

Saurabh Shukla Sir

Java is a hot

APRIL 2020

01

2020 Wk 7b

MAY

MONDAY

[JavaScript]

SUN	MON	TUE	WED	THU	FRI	SAT
10	11	12	13	14	15	16
17	18	19	20	21	22	23

- * Javascript is a high level, un-typed and interpreted programming language.
- * Despite some naming, syntactic, and standard library similarities, javascript and java are otherwise unrelated.

[History of javascript]

→ first vision

- * javascript was created in 10 days in May 1995 by Brendan Eich, then working at ~~is a browser free~~ → Netscape and now of mozilla
- * The original name of javascript was Mocha (Name chose by founder of Netscape)
- * In September 1995, name changed to Livescript
- * Again in December 1995, name changed to javascript

JUNE - 2017

M	T	W	T	F	S	S	M	T	W	T	F	S	S
1	2	3	4	5	6	7	8	9	10	11			
12	13	14	15	16	17	18	19	20	21	22	23	24	25
26	27	28	29	30									

MONDAY

TUESDAY

WEDNESDAY

THURSDAY

FRIDAY

SATURDAY

SUNDAY

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

1

2

3</

APRIL - 2017

M	T	W	T	F	S	S	M	T	W	T	F	S
					1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16	17	18	19	20	21
22	23	24	25	26	27	28	29	30				

03

MAY

WEDNESDAY

[In HTML file]

```
<script type="text/javascript">  
...  
</script>
```

[In External file]

* External Script cannot contain the
<script></script> tags!

* `<script type="text/javascript" src="value.js">`

`</script>`

2017

JUNE - 2017

M	T	W	T	F	S	S	M	T	W	T	F	S
1	2	3	4	5	6	7	8	9	10	11		
12	13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30							

MAY

04

THURSDAY

Where to write Javascript

[JavaScript in HTML file]

- * JavaScript code must be inserted between `<Script>` & `</script>` tags.
- * Script can be placed anywhere in HTML document (do body or head)
- * `<script type="text/javascript"> </script>`
- * The type attribute is not required. JavaScript is the default scripting language in HTML

[JavaScript]

- * Javascript is a programming language
- * Javascript statements are separated by semicolons.
- * JavaScript statements are composed of: values, operators, Expressions, keywords, and comments.

APRIL - 2017

M	T	W	T	F	S	S	M	T	W	T	F	S
					1	2	3	4	5	6	7	8
					9	10	11	12	13	14	15	16
					17	18	19	20	21	22	23	24

05

MAY

FRIDAY

Conctns

[JS literals]

* 34.25 number

* 304 number

* "Hello world" String

[JS variables]

* Variables are used to store data values.

* The var keyword is used to declare variables.

var x = 5;

[JS operators]

* Operators are used to manipulate data and gives a data as result.

3 + 4

2 * 5

x = 5

2017

JUNE - 2017

M	T	W	T	F	S	S	M	T	W	T	F	S
1	2	3	4	5	6	7	8	9	10	11		
12	13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30							

06
MAY
SATURDAY

[JS Expressions]

- * An expression is a combination of values, variables, and operators, which computes to a value.

10 $x = 3 + 4 * 5;$

11 $y = "x = " + x;$

[JS keywords]

- * var is a keyword

- * As of now there are 63 keywords in js

What is an Operator?

Operator :-

An Operator is a symbol that tell the compiler which arithmetic or logic operation to be performed between the respective Operands.

Ex:-

Operator \Rightarrow is a symbol (+, -, ÷, x)

give
to

Operand

give to

Operand

\Rightarrow logic operations
arithmetic operation

07

MAY

SUNDAY

APRIL - 2017						
M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

[JS Comments]

- * Code after double slashes // or /*...*/ multiple comments.
- * Comments are ignored, and will not be Executed

[Identifier naming rules]

- * In javascript, the first character must be a letter, an underscore (-), or a dollar sign (\$).
- * Subsequent characters may be letter, digits, underscores, or dollar signs.

* ~~don't~~ first letter can't be digit

bookstore 1234567890 0987654321

1234567890 0987654321

1234567890 0987654321

1234567890 0987654321

1234567890 0987654321

2017

JUNE - 2017

M	T	W	T	F	S	S	M	T	W	T	F	S
1	2	3	4	5	6	7	8	9	10	11	12	13
14	15	16	17	18	19	20	21	22	23	24	25	26
27	28	29	30									

MONDAY

MAY

08

Variables and operators in JS

JavaScript variables

- * JS variable are containers for storing constants
- * The keyword var is used to declare variable.

* var x = 4;
* var y = 3.5;
* var z = "Hello";
* var a = 3 + x;

Naming rules

* Names can contain letter alphabet, digits, underscores, & dollar sign.

* Name must begin with a letter, \$ or -

* Name are case sensitive

allow:

not allow:

var a-1

var 01\$-A

var a.b1 =

var a\$

var 2abl

var -a

var .a

var \$1

09

(129-200) MAY 10

MAY

TUESDAY

APRIL - 2017											
M	T	W	T	F	S	S	M	T	W	T	F
					1	2	3	4	5	6	7
						8	9				
					10	11	12	13	14	15	16
						17	18	19	20	21	22
						23	24	25	26	27	28
							29	30			

[Re-declaration of variables]

- * If you re-declare a javascript variable, it will not lose its value.

Ex:-

`var x = 4;`

`var x;` { value are not lost }

`var x = 5;` { value is lost, new value is 5 }

$x = 5$ $x = 5$

[Concatenation Operation]

using (+) sign

* `var x = "Hello" + " " + "Students";`

* `var y = "2" + 4 + 5;` // $y = "24" + 5$ $y = "245"$

- * Concatenation, in the context of programming, is the operation of joining two strings together. The term "concatenation" literally means to merge two things together. Also known as a string concatenation.

* Also you concatenated two string.

[Assignment Operator]

Store value

* = is an assignment operator

* == is equal to operator

2017

Types of operators

JUNE - 2017

M	T	W	T	F	S	M	I	W	T	F	S	S
						2	3	4	5	6	7	8
12	13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30							

MAY

WEDNESDAY

10

① Arithmetic operators

8	$+ 3 + 4 \quad (?)$	addition
9	$- 3 - 4 \quad (-)$	subtraction
*	$\text{++} \quad (\text{Increment})$	
	$\text{--} \quad (\text{decrement})$	
%	$* \quad 3 + 4 \quad (?)$	What remainder to leave
=	$\% \rightarrow (\text{Modulus}) \rightarrow$	
/	$3 / 4 \quad (0.75)$	
-	$\rightarrow (\text{Division})$	

② Compound assignment operators

= (Simple Assignment)

$$+ = (\text{Add } \ell \text{ to both sides}) \quad x = x + 4$$

$$x = x - 4$$

$$* = x^* = 10 \quad x = x * 10$$

/ = = = = =

3 $\% =$ = =
→ Comparison operators

③ Relational Operators

(3) Relational

\leftarrow false value return
 $=$ (Equal)

$\text{var } x = " - "$ $x = -4$ TRUE

5 \equiv (Equal)
6 $\equiv \equiv$ (
6 \neq (not Equal)

$$\forall x \exists y \exists z (x = 2 \wedge y = 3 \wedge z = 4) \text{ false}$$

\neq (not Equal)

$>$ (Greater than) $3 > 4 \rightarrow \text{false}$

\leq (less than) $3 \leq 4 \leq 5$

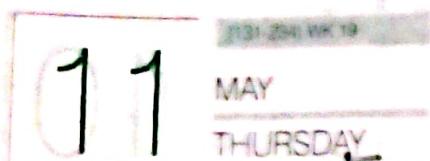
\geq (Greater than or equal to)

\leq (less than or equal to).

You are writing one block page

use configuration

APRIL - 2017



M	T	W	T	F	S	S	M	T	W	F	S
10	11	12	13	14	15	16	17	18	19	20	21
22	23	24	25	26	27	28	29	30			

Control Statements in JS

* JavaScript offers almost the same control statement as we have in C language.

- if, if else, ? :
- Switch
- while, do-while, for, for in

if

```
if (condition) {  
    // code here  
}
```

if else

```
if (condition) {  
    // code here  
} else {  
    // code here  
}
```

else if ladder

```
if (condition 1) {  
    // here code  
} else if (condition 2) {  
    // here code  
} else {  
    // here code  
}
```

2017

JUNE - 2017

M	T	W	T	F	S	S	M	T	W	T	F	S	S
1	2	3	4	5	6	7	8	9	10	11			
12	13	14	15	16	17	18	19	20	21	22	23	24	25
26	27	28	29	30									

WK 19 (22-26)

MAY
FRIDAY

12

{ in left }

? :

Exp 1? exp2 : exp 3;

8

Switch

9. Switch (expression) {

 Case n1:

 // code here

 break;

 Case n2:

 // code here

 break;

 default:

 // code here

}

var t = prompt ("Enter any number");

for (

3. while (condition) {

 // code here

}

For ex:-

5. <html>

<body>

6. <script>

 var x=5, i=1;

 while (i<=10) {

 document.write (x + " x " + i + "=" + x*i);

 i++;

}

</script>

<body>

</html>

2017

APRIL - 2017

M	T	W	T	F	S	S	M	T	W	T	F	S
				1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19	20	21	22
23	24	25	26	27	28	29	30					

13

SATURDAY

MAY

do while

do {

 // Code here
} while (condition);
 <script>Ex:- var x=5, i=1;
 do { document.write(x+"x"+i+" = "+x*i+"
");
 i++;

</script>

Forfor (Statement 1 ; Statement 2 ; Statement 3) {
 // code here
}

Ex:- <script>

var x=8;

for (i=1 ; i<=10 ; i++)

{

 document.write(x+"x"+i+" = "+x*i);
}

}

</script>

2017

<xbox>

window.alert

JUNE - 2017

M	T	W	T	F	S	S	M	T	W	T	F	S	S
1	2	3	4	5	6	7	8	9	10	11			
12	13	14	15	16	17	18	19	20	21	22	23	24	25
26	27	28	29	30									

WR 19 (134-21)

MAY

SUNDAY

14

[PopUp Boxes in JS]

* javascript has three kinds of popup boxes

- Alert box
- Confirm box
- Prompt box

Alert box

* When an alert box pops up, the user will have to click "OK" to proceed.

* `alert ("message to show on alert box");`

Confirm box

* When a confirm box pops up, the user will have to click "OK" or "Cancel" to proceed.

* If the user clicks "OK", the box returns true.
If user clicks "cancel", the box returns false.

* `Confirm ("Some message");`

2017

15

MAY

MONDAY

Fri Tcf

APRIL - 2017

M	T	W	T	F	S	S	M	T	W	T	F	S	S
					1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19	20	21	22	23
24	25	26	27	28	29	30							

```
<Script> if(x) document.body.style.backgroundColor = "red";  
var x = confirm("Shashikumar");  
if(x) document.getElementsByTagName("body")[0].style.backgroundColor = "red";  
backgroundColor = red  
document.getElementsByTagName("body") = red  
<Script>  
backgroundColor = red  
document.getElementsByTagName("body") = red  
backgroundColor = red  
document.getElementsByTagName("body") = red
```

Prompt box

* 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.

* If the user clicks "OK" the box returns the input value. If the user clicks "Cancel" the box returns null.

* `prompt("message", "default value");`

Ex: <Script>

```
var x = prompt("Enter a number", "8");  
if(x != null)  
{  
    for(i=1; i<10; i++)  
        document.write("<br/>" + x + "x" + i + "=" + x * i);  
}
```

JUNE - 2017

M	T	W	T	F	S	S	M	T	W	T	F	S	S
1	2	3	4	5	6	7	8	9	10	11			
12	13	14	15	16	17	18	19	20	21	22	23	24	25
26	27	28	29	30									

MAY

TUESDAY

16

Function in JS

* Function is a block of code designed to perform certain action

9

* Function function name (variable list)

10 {

// code here

11 }

Function execution

* Function is executed when

- function is invoked from another js function
- function is attached with some event and event occurs

3

Why function?

• You can reuse code.

• Script inside function is not executed till function is invoked.

6

Variable in the function

* Local variable

* Global Variable

var x = 1;
x = 5;

2017

17

MAY
WEDNESDAY

APRIL - 2017											
M	T	W	T	F	S	S	M	T	W	F	S
1	2	3	4	5	6	7	8	9	10	11	12
13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30						

<script>

function ashish()

document.write ("Station Ashish");

document.write ("www.Ashish756.com");

}

</script>

<button onclick="ashish ()> My sign </button>

"

Returning Value

* Function may return a value using keyword

return; statement inside function

* It can return many values but function can return only one

* Function can only return one value

* Once a function returns a value, control

returns back to the caller

<script>

function sum(a,b)

{

return (a+b);

}

</script>

<script>

var s = sum(3,4);

document.write ("Sum of 3 and 4 is "+s);

2017 </script>

JUNE - 2017

M	T	W	T	F	S	S	M	T	W	F	S	S
1	2	3	4	5	6	7	8	9	10	11		
12	13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30							



18
THURSDAY

[Event Handling in JS]

Events → response from server, key press, mouse movement

- * Events are actions that can be detected by javascript

- When a user click the mouse
- When a web page has loaded
- When an image has been loaded
- When the mouse move over on element
- When an input field is changed
- When an HTML form is submitted
- When a user strokes a key

Event Handling

- * Sometimes we want to execute a javascript when an event occurs, such as when a user click a button

- * Events are normally used in combination with functions, and the function will not be executed before the event occurs

All HTML

19

MAY
FRIDAY

Event Attributes

APRIL - 2017								
M	T	W	T	F	S	S	M	T
					1	2	3	4
					5	6	7	8
					9	10	11	12
					13	14	15	16
					17	18	19	20
					21	22	23	24
					25	26	27	28
					29	30	31	

~~Event~~ attributes window Event Attribute

Events triggered for the window object (applies to the <body> tag):

Attribute	value	Description
1) onafterprint	Script	Script to be run after the document is printed
2) onbeforeprint	=	Script to be run before the document is printed
3) onbeforeunload	=	Script to be run when the document is about to be unloaded.
4) onerror	=	Script to be run when an error occurs
5) onhashchange	=	Script to be run when there has been changes to the anchor part of the URL.
6) onload	=	Fires after the page is finished loading.
7) onmessage	=	Script to be run when the message is triggered.
8) onoffline	=	Script to be run when the browser starts to work offline.
9) ononline	=	Script to be run when the browser starts to work online

JANUARY - 2018

M	T	W	T	F	S	S	M	T
1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31					

(Contd)

10) onpagehide

=

11) onpopstate

=

12) onresize

=

13) onstorage

=

14) onunload

=

Bootstrap

DECEMBER

TUESDAY

12

Script to be run when a user navigates away from a page

Script to be run when the window's history changes

Fires when the browser window is resized

Script to run when a web storage area is updated

Fires once a page has unloaded
(or the browser window has been closed)

13

(347-018) Wk 50

DECEMBER

WEDNESDAY

NOVEMBER - 2017

M	T	W	T	F	S	S	M	T	W	T	F	S
1	2	3	4	5	6	7	8	9	10	11	12	
13	14	15	16	17	18	19	20	21	22	23	24	25
26	27	28	29	30								

Form Events

- * Events triggered by actions inside a HTML form (applies to almost all HTML elements, but is most used in form elements):

Attribute	=	Description
1) onblur	=	Script to be run when the element loses focus.
2) onchange	=	Fires the moment when the value of the element is changed.
3) oncontextmenu	=	Script to be run when a context menu is triggered.
4) onfocus	=	Fires the moment when the element gets focus.
5) oninput	=	Script to be run when an element gets user input.
6) oninvalid	=	Script to be run when an element is invalid.
7) onreset	=	Fires when the Reset button in a form is clicked.
8) onsearch	=	Fires when the user writes something in a search field (for <input = "Search">)

JANUARY - 2018

M	T	W	T	F	S	S	M	T	W	T	F	S
1	2	3	4	5	6	7	8	9	10	11	12	13
14	15	16	17	18	19	20	21	22	23	24	25	26
27	28	29	30	31								

WK 50 (248-017)

DECEMBER
THURSDAY

14

9) Onselect = Fires after some text has been Selected in an element

10) OnSubmit = Fires when a form is Submitted

Keyboard Events

Attribute	Value	Description
1) Onkeydown	Script	Fires when a user is pressing a key
2) Onkeypress	=	Fires when a user presses a key
3) Onkeyup	=	Fires when a user releases a key

15

2016 WK 50

DECEMBER
FRIDAY

NOVEMBER - 2017

M	T	W	T	F	S	S	M	T	W	T	F	S
1	2	3	4	5	6	7	8	9	10	11	12	
13	14	15	16	17	18	19	20	21	22	23	24	25
26	27	28	29	30								

JANUARY - 2018

M	T	W	T	F	S	S	M	T	W	T	F	S
1	2	3	4	5	6	7	8	9	10	11	12	
13	14	15	16	17	18	19	20	21	22	23	24	25
26	27	28	29	30								

mouse Events

Attribute	Value	Description
1) onclick	Script	Fires on a mouse click on the element
2) ondblclick	=	Fires on a mouse double-click on the element.
3) onmousemove	=	Fires when the mouse pointer is moving while it is over an element
4) onmousedown	=	Fires when a mouse button is pressed down on an element
5) onmouseout	=	Fires when the mouse pointer moves out of an element
6) onmouseover	=	Fires when the mouse over an element
7) onmouseup	=	Fires when the mouse button is released over an element
8) onmousewheel	=	Deprecated. Use the <u>onwheel</u> attribute instead
9) onwheel	=	Fires when the mouse wheel rolls up or down over an element.

8) Attribute

1) ondrag

2) ondragend

3) ondragenter

4) ondragleave

5) ondragover

6) ondragstart

7) ondrop

8) onscroll

T W T F S S
9 10 11 12 13 14
23 24 25 26 27 28

WK 30 DECEMBER
DECEMBER
SATURDAY

16

[Drag Events]

value

Description

=

Script to be run when an element is dragged

=

Script to be run at the end of a drag operation

=

Script to be run when an element has been dragged to a valid drop target

=

Script to be run when an element leaves a valid drop target

=

Script to be run when an element is being dragged over a valid drop target

=

Script to be run at the start of a drag operation

=

Script to be run when dragged element is being dropped.

=

Script to be run when an element's scrollbar is being scrolled.

NOVEMBER - 2017

M	T	W	T	F	S	S	M	T	W	T	F	S
1	2	3	4	5	6	7	8	9	10	11	12	
13	14	15	16	17	18	19	20	21	22	23	24	25
26	27	28	29	30								

17

051-014 WKS

DECEMBER

SUNDAY

Clipboard Events

Attribute	Value	Description
1) oncopy	Script	Fires when the user copies the content of an element.
2) oncut	=	Fires when the user cuts the content of an element.
3) onpaste	=	Fires when the user pastes some content in an element

Media Events

* Events triggered by medias like videos, images and audio (applies to all HTML elements, but is most common in media elements, like `<audio>`, `<embed>`, ``, `<object>`, and `<video>`).

Attribute	Value	Description
1) onabort	Script	Script to be run on abort
2) oncanplay	=	Script to be run when a file is ready to start playing (when it has buffered enough to begin)

2017

JANUARY - 2018

M	W	T	F	S	S	M	T	W	T	F	S	S
1	2	3	4	5	6	7	8	9	10	11	12	13
14	15	16	17	18	19	20	21	22	23	24	25	26
27	28	29	30	31								

Wk 51 (202018)

DECEMBER

MONDAY

18

- 3) oncompletethrough = Script to be run when a file be played all the way to the end without pausing for buffering
- 4) oncuechange = Script to be run when the cue change in a <track> element.
- 5) ondurationchange = Script to be run when the length of the media changes.
- 6) onemptied = Script to be run when something bad happens and the file is suddenly unavailable (like unexpectedly disconnects)
- 7) onended = Script to be run when media has reach the end (a useful for message like "thanks for listening")
- 8) onerror = Script to be run when an error occurs when the file is being loaded
- 9) onloadeddata = Script to be run when media data is loaded.
- 10) onloadedmetadata = Script to be run when meta data (like dimensions and duration) are loaded.

2017

19

DECEMBER

TUESDAY

NOVEMBER

M	T	W	T	F	S	S	M	T	W	T	F
1	2	3	4	5	6	7	8	9	10	11	12
13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30						

Clipboard Events

- | | | |
|-------------------------|---|---|
| 11) <u>onloadstart</u> | = | Script to be run just as the file begins to load before anything is actually loaded. |
| 12) <u>onpause</u> | = | Script to be run when the media is paused either by the user or programmatically. |
| 13) <u>onplay</u> | = | Script to be run when the media is ready to start playing. |
| 14) <u>onplaying</u> | = | Script to be run when the media actually has started playing. |
| 15) <u>onprogress</u> | = | Script to be run when the browser is in the process of getting the meta media data. |
| 16) <u>onratechange</u> | = | Script to be run each time playback rate changes (like when a user to a slow motion or fast forward mode) |
| 17) <u>onseeked</u> | = | Script to be run when the seeking attribute is set to false indicating the seeking has ended. |

JUNE - 2017

M	T	W	T	F	S	S	M	T	W	T	F	S
1	2	3	4	5	6	7	8	9	10	11		
12	13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30							

WK 20 (14-20)

MAY

SATURDAY

20

- 18] onseeking = Script to be run when seeking attributes is set to true indicating that seeking is active.
- 19] onstalled = Script to be run when the browser is unable to fetch the media data for whatever reason.
- 20] onsuspend = Script to be run when fetching the media data is stopped before it is completely loaded for whatever reason.
- 21] ontimeupdate = Script to be run when the playing position has changed (like when the user fast forwards to a different point in the media)
- 22] onvolumechange = Script to be run each time the volume is changed which (includes setting the volume to "mute")
- 23] onwaiting = Script to be run when the media has paused but is expected to resume (like when the media paused to buffer more data)

MISC Events

- ① ontoggle | Script | fires when the user opens or closes the <details> elements

Core JavaScript

getElementsBy

21

(14) 354 Wk 20

MAY

SUNDAY

APRIL - 2017											
M	T	W	T	F	S	S	M	T	W	T	F
1	2	3	4	5	6	7	8	9	10	11	12
13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30						

Editor <html> <head> <script>

- <head> no content
- <script> round to point
- Function effect()
- Function effectback()

```

<html>
<head>
<script>
function effect() {
    var x = document.getElementById("para1");
    x.style.backgroundColor = "lightblue";
}
</script>
</head>
<body>
<h1> Welcome to my page </h1>
<p id="para1" onmouseover="effect()" onmouseout="effectback()"> My Name is: Achint Khatiwala </p>
</body>
</html>

```

6. var getElementsById(id)

7. document.getElementById("para1")

8. var x = document.getElementById("para1")

9. x.style.backgroundColor = "lightblue";

10. function effectback() { }

17

2017

JUNE - 2017												
M	T	W	T	F	S	S	M	T	W	T	F	S
1	2	3	4	5	6	7	8	9	10	11		
12	13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30							

WEDNESDAY

MAY

MONDAY

22

How to access HTML elements in JS

Document Object Methods

- * getElementById() = ~~example~~ ~~pass the id to void~~
- * getElementsByTagName() ~~pass the tag value~~
- * getElementsByName() ~~pass the name~~
- * getElementsByClassName() ~~pass the class name~~

innerHTML print inner Data

```

Ex-
<html>
  <head>
    <script>
      function f1()
      {
        document.getElementById("p1");
        alert(x.tagName);
        alert(x.innerHTML);
        x.innerHTML = "This is a Third Paragraph";
      }
    </script>
  </head>
  <h1> Welcome </h1>
  <p id="p1"> This is a paragraph </p>
  <p> the is a second paragraph </p>
  <button onclick="f1()"> Click </button>

</body>
</html>

```

2017

23

MAY
TUESDAY

APRIL - 2017								
M	T	W	T	F	S	S	M	T
1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30						

How to Set CSS properties in JS

Style

* Obj. style.property = "value"

```
<html>
  <head>
```

```
  <script>
    function f1()
```

```
      document.getElementById("p1");
```

```
      x.style.color = "Red";
```

```
      x.style.backgroundColor = "Black";
```

```
      x.style.fontSize = "25px";
```

```
</script>
```

```
</head>
```

```
<body>
```

```
  <h1> Welcome to my page </h1>
```

```
  <p id="p1"> Welcome to my paragraph </p>
```

```
<button onclick="f1"> Click me </button>
```

```
</body>
```

```
</html>
```

2017

JUNE - 2017

M	T	W	T	F	S	S	M	T	W	F	S	S
1	2	3	4	5	6	7	8	9	10	11		
12	13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30							

MAY

WEDNESDAY

24

form validation in javascript

- * Form is an html element to take information from the user, using various input elements.

```

9 < form action = "Some Uri" method = "get" >
10 < input type = "text" name = "user" />
11 < input type = "password" name = "pwd" />
12 < input type = "number" name = "age" />
< input type = "Submit" />
< /form >
```

```

1 < html >
2   < head >
3     < script >
4       function validation()
5         var result = true;
6         var i = document.getElementsByTagName("input");
7         if (i[0].value.length == 0)
8           result = false;
9         return (result);
10    < /script >
11  < body >
12    < form action = "registration.php" method = "get"
13      onsubmit = "return validation()" >
14      Name < input type = "text" name = "username" />
15    < /form >
16  < /body >
17 < /html >
```