

Javascript

- 1) Javascript Implementation
 1. The Core(ECMAScript)
 2. DOM
 3. BOM
- 2) ECMAScript describes
 1. Syntax
 2. Types
 3. Statements
 4. Keywords
 5. Reserved words
 6. Operators
 7. Objects
- 3) DOM
 1. Tree structure of page
- 4) BOM
 1. Browser specific features
- 5) Script Element
 1. Normally script tag is placed in head
 2. Src attribute can use url of js in other server
 3. Executed in order as mentioned in page
 4. Defer attribute
- 6) Inline code vs external files
 1. Maintainability
 2. Caching – same file used in multiple pages
- 7) NoScript element
- 8) Language Basics
 1. Syntax C Like
 2. Case sensitivity
 3. Identifiers
 - i) First character – Letter,_, \$
 - ii) All other character letters,_, \$,numbers
 4. Comments - //, /* */

5. Strict mode – “use strict”;
6. Statement – without semicolon, with semicolon
7. Keywords
8. Variables – var keyword and global scope variable
9. Datatypes – undefined,null,number,string,bool,object
10. typeof operator
11. Undefined type – declared but not initialized
12. Null type
13. Boolean type
14. Boolean casting function
15. Floating point values
16. Number max value
17. Number min value
18. NaN – isNaN
19. Number function
20. parseInt
21. parseFloat
22. string type – “ ”, ‘ ’, ` `”
23. string.length
24. .toString()
25. Object type
 - i) Var x = new Object();
 - ii) Var y = new Object;
26. Unary Operator - ++,--, +1
27. Boolean operator
 - i) !false
 - ii) !
 - iii) &&
 - iv) ||
28. Multiplicative Operator - *
29. Divide - /
30. Modulus - %
31. Additive Operators - +
32. Subtract - -

33. Relational Operator –
 - i) >
 - ii) <
 - iii) <=
 - iv) >=
34. Equality Operator
 - i) ==
 - ii) !=
 - iii) ===
35. Conditional Operator
 - i) Var = expression ?true:false
36. Assignment Operator
37. Comma Operator
 - i) Var a=1,b=2,c=3
 - ii) Var b = (1,5,6,8);
38. If Statement
39. If Else
40. If else if else
41. Do while
42. While
43. For
 - i) For(;;) – infinite loop
44. For in
 - i) Arr[i]
45. Break
46. Continue
47. With keyword
48. Switch
49. Function
 - i) With return
 - ii) Without return
50. Arguments keyword in function
51. No overloading

52. Primitive values
53. Reference values
54. Argument passing (by Value)
55. Variable scope
56. Object declaration
 - i) New Object();
 - ii) {}
 - iii) {name:"Ram"}
 - iv) Person.Name
 - v) Person["name"]
 - vi) Person["first name"]
57. Array
 - i) New Array();
 - ii) New Array(20);
 - iii) New array("aa","bb","cc");
 - iv) New Array("aa");
 - v) Array(3);
 - vi) Array("greg")
 - vii) [5,6,8,7];
 - viii) [];
58. Array.length property
 - i) To read array length
 - ii) To set array length
 - iii) To add items in the array
59. Array.push() – Add last item
60. Array.pop() – remove and return last item
61. Array.shift() – remove and return first item
62. Array.unshift() – add first items and return count
63. Array.reverse()
64. Array.sort()
65. Array.concat(Array)
66. Array.slice()
 - i) Array.slice(1)

ii) `Array.slice(1,4)`

9) Array iterative functions

1. `Every(item,index,arr)`
2. `Some(item,index,arr)`
3. `Filter(item,index,arr)`
4. `Map(item,index,arr)`
5. `forEach(item,index,arr)`

10) Date type

1. `New Date()`
2. `New Date(2016,5,9)`
3. `getFullYear()`
4. `setFullYear()`
5. `setDate()`

11) Function types

1. `Var aa = function(){}`
2. Functions as values (Pass functions as parameters)
3. Callback functions
4. `functionName.length = number of arguments`

12) string functions

1. `substring()`
2. `concat()`
3. `stringValue.indexOf()`
4. `stringValue.lastIndexOf()`
5. `trim()`
6. `toUpperCase()`
7. `toLowerCase()`
8. `replace()`
9. `split()`

13) Character functions

1. `charAt(0)`
2. `stringValue[0]`

14) Object Properties

1. `defineProperty`

2. defineProperties

15) Creating Object

1. Factory Pattern – return object
2. Constructor Pattern – this.propName
3. Prototype Pattern

16) Functions

1. Recursive functions
2. Block Scopes

17) Window Object

1. Window.open()
2. Window.close()
3. setTimeout()
4. clearTimeout()
5. setInterval()
6. Dialogues
 - i) Alert()
 - ii) Confirm()
 - iii) Prompt()
 - iv) Window.print()

18) Window object

1. Window.location
2. Get query string data
3. Location.href
4. Location.assign
5. Location.hash
6. Location.search
7. Location.pathname
8. Location.hostname
9. Global scope
10. Location.reload()/true
11. Window.history.go(1)
12. History.back() – 1 page back
13. History.forward() – 1 page

14. History.length

19) DOM

1. Document.title

2. Document.domain

3. Document.getElementById

4. Document.getElementsByTagName()

5. Document.getElementsByName()

6. Document.anchors – All anchor with a name attribute

7. Document.forms

8. Document.links – all anchor with a href

9. Document.images

10. Getting attribute's value from getElementById

11. GetAttribute()

12. setAttribute()

13. removeAttribute()

14. document.createElement("div");

15. document.body.appendChild(div);

16. createTextNode()

17. Events

i) Onclick

ii) Removing event handler by setting to null

iii) Add event listener

iv) Remove event listener

v) Event object

vi) Event.preventDefault()

vii) Switch event type and common handler for all events

viii) Event.pageX, event.pageY

ix) Attach eventHandler with setTimeout

x) Hash change event

xi) All the other events

xii) Event.keyCode

18. Form

i) Submit()

- ii) Reset()
 - iii) Form.elements
 - iv) Submit form when enter is pressed
19. Error Handling and Debugging
- i) Try Catch
 - ii) Finally
 - iii) Throw
 - iv) Console.log()
 - v) Console.info()
 - vi) Console.error()
 - vii) Console.warn()
20. Json
- i) Json object and Javascript object difference
 - ii) JSON.stringify()
 - iii) Filtering result in Json.Stringify
 - (1)JSON.stringify(object,[“propname”,[“propname”]])
 - (2)Json.parse(jsonobject)
21. AJAX
- i) Var xhr = new XMLHttpRequest();
 - ii) Xhr.open()
 - iii) Xhr.send
 - iv) Xhr.onreadystatechange

jQuery

Selectors

- All Selector – (“*”)
- Element Selector – (“div”)
- Id Selector
- Select with Class Name – “.myClass”
- Attribute Equals – “[name=’divName’]”

- Attribute not equals - "[name!='divName']"
- Attribute starts with - "[name^='divName']"
- Button Selector - :Button
- Contains - :contains()
- Parent Child Selector
- Ascendant Descendant Selector
- :odd Selector
- :even Selector
- :eq() selector – Index selector
- :first-child
- :first-of-type
- :last-child
- :last-of-type
- :gt() –
- :lt() – all elements less than index
- Multiple attribute selector
- Multiple selector
- :nth-child
- :nth-of-type

Events

- .click()
- .dblclick()
- .on()
- .off()
- .one()
- .focus()
- .focusin()
- .focusout()
- .hover()
- Keydown()
- Keypress()
- Keyup()
- Mouseenter()
- Mouseleave()
- mouseUp()
- mouseDown()
- ready()
- submit()
- trigger()
- event object
- event.pageX

- `event.pageY`
- `change()`

CSS

- `css()`
- `addClass()`
- `hasClass()`
- `removeClass()`
- `toggleClass()`

Manipulation

- `.after()`
- `.before()`
- `.append()`
- `.appendTo()`
- `.attr()`
- `.clone()`
- `.empty()`
- `.html()`
- `.prepend()`
- `.prependTo()`
- `.removeAttr()`
- `.text()`
- `.val()`

Traversing

- `.add()`
- `.addBack()`
- `.children()`
- `.parent()`
- `.closest()`
- `.each()`
- `.find()`
- `.filter()`

Ajax

- `.ajax()`
- `.get()`
- `.getJSON()`

