd by developers  
If your script code is very short and only for single page then following ways is best  
a)You can place <script type="text/javascript"> tag inside the <head> element.

Code:

<head>  
<title>Page Title</title>  
<script language="JavaScript" type="text/javascript">  
   var name = "Vikas Ahlawta"  
   alert(name);  
</script>  
</head>

b).If your script code is very large then you can make a javascript file and add its path in the following way..

Code:

<head>  
<title>Page Title</title>  
<script type="text/javascript" src="myjavascript.js"></script>  
</head>

**4).Is JavaScript case sensitive?**  
Ans:-Yes!  
A function getElementById is not the same as getElementbyID  
  
  
**5).What are the types used in JavaScript?**   
Ans:-String, Number, Boolean, Function, Object, Null, Undefined.  
  
  
**6).What are the boolean operators sported by JavaScript?**  
And Operator: &&  
Or Operator: ||  
Not Operator: !  
  
  
**7).What is the difference between “==” and “===”?**  
Ans:-  
“==” checks equality only,   
“===” checks for equality as well as the type.  
  
  
**8).How to access the value of a textbox using JavaScript?**  
Ans:-  
ex:-

Code:

<!DOCTYPE html>  
<html>  
<body>  
Full name: <input type="text" id="txtFullName" name="FirstName" value="Vikas Ahlawat">  
</body>  
</html>

There are following way to access the value of the above textbox  
var name = document.getElementById('txtFullName').value;  
alert(name);  
or   
we can use the old way  
document.forms[0].mybutton.  
var name = document.forms[0].FirstName.value;  
alert(name);  
Note:- this uses the "name" attribute of the element to locate it.  
  
  
**9).What are the way of make comment in Javascript?**  
Ans:-  
// is used for line comments  
ex:- var x=10; //comment text  
  
/\*  
\*/ is used for block comments  
ex:-  
var x= 10; /\* this is  
block comment example.\*/  
  
  
**10).How you will get the CheckBox status whether it is checked or not?**  
Ans:-  
var status = document.getElementById('checkbox1').checked;   
alert(status);   
it will return true or false  
  
  
**11).How to create arrays in JavaScript?**   
Ans:-  
There are Two way dor create array in Javascript like other languages..  
a) first way to create array  
Declare Array:-

Code:

var names = new Array();   
Add Elements in Array:-  
names[0] = "Vikas";  
names[1] = "Ashish";  
names[2] = "Nikhil";

b) this is second way  
var names = new Array("Vikas", "Ashish", "Nikhil");  
  
  
**12).If an array with name as "names" contain three elements then how you will print the third element of this array?**  
Ans:- Print third array element document.write(names[2]);   
Note:- array index start with 0  
  
  
**13).How do you submit a form using Javascript?**   
Ans:-Use document.forms[0].submit();  
  
  
**14).What does isNaN function do?**   
Ans:-  
It Return true if the argument is not a number.  
ex:-

Code:

document.write(isNaN("Hello")+ "<br>");  
document.write(isNaN("2013/06/23")+ "<br>");  
document.write(isNaN(123)+ "<br>");

output will be:-  
true  
true  
false  
  
  
**15).What is the use of Math Object in Javascript?**  
Ans:-  
The math object provides you properties and methods for mathematical constants and functions.  
ex:-

Code:

var x = Math.PI; // Returns PI  
var y = Math.sqrt(16); // Returns the square root of 16  
var z = Math.sin(90);    Returns the sine of 90

**16). What do you understand by this keyword in javascript?**   
Ans:-In JavaScript the this is a context-pointer and not an object pointer. It gives you the top-most context that is placed on the stack. The following gives two different results (in the browser, where by-default the window object is the 0-level context):

Code:

var obj = { outerWidth : 20 };  
  
function say() {  
    alert(this.outerWidth);  
}  
say();//will alert window.outerWidth  
say.apply(obj);//will alert obj.outerWidth

**17).What does "1"+2+4 evaluate to?**   
Ans:-Since 1 is a string, everything is a string, so the result is 124.   
  
  
**18).What does 3+4+"7" evaluate to?**  
Ans:-Since 3 and 4 are integers, this is number arithmetic, since 7 is a string, it’s concatenation, so 77 is the result.  
  
  
**19).How do you change the style/class on any element using javascript?**  
Ans:-

Code:

document.getElementById(“myText”).style.fontSize = “10";  
-or-  
document.getElementById(“myText”).className = “anyclass”;

**20).Does javascript support foreach loop?**  
Ans:- Yes, See example here <http://jsfiddle.net/gpDWk/>   
  
  
**21).What looping structures are there in JavaScript?**   
Ans:-for, while, do-while loops  
  
  
**22).what is an object in JavaScript, give an example?**  
Ans:-  
An object is just a container for a collection of named values  
  
// Create the man object

Code:

var man = new Object();  
man.name = 'Vikas Ahlawat';  
man.living = true;  
man.age = 27;

**23).How you will add function as a property in a JavaScript object? Give example.**  
Ans:-

Code:

var man = new Object();  
man.name = 'Vikas Ahlawat';  
man.living = true;  
man.age = 27;

man.getName = function() { return man.name;}  
console.log(man.getName()); // Logs 'Vikas Ahlawat'.  
  
  
**24).What is the similarity between 1st and 2nd statement?**  
1st:- var myString = new String('male'); // An object.  
2nd:- var myStringLiteral = 'male'; // Primitive string value, not an object.  
Ans:- Both will call String() constructor function  
you can confirm it by run the following statement  
console.log(myString.constructor, myStringLiteral.constructor);  
  
  
**25).What will be the output of the following statements?**

Code:

var myString = 'Vikas' // Create a primitive string object.  
var myStringCopy = myString; // Copy its value into a new variable.  
var myString = null; // Manipulate the value  
console.log(myString, myStringCopy);

Ans:- // Logs 'null Vikas'  
  
  
**26).Consider the following statements and tell what would be the output of the logs statements?**  
var price1 = 10;  
var price2 = 10;  
var price3 = new Number('10'); // A complex numeric object because new was used.  
console.log(price1 === price2);   
console.log(price1 === price3);  
Ans:-  
console.log(price1 === price2); // Logs true.  
console.log(price1 === price3); /\* Logs false because price3 contains a complex number object and price 1  
is a primitive value. \*/  
  
  
**27).What would be the output of the following statements?**  
var object1 = { same: 'same' };  
var object2 = { same: 'same' };  
console.log(object1 === object2);  
Ans:- // Logs false, JavaScipt does not care that they are identical and of the same object type.  
When comparing complex objects, they are equal only when they reference the same  
object (i.e. have the same address). Two variables containing identical objects are not  
equal to each other since they do not actually point at the same object.  
  
  
**28).What would be the output of the following statements?**

Code:

var object1 = { same: 'same' };  
var object2 = object1;  
console.log(object1 === object2);

Ans:- // Logs true  
  
  
**29).What is this?**  
var myArray = [[[]]];  
Ans:- Three dimantional array  
  
  
**30).Name any two JavaScript functions which are used for convert nonnumeric values into numbers?**  
Ans:-  
Number()  
parseInt()  
parseFloat()

Code:

var n1 = Number(“Hello world!”); //NaN  
var n2 = Number(“”);             //0  
var n3 = Number(“000010”);       //10  
var n4 = Number(true);           //1  
var n5 = Number(NaN);            //NaN