

JavaScript Interview Questions & Answers

1. What is JavaScript?

JavaScript is a scripting language developed by Netscape. It can be used to program web browser or even servers. It can dynamically update the contents of the webpage, which is the beauty of JavaScript.

2. What are the Advantages of JavaScript?

JavaScript Scripting language has many advantages as stated below.

- Lightweight: JavaScript is easy to implement. It has small memory footprints.
- Interpreted: It is an interpreted language. Instructions are executed directly.
- Object-oriented: JavaScript is an object-oriented language.
- First class functions: In JavaScript, a function can be used as a value.
- Scripting Language: It's a language in which instructions are written for a run-time environment.

3. What are the advantages of using External JavaScript?

Using External JavaScript in our code has many advantages as stated below.

- Separation of Code is done.
- Code Maintainability is Easy.
- Performance is better.

4. What is the difference between test () and exec () methods?

Both test () and exec () are RegExp expression methods.

Using test (), we will search a string for a given pattern, if it finds the matching text then it returns the Boolean value 'true' and else it returns 'false'.

But in exec (), we will search a string for a given pattern, if it finds the matching text then it returns the pattern itself and else it returns 'null' value.

5. Have you used any browser for debugging? If yes, how is it done?

By, Pressing 'F12' key in the keyboard we can enable debugging in the browser. Chose the 'Console' tab to view the result.

In Console, we can set breakpoints and View the value in variables. All the modern browsers have a built-in debugger with them (For Example: Chrome, Firefox, Opera, and Safari). This feature can be turned ON and OFF.

6. What is the use of 'debugger' keyword in JavaScript code?

Using the 'debugger' keyword in the code is like using breakpoints in the debugger.

7. What are the distinct types of Error Name Values?

There are 6 types of values in 'Error Name' Property.

- Range Error -> We will get this error if we use a number outside the range
- Syntax Error -> This error raises when we use the incorrect syntax.
- Reference Error -> This error is thrown if used an undeclared variable
- Eval Error -> Thrown due to the error in eval(). New JavaScript version doesn't have this error
- Type Error -> Value is outside the range of types used.
- URI Error -> Due to the usage of illegal characters.

8. What is JavaScript Hoisting?

All variables should be declared at the top of the original /current scope.

If we have declared anywhere inside the javascript then it should be brought to the top. This is only applicable for declaration we have initialize a variables anywhere.

9. What is JavaScript 'Strict Mode'?

Usually, JavaScript is 'not very strict' in throwing errors.

'Strict mode' is a restricted variant of JavaScript.

In 'Strict mode' it will throw all types of errors, even the silent errors. Thus, the process of debugging becomes easier.

10. What are the characteristics of JavaScript 'Strict Mode'?

- 'Strict Mode' will stop developers from creating global variables.
- Developers are restricted from using duplicate parameters.
- Strict mode will restrict you from using JavaScript keyword as a variable name or function name.
- Strict mode is declared with 'use strict' keyword at the beginning of the script.
- All browser support strict mode.

11. What are Self Invoking Functions?

They are also known as 'Immediately Invoked Function Expressions' or 'Self Executing Anonymous Functions'. These functions are invoked automatically in the code, hence they are named as 'Self Invoking Functions'.

12. What is the syntax of 'Self Invoking Function'? Give an example?

```
(function () {  
    return () } () ;
```

Here the last '()' parenthesis in the syntax states that it is a function expression.

Ex:

```
(function (){  
    elem = document.getElementById("dispaly_num");  
    elem.innerHTML = "This function has no name.<br>It is called automatically";  
})();
```

13. What is the difference between 'var' and 'let' keyword?

Var:

- 'var' keyword was introduced in JavaScript code from the beginning Stage itself.
- 'Var' keyword has function scope. This is available anywhere within the function
- The variable declared with 'var' be hoisted

Let:

- 'let' keyword is introduced in 2015 only.
- 'let' keyword has a block scope. This is available only within the block
- The variable declared with 'let' be hoisted

14. What is the difference between '==' and '==='?

Both '==' and '===' are comparison operators.

==;

- It is known as 'Type Converting Operator'
- It compares Value, do not compare type

===

- It is known as 'Strict Equality Operator'
- It compares both value and type.

15. What is the difference between 'let' and 'const'?

let:

using 'let' we can change the value of variable any number of times

const:

using 'const', after the first assignment of the value we cannot redefine the value again

16. What is the difference between 'null' and 'undefined'?

Both the keywords represent empty values.

In 'undefined', we will define a variable, but we won't assign a value to that variable. On the other hand, in 'null' we will define a variable and assign the 'null' value to the variable.

17. What is the difference between 'function declaration' and 'function expression'?

Function declarations are hoisted but function expressions are not hoisted.

Function Declaration:

```
function bar() {  
  return 3;  
}
```

Function Expressions:

```
var x = function (a, b) {return a * b};
```

18. What are 'setTimeout()'?

It calls a function after a specified number of milliseconds.

19. What is a Closure and How do you use it?

closure is an inner function that has access to the variables in the outer (enclosing) function's scope chain.

```
function showName (firstName, lastName) {  
  var nameIntro = "Your name is ";  
  function makeFullName () {  
    return nameIntro + firstName + " " + lastName;  
  }  
  return makeFullName ();  
}  
  
alert(showName ("Michael", "Jackson"));
```

20. If you need to hide the JavaScript code from the older browser versions, how will you perform it?

```
<script><!-- .... --></script>
```

21. What's An Object In Javascript And How Do We Create Them?

A JavaScript object is an entity having properties and method. Since JavaScript is an object-based language, it treats everything as an object.

JavaScript is a template-based language. It doesn't need to define a class for creating an object instead creates it directly

3 ways to create an object:

- By Object Literal.
- By Creating An Instance Of The Object (Using New Keyword).
- By Using An Object Constructor

22. What Does A Scope Mean In JavaScript?

scope determines the accessibility of variables, objects, and functions in particular part of your code.

2 types : Global and local scope

23. What Is <This> In JavaScript?

The keyword 'this' refers to the current instance of the object when used inside a function. But, when used outside a function, it refers to the window object.

24. What Is The Prototype Property In JavaScript?

By Using Prototype we can add new members to an existing object. Every JavaScript object has this property internally. Initially it is an empty object.

```
function Person(first, last, age, eye) {  
    this.firstName = first;
```

```
this.lastName = last;
this.age = age;
this.eyeColor = eye;
}
Person.prototype.nationality = "English";
var myFather = new Person("John", "Doe", 50, "blue");
document.getElementById("demo").innerHTML =
"The nationality of my father is " + myFather.nationality;
```

26. What Is An Anonymous Function And When Should You Use It?

Dynamically declared at runtime, they don't have a name like normal functions.

- Code brevity.

Use them in Callbacks, and Event handlers.

- Scope management.

They are useful in the following scenario.

To create a temporary/private scope.

In Closures and Recursions.

27. What Is Prototypal Inheritance In JavaScript?

In JavaScript, the inheritance is prototype-based. It means that there are no classes. Instead, there is an object that inherits from another object.

```
function Rectangle(x, y) {
    this.x = x;
    this.y = y;
}
Rectangle.prototype.perimeter = function() {
```



```
    return 2 * (this.x + this.y);  
}  
  
var rect = new Rectangle(4, 2);  
  
console.log(rect.perimeter()); // outputs '12'
```

28. What Is Asynchronous Programming? Why Is It Important In JavaScript?

An asynchronous model allows multiple things to happen at the same time. When we start an action our program continues to run. When the action finishes the program is informed and gets access to the result.

29. What Is Variable Typing In JavaScript?

JS is a very loosely typed language. It means that the variables are declared without a type. Its type is determined once a value is assigned to it.

30. What Are The Different Ways To Create An Array In JavaScript?

Using An Array Initializer (Array Literal)., Using The Array Constructor

31. What Are JavaScript Cookies?

Cookie is a piece of data which is sent from a website (that owns the requested web page) and gets stored locally by the browser at the user end.

32. Which Built-In Method Adds One Or More Elements To The End Of An Array And Returns The New Length Of The Array?

push()

33. What Is The Naming Conventions For The Variables In JavaScript?

- Do not use any of the JavaScript reserved keywords as a name for your variable. For example, the “break” or “boolean” is the JavaScript keywords, and if used as variable names, it’s invalid.

- JavaScript variable names should not start with a numeral (0-9). It must begin with a letter or the underscore character. For example, 123var is an invalid variable name, but _123var is a valid one.

- Also, JavaScript variable names are case sensitive. For example, test and Test are two different variables.

34. How Will You Create A Cookie Using JavaScript?

The simplest way to create a cookie is to assign a string value to the <document.cookie> object.

ex: document.cookie = "key1 = value1; key2 = value2; expires = date";

35. How To Read A Cookie Using JavaScript?

To read a Cookie, we have to access the value of the <document.cookie> object. This <document.cookie> string maintains a list of <name = value> pairs that is separated with semicolons.

Where,

"name" is the name of a cookie and "value" is its string value.

36. How To Delete A Cookie Using JavaScript?

To delete a Cookie, we have to set its expiry date to a time that occurred in the past. If attempts

are made to read a deleted Cookie then, nothing is returned.

37. How Do You Submit A Form Using JavaScript?

document.forms[0].submit();

38. Why Should You Not Prefer To Use Global Variables In JavaScript And How Can You Prevent It?

The principal issue in using a global variable is that someone else can create another variable with the same name. And you may not know it until the duplicate could overwrite the value of your variable.

1. Create a single global variable that holds all your other variables.
2. Enclose all of your code in a self-executing method/function so that any variable declared inside remain in the function scope.

39. What Are The Different Objects Used In JavaScript?

Window Object.

Document Object.

Form Object.

40. What Do We Achieve By Deferring The Loading Of JavaScript?

By deferring the loading of JavaScript, we instruct the browser to load the script, only after the loading of the web page(DOM) has finished. It reduces the loading time of the webpage, and it gets displayed faster.

```
<script>  
// this piece of code runs first  
  
</script>  
  
<script defer="defer">  
//do stuff, but defer it (runs last)  
  
</script>  
  
<script>  
//do more stuff (runs second)  
  
</script>
```

41. What Are Event Handlers In JavaScript And How To Use Them?

`<onclick()>`, `<onmouseover()>`, `<onkeydown()>` ... like this

42. What Does A JavaScript Function Result When It Has No Return Statement?

If a function doesn't use the return statement, then it returns undefined as the return value.

43. What Is EncodeURI() Function?

The `encodeURI()` function is used to encode a URI. This function encodes all special characters, except these `< , / ? : @ & = + $ # >`.

44. How Does The `<Array()>` Differ From `<[]>` While Creating A JavaScript Array?

Both the `<Array()>` and `<[]>` works almost the same in JavaScript.

If we use them as is (i.e., without any argument) to create an array object, then they will result in an array object of zero length. Also, if we pass a string or a list of strings as arguments, even then the result will be similar.

45. How Will You Replace All Occurrences Of A String In JavaScript?

We can use String's `<replace()>` method to substitute any string.

46. How can you read properties of an Object in JavaScript?

You can write and read properties of an object using the dot notation

47. How to read elements of an array in JavaScript?

An array has a length property that is useful for iteration. We can read elements of an array as follows –

```
var x = [1, 2, 3, 4, 5];  
for (var i = 0; i < x.length; i++) {  
    // Do something with x[i]  
}
```

48. What is a named function in JavaScript? How to define a named function?

A named function has a name when it is defined. A named function can be defined using function keyword as follows –

```
function named(){  
    // do some stuff here  
}
```

49. How many types of functions JavaScript supports?

A function in JavaScript can be either named or anonymous.

50. Can you assign a anonymous function to a variable?

Yes! An anonymous function can be assigned to a variable.

51. Can you pass a anonymous function as an argument to another function?

Yes! An anonymous function can be passed as an argument to another function.

52. What is arguments object in JavaScript?

JavaScript variable arguments represents the arguments passed to a function.

53. How can you get the type of arguments passed to a function?

Using typeof operator, we can get the type of arguments passed to a function. For example –

```
function func(x){  
    console.log(typeof x, arguments.length);  
}  
func();           //==> "undefined", 0  
func(1);          //==> "number", 1  
func("1", "2", "3"); //==> "string", 3
```

54. How can you get the total number of arguments passed to a function?

Using arguments.length property, we can get the total number of arguments passed to a function.

55. How can you get the reference of a caller function inside a function?

The arguments object has a callee property, which refers to the function you're inside of. For example –

```
function func() {  
    return arguments.callee;  
}  
func();           // ==> func
```

56. Which type of variable among global and local, takes precedence over other if names are same?

A local variable takes precedence over a global variable with the same name.

57. What is callback?

The callback function is a mechanism to send one function to another function as an argument; i.e., passing func as an argument to another function.

58. Which built-in method returns the character at the specified index?

charAt() method returns the character at the specified index.

59. Which built-in method combines the text of two strings and returns a new string?

concat() method returns the character at the specified index.

60. Which built-in method calls a function for each element in the array?

forEach() method calls a function for each element in the array.

61. Which built-in method returns the index within the calling String object of the first occurrence of the specified value?

indexOf() method returns the index within the calling String object of the first occurrence of the specified value, or -1 if not found.

62. Which built-in method returns the length of the string?

length() method returns the length of the string.

63. Which built-in method removes the last element from an array and returns that element?

pop() method removes the last element from an array and returns that element.

64. Which built-in method adds one or more elements to the end of an array and returns the new length of the array?

push() method adds one or more elements to the end of an array and returns the new length of the array.

65. Which built-in method reverses the order of the elements of an array?

reverse() method reverses the order of the elements of an array -- the first becomes the last, and the last becomes the first.

66. Which built-in method sorts the elements of an array?

sort() method sorts the elements of an array.

67. Which built-in method returns the characters in a string beginning at the specified location?

substr() method returns the characters in a string beginning at the specified location through the specified number of characters.

68. Which built-in method returns the calling string value converted to lower case?

toLowerCase() method returns the calling string value converted to lower case.

69. Which built-in method returns the calling string value converted to upper case?

toUpperCase() method returns the calling string value converted to upper case.

70. Which built-in method returns the string representation of the number's value?

toString() method returns the string representation of the number's value.

71. How typeof operator works?

The typeof 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.

The typeof operator evaluates to "number", "string", or "boolean" if its operand is a number, string, or boolean value and returns true or false based on the evaluation.

72. What typeof returns for a null value?

It returns "object".

73. Can you access Cookie using javascript?

JavaScript can also manipulate cookies using the cookie property of the Document object. JavaScript can read, create, modify, and delete the cookie or cookies that apply to the current web page.

74. How to redirect a url using JavaScript?

This is very simple to do a page redirect using JavaScript at client side. To redirect your site visitors to a new page, you just need to add a line in your head section as follows –

```
<head>
<script type="text/javascript">
<!--
    window.location="http://www.newlocation.com";
//-->
</script>
</head>
```

75. How to print a web page using javascript?

JavaScript helps you to implement this functionality using print function of window object. The JavaScript print function window.print() will print the current web page when executed.

76. What is Date object in JavaScript?

The Date object is a datatype built into the JavaScript language. Date objects are created with the new Date().

Once a Date object is created, a number of methods allow you to operate on it. Most methods simply allow you to get and set the year, month, day, hour, minute, second, and millisecond fields of the object, using either local time or UTC (universal, or GMT) time.

77. What is Number object in JavaScript?

The Number object represents numerical data, either integers or floating-point numbers. In general, you do not need to worry about Number objects because the browser automatically converts number literals to instances of the number class.

Syntax –

Creating a number object –

```
var val = new Number(number);
```

78. How to handle exceptions in JavaScript?

The latest versions of JavaScript added exception handling capabilities. JavaScript implements the try...catch...finally construct as well as the throw operator to handle exceptions.

You can catch programmer-generated and runtime exceptions, but you cannot catch JavaScript syntax errors.

79. What is purpose of onError event handler in JavaScript?

The onerror event handler provides three pieces of information to identify the exact nature of the error –

- Error message – The same message that the browser would display for the given error.
- URL – The file in which the error occurred.
- Line number – The line number in the given URL that caused the error.

80. What is the difference between JavaScript and Jscript?

Both JavaScript and Jscript are almost similar. JavaScript was developed by Netscape. Microsoft reverse engineered Javascript and called it JScript.

81. What are the boolean operators supported by JavaScript?

And Operator: &&

Or Operator: ||

Not Operator: !

82. How will you get the Checkbox status whether it is checked or not?

```
var status = document.getElementById('checkbox1').checked;
```

```
alert(status);
```

will return true or false.

83. If an array with name as "names" contain three elements, then how will you print the third element of this array?

```
Print third array element document.write(names[2]);
```

83. What does isNaN function do?

It returns true if the argument is not a number.

84. What is the use of Math Object in JavaScript?

The math object provides you properties and methods for mathematical constants and functions.

85. How do you change the style/class on any element using JavaScript?

```
document.getElementById("myText").style.fontSize = "10";
```

-or-

```
document.getElementById("myText").className = "anyclass";
```

86. Does JavaScript support foreach loop?

JavaScript 1.6(ECMAScript 5th Edition) support foreach loop

87. Name any two JavaScript functions which are used to convert nonnumeric values into numbers?

Number()

parseInt()

parseFloat()

91. What is the definition of prompt box?

A prompt box is a box which allows the user to enter input by providing a text box. The prompt() method displays a dialog box that prompts the visitor for input. A prompt box is often used if you want the user to input a value before entering a page. When a prompt box pops up, the user will have to click either "OK" or "Cancel" to proceed after entering an input value.

92. What are the disadvantages of using inner HTML in JavaScript?

1. The process of using inner HTML is much slower than the rest of the variables as its content is slowly built into different elects and takes time to get re-parsed.

2. While using the inner HTML, the content gets replaced in JavaScript.

3. Appending to inner HTML can't be used properly.

4. Using inner HTML can also break the document of the JavaScript.

Since no validation is required by it, any type of valid inner HTML can be used. Even broken HTML can also be used and this can cause problems.

5. The old content also gets easily replaced.

93. What are the meanings of undefined variables and undeclared variables?

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.

94. Define the types of functional components which is available in JavaScript programming language?

1. Nested functions - As the name suggests, nested functions are the ones which are included in other functions. They are basically called each time when the invoked functions are provided.

2. First class functions - These are the types of functions which are used as first-class objects. Such functions can be used as arguments and opinions against other types of functions. The data structures can store the values of such functional components and the variables can also be stored properly in the data structures.

95. How to declare a private and a public member?

Private members are declared using var keyword and constructor function.

```
Function Student (name, roll){  
var id= ABCD123;  
this.name = name;  
this.roll = roll;  
}
```

96. Define Screen Objects. What is its proper?

Screen Objects are used to read a different type of information from the screen of the client.

97. How can one handle exceptions in JavaScript?

The operators used to handle exceptions in JavaScript programming language are try, catch and construct. These are the latest versions.

98. What is constructor property?

Constructor property of an object maintains a reference to its creator function.

99. How to call other class methods?

Using call () and apply () method

call (): It is used to calls a function with a given this value and arguments provided individually.

```
fun.call(thisArg[, arg1[, arg2[, ...]]])
```

apply (): It is used to call a function with a given this value and arguments provided as an array.

```
fun.apply(thisArg, [argsArray])
```

100. How many types of objects are there in JavaScript? Define them?

1. Date object - q.no 76
2. Number object - q.no 78

101. Explain method overriding with an Example?

We can override any inbuilt method of JavaScript by declaring its definition again. The existing definition is accessible at override by the Prototype property.

102. What is the meaning of continuing and break statements?

Continue statements are the ones which continue from the next loop with the next set of statements. On the other hand, break statements are the ones that start from the current existing loop.

103. How to inherit form a class?

Inheritance can be achieved in JavaScript using Prototype property.

Step1:

Child class prototype should point to parent class object.

`.prototype = new ();`

Step2:

Reset the child class prototype constructor to point self. `.prototype .`

`constructor = student;`

104. What is the difference between Attributes and Property?

Attributes- provide more details on an element like id, type, value etc.

Property- is the value assigned to the property like `type="text"`, `value='Name'` etc.

105. List out the different 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.

106. What is a Typed language?

Typed Language is in which the values are associated with values and not with variables. It is of two types:

Dynamically: in this, the variable can hold multiple types; like in JS a variable can take number, chars.

Statically: in this, the variable can hold only one type, like in Java a variable declared of string can take only set of characters and nothing else.

107. 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.

108. Name some of the JavaScript Frameworks

Angular

React

Vue

109. What is the difference between window & document in JavaScript?

Window:

JavaScript window is a global object which holds variables, functions, history, location.

Document:

The document also comes under the window and can be considered as the property of the window.

110. What is the difference between innerHTML & innerText?

innerHTML – It will process an HTML tag if found in a string

innerText – It will not process an HTML tag if found in a string

111. What is an event bubbling in JavaScript?

When an event is fired on an HTML element, the execution starts from that event and goes to its parent element. From there, the execution passes to its parent element and so on till the body element.

stop: <button onclick="event.stopPropagation()">

112. What is NaN in JavaScript?

NaN is a short form of Not a Number. Since NaN always compares unequal to any number, including NaN, it is usually used to indicate an error condition for a function that should return a valid number. When a string or

something else is being converted into a number and that cannot be done, then we get to see NaN.

113. How do JavaScript primitive/object types passed in functions?

One of the differences between the two is that Primitive Data Types are passed By Value and Objects are passed By Reference.

By Value means creating a COPY of the original. Picture it like twins: they are born exactly the same, but the first twin doesn't lose a leg when the second twin loses his in the war.

By Reference means creating an ALIAS to the original. When your Mom calls you "Pumpkin Pie" although your name is Margaret, this doesn't suddenly give birth to a clone of yourself: you are still one, but you can be called by these two very different names.

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

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

115. What are Exports & Imports?

Imports and exports help us to write modular JavaScript code. Using Imports and exports we can split our code into multiple files.

116. What happens when the recursion calling is applied on two functions?

The calling of recursion is possible in two functions, but the call comes to an end after some time.

117. Explain the terms synchronous and asynchronous code.

The synchronous code is something that should be finished before anything else can happen, or in other words, the synchronous code is blocking. And the Asynchronous code is something in which actions can happen and is not dependent on other actions- in other words, it is non-blocking.

118. Name the different types of pop up boxes in Javascript.

There are three types of pop up boxes in Javascript (i) alert() provides some information to the user with just an OK button (ii) confirm() asks a question to the user with two options Ok and cancel, and (iii) prompt() takes an input from the user.

119. What is the difference between typeof and instanceof operators in Javascript?

The typeof operator returns a string of what type the operand is. Whereas, the instanceof operator does not work with primitive data types; but works with objects and checks on what type the object is.

120. What is DOM?

DOM (Document Object Model) is an object-oriented representation of the HTML elements. All the elements (or nodes) are part of window.document.

121. Expand BOM and explain it.

BOM stands for Browser Object Model. Using BOM interaction with a browser is possible. Default object of the browser is a window.

122. What is the difference between textContent and innerText?

Let us have a paragraph element and a span element in it as a child element.

`<p>some text and a span tag hidden
<\span>in it</p>`

Now, if the following two steps would result in the following-

`console.log(document.querySelector('p').textContent);` gives some text and a span tag hidden in it.

`console.log(document.querySelector('p').innerText);` gives some text and a in it.

123. What is the difference between HTMLCollection and NodeList?

The functions `querySelectorAll()` returns `NodeList` in which the `forEach` can be used directly to traverse the elements. Whereas, the `getElementsByClassName()` or `getElementsByTagName()` returns an `HTMLCollection`, which does not have a `forEach` by default.

124. How can an HTMLCollection be traversed?

The `HTMLCollection` by default does not have `forEach`, it needs to be converted into an array (`eleName = Array.from(eleName)`) and then the traversal is possible using `forEach`.

125. What is the difference between `childNodes` and `children`?

A `childNodes`, when used returns a `NodeList`. Whereas, `children`, when used; returns an `HTMLCollection`.

126. What is the difference between `firstChild` and `firstElementChild`?

A `firstChild` when used returns the first node. It could be an HTML element or even a space, or a new line. Whereas, `firstElementChild`, when used returns the first HTML element only.

127. Name the two functions that are used to create an HTML element dynamically.

To generate an html element dynamically, we need to use two functions:

- (i) `var ele = createElement ('elementName')`, which creates an element and
- (ii) `parentElement.appendChild(ele)`

128. What is the difference between `remove()` and `removeChild()`?

The `remove()` function just removes the element. Whereas, the `removeChild()` returns the deleted element.

129. What is the use of history object?

By using the history object you can change or switch back to history pages or forward from current page to another page.

130. List out all the falsifying tokens in Javascript.

There are 6 tokens that falsify in Javascript and they are false, null, undefined, 0, NaN.

131. What is Currying in Javascript?

A partial invocation of a Javascript function is called Currying. Few arguments of a function are processed and a function is returned. Few more arguments are added by the returning function.

132. When do we use `JSON.stringify()`?

The `JSON.stringify()` method is used to convert a Javascript data to a string.

133. 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.

134. Explain how to read and write a file using JavaScript?

There are two ways to read and write a file using JavaScript

Using JavaScript extensions

Using a web page and Active X objects

135. Explain how to detect the operating system on the client machine?

In order to detect the operating system on the client machine, the navigator.platform string (property) should be used.

136. Whether JavaScript has concept level scope?

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

137. Explain window.onload and onDocumentReady?

The onload function is 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.

138. 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.

139. What is namespaces 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.

140. What is the difference between classical inheritance and prototypal inheritance?

Class Inheritance: instances inherit from classes (like a blueprint - a description of the class), and create sub-class relationships:

hierarchical class taxonomies. Instances are typically instantiated via constructor functions with the new keyword. Class

inheritance may or may not use the class keyword from ES6.

Prototypal Inheritance: instances inherit directly from other objects. Instances are typically instantiated via factory functions

or Object.create(). Instances may be composed from many different objects, allowing for easy selective inheritance.

141. Map

It creates a new array element with the result of calling a function with every array element

```
var numbers = [2, 4, 6, 8]
```

```
var half = numbers.map(x=>x/2);
```


142. Filters

Its create a new array with all element that pass the text implemented by the provider function

```
const words = ["spray", "limit", "elite", "destruction"];  
const longwords = words.filter(word=>word.length>6);
```

143. Reduce

```
const total = [0, 1, 2, 3].reduce((sum,value)=>sum+value, )
```

