jQuery Cheat Sheet

Selectors

General
#id .class tag *
ancestor descendant
parent > child
target + next
target ~ siblings

Attributes
[attr] exists
[attr=val] equals
[attr!=val] not equal
[attr^=val] begins
[attr$=val] ends
[attr*=val] contains

Forms
:input :reset
:text :button
:password :file
:radio :enabled
:checkbox :disabled
:submit :checked
:image :selected

Filters
:first / :last
:even / :odd
:eq(index)
:gt(index)
:lt(index)
:header
:animated
:first-child
:last-child
:only-child
:empty / :parent
:visible / :hidden
:nth-child(num)
:contains(text)
:has(selector)
:not(selector)

Events

Event Obj
e.data e.which
e.target
e.currentTarget
e.relatedTarget
e.pageX e.pageY
e.result
e.timeStamp

Shortcut
.blur(fn) .change(fn)
.click(fn) .dblclick(fn)
.focus(fn) .focusin(fn)
.focusout(fn)
.keydown(fn) .keyup(fn)
.keypress(fn)
.mouseover(fn) .mouseout(fn)
.mouseenter(fn) .mouseleave(fn)
.mousemove(fn) .mouseup(fn)
.submit(fn)

Window | Elements
load(fn) .resize(fn)
.unload(fn) .scroll(fn)

.live (type, data, fn)
.die (tyep, handler)
.delegate (selector, data, type, fn)
.undelegate (selector, type, handler)

General
.bind (type, data, fn)
.unbind (type)
.one (type, data, fn)
.toggle (fn1, fn2)
.hover (overfn, outfn)
.trigger (type, data)
.triggerHandler (t, d)

Effects

General
.fadeIn(d,c) .fadeout(d,c)
.fadeTo(d,opacity,c)
.slideDown(d,c) .slideUp(d, c)
slideToggle(d,c) .show(d,c)
.hide(d,c) .toggle(d,c)
.toggle(boolean)
.clearQueue( ) .queue(name,newFn)
.dequeue(name)
.delay(ms) .stop( )
.stop(true) $.fx.off

Animate
.animate({properties}, d, e, c)
.animate({properties},{options})

Traverse

.add(s) (selector) | .children(s)
.closest(s) | .contents()
.eq(s) | .filter(s) | .find(s)
.first() | .has(s) | .is(s)
.last() | .next(s) | .nextAll(s)
.nextUntil(s) | .not(s)
.offsetParent() | .parent(s)
.parents(s) | .parentsUntil(s)
.prev(s) | .prevAll(s) | .prevUntil(s)
.siblings(s) | .slice(start,end)

Values

CSS | Classes
.css(name)
.css(name, value)
.css({properties})
.addClass(class)
.removeClass(class)
.hasClass(class)
.toggleClass(class)

Sizes
.height(num)
.width(num)
.innerHeight()
.innerWidth()
.outerHeight(boolean)
.outerWidth(boolean)

Forms
.val()
.val(value)
.serialize()
.serializeArray()
$.param(obj)

Attributes
.attr(name)
.attr(name, value)
.attr({properties})
.removeAttr(name)

HTML
.html(string)
.html()
.text(string)
.text()

Position
.offset() .top .left
.position() .top .left
.scrollLeft(num)
.scrollTop(num)

Manipulating

Clone
.clone()
.clone(true)

Parent
.wrap(h)
.wrapAll(h)
.wrapInner(h)

Siblings
.after(h)
.before(h)
.insertAfter(s)
.insertBefore(s)

Children
.append(h)
.appendTo(s)
.prepend(h)
.prependTo(s)

Replace
.replaceWith(h)
.replaceAll(s)

Deleting
.empty()
.remove(s)
.detach(s)
.unwrap(s)

AJAX

Setups
.ajaxComplete(c)
.ajaxError(c)
.ajaxSuccess(c)
.ajaxStart(c)
.ajaxStop(c)
.ajaxSend(c)

$.ajaxSetup({options})

Requests
$.get(u, d, c, t)
$.getJSON(u, d, c, t)
$.getScript(u, c)
$.post(u, d, c, t)
.load(u, d, c)

$.ajax({options})
url, timeout, async, cache, type, contentType
data, dataFilter, dataType, global, ifModified
jscrip, jsonpCallback

complete (xhr, status)
success (data, status, xhr)
error(xhr, status, err)
beforeSend(xhr)

Utilities

.length
.context
.each(callback)
.index(obj)
.get(index)
.data(name)
.data(name, value)
.removeData(name)

$.noConflict()
$.support
$.proxy(fn, context)
$.parseJSON(string)
$.contains(parent, obj)
$.each(obj, callback)
$.extend(deep, (), objN)
$.trim(string)
$.unique(array)

$.merge(array, array)
$.inArray (value, array)
$.grep(array, fn, invert)
$.map(array, fn)
.map(fn)

JavaScript

Methods

Object
toString
toLocaleString
valueOf
hasOwnProperty
isPrototypeOf
propertyIsEnumerable

String
charAt
charCodeAt
fromCharCode
concat
indexOf
lastIndexOf
localeCompare
match
replace
search
slice
split
substring
substr
toLowerCase
toUpperCase
toLocaleLowerCase
toLocaleUpperCase

RegEx
test
match
exec

Array
concat
join
push
pop
reverse
shift
slice
sort
splice
unshift

Number
toFixed
toExponential
toPrecision

Date
parse
toDateString
toTimeString
getDate
getDay
getFullYear
getHours
getMilliseconds
getMinutes
getMonth
getSeconds
getTime
getTimezoneOffset
getYear
setDate
setHours
setMilliseconds
setMinutes
setMonth
setSeconds
setYear
toLocaleTimeString

XMLHttpRequest

Safari, Mozilla, Opera:

```
var req = new XMLHttpRequest();
```

Internet Explorer:

```
var req = new
ActiveXObject("Microsoft.XMLHTTP");
```

XMLHttpRequest Object Methods

abort()
getAllResponseHeaders()
getResponseHeader(header)
open(method, URL)
send(body)
setRequestHeader(header, value)

XMLHttpRequest Object Properties

onreadystatechange
readyState
responseText
responseXML
status
statusText

XMLHttpRequest readyState Values

0	Uninitiated
1	Loading
2	Loaded
3	Interactive
4	Complete

JAVASCRIPT IN HTML

External JavaScript File

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

Inline JavaScript

```
<script type="text/javascript">
  <!--
     // JavaScript Here
  //-->
</script>
```

Functions

Window
alert
blur
clearTimeout
close
focus
open
print
setTimeout

Built In
eval
parseInt
parseFloat
isNaN
isFinite
decodeURI
decodeURIComponent
encodeURI
encodeURIComponent
escape
unescape

REGULAR EXPRESSIONS - FORMAT

Regular expressions in JavaScript take the form:

var RegEx = **/pattern/modifiers**;

REGULAR EXPRESSIONS - MODIFIERS

/g	Global matching
/i	Case insensitive
/s	Single line mode
/m	Multi line mode

REGULAR EXPRESSIONS - PATTERNS

^	Start of string
$	End of string
.	Any single character
(a|b)	a or b
(...)	Group section
[abc]	Item in range (a or b or c)
[^abc]	Not in range (not a or b or c)
a?	Zero or one of a
a*	Zero or more of a
a+	One or more of a
a{3}	Exactly 3 of a
a{3,}	3 or more of a
a{3,6}	Between 3 and 6 of a
!(pattern)	"Not" prefix. Apply rule when URL does not match pattern.

EVENT HANDLERS

onAbort	onMouseDown
onBlur	onMouseMove
onChange	onMouseOut
onClick	onMouseOver
onDblClick	onMouseUp
onDragDrop	onMove
onError	onReset
onFocus	onResize
onKeyDown	onSelect
onKeyPress	onSubmit
onKeyUp	onUnload
onLoad	

FUNCTIONS AND METHODS

A method is a type of function, associated with an object. A normal function is not associated with an object.

Available free from www.ILoveJackDaniels.com

DOM Methods

Document
clear
createDocument
createDocumentFragment
createElement
createEvent
createEventObject
createRange
createTextNode
getElementsByTagName
getElementById
write

Node
addEventListener
appendChild
attachEvent
cloneNode
createTextRange
detachEvent
dispatchEvent
fireEvent
getAttributeNS
getAttributeNode
hasChildNodes
hasAttribute
hasAttributes
insertBefore
removeChild
removeEventListener
replaceChild
scrollIntoView

Form
submit

DOM Collections
item

Range
collapse
createContextualFragment
moveEnd
moveStart
parentElement
select
setStartBefore

Style
getPropertyValue
setProperty

Event
initEvent
preventDefault
stopPropagation

XMLSerializer
serializeToString

XMLHTTP
open
send

XMLDOM
loadXML

DOMParser
parseFromString

Regular Expressions Syntax

^	Start of string
$	End of string
.	Any single character
(a|b)	a or b
(...)	Group section
[abc]	In range (a, b or c)
[^abc]	Not in range
\s	White space
a?	Zero or one of a
a*	Zero or more of a
a*?	Zero or more, ungreedy
a+	One or more of a
a+?	One or more, ungreedy
a{3}	Exactly 3 of a
a{3,}	3 or more of a
a{,6}	Up to 6 of a
a{3,6}	3 to 6 of a
a{3,6}?	3 to 6 of a, ungreedy
\	Escape character
[:punct:]	Any punctuation symbol
[:space:]	Any space character
[:blank:]	Space or tab

There's an excellent regular expression tester at: http://regexpal.com/

Pattern Modifiers

g	Global match
i *	Case-insensitive
m *	Multiple lines
s *	Treat string as single line

Pattern Modifiers (cont)

x *	Allow comments and whitespace in pattern
e *	Evaluate replacement
U *	Ungreedy pattern

* PCRE modifier

JavaScript RegExp Object

compile()	lastParen
exec()	leftCOntext
global	multiline
ignoreCase	rightContext
input	source
lastIndex	test()
lastMatch	

JavaScript Event Handlers

onabort	onmousedown
onblur	onmousemove
onchange	onmouseout
onclick	onmouseover
ondblclick	onmouseup
ondragdrop	onmove
onerror	onreset
onfocus	onresize
onkeydown	onselect
onkeypress	onsubmit
onkeyup	onunload
onload	

JavaScript Arrays

concat()	slice()
join()	sort()
length	splice()
pop()	toSource()
push()	toString()
reverse()	unshift()
shift()	valueOf()

JavaScript Numbers and Maths

abs()	min()
acos()	NEGATIVE_INFINITY
asin()	PI
atan()	POSITIVE_INFINITY
atan2()	pow()
ceil()	random()
cos()	round()
E	sin()
exp()	sqrt()
floor()	SQRT1_2
LN10	SQRT2
LN2	tan()
log()	toSource()
LOG10E	toExponential()
LOG2E	toFixed()
max()	toPrecision()
MAX_VALUE	toString()
MIN_VALUE	valueOf()
NaN	

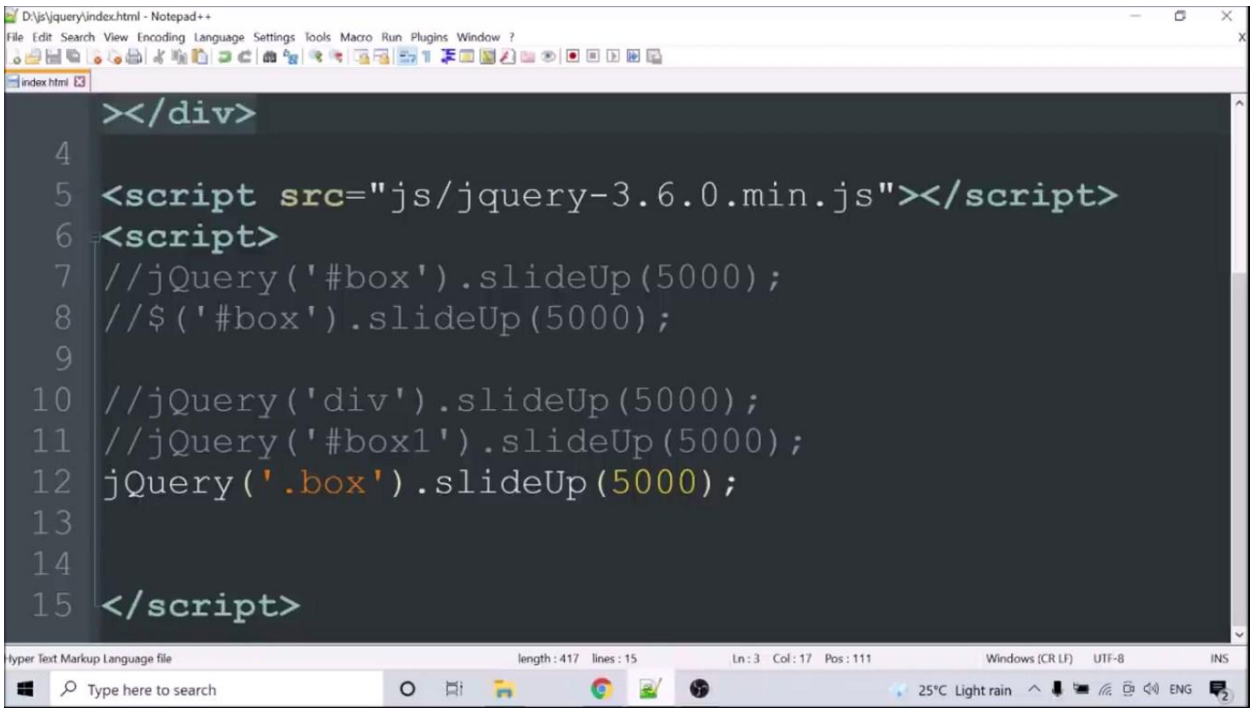
JavaScript Booleans	
toSource()	valueOf()
toString()	

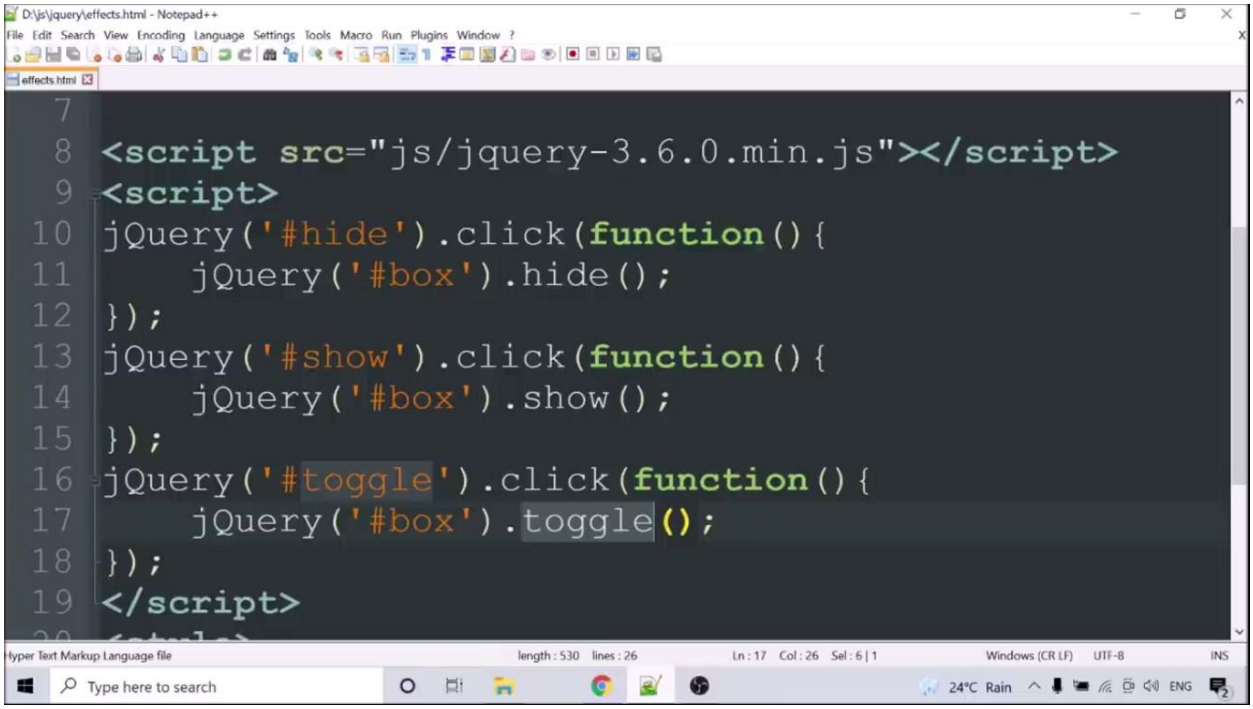
JavaScript Dates	
Date()	setMonth()
getDate()	setFullYear()
getDay()	setHours()
getMonth	setMinutes()
getFullYear	setSeconds()
getYear	setMilliseconds()
getHours	setTime()
getMinutes	setUTCDate()
getSeconds	setUTCDay()
getMilliseconds	setUTCMonth()
getTime	setUTCFullYear()
getTimezoneO-ffset()	setUTCHours()
getUTCDate()	setUTCMinutes()
getUTCDay()	setUTCSeconds()
getUTCMonth()	setUTCMilliseco-nds()
getUTCFullYear()	toSource()
getUTCHours()	toString()
getUTCMinutes()	toGMTString()
getUTCSeconds()	toUTCString()
getUTCMilliseco-nds()	toLocaleString()
parse()	UTC()
setDate()	valueOf()

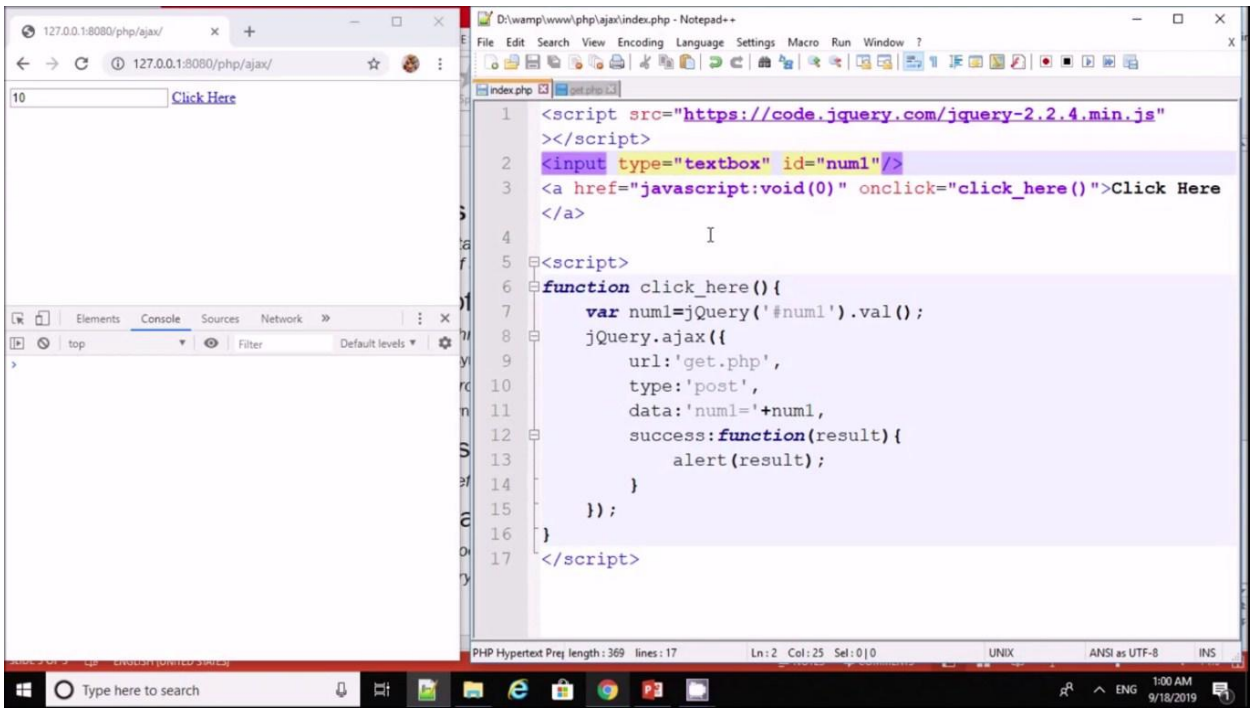
JavaScript Strings	
charAt()	slice()
charCodeAt()	split() x
concat()	substr()
fromCharCode()	substring()
indexOf()	toLowerCase()
lastIndexOf()	toUpperCase()
length	toLocaleLowerCase()
localeCompare()	toLocaleUpperCase()
match() x	toSource()
replace() x	valueOf()
search() x	

String object methods with an x support regular expressions.

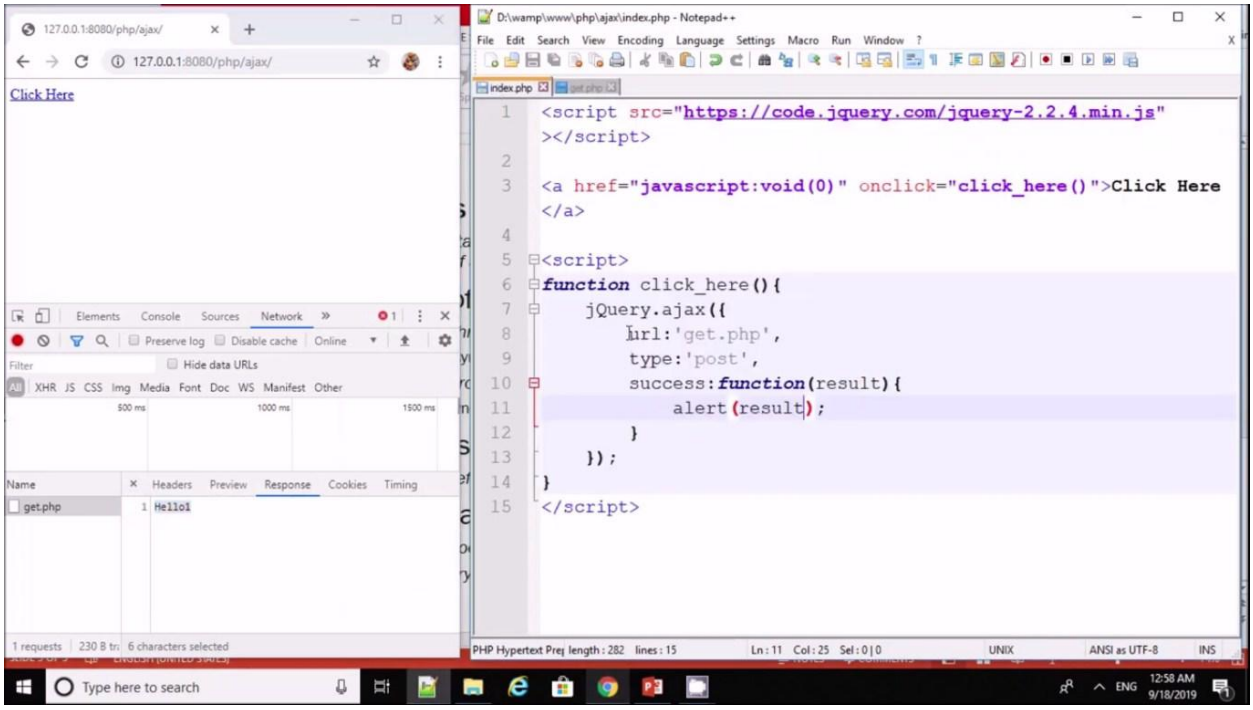
JavaScript Functions	
decodeURI()	isNaN()
decodeURIComponent()	Number()
encodeURI()	parseFloat()
encodeURIComponent()	parseInt()
escape()	String()
eval()	unescape()
isFinite()	


D:\js\jquery\index.html - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
index.html
>
<script src="js/jquery-3.6.0.min.js"></script>
<script>
//jQuery('#box').slideUp(5000);
//$('#box').slideUp(5000);
//jQuery('div').slideUp(5000);
//jQuery('#box1').slideUp(5000);
jQuery('.box').slideUp(5000);
</script>
Hyper Text Markup Language file
length : 417 lines : 15
Ln : 3 Col : 17 Pos : 111
Windows (CR LF) UTF-8 INS
Type here to search
25°C Light rain
ENG

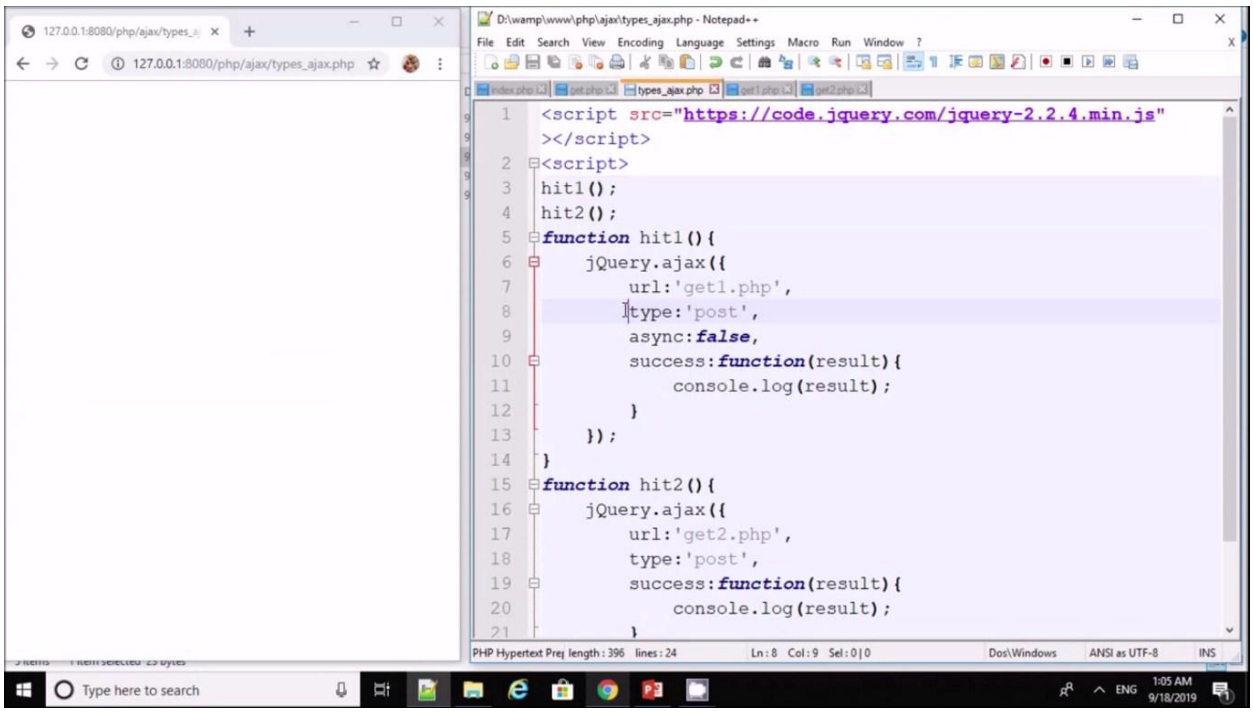

D:\js\jquery\effects.html - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
effects.html
<script src="js/jquery-3.6.0.min.js"></script>
<script>
jQuery('#hide').click(function(){
jQuery('#box').hide();
});
jQuery('#show').click(function(){
jQuery('#box').show();
});
jQuery('#toggle').click(function(){
jQuery('#box').toggle();
});
</script>
Hyper Text Markup Language file
length : 530 lines : 26
Ln : 17 Col : 26 Sel : 6 | 1
Windows (CR LF) UTF-8 INS
Type here to search
24°C Rain
ENG

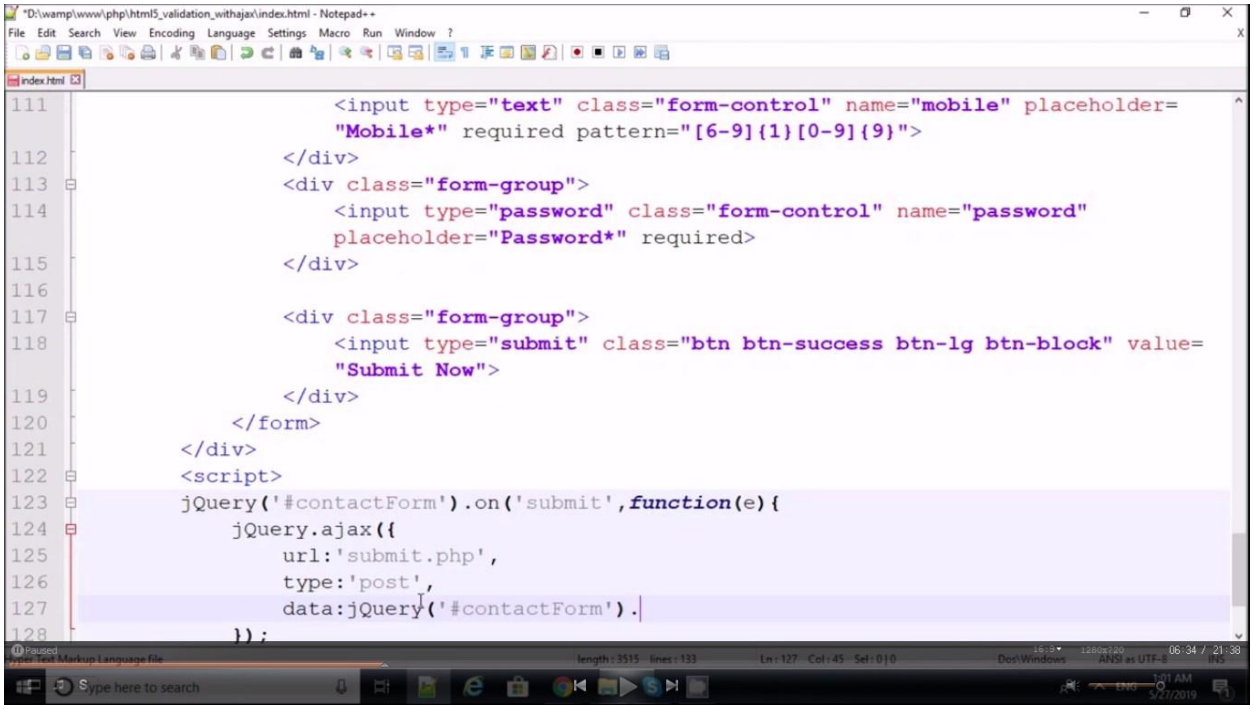

127.0.0.1:8080/php/ajax/
10
Click Here
Elements
Console
Sources
Network
top
Filter
Default levels
D:\wamp\www\php\ajax\index.php - Notepad++
File Edit Search View Encoding Language Settings Macro Run Window ?
index.php
get.php
1 <script src="https://code.jquery.com/jquery-2.2.4.min.js"
></script>
2 <input type="textbox" id="num1"/>
3 Click Here

4
5 <script>
6 function click_here(){
7 var num1=jQuery('#num1').val();
8 jQuery.ajax({
9 url:'get.php',
10 type:'post',
11 data:'num1='+num1,
12 success:function(result){
13 alert(result);
14 }
15 });
16 }
17 </script>
PHP Hypertext Prej length : 369 lines : 17
Ln : 2 Col : 25 Sel : 0 | 0
UNIX
ANSI as UTF-8
INS
Type here to search
ENG
1:00 AM
9/18/2019


127.0.0.1:8080/php/ajax/
Click Here
Elements
Console
Sources
Network
Preserve log
Disable cache
Online
Filter
Hide data URLs
All XHR JS CSS Img Media Font Doc WS Manifest Other
500 ms
1000 ms
1500 ms
Name
Headers
Preview
Response
Cookies
Timing
get.php
1 Hello1
1 requests
230 B tr
6 characters selected
D:\wamp\www\php\ajax\index.php - Notepad++
File Edit Search View Encoding Language Settings Macro Run Window ?
index.php
get.php
1 <script src="https://code.jquery.com/jquery-2.2.4.min.js"
></script>
2
3 Click Here

4
5 <script>
6 function click_here(){
7 jQuery.ajax({
8 url:'get.php',
9 type:'post',
10 success:function(result){
11 alert(result);
12 }
13 });
14 }
15 </script>
PHP Hypertext Prej length : 282 lines : 15
Ln : 11 Col : 25 Sel : 0 | 0
UNIX
ANSI as UTF-8
INS
Type here to search
ENG
12:58 AM
9/18/2019


127.0.0.1:8080/php/ajax/types_ajax.php
D:\wamp\www\php\ajax\types_ajax.php - Notepad++
File Edit Search View Encoding Language Settings Macro Run Window ?
types_ajax.php
<script src="https://code.jquery.com/jquery-2.2.4.min.js"
></script>
<script>
hit1();
hit2();
function hit1(){
jQuery.ajax({
url:'get1.php',
type:'post',
async:false,
success:function(result){
console.log(result);
}
});
}
function hit2(){
jQuery.ajax({
url:'get2.php',
type:'post',
success:function(result){
console.log(result);
PHP Hypertext Prej length : 396 lines : 24
Ln : 8 Col : 9 Sel : 0 | 0
Dos\Windows
ANSI as UTF-8
INS
Type here to search
ENG
1:05 AM
9/18/2019

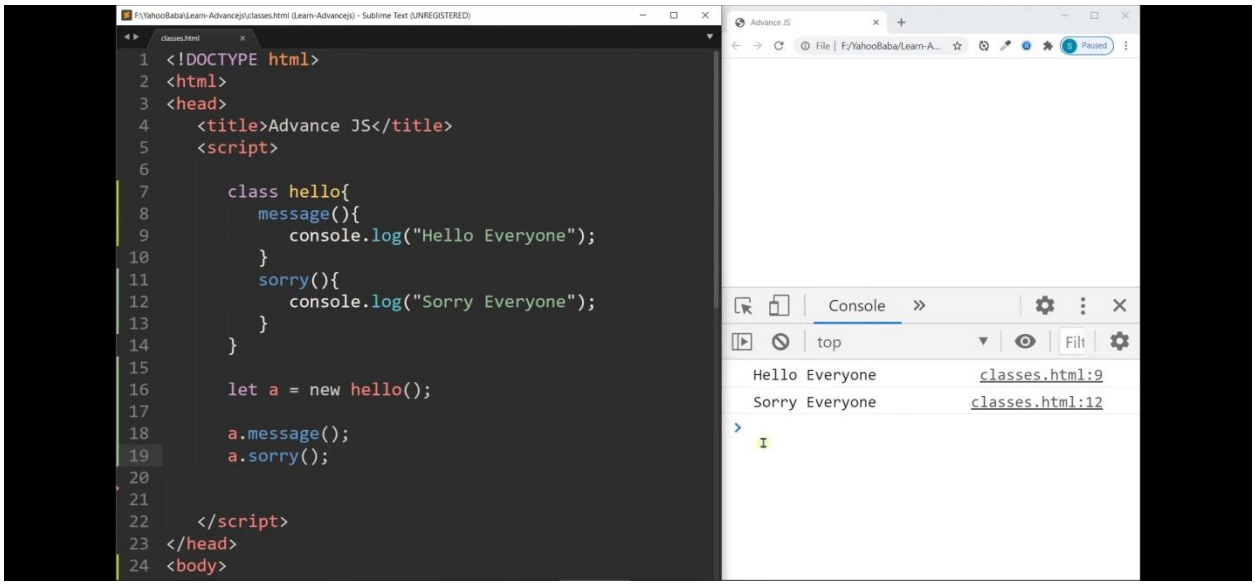

*D:\wamp\www\php\html5_validation_withajax\index.html - Notepad++
File Edit Search View Encoding Language Settings Macro Run Window ?
index.html
<input type="text" class="form-control" name="mobile" placeholder=
"Mobile*" required pattern="[6-9]{1}[0-9]{9}">

<input type="password" class="form-control" name="password"
placeholder="Password*" required>

<input type="submit" class="btn btn-success btn-lg btn-block" value=
"Submit Now">

</form>

<script>
jQuery('#contactForm').on('submit',function(e){
jQuery.ajax({
url:'submit.php',
type:'post',
data:jQuery('#contactForm').
});
length : 3515 lines : 133
Ln : 127 Col : 45 Sel : 0 | 0
Dos\Windows
ANSI as UTF-8
Type here to search
1:01 AM
5/27/2019


F:\YahooBaba\Learn-Advancejs\classes.html (Learn-Advancejs) - Sublime Text (UNREGISTERED)
classes.html
<!DOCTYPE html>
<html>
<head>
<title>Advance JS</title>
<script>
class hello{
message(){
console.log("Hello Everyone");
}
sorry(){
console.log("Sorry Everyone");
}
}
let a = new hello();
a.message();
a.sorry();
</script>
</head>
<body>
Advance JS
File | F:/YahooBaba/Learn-A...
Paused
Console
top
Hello Everyone
classes.html:9
Sorry Everyone
classes.html:12

JS Class & Object in PHP

```
class hello{
    message(){
      console.log("Hello Everyone");
    }

}

let a = new hello();

a.message();
```


JS Type of Methods :

Constructor

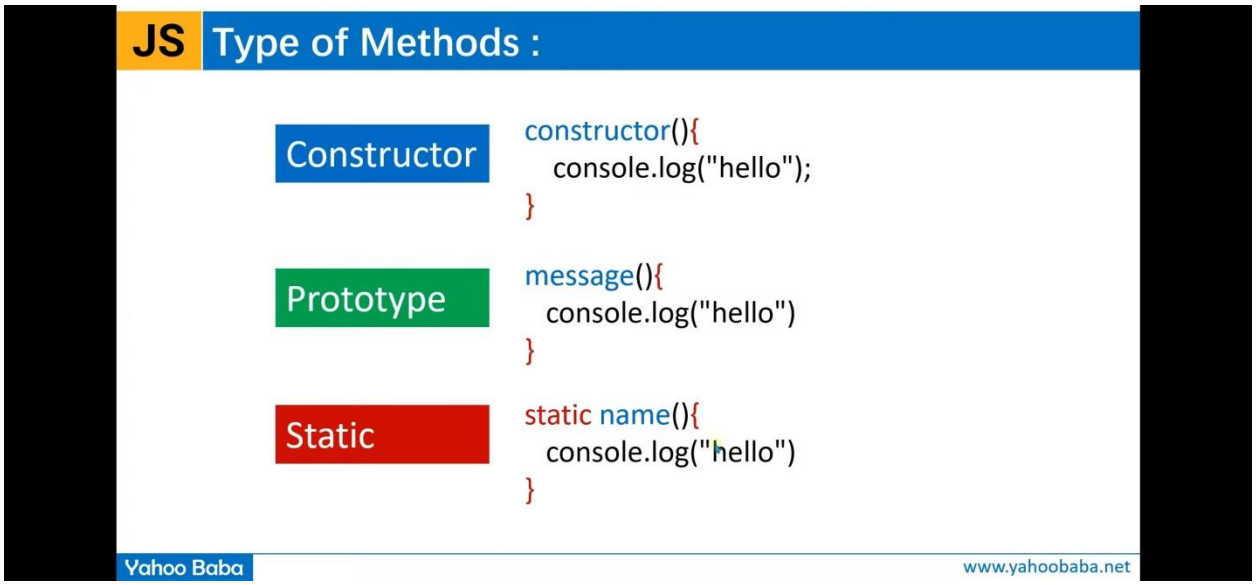
```
constructor(){
   console.log("hello");
}
```

Prototype

```
message(){
  console.log("hello")
}
```

Static

```
static name(){
  console.log("hello")
}
```



Pillar of OOPS

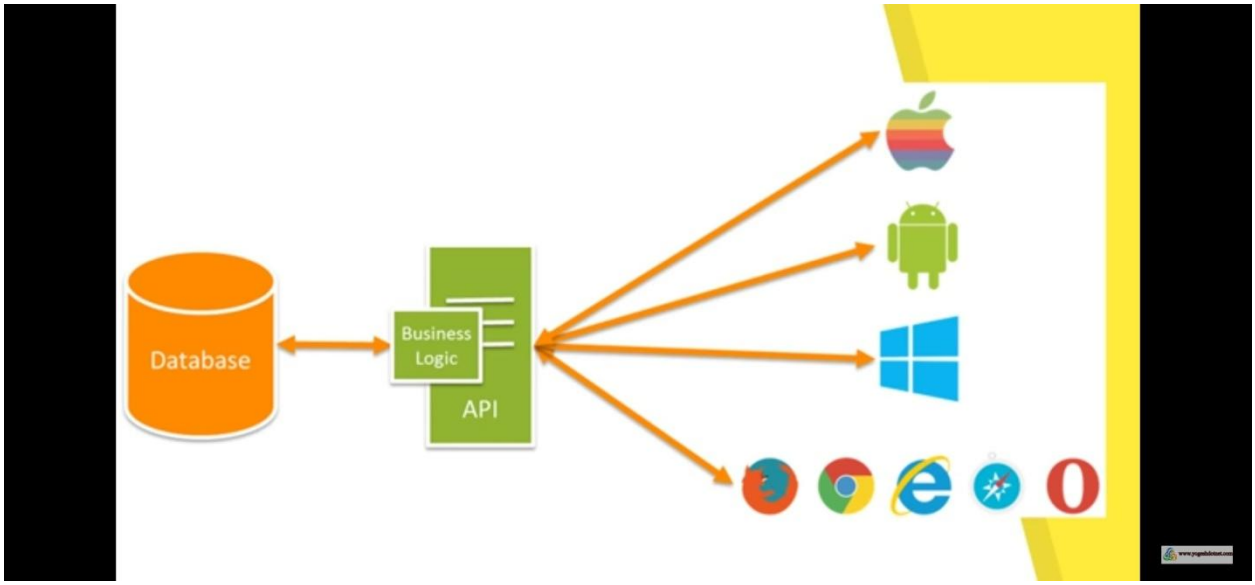
- Data Abstraction(Data hide)
- Inheritance. (Reusability)
- Polymorphism. (Object to take many forms)
- Encapsulation. (Data hide)

Activate Windows
Go to Settings to activate Windows.

REST API

CRUD Operations:-

Operation	HTTP Methods	Description
Create	POST	Creating/Posting/Inserting Data
Read	GET	Reading/Getting/Retrieving Data
Update	PUT, PATCH	Updating Data Complete Update - PUT Partial Update - PATCH
Delete	DELETE	Deleting Data

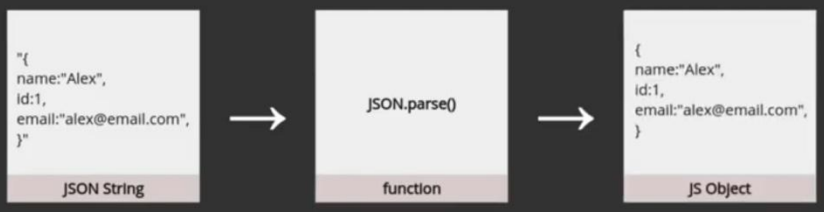


JSON.stringify()

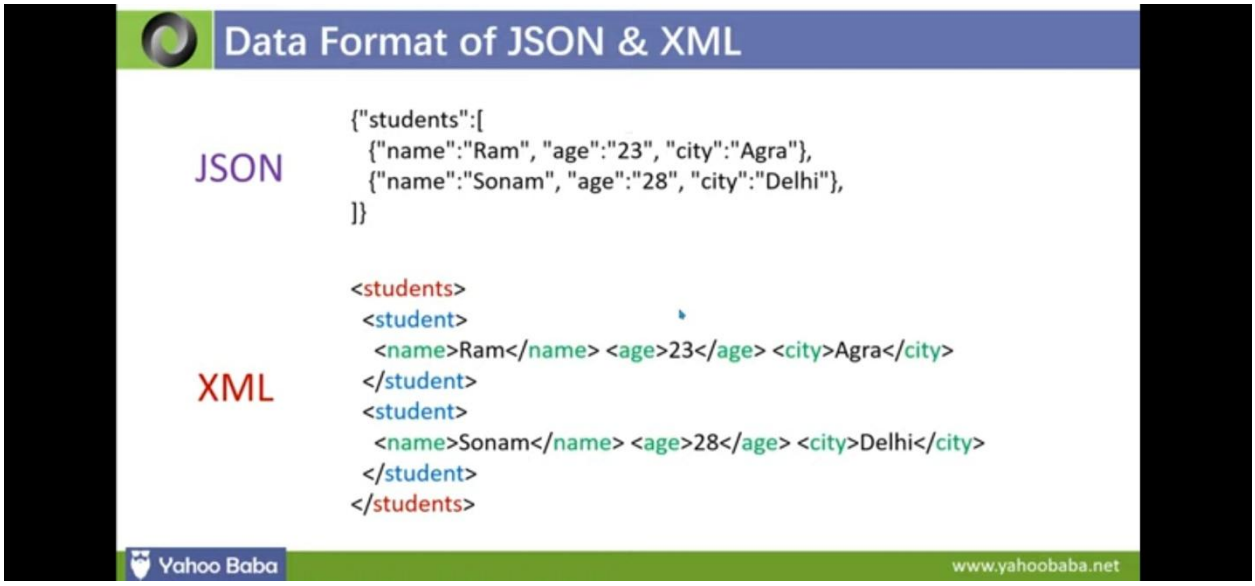
Transforms a JavaScript object into a JSON string.

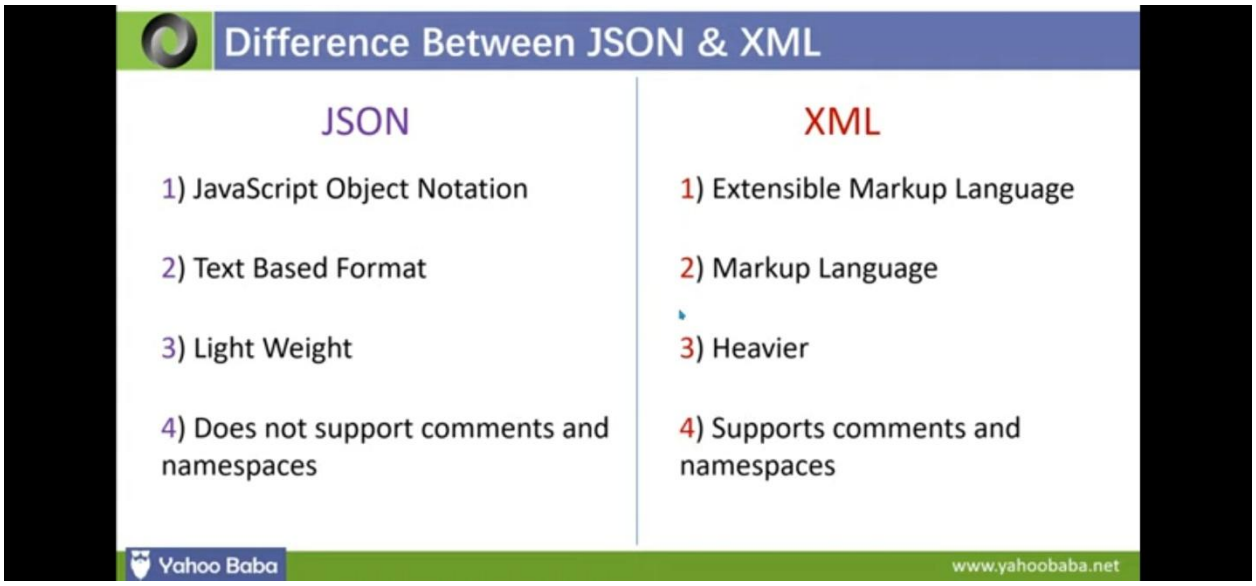
JSON.parse()

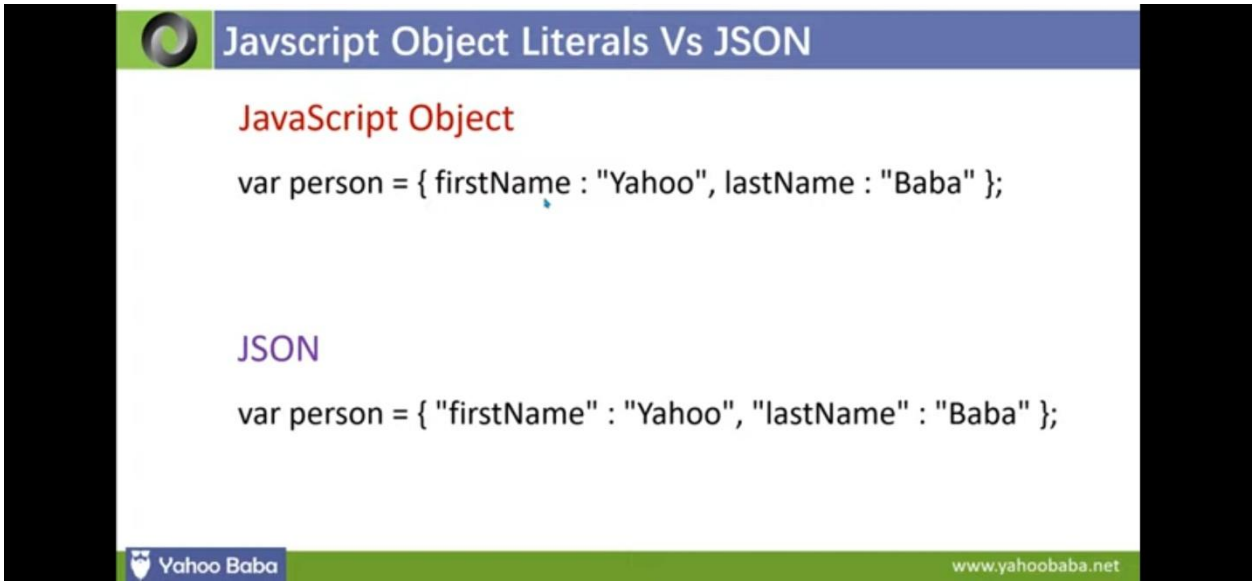
Takes a JSON string and transforms it into a JavaScript object.



```
> const student = {
          name:"Alex",
          age:20,
          email:"alex@email.com"
  };

> const studentStr = JSON.stringify(student);

> console.log(studentStr);

  {"name":"Alex","age":20,"email":"alex@email.com"}

> const jsObject = JSON.parse(studentStr);

> console.log(jsObject);

  Object { age: 20, email: "alex@email.com", name: "Alex" }

>
```




Difference Between JSON & XML
JSON
1) JavaScript Object Notation
2) Text Based Format
3) Light Weight
4) Does not support comments and namespaces
XML
1) Extensible Markup Language
2) Markup Language
3) Heavier
4) Supports comments and namespaces
Yahoo Baba
www.yahoobaba.net


Javscript Object Literals Vs JSON
JavaScript Object
var person = { firstName : "Yahoo", lastName : "Baba" };
JSON
var person = { "firstName" : "Yahoo", "lastName" : "Baba" };
Yahoo Baba
www.yahoobaba.net

Javscript Object Literals Vs JSON

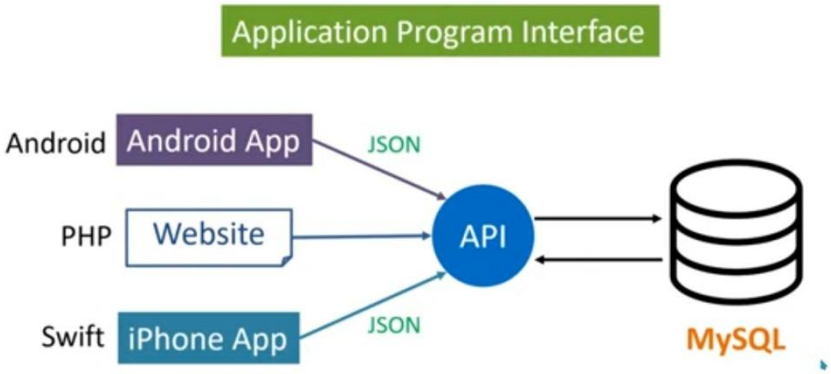
JavaScript Object

```
var person = { firstName : "Yahoo", lastName : "Baba" };
alert(person.firstName + " " + person.lastName);
```

JSON

```
var person = { "firstName" : "Yahoo", "lastName" : "Baba" };
alert(person.firstName + " " + person.lastName);
```


What is API ?



Advantages of JSON :

- Human Readable Format
- Language Independent
- Supports all Popular Programming languages
- Easy to organised and access
- It is light-weight

Can not use it for transfer video, audio, images or any other binary information.

What we learn in next videos ?

- How to read JSON data in Website Front End
- How to produce JSON data in PHP
- How to work with APIs

What is Object ?

0 1 2 3

```
var a = ["Ram", "Kumar", 18 ,"India"];

var a = {
        firstName : "Ram" ,
        lastName : "Kumar" ,
        age : 18 ,
        country : "India"
};
```

Yahoo Baba

How to Target DOM Object :

- Id → document.getElementById(id)
- Class Name → document.getElementsByClassName(name)
- Tag Name → document.getElementsByTagName(name)

Yahoo Baba

Other DOM Targeting Methods :

- document
- document.all
- document.documentElement
- document.head
- document.title
- document.body
- document.images
- document.anchors
- document.links
- document.forms
- document.doctype
- document.URL
- document.baseURI
- document.domain

Yahoo Baba

DOM Get Methods :

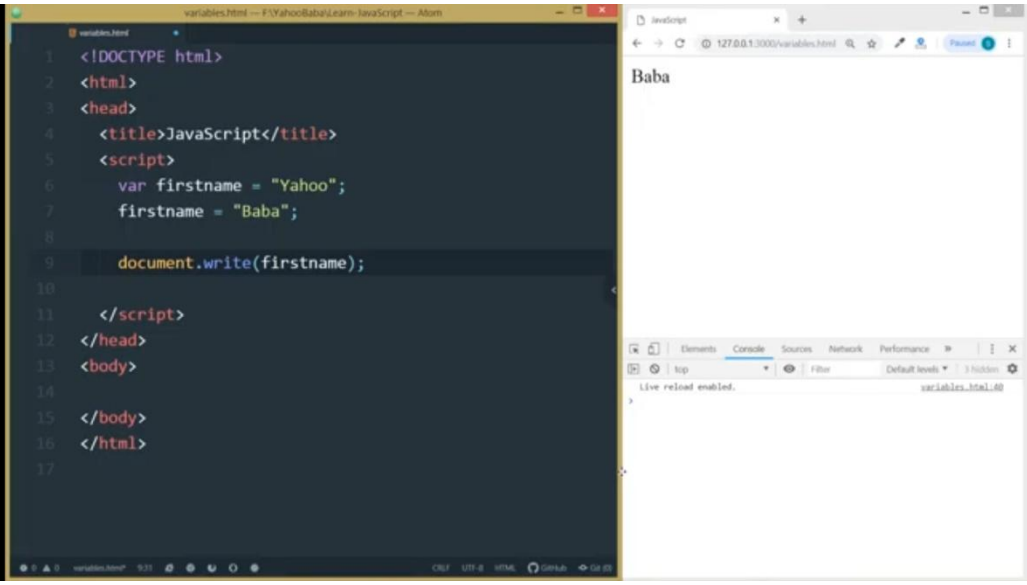
- innerText
- innerHTML
- getAttribute
- getAttributeNode
- Attributes

Yahoo Baba

DOM Set Methods :

- innerText
- innerHTML
- setAttribute
- Attribute
- removeAttribute

Yahoo Baba



Yahoo Baba

Difference between Variable Types:

Var	Let	Const
var x = “Hello”;	let x = “Hello”;	const x = “Hello”;
var x = “World”;	let x = “World”;	const x = “World”;
x = “WoW”;	x = “WoW”;	x = “WoW”;

Different Type of Assignment Operators :

Operator	Example	Same As
=	x = y	x = y
+=	x += y	x = x + y
-=	x -= y	x = x - y
*=	x *= y	x = x * y
/=	x /= y	x = x / y
%=	x %= y	x = x % y
**=	x **= y	x = x ** y

Yahoo Baba

Different Type of Data Types :

```
var x = "Hello World";        → String
var x = 25;                   → Number
var x = true;                 → Boolean
var x = ["HTML","CSS","JS"];  → Array
var x = {first:"Jane", last:"Doe"};  → Object
var x = null;                 → Null
var x;                        → Undefined
```

Yahoo Baba

Different Type of Comparison Operators :

Operator	Description
==	equal to
===	equal value and equal type
!=	not equal
!==	not equal value or not equal type
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to

Yahoo Baba

JavaScript Basic Events

- Click
- Double Click
- Right Click
- Mouse Hover
- Mouse Out
- Mouse Down
- Mouse Up
- Key Press
- Key Up
- Load
- Unload
- Resize
- Scroll

Yahoo Baba

For Loop Syntax in JavaScript :

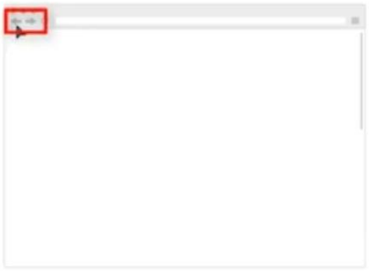
Initialization Condition Increment / Decrement

```
for(var a = 1; a <= 10; a++){
   document.write("Yahoo Baba");
}
```

Yahoo Baba

Browser Object Model

- Get Width & Height of Browser Window
- Open & Close Browser Window
- Move & Resize Browser Window
- Scroll to Browser Window
- Get URL, Hostname, Protocol of Browser Window
- Get History of Browser Window



Yahoo Baba

JavaScript Math Methods :

- ceil(x)
- floor(x)
- round(x)
- trunc(x)
- max(x, y, z, ..., n)
- min(x, y, z, ..., n)
- sqrt(x)
- cbrt(x)
- pow(x, y)
- random()
- abs(x)
- PI

Yahoo Baba

JavaScript Math Methods :

- toDateString()
- getDate()
- getFullYear()
- getMonth()
- getDay()
- getHours()
- getMinutes()
- getSeconds()
- getMilliseconds()
- setDate()
- setFullYear()
- setHours()
- setMilliseconds()
- setMinutes()
- setMonth()
- setSeconds()

Yahoo Baba

JavaScript Number Methods :

- number()
- parseInt()
- parseFloat()
- isFinite()
- isInteger()
- toFixed(x)
- toPrecision(x)

Yahoo Baba

JavaScript Array Methods :

- sort()
- reverse()
- pop()
- push()
- shift()
- unshift()
- concat()
- join()
- slice()
- splice()
- isArray()
- indexOf()
- lastIndexOf()
- entries()
- every()
- filter()
- find()
- findIndex()
- includes()
- some()
- forEach()
- toString()
- valueOf()
- fill()

Yahoo Baba

For / in Loop :

```
var a = {
        firstName : "Ram" ,
        lastName : "Kumar" ,
        age : 18 ,
        country : "India"
};
```

```
for(var i in a){
   Statement
}
```

Yahoo Baba

Array – Map() Function :

```
var a = [1 , 3 , 5 , 8 , 10];

a.map(function(){
  Statement
});
```

Yahoo Baba

How to Target DOM Object :

- Id ⟶ document.getElementById(id)
- Class Name ⟶ document.getElementsByClassName(name)
- Tag Name ⟶ document.getElementsByTagName(name)

Yahoo Baba

Other DOM Targeting Methods :

- document
- document.all
- document.documentElement
- document.head
- document.title
- document.body
- document.images
- document.anchors
- document.links
- document.forms
- document.doctype
- document.URL
- document.baseURI
- document.domain

Yahoo Baba

JavaScript Basic Events

- Click (onclick)
- Double Click (ondblclick)
- Right Click (oncontextmenu)
- Mouse Hover (onmouseenter)
- Mouse Out (onmouseout)
- Mouse Down (onmousedown)
- Mouse Up (onmouseup)
- Key Press (onkeypress)
- Key Up (onkeyup)
- Load (onload)
- Unload (onunload)
- Resize (onresize)
- Scroll (onscroll)

Yahoo Baba

jQuery Quick API Reference | 3.0 | Search... | Preferences...

SELECTORS

Basics
*
.class
element
#id
selector1, selectorN, ...

Hierarchy
parent > child
ancestor descendant
prev + next
prev ~ siblings

Basic Filters
:animated
:eq()
:even
:first
:gt()
:header
:lang()
:last
:lt()
:not()
:odd
:root
:target

Content Filters
:contains()
:empty
:has()
:parent

Visibility Filters
:hidden
:visible

Attribute
[name|="value"]
[name*="value"]
[name~="value"]
[name$="value"]
[name="value"]
[name!="value"]
[name^="value"]
[name]
[name="value"][name2="value2"]

Child Filters
:first-child
:first-of-type
:last-child
:last-of-type
:nth-child()
:nth-last-child()
:nth-last-of-type()
:nth-of-type()
:only-child
:only-of-type()

Forms
:button
:checkbox
:checked
:disabled
:enabled
:focus
:file
:image
:input
:password
:radio
:reset
:selected
:submit
:text

ATTRIBUTES / CSS

Attributes
.attr()
.prop()
.removeAttr()
.removeProp()
.val()

CSS
.addClass()
.css()
jQuery.cssHooks
jQuery.cssNumber
jQuery.escapeSelector()
.hasClass()
.removeClass()
.toggleClass()

Dimensions
.height()
.innerHeight()
.innerWidth()
.outerHeight()
.outerWidth()
.width()

Offset
.offset()
.offsetParent()
.position()
.scrollLeft()
.scrollTop()

Data
jQuery.data()
.data()
jQuery.hasData()
jQuery.removeData()
.removeData()

MANIPULATION

Copying
.clone()

DOM Insertion, Around
.wrap()
.wrapAll()
.wrapInner()

DOM Insertion, Inside
.append()
.appendTo()
.html()
.prepend()
.prependTo()
.text()

DOM Insertion, Outside
.after()
.before()
.insertAfter()
.insertBefore()

DOM Removal
.detach()
.empty()
.remove()
.unwrap()

DOM Replacement
.replaceAll()
.replaceWith()

Short Selectors Example :

```
document.getElementsByClassName('classname')
$('.classname')

document.getElementById('idname')
$('#idname')

document.getElementsByTagName('tagname')
$('tagname')
```

Yahoo Baba

JavaScript

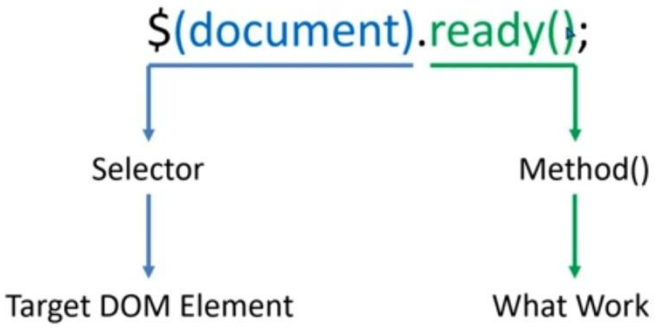
```
<script>
    var a = document.getElementById("idName").innerHTML();
    console.log(a);
</script>
```

jQuery

```
<script>
    $(document).ready(function(){

        Further jQuery Code

    });
</script>
```

Yahoo Baba

jQuery Basic Syntax



Yahoo Baba

Another Method to start jQuery Code :

```
$(document).ready(function(){
});
```

```
$(function(){
});
```

Yahoo Baba

Targeting DOM Element with ID, Class & Tag

Select by Id : $("#idName")

Select by Class Name : $(".className")

Select by Tag Name : $("tagName") ⟶ $("p")

Yahoo Baba

Advance Selectors

$("*")

$("ul li")

$(".abc , .xyz")

$("h1, div, p")

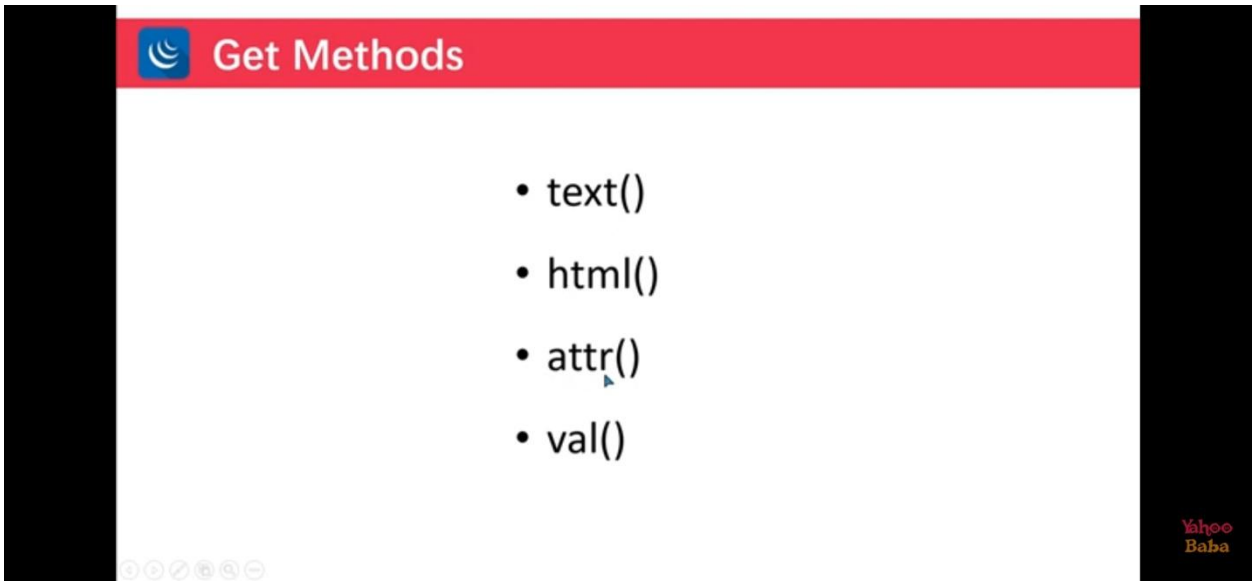
$("p:first")

$("p:last")

$("li:even")

$("li:odd")

Yahoo Baba


Get Methods
• text()
• html()
• attr()
• val()
Yahoo Baba

5:49 4 Vo) LTE1 Vo) LTE2 LTE 69%

Syntax

```
$(document).ready(function(){
$(".demo").click(function(){
        $(this).hide(200);
});
});
$(function(){
// Short Document Ready
});
```

Each

```
$(".demo").each(function() {
document.write($(this).text() + "\n");  // output
});
```

Trigger

```
$("a#mylink").trigger("click"); // triggers event or
```

noConflict

```
var jq = $.noConflict();            // avoid conflic
jq(document).ready(function(){
jq("#demo").text("Hello World!");
});
```

Selectors

```
$("*")                      // all elements
$("p.demo")                 // <p> elements with cla
$("p:first")                // the first <p> element
$("p span")                 // span, descendant of p
$("p > span")               // span, direct child of
$("p + span")               // span immediately proc
$("p ~ span")               // strong element proce
$("ul li:first")            // the first <li> elemer
$("ul li:first-child")      // the first <li> elemer
$("ul li:nth-child(3)")     // third child
$("[href]")                 // any element with an h
$("a[target='_blank']")     // <a> elements with a t
$("a[target!='_blank']")    // <a> elements with a t
$(":input")                 // all form elements
$(":button")                // <button> and <input>
$("tr:even")                // even <tr> elements
$("tr:odd")                 // odd <tr> elements
$("span:parent")            // element which has chi
$("span:contains('demo')")  // element conaining the
```

Actions

$(selector).action()

```
$(this).hide()      // the current element
$("div").hide()     // all <div> elements
$(".demo").hide()   // all elements with class="demc
$("#demo").hide()   // the element with id="demo"
```

Events

```
$(".demo").click(function(){
$(this).hide(200);
});
```

Mouse

scroll, click, dblclick, mousedown, mouseup, mousemove, mouseover, mouseout, mouseenter, mouseleave, load, resize, scroll, unload, error

Keyboard

keydown, keypress, keyup

Form

submit, change, focus, blur

DOM Element

blur, focus, focusin, focusout, change, select, submit

5:50 4 Vo)) LTE1 Vo)) LTE2 LTE 69%

Syntax

```
$(document).ready(function(){
$(".demo").click(function(){
        $(this).hide(200);
});
});
$(function(){
// Short Document Ready
});
```

Each

```
$(".demo").each(function() {                    // pa
document.write($(this).text() + "\n");  // outpu
});
```

Trigger

```
$("a#mylink").trigger("click"); // triggers event or
```

noConflict

```
var jq = $.noConflict();            // avoid conflic
jq(document).ready(function(){
jq("#demo").text("Hello World!");
});
```

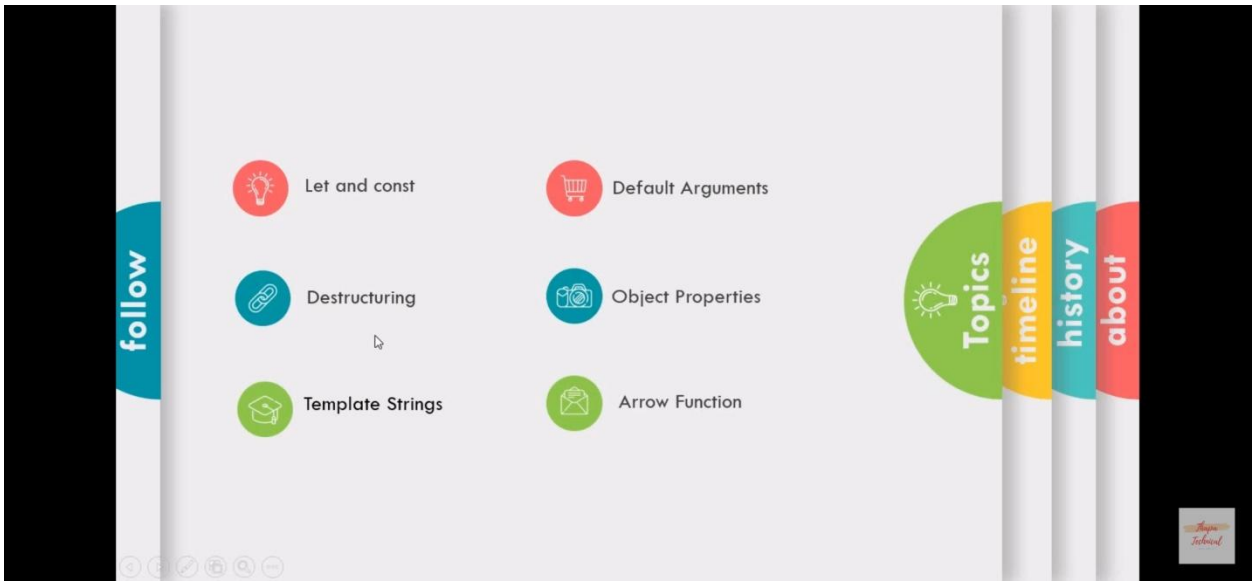
Selectors

```
$("*")                      // all elements
$("p.demo")                 // <p> elements with cla
$("p:first")                // the first <p> element
$("p span")                 // span, descendant of p
$("p > span")               // span, direct child of
$("p + span")               // span immediately proc
$("p ~ span")               // strong element procee
$("ul li:first")            // the first <li> elemer
$("ul li:first-child")      // the first <li> elemer
$("ul li:nth-child(3)")     // third child
$("[href]")                 // any element with an h
$("a[target='_blank']")     // <a> elements with a t
$("a[target!='_blank']")    // <a> elements with a t
$(":input")                 // all form elements
$(":button")                // <button> and <input>
$("tr:even")                // even <tr> elements
$("tr:odd")                 // odd <tr> elements
$("span:parent")            // element which has chi
$("span:contains('demo')")  // element conaining the
```

Actions

$(selector).action()

```
$(this).hide()      // the current element
$("div").hide()     // all <div> elements
$(".demo").hide()   // all elements with class="demo
```



JS Promise Syntax

```
let prom = new Promise(function(resolve, reject){
    if(condition){
        resolve();
    else{
        reject();
    }
});
```

```
let onfulfilment = (result) => {
    console.log(result);
}

let onRejection = (error) => {
    console.log(error);
}
```

JS Fetch() Syntax

```
fetch();

fetch(file / URL);  ──────→ Promise

fetch(file / URL).then();  ──────→ Promise

fetch(file / URL).then(function(response){
    return response.data;
});
```

text() json()

Yahoo Baba

JS Different type of Variable :

Var	Let (ES6)	Const (ES6)
var x = "Hello";	let x = "Hello";	const x = "Hello";
var x = "World";	let x = "World"; ⊗	const x = "World"; ⊗
x = "WoW";	x = "WoW";	x = "WoW"; ⊗

Yahoo Baba

JS What is Fetch() Method ?

AJAX

jQuery

$.ajax();

$.get();

$.post();

JavaScript

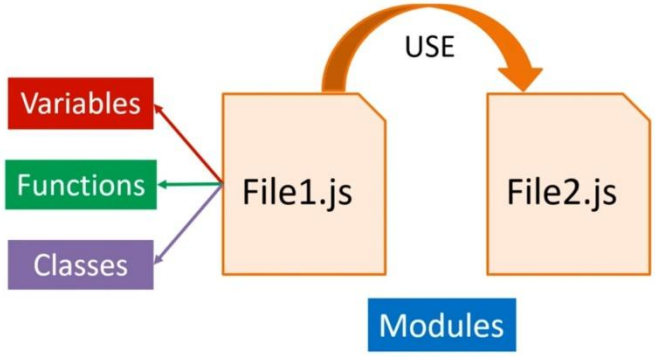
XMLHttpRequest

JavaScript ES6

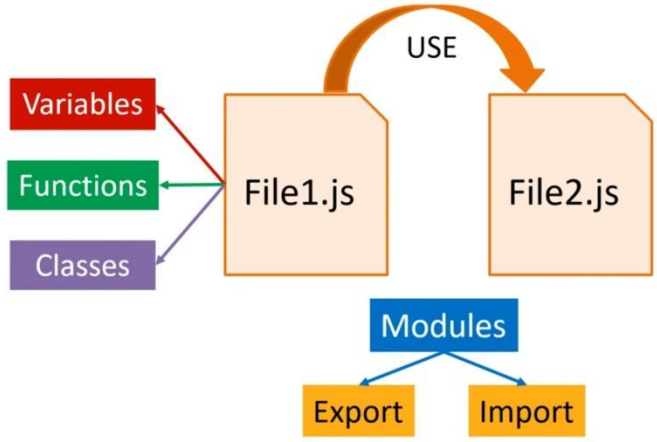
Fetch()

- Insert
- Update
- Read
- Delete

JS What is Modules ?



JS What is Modules ?



JS How to work with Modules ?

File1.js

```
export let name = "Yahoo Baba";

export function hello{

        }

export class user{

        }
```

File2.js

```
import { name } from './File1.js'
import { hello } from './File1.js'
import { user } from './File1.js'

console.log(name);
hello();
let a = new user();
```

HTML File `<script type="module" src="./File2.js"></script>`

JS JavaScript : Data Types

```
var x = "Hello World";   → String
var x = 25;              → Number
var x = true;            → Boolean
var x = ["HTML","CSS","JS"];   → Array
var x = {first:"Jane", last:"Doe"};   → Object
var x = null;            → Null
var x;                   → Undefined
```


JS Symbol Data Type

```
var x = Symbol();
var x = Symbol("Hello");   → Unique Value
var y = Symbol("Hello");
console.log(x === y)   → False
```

JS JavaScript : Generators

```
function *test() {
    yield "First";
    yield "Second";
    yield "Third";
}

let g = test();
g.next();
g.next();
g.next();
```

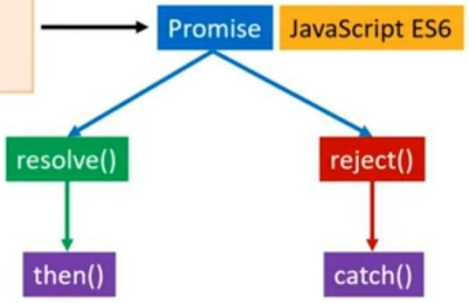
Pause ← yield "First";

g.next(); → First

JS Async Function

```
async function test(){

}
```

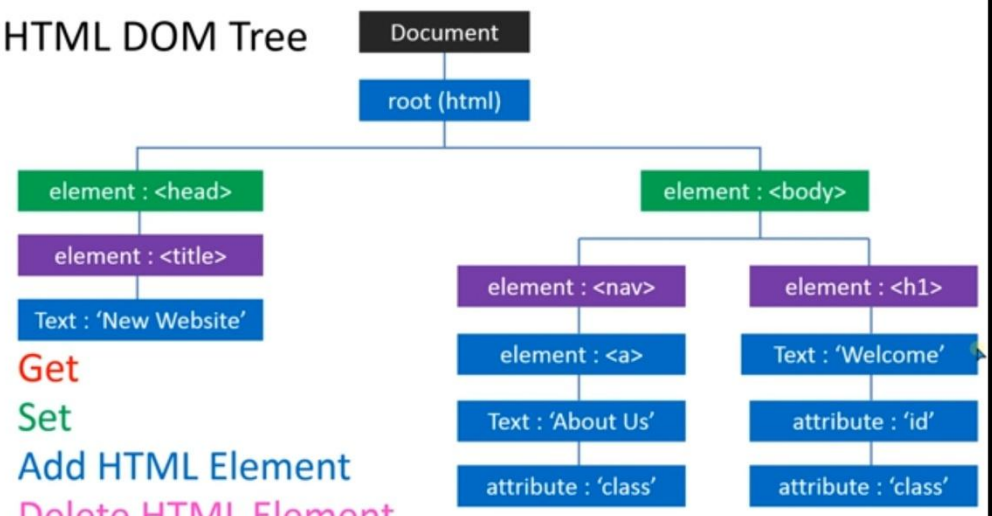
JavaScript ES 2017



JS Await Method

```
async function test(){
    console.log("A");
await console.log("B");
    console.log("C");
}

test();
console.log("D");
console.log("E");
```


HTML DOM Tree



Get
Set
Add HTML Element
Delete HTML Element

DOM Get Methods :

- innerText
- innerHTML
- getAttribute
- getAttributeNode
- Attributes

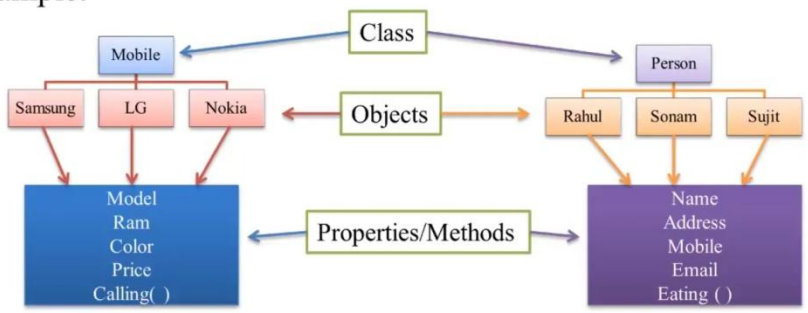
DOM Set Methods :

- innerText
- innerHTML
- setAttribute
- Attribute
- removeAttribute

Class

A specific category can be defined as class.

Example:-



www.geekyshows.com

Defining a Class

We define class in JavaScripts using custom constructor.

```
var Mobile = function(model_no, sprice) {
        this.model = model_no;
        this.color = 'white';
        this.price = 3000;
        this.sp = sprice;
        this.sellingprice = function() {
                return (this.price + this.sp);
        };
};
var samsung = new Mobile('Galaxy', 2000);
```

www.geekyshows.com

Defining a Class

We define class in JavaScripts using custom constructor.

```
var Mobile = function(model_no, sprice) {
        this.model = model_no;
        this.color = 'white';
        this.price = 3000;
        this.sp = sprice;
        this.sellingprice = function() {
                return (this.price + this.sp);
        };
};
var samsung = new Mobile('Galaxy', 2000);
var nokia = new Mobile('3310', 1000);
```

www.geekyshows.com

Defining a Class

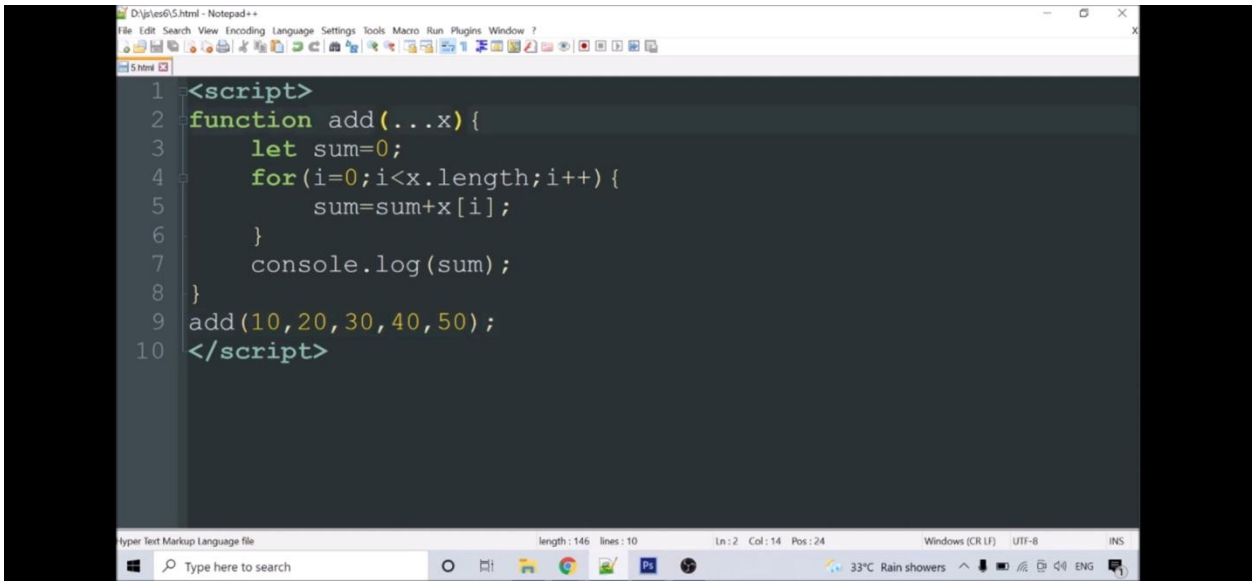
We define class in JavaScripts using custom constructor.

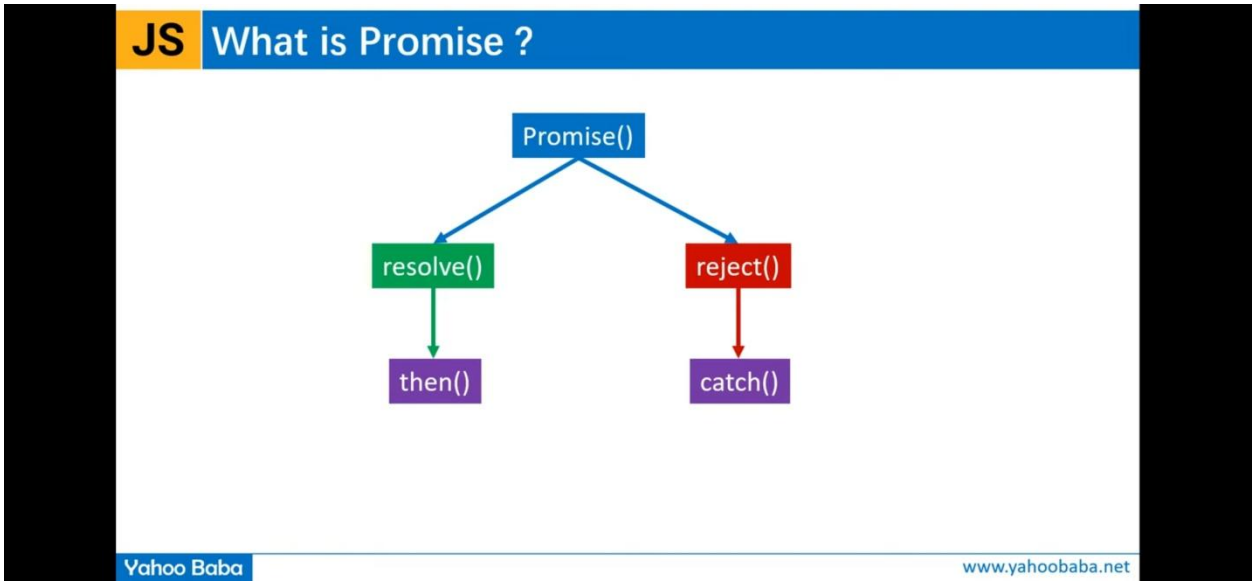
```
var Mobile = function(mod
        this.model = mo
        this.color = 'white';
        this.price = 3000;
        this.sp = sprice;
        this.sellingprice = function() {
                return (this.price + this.sp);
        };
};
var samsung = new Mobile('Galaxy', 2000);
var nokia = new Mobile('3310', 1000);
```

ES6 Class

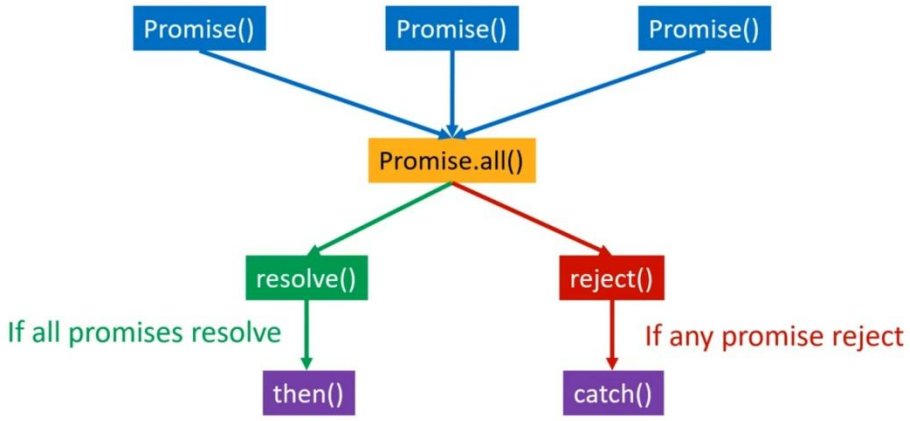
```
class Mobile {
        constructor ( ) {
            this.model = 'Galaxy';
        }
        show() { return this.model +
                        "Price 3000";
        }
}
var nokia = new Mobile( );
```

www.geekyshows.com


D:\js\es6\5.html - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
5.html
<script>
function add(...x){
let sum=0;
for(i=0;i<x.length;i++){
sum=sum+x[i];
}
console.log(sum);
}
add(10,20,30,40,50);
</script>
Hyper Text Markup Language file
length : 146 lines : 10
Ln : 2 Col : 14 Pos : 24
Windows (CR LF)
UTF-8
INS
Type here to search
33°C Rain showers
ENG


JS
What is Promise ?
Promise()
resolve()
reject()
then()
catch()
Yahoo Baba
www.yahoobaba.net

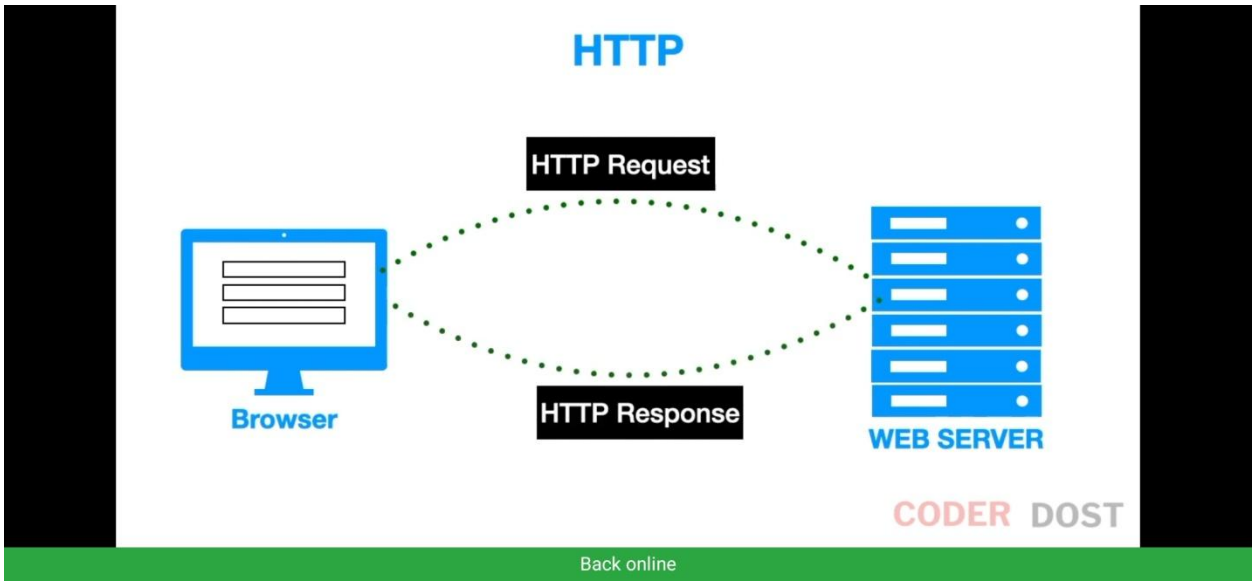
JS What is Promise.all() ?



Promise : .then() & .catch()

```
const p  = new Promise(function(resolve,reject){

})
p.then(function(data){

}).catch(function(data){

})
```

CODER DOST



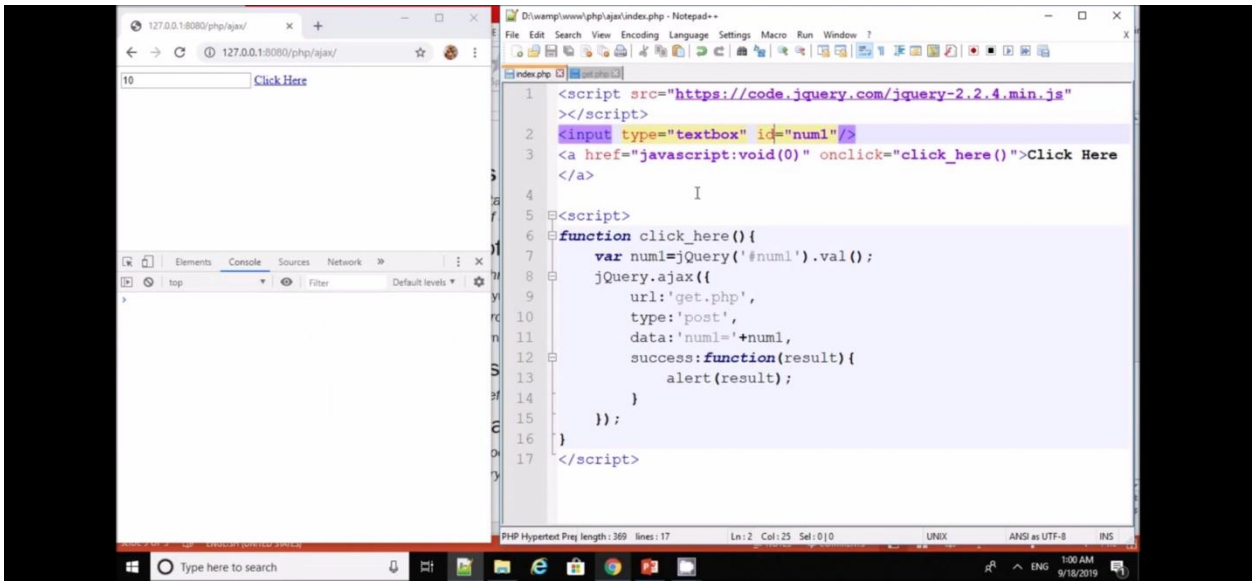
Back online

Recursion in Function

It is the process in which a function calls itself again and again to perform any repetitive task.

For Example

```
function fact(a)
{
If(a==1)
return 1;
else
return a*fact(a-1);
}
```

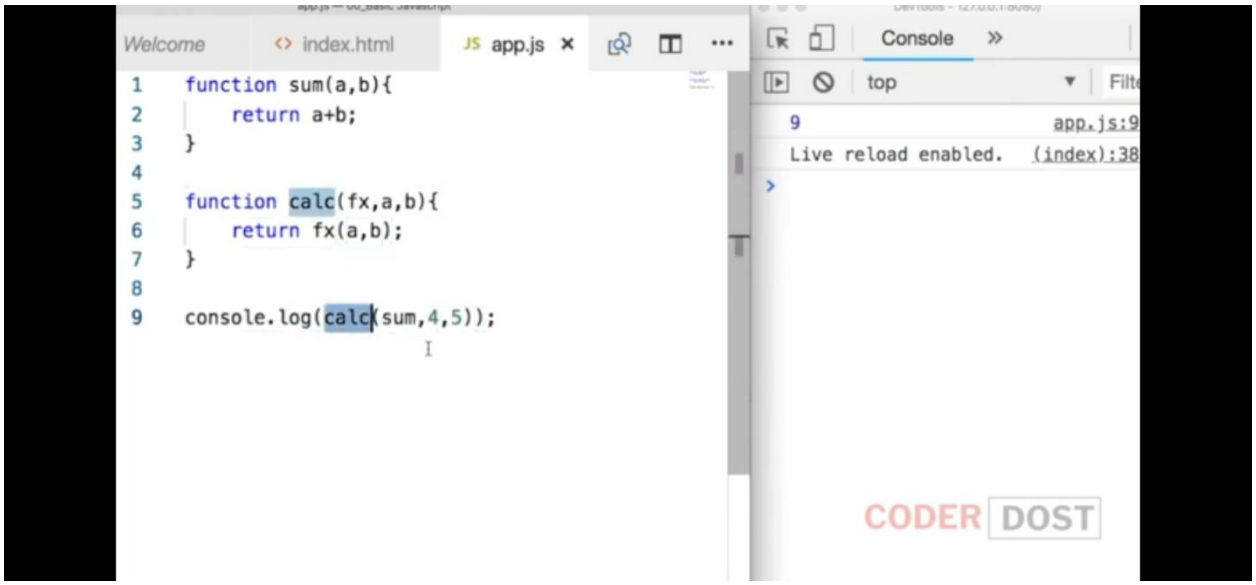


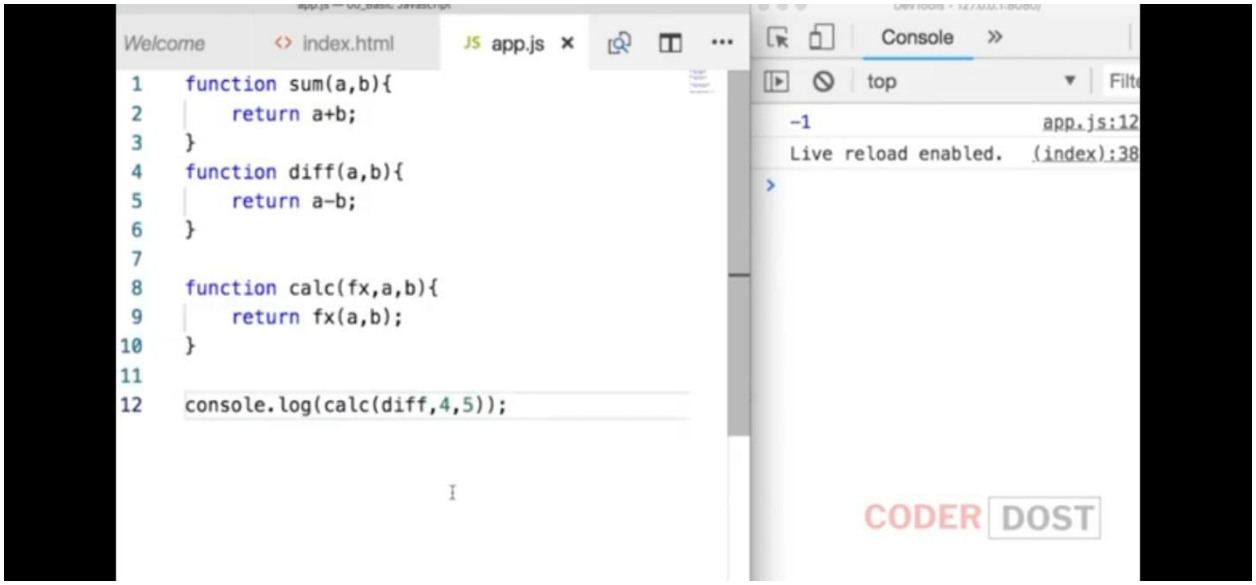
Callback Functions

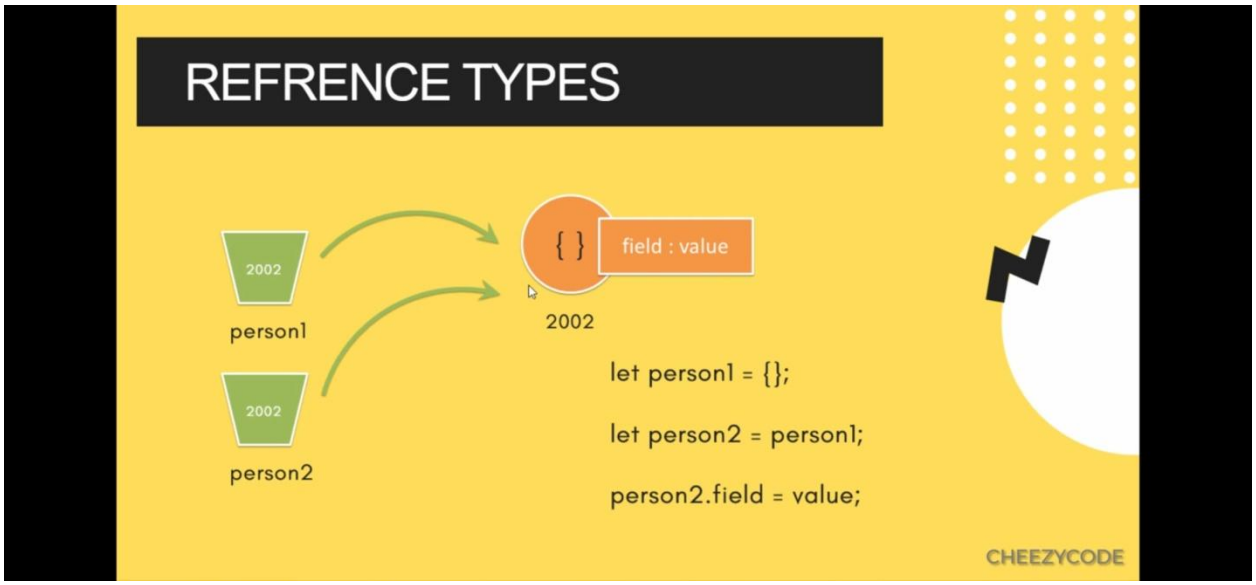
```
var calc = function(fx,a,b){
      return fx(a,b);
}

var sum = function(x,y){
    return a+b;
}
```

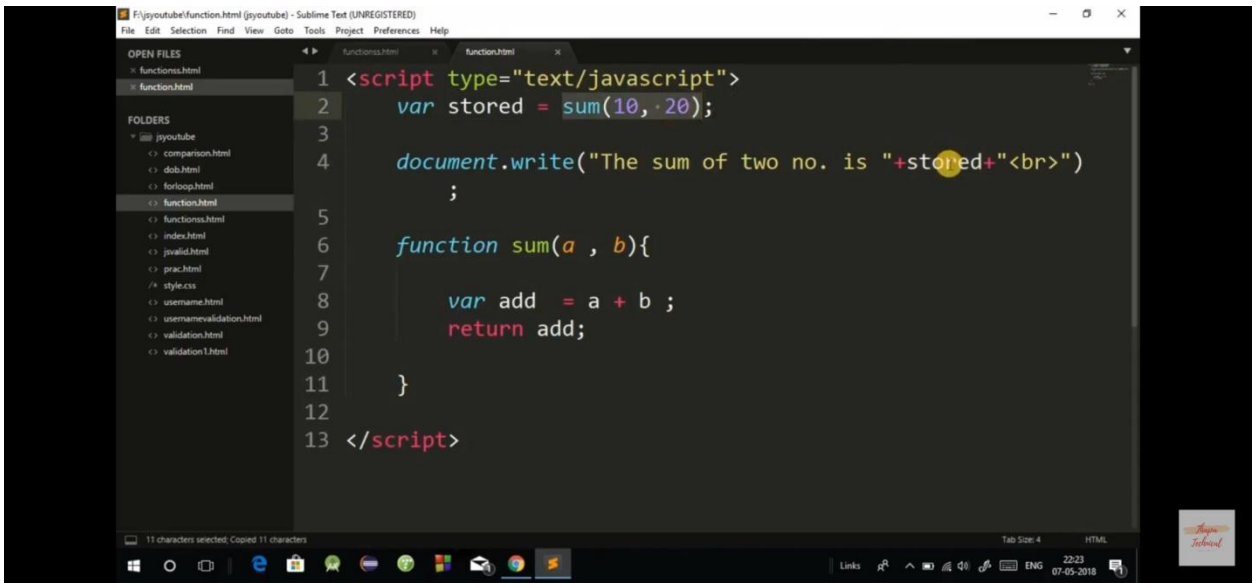
calc(sum,4,5)

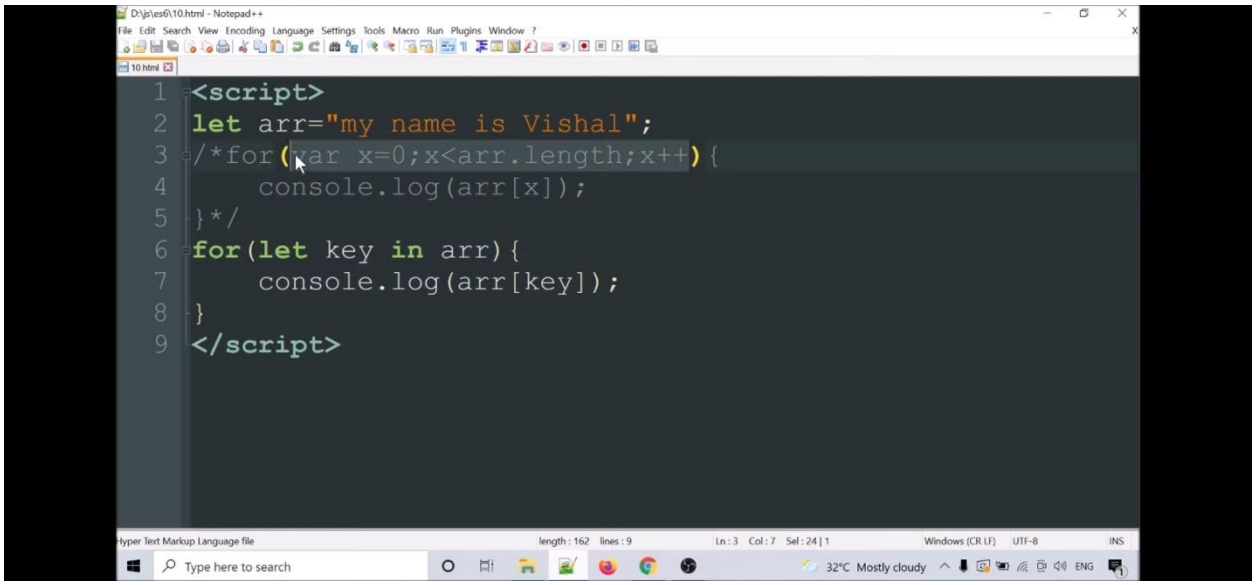
CODER DOST

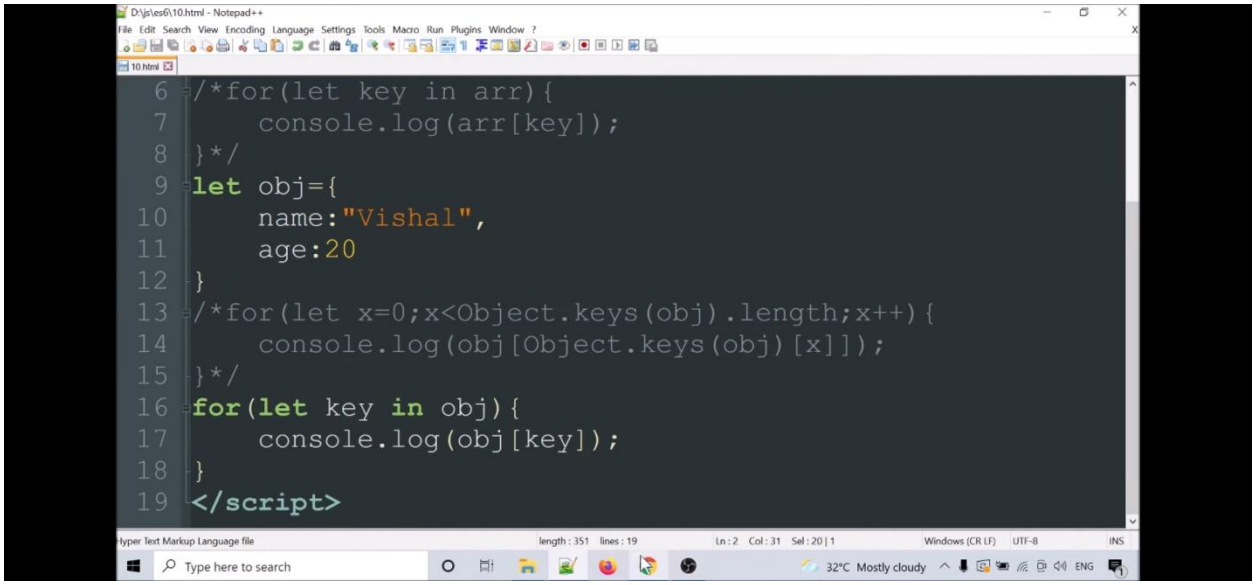

Welcome
index.html
app.js
function sum(a,b){
return a+b;
}
function calc(fx,a,b){
return fx(a,b);
}
console.log(calc(sum,4,5));
Console
top
9
app.js:9
Live reload enabled.
(index):38
CODER DOST


Welcome
index.html
app.js
function sum(a,b){
return a+b;
}
function diff(a,b){
return a-b;
}
function calc(fx,a,b){
return fx(a,b);
}
console.log(calc(diff,4,5));
Console
top
-1
app.js:12
Live reload enabled.
(index):38
CODER DOST


REFRENCE TYPES
2002
person1
2002
person2
{ }
field : value
2002
let person1 = {};
let person2 = person1;
person2.field = value;
CHEEZYCODE


F:\jsyoutube\function.html (jsyoutube) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
OPEN FILES
functionss.html
function.html
FOLDERS
jsyoutube
comparison.html
dob.html
forloop.html
function.html
functionss.html
index.html
jsvalid.html
prac.html
style.css
username.html
usernamevalidation.html
validation.html
validation1.html
<script type="text/javascript">
var stored = sum(10, 20);
document.write("The sum of two no. is "+stored+"
")
;
function sum(a , b){
var add = a + b ;
return add;
}
</script>
11 characters selected; Copied 11 characters
Tab Size: 4
HTML
ENG
22:23
07-05-2018


D:\js\es6\10.html - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
10.html
<script>
let arr="my name is Vishal";
/*for(var x=0;x<arr.length;x++){
console.log(arr[x]);
}*/
for(let key in arr){
console.log(arr[key]);
}
</script>
Hyper Text Markup Language file
length : 162 lines : 9
Ln : 3 Col : 7 Sel : 24 | 1
Windows (CR LF)
UTF-8
INS
Type here to search
32°C Mostly cloudy
ENG


D:\js\es6\10.html - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
10.html
/*for(let key in arr){
console.log(arr[key]);
}*/
let obj={
name:"Vishal",
age:20
}
/*for(let x=0;x<Object.keys(obj).length;x++){
console.log(obj[Object.keys(obj)[x]]);
}*/
for(let key in obj){
console.log(obj[key]);
}
</script>
Hyper Text Markup Language file
length : 351 lines : 19
Ln : 2 Col : 31 Sel : 20 | 1
Windows (CR LF)
UTF-8
INS
Type here to search
32°C Mostly cloudy
ENG

1. Encapsulation: -

Encapsulation means wrapping up data and member function (Method) together into a single unit i.e. class.

2. Abstraction: -

Abstraction is the process of showing only essential/necessary features of an entity/object to the outside world and hide the other irrelevant information. For example to open your TV we only have a power button, It is not required to understand how infra-red waves are getting generated in TV remote control.

3. Inheritance: -

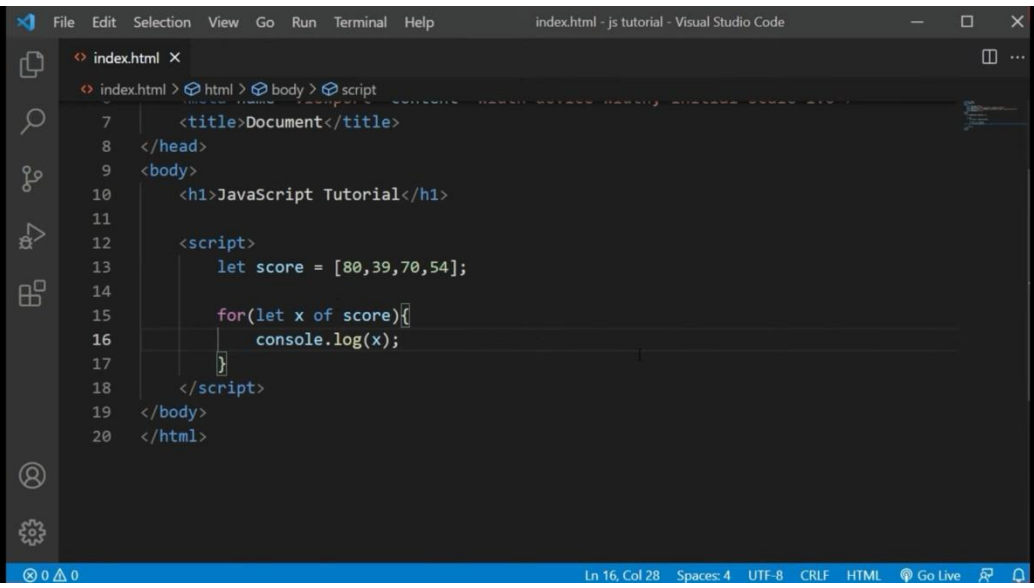
Inheritance allows a class (subclass) to acquire the properties and behavior of another class (super-class). It helps to reuse, customize and enhance the existing code. So it helps to write a code accurately and reduce the development time.

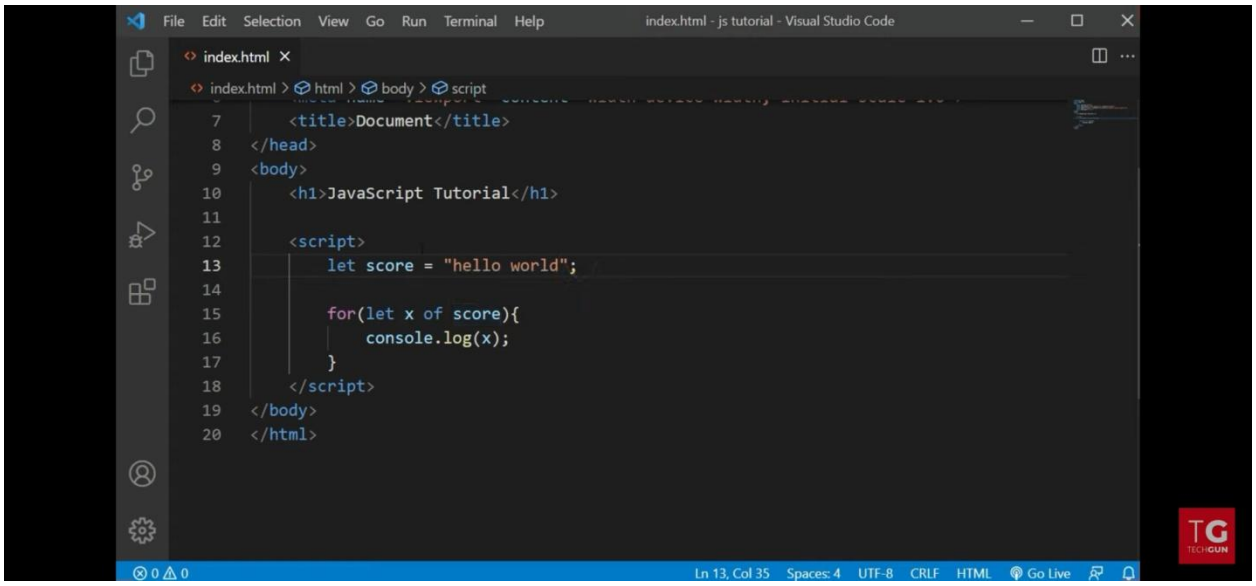
4. Polymorphism: -

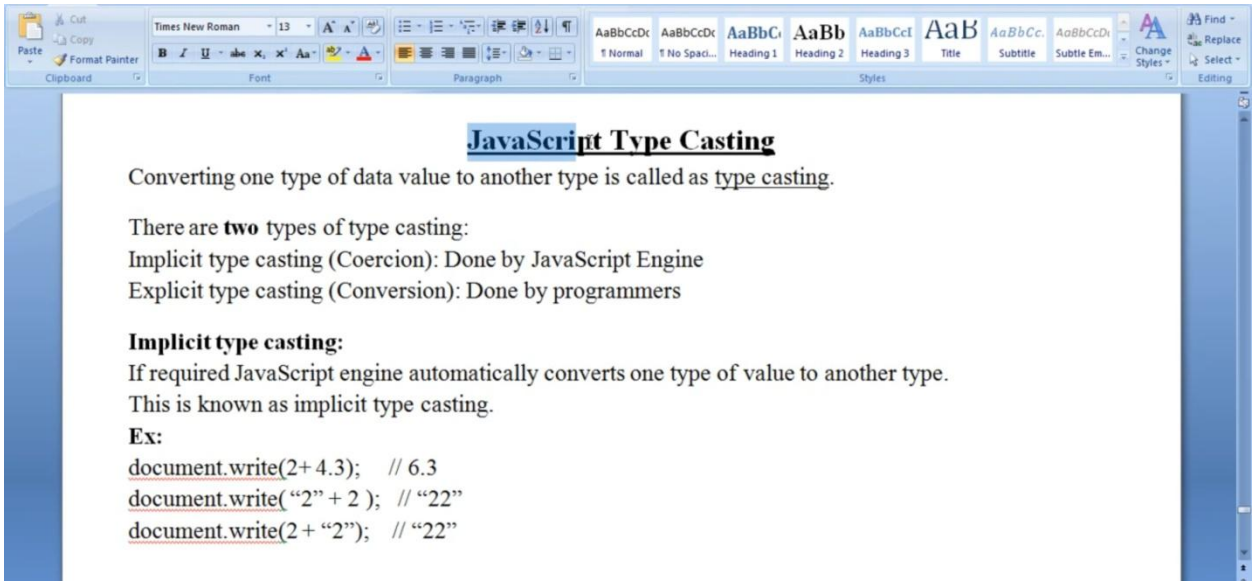
So polymorphism means "many forms". A subclass can define its own unique behavior and still share the same functionalities or behavior of its parent/base class.

```
class square(){
 area()
}

Var S1 =  new square();
S1->area();
```

```
class circle(){
 area()
}

Var C1 =  new circle();
C1->area();
```







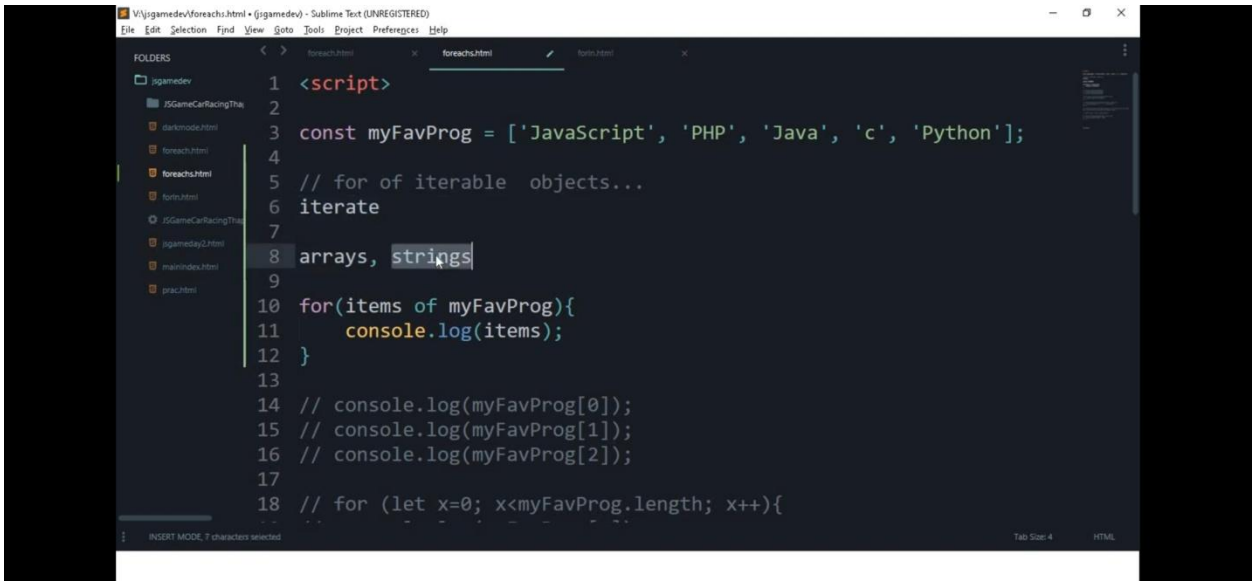
JavaScript Type Casting

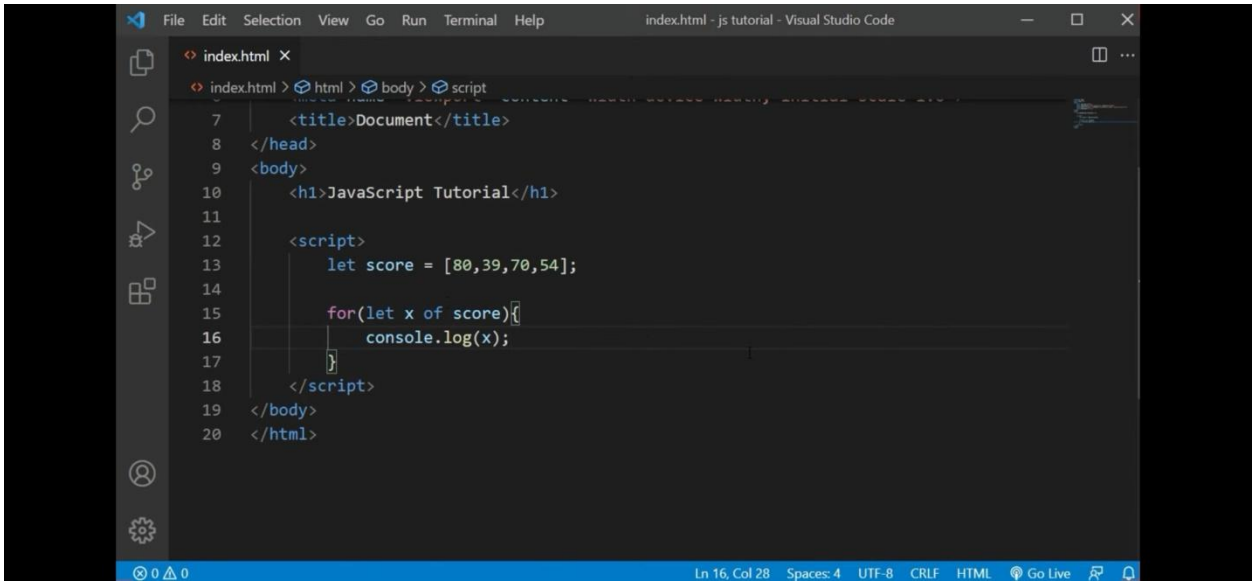
Converting one type of data value to another type is called as type casting.

There are **two** types of type casting:
Implicit type casting (Coercion): Done by JavaScript Engine
Explicit type casting (Conversion): Done by programmers

Implicit type casting:
If required JavaScript engine automatically converts one type of value to another type.
This is known as implicit type casting.
Ex:

```
document.write(2+4.3);     // 6.3
document.write( "2" + 2 );   // "22"
document.write(2 + "2");    // "22"
```


V:\jsgamedev\foreachs.html • (jsgamedev) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
jsgamedev
foreach.html
foreachs.html
forin.html
<script>
const myFavProg = ['JavaScript', 'PHP', 'Java', 'c', 'Python'];
// for of iterable objects...
iterate
arrays, strings
for(items of myFavProg){
console.log(items);
}
// console.log(myFavProg[0]);
// console.log(myFavProg[1]);
// console.log(myFavProg[2]);
// for (let x=0; x<myFavProg.length; x++){
INSERT MODE, 7 characters selected
Tab Size: 4
HTML

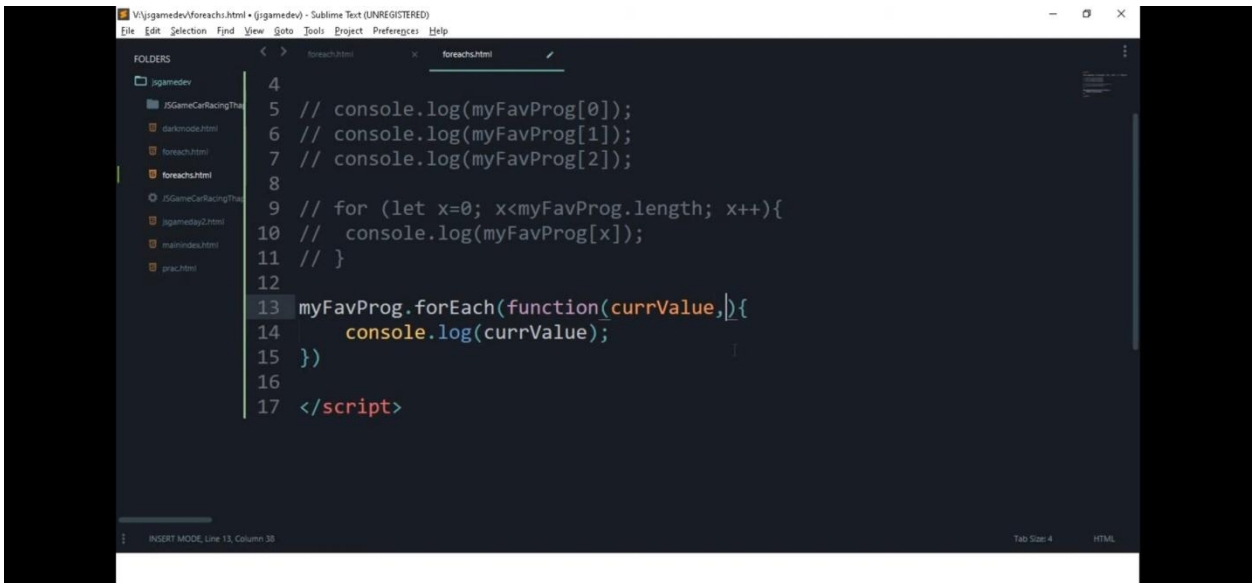

File Edit Selection View Go Run Terminal Help
index.html - js tutorial - Visual Studio Code
index.html
index.html > html > body > script
<title>Document</title>
</head>
<body>
<h1>JavaScript Tutorial</h1>
<script>
let score = [80,39,70,54];
for(let x of score){
console.log(x);
}
</script>
</body>
</html>
Ln 16, Col 28 Spaces: 4 UTF-8 CRLF HTML Go Live

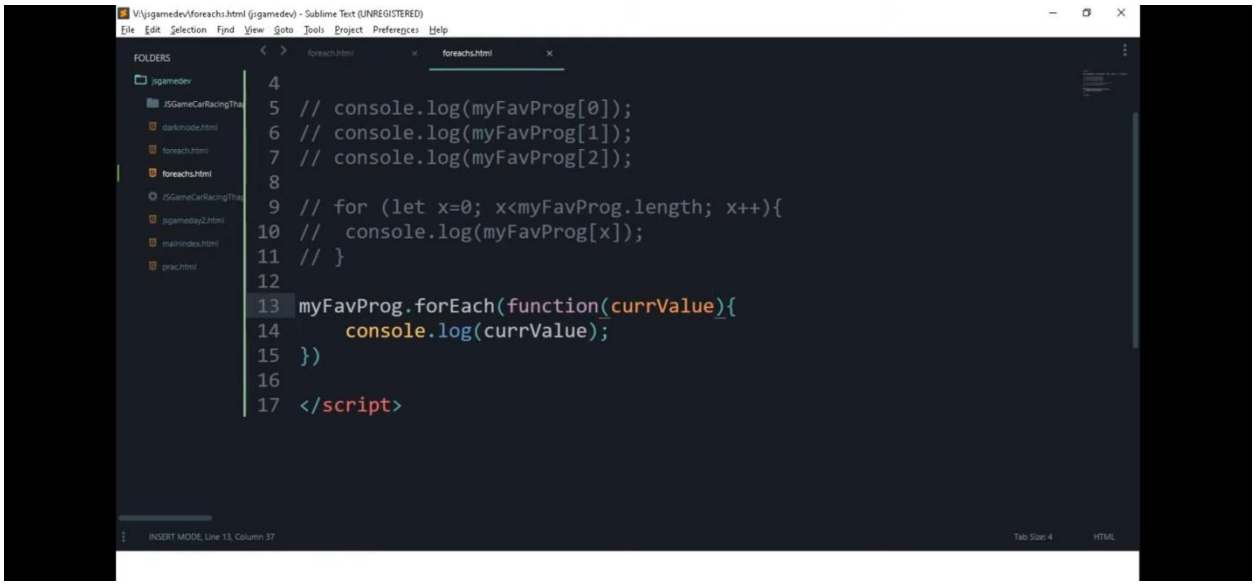
The forEach() method calls a function once for each element in an array, in order.

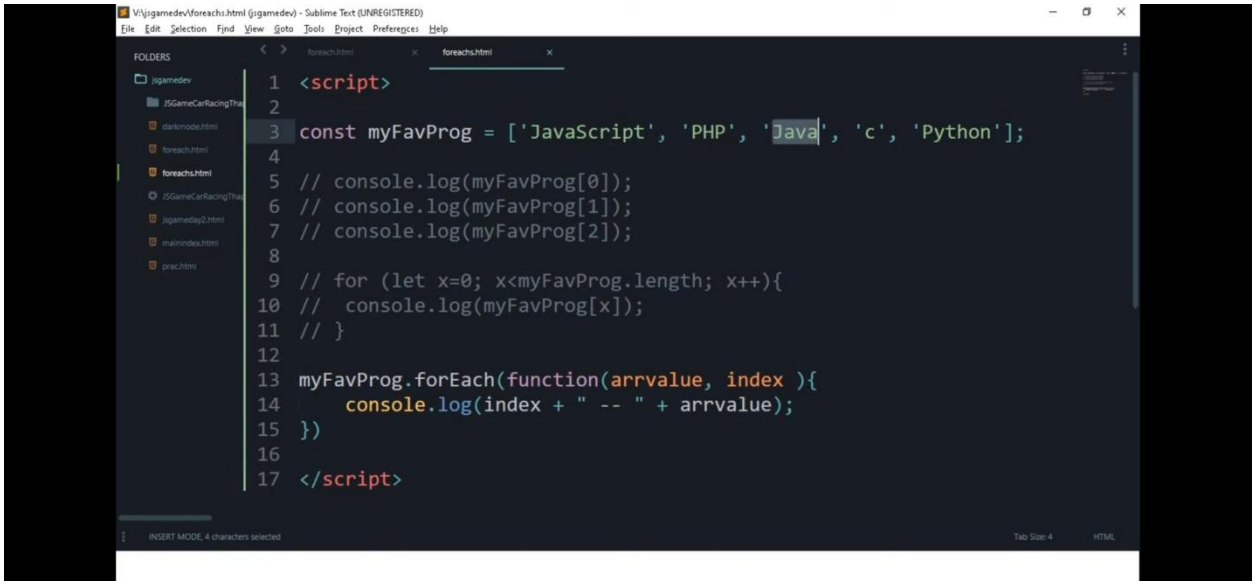
forEach() Method

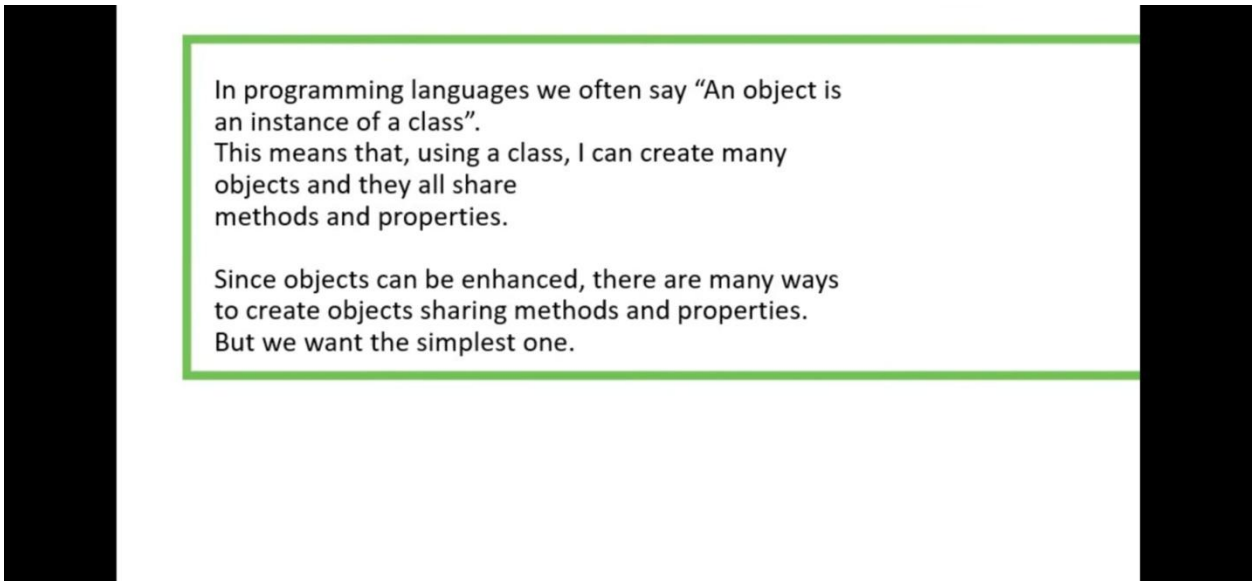
Syntax: arr.forEach(callback(currentValue [, index [, array]])[, thisArg])

Argument	Description
currentValue	Required. The value of the current element
index	Optional. The array index of the current element
arr	Optional. The array object the current element belongs to




V:\jsgamedev\foreachs.html (jsgamedev) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
jsgamedev
foreach.html
foreachs.html
// console.log(myFavProg[0]);
// console.log(myFavProg[1]);
// console.log(myFavProg[2]);
// for (let x=0; x<myFavProg.length; x++){
// console.log(myFavProg[x]);
// }
myFavProg.forEach(function(currValue){
console.log(currValue);
})
</script>
INSERT MODE, Line 13, Column 37
Tab Size: 4
HTML


V:\jsgamedev\foreachs.html (jsgamedev) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
jsgamedev
foreach.html
foreachs.html
<script>
const myFavProg = ['JavaScript', 'PHP', 'Java', 'c', 'Python'];
// console.log(myFavProg[0]);
// console.log(myFavProg[1]);
// console.log(myFavProg[2]);
// for (let x=0; x<myFavProg.length; x++){
// console.log(myFavProg[x]);
// }
myFavProg.forEach(function(arrvalue, index){
console.log(index + " -- " + arrvalue);
})
</script>
INSERT MODE, 4 characters selected
Tab Size: 4
HTML



Bind `this`

For methods in React, the `this` keyword should represent the component that owns the method.

That is why you should use arrow functions. With arrow functions, `this` will always represent the object that defined the arrow function.

Why Arrow Functions?

In class components, the `this` keyword is not defined by default, so with regular functions the `this` keyword represents the object that called the method, which can be the global window object, a HTML button, or whatever.

Read more about binding `this` in our React ES6 'What About this?' chapter.

If you *must* use regular functions instead of arrow functions you have to bind `this` to the component instance using the `bind()` method:

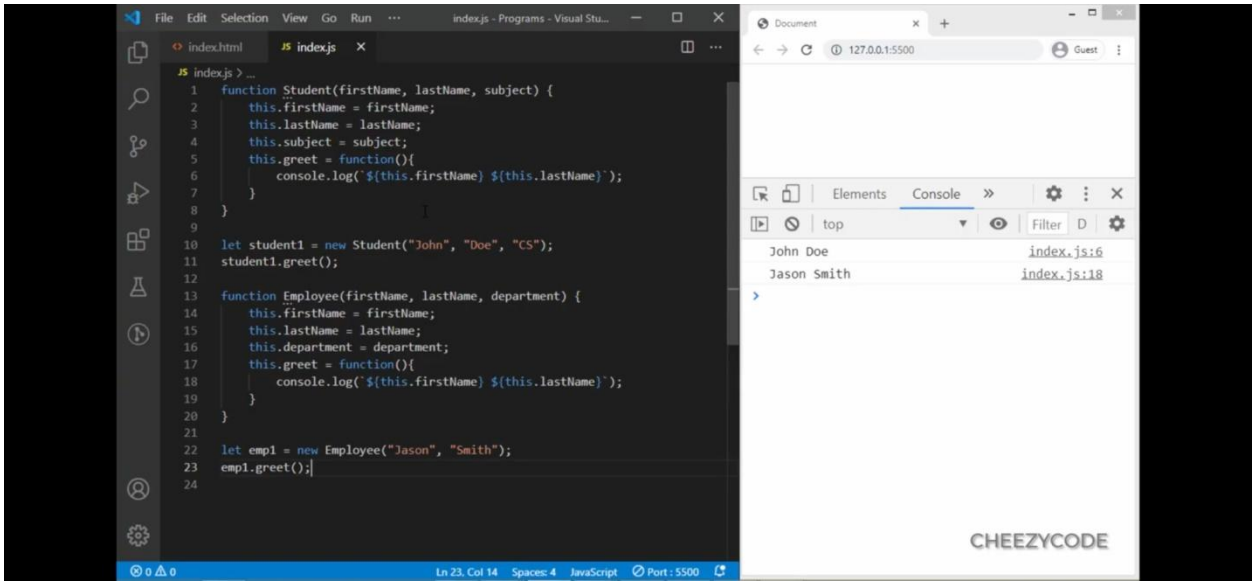
bind() Method

by this method, we can bind an object to a common function, so that the function gives different result when its need.

```
<script>
        function Employee(firstName,lastName)
        {
          this.firstName=firstName;
          this.lastName=lastName;
        }

        Employee.prototype.company="Javatpoint"

        var employee1=new Employee("Martin","Roy");
        var employee2=new Employee("Duke", "William");
        document.writeln(employee1.firstName+" "+employee1.lastName+" "+employee1.company+"<b
        r>");
        document.writeln(employee2.firstName+" "+employee2.lastName+" "+employee2.company);
</script>
```

Let's see an example to add a new property to the constructor function.

```
<script>
        function Employee(firstName,lastName)
        {
         this.firstName=firstName;
         this.lastName=lastName;
        }

        Employee.prototype.company="Javatpoint"

        var employee1=new Employee("Martin","Roy");
        var employee2=new Employee("Duke", "William");
        document.writeln(employee1.firstName+" "+employee1.lastName+" "+employee1.company+"<b
        r>");
        document.writeln(employee2.firstName+" "+employee2.lastName+" "+employee2.company);
</script>
```



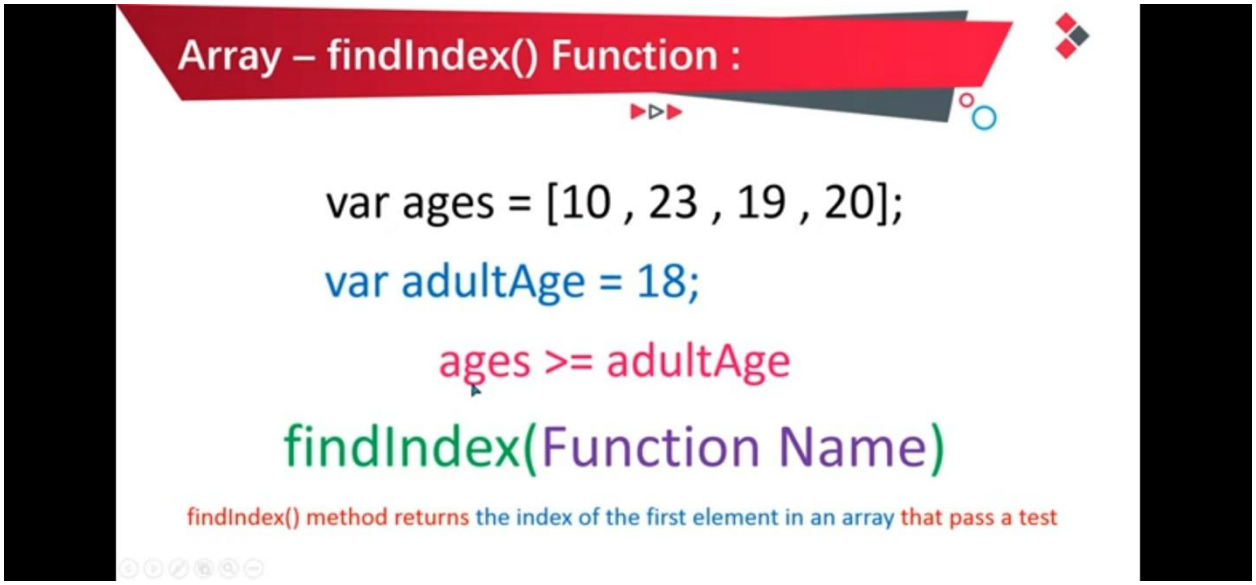

const person = {
firstName:"John",
lastName: "Doe",
fullName: function () {
return this.firstName + " " + this.lastName;
}
}
console.log(person.fullName()); // John Doe
JS
SUBSCRIBE


const person = {
fullName: function() {
return this.firstName + " " + this.lastName;
}
}
const person1 = {
firstName:"John",
lastName: "Doe"
}
const person2 = {
firstName:"Mary",
lastName: "Doe"
}
console.log(person.fullName.call(person1)); // John Doe
console.log(person.fullName.call(person2)); // Mary Doe
JS
SUBSCRIBE


const obj = { name: "John" };
let greeting = function(a,b){
return `${a} ${this.name}. ${b}`;
};
console.log(greeting.call(obj, "Hello", "How are you?"));
// returns: Hello John. How are you?
JS
SUBSCRIBE


const obj = { name: "John" };
let greeting = function(a,b){
return `${a} ${this.name}. ${b}`;
};
console.log(greeting.apply(obj, ["Hello", "How are you?"]));
// returns: Hello John. How are you?
JS
SUBSCRIBE

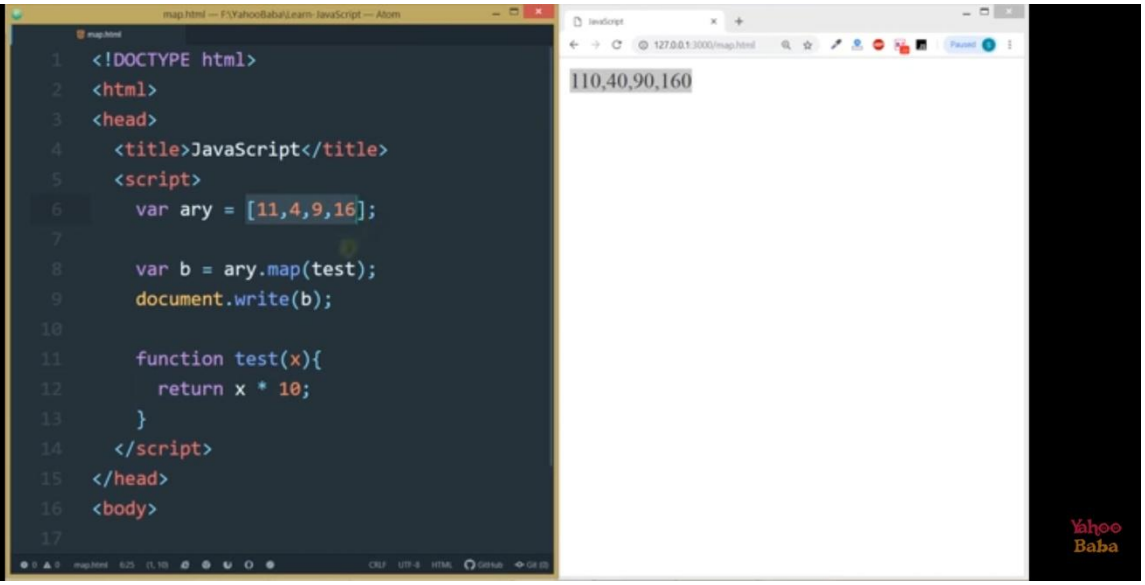

const obj = { name: "John" };
let greeting = function(a,b){
return `${a} ${this.name}. ${b}`;
}
let bound = greeting.bind(obj);
console.log(bound("Hello", "How are you?"));
// returns: Hello John. How are you?


Array – findIndex() Function :
var ages = [10 , 23 , 19 , 20];
var adultAge = 18;
ages >= adultAge
findIndex(Function Name)
findIndex() method returns the index of the first element in an array that pass a test

Array – find() Function :

```
var ages = [10 , 23 , 19 , 20];
var adultAge = 18;
ages >= adultAge
find(Function Name)
```

find() method returns the value of the first element in an array that pass a test



Access Modifiers

	Class Itself	Outside Class	Derived Class
Public	✓	✓	✓
Protected	✓	✗	✓
Private	✓	✗	✗

Namespace

First.php

```
namespace first;
class test{

}
```

Second.php

```
namespace second;
class test{

}
```

Third.php

```
require First.php;
require second.php;

$obj = new first\test();
```

Web notes

Class is a blue print of object.

memory is allocated for an object by Constructor.

No need of define variable in constructor.

Constructor function is a same as class.

Constructor is a method wchic is use to initialize object state.

this keyword ka use kewal object ya constructor function ke under hota hai

History of JavaScript

- JavaScript is a scripting language that enables you to create dynamically updating content, control multimedia, animate images etc.
- Written by Brendan Eich in 1995.
- ECMAScript or ES released in 1997
- ES2 & ES3 released in 1998, 1999 resp.
- ES5 was released in 2009
- ES6 (or ES2015) was released in 2015
- Anything greater than ES5 is called ESNext or ES2015+

Prerequisites

- JavaScript
- ES6
- JSX
- Babel
- Webpack
- Node
- PHP

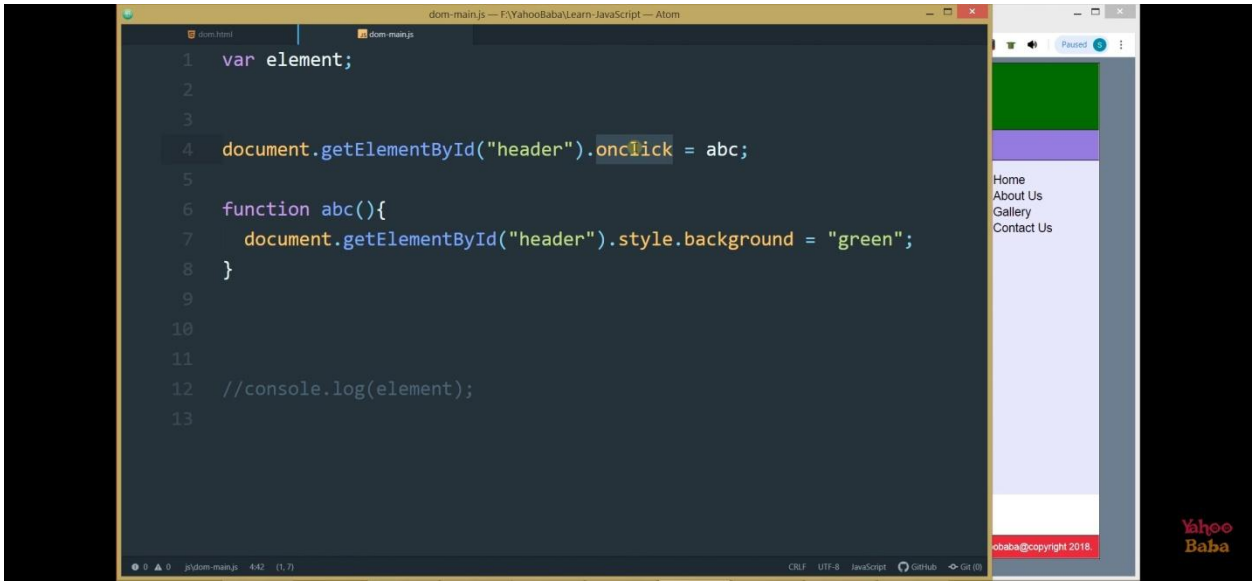
Why do we use ESNext, JSX, Babel & Webpack?

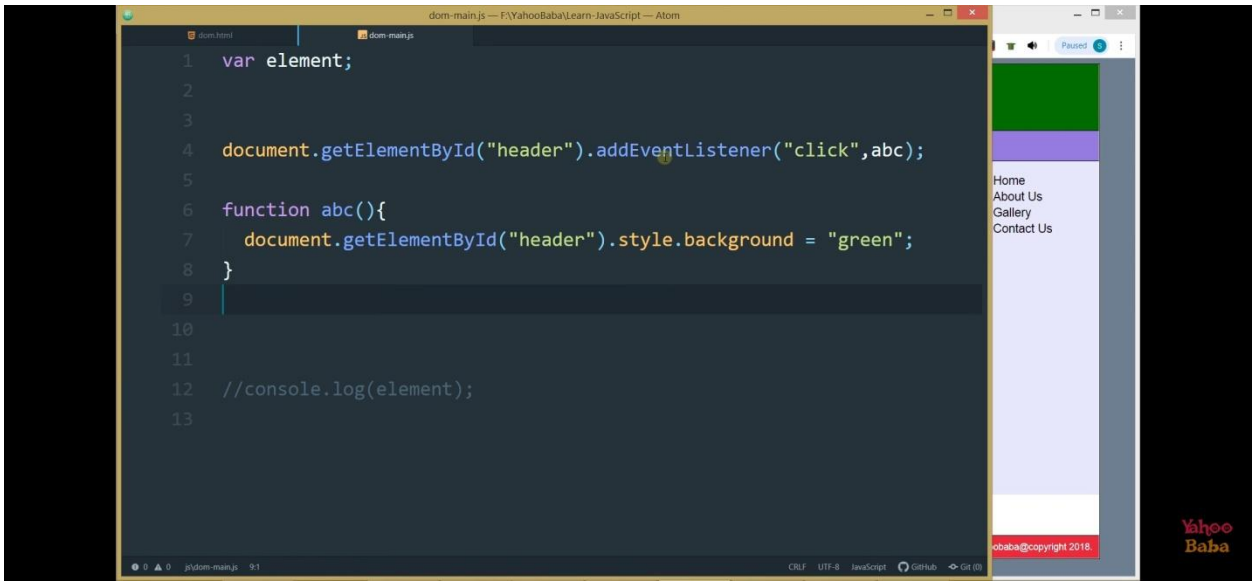
- ESNext makes for simpler code that is easier to read and write.
- Convert ESNext & JSX syntax to code browser can understand
- Using transformation tools help in optimizing our code for widest variety of browsers.
- Organize your code into smaller modules and then bundle them together using Webpack into a single download

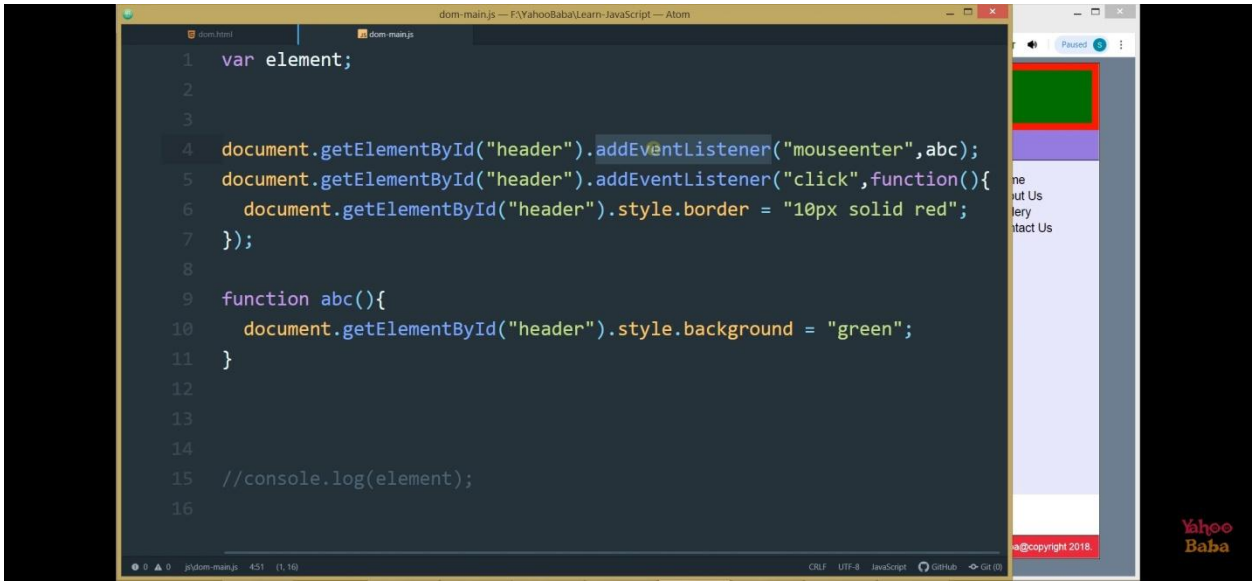
Why do we use ESNext, JSX, Babel & Webpack?

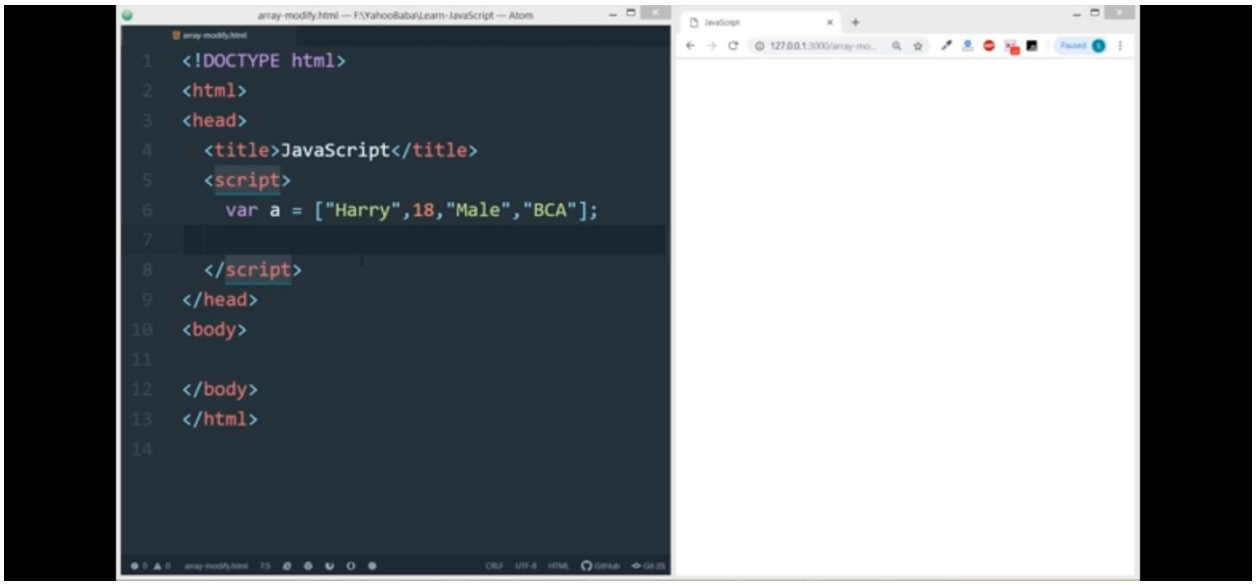
- Both webpack and Babel are tools written in JavaScript and run using Node.js (node)
- Node.js is a runtime environment for JavaScript outside of a browser
- node allows you to run JavaScript code on the command-line..


Cyber Technophile
What are COOKIES ?
Cookies are small files, that a website stores in your browser which contain information pertaining to that particular website.
Each time you return, the data, or 'cookie' is sent back to the site.
After certain interval of time it gets deleted automatically.
Cookies may contain:
Username and password
links that you have clicked on
Visitor tracking information
Your cart information for buying online
Your computer's general location in the world
Videos you have watched
One Fine Monday Morning
HANG ONTO THIS, OK?
NO PROBLEM!
msn.com
The Following Wednesday Afternoon
MSN! HERE'S THAT COOKIE FROM MONDAY.
THANKS DUDE!
msn.com


dom-main.js — F:\YahooBaba\Learn-JavaScript — Atom
dom.html
dom-main.js
var element;
document.getElementById("header").onclick = abc;
function abc(){
document.getElementById("header").style.background = "green";
}
//console.log(element);
Home
About Us
Gallery
Contact Us
obaba@copyright 2018.
Yahoo Baba


dom-main.js — F:\YahooBaba\Learn-JavaScript — Atom
dom.html
dom-main.js
var element;
document.getElementById("header").addEventListener("click",abc);
function abc(){
document.getElementById("header").style.background = "green";
}
//console.log(element);
Home
About Us
Gallery
Contact Us
Yahoo Baba


dom-main.js — F:\YahooBaba\Learn-JavaScript — Atom
dom.html
dom-main.js
var element;
document.getElementById("header").addEventListener("mouseenter",abc);
document.getElementById("header").addEventListener("click",function(){
document.getElementById("header").style.border = "10px solid red";
});
function abc(){
document.getElementById("header").style.background = "green";
}
//console.log(element);
Yahoo Baba



Modify Array Element

```
var a = [“Harry” , 18 , ”BCA”];
var a[1] = 19;
```

Remove Array Element

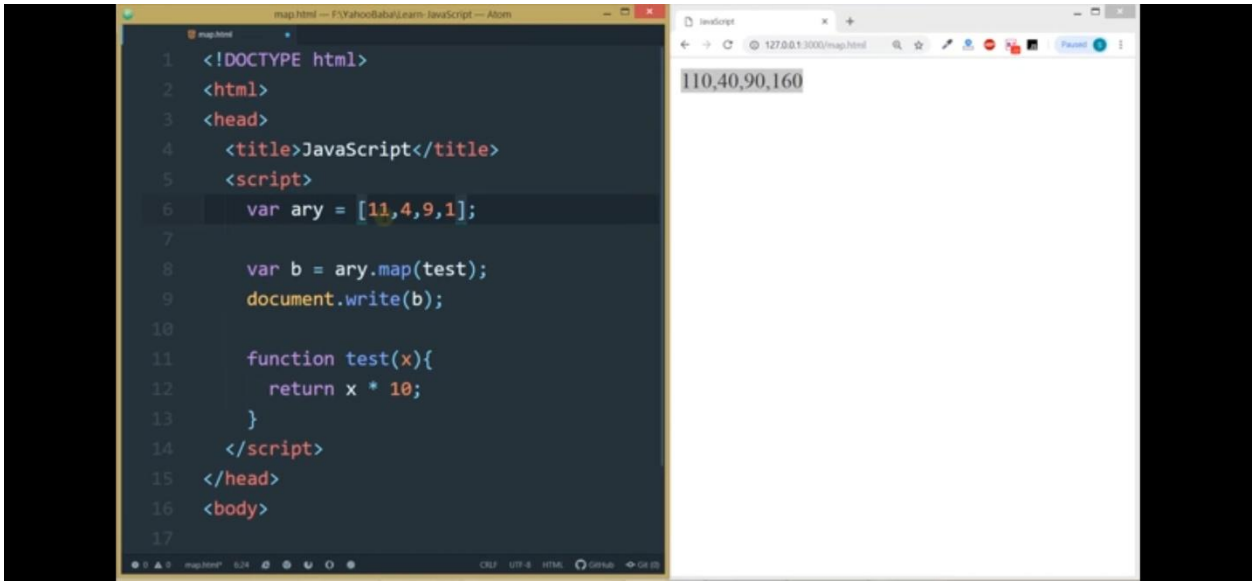
```
var a = [“Harry” , 18 , ”BCA”];
delete a[1];
```

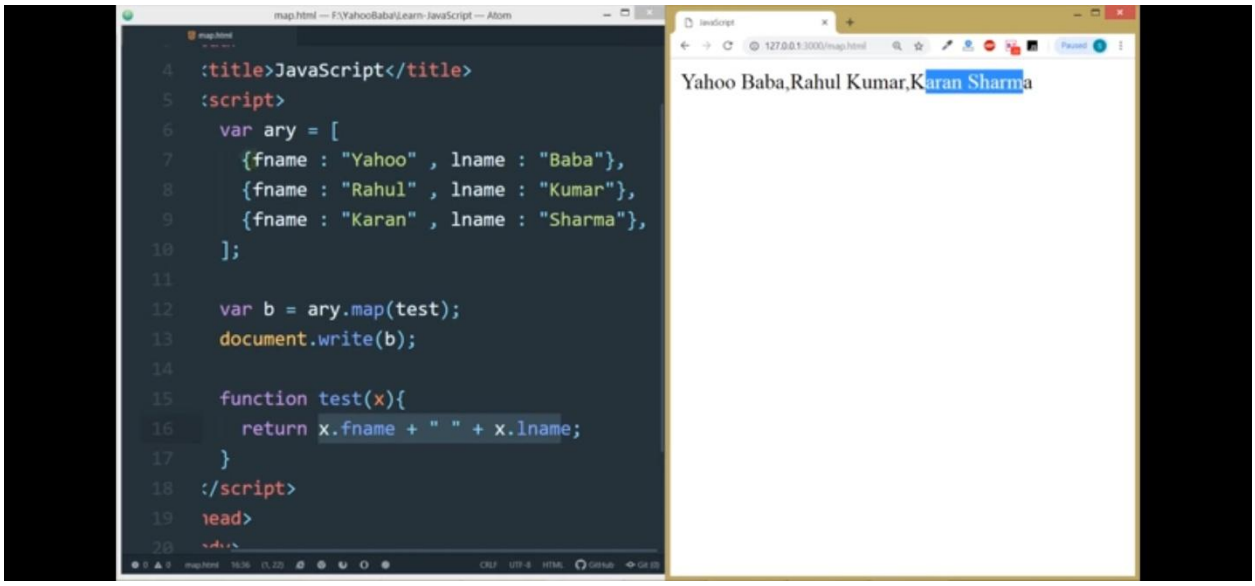
Array – Map() Function :

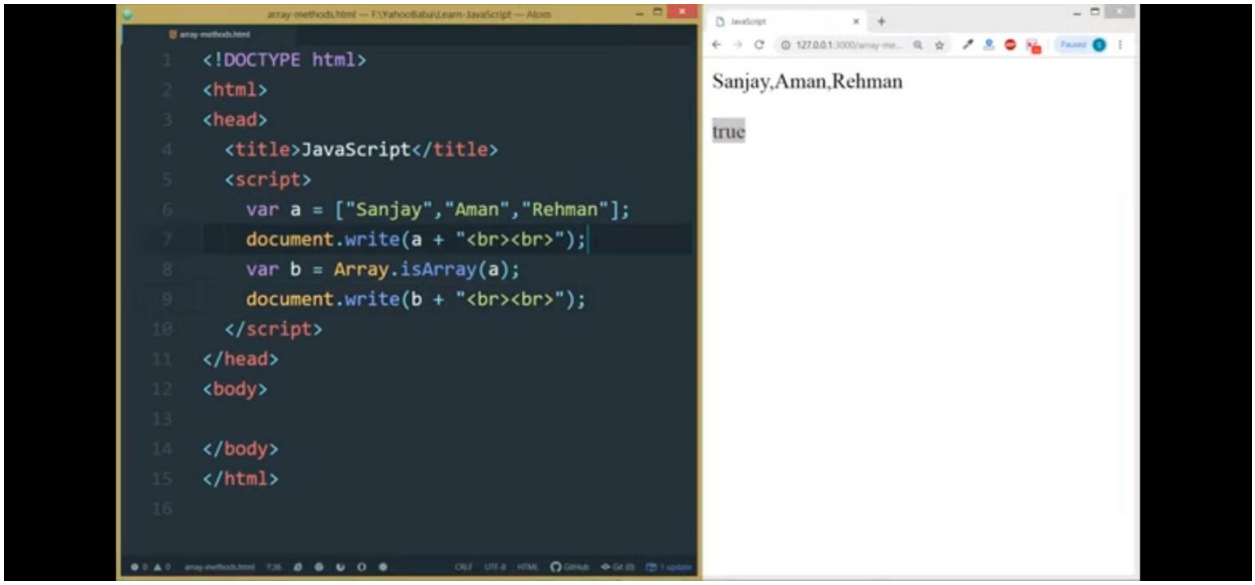
```
var a = [1 , 3 , 5 , 8 , 10];
a.map(function(){
  Statement
});
```

Array – Map() Function :

```
var a = [1 , 3 , 5 , 8 , 10];

a.map(function(){
  Statement
});
```

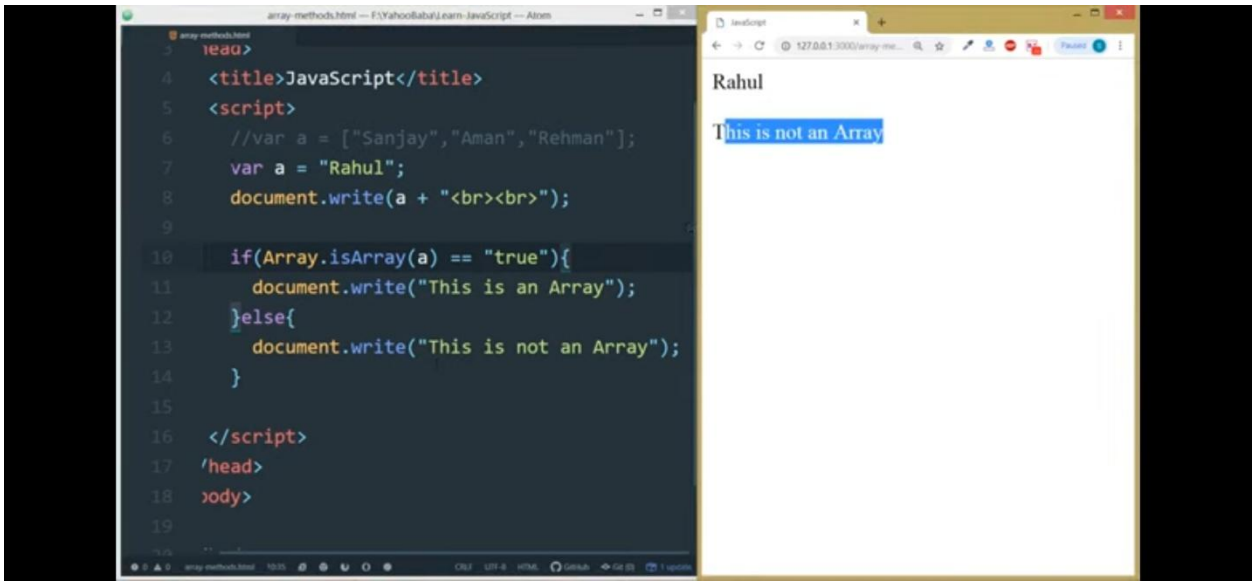



map.html — F:\YahooBaba\Learn-JavaScript — Atom
:title>JavaScript</title>
:script>
var ary = [
{fname : "Yahoo" , lname : "Baba"},
{fname : "Rahul" , lname : "Kumar"},
{fname : "Karan" , lname : "Sharma"},
];
var b = ary.map(test);
document.write(b);
function test(x){
return x.fname + " " + x.lname;
}
:/script>
ıead>
JavaScript
127.0.0.1:3000/map.html
Yahoo Baba,Rahul Kumar,Karan Sharma

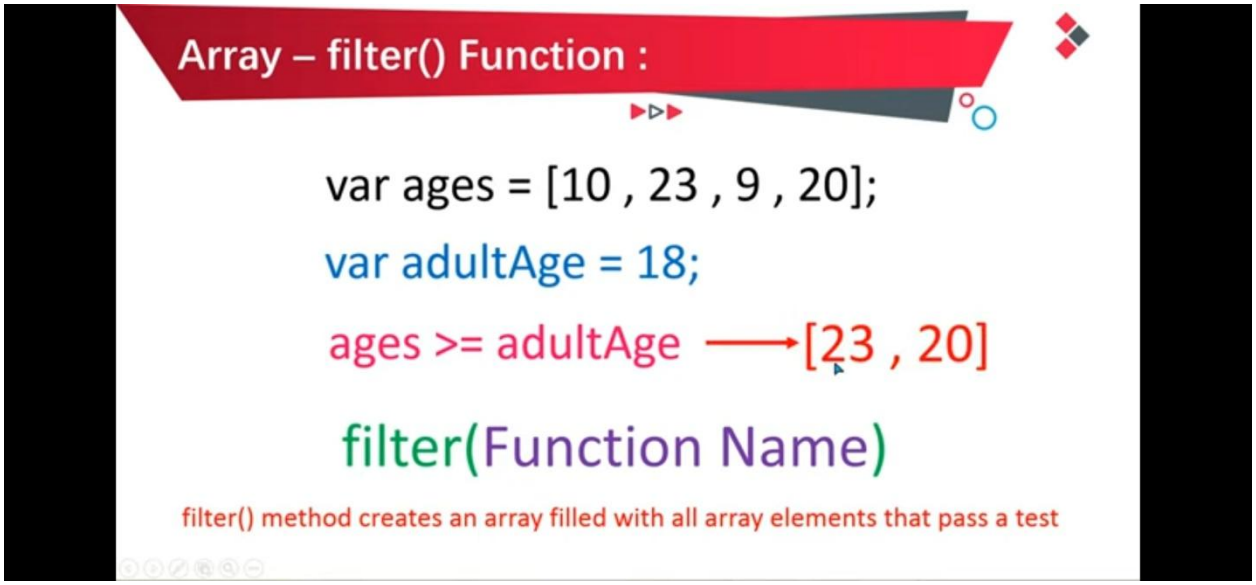

array-methods.html — F:\YahooBaba\Learn-JavaScript — Atom
<!DOCTYPE html>
<html>
<head>
<title>JavaScript</title>
<script>
var a = ["Sanjay","Aman","Rehman"];
document.write(a + "

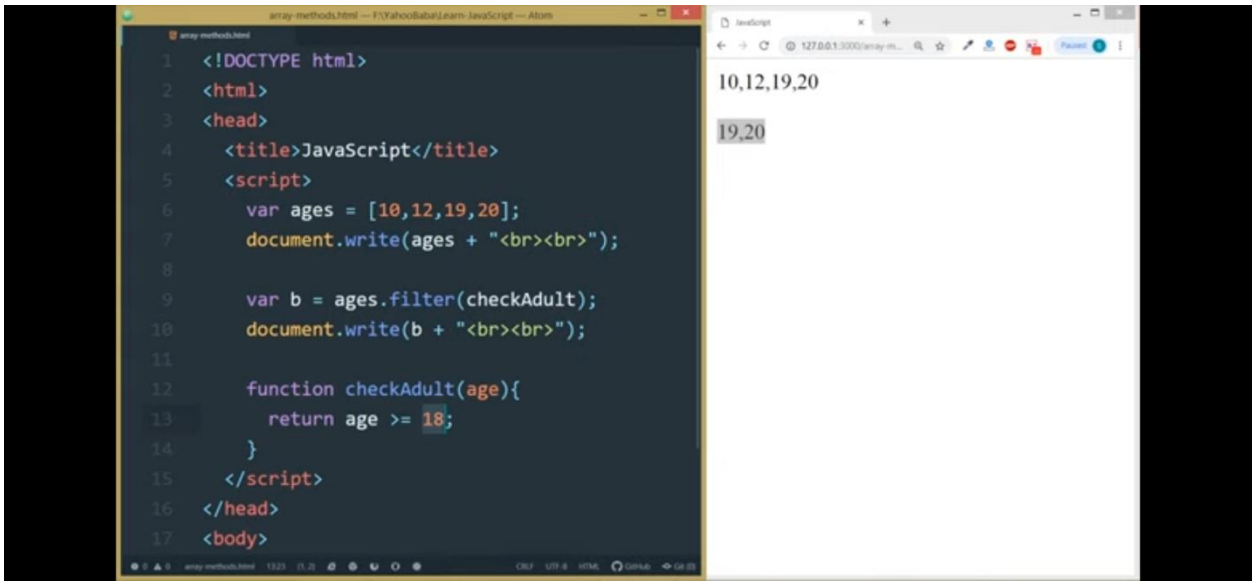
");
var b = Array.isArray(a);
document.write(b + "

");
</script>
</head>
<body>
</body>
</html>
JavaScript
Sanjay,Aman,Rehman
true


array-methods.html — F:\YahooBaba\Learn-JavaScript — Atom
<title>JavaScript</title>
<script>
//var a = ["Sanjay","Aman","Rehman"];
var a = "Rahul";
document.write(a + "

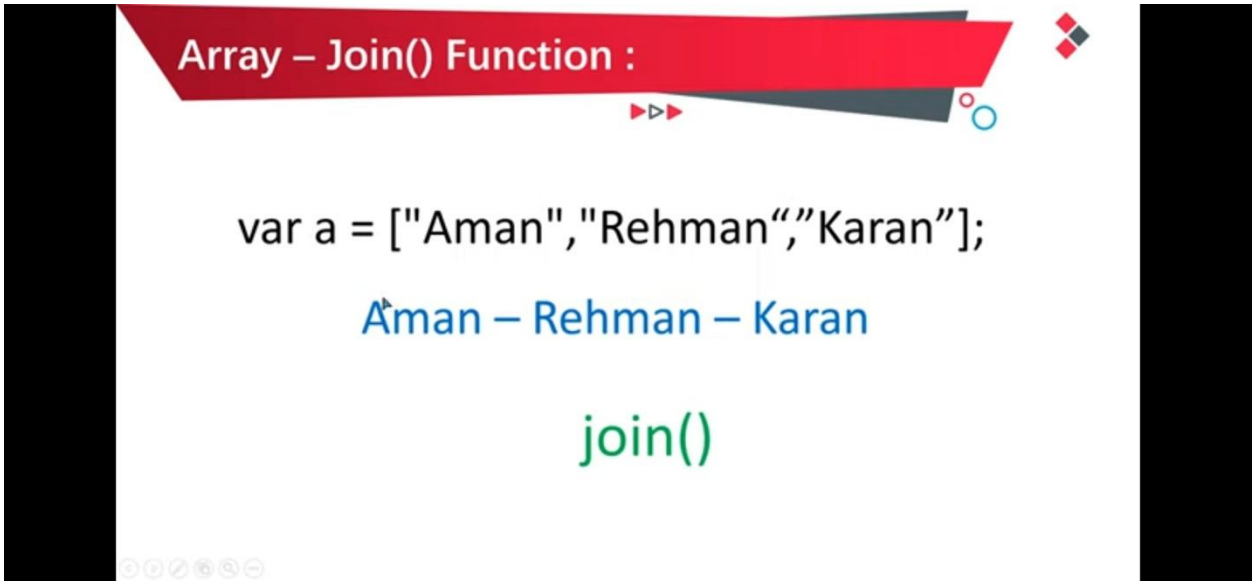
");
if(Array.isArray(a) == "true"){
document.write("This is an Array");
}else{
document.write("This is not an Array");
}
</script>
Rahul
This is not an Array

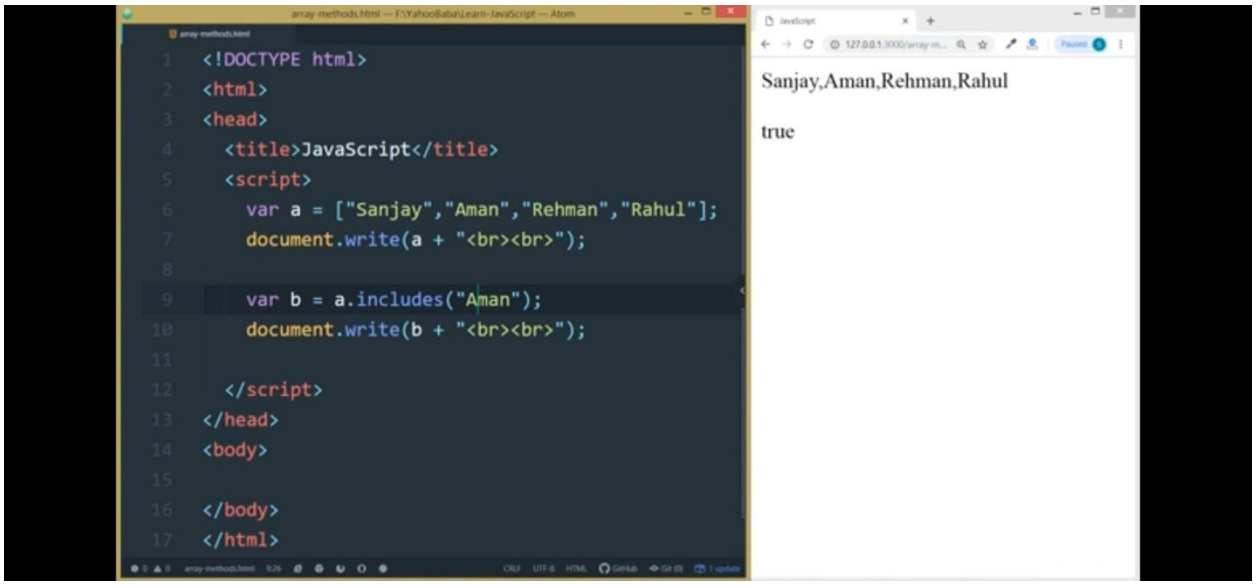

Array – filter() Function :
var ages = [10 , 23 , 9 , 20];
var adultAge = 18;
ages >= adultAge ⟶ [23 , 20]
filter(Function Name)
filter() method creates an array filled with all array elements that pass a test


array-methods.html
<!DOCTYPE html>
<html>
<head>
<title>JavaScript</title>
<script>
var ages = [10,12,19,20];
document.write(ages + "

");
var b = ages.filter(checkAdult);
document.write(b + "

");
function checkAdult(age){
return age >= 18;
}
</script>
</head>
<body>
10,12,19,20
19,20

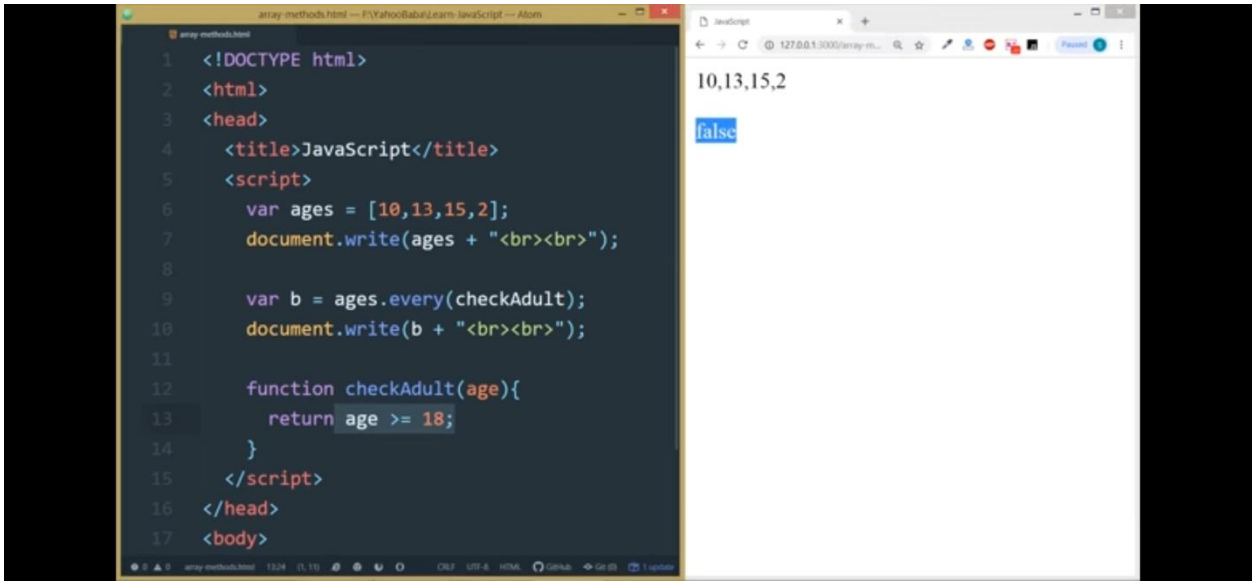

Array – Join() Function :
var a = ["Aman","Rehman","Karan"];
Aman – Rehman – Karan
join()



```
<!DOCTYPE html>
<html>
<head>
  <title>JavaScript</title>
  <script>
    var a = ["Sanjay","Aman","Rehman","Rahul"];
    document.write(a + "<br><br>");

    var b = a.includes("Aman");
    document.write(b + "<br><br>");

  </script>
</head>
<body>

</body>
</html>
```

Sanjay,Aman,Rehman,Rahul

true



```
<!DOCTYPE html>
<html>
<head>
  <title>JavaScript</title>
  <script>
    var ages = [10,13,15,2];
    document.write(ages + "<br><br>");

    var b = ages.every(checkAdult);
    document.write(b + "<br><br>");

    function checkAdult(age){
      return age >= 18;
    }
  </script>
</head>
<body>
```

10,13,15,2

false

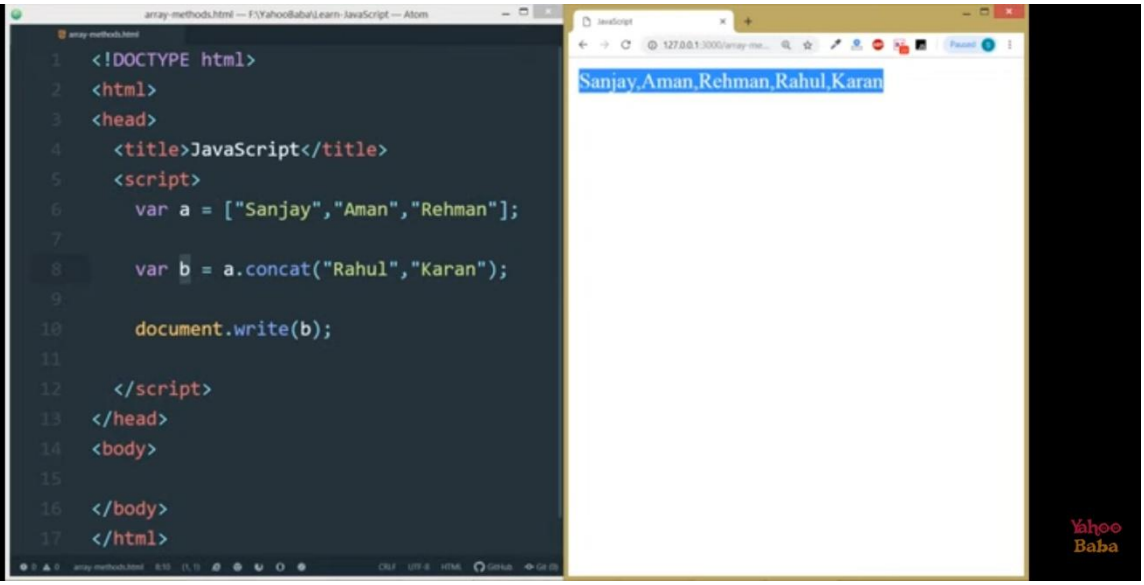
Array – indexOf() Function :

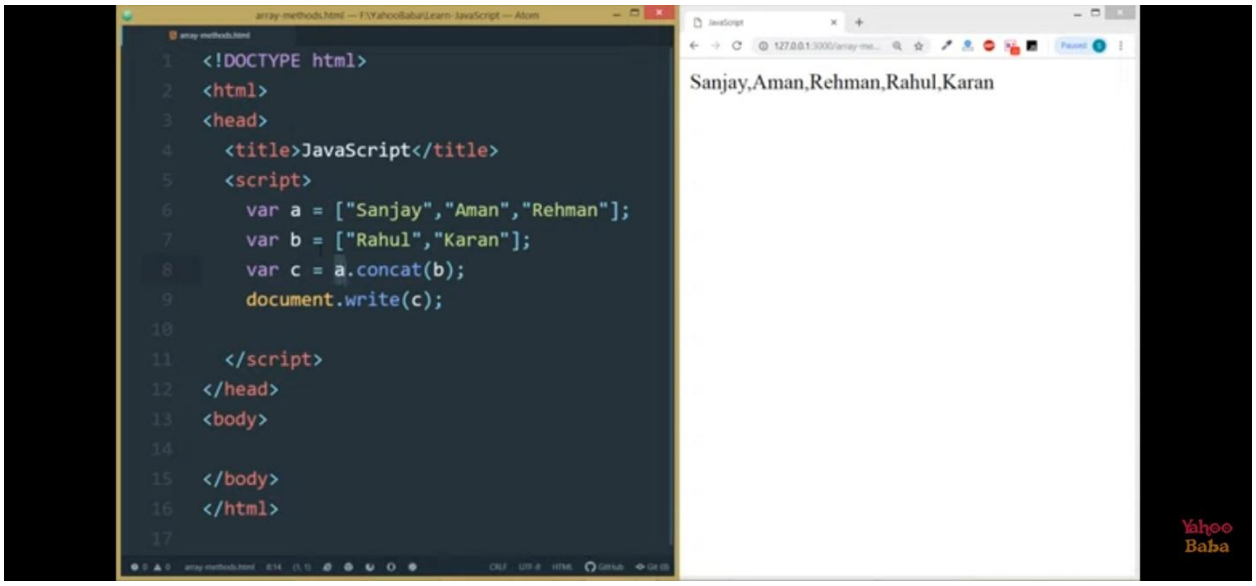
0 1 2 3

var a = ["Sanjay","Aman","Rehman","Aman"];

Aman ⟶ 1

indexOf("Search item" , Start)





Converting an Arrays to Strings

JS

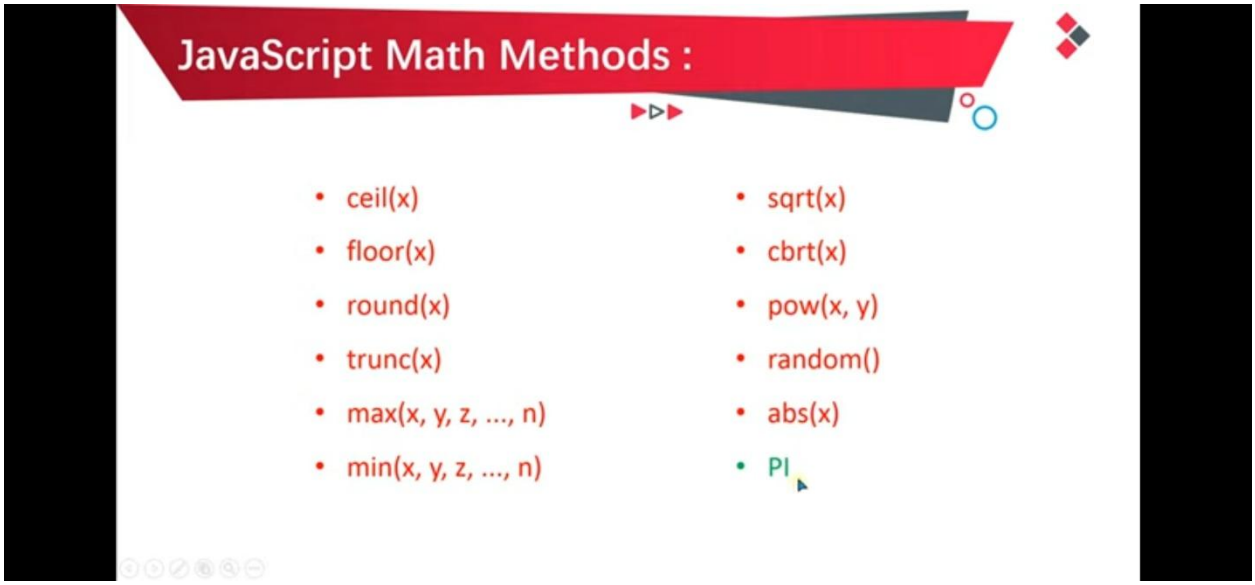
The JavaScript method toString() converts an array to a string of (comma separated) array values.

JavaScript automatically converts an array to a comma separated string when a primitive value is expected.

Activate Windows
Go to Settings to activate Windows.

```
97 <script>
98 // array.reduce(function(total, currentValue,
   currentIndex, arr), initialValue)
99 let array = [200,100,200,500];
100 var sum = 0;
101 array.forEach(function(item){
102  sum+=item;
103 })
104 console.log(sum);
105 </script>
106 </body>
107 </html>
```

<dsc />



JavaScript Number Methods :

- number()
- parseInt()
- parseFloat()
- isFinite()
- isInteger()
- toFixed(x)
- toPrecision(x)

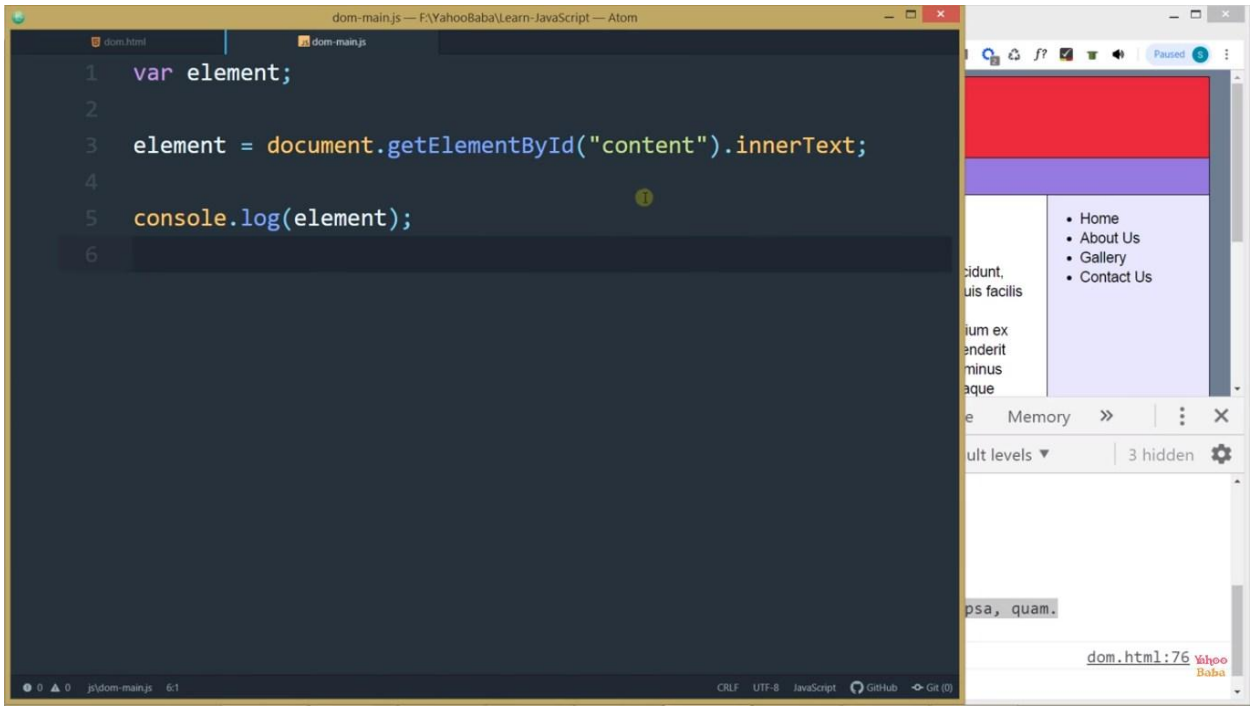
JavaScript Math Methods :

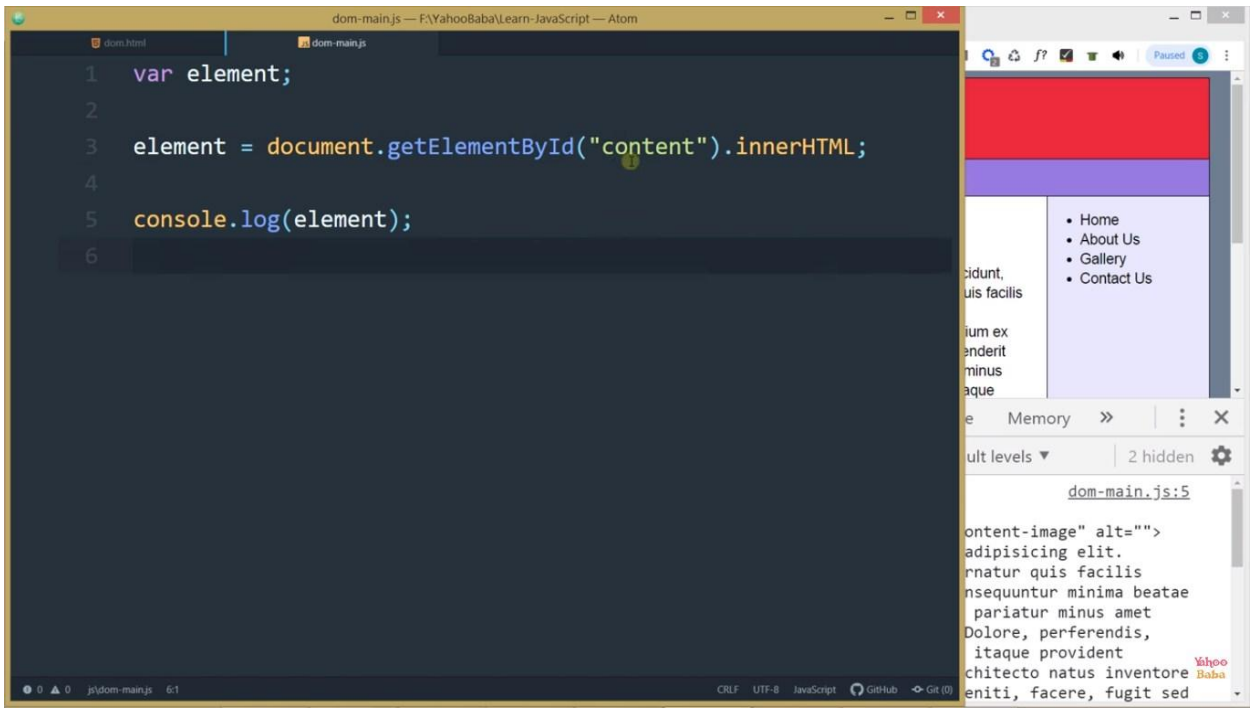
- toDateString()
- getDate()
- getFullYear()
- getMonth()
- getDay()
- getHours()
- getMinutes()
- getSeconds()
- getMilliseconds()
- setDate()
- setFullYear()
- setHours()
- setMilliseconds()
- setMinutes()
- setMonth()
- setSeconds()

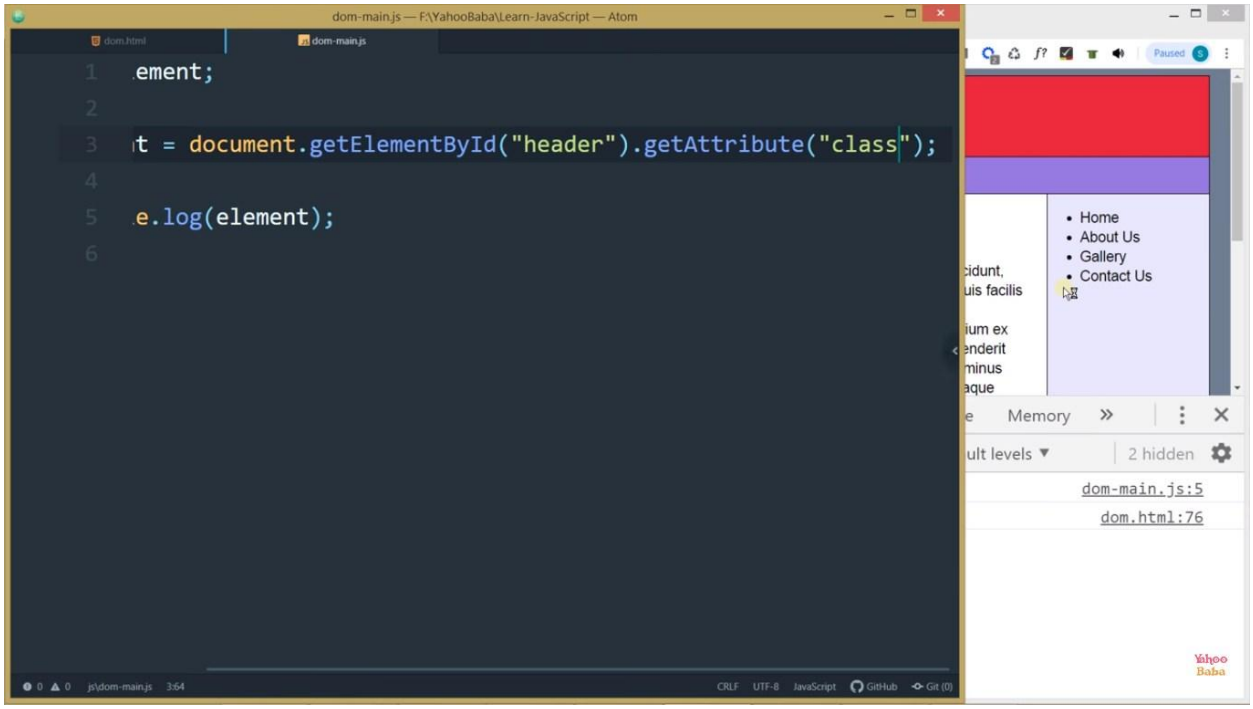
DOM Get Methods :

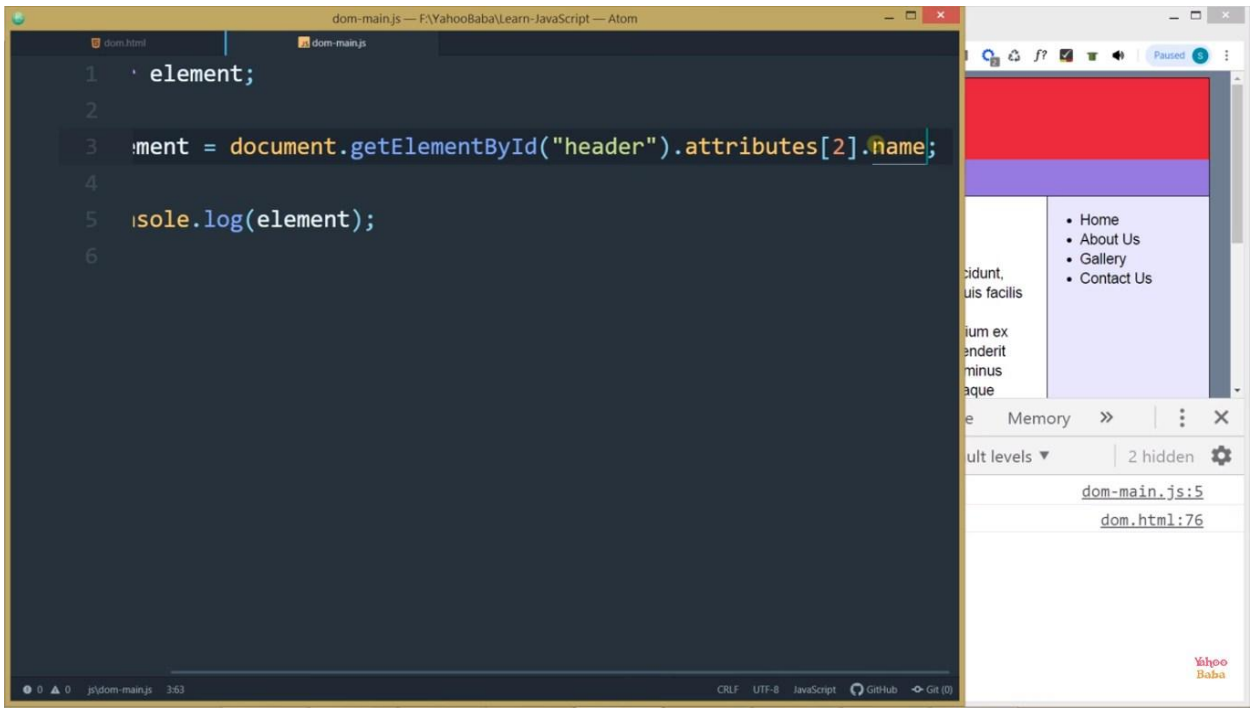
- innerText
- innerHTML
- getAttribute
- getAttributeNode
- Attributes

Yahoo Baba




dom-main.js — F:\YahooBaba\Learn-JavaScript — Atom
dom.html
dom-main.js
var element;
element = document.getElementById("content").innerHTML;
console.log(element);
js\dom-main.js 6:1
CRLF UTF-8 JavaScript GitHub Git (0)
Home
About Us
Gallery
Contact Us
Memory
2 hidden
dom-main.js:5
ontent-image" alt="">
adipisicing elit.
rnatur quis facilis
nsequuntur minima beatae
pariatur minus amet
Dolore, perferendis,
itaque provident
chitecto natus inventore
eniti, facere, fugit sed

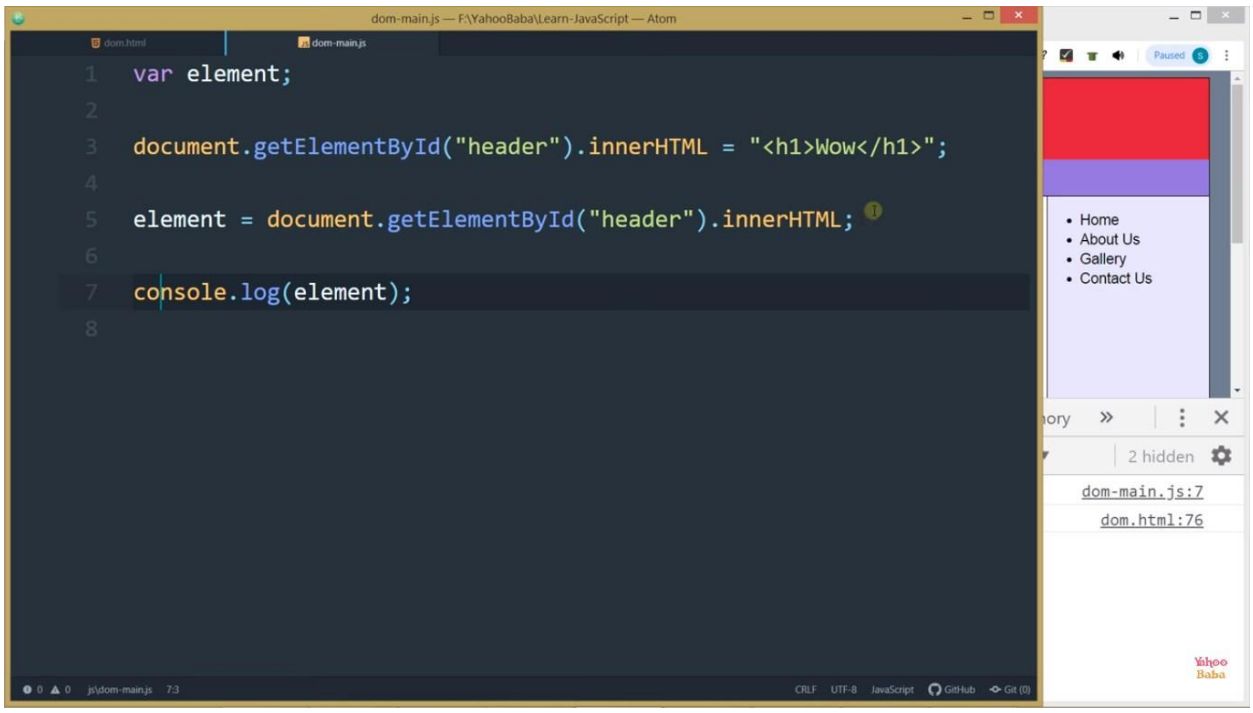

dom-main.js — F:\YahooBaba\Learn-JavaScript — Atom
dom.html
dom-main.js
ement;
t = document.getElementById("header").getAttribute("class");
e.log(element);
js\dom-main.js 3:64
CRLF UTF-8 JavaScript GitHub Git (0)
Home
About Us
Gallery
Contact Us
Memory
2 hidden
dom-main.js:5
dom.html:76

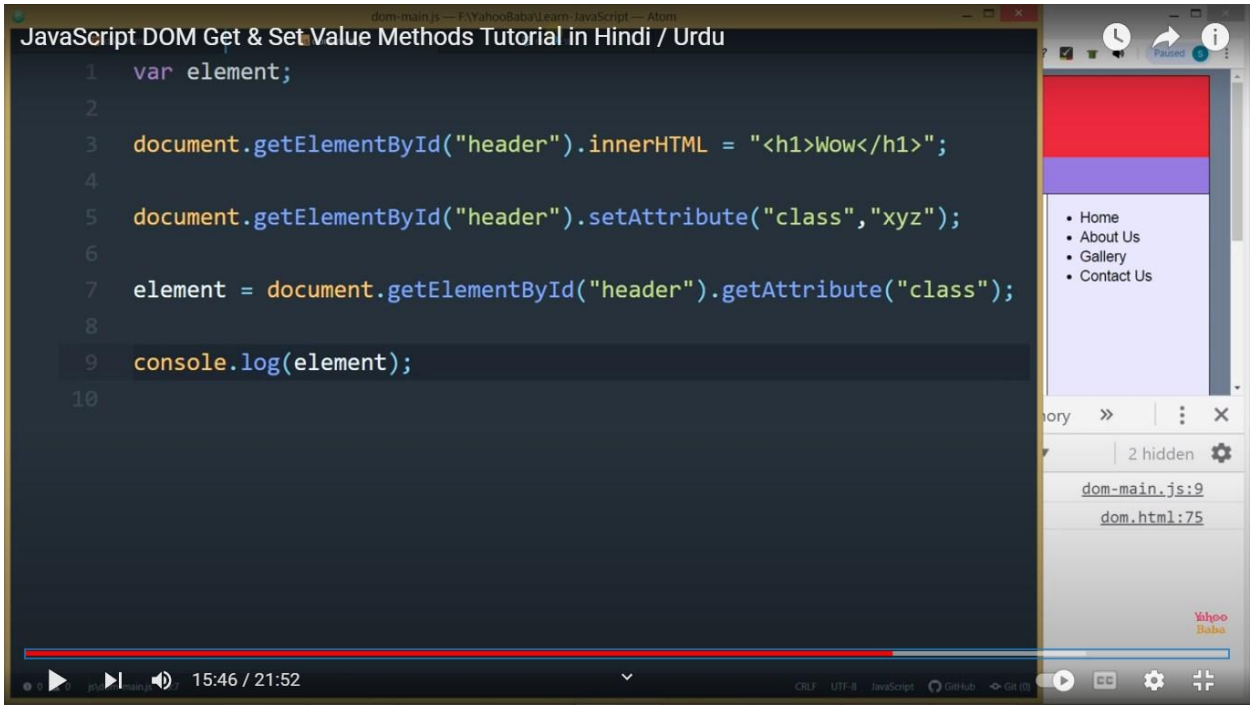


DOM Set Methods :

- innerText
- innerHTML
- setAttribute
- Attribute
- removeAttribute

Yahoo Baba

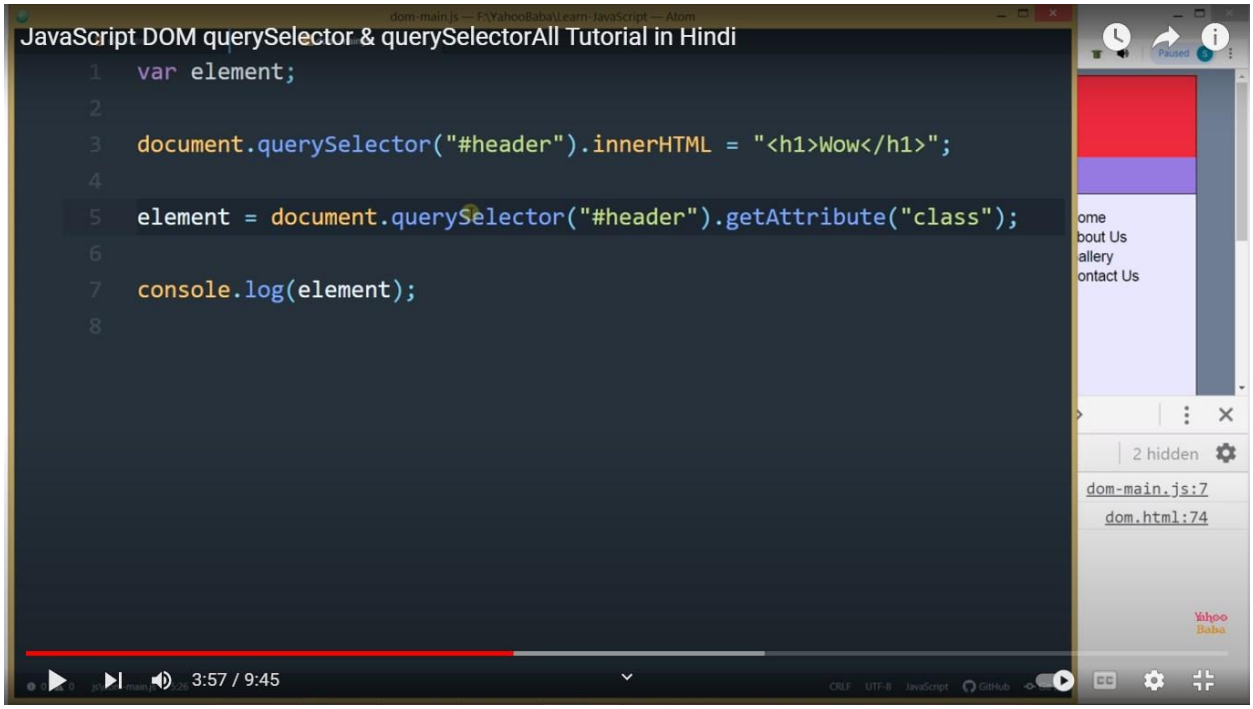

dom-main.js — F:\YahooBaba\Learn-JavaScript — Atom
dom.html
dom-main.js
1 var element;
2
3 document.getElementById("header").innerHTML = "<h1>Wow</h1>";
4
5 element = document.getElementById("header").innerHTML;
6
7 console.log(element);
8
Home
About Us
Gallery
Contact Us
2 hidden
dom-main.js:7
dom.html:76
Yahoo Baba

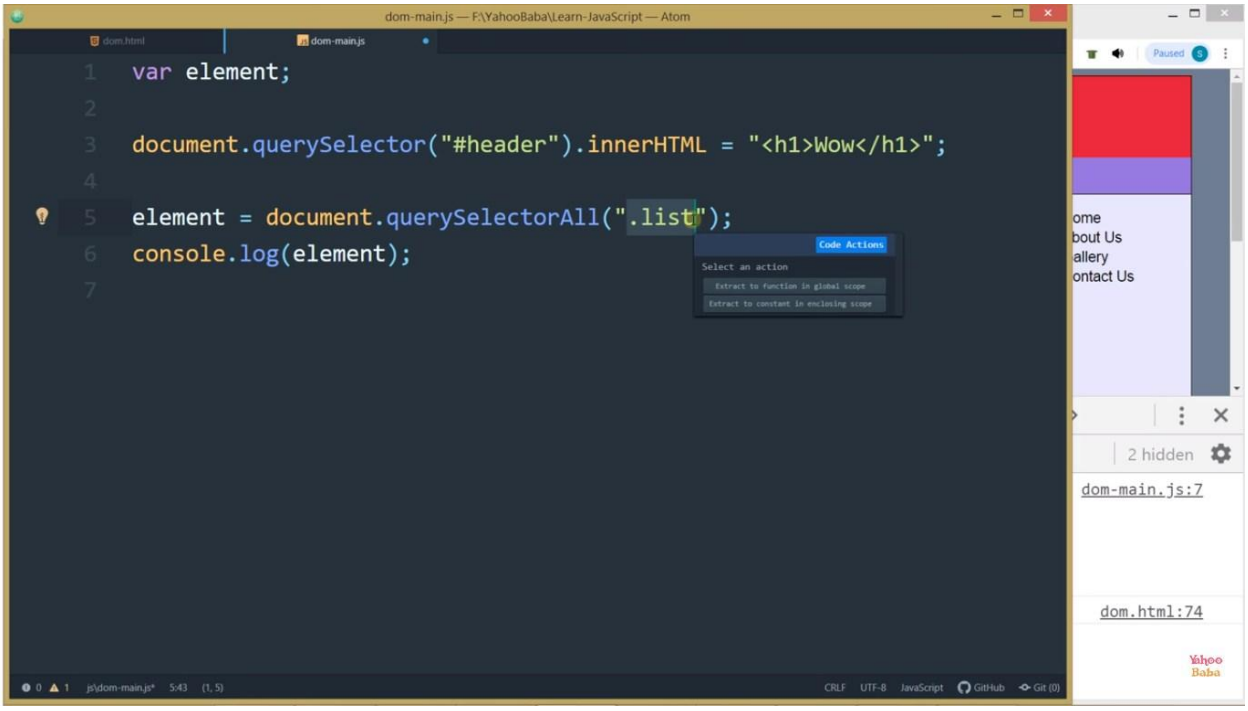

JavaScript DOM Get & Set Value Methods Tutorial in Hindi / Urdu
1 var element;
2
3 document.getElementById("header").innerHTML = "<h1>Wow</h1>";
4
5 document.getElementById("header").setAttribute("class","xyz");
6
7 element = document.getElementById("header").getAttribute("class");
8
9 console.log(element);
10
Home
About Us
Gallery
Contact Us
2 hidden
dom-main.js:9
dom.html:75
Yahoo Baba
15:46 / 21:52

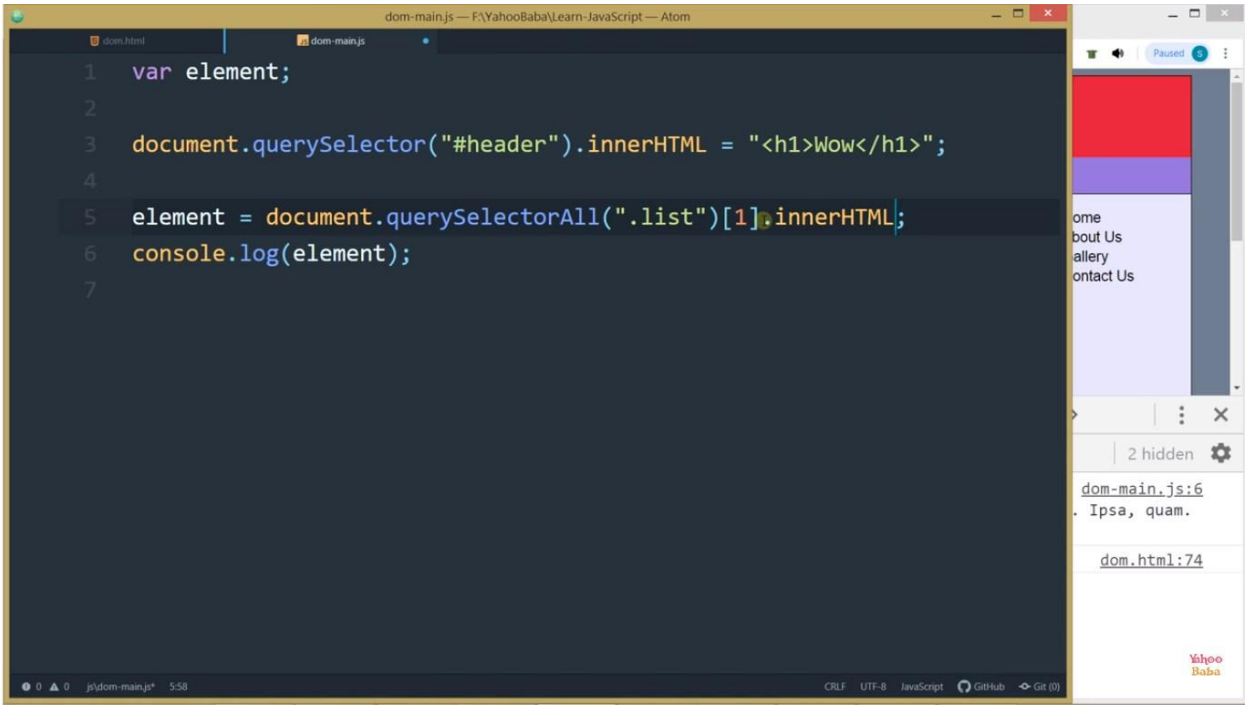
New DOM Targeting Methods :

- querySelector ⟶ document.querySelector(CSS Selector)
- querySelectorAll ⟶ document.querySelectorAll(CSS Selector)

Yahoo Baba



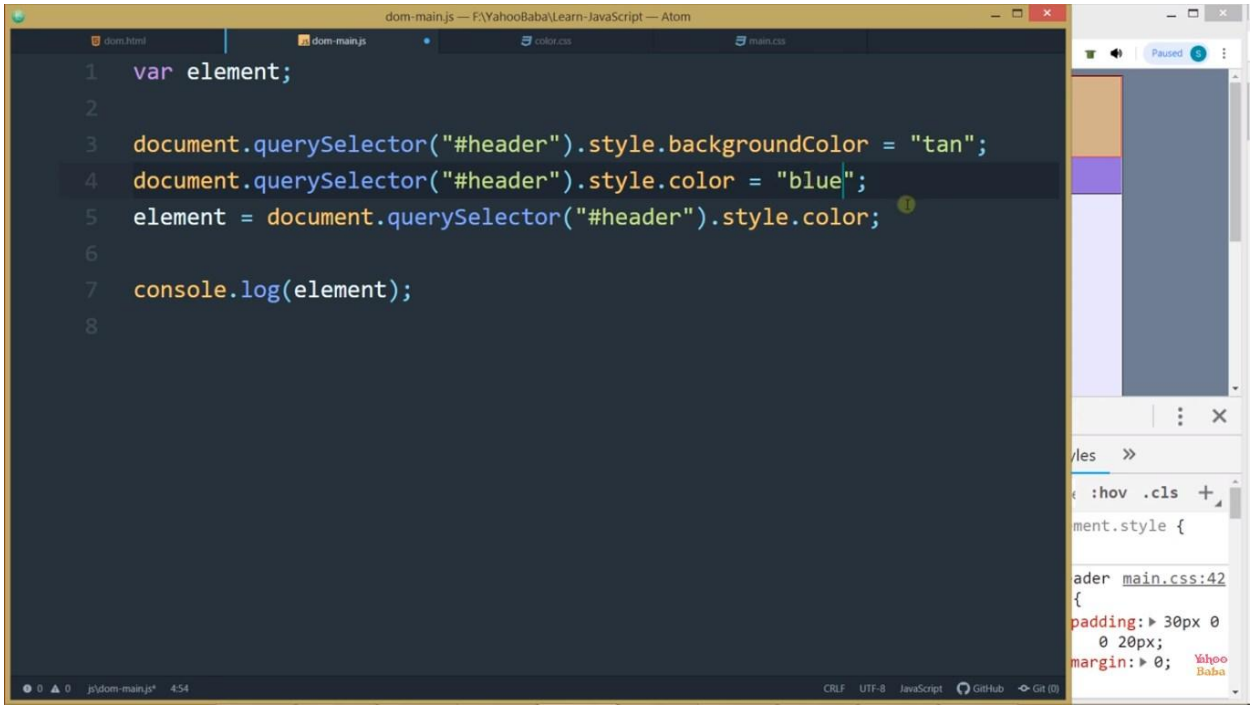

dom-main.js — F:\YahooBaba\Learn-JavaScript — Atom
dom.html
dom-main.js
var element;
document.querySelector("#header").innerHTML = "<h1>Wow</h1>";
element = document.querySelectorAll(".list");
console.log(element);
Code Actions
Select an action
Extract to function in global scope
Extract to constant in enclosing scope
ome
bout Us
allery
ontact Us
2 hidden
dom-main.js:7
dom.html:74
Yahoo
Baba

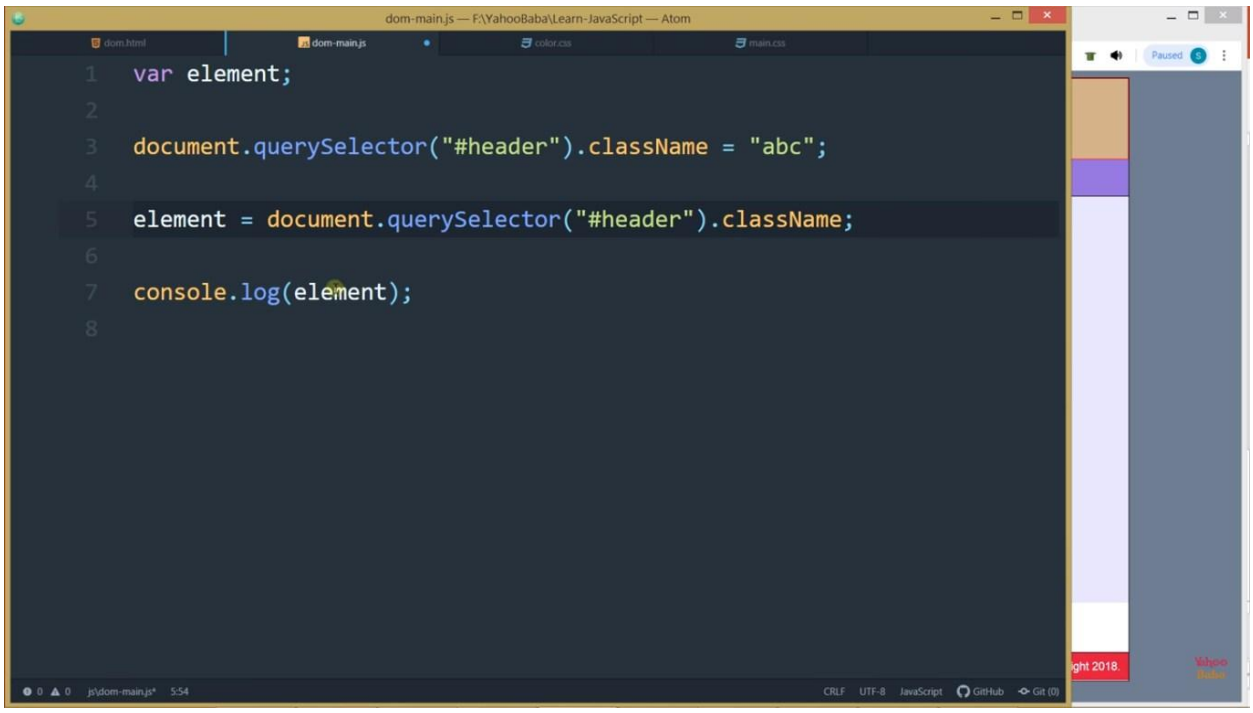

dom-main.js — F:\YahooBaba\Learn-JavaScript — Atom
dom.html
dom-main.js
var element;
document.querySelector("#header").innerHTML = "<h1>Wow</h1>";
element = document.querySelectorAll(".list")[1].innerHTML;
console.log(element);
ome
bout Us
allery
ontact Us
2 hidden
dom-main.js:6
. Ipsa, quam.
dom.html:74
Yahoo
Baba

DOM CSS Styling Methods :

- Style
- className
- classList

Yahoo Baba





JavaScript Basic Events

- Click (onclick)
- Double Click (ondblclick)
- Right Click (oncontextmenu)
- Mouse Hover (onmouseenter)
- Mouse Out (onmouseout)
- Mouse Down (onmousedown)
- Mouse Up (onmouseup)
- Key Press (onkeypress)
- Key Up (onkeyup)
- Load (onload)
- Unload (onunload)
- Resize (onresize)
- Scroll (onscroll)

Yahoo Baba

HTML Event Attributes :

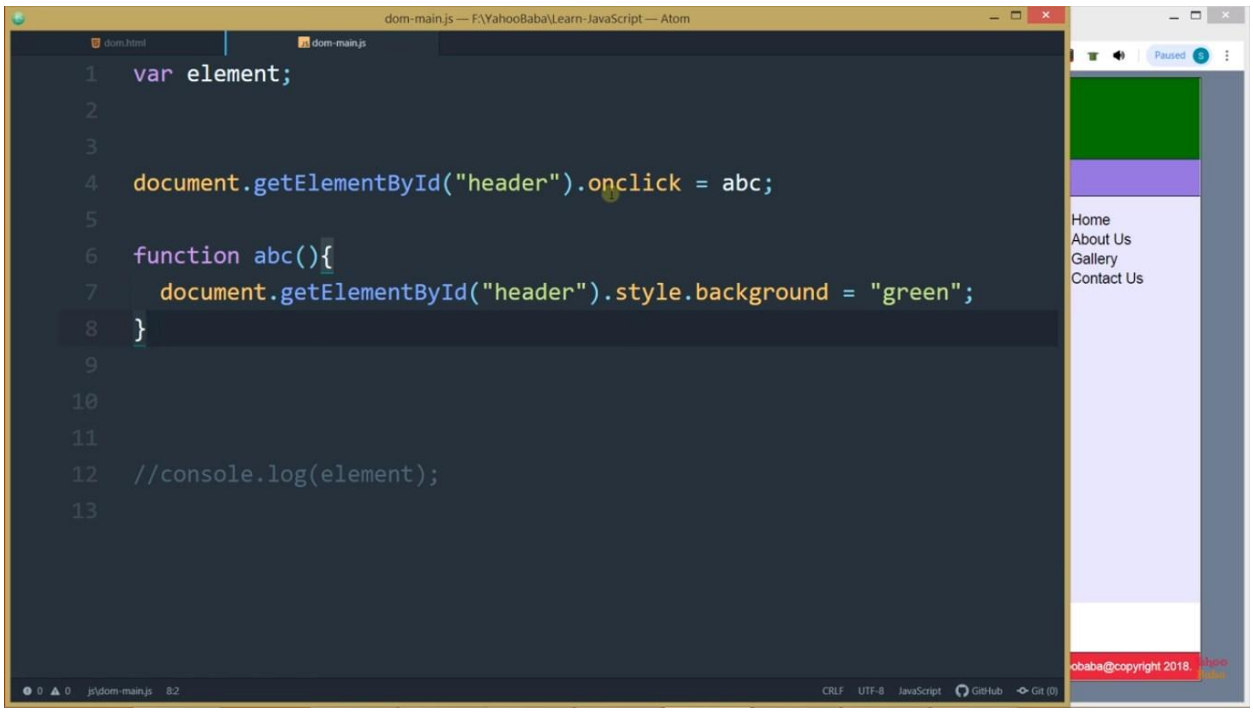
```
<button onClick="abc()"></button>

<img src=".." onmouseenter="xyz()">
```

Yahoo Baba

Assign Events Using the HTML DOM :

```
document.getElementById(Id).onclick = functionName;
```

Yahoo Baba



DOM addEventListener() Method :

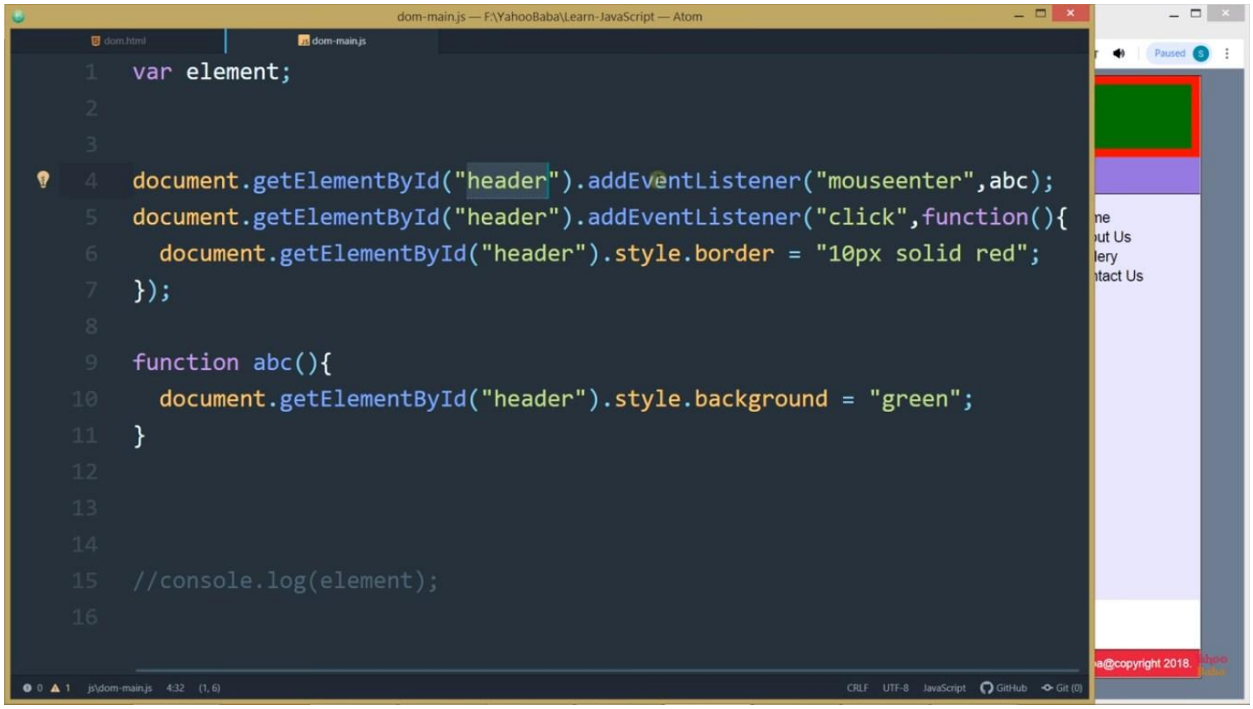
```
document.getElementById(Id). addEventListener("click", functionName);
```

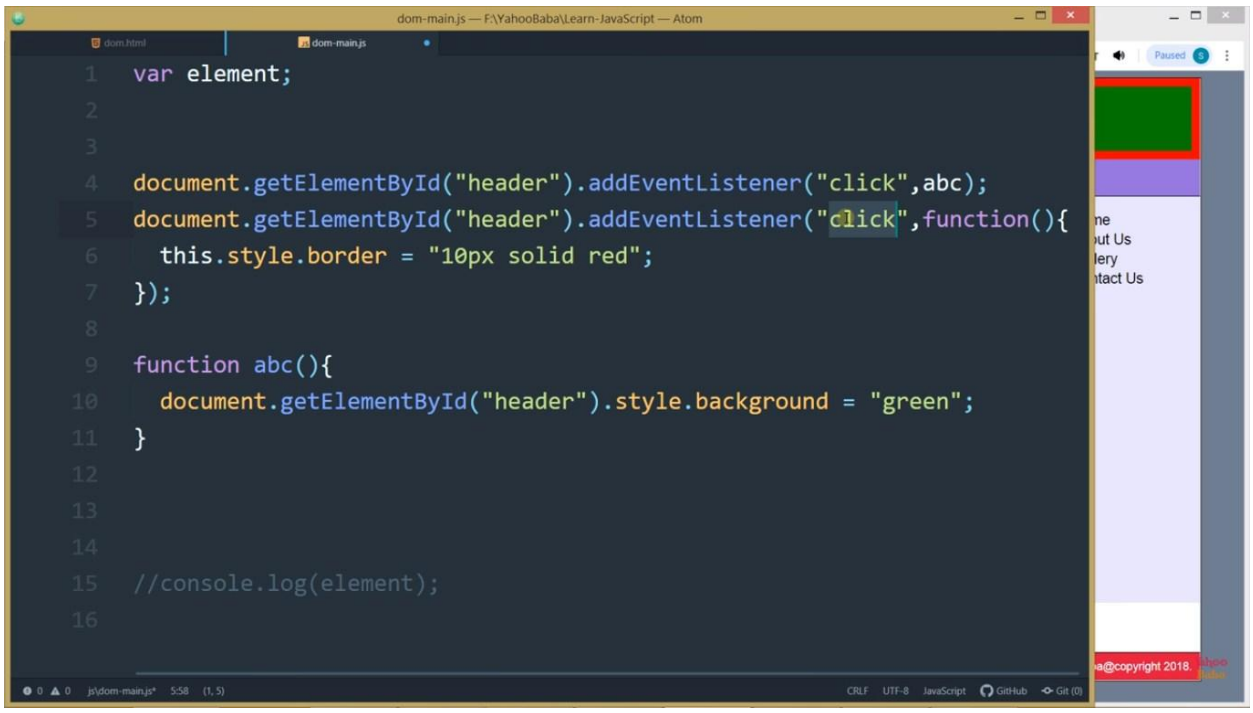
Yahoo Baba

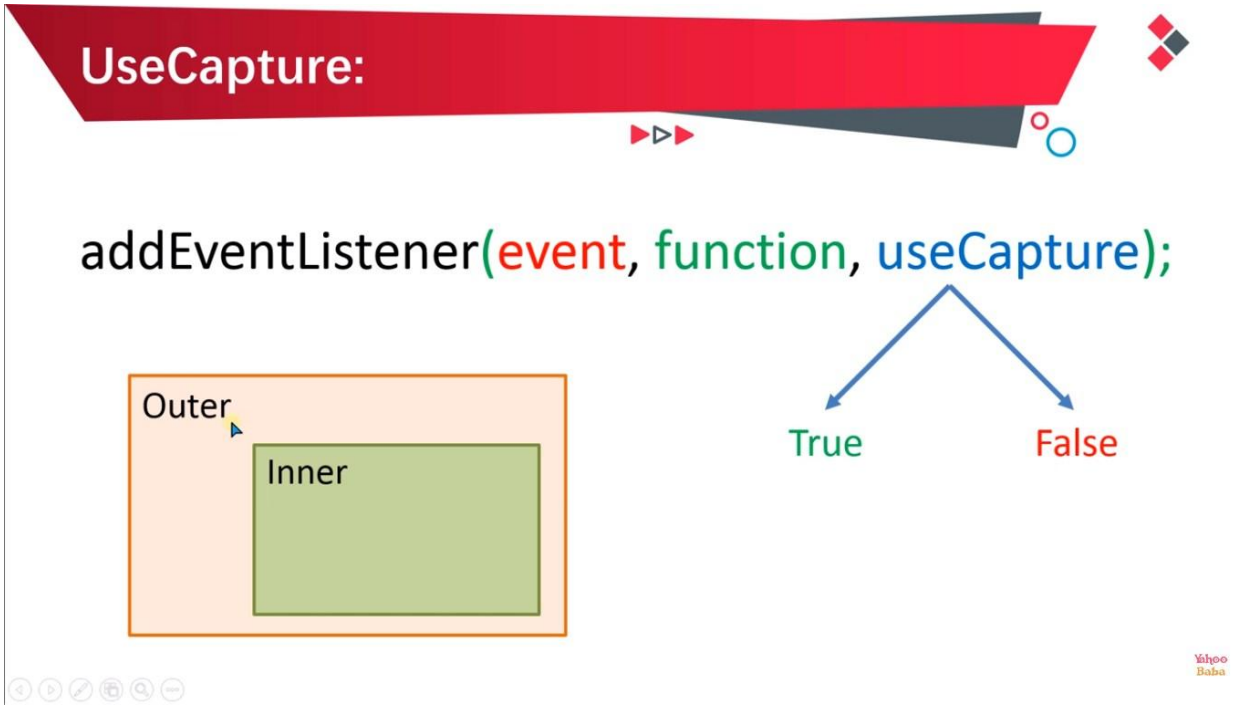
DOM addEventListener() Method :

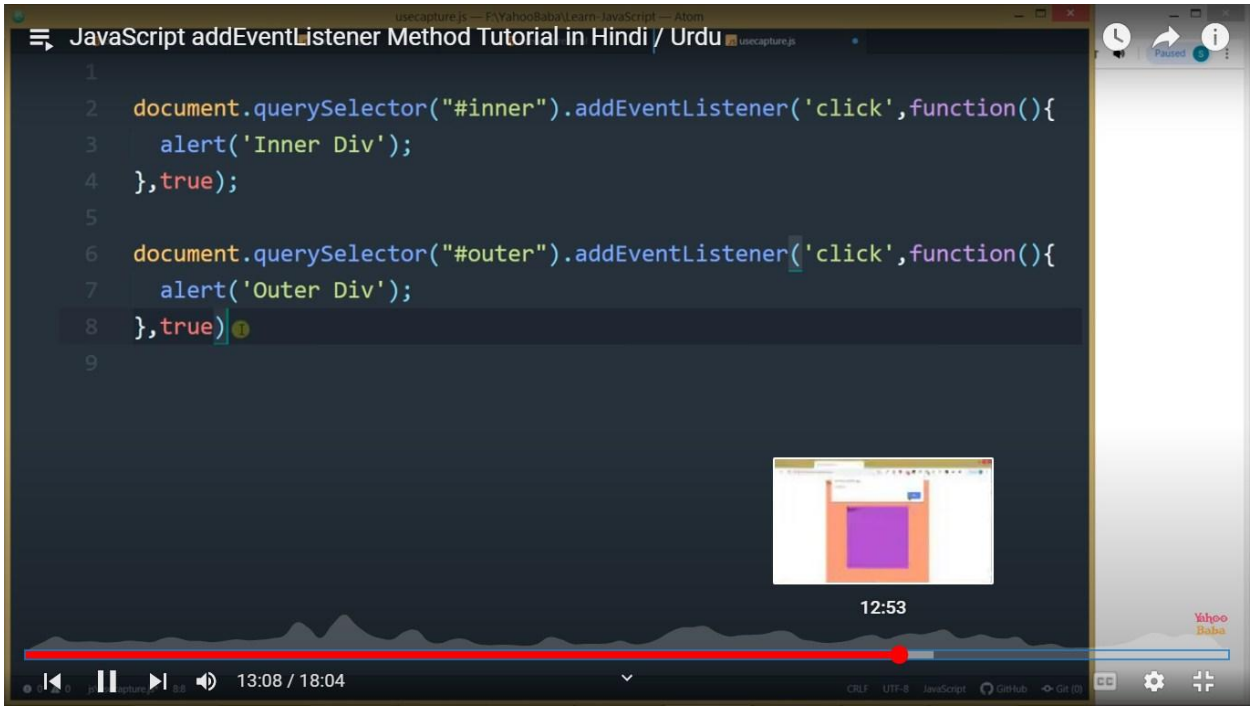
```
document.getElementById(Id). addEventListener("click", functionName);

document.getElementById(Id). addEventListener("click", function(){
        Statement
});
```

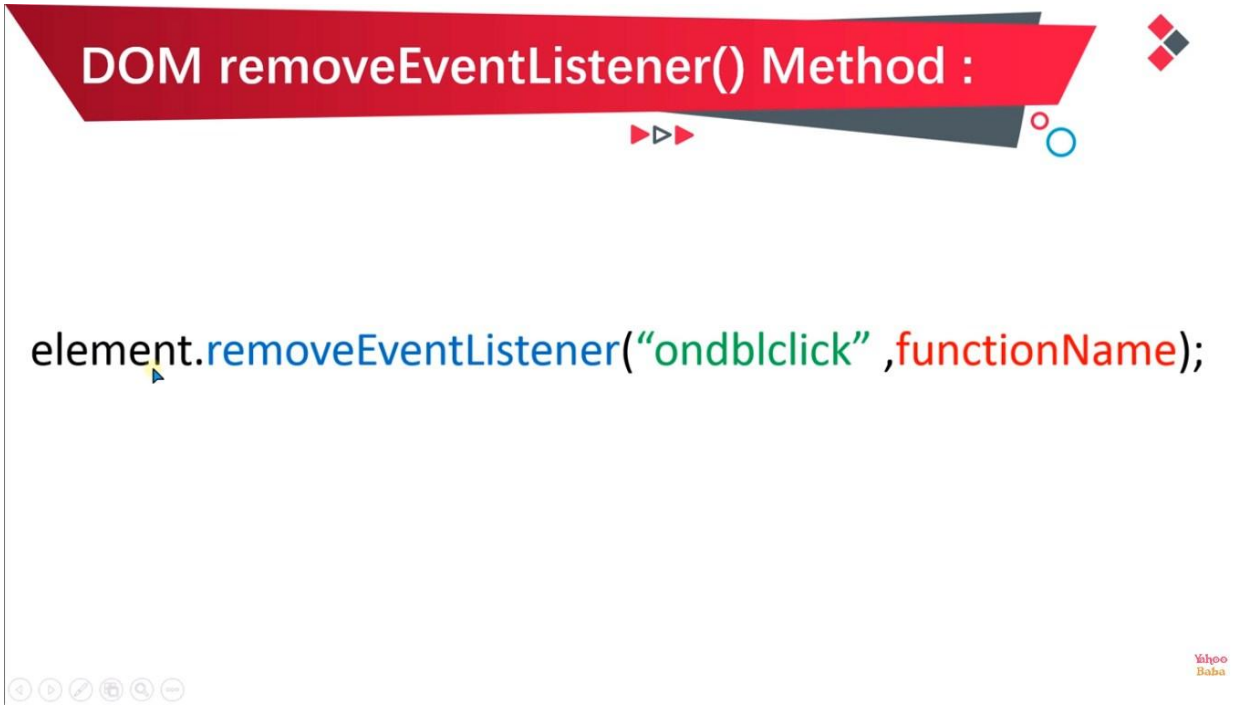
Yahoo Baba

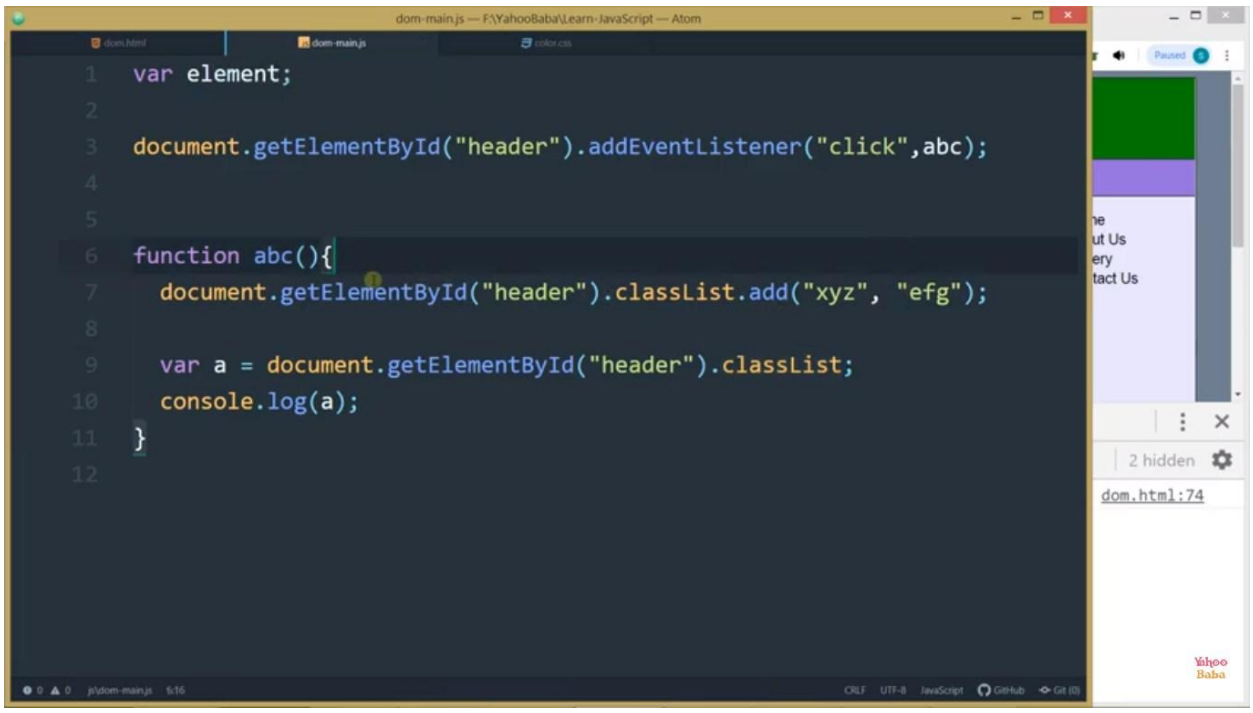


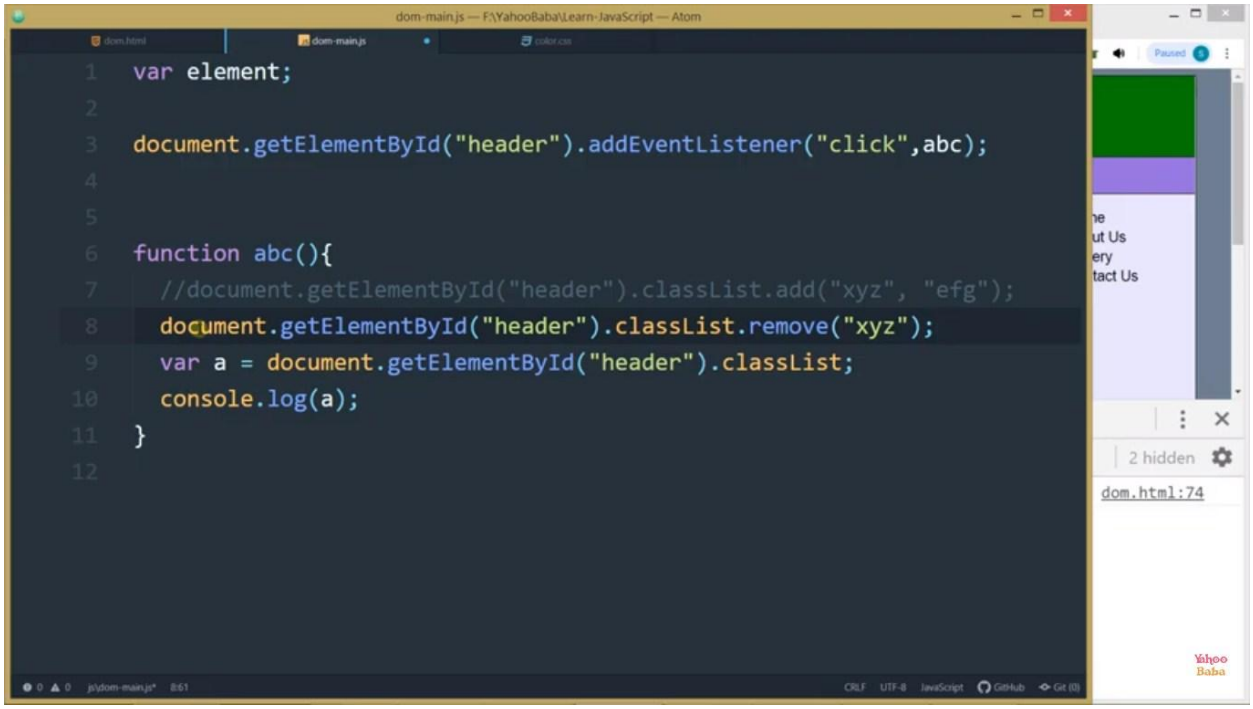

dom-main.js — F:\YahooBaba\Learn-JavaScript — Atom
dom.html
dom-main.js
var element;
document.getElementById("header").addEventListener("click",abc);
document.getElementById("header").addEventListener("click",function(){
this.style.border = "10px solid red";
});
function abc(){
document.getElementById("header").style.background = "green";
}
//console.log(element);
Paused
ne
ut Us
lery
ntact Us
@copyright 2018.


UseCapture:
addEventListener(event, function, useCapture);
Outer
Inner
True
False
Yahoo
Baba


JavaScript addEventListener Method Tutorial in Hindi / Urdu
document.querySelector("#inner").addEventListener('click',function(){
alert('Inner Div');
},true);
document.querySelector("#outer").addEventListener('click',function(){
alert('Outer Div');
},true)
12:53
13:08 / 18:04

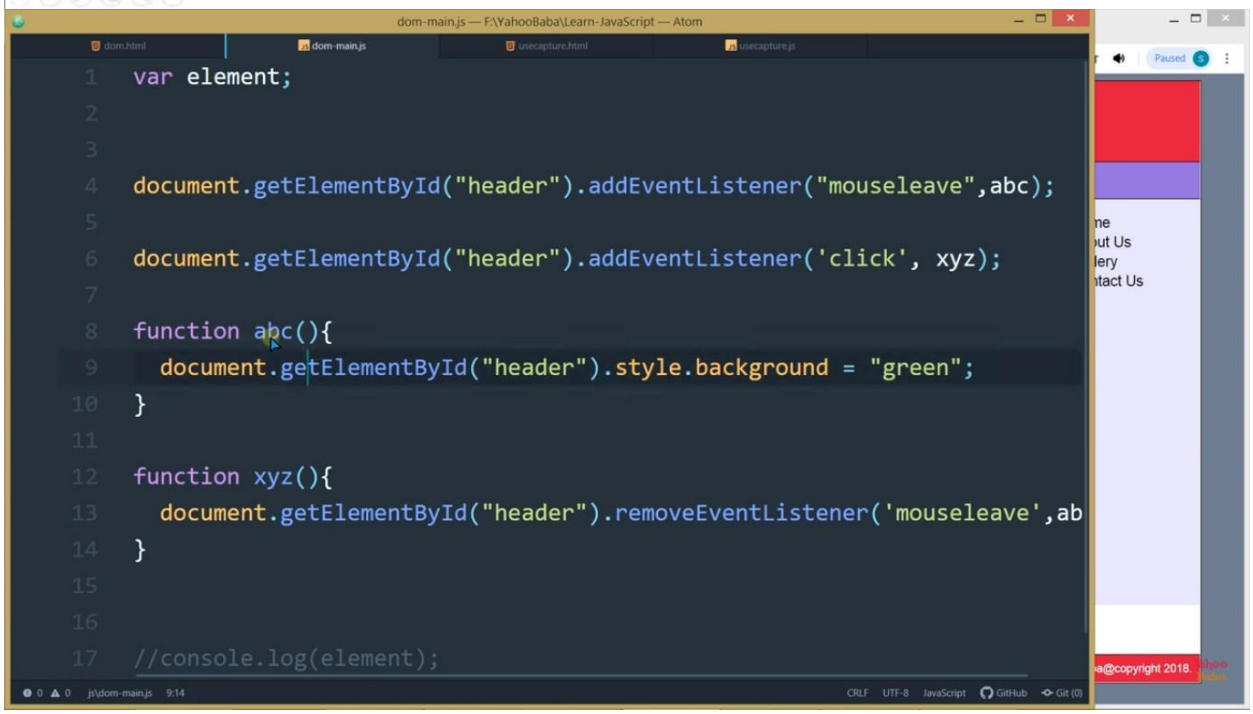

DOM removeEventListener() Method :
element.removeEventListener("ondblclick" ,functionName);


dom-main.js — F:\YahooBaba\Learn-JavaScript — Atom
dom.html
dom-main.js
color.css
var element;
document.getElementById("header").addEventListener("click",abc);
function abc(){
document.getElementById("header").classList.add("xyz", "efg");
var a = document.getElementById("header").classList;
console.log(a);
}
Paused
2 hidden
dom.html:74
Yahoo Baba


dom-main.js — F:\YahooBaba\Learn-JavaScript — Atom
dom.html
dom-main.js
color.css
var element;
document.getElementById("header").addEventListener("click",abc);
function abc(){
//document.getElementById("header").classList.add("xyz", "efg");
document.getElementById("header").classList.remove("xyz");
var a = document.getElementById("header").classList;
console.log(a);
}
Paused
2 hidden
dom.html:74
Yahoo Baba

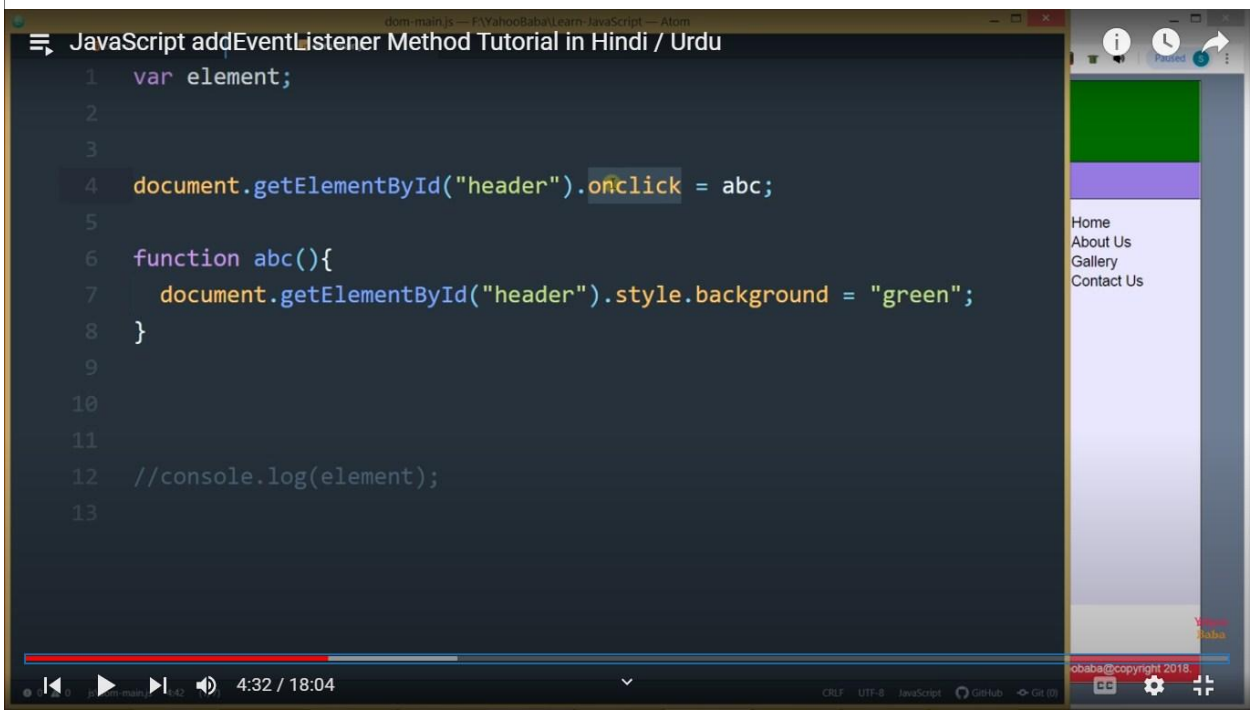
JavaScript classList Methods :

- add(class1, class2, ...)
- remove(class1, class2, ...)
- toggle(class)
- contains(class)
- item(index)
- Length



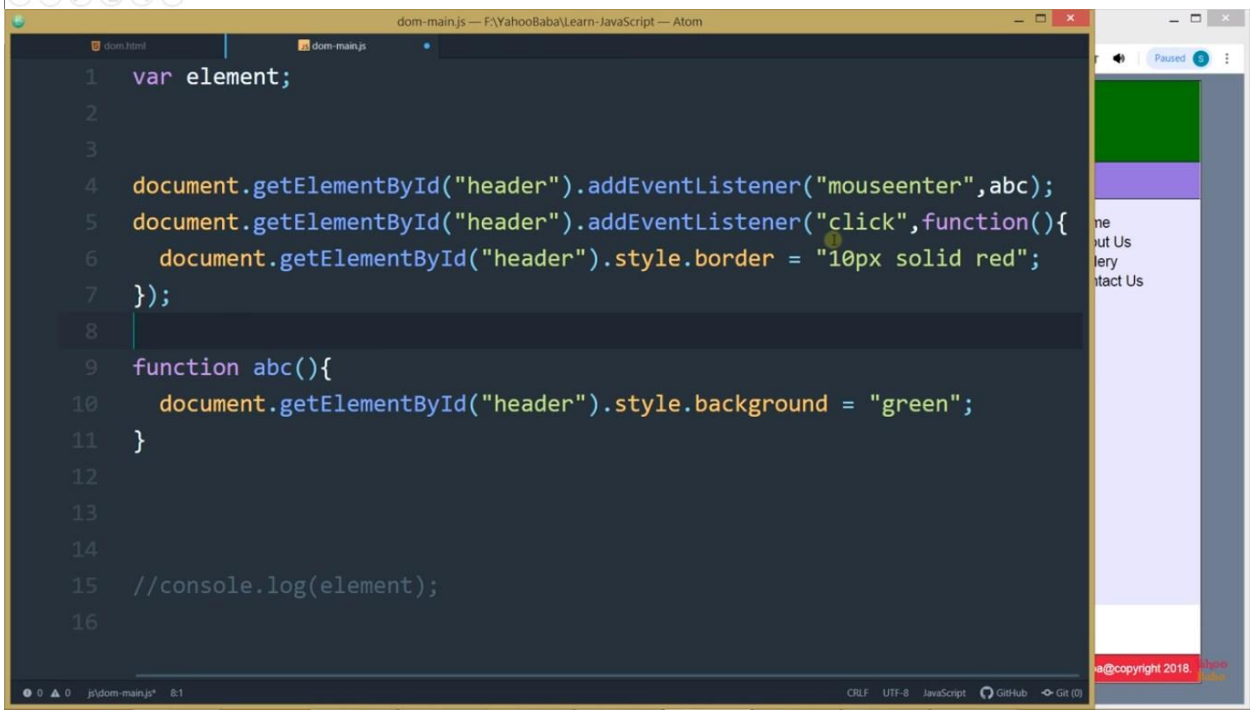
HTML Event Attributes :

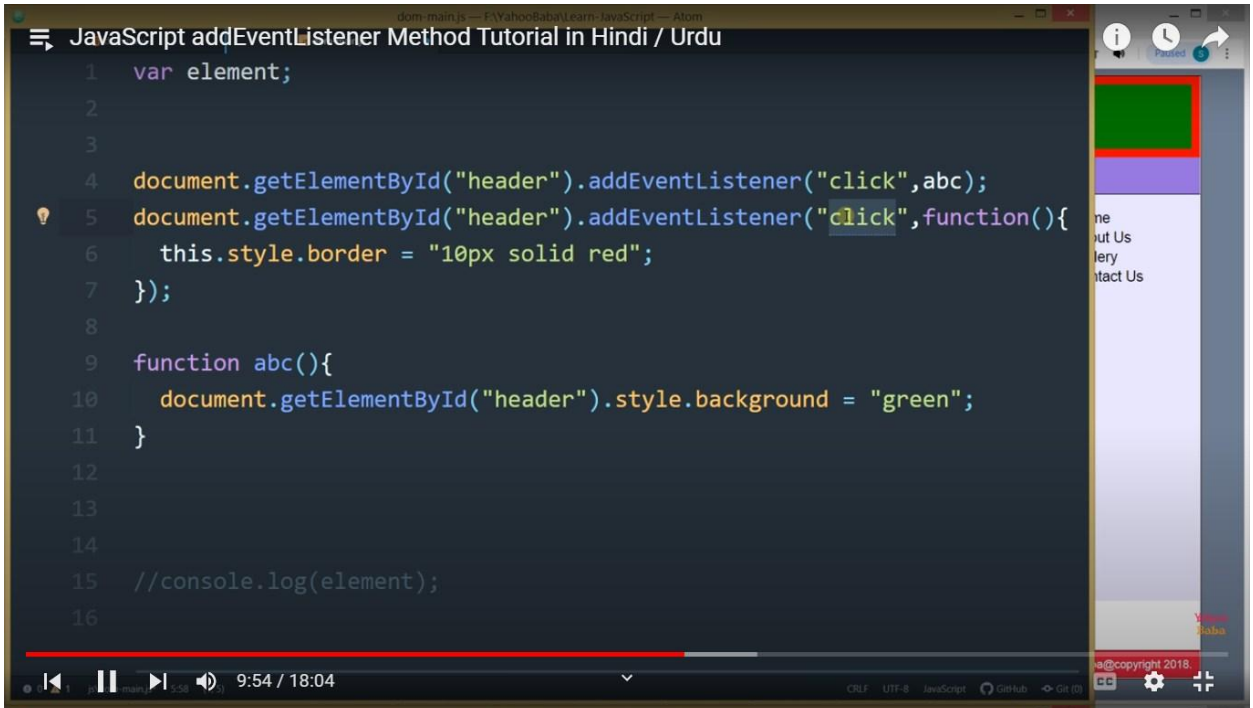
```
<button onClick="abc()"></button>

<img src=".." onmouseenter="xyz()">
```

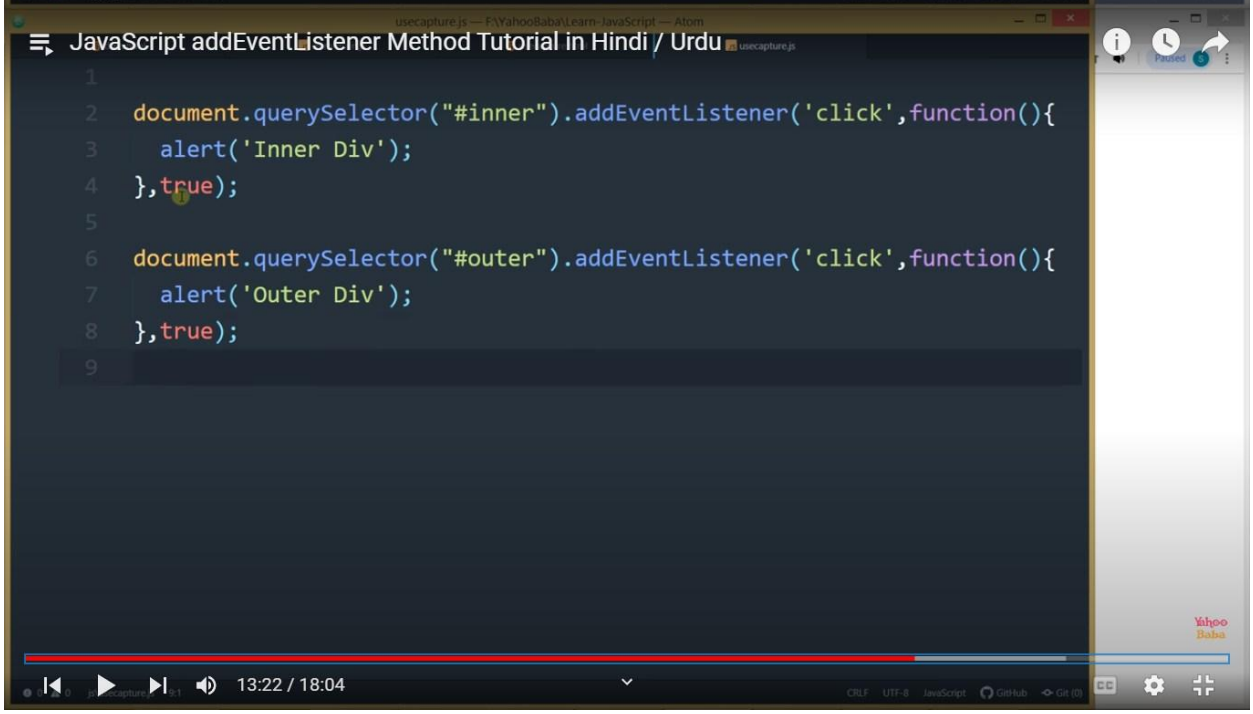


DOM addEventListener() Method :

```
document.getElementById(Id). addEventListener("click", functionName);

document.getElementById(Id). addEventListener("click", function(){

});
```




JavaScript addEventListener Method Tutorial in Hindi / Urdu
var element;
document.getElementById("header").addEventListener("click",abc);
document.getElementById("header").addEventListener("click",function(){
this.style.border = "10px solid red";
});
function abc(){
document.getElementById("header").style.background = "green";
}
//console.log(element);
9:54 / 18:04


JavaScript addEventListener Method Tutorial in Hindi / Urdu
document.querySelector("#inner").addEventListener('click',function(){
alert('Inner Div');
},true);
document.querySelector("#outer").addEventListener('click',function(){
alert('Outer Div');
},true);
13:22 / 18:04

DOM removeEventListener() Method :

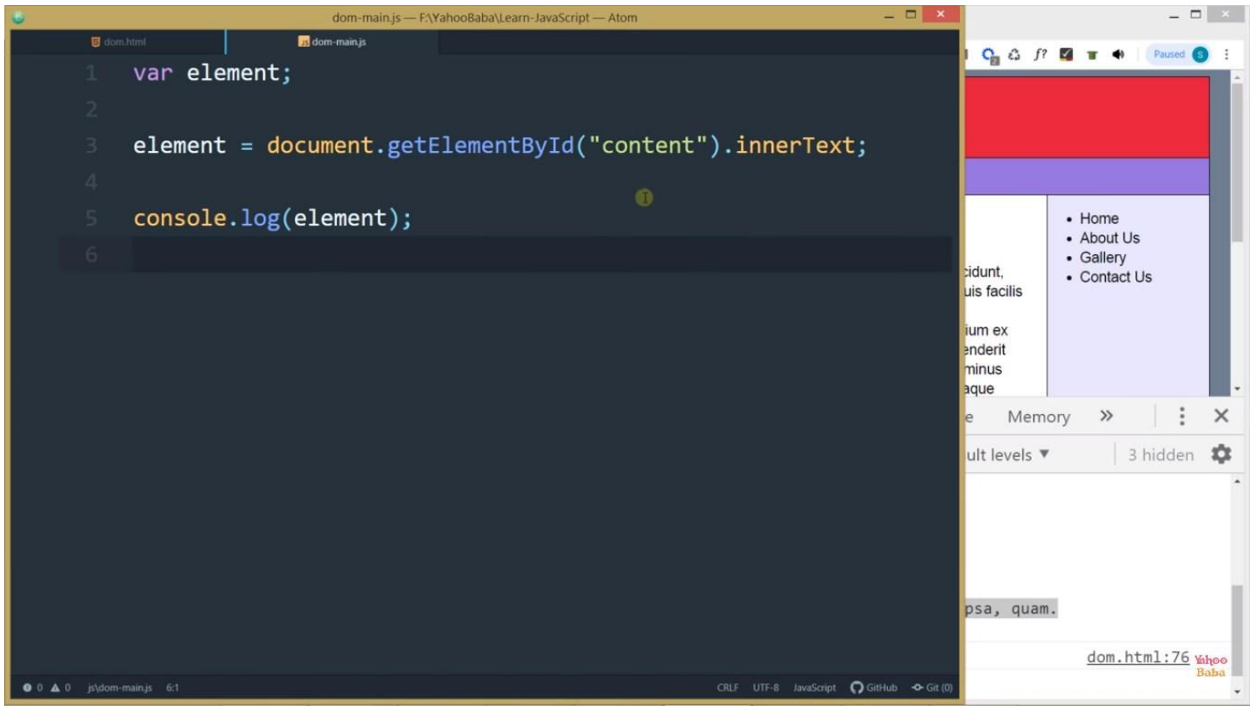
```
element.removeEventListener("ondblclick" ,functionName);
```

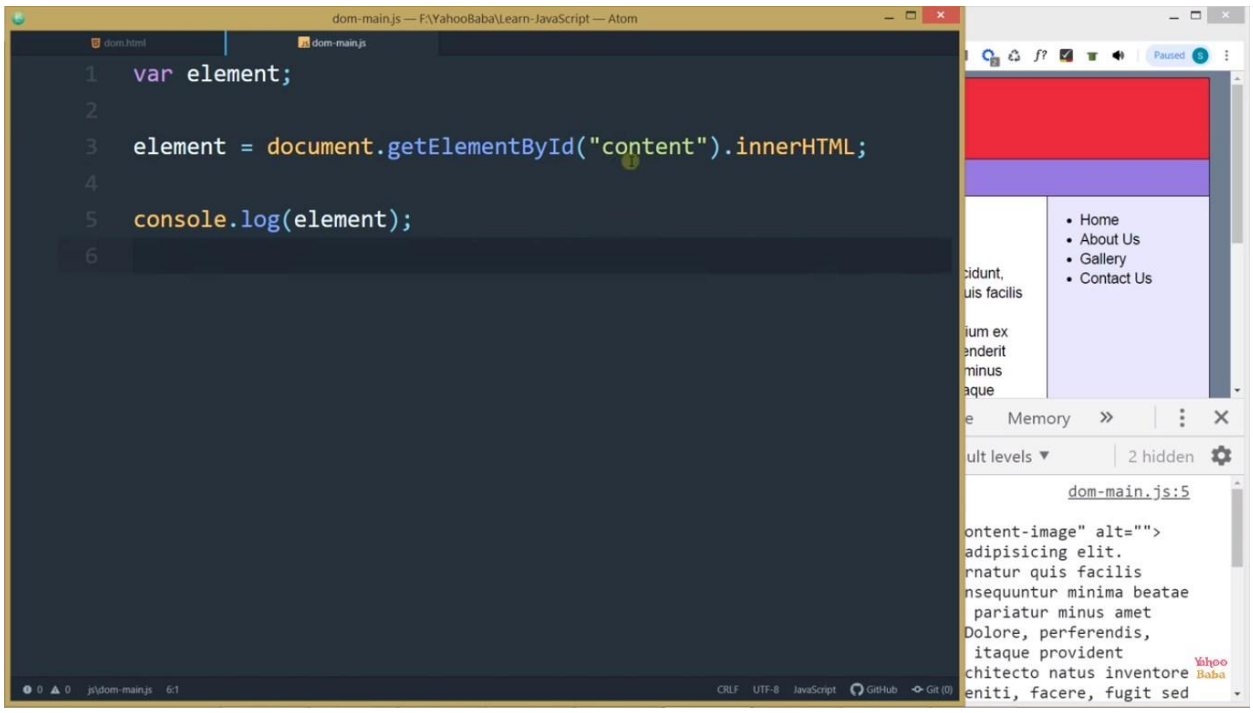
Yahoo Baba

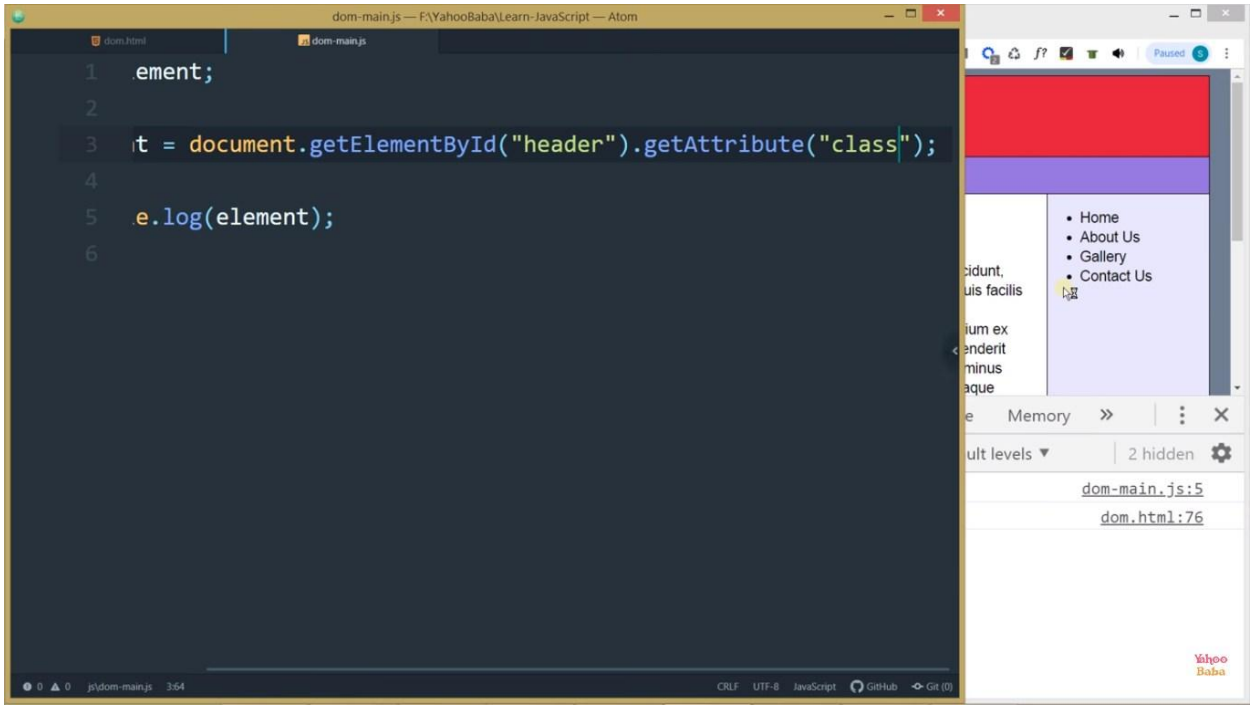
DOM Get Methods :

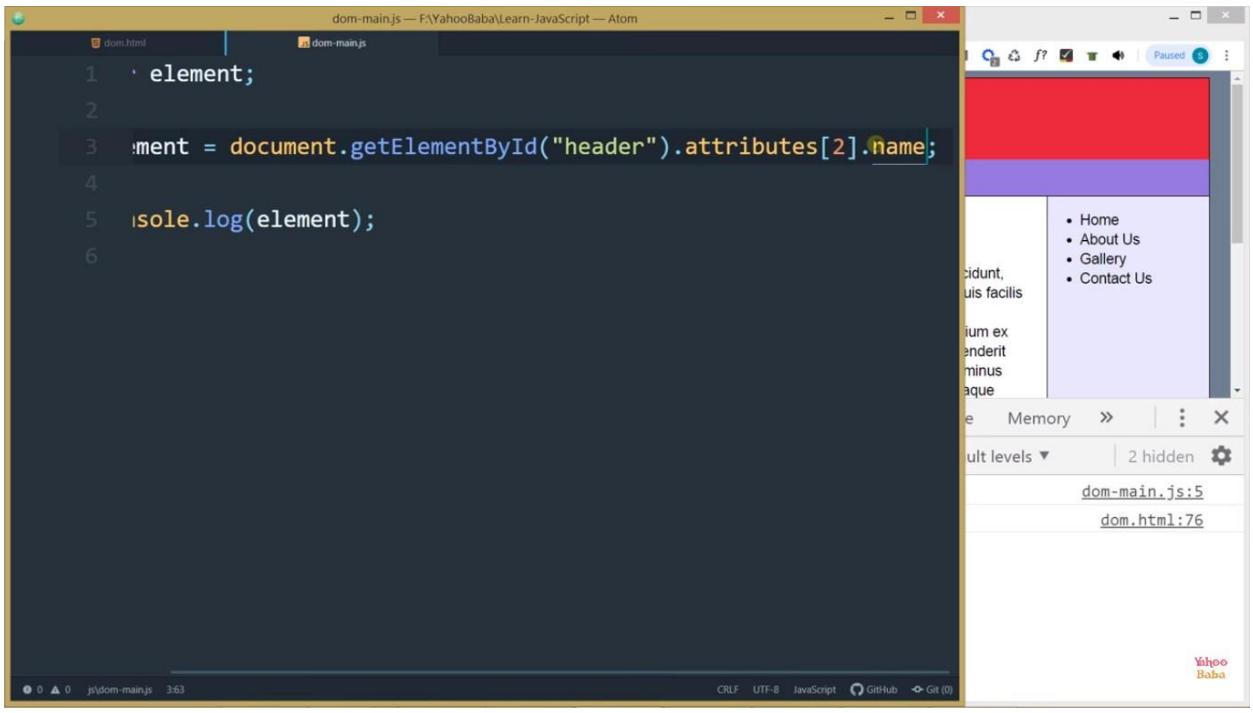
- innerText
- innerHTML
- getAttribute
- getAttributeNode
- Attributes

Yahoo Baba




dom-main.js — F:\YahooBaba\Learn-JavaScript — Atom
dom.html
dom-main.js
var element;
element = document.getElementById("content").innerHTML;
console.log(element);
js\dom-main.js 6:1
CRLF UTF-8 JavaScript GitHub Git (0)
Home
About Us
Gallery
Contact Us
Memory
2 hidden
dom-main.js:5
ontent-image" alt="">
adipisicing elit.
rnatur quis facilis
nsequuntur minima beatae
pariatur minus amet
Dolore, perferendis,
itaque provident
chitecto natus inventore
eniti, facere, fugit sed

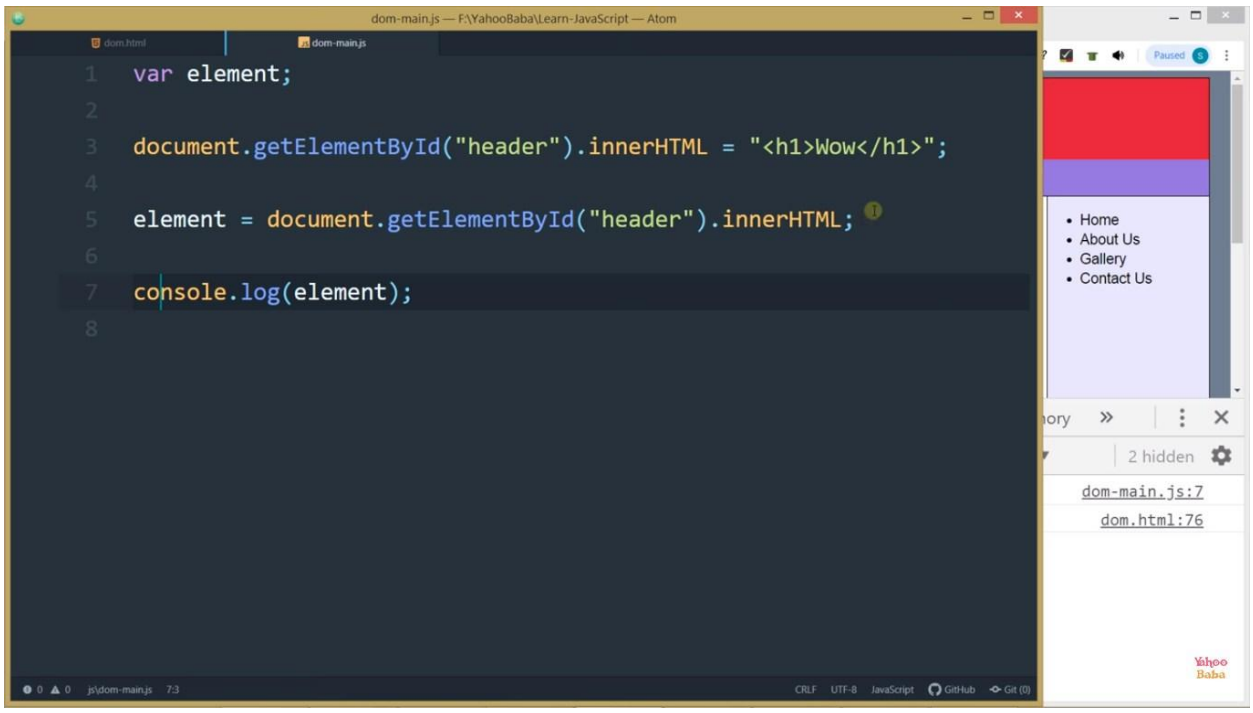

dom-main.js — F:\YahooBaba\Learn-JavaScript — Atom
dom.html
dom-main.js
ement;
t = document.getElementById("header").getAttribute("class");
e.log(element);
js\dom-main.js 3:64
CRLF UTF-8 JavaScript GitHub Git (0)
Home
About Us
Gallery
Contact Us
Memory
2 hidden
dom-main.js:5
dom.html:76

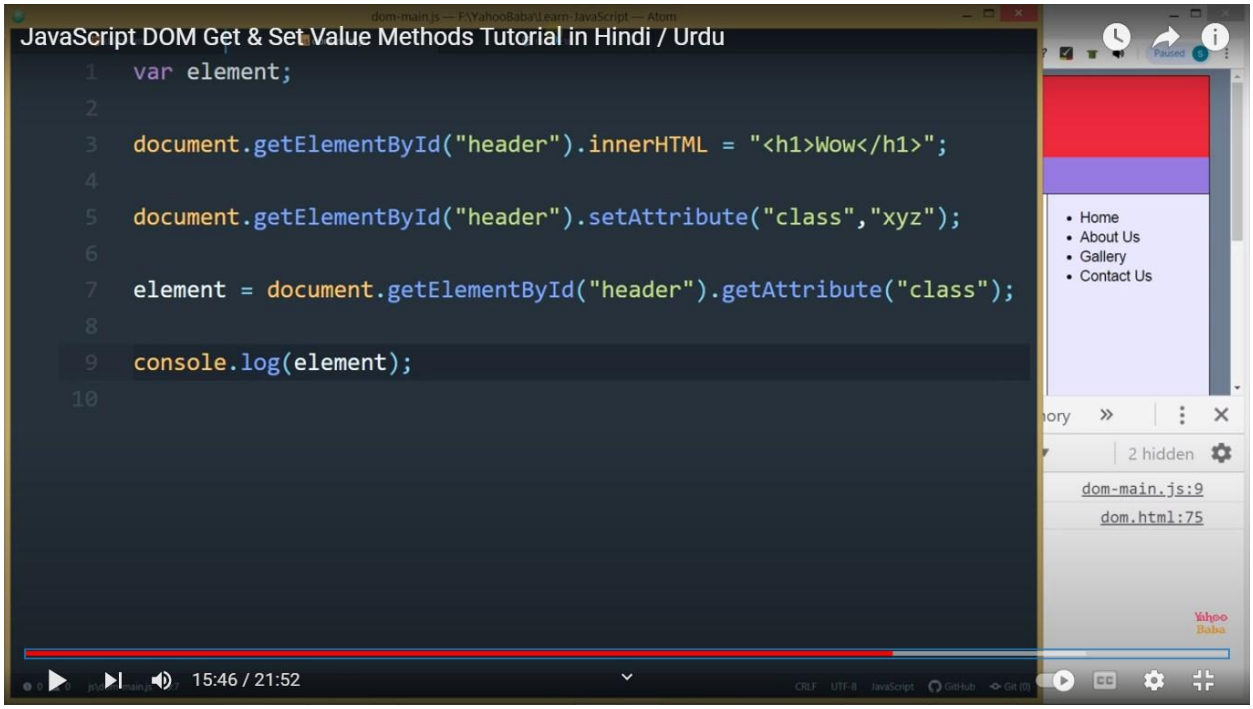


DOM Set Methods :

- innerText
- innerHTML
- setAttribute
- Attribute
- removeAttribute

Yahoo Baba

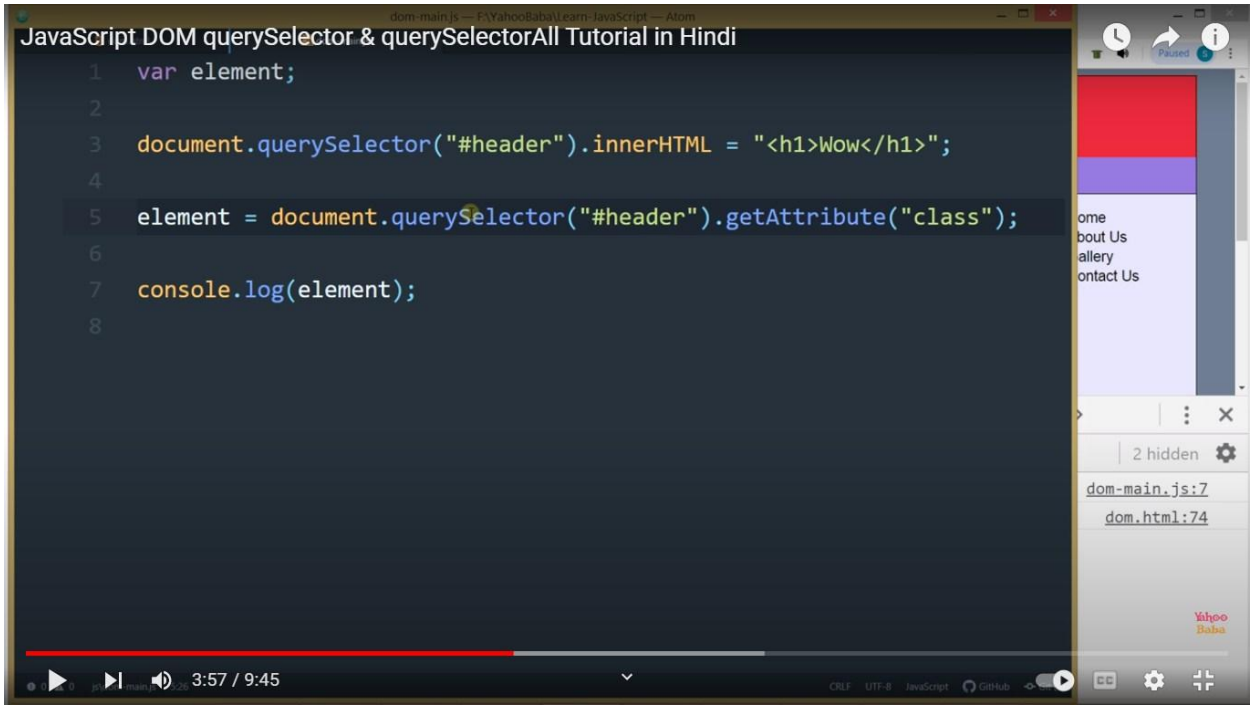

dom-main.js — F:\YahooBaba\Learn-JavaScript — Atom
dom.html
dom-main.js
1 var element;
2
3 document.getElementById("header").innerHTML = "<h1>Wow</h1>";
4
5 element = document.getElementById("header").innerHTML;
6
7 console.log(element);
8
Home
About Us
Gallery
Contact Us
2 hidden
dom-main.js:7
dom.html:76
Yahoo Baba

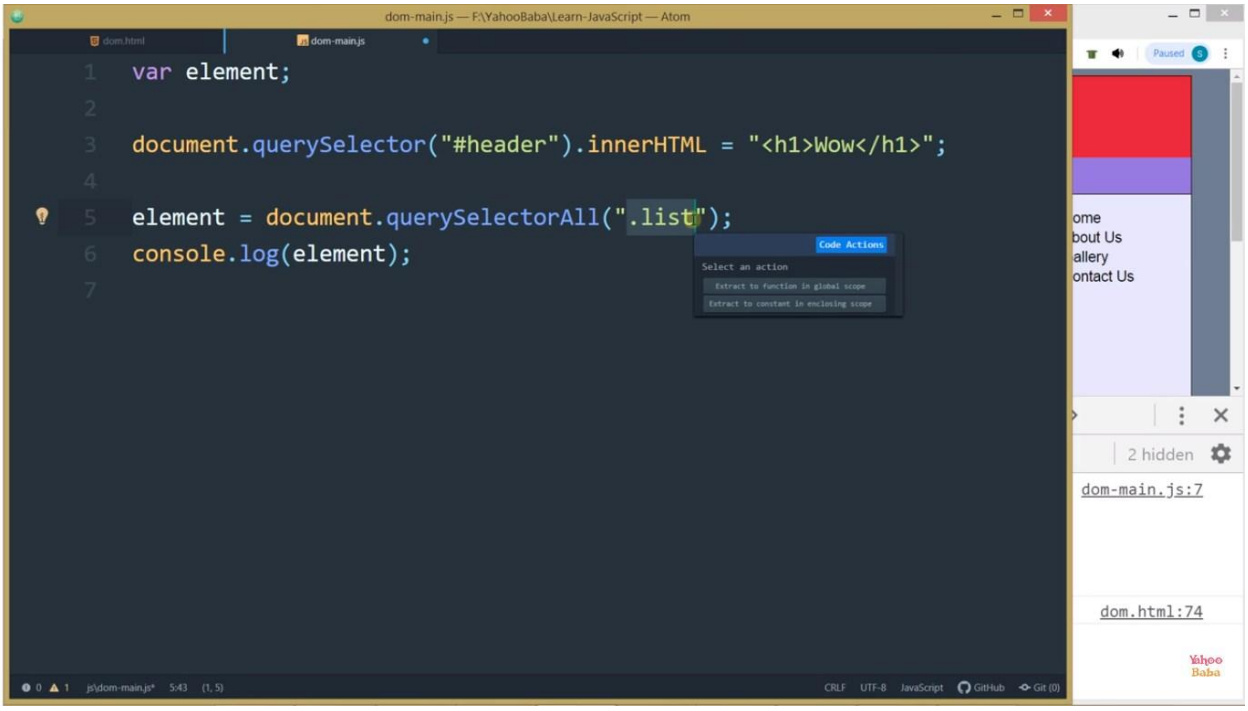

JavaScript DOM Get & Set Value Methods Tutorial in Hindi / Urdu
1 var element;
2
3 document.getElementById("header").innerHTML = "<h1>Wow</h1>";
4
5 document.getElementById("header").setAttribute("class","xyz");
6
7 element = document.getElementById("header").getAttribute("class");
8
9 console.log(element);
10
Home
About Us
Gallery
Contact Us
2 hidden
dom-main.js:9
dom.html:75
Yahoo Baba
15:46 / 21:52

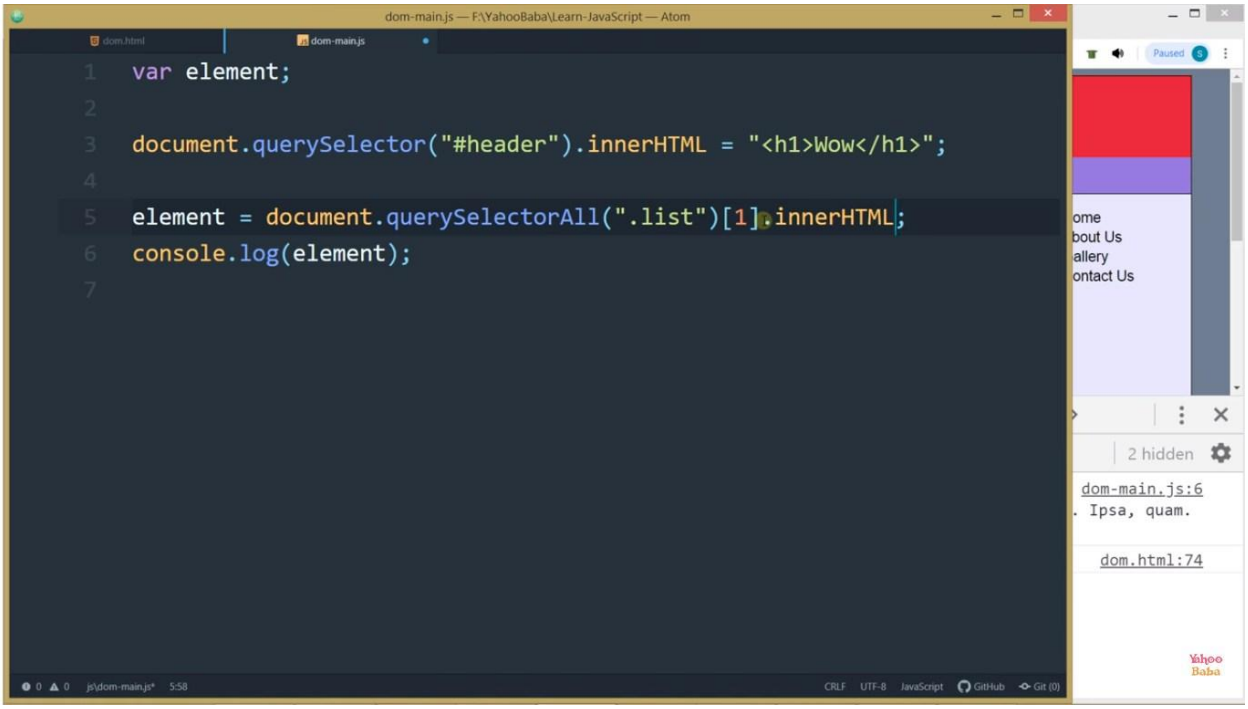
New DOM Targeting Methods :

- querySelector ⟶ document.querySelector(CSS Selector)
- querySelectorAll ⟶ document.querySelectorAll(CSS Selector)

Yahoo Baba



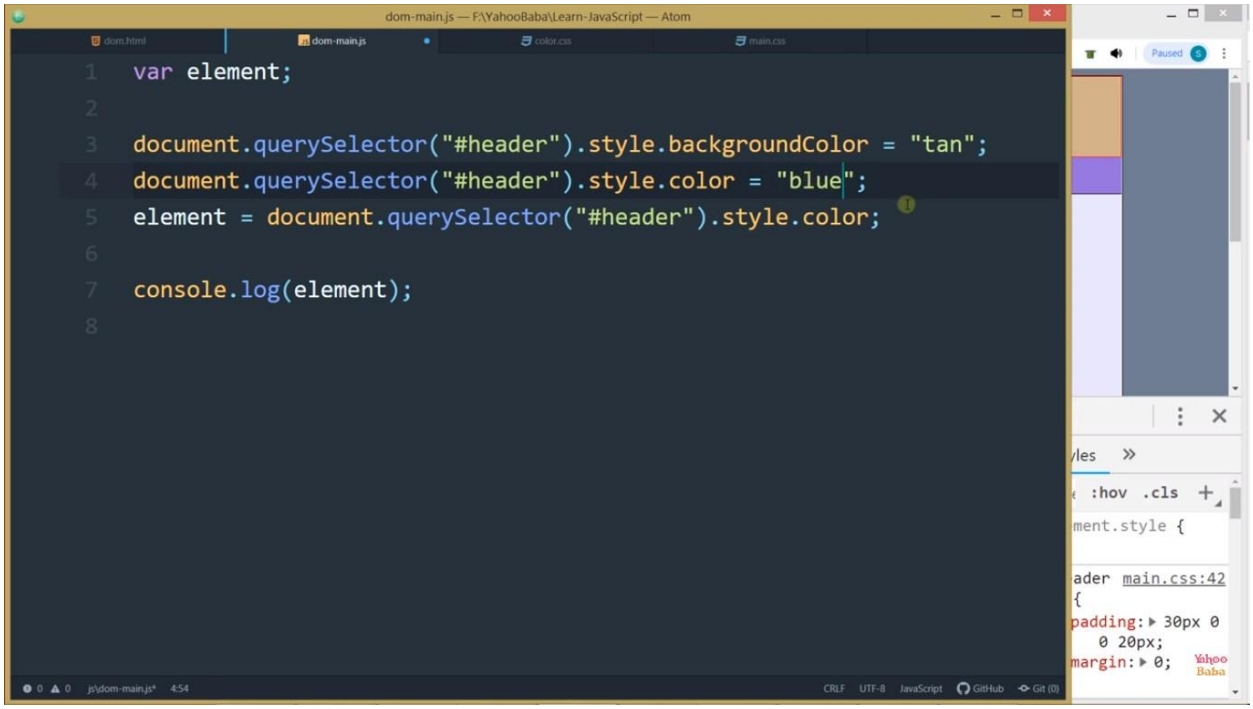

dom-main.js — F:\YahooBaba\Learn-JavaScript — Atom
dom.html
dom-main.js
var element;
document.querySelector("#header").innerHTML = "<h1>Wow</h1>";
element = document.querySelectorAll(".list");
console.log(element);
Code Actions
Select an action
Extract to function in global scope
Extract to constant in enclosing scope
ome
bout Us
allery
ontact Us
2 hidden
dom-main.js:7
dom.html:74
Yahoo
Baba

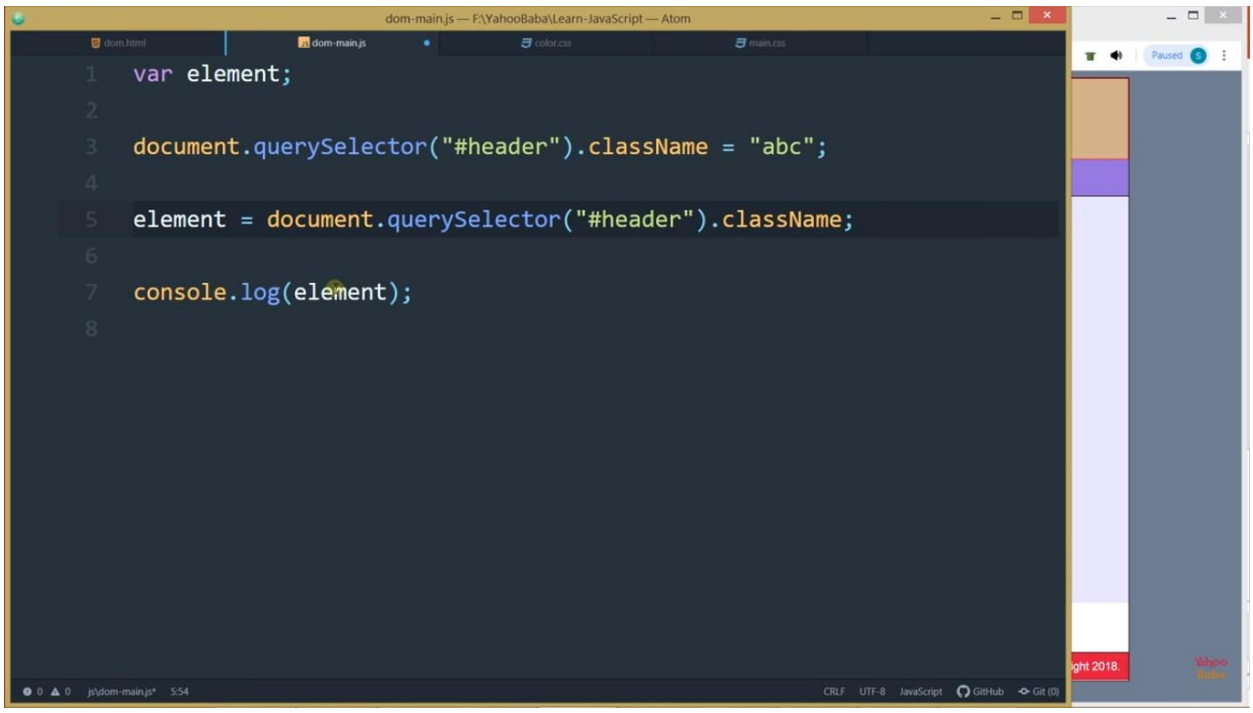

dom-main.js — F:\YahooBaba\Learn-JavaScript — Atom
dom.html
dom-main.js
var element;
document.querySelector("#header").innerHTML = "<h1>Wow</h1>";
element = document.querySelectorAll(".list")[1].innerHTML;
console.log(element);
ome
bout Us
allery
ontact Us
2 hidden
dom-main.js:6
. Ipsa, quam.
dom.html:74
Yahoo
Baba

DOM CSS Styling Methods :

- Style
- className
- classList

Yahoo Baba





JavaScript Basic Events

- Click (onclick)
- Double Click (ondblclick)
- Right Click (oncontextmenu)
- Mouse Hover (onmouseenter)
- Mouse Out (onmouseout)
- Mouse Down (onmousedown)
- Mouse Up (onmouseup)
- Key Press (onkeypress)
- Key Up (onkeyup)
- Load (onload)
- Unload (onunload)
- Resize (onresize)
- Scroll (onscroll)

Yahoo Baba

HTML Event Attributes :

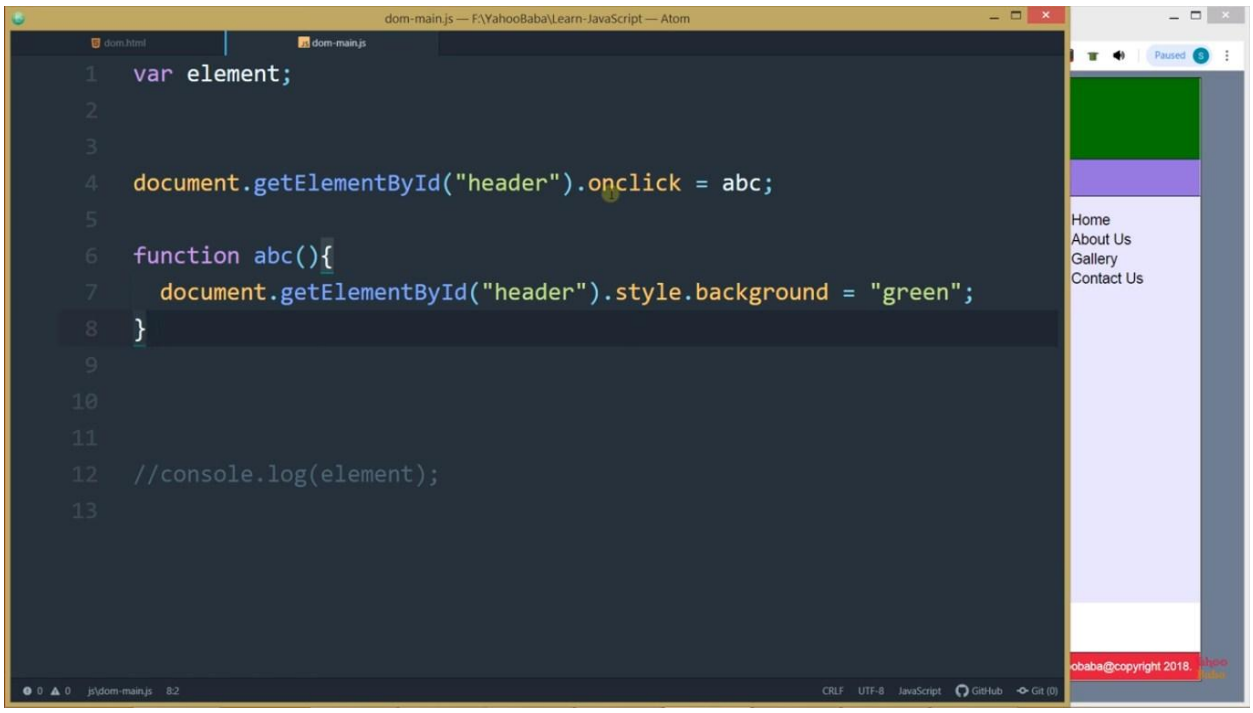
```
<button onClick="abc()"></button>

<img src=".." onmouseenter="xyz()">
```

Yahoo Baba

Assign Events Using the HTML DOM :

```
document.getElementById(Id).onclick = functionName;
```

Yahoo Baba



DOM addEventListener() Method :

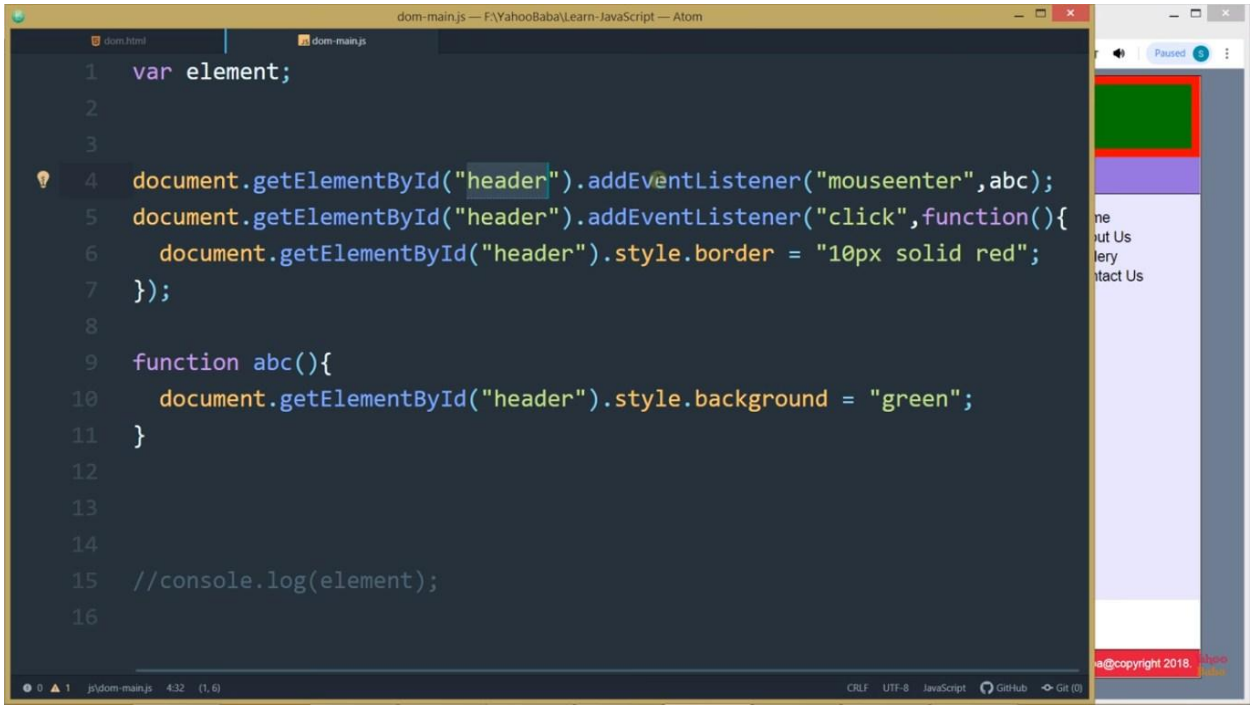
```
document.getElementById(Id). addEventListener("click", functionName);
```

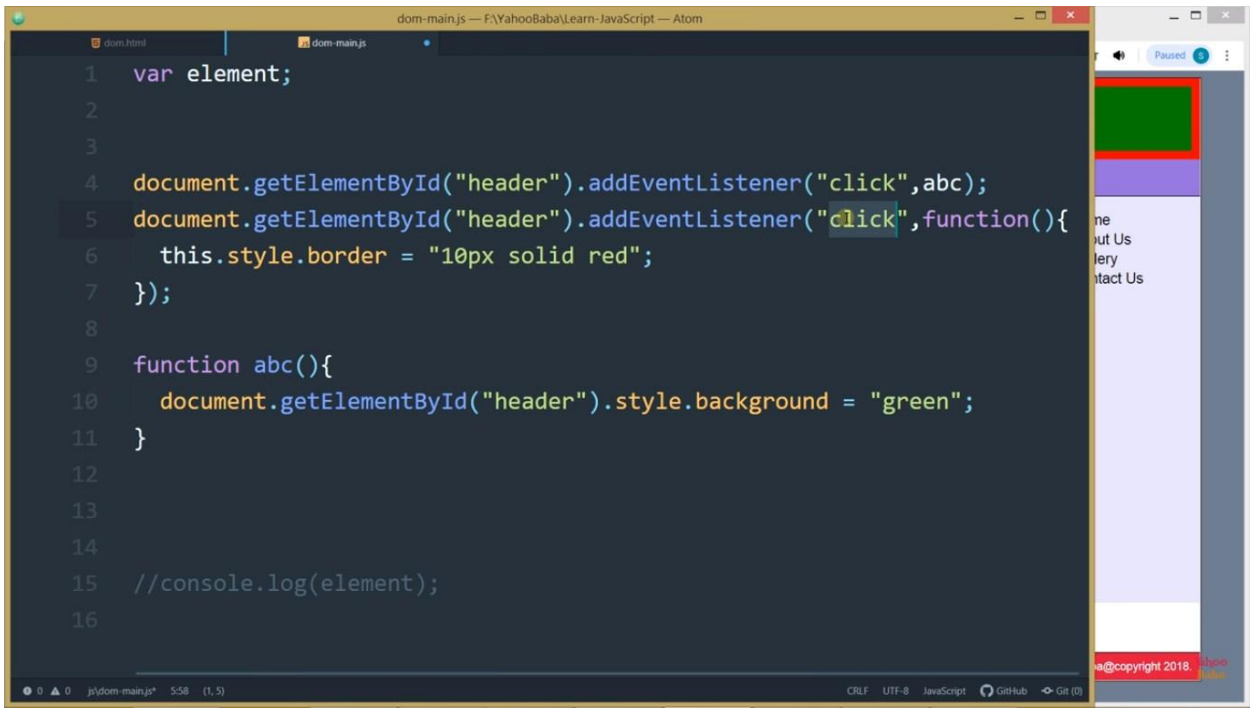
Yahoo Baba

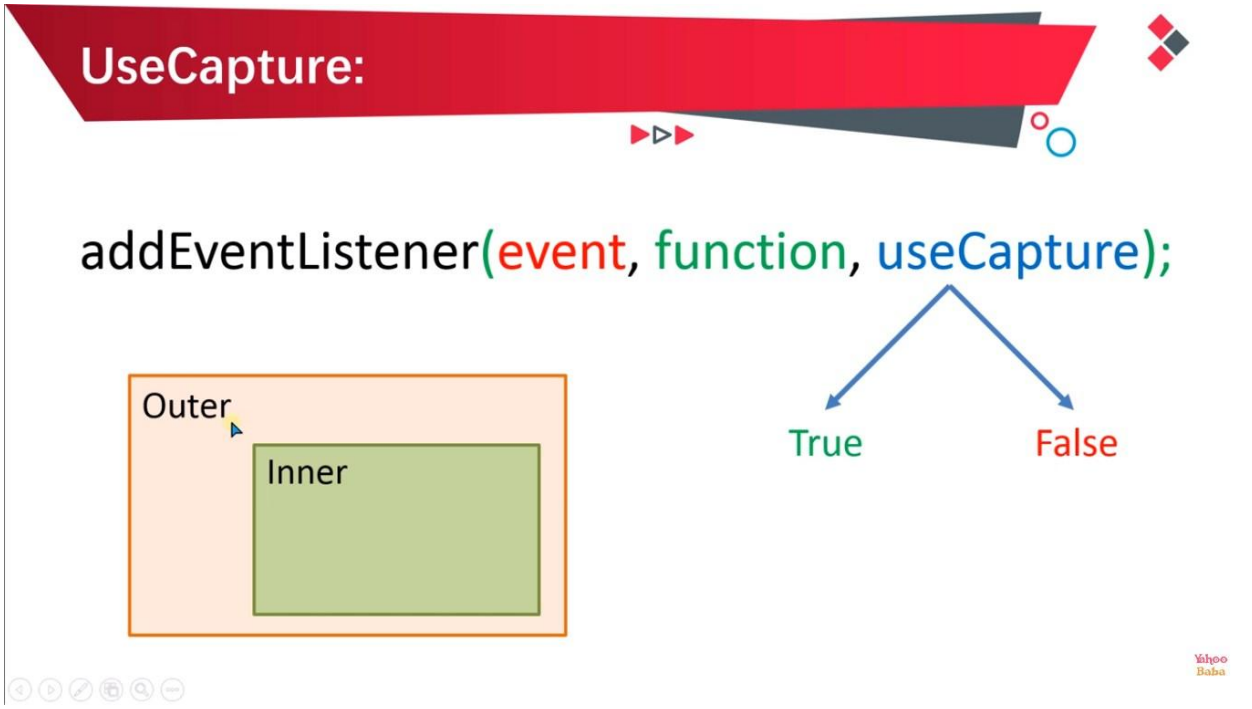
DOM addEventListener() Method :

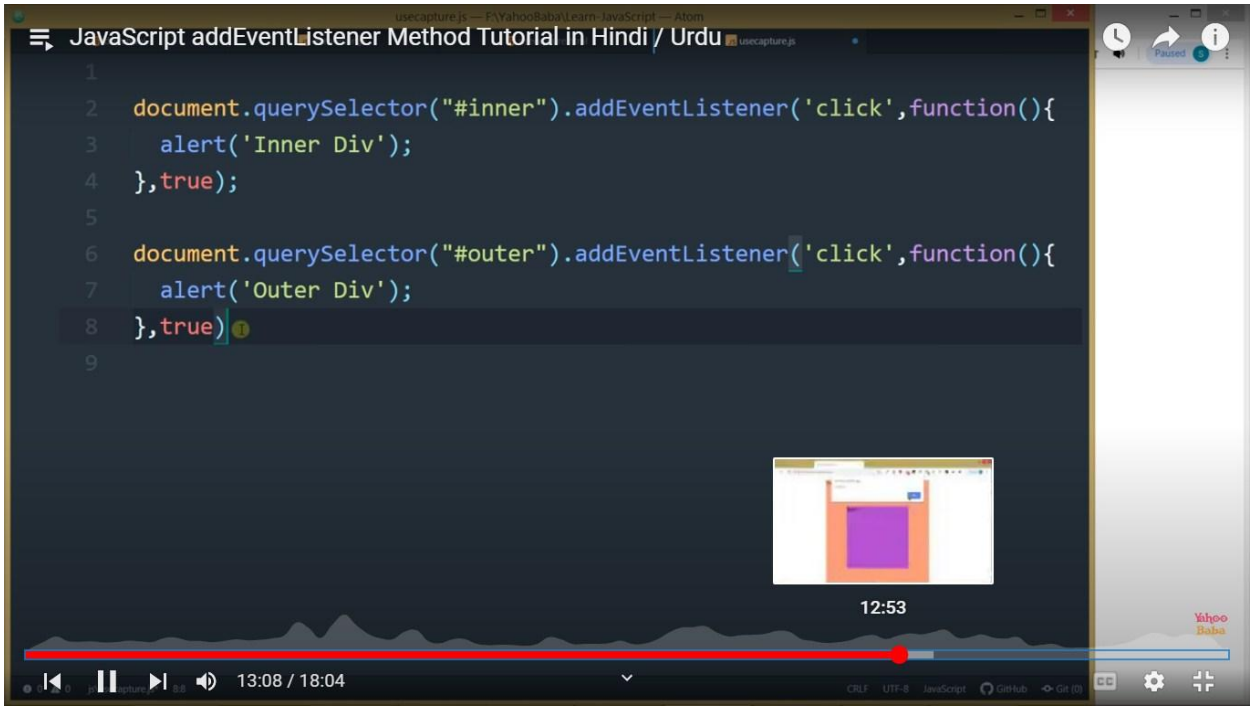
```
document.getElementById(Id). addEventListener("click", functionName);

document.getElementById(Id). addEventListener("click", function(){
        Statement
});
```

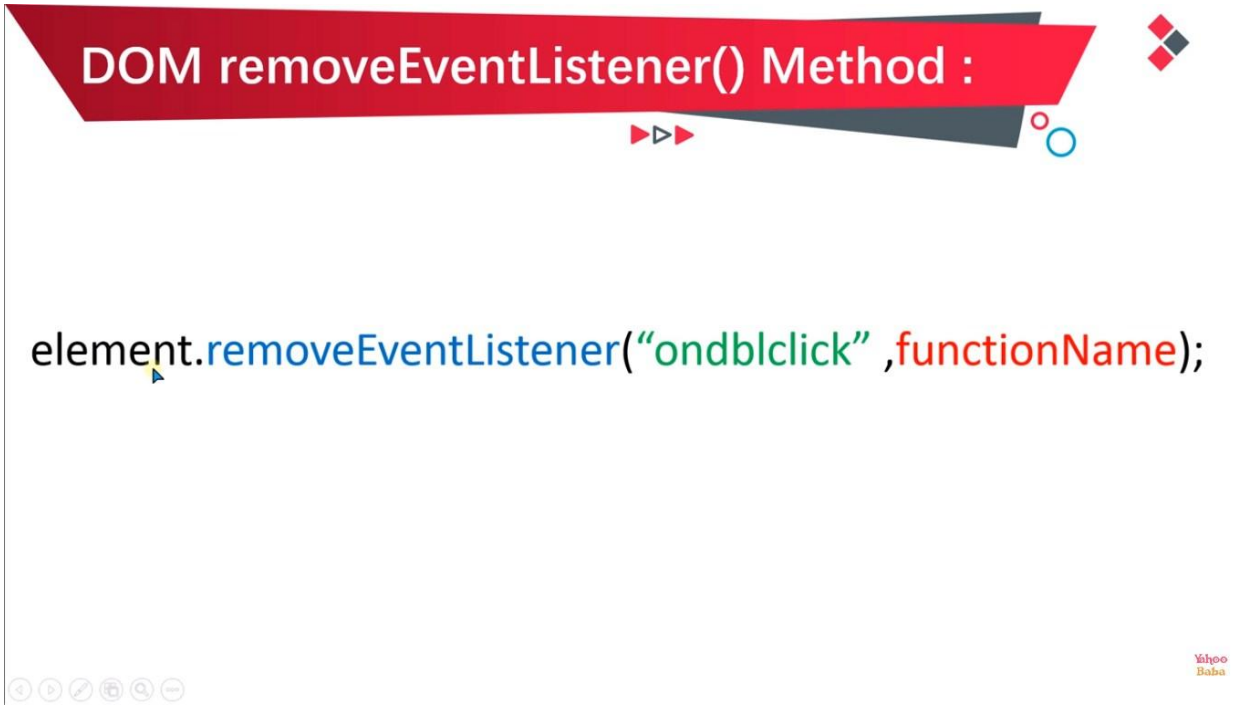
Yahoo Baba

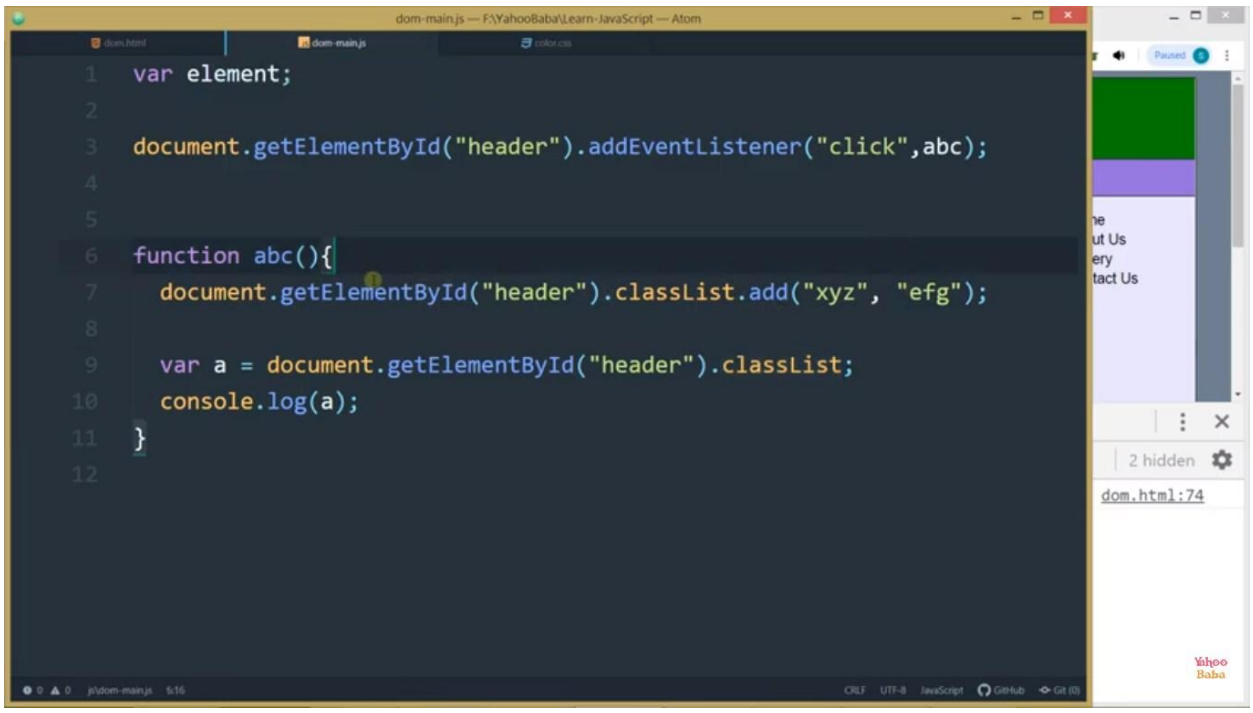


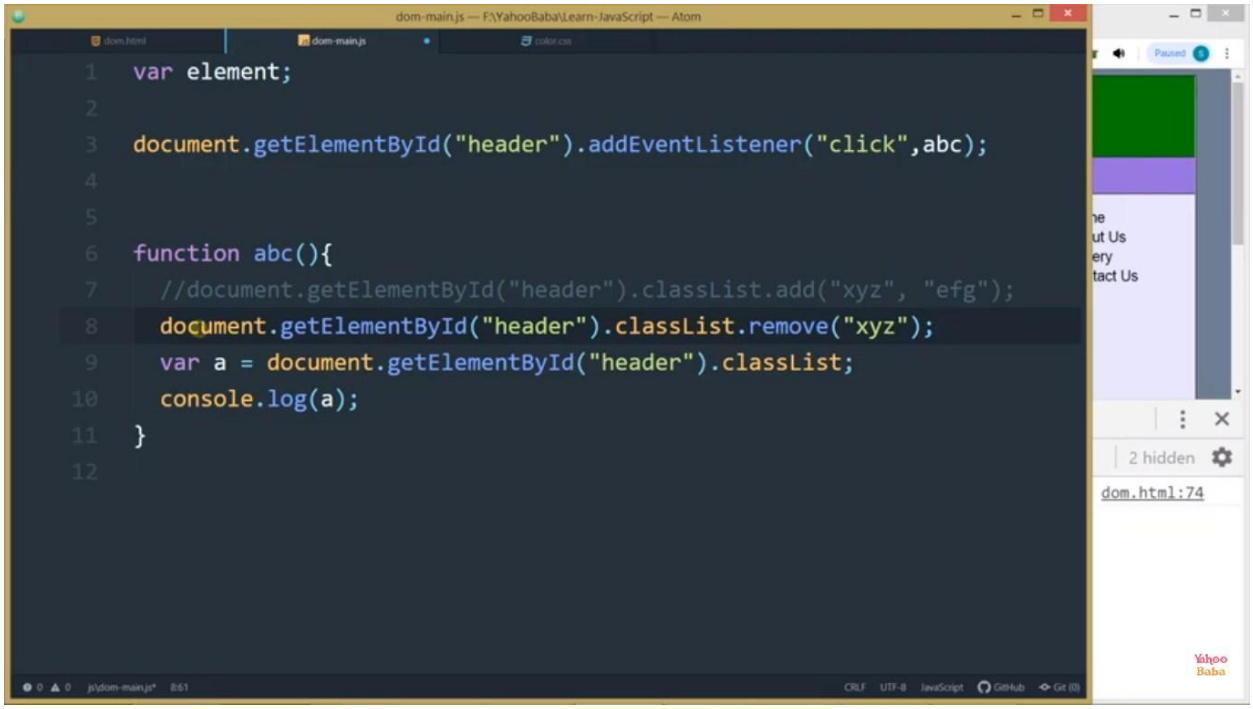

dom-main.js — F:\YahooBaba\Learn-JavaScript — Atom
dom.html
dom-main.js
1 var element;
2
3
4 document.getElementById("header").addEventListener("click",abc);
5 document.getElementById("header").addEventListener("click",function(){
6 this.style.border = "10px solid red";
7 });
8
9 function abc(){
10 document.getElementById("header").style.background = "green";
11 }
12
13
14
15 //console.log(element);
16
Paused
ne
ut Us
lery
ntact Us
@copyright 2018.
CRLF UTF-8 JavaScript GitHub Git (0)


UseCapture:
addEventListener(event, function, useCapture);
Outer
Inner
True
False
Yahoo
Baba


JavaScript addEventListener Method Tutorial in Hindi / Urdu
document.querySelector("#inner").addEventListener('click',function(){
alert('Inner Div');
},true);
document.querySelector("#outer").addEventListener('click',function(){
alert('Outer Div');
},true)
12:53
13:08 / 18:04

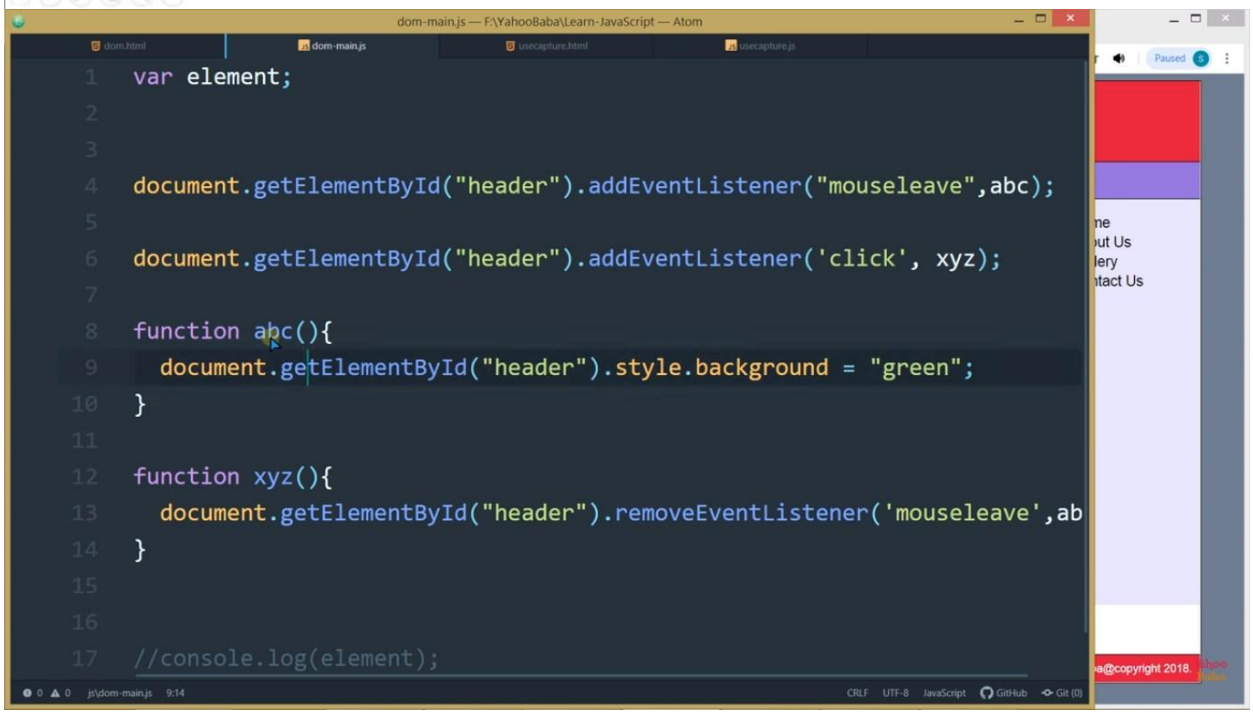

DOM removeEventListener() Method :
element.removeEventListener("ondblclick" ,functionName);


dom-main.js — F:\YahooBaba\Learn-JavaScript — Atom
dom.html
dom-main.js
color.css
var element;
document.getElementById("header").addEventListener("click",abc);
function abc(){
document.getElementById("header").classList.add("xyz", "efg");
var a = document.getElementById("header").classList;
console.log(a);
}
Paused
2 hidden
dom.html:74
Yahoo Baba


dom-main.js — F:\YahooBaba\Learn-JavaScript — Atom
dom.html
dom-main.js
color.css
var element;
document.getElementById("header").addEventListener("click",abc);
function abc(){
//document.getElementById("header").classList.add("xyz", "efg");
document.getElementById("header").classList.remove("xyz");
var a = document.getElementById("header").classList;
console.log(a);
}
Paused
2 hidden
dom.html:74
Yahoo Baba

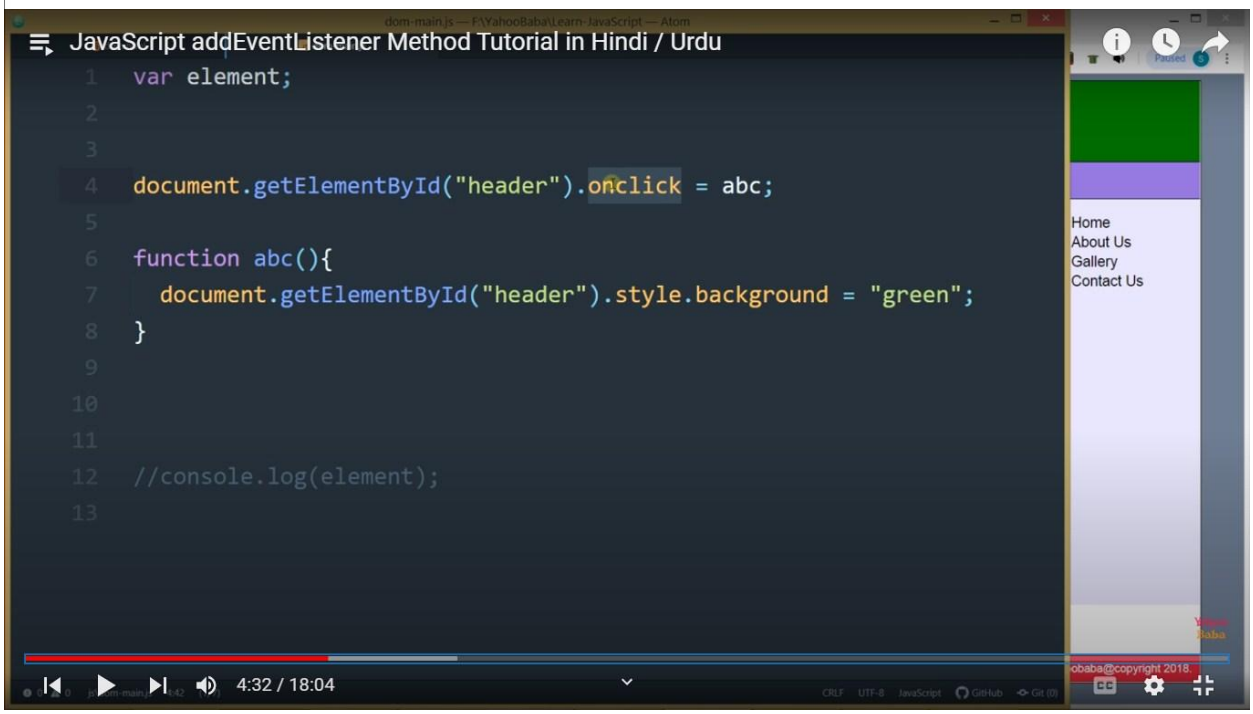
JavaScript classList Methods :

- add(class1, class2, ...)
- remove(class1, class2, ...)
- toggle(class)
- contains(class)
- item(index)
- Length



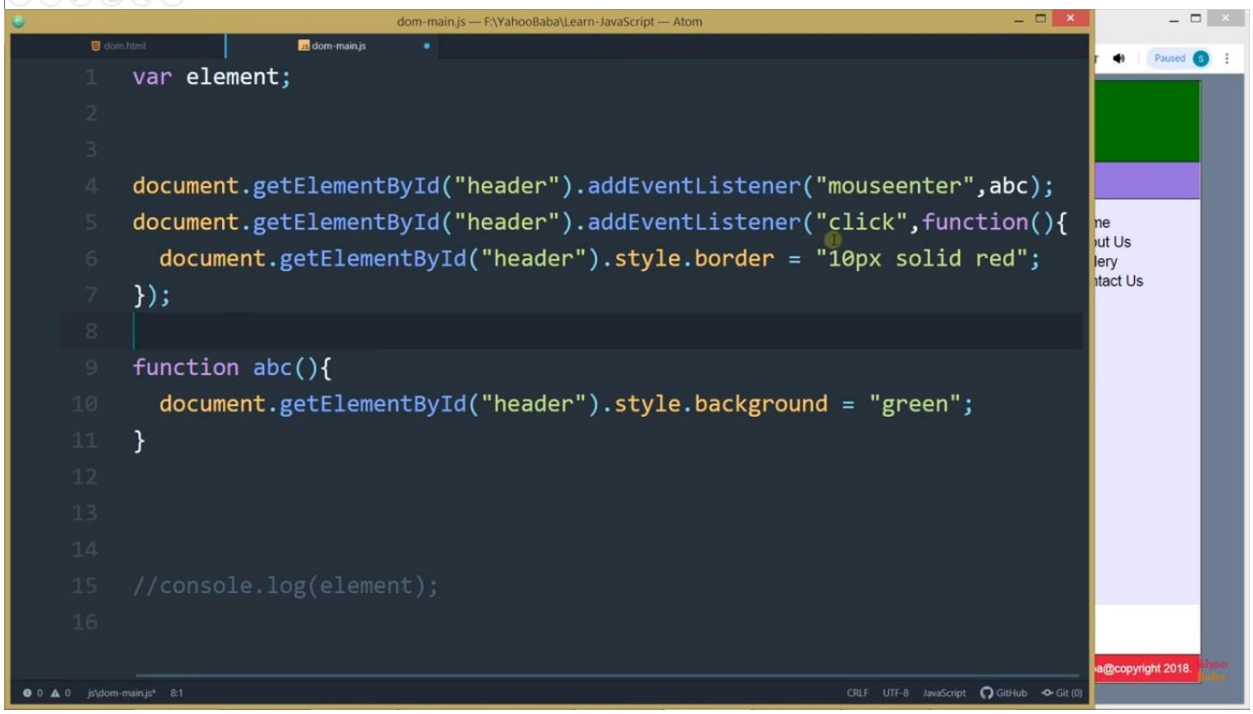
HTML Event Attributes :

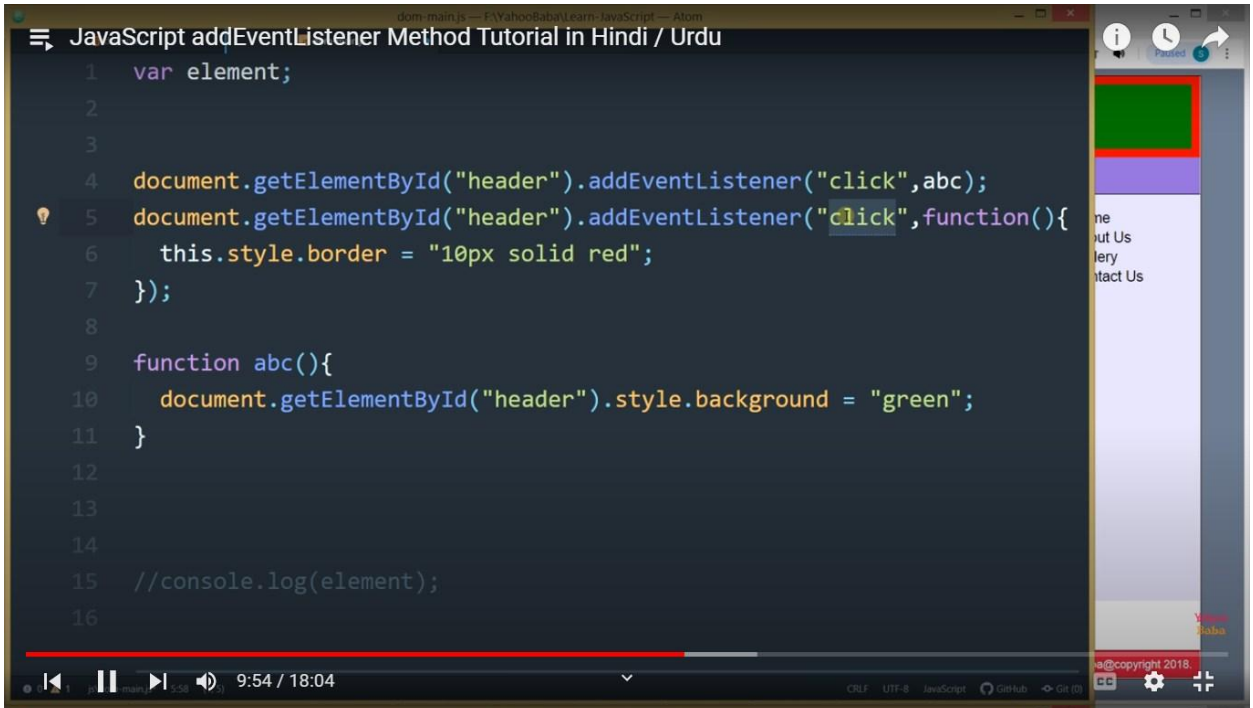
```
<button onClick="abc()"></button>

<img src=".." onmouseenter="xyz()">
```

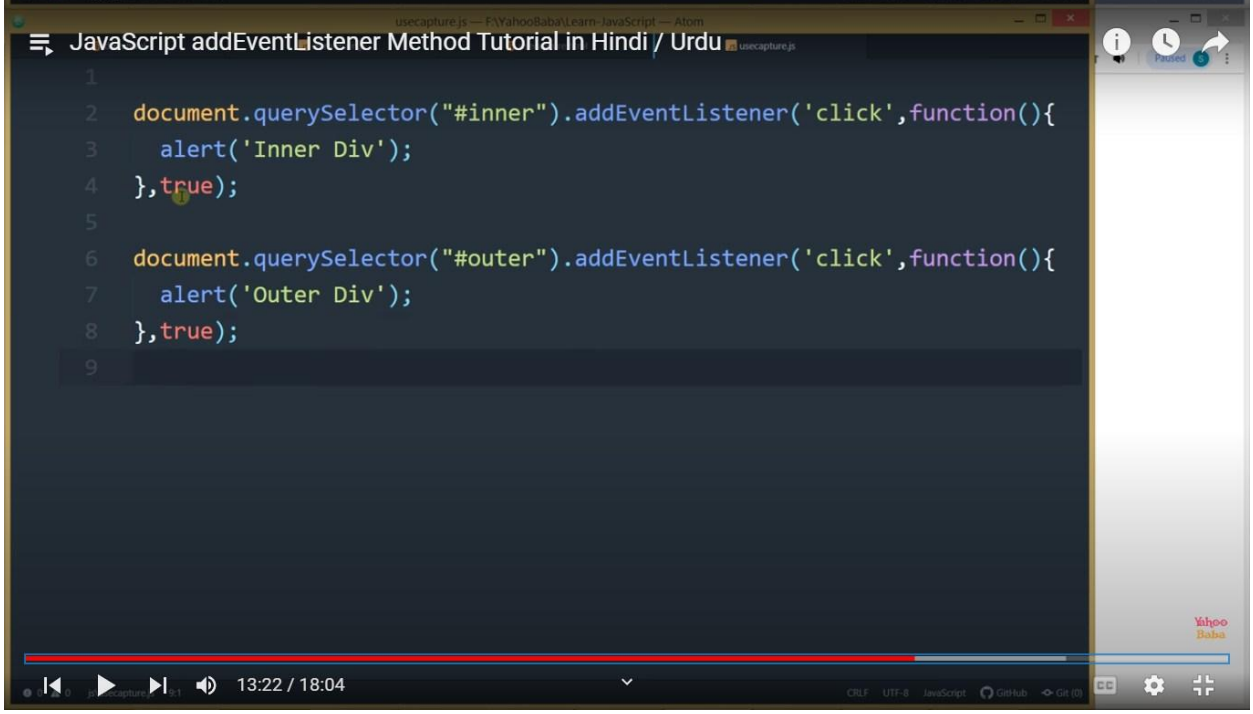


DOM addEventListener() Method :

```
document.getElementById(Id). addEventListener("click", functionName);

document.getElementById(Id). addEventListener("click", function(){

});
```



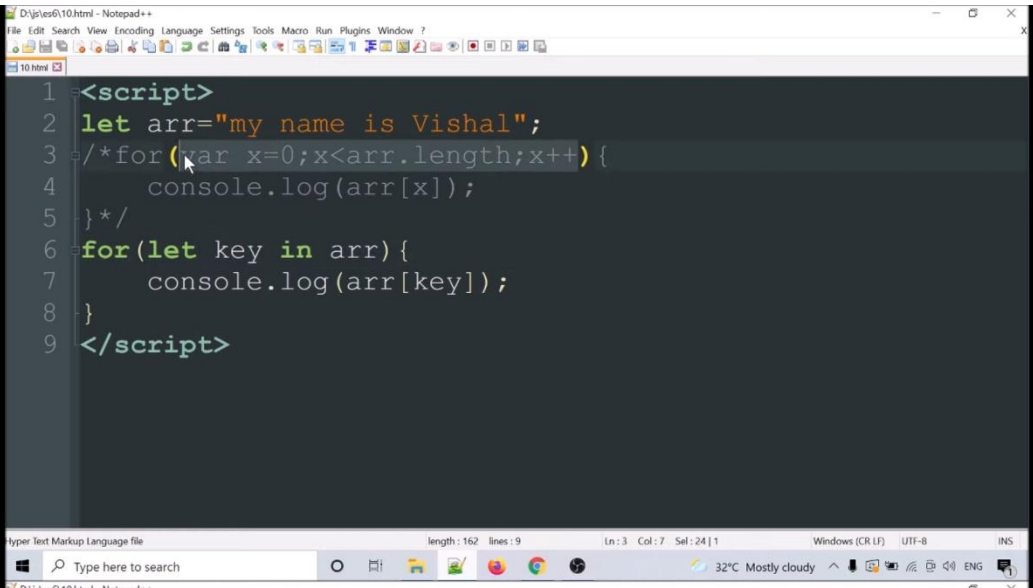

JavaScript addEventListener Method Tutorial in Hindi / Urdu
var element;
document.getElementById("header").addEventListener("click",abc);
document.getElementById("header").addEventListener("click",function(){
this.style.border = "10px solid red";
});
function abc(){
document.getElementById("header").style.background = "green";
}
//console.log(element);
9:54 / 18:04

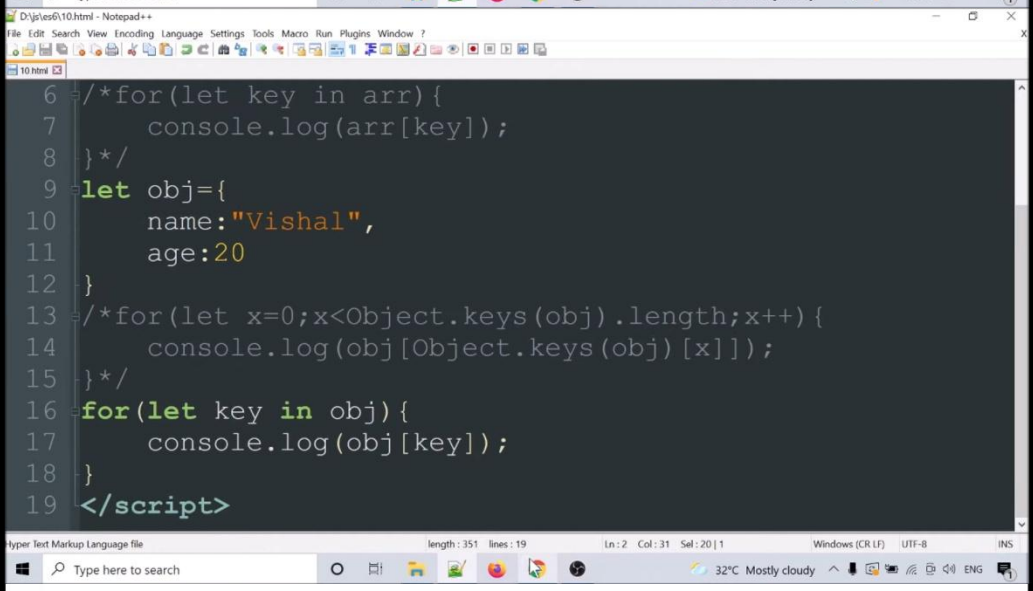

JavaScript addEventListener Method Tutorial in Hindi / Urdu
document.querySelector("#inner").addEventListener('click',function(){
alert('Inner Div');
},true);
document.querySelector("#outer").addEventListener('click',function(){
alert('Outer Div');
},true);
13:22 / 18:04

DOM removeEventListener() Method :

```
element.removeEventListener("ondblclick" ,functionName);
```

Yahoo Baba





1. Encapsulation: -

Encapsulation means wrapping up data and member function (Method) together into a single unit i.e. class.

2. Abstraction: -

Abstraction is the process of showing only essential/necessary features of an entity/object to the outside world and hide the other irrelevant information. For example to open your TV we only have a power button, It is not required to understand how infra-red waves are getting generated in TV remote control.

3. Inheritance: -

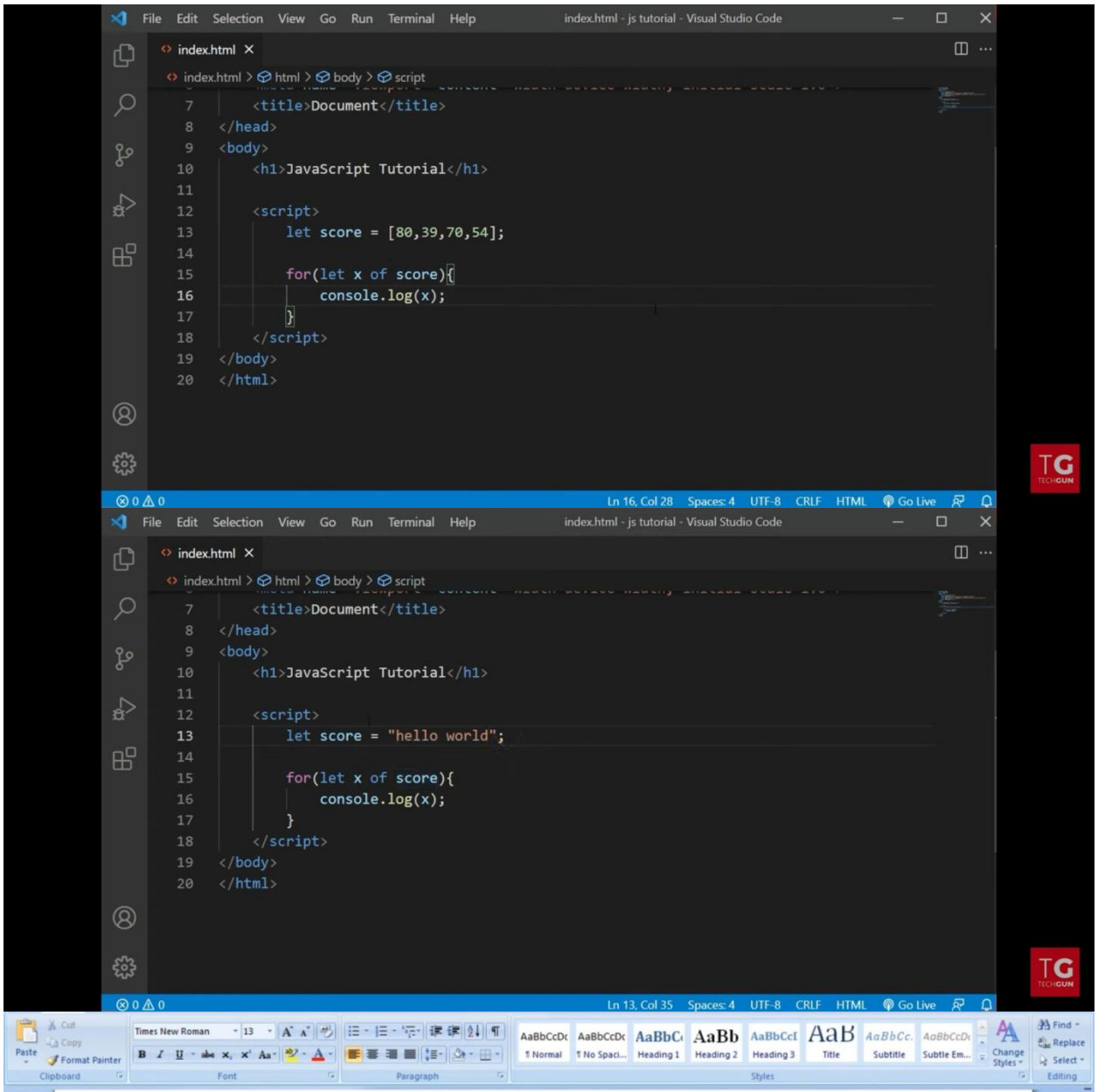
Inheritance allows a class (subclass) to acquire the properties and behavior of another class (super-class). It helps to reuse, customize and enhance the existing code. So it helps to write a code accurately and reduce the development time.

4. Polymorphism: -

So polymorphism means "many forms". A subclass can define its own unique behavior and still share the same functionalities or behavior of its parent/base class.

```
class square(){
  area()
}

Var S1 =  new square();
S1->area();
```

```
class circle(){
  area()
}

Var C1 =  new circle();
C1->area();
```



JavaScript Type Casting

Converting one type of data value to another type is called as type casting.

There are **two** types of type casting:
Implicit type casting (Coercion): Done by JavaScript Engine
Explicit type casting (Conversion): Done by programmers

Implicit type casting:
If required JavaScript engine automatically converts one type of value to another type.
This is known as implicit type casting.
Ex:

```
document.write(2+4.3);     // 6.3
document.write( "2" + 2 );   // "22"
document.write(2 + "2");    // "22"
```

```
<script>

const myFavProg = ['JavaScript', 'PHP', 'Java', 'c', 'Python'];

// for of iterable  objects...
iterate

arrays, strings

for(items of myFavProg){
    console.log(items);
}

// console.log(myFavProg[0]);
// console.log(myFavProg[1]);
// console.log(myFavProg[2]);

// for (let x=0; x<myFavProg.length; x++){
```

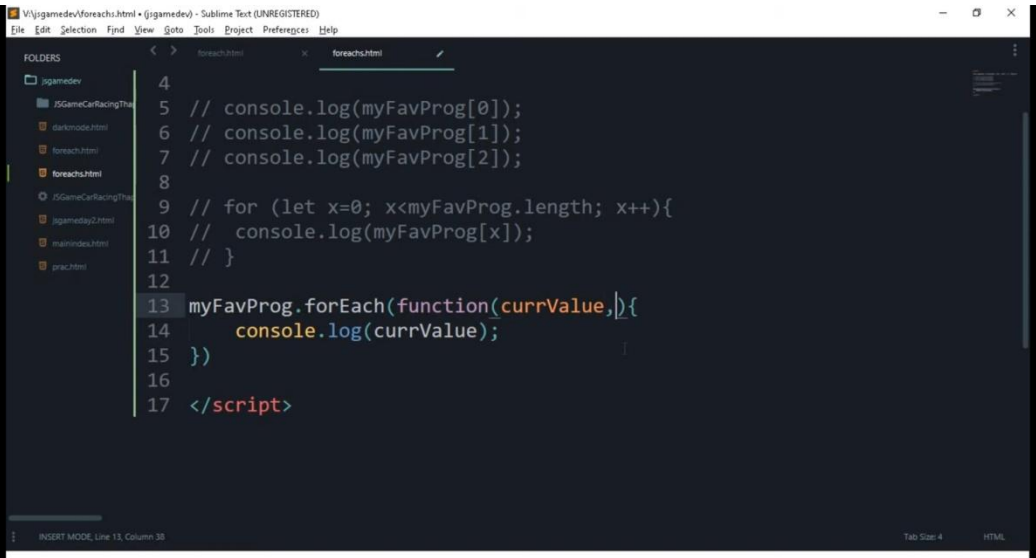
```
    <title>Document</title>
</head>
<body>
    <h1>JavaScript Tutorial</h1>

    <script>
        let score = [80,39,70,54];

        for(let x of score){
            console.log(x);
        }
    </script>
</body>
</html>
```

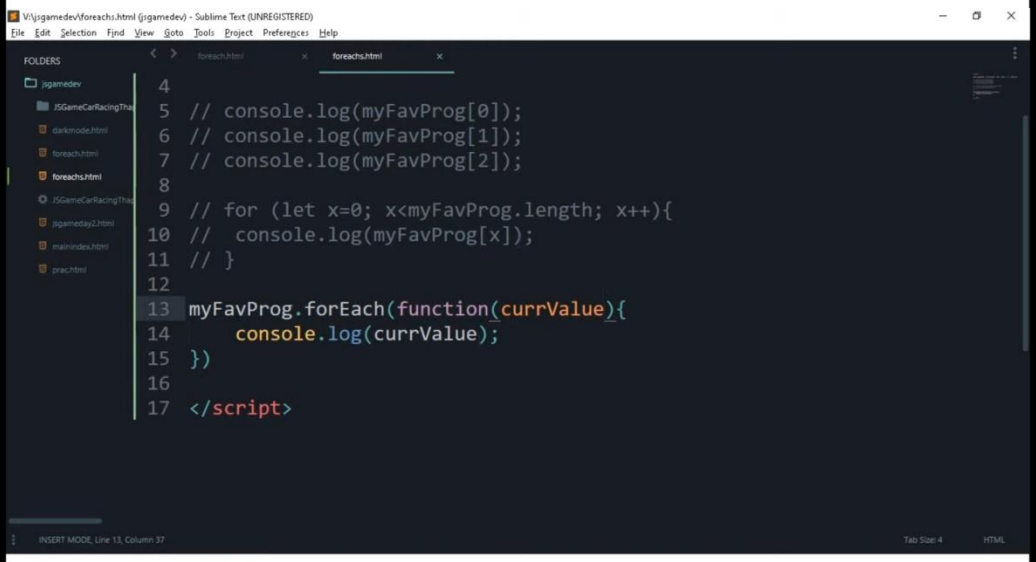
The forEach() method calls a function once for each element in an array, in order.

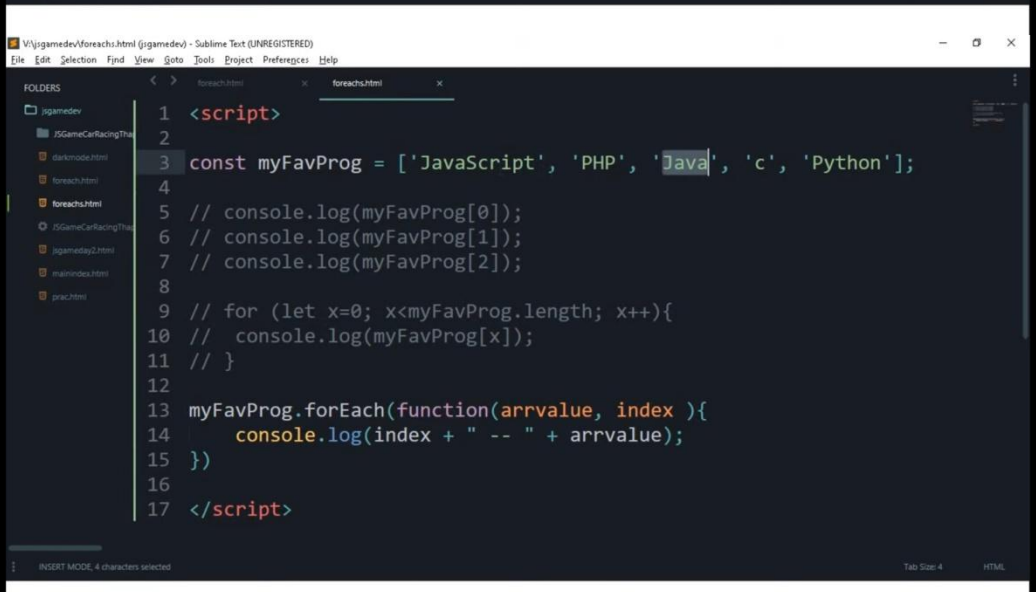
forEach() Method

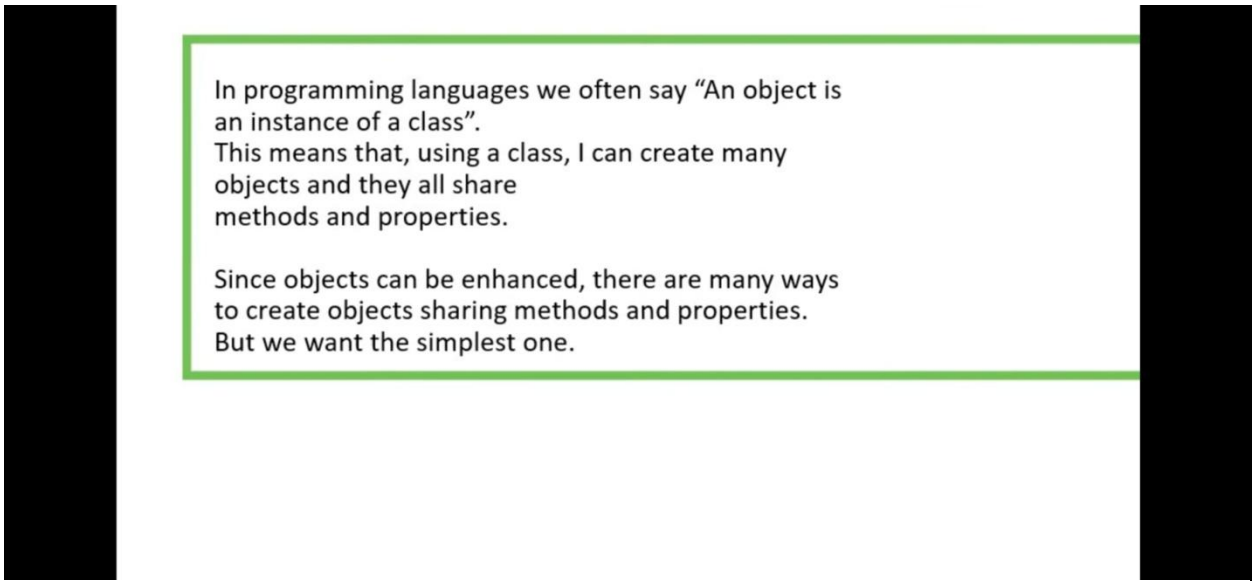
Syntax: arr.forEach(callback(currentValue [, index [, array]])[, thisArg])

Argument	Description
currentValue	Required. The value of the current element
index	Optional. The array index of the current element
arr	Optional. The array object the current element belongs to


V:\jsgamedev\foreachs.html • (jsgamedev) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
jsgamedev
foreach.html
foreachs.html
// console.log(myFavProg[0]);
// console.log(myFavProg[1]);
// console.log(myFavProg[2]);
// for (let x=0; x<myFavProg.length; x++){
// console.log(myFavProg[x]);
// }
myFavProg.forEach(function(currValue,){
console.log(currValue);
})
</script>
INSERT MODE, Line 13, Column 38
Tab Size: 4
HTML


V:\jsgamedev\foreachs.html (jsgamedev) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
jsgamedev
foreach.html
foreachs.html
// console.log(myFavProg[0]);
// console.log(myFavProg[1]);
// console.log(myFavProg[2]);
// for (let x=0; x<myFavProg.length; x++){
// console.log(myFavProg[x]);
// }
myFavProg.forEach(function(currValue){
console.log(currValue);
})
</script>
INSERT MODE, Line 13, Column 37
Tab Size: 4
HTML


V:\jsgamedev\foreachs.html (jsgamedev) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
jsgamedev
foreach.html
foreachs.html
<script>
const myFavProg = ['JavaScript', 'PHP', 'Java', 'c', 'Python'];
// console.log(myFavProg[0]);
// console.log(myFavProg[1]);
// console.log(myFavProg[2]);
// for (let x=0; x<myFavProg.length; x++){
// console.log(myFavProg[x]);
// }
myFavProg.forEach(function(arrvalue, index){
console.log(index + " -- " + arrvalue);
})
</script>
INSERT MODE, 4 characters selected
Tab Size: 4
HTML



Bind `this`

For methods in React, the `this` keyword should represent the component that owns the method.

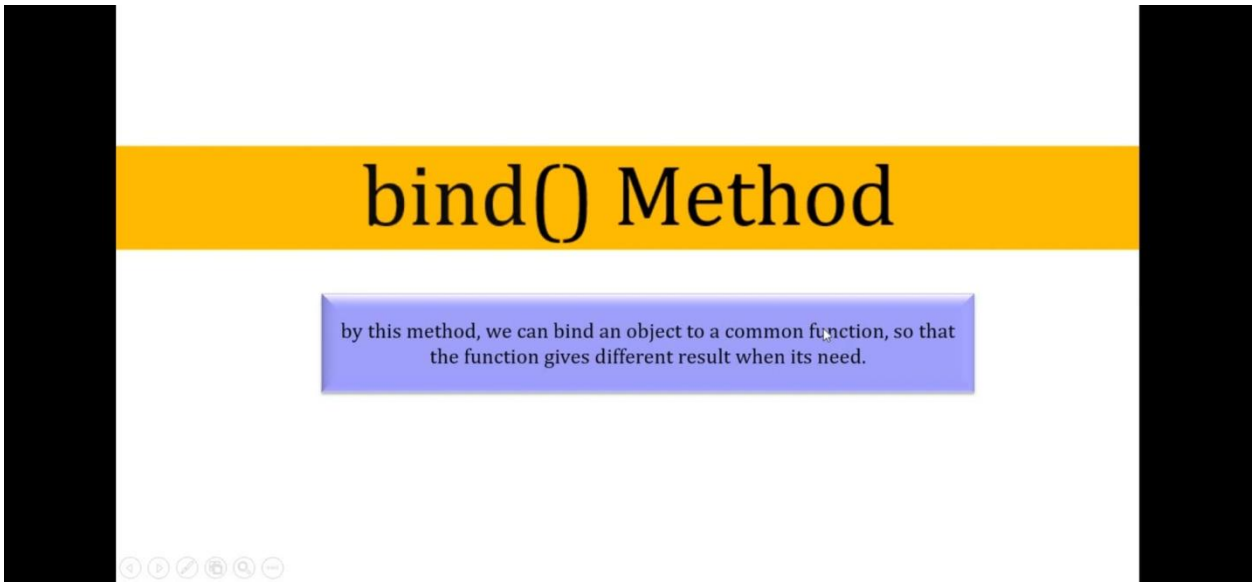
That is why you should use arrow functions. With arrow functions, `this` will always represent the object that defined the arrow function.

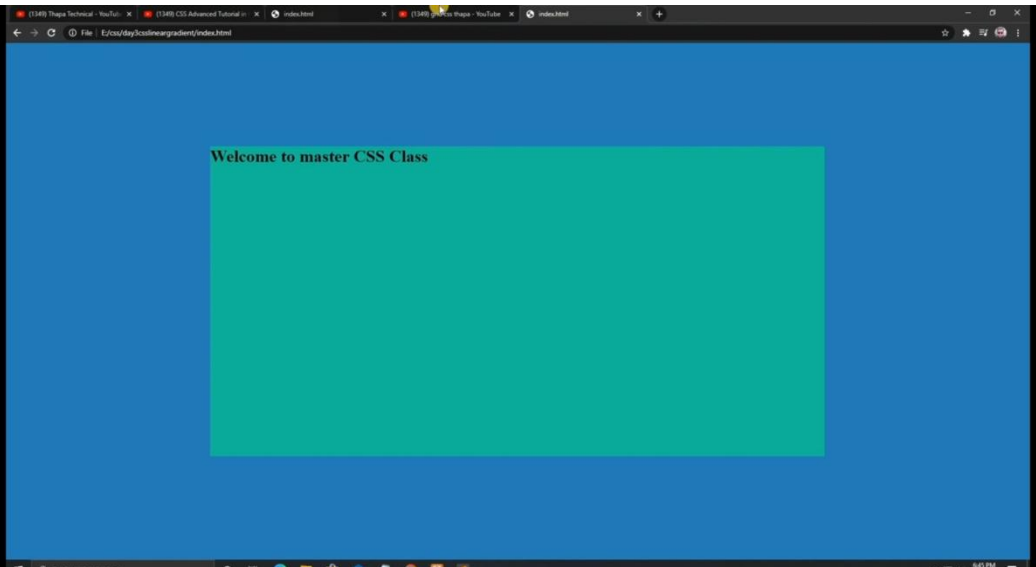
Why Arrow Functions?

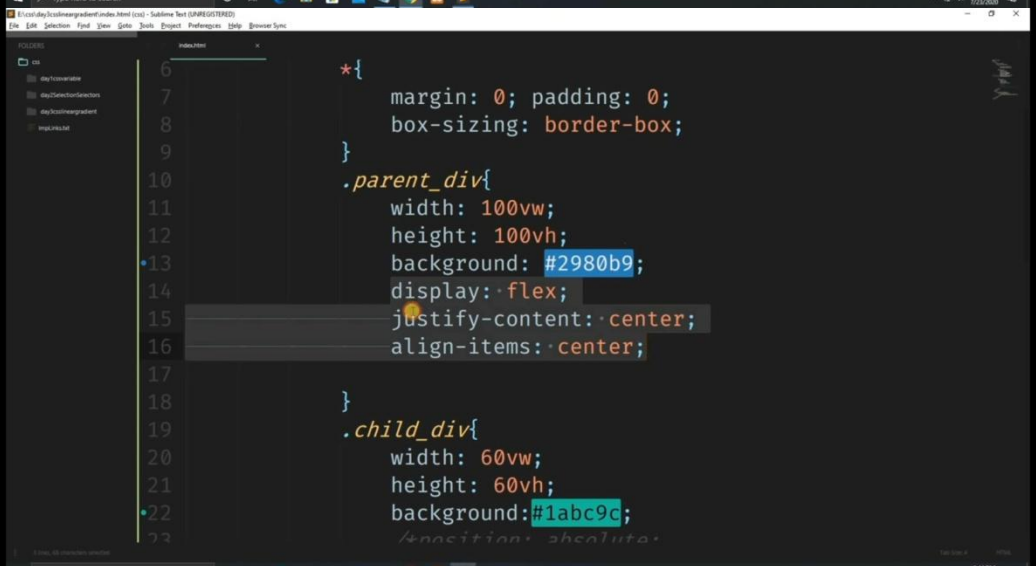
In class components, the `this` keyword is not defined by default, so with regular functions the `this` keyword represents the object that called the method, which can be the global window object, a HTML button, or whatever.

Read more about binding `this` in our React ES6 'What About this?' chapter.

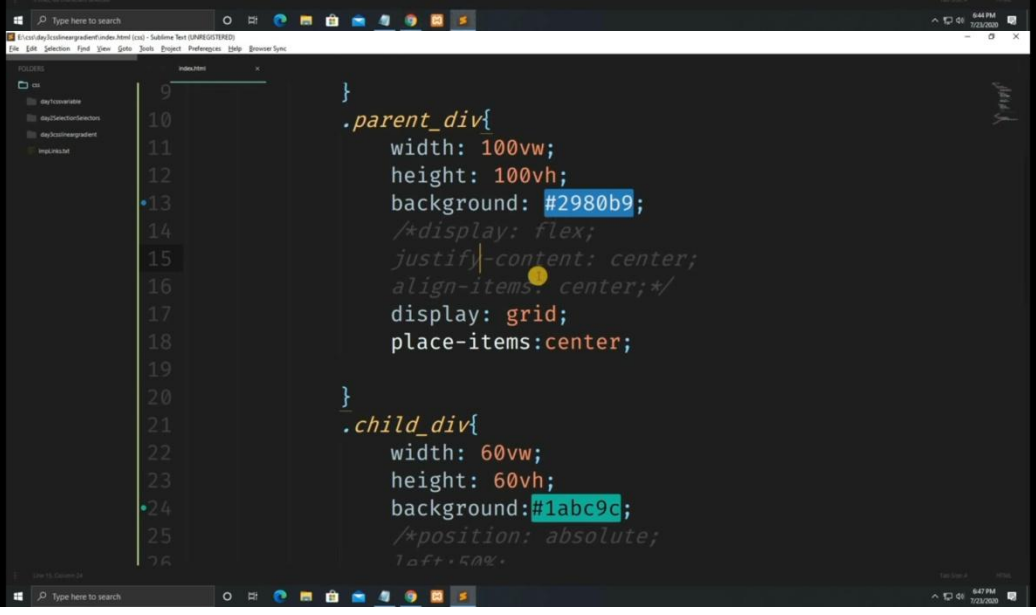
If you *must* use regular functions instead of arrow functions you have to bind `this` to the component instance using the `bind()` method:







```
*{
    margin: 0; padding: 0;
    box-sizing: border-box;
}
.parent_div{
    width: 100vw;
    height: 100vh;
    background: #2980b9;
    display: flex;
    justify-content: center;
    align-items: center;

}
.child_div{
    width: 60vw;
    height: 60vh;
    background:#1abc9c;
```

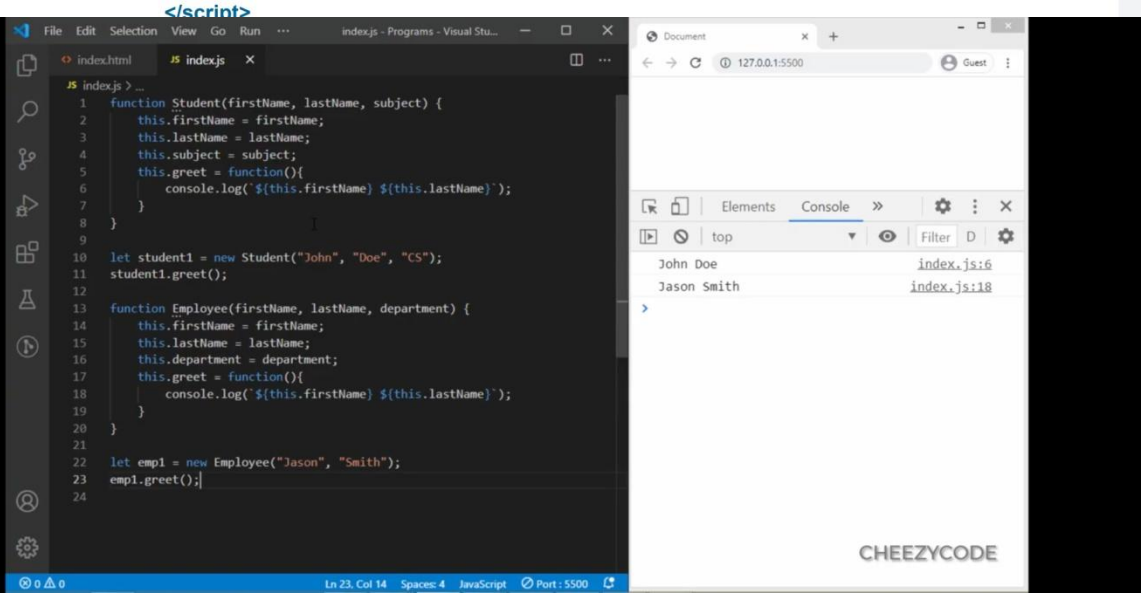


```
}
.parent_div{
    width: 100vw;
    height: 100vh;
    background: #2980b9;
    /*display: flex;
    justify-content: center;
    align-items: center;*/
    display: grid;
    place-items:center;

}
.child_div{
    width: 60vw;
    height: 60vh;
    background:#1abc9c;
    /*position: absolute;
```

```
<script>
        function Employee(firstName,lastName)
        {
          this.firstName=firstName;
          this.lastName=lastName;
        }

        Employee.prototype.company="Javatpoint"

        var employee1=new Employee("Martin","Roy");
        var employee2=new Employee("Duke", "William");
        document.writeln(employee1.firstName+" "+employee1.lastName+" "+employee1.company+"<b
        r>");
        document.writeln(employee2.firstName+" "+employee2.lastName+" "+employee2.company);
</script>
```

Let's see an example to add a new property to the constructor function.

```
<script>
        function Employee(firstName,lastName)
        {
          this.firstName=firstName;
          this.lastName=lastName;
        }

        Employee.prototype.company="Javatpoint"

        var employee1=new Employee("Martin","Roy");
        var employee2=new Employee("Duke", "William");
        document.writeln(employee1.firstName+" "+employee1.lastName+" "+employee1.company+"<b
        r>");
        document.writeln(employee2.firstName+" "+employee2.lastName+" "+employee2.company);
</script>
```



```
const person = {
  firstName:"John",
  lastName: "Doe",
  fullName: function () {
    return this.firstName + " " + this.lastName;
  }
}

console.log( person.fullName() ); // John Doe
```

JS


```
const person = {
  fullName: function() {
    return this.firstName + " " + this.lastName;
  }
}

const person1 = {
  firstName:"John",
  lastName: "Doe"
}
const person2 = {
  firstName:"Mary",
  lastName: "Doe"
}

console.log( person.fullName.call(person1) ); // John Doe
console.log( person.fullName.call(person2) ); // Mary Doe
```

JS


```
const obj = { name: "John" };

let greeting = function(a,b){
    return `${a} ${this.name}. ${b}`;
};

console.log( greeting.call(obj, "Hello", "How are you?") );
// returns: Hello John. How are you?
```

JS

```js
const obj = { name: "John" };

let greeting = function(a,b){
    return `${a} ${this.name}. ${b}`;
};

console.log( greeting.apply(obj, ["Hello", "How are you?"]) );
// returns: Hello John. How are you?
```

JS

SUBSCRIBE

```js
const obj = { name: "John" };

let greeting = function(a,b){
    return `${a} ${this.name}. ${b}`;
}

let bound = greeting.bind(obj);

console.log( bound("Hello", "How are you?") );
// returns: Hello John. How are you?
```

JS

Array – findIndex() Function :

var ages = [10 , 23 , 19 , 20];

var adultAge = 18;

ages >= adultAge

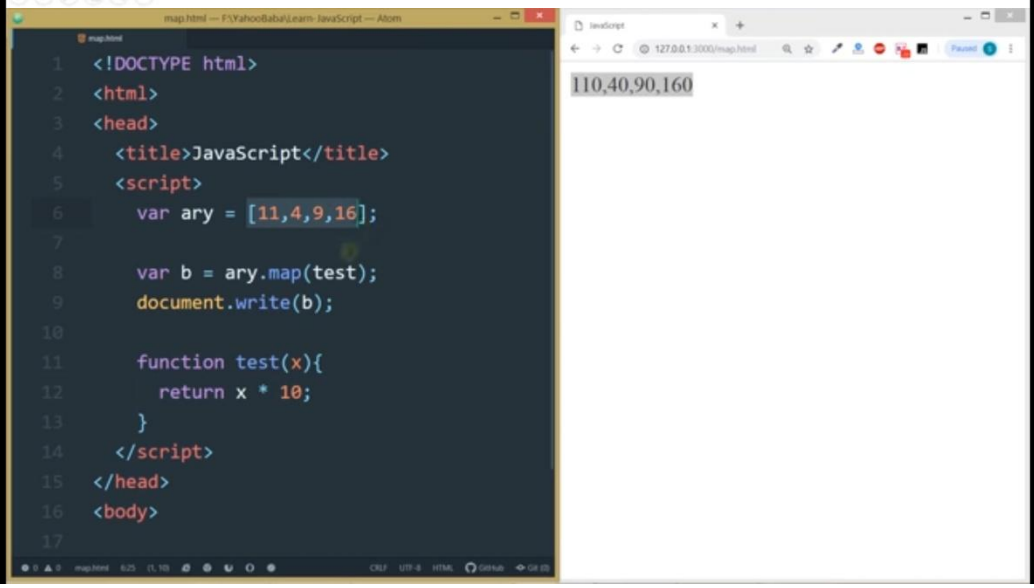
findIndex(Function Name)

findIndex() method returns the index of the first element in an array that pass a test

Array – find() Function :

```
var ages = [10 , 23 , 19 , 20];
var adultAge = 18;
ages >= adultAge
find(Function Name)
```

find() method returns the value of the first element in an array that pass a test



```
<!DOCTYPE html>
<html>
<head>
  <title>JavaScript</title>
  <script>
    var ary = [11,4,9,16];

    var b = ary.map(test);
    document.write(b);

    function test(x){
      return x * 10;
    }
  </script>
</head>
<body>
```

```
110,40,90,160
```

Yahoo Baba

JavaScript Interview Questions

JavaScript interview questions and answers for provides a list of top 20 interview questions. The frequently asked JavaScript interview questions with answers for beginners and professionals are given below.

1) What is JavaScript?

JavaScript is *a scripting language*. It is different from Java language. It is object-based, lightweight, cross-platform translated language. It is widely used for client-side validation. The JavaScript Translator (embedded in the browser) is responsible for translating the JavaScript code for the web browser. More details.

---

2) List some features of JavaScript.

Some of the features of JavaScript are:

- Lightweight
- Interpreted programming language
- Good for the applications which are network-centric
- Complementary to Java
- Complementary to HTML
- Open source

  Cross-platform

---

3) Who developed JavaScript, and what was the first name of JavaScript?

JavaScript was developed by Brendan Eich, who was a Netscape programmer. Brendan Eich developed this new scripting language in just ten days in the year September 1995. At the time of its launch, JavaScript was initially called Mocha. After that, it was called Live Script and later known as JavaScript.

4) List some of the advantages of JavaScript.

Some of the advantages of JavaScript are:

- Server interaction is less
- Feedback to the visitors is immediate
- Interactivity is high
- Interfaces are richer

---

5) List some of the disadvantages of JavaScript.

Some of the disadvantages of JavaScript are:

- No support for multithreading
- No support for multiprocessing
- Reading and writing of files is not allowed
- No support for networking applications.

---

6) Define a named function in JavaScript.

The function which has named at the time of definition is called a named function. For example

```
1. function msg()
2. {
3.   document.writeln("Named Function");
4. }
5. msg();
```

---

7) Name the types of functions

The types of function are:

- Named - These type of functions contains name at the time of definition. For Example:

```
function display()
{
  document.writeln("Named Function");
}
display();
```

- Anonymous - These type of functions doesn't contain any name. They are declared dynamically at runtime.

```
var display=function()
{
  document.writeln("Anonymous Function");
}
display();
```

---

8) Define anonymous function

It is a function that has no name. These functions are declared dynamically at runtime using the function operator instead of the function declaration. The function operator is more flexible than a function declaration. It can be easily used in the place of an expression. For example:

```
var display=function()
{
  alert("Anonymous Function is invoked");
}
display();
```

---

9) Can an anonymous function be assigned to a variable?

Yes, you can assign an anonymous function to a variable.

10) In JavaScript what is an argument object?

The variables of JavaScript represent the arguments that are passed to a function.

11) Define closure.

In JavaScript, we need closures when a variable which is defined outside the scope in reference is accessed from some inner scope.

```
1. var num = 10;
2. function sum()
3. {
4. document.writeln(num+num);
5. }
6. sum();
```

12) If we want to return the character from a specific index which method is used?

The JavaScript string charAt() method is used to find out a char value present at the specified index. The index number starts from 0 and goes to n-1, where n is the length of the string. The index value can't be a negative, greater than or equal to the length of the string. For example:

```
1. var str="Javatpoint";
2. document.writeln(str.charAt(4));
```

13) What is the difference between JavaScript and JScript?

Netscape provided the JavaScript language. Microsoft changed the name and called it JScript to avoid the trademark issue. In other words, you can say JScript is the same as JavaScript, but Microsoft provides it.

14) How to write a hello world example of JavaScript?

A simple example of JavaScript hello world is given below. You need to place it inside the body tag of HTML.

```
<script type="text/javascript">
document.write("JavaScript Hello World!");
</script>
```

More details.

15) What are the key differences between Java and JavaScript? / How is JavaScript different from Java?

JavaScript is a lightweight programming language (most commonly known as scripting language) developed by Netscape, Inc. It is used to make web pages interactive. It is not a part of the Java platform. Following is a list of some key differences between Java and JavaScript

A list of key differences between Java and JavaScript

Java	JavaScript
Java is a complete and strongly typed programming language used for backend coding. In Java, variables must be declared first to use in the program, and the type of a variable is checked at compile-time.	JavaScript is a weakly typed, lightweight programming language (most commonly known as scripting language) and has more relaxed syntax and rules.
Java is an object-oriented programming (OOPS) language or structured programming languages such as C, C++, or .Net.	JavaScript is a client-side scripting language, and it doesn't fully support the OOPS concept. It resides inside the HTML documents and is used to make web pages interactive (not achievable with simple HTML).
Java creates applications that can run in any virtual machine (JVM) or browser.	JavaScript code can run only in the browser, but it can now run on the server via Node.js.

The Java code needs to be compiled.	The JavaScript code doesn't require to be complied.
Java Objects are class-based. You can't make any program in Java without creating a class.	JavaScript Objects are prototype-based.
Java is a Complete and Standalone language that can be used in backend coding.	JavaScript is assigned within a web page and integrates with its HTML content.
Java programs consume more memory.	JavaScript code is used in HTML web pages and requires less memory.
The file extension of the Java program is written as ".Java" and it translates source code into bytecodes which are then executed by JVM (Java Virtual Machine).	The JavaScript file extension is written as ".js" and it is interpreted but not compiled. Every browser has a JavaScript interpreter to execute the JS code.
Java supports multithreading.	JavaScript doesn't support multithreading.
Java uses a thread-based approach to concurrency.	JavaScript uses an event-based approach to concurrency.

16) How to use external JavaScript file?

I am assuming that js file name is message.js, place the following script tag inside the head tag.

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

More details.

17) Is JavaScript case sensitive language?

Yes, JavaScript is a case sensitive language. For example:

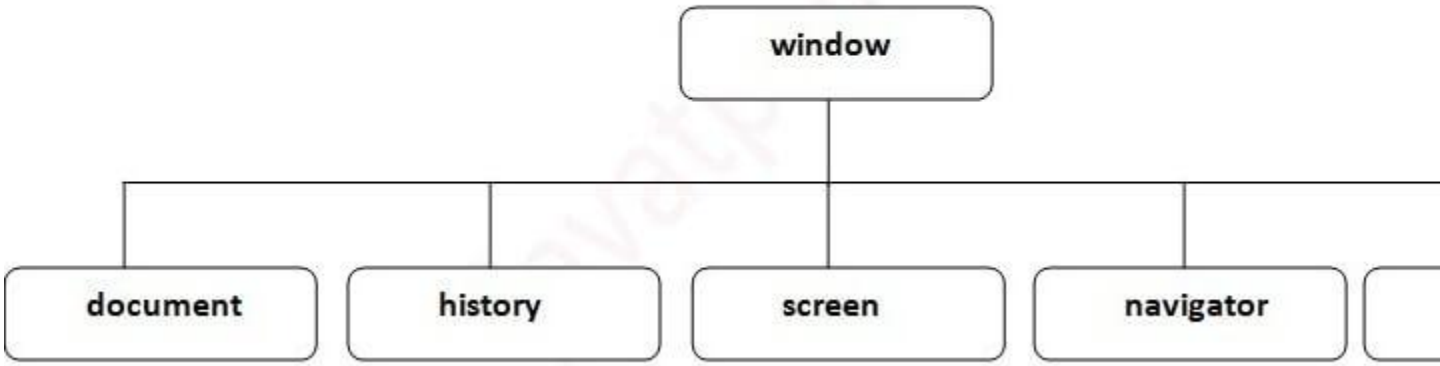
```
Var msg = "JavaScript is a case-
sensitive language"; //Here, var should be used to declare a variable
function display()
{
```

```
4. document.writeln(msg); // It will not display the result.
5. }
6. display();
```

---

18) What is BOM?

BOM stands for *Browser Object Model*. It provides interaction with the browser. The default object of a browser is a window. So, you can call all the functions of the window by specifying the window or directly. The window object provides various properties like document, history, screen, navigator, location, innerHeight, innerWidth,



More Details: Browser Object Model

---

19) What is DOM? What is the use of document object?

DOM stands for *Document Object Model*. A document object represents the HTML document. It can be used to access and change the content of HTML.

More Details: Document Object Model

---

20) What is the use of window object?

The window object is created automatically by the browser that represents a window of a browser. It is not an object of JavaScript. It is a browser object.

The window object is used to display the popup dialog box. Let's see with description.

Method	Description
alert()	displays the alert box containing the message with ok button.
confirm()	displays the confirm dialog box containing the message with ok and cancel button.
prompt()	displays a dialog box to get input from the user.
open()	opens the new window.
close()	closes the current window.
setTimeout()	performs the action after specified time like calling function, evaluating expressions.

More details.

---

21) What is the use of history object?

The history object of a browser can be used to switch to history pages such as back and forward from the current page or another page. There are three methods of history object.

1. history.back() - It loads the previous page.
2. history.forward() - It loads the next page.
3. history.go(number) - The number may be positive for forward, negative for backward. It loads the given page number.

More details.

---

22) How to write a comment in JavaScript?

There are two types of comments in JavaScript.

1. Single Line Comment: It is represented by // (double forward slash)
2. Multi-Line Comment: Slash represents it with asterisk symbol as /* write comment here */

More details.

---

23) How to create a function in JavaScript?

To create a function in JavaScript, follow the following syntax.

```
function function_name(){
//function body
}
```

More details.

---

24) What are the different data types present in JavaScript?

There are two types of data types in JavaScript:

- Primitive data types
- Non- Primitive data types

Primitive data types

The primitive data types are as follows:

String: The string data type represents a sequence of characters. It is written within quotes and can be represented using a single or a double quote.

Example:

```
var str1 = "Hello JavaTpoint"; //using double quotes
var str2 = 'Hello Javatpoint'; //using single quotes
```

Number: The number data type is used to represent numeric values and can be written with or without decimals.

Example:

```
var x = 5; //without decimal
var y = 5.0; //with decimal
```

Boolean: The Boolean data type is used to represent a Boolean value, either false or true. This data type is generally used for conditional testing.

Example:

```
var x = 5;
var y =  6;
var z =  5;
(x == y) // returns false
(x == z) //returns true
```

BigInt: The BigInt data type is used to store numbers beyond the Number data type limitation. This data type can store large integers and is represented by adding "n" to an integer literal.

Example:

```
var bigInteger =  1234567890123456789012345678901234567890;
// This is an example of bigInteger.
```

Undefined: The Undefined data type is used when a variable is declared but not assigned. The value of this data type is undefined, and its type is also undefined.

Example:

```
var x; // value of x is undefined
var y = undefined; // You can also set the value of a variable as undefined.
```

Null: The Null data type is used to represent a non-existent, null, or a invalid value i.e. no value at all.

Example:

```
var  x = null;
```

Symbol: Symbol is a new data type introduced in the ES6 version of JavaScript. It is used to store an anonymous and unique value.

Example:

```
1. var symbol1 = Symbol('symbol');
```

typeof: The typeof operator is used to determine what type of data a variable or operand contains. It can be used with or without parentheses (typeof(x) or typeof x). This is mainly used in situations when you need to process the values of different types.

Example:

```
1. typeof 10;  // Returns: "number"
2. typeof 10.0;  // Returns: "number"
3. typeof 2.5e-4;  // Returns: "number"
4. typeof Infinity;  // Returns: "number"
5. typeof NaN;  // Returns: "number". Despite being "Not-A-Number"
6. // Strings
7. typeof '';  // Returns: "string"
8. typeof 'Welcome to JavaTpoint';  // Returns: "string"
9. typeof '12';  // Returns: "string". Number within quotes is typeof string
10. // Booleans
11. typeof true;  // Returns: "boolean"
12. typeof false;  // Returns: "boolean"
13. // Undefined
14. typeof undefined;  // Returns: "undefined"
15. typeof undeclaredVariable; // Returns: "undefined"
16. // Null
17. typeof Null;  // Returns: "object"
18. // Objects
19. typeof {name: "John", age: 18};  // Returns: "object"
20. // Arrays
21. typeof [1, 2, 3];  // Returns: "object"
22. // Functions
23. typeof function(){};  // Returns: "function"
```

Non-Primitive data types

In the above examples, we can see that the primitive data types can store only a single value. To store multiple and complex values, we have to use non-primitive data types.

The non-primitive data types are as follows:

Object: The Object is a non-primitive data type. It is used to store collections of data. An object contains properties, defined as a key-value pair. A property key (name) is always a string, but the value can be any data type, such as strings, numbers, Booleans, or complex data types like arrays, functions, and other objects.

Example:

```
// Collection of data in key-value pairs
var obj1 = {
   x:  123,
   y:  "Welcome to JavaTpoint",
   z: function(){
      return this.x;
   }
}
```

Array: The Array data type is used to represent a group of similar values. Every value in an array has a numeric position, called its index, and it may contain data of any data type-numbers, strings, Booleans, functions, objects, and even other arrays. The array index starts from 0 so that the first array element is arr[0], not arr[1].

Example:

```
var colors = ["Red", "Yellow", "Green", "Orange"];
var cities = ["Noida", "Delhi", "Ghaziabad"];
alert(colors[2]);   // Output: Green
alert(cities[1]);   // Output: Delhi
```

More details.

---

25) What is the difference between == and ===?

The == operator checks equality only whereas === checks equality, and data type, i.e., a value must be of the same type.

---

26) How to write HTML code dynamically using JavaScript?

The innerHTML property is used to write the HTML code using JavaScript dynamically. Let's see a simple example:

```
1. document.getElementById('mylocation').innerHTML="<h2>This is heading using JavaScript</h2>";
```

More details.

---

27) How to write normal text code using JavaScript dynamically?

The innerText property is used to write the simple text using JavaScript dynamically. Let's see a simple example:

```
1. document.getElementById('mylocation').innerText="This is text using JavaScript";
```

More details.

---

28) How to create objects in JavaScript?

There are 3 ways to create an object in JavaScript.

1. By object literal
2. By creating an instance of Object
3. By Object Constructor

Let's see a simple code to create an object using object literal.

```
1. emp={id:102,name:"Rahul Kumar",salary:50000}
```

More details.

---

29) How to create an array in JavaScript?

There are 3 ways to create an array in JavaScript.

1. By array literal

2. By creating an instance of Array
3. By using an Array constructor

Let's see a simple code to create an array using object literal.

```
var emp=["Shyam","Vimal","Ratan"];
```

More details.

---

30) What does the isNaN() function?

The isNan() function returns true if the variable value is not a number. For example:

```
function number(num) {
  if (isNaN(num)) {
    return "Not a Number";
  }
  return "Number";
}
console.log(number('1000F'));
// expected output: "Not a Number"

console.log(number('1000'));
// expected output: "Number"
```

---

31) What is the output of 10+20+"30" in JavaScript?

3030 because 10+20 will be 30. If there is numeric value before and after +, it treats as binary + (arithmetic operator).

```
function display()
{
  document.writeln(10+20+"30");
}
display();
```

---

32) What is the output of "10"+20+30 in JavaScript?

102030 because after a string all the + will be treated as string concatenation operator (not binary +).

```
1. function display()
2. {
3.   document.writeln("10"+20+30);
4. }
5. display();
```

---

33) Difference between Client side JavaScript and Server side JavaScript?

Client-side JavaScript comprises the basic language and predefined objects which are relevant to running JavaScript in a browser. The client-side JavaScript is embedded directly by in the HTML pages. The browser interprets this script at runtime.

Server-side JavaScript also resembles client-side JavaScript. It has a relevant JavaScript which is to run in a server. The server-side JavaScript are deployed only after compilation.

---

34) In which location cookies are stored on the hard disk?

The storage of cookies on the hard disk depends on the OS and the browser.

The Netscape Navigator on Windows uses a cookies.txt file that contains all the cookies. The path is c:\Program Files\Netscape\Users\username\cookies.txt

The Internet Explorer stores the cookies on a file username@website.txt. The path is: c:\Windows\Cookies\username@Website.txt.

---

35) What's the difference between event.preventDefault() and event.stopPropagation() methods in JavaScript?

In JavaScript, the event.preventDefault() method is used to prevent the default behavior of an element.

For example: If you use it in a form element, it prevents it from submitting. If used in an anchor element, it prevents it from navigating. If used in a contextmenu, it prevents it from showing or displaying.

On the other hand, the event.stopPropagation() method is used to stop the propagation of an event or stop the event from occurring in the bubbling or capturing phase.

---

36) What is the real name of JavaScript?

The original name was **Mocha**, a name chosen by Marc Andreessen, founder of Netscape. In September of 1995, the name was changed to LiveScript. In December 1995, after receiving a trademark license from Sun, the name JavaScript was adopted.

---

37) How can you check if the event.preventDefault() method was used in an element?

When we use the event.defaultPrevent() method in the event object returns a Boolean indicating that the event.preventDefault() was called in a particular element.

---

38) What is the difference between undefined value and null value?

Undefined value: A value that is not defined and has no keyword is known as undefined value. For example:

```
1. int number;//Here, a number has an undefined value.
```

Null value: A value that is explicitly specified by the keyword "null" is known as a null value. For example:

```
1. String str=null;//Here, str has a null value.
```

---

39) How to set the cursor to wait in JavaScript?

The cursor can be set to wait in JavaScript by using the property "cursor". The following example illustrates the usage:

```
<script>
window.document.body.style.cursor = "wait";
</script>
```

---

40) What is this [[[]]]?

This is a three-dimensional array.

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

---

41) Are Java and JavaScript same?

No, Java and JavaScript are the two different languages. Java is a robust, secured and object-oriented programming language whereas JavaScript is a client-side scripting language with some limitations.

---

42) What is negative infinity?

Negative Infinity is a number in JavaScript which can be derived by dividing the negative number by zero. For example:

```
var num=-5;
function display()
{
  document.writeln(num/0);
}
display();
//expected output: -Infinity
```

---

43) What is the difference between View state and Session state?

"View state" is specific to a page in a session whereas "Session state" is specific to a user or browser that can be accessed across all pages in the web application.

---

44) What are the pop-up boxes available in JavaScript?

- Alert Box
- Confirm Box
- Prompt Box

Example of alert() in JavaScript

```
<script type="text/javascript">
function msg(){
 alert("Hello Alert Box");
}
</script>
<input type="button" value="click" onclick="msg()"/>
```

Example of confirm() in JavaScript

```
<script type="text/javascript">
function msg(){
var v= confirm("Are u sure?");
if(v==true){
alert("ok");
}
else{
alert("cancel");
}

}
</script>

```

```
14. <input type="button" value="delete record" onclick="msg()"/>
```

Example of prompt() in JavaScript

```
1. <script type="text/javascript">
2. function msg(){
3. var v= prompt("Who are you?");
4. alert("I am "+v);
5. 
6. }
7. </script>
8. 
9. <input type="button" value="click" onclick="msg()"/>
```

---

45) How can we detect OS of the client machine using JavaScript?

The **navigator.appVersion** string can be used to detect the operating system on the client machine.

---

46) How to submit a form using JavaScript by clicking a link?

Let's see the JavaScript code to submit the form by clicking the link.

```
1. <form name="myform" action="index.php">
2. Search: <input type='text' name='query' />
3. <a href="javascript: submitform()">Search</a>
4. </form>
5. <script type="text/javascript">
6. function submitform()
7. {
8.   document.myform.submit();
9. }
10. </script>
```

47) Is JavaScript faster than ASP script?

Yes, because it doesn't require web server's support for execution.

48) How to change the background color of HTML document using JavaScript?

```
<script type="text/javascript">
document.body.bgColor="pink";
</script>
```

49) How to handle exceptions in JavaScript?

By the help of try/catch block, we can handle exceptions in JavaScript. JavaScript supports try, catch, finally and throw keywords for exception handling.

50) How to validate a form in JavaScript?

```
<script>
function validateform(){
var name=document.myform.name.value;
var password=document.myform.password.value;

if (name==null || name==""){
  alert("Name can't be blank");
  return false;
}else if(password.length<6){
  alert("Password must be at least 6 characters long.");
  return false;
  }
}
```

```
14. </script>
15. <body>
16. <form name="myform" method="post" action="abc.jsp" onsubmit="return validatefor
    m()" >
17. Name: <input type="text" name="name"><br/>
18. Password: <input type="password" name="password"><br/>
19. <input type="submit" value="register">
20. </form>
```

Visit here: JavaScript form validation.

---

51) How to validate email in JavaScript?

```
1.  <script>
2.  function validateemail()
3.  {
4.  var x=document.myform.email.value;
5.  var atposition=x.indexOf("@");
6.  var dotposition=x.lastIndexOf(".");
7.  if (atposition<1 || dotposition<atposition+2 || dotposition+2>=x.length){
8.    alert("Please enter a valid e-
    mail address \n atpostion:"+atposition+"\n dotposition:"+dotposition);
9.    return false;
10.   }
11. }
12. </script>
13. <body>
14. <form name="myform"  method="post" action="#" onsubmit="return validateemail();"
    >
15. Email: <input type="text" name="email"><br/>
16.
17. <input type="submit" value="register">
18. </form>
```

Visit here: JavaScript Email validation.

---

52) What is this keyword in JavaScript?

The this keyword is a reference variable that refers to the current object. For example:

```
var address=
{
company:"Javatpoint",
city:"Noida",
state:"UP",
fullAddress:function()
{
return this.company+" "+this.city+" "+this.state;
}
};
var fetch=address.fullAddress();
document.writeln(fetch);
```

---

53) What is the requirement of debugging in JavaScript?

JavaScript didn't show any error message in a browser. However, these mistakes can affect the output. The best practice to find out the error is to debug the code. The code can be debugged easily by using web browsers like Google Chrome, Mozilla Firebox.

To perform debugging, we can use any of the following approaches:

- Using console.log() method
- Using debugger keyword

---

54) What is the use of debugger keyword in JavaScript?

JavaScript debugger keyword sets the breakpoint through the code itself. The debugger stops the execution of the program at the position it is applied. Now, we can start the flow of execution manually. If an exception occurs, the execution will stop again on that particular line.. For example:

```
1. function display()
2. {
3. x = 10;
4. y = 15;
5. z = x + y;
6. debugger;
7. document.write(z);
8. document.write(a);
9. }
10. display();
```

---

55) What is the role of a strict mode in JavaScript?

The JavaScript strict mode is used to generates silent errors. It provides "use strict"; expression to enable the strict mode. This expression can only be placed as the first statement in a script or a function. For example:

```
1. "use strict";
2. x=10;
3. console.log(x);
```

---

57) What is the use of Math object in JavaScript?

The JavaScript math object provides several constants and methods to perform a mathematical operation. Unlike date object, it doesn't have constructors. For example:

```
1. function display()
2. {
3.   document.writeln(Math.random());
4. }
```

```
5. display();
```

---

58) What is the use of a Date object in JavaScript?

The JavaScript date object can be used to get a year, month and day. You can display a timer on the webpage by the help of JavaScript date object.

```
1. function display()
2. {
3.   var date=new Date();
4. var day=date.getDate();
5. var month=date.getMonth()+1;
6. var year=date.getFullYear();
7. document.write("<br>Date is: "+day+"/"+month+"/"+year);
8. }
9. display();
```

---

59) What is the use of a Number object in JavaScript?

The JavaScript number object enables you to represent a numeric value. It may be integer or floating-point. JavaScript number object follows the IEEE standard to represent the floating-point numbers.

```
1. function display()
2. {
3. var x=102;//integer value
4. var y=102.7;//floating point value
5. var z=13e4;//exponent value, output: 130000
6. var n=new Number(16);//integer value by number object
7. document.write(x+" "+y+" "+z+" "+n);
8. }
9. display();
```

---

60) What is the use of a Boolean object in JavaScript?

The JavaScript Boolean is an object that represents value in two states: true or false. You can create the JavaScript Boolean object by Boolean() constructor.

```
1. function display()
2. {
3. document.writeln(10<20);//true
4. document.writeln(10<5);//false
5. }
6. display();
```

---

61) What is the use of a TypedArray object in JavaScript?

The JavaScript TypedArray object illustrates an array like a view of an underlying binary data buffer. There is any number of different global properties, whose values are TypedArray constructors for specific element types.

```
1. function display()
2. {
3. var arr1= [1,2,3,4,5,6,7,8,9,10];
4.       arr1.copyWithin(2) ;
5.       document.write(arr1);
6. }
7. display();
```

---

62) What is the use of a Set object in JavaScript?

The JavaScript Set object is used to store the elements with unique values. The values can be of any type i.e. whether primitive values or object references. For example:

```
1. function display()
2. {
3. var set = new Set();
4. set.add("jQuery");
```

```
5. set.add("AngularJS");
6. set.add("Bootstrap");
7. for (let elements of set) {
8.  document.writeln(elements+"<br>");
9. }
10. }
11. display();
```

---

63) What is the use of a WeakSet object in JavaScript?

The JavaScript WeakSet object is the type of collection that allows us to store weakly held objects. Unlike Set, the WeakSet are the collections of objects only. It doesn't contain the arbitrary values. For example:

```
1. function display()
2. {
3. var ws = new WeakSet();
4. var obj1={};
5. var obj2={};
6. ws.add(obj1);
7. ws.add(obj2);
8. //Let's check whether the WeakSet object contains the added object
9. document.writeln(ws.has(obj1)+"<br>");
10. document.writeln(ws.has(obj2));
11. }
12. display()
```

---

64) What is the use of a Map object in JavaScript?

The JavaScript Map object is used to map keys to values. It stores each element as key-value pair. It operates the elements such as search, update and delete on the basis of specified key. For example:

```
1. function display()
```

```
{
var map=new Map();
map.set(1,"jQuery");
map.set(2,"AngularJS");
map.set(3,"Bootstrap");
document.writeln(map.get(1)+"<br>");
document.writeln(map.get(2)+"<br>");
document.writeln(map.get(3));
}
display();
```

---

65) What is the use of a WeakMap object in JavaScript?

The JavaScript WeakMap object is a type of collection which is almost similar to Map. It stores each element as a key-value pair where keys are weakly referenced. Here, the keys are objects and the values are arbitrary values. For example:

```
function display()
{
var wm = new WeakMap();
var obj1 = {};
var obj2 = {};
var obj3= {};
wm.set(obj1, "jQuery");
wm.set(obj2, "AngularJS");
wm.set(obj3,"Bootstrap");
document.writeln(wm.has(obj2));
}
display();
```

---

66) What are the falsy values in JavaScript, and how can we check if a value is falsy?

Those values which become false while converting to Boolean are called falsy values.

```
1. const falsyValues = ['', 0, null, undefined, NaN, false];
```

We can check if a value is falsy by using the Boolean function or the Double NOT operator (!!).

---

67) What do you understand by hoisting in JavaScript?

Hoisting is the default behavior of JavaScript where all the variable and function declarations are moved on top. In simple words, we can say that Hoisting is a process in which, irrespective of where the variables and functions are declared, they are moved on top of the scope. The scope can be both local and global.

Example 1:

```
1. hoistedVariable = 12;
2. console.log(hoistedVariable); // outputs 12 even when the variable is declared after it is i
   nitialized
3. var hoistedVariable;
```

Example2:

```
1. hoistedFunction();  // Outputs " Welcome to JavaTpoint " even when the function is decl
   ared after calling
2. function hoistedFunction(){
3.   console.log(" Welcome to JavaTpoint ");
4. }
5. Example3:
6. // Hoisting in a local scope
7. function doSomething(){
8.   x = 11;
9.   console.log(x);
10.  var x;
11. }
12. doSomething(); // Outputs 11 since the local variable "x" is hoisted inside the local scop
    e
```

ES6 Interview Questions

A list of frequently asked **ES6 Interview Questions** and Answers are given below.

1) What is ES6 or ECMAScript 2015?

ES6 was released in June 2015, which is stated as the sixth edition of the language. Initially, it was named **ES6** and later renamed to ECMAScript 2015. This edition includes several new features that are modules, iterators, class, arrow functions, for...of loop, promises, and many more. Brendan Eich developed it.

---

2) Define ECMAScript.

It is the specification that is defined in the ECMA-262 standard to create a general-purpose scripting language.

---

3) What are the new features introduced in ES6?

The new features that are introduced in ES6 are listed as follows:

- Let and const keywords.
- Default Parameters.
- Arrow functions.
- Template Literals.
- Object Literals.
- Rest and spread operators.
- Destructuring assignment.
- Modules, Classes, Generators, and iterators.
- Promises, and many more.

---

4) Define let and const keywords.

let: The variables declared using **let** keyword will be mutable, i.e., the values of the variable can be changed. It is similar to **var** keyword except that it provides block scoping.

const: The variables declared using **the const** keyword are immutable and block-scoped. The value of the variables cannot be changed or re-assigned if they are declared by using **the const** keyword.

---

5) What is the arrow function, and how to create it?

Arrow functions are introduced in ES6. Arrow functions are the shorthand notation to write ES6 functions. The definition of the arrow function consists of parameters, followed by an arrow (=>) and the body of the function.

An Arrow function is also called as **'fat arrow'** function. We cannot use them as constructors.

Syntax

```
const functionName = (arg1, arg2, ...) => {
    //body of the function
}
```

---

6) Give an example of an Arrow function in ES6? List down its advantages.

Arrow function provides us a more accurate way of writing the functions in JavaScript. They allow us to write smaller function syntax.

The context within the arrow functions is lexically or statically scoped. Arrow functions do not include any prototype property, and cannot be used with the new keyword.

You can learn more about arrow functions by clicking on this link ES6 Arrow Function.

Example

```
var myfun = () => {
```

```
2.      console.log("It is an Arrow Function");
3.     };
4.     myfun();
```

Output

```
It is an Arrow Function
```

Advantages of Arrow Function

The advantages of the arrow function are listed below:

- It reduces code size.
- The return statement is optional for a single line function.
- Lexically bind the context.
- Functional braces are optional for a single-line statement.

---

7) Discuss spread operator in ES6 with an example.

The spread operator is represented by three dots (...) to obtain the list of parameters. It allows the expansion of an iterable such as array or string in places where more than zero arguments are expected.

The spread operator syntax is similar to the rest operator, but functionality is entirely opposite to it. It is also used to combine or to perform the concatenation between arrays. Let's understand it by an example.

Example

```
1. let num1 = [40,50,60];
2.
3. let num2 = [10,20,30,...num1,70,80,90,100];
4.
5. console.log(num2);
```

Output

```
[
```

```
  10, 20, 30, 40,  50,
  60, 70, 80, 90, 100
]
```

8) Discuss the Rest parameter in ES6 with an example.

It is introduced in ES6 that improves the ability to handle the parameters. With rest parameters, it is possible to represent indefinite parameters as an array. By using the rest parameter, we can call a function with any number of arguments.

Example

```
function show(...args) {
    let sum = 0;
    for (let i of args) {
        sum += i;
    }
    console.log("Sum = "+sum);
 }

 show(10, 20, 30);
```

Output

```
Sum = 60
```

9) What are the template literals in ES6?

Template literals are a new feature introduced in ES6. It provides an easy way of creating multiline strings and perform string interpolation.

Template literals allow embedded expressions and also called as string literals.

Prior to ES6, template literals were referred to as **template strings**. Template literals are enclosed by the **backtick (` `) character**. Placeholders in template literals are represented by the dollar sign and the curly braces **(${expression})**. If we require to use an expression within the backticks, then we can place that expression in the **(${expression})**.

To learn more about template literals in ES6, follow this link ES6 Template Literals.

Example

```
let str1 = "Hello";

let str2 = "World";

let str = `${str1} ${str2}`;
console.log(str);
```

Output

```
Hello World
```

---

10) Discuss Destructuring Assignment in ES6.

Destructuring is introduced in ECMAScript 2015 or ES6 to extract data from objects and arrays into separate variables. It allows us to extract smaller fragments from objects and arrays.

To learn more about array destructuring in ES6, follow this link ES6 Array Destructuring.

To learn more about object destructuring in ES6, follow this link ES6 Object Destructuring.

Example

```
let fullname =['Alan','Rickman'];
let [fname,lname] = fullname;
console.log (fname,lname);
```

Output

```
Alan Rickman
```

---

11) How to create a class in ES6?

This keyword is used for creating the class. We can include the classes in our code either by using class expression or by class declaration. A class definition can only include **functions** and **constructors**. These components are together called as data members of the class.

Constructors in classes allocate the memory to the objects of the class. Functions in a class are responsible for performing the actions to the objects.

To learn more about classes in ES6, follow this link ES6 Classes.

Let us see the syntax for creating classes.

Syntax: In ES5

```
var var_name = new class_name {
}
```

Syntax: In ES6 (Using class keyword)

```
class class_name{
}
```

---

12) What do you understand by Generator function?

A generator provides us a new way to work with iterators and functions. The generator is a special kind of function that may be paused in the middle either one or many times and can be resumed later. The declaration **function* (*function* keyword followed by an asterisk)** is used to define a generator function.

When the generator gets called, it does not run its code. Instead, it returns a special object, which is called a **Generator object** to manage the execution. Let us see an example of generators in ES6.

To learn more about Generators in ES6, follow this link ES6 Generators.

Example

```
function* gen()
{
```

```
3. yield 100;
4. yield;
5. yield 200;
6. }
7. // Calling the Generator Function
8. var mygen = gen();
9. console.log(mygen.next().value);
10. console.log(mygen.next().value);
11. console.log(mygen.next().value);
```

Output

```
100
undefined
200
```

---

13) What are the default parameters?

By using the default parameters, we can initialize named parameters with default values if there is no value or **undefined** is passed.

Example

```
1. var show = (a, b=200) => {
2.     console.log(a + " " + b);
3. }
4. show(100);
```

Output

```
100 200
```

---

14) What do you mean by IIFE (Immediately invoked function expressions)?

IIFE is a function in JavaScript that runs as soon as it is defined. It is also called as the **Self-Executing Anonymous Function**. It includes two major parts that are as follows:

- The first part is an anonymous function that has a lexical scope (static scope), which is enclosed within the **Grouping operator ()**.
- The second part creates the IIFE by which the JavaScript engine will interpret the function directly.

You can learn more about arrow functions by clicking on this link ES6 IIFE.

Example

```
(function()
 {
 console.log("Hello World");
})();
```

Output

```
Hello World
```

---

15) Discuss the for...in loop.

It is similar to for loop that iterates through the properties of an object. It is useful when we require to visit the properties or keys of the object.

Example

```
function Mobile(model_no){
    this.Model = model_no;
    this.Color = 'White';
    this.RAM = '4GB';
    }
    var Samsung = new Mobile("Galaxy");
    for(var props in Samsung)
    {
    console.log(props+ " : " +Samsung[props]);
    }
```

Output

```
Model: Galaxy
Color:  White
RAM: 4GB
```

---

16) Discuss the for...of loop.

This loop is used for iterating the iterables (arrays, string, etc.).

Example

```
var fruits = ['Apple', 'Banana', 'Mango', 'Orange'];
for(let value of fruits)
{
  console.log(value);
}
```

Output

```
Apple
Banana
Mango
Orange
```

---

17) Define set.

A set is a data structure that allows us to create a collection of unique values. It is a collection of values that are similar to arrays, but it does not include any duplicates. It supports both object references and primitive values.

To learn more about Sets in ES6, follow this link ES6 Sets.

Example

```
let colors = new Set(['Green', 'Red', 'Orange', 'Yellow', 'Red']);
console.log(colors);
```

Output

```
Set { 'Green', 'Red', 'Orange', 'Yellow' }
```

---

18) Define Map.

Prior to ES6, when we require the mapping of keys and values, we often use an object. **Map object** is a new collection type, which is introduced in ES6. It holds the key-value pairs in which any type of values can be used as either keys or values.

A map object always remembers the actual insertion order of the keys. Maps are ordered, so they traverse the elements in their insertion order.

To learn more about Map in ES6, follow this link ES6 Maps.

---

19) What do you understand by Weakset?

Using weakset, it is possible to store weakly held objects in a collection. As similar to set, weakset cannot store duplicate values. Weakset cannot be iterated.

Weakset only includes **add(value), delete(value)** and **has(value)** methods of the set object.

---

20) What do you understand by Weakmap?

Weak maps are almost similar to maps, but the keys in weak maps must be objects. It stores each element as a key-value pair where keys are weakly referenced. Here, the keys are objects, and the values are arbitrary.

A weak map object iterates the element in their insertion order. It only includes **delete(key), get(key), has(key)** and **set(key, value)** method.

---

21) Explain Promises in ES6.

ES6 promises are the easiest way to work with asynchronous programming in JavaScript. Asynchronous programming includes running of processes individually from the main thread, and it notifies the main thread when it gets complete.

Prior to ES6, there is the use of **Callbacks** for performing asynchronous programming. Promises are used to overcome the problem of **Callback hell**.

To learn more about promises, follow this link: ES6 Promises.

---

22) What are the states of promises in ES6?

Promises have mainly three states that are as follows:

- **Pending:** It is the initial state of every promise. It represents that the result has not been computed yet.
- **Fulfilled:** It represents the completion of an operation.
- **Rejected:** It represents the failure that occurs during computation.

Once the promise is fulfilled or rejected, then it will be immutable. The **Promise()** constructor takes two arguments that are **rejected** function and a **resolve** function. Based on the asynchronous operation, it returns either the first argument or the second argument.

---

23) What do you understand by Callback and Callback hell in JavaScript?

Callback: It is used to handle the execution of function after the completion of the execution of another function. A callback would be helpful in working with events. In the callback, a function can be passed as an argument to another function. It is a great way when we are dealing with basic cases such as minimal asynchronous operations.

Callback hell: When we develop a web application that includes a lot of code, then working with callback is messy. This excessive Callback nesting is often referred to as **Callback hell**.

---

24) List the comparisons between ES5 and ES6.

ES5 and ES6 are similar in their nature, but there are some differences between them. The comparison between ES5 and ES6 are tabulated as follows:

Based on	ES5	ES6
Definition	ES5 is the fifth edition of the ECMAScript (a trademarked scripting language specification defined by ECMA International)	ES6 is the sixth edition of the ECMAScript (a trademarked scripting language specification defined by ECMA International).
Release	It was introduced in 2009.	It was introduced in 2015.
Data-types	ES5 supports primitive data types that are **string, boolean, number, null,** and **undefined**.	In ES6, there are some additions to JavaScript data types. It introduced a new primitive data type **'symbol'** for supporting unique values.
Defining Variables	In ES5, we could only define the variables by using the **var** keyword.	In ES6, there are two new ways to define variables that are **let** and **const**.
Performance	As ES5 is prior to ES6, there is a non-presence of some features, so it has a lower performance than ES6.	Because of new features and the shorthand storage implementation ES6 has a higher performance than ES5.
Support	A wide range of communities supports it.	It also has a lot of community support, but it is lesser than ES5.
Object Manipulation	ES5 is time-consuming than ES6.	Due to destructuring and speed operators, object manipulation can be processed more smoothly in ES6.
Arrow Functions	In ES5, both **function** and **return** keywords are used to define a function.	An arrow function is a new feature introduced in ES6 by which we don't require the **function** keyword to define the function.
Loops	In ES5, there is a use of **for** loop to iterate over elements.	ES6 introduced the concept of **for...of** loop to perform an iteration over the values of the iterable objects.

To learn more about the difference between ES5 and ES6, follow this link: ES5 v/s ES6

---

25) Define Modules in JavaScript.

Modules are the piece of JavaScript code written in a file. By using Modules, it is easy to maintain the code, debug the code, and reuse the code. Each module is a piece of code that gets executed once it is loaded.

---

26) What do you understand by the term Hoisting in JavaScript?

It is a JavaScript's default behavior, which is used to move all the declarations at the top of the scope before the execution of code. It can be applied to functions as well as on variables. It allows the JavaScript to use the component before its declaration. It does not apply to scripts that run in strict mode.

---

27) List the new Array methods introduced in ES6?

There are many array methods available in ES6, which are listed below:

- Array.of()
- Array.from()
- Array.prototype.copyWithin()
- Array.prototype.find()
- Array.prototype.findIndex()
- Array.prototype.entries()
- Array.prototype.keys()
- Array.prototype.values()
- Array.prototype.fill()

To learn more about the above array methods, follow this link: ES6 Array methods.

---

28) What are the new String methods introduced in ES6?

There are four string methods introduced in ES6 that are listed as follows:

- string.startsWith()
- string.endsWith()
- string.includes()
- string.repeat()

To learn more about the strings, follow this link: ES6 Strings.

---

29) Define Babel.

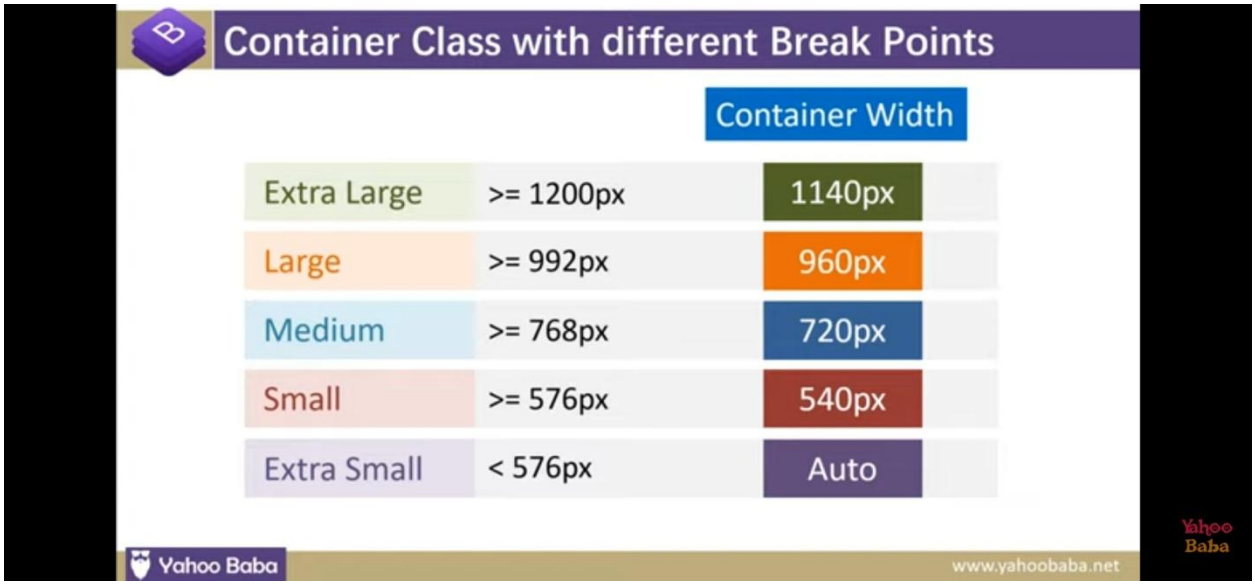
Babel is one of the popular transpilers of JavaScript. It is mainly used for converting the ES6 plus code into the backward-compatible version of JavaScript that can be run by previous JavaScript engines.

---

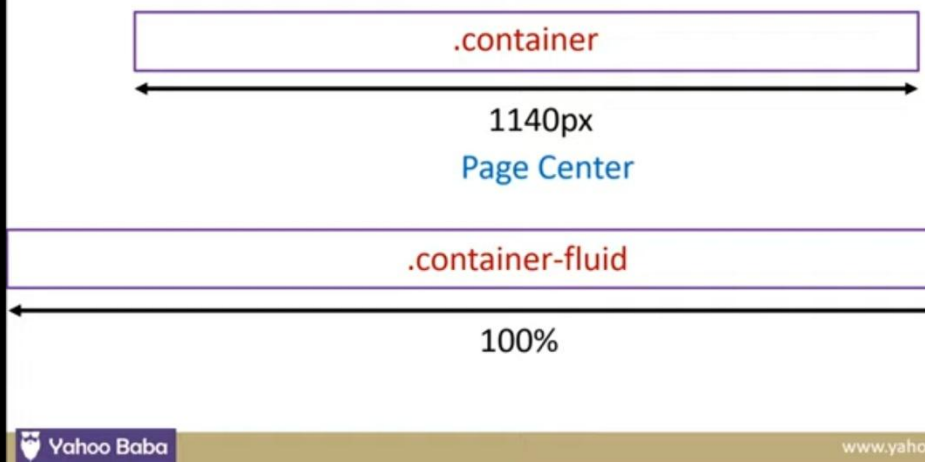
30) Define Webpack.

It is an open-source JavaScript module bundler that takes modules with dependencies. It allows us to run an environment that hosts Babel.

Component	Bootstrap 3	Bootstrap 4
Big Change	-	**Flexbox**
CSS preprocessors	**LESS** LESS is JavaScript based	**SCSS** Sass is Ruby based
Grid System	**4 Columns** (xs, sm, md, lg)	**5 Columns** (xs, sm, md, lg, xl)
Glyphicons	**Supported**	**Not Supported**
Primary CSS Unit	**px**	**rem**
Global Font Size	**14px**	**16px**
Dropdown Structure	**Created with ul and li**	**Created with ul or div**
Offsetting Columns	**col-md-offset-4**	**offset-md-4**
Images	**.img-responsive**	**.img-fluid**

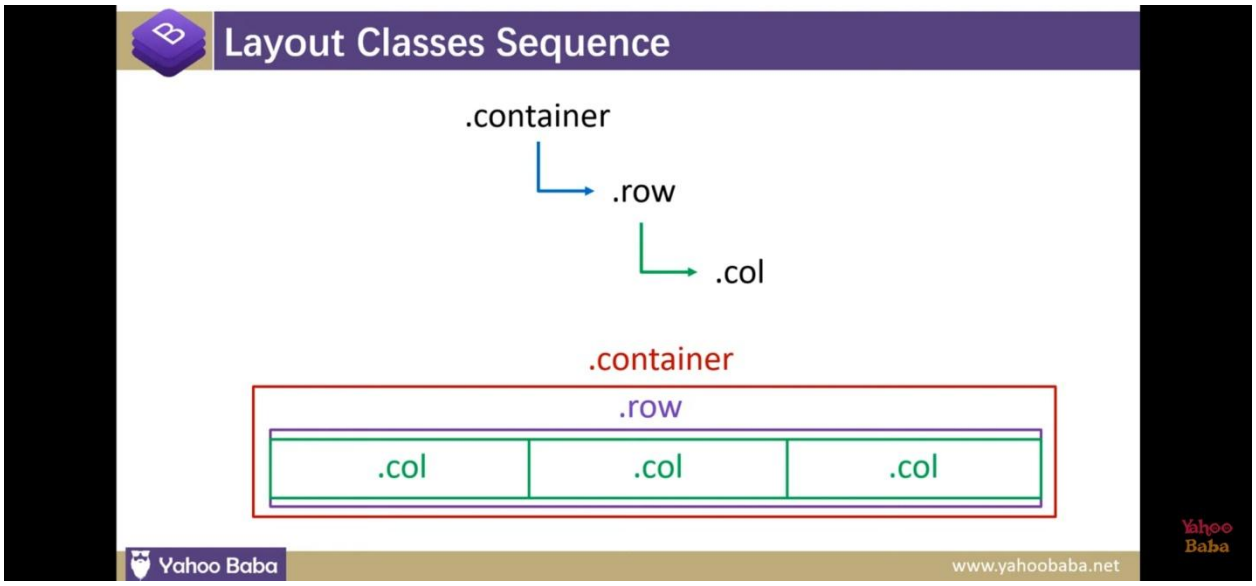


Container Class

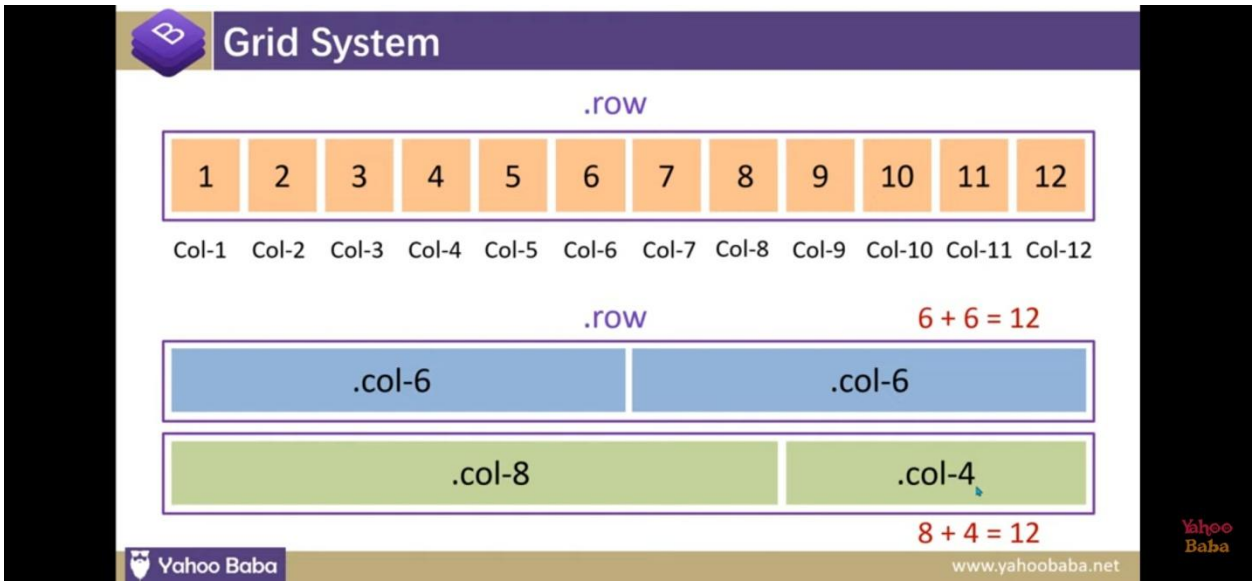


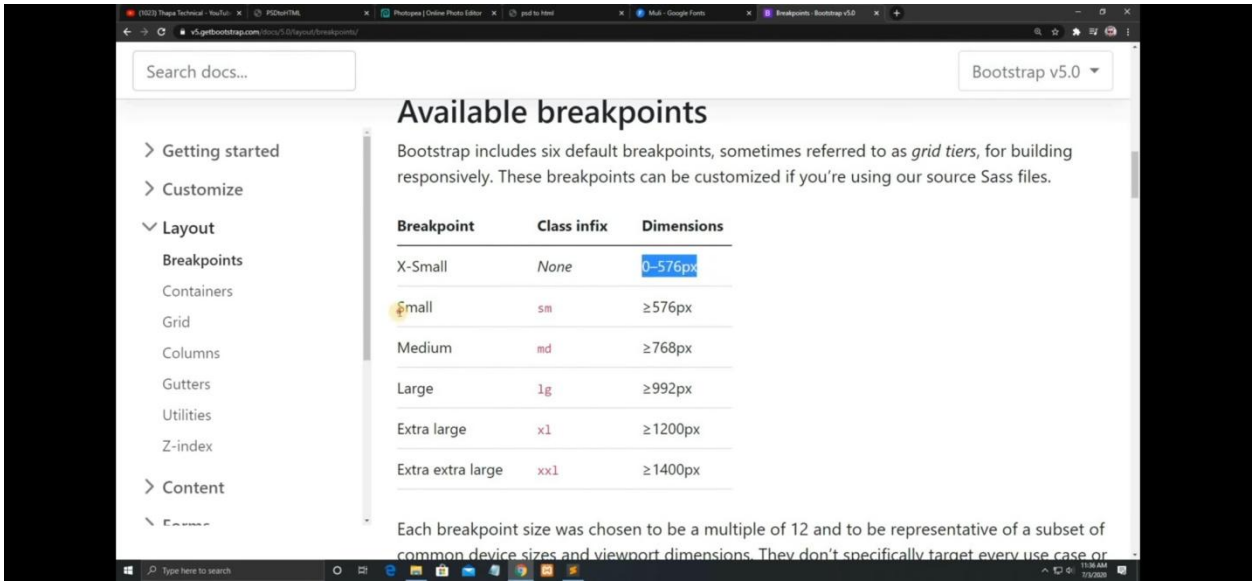
Container Class with different Break Points

		Container Width
Extra Large	>= 1200px	1140px
Large	>= 992px	960px
Medium	>= 768px	720px
Small	>= 576px	540px
Extra Small	< 576px	Auto


Layout Classes Sequence
.container
.row
.col
.container
.row
.col
.col
.col
Yahoo Baba
www.yahoobaba.net
Yahoo Baba


Different Position Values
1). Relative
2). Absolute
3). Fixed
4). Sticky
5). Static


Grid System
.row
1 2 3 4 5 6 7 8 9 10 11 12
Col-1 Col-2 Col-3 Col-4 Col-5 Col-6 Col-7 Col-8 Col-9 Col-10 Col-11 Col-12
.row
6 + 6 = 12
.col-6
.col-6
.col-8
.col-4
8 + 4 = 12
Yahoo Baba
www.yahoobaba.net
Yahoo Baba


Search docs...
Bootstrap v5.0
Getting started
Customize
Layout
Breakpoints
Containers
Grid
Columns
Gutters
Utilities
Z-index
Content
Available breakpoints
Bootstrap includes six default breakpoints, sometimes referred to as grid tiers, for building responsively. These breakpoints can be customized if you're using our source Sass files.
Breakpoint
Class infix
Dimensions
X-Small
None
0–576px
Small
sm
≥576px
Medium
md
≥768px
Large
lg
≥992px
Extra large
xl
≥1200px
Extra extra large
xxl
≥1400px
Each breakpoint size was chosen to be a multiple of 12 and to be representative of a subset of

Padding Classes for different Break Points

	Padding	Padding-right	Padding-left	Padding-top	Padding-bottom
Extra Large	.p-xl-*	.pr-xl-*	.pl-xl-*	.pt-xl-*	.pb-xl-*
Large	.p-lg-*	.pr-lg-*	.pl-lg-*	.pt-lg-*	.pb-lg-*
Medium	.p-md-*	.pr-md-*	.pl-md-*	.pt-md-*	.pb-md-*
Small	.p-sm-*	.pr-sm-*	.pl-sm-*	.pt-sm-*	.pb-sm-*
Extra Small					

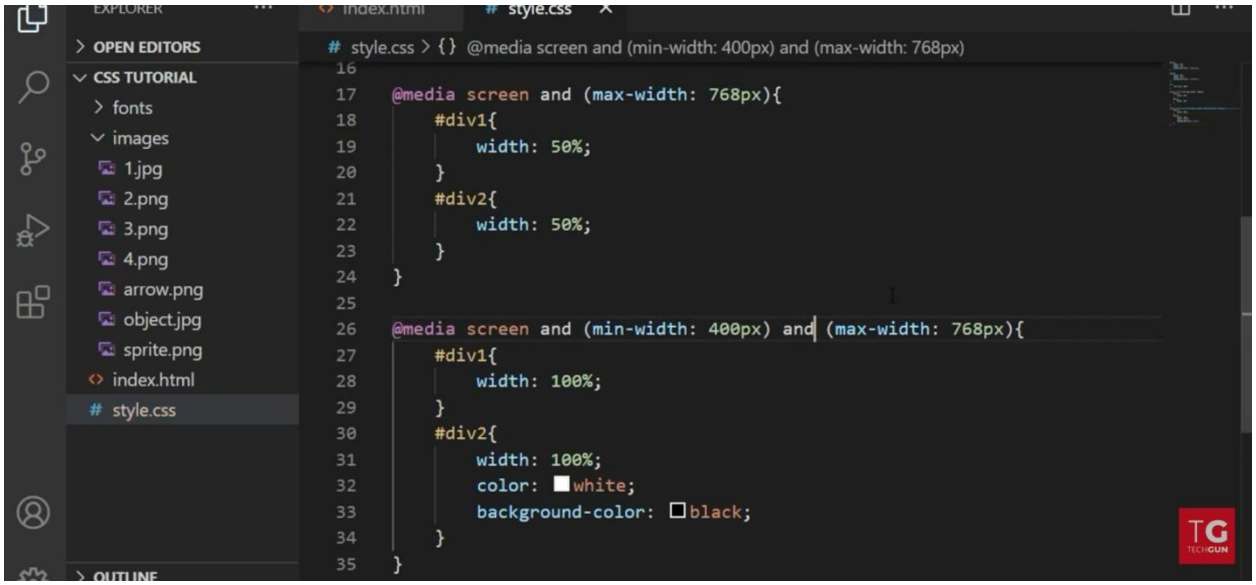
0 - 5

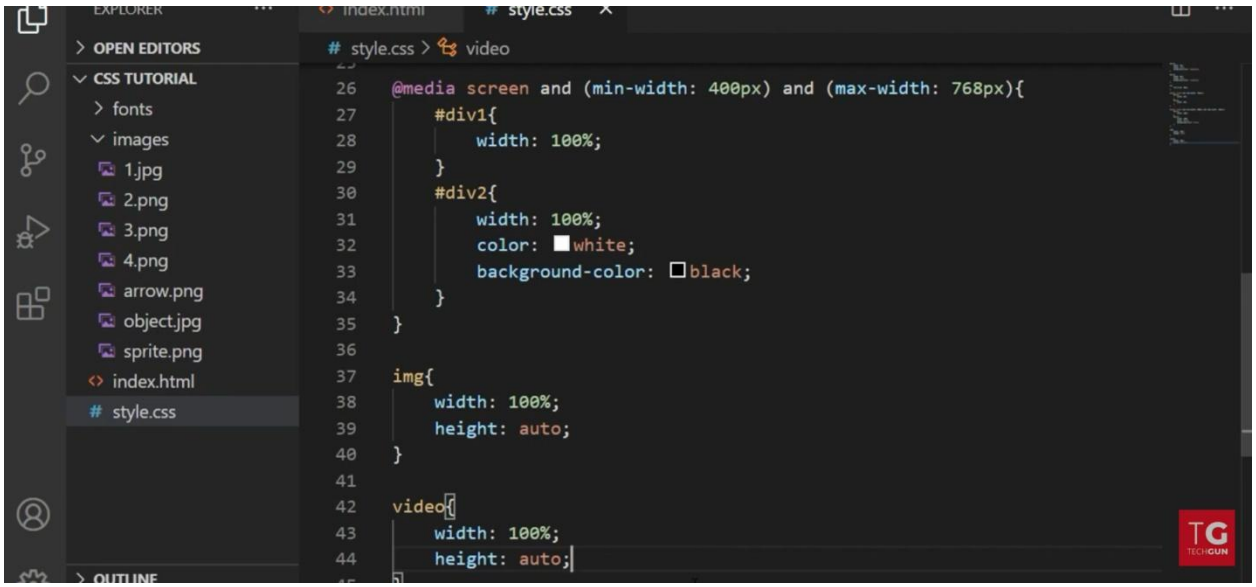
Yahoo Baba

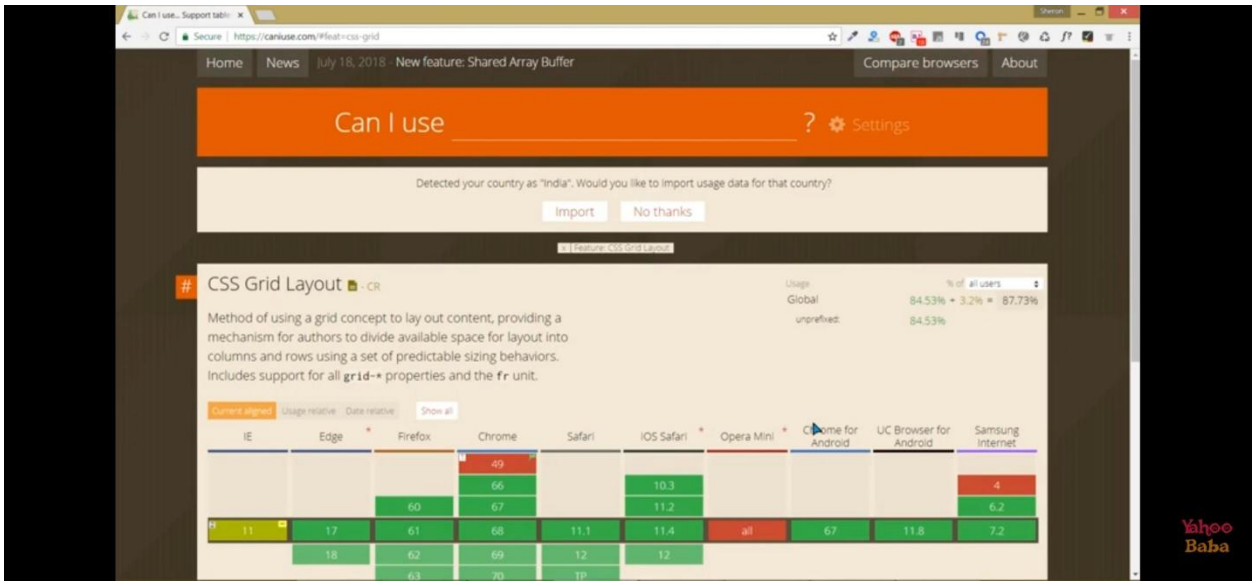
About CSS Flexbox :

- New module in CSS3 to easy align elements in different directions and orders
- Flexbox is to gives container the ability to expand and shrink elements to best use all the available space
- CSS Flexbox Layout replaces float layout
- New way to build one-dimensional layouts

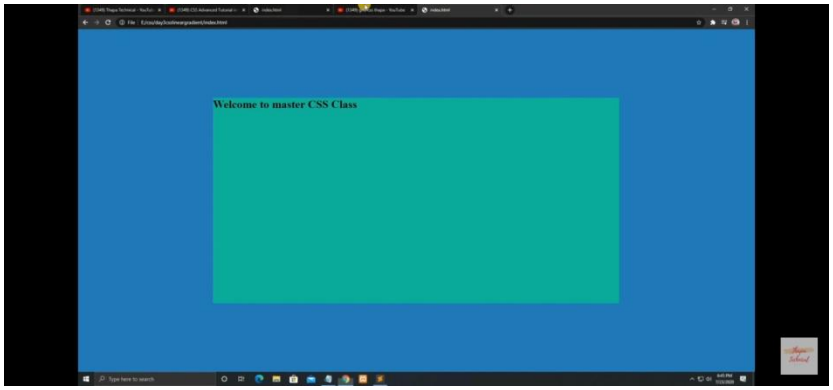
Yahoo Baba

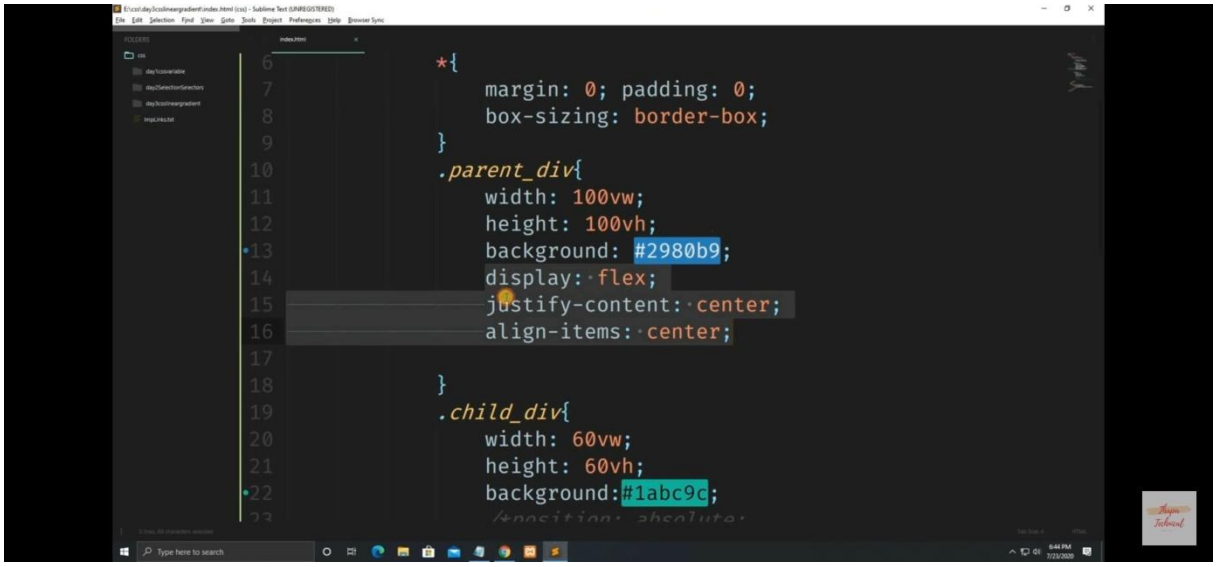

EXPLORER
index.html
style.css
OPEN EDITORS
CSS TUTORIAL
fonts
images
1.jpg
2.png
3.png
4.png
arrow.png
object.jpg
sprite.png
index.html
style.css
OUTLINE
style.css > {} @media screen and (min-width: 400px) and (max-width: 768px)
@media screen and (max-width: 768px){
#div1{
width: 50%;
}
#div2{
width: 50%;
}
}
@media screen and (min-width: 400px) and (max-width: 768px){
#div1{
width: 100%;
}
#div2{
width: 100%;
color: white;
background-color: black;
}
}


EXPLORER
index.html
style.css
OPEN EDITORS
CSS TUTORIAL
fonts
images
1.jpg
2.png
3.png
4.png
arrow.png
object.jpg
sprite.png
index.html
style.css
OUTLINE
style.css > video
@media screen and (min-width: 400px) and (max-width: 768px){
#div1{
width: 100%;
}
#div2{
width: 100%;
color: white;
background-color: black;
}
}
img{
width: 100%;
height: auto;
}
video{
width: 100%;
height: auto;

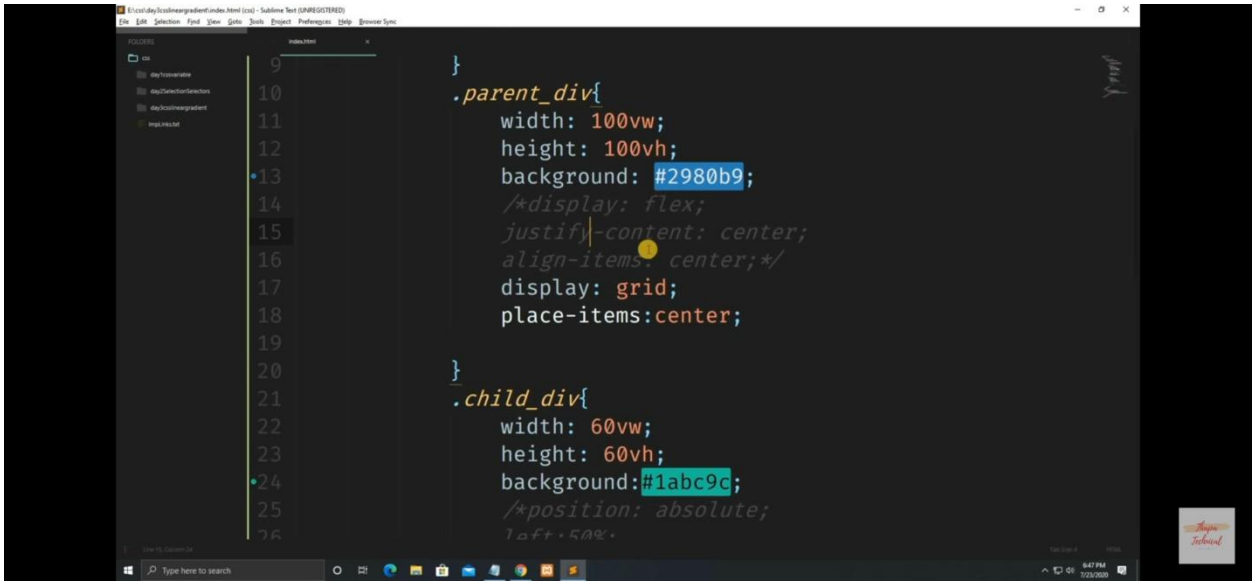

Home
News
July 18, 2018 · New feature: Shared Array Buffer
Compare browsers
About
Can I use
Settings
Detected your country as "India". Would you like to import usage data for that country?
Import
No thanks
CSS Grid Layout
Method of using a grid concept to lay out content, providing a mechanism for authors to divide available space for layout into columns and rows using a set of predictable sizing behaviors. Includes support for all grid-* properties and the fr unit.
Global
84.53% + 3.2% = 87.73%
84.53%
IE
Edge
Firefox
Chrome
Safari
iOS Safari
Opera Mini
Chrome for Android
UC Browser for Android
Samsung Internet
Yahoo Baba


CSS Grid Properties:
grid-template-rows
grid-template-columns
grid-template-areas
grid-template
grid-row-gap
grid-column-gap
grid-gap
justify-items
align-items
justify-content
align-content
grid-auto-rows
grid-auto-columns
grid-auto-flow
grid-row-start
grid-row-end
grid-row
grid-column-start
grid-column-end
grid-column
grid-area
justify-self
align-self
order
Yahoo Baba

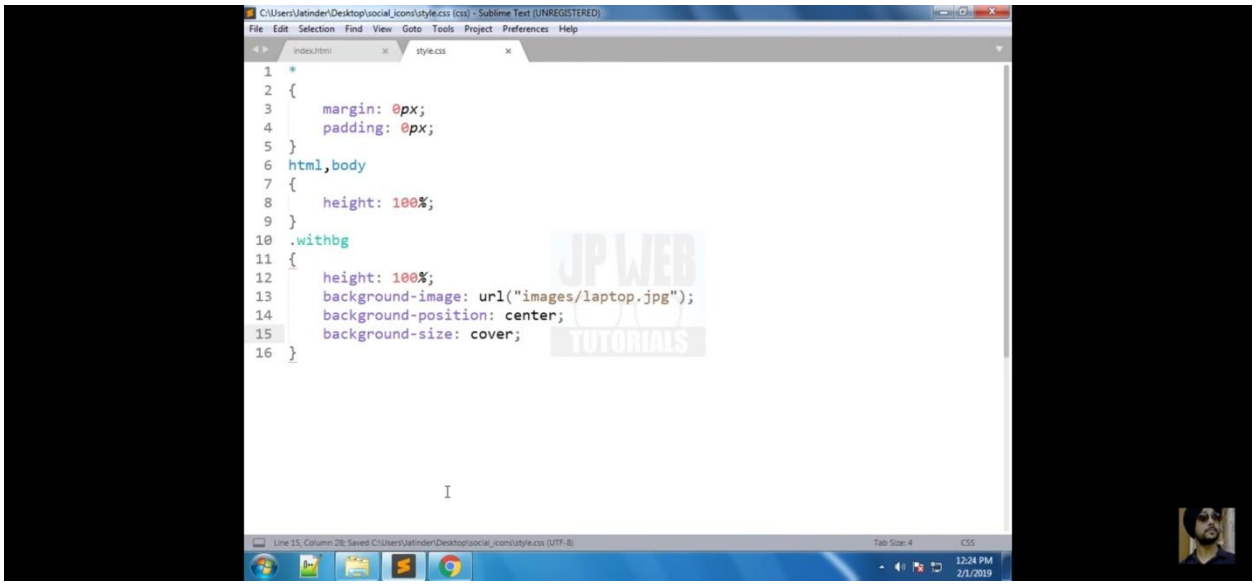


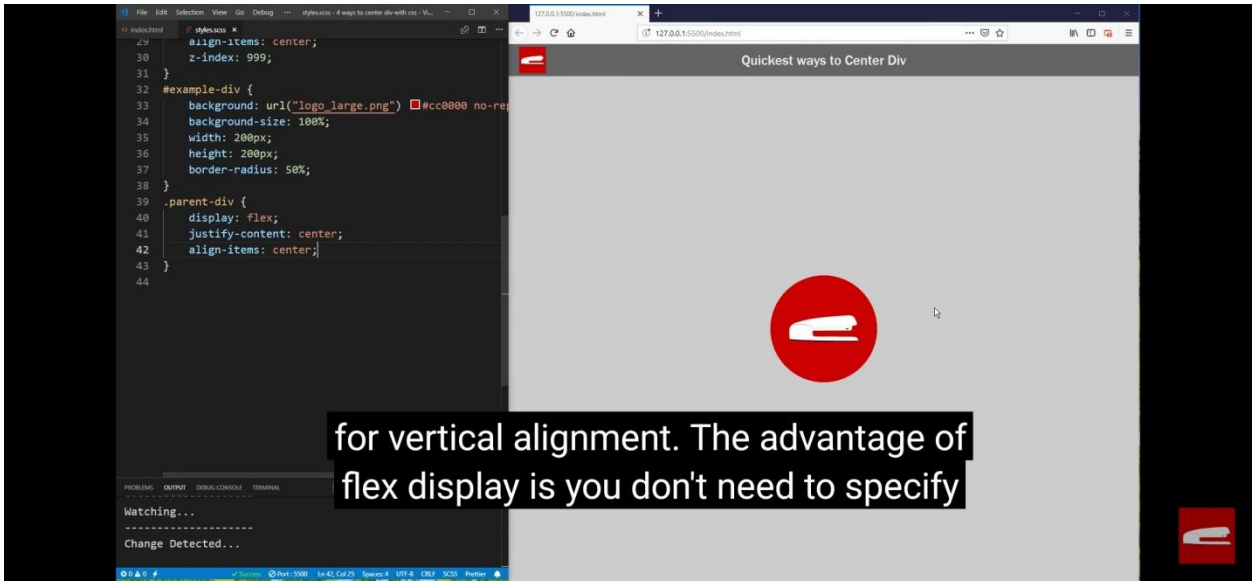


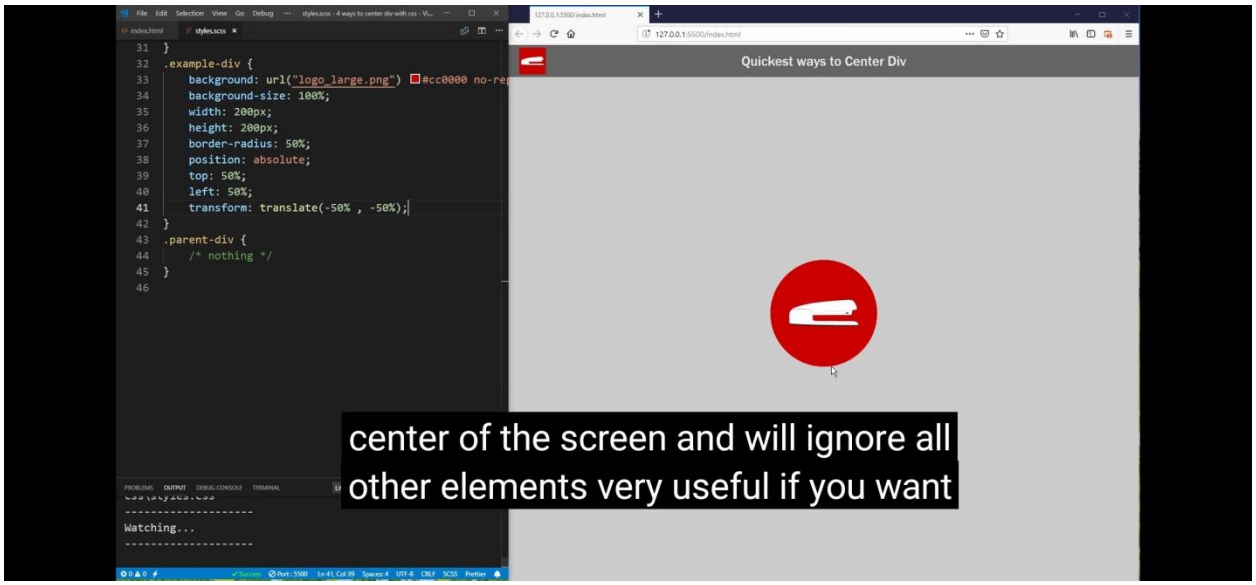
```
*{
    margin: 0; padding: 0;
    box-sizing: border-box;
}
.parent_div{
    width: 100vw;
    height: 100vh;
    background: #2980b9;
    display: flex;
    justify-content: center;
    align-items: center;

}
.child_div{
    width: 60vw;
    height: 60vh;
    background:#1abc9c;
```



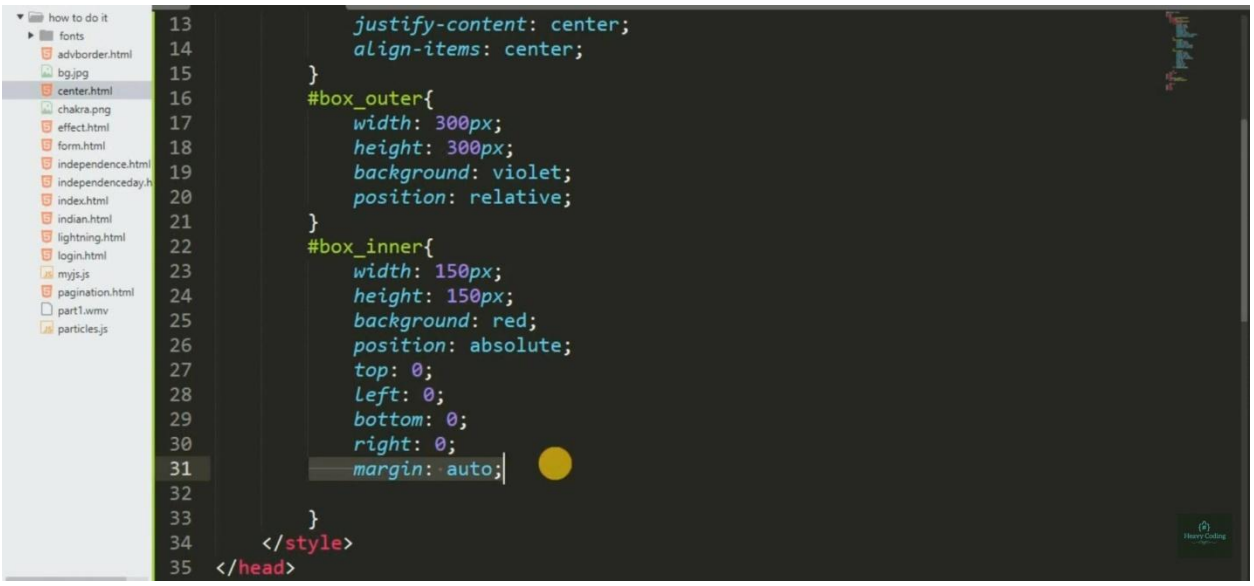
```
}
.parent_div{
    width: 100vw;
    height: 100vh;
    background: #2980b9;
    /*display: flex;
    justify-content: center;
    align-items: center;*/
    display: grid;
    place-items:center;

}
.child_div{
    width: 60vw;
    height: 60vh;
    background:#1abc9c;
    /*position: absolute;
```


C:\Users\Jatinder\Desktop\social_icons\style.css (css) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
index.html
style.css
1 *
2 {
3 margin: 0px;
4 padding: 0px;
5 }
6 html,body
7 {
8 height: 100%;
9 }
10 .withbg
11 {
12 height: 100%;
13 background-image: url("images/laptop.jpg");
14 background-position: center;
15 background-size: cover;
16 }
JP WEB TUTORIALS
Line 15, Column 28; Saved C:\Users\Jatinder\Desktop\social_icons\style.css (UTF-8)
Tab Size: 4
CSS
12:24 PM
2/1/2019

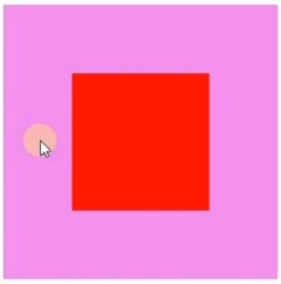

align-items: center;
z-index: 999;
}
#example-div {
background: url("logo_large.png") #cc0000 no-re
background-size: 100%;
width: 200px;
height: 200px;
border-radius: 50%;
}
.parent-div {
display: flex;
justify-content: center;
align-items: center;
}
Watching...
Change Detected...
127.0.0.1:5500/index.html
Quickest ways to Center Div
for vertical alignment. The advantage of
flex display is you don't need to specify

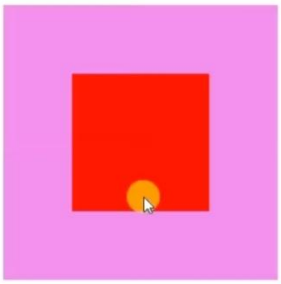

}
.example-div {
background: url("logo_large.png") #cc0000 no-re
background-size: 100%;
width: 200px;
height: 200px;
border-radius: 50%;
position: absolute;
top: 50%;
left: 50%;
transform: translate(-50% , -50%);
}
.parent-div {
/* nothing */
}
Watching...
Quickest ways to Center Div
center of the screen and will ignore all
other elements very useful if you want


how to do it
fonts
advborder.html
bg.jpg
center.html
chakra.png
effect.html
form.html
independence.html
independenceday.h
index.html
indian.html
lightning.html
login.html
myjs.js
pagination.html
part1.wmv
particles.js
<meta charset="utf-8">
<title></title>
<style type="text/css">
body,html{
width: 100%;
height: 100%;
margin: 0;
padding: 0;
display: flex;
justify-content: center;
align-items: center;
}
#box_outer{
width: 300px;
height: 300px;
background: violet;
}
</style>
</head>
<body>


how to do it
fonts
advborder.html
bg.jpg
center.html
chakra.png
effect.html
form.html
independence.html
independenceday.h
index.html
indian.html
lightning.html
login.html
myjs.js
pagination.html
part1.wmv
particles.js
13 justify-content: center;
14 align-items: center;
15 }
16 #box_outer{
17 width: 300px;
18 height: 300px;
19 background: violet;
20 position: relative;
21 }
22 #box_inner{
23 width: 150px;
24 height: 150px;
25 background: red;
26 position: absolute;
27 top: 0;
28 left: 0;
29 bottom: 0;
30 right: 0;
31 margin: auto;
32
33 }
34 </style>
35 </head>


how to do it
fonts
advborder.html
bg.jpg
center.html
chakra.png
effect.html
form.html
independence.html
independenceday.h
index.html
indian.html
lightning.html
login.html
myjs.js
pagination.html
part1.wmv
particles.js
13 justify-content: center;
14 align-items: center;
15 }
16 #box_outer{
17 width: 300px;
18 height: 300px;
19 background: violet;
20 position: relative;
21 }
22 #box_inner{
23 width: 150px;
24 height: 150px;
25 background: red;
26 position: absolute;
27 top: 50%;
28 left: 50%;
29 transform: translate(-50%,-50%);
30 }
31 </style>
32 </head>
33 <body>
34
35






how to do it
fonts
advborder.html
bg.jpg
center.html
chakra.png
effect.html
form.html
independence.html
independenceday.h
index.html
indian.html
lightning.html
login.html
myjs.js
pagination.html
part1.wmv
particles.js
justify-content: center;
align-items: center;
}
#box_outer{
width: 300px;
height: 300px;
background: violet;
display: grid;
place-content: center;
}
#box_inner{
width: 150px;
height: 150px;
background: red;
}
</style>
</head>
<body>


how to do it
fonts
advborder.html
bg.jpg
center.html
chakra.png
effect.html
form.html
independence.html
independenceday.h
index.html
indian.html
lightning.html
login.html
myjs.js
pagination.html
part1.wmv
particles.js
justify-content: center;
align-items: center;
}
#box_outer{
width: 300px;
height: 300px;
background: violet;
display: grid;
align-items: center;
justify-content: center;
}
#box_inner{
width: 150px;
height: 150px;
background: red;
}
</style>
</head>
<body>

Bootstrap 5 first alpha was officially released on June 16, 2020.

Summary of the most important changes:

- jQuery was removed
- Switch to Vanilla JavaScript
- Drop Internet Explorer 10 and 11 support
- Improved grid system
- Improved documentation
- Improved modularity
- Improved forms
- New responsive font
- New utilities & helpers
- Easier customization & theming
- Lighter package
- New API available

About CSS Flexbox :

- New module in CSS3 to easy align elements in different directions and orders
- Flexbox is to gives container the ability to expand and shrink elements to best use all the available space
- CSS Flexbox Layout replaces float layout
- New way to build one-dimensional layouts

Yahoo Baba

Justify-content Values:

Horizontal Alignment

- flex-start
- flex-end
- center
- space-around
- space-between
- space-evenly

Yahoo Baba

CSS Flex-direction values:

- row
- row-reverse
- column
- column-reverse

Yahoo Baba

Align-items Values:

Vertical Alignment

- flex-start
- flex-end
- center
- stretch
- baseline

Yahoo Baba

Rows - Grid System

Rows are in .container or .container-fluid

Columns

.col-xs-	Always	Full
.col-sm-	>768	750 wide
.col-md-	>992	970 wide
.col-lg-	>1200	1170 wide

Resets, Offsets, Pushes, Pulls

.clearfix

.col-xs-offset- (*sm, md, lg*)

Nest using a .row inside a col

.col-xs-push- (*sm, md, lg*)

.col-xs-pull- (*sm, md, lg*)

.visible-xs (*sm, md, lg*)

.hidden-xs (*sm, md, lg*)

.visible-print

.hidden-print

Grid Variables

@grid-columns:	12
@grid-gutter-width:	30px
@grid-float-breakpoint:	768px

Using Grid Mixin CSS

```
.wrapper {.make-row();}
.content-main {.make-lg-column(8);}
.content-secondary {
.make-lg-column(3);
.make-lg-column-offset(1);
}
```

Using Grid Mixin HTML

```
<div class="wrapper">
<div class="content-main">...</div>
<div class="content-secondary">...</div>
</div>
```

Buttons

```
<button type="button" class="btn btn-default">Default</button>

<a href="#" class="btn btn-primary btn-lg active" role="button">Primary link</a>
```

Button Classes

.btn-default	.btn-primary
.btn-success	.btn-info
.btn-warning	.btn-danger

.btn-link *(to look like a link)*

.btn-lg	.btn-sm
.btn-xs	.btn-block
.active (A tag)	.disabled (A tag)

disabled="disabled"

Typography

.h1	.h2	.h3
.h4	.h5	.h6
.small	<small>	.lead

@font-size-base | @line-height-base

Align

.text-left	.text-right
.text-center	.text-justify

Text Helpers

.text-muted	.text-primary
.text-success	.text-info
.text-warning	.text-danger
.bg-primary	.bg-success
.bg-info	.bg-warning
.bg-danger	
.text-hide	**.sr-only**

Lists

.list-unstyled	.list-inline

Descriptions

<dl> <dt>...</dt> <dd>...</dd> </dl>

.dl-horizontal

Images

.img-responsive (max-width:100%, height:auto)

.img-rounded

.img-circle

.img-thumbnail (double border)

Forms

NEED TO DO

Labels

.label	.label-default

primary, success, info, warning, danger

```
<span class="label label-default">New</span>
```

Badge

```
<span class="badge">42</span>
```

Alerts

.alert .alert-success

success, info, warning, or danger

.alert-dismissable

```
<button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>
```

```
<a href="#" class="alert-link">...</a>
```

Tables

.table	.table-striped
.table-bordered	.table-hover
.table-condensed	
.table-responsive	

Wrap .table in .table-responsive

```
<div class='table-responsive'>
...table
</div>
```

Rows or Cells

.active	.success	.info
.warning	.danger	

Tags

abbr

```
<abbr title="attribute">attr</abbr>
```

address

```
<address> <strong>Twitter, Inc.</strong><br> 795 Folsom Ave, Suite 600<br> San Francisco, CA 94107<br> <abbr title="Phone">P:</abbr> (123) 456-7890 </address>
```

blockquote

```
<blockquote> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer</p> </blockquote>
```

blockquote **footer**

```
<footer>Someone famous in <cite title="Source Title">Source Title</cite></footer>
```

blockquote right justify

.blockquote-reverse

code

```
<code>&lt;section&gt;</code>
```

kbd Keyboard Entry

```
<kbd>cd</kbd>
```

.pre-scrollable

Max Height 350px box, with y-axis scrollbar. Used with **pre** tag

close button

```
<button type="button" class="close" aria-hidden="true">&times;</button>
```

Caret

```
<span class="caret"></span>
```

Floats

.pull-left .pull-right

Float Mixin

.element { .pull-left(); }

Clear Floats

.clearfix

Center Block (*mixin also*)

.center-block (display:block, margin auto) | .center-block();

Show (*mixin also*)

.show | *.show();*

Hide (*mixin also*)

.hide | *.hide();*

Glyphicons

```
<span class="glyphicon glyphicon-search"></span>
```

Cheatography

Bootstrap Cheat Sheet

by John Mason (masonjo)

Default CSS Property Values Desk Reference

Default CSS Property Values

```
animation : none;
animation-delay : 0;
animation-direction : normal;
animation-duration : 0;
animation-fill-mode : none;
animation-iteration-count : 1;
animation-name : none;
animation-play-state : running;
animation-timing-function : ease;
backface-visibility : visible;
background : 0;
background-attachment : scroll;
background-clip : border-box;
background-color : transparent;
background-image : none;
background-origin : padding-box;
background-position : 0 0;
background-position-x : 0;
background-position-y : 0;
background-repeat : repeat;
background-size : auto auto;
border : 0;
border-style : none;
border-width : medium;
border-color : inherit;
border-bottom : 0;
border-bottom-color : inherit;
border-bottom-left-radius : 0;
border-bottom-right-radius : 0;
border-bottom-style : none;
border-bottom-width : medium;
border-collapse : separate;
border-image : none;
border-left : 0;
```

CSS Vendor Prefixes

`-ms-`
Microsoft

`mso-`
Microsoft Office

`-moz-`
Mozilla Foundation (Gecko-based browsers)

`-o-, -xv-`
Opera Software

`-atsc-`
Advanced Television Standards Committee

`-wap-`
The WAP Forum

`-webkit-`
Safari, Chrome (and other WebKit-based browsers)

`-khtml-`
Konqueror browser

`-apple-`
Webkit supports properties using the `-apple-` prefixes as well

`prince-`
YesLogic

`-ah-`
Antenna House

`-hp-`
Hewlett Packard

`-ro-`
Real Objects

`-rim-`
Research In Motion

`-tc-`
Tall Components

bootstrap creative

Free Bootstrap Video Tutorial Course
BootstrapCreative.com/bootstrap-tutorial/

Default CSS Property Values Desk Reference

Default CSS Property Values cont.

```
border-left-color : inherit;
border-left-style : none;
border-left-width : medium;
border-radius : 0;
border-right : 0;
border-right-color : inherit;
border-right-style : none;
border-right-width : medium;
border-spacing : 0;
border-top : 0;
border-top-color : inherit;
border-top-left-radius : 0;
border-top-right-radius : 0;
border-top-style : none;
border-top-width : medium;
bottom : auto;
box-shadow : none;
box-sizing : content-box;
caption-side : top;
clear : none;
clip : auto;
color : inherit;
columns : auto;
column-count : auto;
column-fill : balance;
column-gap : normal;
column-rule : medium none currentColor;
column-rule-color : currentColor;
column-rule-style : none;
column-rule-width : none;
column-span : 1;
column-width : auto;
content : normal;
counter-increment : none;
```

CSS Selectors

```
.class
#id
*
element
element,element
element element
element>element
element+element
element1~element2
[attribute]
[attribute=value]
[attribute~=value]
[attribute|=value]
[attribute^=value]
[attribute$=value]
[attribute*=value]
:active
::after
::before
:checked
:disabled
:empty
:enabled
:first-child
::first-letter
::first-line
:first-of-type
:focus
:hover
:in-range
:invalid
:lang(language)
:last-child
:last-of-type
:link
:not(selector)
:nth-child(n)
:nth-last-child(n)
:nth-last-of-type(n)
:nth-of-type(n)
:only-of-type
:only-child
:optional
:out-of-range
:read-only
:read-write
:required
:root
::selection
:target
:valid
:visited
```

bootstrap creative

Free Bootstrap Video Tutorial Course
BootstrapCreative.com/bootstrap-tutorial/

Default CSS Property Values Desk Reference

Default CSS Property Values cont.

```
counter-reset : none;
cursor : auto;
direction : ltr;
display : inline;
empty-cells : show;
float : none;
font : normal;
font-family : inherit;
font-size : medium;
font-style : normal;
font-variant : normal;
font-weight : normal;
height : auto;
hyphens : none;
left : auto;
letter-spacing : normal;
line-height : normal;
list-style : none;
list-style-image : none;
list-style-position :
outside;
list-style-type : disc;
margin : 0;
margin-bottom : 0;
margin-left : 0;
margin-right : 0;
margin-top : 0;
max-height : none;
max-width : none;
min-height : 0;
min-width : 0;
opacity : 1;
orphans : 0;
outline : 0;
outline-color : invert;
outline-style : none;
outline-width : medium;
overflow : visible;
overflow-x : visible;
overflow-y : visible;
padding : 0;
padding-bottom : 0;
padding-left : 0;
padding-right : 0;
padding-top : 0;
page-break-after : auto;
page-break-before : auto;
page-break-inside : auto;
perspective : none;
perspective-origin : 50%
50%;
position : static;
/* May need to alter quotes
for different locales (e.g
fr) */
quotes : '\201C' '\201D'
'\2018' '\2019';
right : auto;
tab-size : 8;
table-layout : auto;
text-align : inherit;
text-align-last : auto;
text-decoration : none;
text-decoration-color :
inherit;
text-decoration-line : none;
text-decoration-style :
solid;
text-indent : 0;
text-shadow : none;
text-transform : none;
top : auto;
transform : none;
transform-style : flat;
transition : none;
transition-delay : 0s;
transition-duration : 0s;
transition-property : none;
transition-timing-function :
ease;
unicode-bidi : normal;
vertical-align : baseline;
visibility : visible;
white-space : normal;
widows : 0;
width : auto;
word-spacing : normal;
z-index : auto;
```

bootstrap creative

Free Bootstrap Video Tutorial Course

BootstrapCreative.com/bootstrap-tutorial/

Bootstrap 3 CSS Classes Desk Reference

Grid

Basic grid - full width is 12 columns wide

```
<!-- change .container to .container-fluid to be full width -->
<div class="container">
 <!-- Columns are always 50% wide, on mobile and desktop -->
 <div class="row">
   <div class="col-xs-6">.col-xs-6</div>
   <div class="col-xs-6">.col-xs-6</div>
 </div>
 <!-- nested columns example -->
 <div class="row">
   <div class="col-xs-6">.col-xs-6</div>
   <div class="col-xs-6">.col-xs-6
    <div class="row">
      <div class="col-md-6">100% mobile 50% everywhere else</div>
      <div class="col-md-6">100% mobile 50% everywhere else</div>
    </div>
   </div>
 </div>
</div>
```

Media queries

```
/* Extra small devices (phones, less than 768px) No media query since this is the default in Bootstrap */
/* small (tablets, 768px and up) */
@media (min-width: @screen-sm-min) { ... }
/* medium (desktops, 992px and up) */
@media (min-width: @screen-md-min) { ... }
/* large (large desktops, 1200px and up) */
@media (min-width: @screen-lg-min) { ... }
```

Text & Images

`.text-left` Left aligned text

`.text-center` Center aligned text

`.text-right` Right aligned text

`.text-justify` Justified text

`.text-nowrap` No wrap text

`.text-lowercause` Lowercase text

`.text-uppercase` Uppercase text

`.text-capitalize` Capitalized text

`.lead` Good for first paragraph of article

`.list-unstyled` Removes default list margin/ padding

`.list-inline` Makes list items inline

`.dl-horizontal` Makes list items two columns

`.img-responsive` Make an image responsive

`.img-rounded` Adds rounded corners to image

`.img-circle` Crops image to be circle

`.img-thumbnail` Adds rounded corner and border to an image

`.pull-left` Floats item left

`.pull-right` Floats item right

`.center-block` Set an elemennt to block with auto left and right margin

`.clearfix` Clear floats by adding this class to the parent container

Blockquote

```
<blockquote><p>Lorem ipsum dolor</p>
<footer>Someone famous in <cite title="Source Title">Source Title</cite></ footer></blockquote>
```

Headings

```
<h1>h1. Bootstrap heading <small>Secondary text</small></h1>
<p class="h1">Paragraph that looks like heading</p>
```

bootstrap creative

Free Bootstrap Video Tutorial Course
BootstrapCreative.com/bootstrap-tutorial/

Navbar

```
<!-- Fixed top navbar with brand as logo image tags -->
<nav class="navbar navbar-default navbar-fixed-top">
  <div class="container-fluid">
    <!-- Brand and toggle get grouped for better mobile display -->
    <div class="navbar-header">
      <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-
target="#bs-example-navbar-collapse-1" aria-expanded="false">
        <span class="sr-only">Toggle navigation</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <a class="navbar-brand" href="#"><img alt="Brand" src="..."></a>
    </div>
    <!-- Collect the nav links, forms, and other content for toggling -->
    <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
      <ul class="nav navbar-nav">
        <li class="active"><a href="#">Link <span class="sr-only">(current)</span></a></
li>
        <li><a href="#">Link</a></li>
        <li class="dropdown">
          <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-
haspopup="true" aria-expanded="false">Dropdown <span class="caret"></span></a>
          <ul class="dropdown-menu">
            <li><a href="#">Action</a></li>
            <li role="separator" class="divider"></li>
            <li><a href="#">Separated link</a></li>
          </ul>
        </li>
      </ul>
    </div><!-- /.navbar-collapse -->
  </div><!-- /.container-fluid -->
</nav>
```

Bootstrap 3 CSS Classes Desk Reference

Forms

```
<form>
  <div class="form-group">
    <label for="exampleInputEmail1">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail1" placeholder="Email">
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">Password</label>
    <input type="password" class="form-control" id="exampleInputPassword1"
placeholder="Password">
  </div>
  <div class="form-group">
    <label for="exampleInputFile">File input</label>
    <input type="file" id="exampleInputFile">
    <p class="help-block">Example block-level help text here.</p>
  </div>
  <div class="checkbox">
    <label>
      <input type="checkbox"> Check me out
    </label>
  </div>
  <button type="submit" class="btn btn-default">Submit</button>
</form>
```

Buttons

.btn Needs to be added to all buttons because it adds padding and margin
.btn-default The default button style
.btn-primary The button that has the primary action in a group
.btn-success Could be used on the last submit button in a form
.btn-info Informational button
.btn-link Removes background color and add text color
.btn-lg Large buttom
.btn-sm Smaller than default button
.btn-xs Even smaller
.btn-block Button that spans full width of parent

```
<a class="btn btn-default"
href="#" role="button">Link</a>
<button class="btn btn-primary"
type="submit">Button</button>
```

bootstrap creative

Free Bootstrap Video Tutorial Course
BootstrapCreative.com/bootstrap-tutorial/

Carousel

```
<div id="carousel-example-generic" class="carousel slide" data-ride="carousel">
  <!-- Indicators -->
  <ol class="carousel-indicators">
    <li data-target="#carousel-example-generic" data-slide-to="0" class="active"></li>
    <li data-target="#carousel-example-generic" data-slide-to="1"></li>
  </ol>
  <!-- Wrapper for slides -->
  <div class="carousel-inner" role="listbox">
    <div class="item active">
      <img src="..." alt="...">
      <div class="carousel-caption">
        ...
      </div>
    </div>
    <div class="item">
      <img src="..." alt="...">
      <div class="carousel-caption">
        ...
      </div>
    </div>
    ...
  </div>
  <!-- Controls -->
  <a class="left carousel-control" href="#carousel-example-generic" role="button" data-
slide="prev">
    <span class="glyphicon glyphicon-chevron-left" aria-hidden="true"></span>
    <span class="sr-only">Previous</span>
  </a>
  <a class="right carousel-control" href="#carousel-example-generic" role="button" data-
slide="next">
    <span class="glyphicon glyphicon-chevron-right" aria-hidden="true"></span>
    <span class="sr-only">Next</span>
  </a>
</div>
```

Jumbotron

```
<div class="jumbotron">
  <h1>Hello, world!</h1>
  <p>...</p>
  <p><a class="btn btn-primary btn-lg" href="#" role="button">Learn more</a></p>
</div>
```

To make the jumbotron full width, and without rounded corners, place it outside all .containers and instead add a .container within.

```
<div class="jumbotron">
  <div class="container">
    ...
  </div>
</div>
```

Page header

```
<div class="page-header">
  <h1>Example page header <small>Subtext for header</small></h1>
</div>
```

Breadcrumbs

```
<ol class="breadcrumb">
  <li><a href="#">Home</a></li>
  <li><a href="#">Library</a></li>
  <li class="active">Data</li>
</ol>
```

Responsive embed

```
<!-- 16:9 aspect ratio - change aspect ratio by replacing 16by9 with 4by3 -->
<div class="embed-responsive embed-responsive-16by9">
  <iframe class="embed-responsive-item" src="..."></iframe>
</div>
```

Tables

```
<!-- Responsive table with all of the options applied  -->
<div class="table-responsive">
  <table class="table table-condensed table-hover table-bordered table-striped">
 <tr class="active">...</tr>
 <tr>
   <td class="info">...</td>
 </tr>
  </table>
</div>
```

Alphabetical Index of CSS Classes

.active
.affix
.alert
.alert-danger
.alert-dismissible
.alert-info
.alert-link
.alert-success
.alert-warning
.arrow
.badge
.bg-danger
.bg-info
.bg-primary
.bg-success
.bg-warning
.bottom
.breadcrumb
.btn
.btn-block
.btn-danger
.btn-default
.btn-group
.btn-group-justified
.btn-group-vertical
.btn-info
.btn-link
.btn-primary
.btn-sm
.btn-success
.btn-toolbar
.btn-warning
.btn-xs
.caption
.caret
.carousel
.carousel-caption
.carousel-control
.carousel-indicators
.carousel-inner
.center-block
.checkbox
.checkbox-inline
.close
.col-lg-* /*(1-12)*/
.col-lg-offset-* /*(0-12)*/
.col-lg-pull-* /*(0-12)*/
.col-lg-push-* /*(0-12)*/
.col-md-* /*(1-12)*/
.col-md-offset-* /*(0-12)*/
.col-md-pull-* /*(0-12)*/
.col-md-push-* /*(0-12)*/
.col-sm-* /*(1-12)*/
.col-sm-offset-* /*(0-12)*/
.col-sm-pull-* /*(0-12)*/
.col-sm-push-* /*(0-12)*/
.col-xs-* /*(1-12)*/
.col-xs-offset-* /*(0-12)*/
.col-xs-pull-* /*(0-12)*/
.col-xs-push-* /*(0-12)*/

bootstrap creative

Free Bootstrap Video Tutorial Course
BootstrapCreative.com/bootstrap-tutorial/

Bootstrap 3 CSS Classes Desk Reference

.collapse
.collapsing
.container
.container-fluid
.control-label
.divider
.dropdown
.dropdown-backdrop
.dropdown-header
.dropdown-menu
.dropdown-menu-left
.dropdown-menu-right
.dropdown-toggle
.embed-responsive
.embed-responsive-16by9
.embed-responsive-4by3
.fade
.form-control
.form-control-feedback
.form-control-static
.form-group
.glyphicon
.glyphicon-chevron-left
.glyphicon-chevron-right
.h1
.h2
.h3
.h4
.h5
.h6
.has-feedback
.help-block
.hidden
.hidden-lg
.hidden-md
.hidden-print
.hidden-sm
.hidden-xs
.hide
.icon-bar
.icon-next
.icon-prev
.img-circle
.img-rounded
.img-thumbnail
.in
.initialism
.input-group
.input-group-addon
.input-group-btn
.input-lg
.input-sm
.invisible
.item
.jumbotron
.label
.label-danger
.label-default
.label-info
.label-primary
.label-success
.label-warning
.lead
.left
.list-group
.list-group-item
.list-group-item-danger
.list-group-item-heading
.list-group-item-info
.list-group-item-success
.list-group-item-text
.list-group-item-warning
.list-inline
.list-unstyled
.mark
.media
.media-body
.media-heading
.media-list
.media-object
.modal
.modal-backdrop
.modal-body
.modal-content
.modal-dialog
.modal-footer
.modal-header
.modal-lg
.modal-open
.modal-scrollbar-measure
.modal-sm
.modal-title
.nav
.nav-divider
.nav-justified
.nav-tabs
.nav-tabs-justified
.navbar
.navbar-brand

bootstrap creative

Free Bootstrap Video Tutorial Course
BootstrapCreative.com/bootstrap-tutorial/

Bootstrap 3 CSS Classes Desk Reference

.navbar-btn
.navbar-collapse
.navbar-default
.navbar-fixed-bottom
.navbar-fixed-top
.navbar-form
.navbar-header
.navbar-inverse
.navbar-left
.navbar-link
.navbar-nav
.navbar-right
.navbar-static-top
.navbar-text
.navbar-toggle
.next
.page-header
.pager
.pagination
.panel
.panel-body
.panel-danger
.panel-default
.panel-footer
.panel-group
.panel-heading
.panel-info
.panel-primary
.panel-success
.panel-title
.panel-warning
.popover
.popover-content
.popover-title
.pre-scrollable
.prev
.progress
.progress-bar
.progress-bar-danger
.progress-bar-info
.progress-bar-striped
.progress-bar-success
.progress-bar-warning
.pull-left
.pull-right
.right
.row
.show
.small
.sr-only
.tab-pane
.table
.table-bordered
.table-responsive
.text-capitalize
.text-center
.text-danger
.text-hide
.text-info
.text-justify
.text-left
.text-lowercase
.text-muted
.text-nowrap
.text-primary
.text-right
.text-success
.text-uppercase
.text-warning
.thumbnail
.tooltip
.tooltip-arrow
.tooltip-inner
.top
.visible-lg
.visible-lg-block
.visible-lg-inline
.visible-lg-inline-block
.visible-md
.visible-md-block
.visible-md-inline
.visible-md-inline-block
.visible-print
.visible-print-block
.visible-print-inline
.visible-print-inline-block
.visible-sm
.visible-sm-block
.visible-sm-inline
.visible-sm-inline-block
.visible-xs
.visible-xs-block
.visible-xs-inline
.visible-xs-inline-block
.well
.well-lg
.well-sm

bootstrap creative

Free Bootstrap Video Tutorial Course
BootstrapCreative.com/bootstrap-tutorial/

CSS Cheat Sheet http://w3schools.com

Selectors

*	All elements
p	Selects all elements
div *	All elements within
div p	<p> within
div, p	and <p>
div > p	<p> with parent
div + p	<p> preceded by
.class	Elements of class "class"
div.class	of class "class"
#itemid	Element with id "itemid"
div#itemid	with id "itemid"
a[attr]	<a> with attribute "attr"
a[attr="x"]	<a> when "attr" is "x"
a[class~="x"]	<a> when class is a list containing "x"
a[lang|="en"]	<a> when lang begins "en"

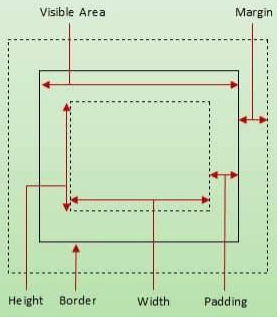
Pseudo-Selectors and Pseudo-Classes

:first-child	First child element
:first-line	First line of element
:first-letter	First letter of element
:hover	Element with mouse over
:active	Active element
:focus	Element with focus
:link	Unvisited links
:visited	Visited links
:lang(var)	Element with language "var"
:before	Before element
:after	After element

Positioning

display	clear
position	z-index
top	direction
right	unicode-bidi
bottom	overflow
left	clip
float	visibility

Box Model



Sizes and Colours

0	0 requires no unit
Relative Sizes	
em	1em equal to font size of parent (same as 100%)
ex	Height of lower case "x"
Percentage	Percentage
Absolute Sizes	
px	Pixels
cm	Centimeters
mm	Millimeters
in	Inches
pt	1pt = 1/72in
pc	1pc = 12pt
Colours	
#789abc	
#acf	RGB Hex Notation
rgb(0,25,50)	Equates to "#aaccff" Value of each of red, green and blue. 0 to 255, may be swapped for percentages.

Color / Background

color	background-repeat
background	background-image
background-color	background-position
background-attachment	

Dimensions

width	min-height
min-width	max-height
max-width	vertical-align
height	

Text

text-indent	word-spacing
text-align	text-transform
text-decoration	white-space
text-shadow	line-height
letter-spacing	

Boxes

margin	border-color
margin-top	border-top-color
margin-bottom	border-bottom-color
margin-right	border-left-color
margin-left	border-right-color
padding	border-style
padding-top	border-top-style
padding-bottom	border-bottom-style
padding-left	border-left-style
padding-right	border-right-style
border	border-width
border-top	border-top-width
border-bottom	border-bottom-width
border-left	border-left-width
border-right	border-right-width

Font

font	font-weight
font-family	font-stretch
font-style	font-size
font-variant	font-size-adjust

Tables

caption-side	border-spacing
table-layout	empty-cells
border-collapse	speak-header

Paging

page-break-before	page-break-inside
page-break-after	

Interface

cursor	outline-width
outline	outline-style
	outline-color

Aural

volume	elevation
speak	speech-rate
pause	voice-family
pause-before	pitch
pause-after	pitch-range
cue	stress
cue-before	richness
cue-after	speak-punctuation
play-during	speak-numeral
azimuth	

Miscellaneous

content	list-style
quotes	list-style-type
counter-reset	list-style-image
counter-increment	list-style-position

Resource

Online
Official Website

Download
CSS in one page [html] (css.su)
CSS Cheat Sheet [.png]
CSS Cheat Sheet [.pdf]
CSS Level 1 Quick Reference D eepX Ltd [.pdf]
CSS Level 2 Quick Reference B enjamin Jung [.pdf]

Related
Blueprint
HTML DOM
HTML
JavaScript
Selenium
XHTML
XPath

Page

Content for Page Media
bookmark-label
bookmark-level
bookmark-target
float-offset
hyphenate-after
hyphenate-before
hyphenate-character
hyphenate-lines
hyphenate-resource
hyphens
image-resolution
marks
string-set

Generated Content

Box Model

Margin
margin
margin-bottom
margin-left
margin-right
margin-top

Padding
padding
padding-bottom
padding-left
padding-right
padding-top

Dimension
height
width
max-height
max-width
min-height
min-width

Border and Outline
border
border-bottom
border-bottom-color
border-bottom-style
border-bottom-width
border-color
border-left
border-left-color
border-left-style
border-left-width
border-right
border-right-color
border-right-style
border-right-width
border-style
border-top
border-top-color
border-top-style
border-top-width
border-width
outline

Selectors

Basic Selectors
.class
#id
*
element
element,element
element element
element>element
element+element
[*attribute*]
[*attribute=value*]
[*attribute~=value*]
[*attribute|=value*]
element1~element2
[*attribute^=value*]
[*attribute$=value*]
[*attribute*=value*]

Pseudo-Selectors
:link
:visited
:active
:hover
:focus
:first-letter
:first-line
:first-child
:before
:after
:lang(*language*)
:first-of-type
:last-of-type
:only-of-type
:only-child
:nth-child(*n*)
:nth-last-child(*n*)
:nth-of-type(*n*)
:last-child
:root
:empty
:target
:enabled
:disabled
:checked

Style

Background
background
background-attachment
background-color
background-image
background-position
background-repeat
background-clip
background-origin
background-size

Color
color-profile
opacity
rendering-intent

Font
font
font-family
font-size
font-style
font-variant
font-weight
@font-face
font-size-adjust
font-stretch

Positioning
bottom
clear
clip
cursor
display
float
left
overflow
position
right
top
visibility
z-index

Elements

Hyperlink
target
target-name
target-new
target-position

Text
color
direction
letter-spacing
line-height
text-align
text-decoration
text-indent
text-transform
unicode-bidi
vertical-align
white-space
word-spacing
text-outline
text-overflow
text-shadow
text-wrap
word-break
word-wrap

List
list-style
list-style-image
list-style-position
list-style-type

Table
border-collapse
border-spacing
caption-side
empty-cells
table-layout

Marquee
marquee-direction
marquee-play-count
marquee-speed

Cascading Style Sheets (CSS 3)

BACKGROUND	
background	*background-image* *background-position* *background-size* *background-repeat* *background-attachment* *background-origin* *background-clip* *background-color*
background-attachment	scroll | fixed
background-break	bounding-box | each-box | continuous
background-clip	*length* % border-box | padding-box | content-box | no-clip
background-color	*color* transparent
background-image	*url* none
background-origin	border-box | padding-box | content-box
background-position	top left | top center | top right | center left | center center | center right | bottom left | bottom center | bottom right *x-% y-%* *x-pos y-pos*
background-repeat	repeat | repeat-x | repeat-y | no-repeat
background-size	*length* % auto | cover | contain

BORDER	
border	*border-width* *border-style* *border-color*
border-break	*border-width* *border-style* *color* close
border-bottom	*border-bottom-width* *border-style* *border-color*
border-bottom-color	*border-color*
border-bottom-style	*border-style*
border-bottom-width	thin | medium | thick *length*
border-collapse	collapse | separate
border-color	*color*
border-image	*image* [*number* / % *border-width* stretch | repeat | round] none
border-left	*border-left-width* *border-style* *border-color*
border-left-color	*border-color*
border-left-style	*border-style*
border-left-width	thin | medium | thick *length*
border-right	*border-right-width* *border-style* *border-color*
border-right-color	*border-color*
border-right-style	*border-style*
border-right-width	thin | medium | thick *length*

BORDER	
border-top	*border-top-width* *border-style* *border-color*
border-top-color	*border-color*
border-top-style	*border-style*
border-top-width	thin | medium | thick *length*
border-width	thin | medium | thick *length*
border-radius	*border-top-right-radius* *border-bottom-right-radius* *border-bottom-left-radius* *border-top-left-radius*
border-top-right-radius	*length*
border-bottom-right-radius	*length*
border-bottom-left-radius	*length*
border-top-left-radius	*length*
box-shadow	inset || [*length*, *length*, *length*, *length* || <*color*>] none
border-style	none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset

FONT	
font	*font-style* *font-variant* *font-weight* *font-size/line-height* *font-family* caption | icon | menu | message-box | small-caption | status-bar
font-family	*family-name* *generic-family* inherit
font-size	xx-small | x-small | small | medium | large | x-large | xx-large | smaller | larger | inherit *length* %
font-size-adjust	none| inherit *number*
font-stretch	normal | wider | narrower | ultra-condensed | extra-condensed | condensed | semi-condensed | semi-expanded | expanded | extra-expanded | ultra-expanded | inherit
font-style	normal | italic | oblique | inherit
font-variant	normal | small-caps | inherit
font-weight	normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | inherit

BOX MODEL	
clear	left | right | both | none
display	none | inline | block | inline-block | list-item | run-in | compact | table | inline-table | table-row-group | table-header-group | table-footer-group | table-row | table-column-group | table-column | table-cell | table-caption | ruby | ruby-base | ruby-text | ruby-base-group | ruby-text-group

BOX MODEL	
float	left | right | none
height	auto *length* %
max-height	none *length* %
max-width	none *length* %
min-height	none | inherit *length* %
min-width	none | inherit *length* %
width	auto % *length*
margin	*margin-top* *margin-right* *margin-bottom* *margin-left*
margin-bottom	auto *length* %
margin-left	auto *length* %
margin-right	auto *length* %
margin-top	auto *length* %
padding	*padding-top* *padding-right* *padding-bottom* *padding-left*
padding-bottom	*length* %
padding-left	*length* %
padding-right	*length* %
padding-top	*length* %
marquee-direction	forward | reverse
marquee-loop	infinite *number*
marquee-play-count	infinite *integer*
marquee-speed	slow | normal | fast
marquee-style	scroll | slide | alternate
overflow	visible | hidden | scroll | auto | no-display | no-content *overflow-x* *overflow-y*
overflow-style	auto | marquee-line | marquee-block
overflow-x	visible | hidden | scroll | auto | no-display | no-content
overflow-y	visible | hidden | scroll | auto | no-display | no-content
rotation	*angle*
rotation-point	*position (paired value offset)*
visibility	visible | hidden | collapse

©2009 Veign, All Rights Reserved

<html>

Document Outline

<!DOCTYPE>	Version of (X)HTML
<html>	HTML document
<head>	Page information
<body>	Page contents

Comments

<!-- Comment Text -->

Page Information

<base />	Base URL
<meta />	Meta data
<title>	Title
<link />	Relevant resource
<style>	Style resource
<script>	Script resource

Document Structure

<h[1-6]>	Heading
	Page section
	Inline section
<p>	Paragraph
 	Line break
<hr />	Horizontal rule

Links

<a href="">	Page link
<a href="mailto:">	Email link
<a name="name">	Anchor
<a href="#name">	Link to anchor

Text Markup

	Strong emphasis
	Emphasis
<blockquote>	Long quotation
<q>	Short quotation
<abbr>	Abbreviation
<acronym>	Acronym
<address>	Address
<pre>	Pre-formatted text
<dfn>	Definition
<code>	Code
<cite>	Citation
<del>	Deleted text
<ins>	Inserted text
<sub>	Subscript
<sup>	Superscript
<bdo>	Text direction

Lists

	Ordered list
	Unordered list
	List item
<dl>	Definition list
<dt>	Definition term
<dd>	Term description

Forms

<form>	Form
<fieldset>	Collection of fields
<legend>	Form legend
<label>	Input label
<input />	Form input
<select>	Drop-down box
<optgroup>	Group of options
<option>	Drop-down options
<textarea>	Large text input
<button>	Button

Tables

<table>	Table
<caption>	Caption
<thead>	Table header
<tbody>	Table body
<tfoot>	Table footer
<colgroup>	Column group
<col />	Column
<tr>	Table row
<th>	Header cell
<td>	Table cell

Images and Image Maps

<img />	Image
<map>	Image Map
<area />	Area of Image Map

Common Character Entities

"	"	Quotation mark
&	&	Ampersand
<	<	Less than
>	>	Greater than
@	@	"At" symbol
€	€	Euro
•	•	Small bullet
™	™	Trademark
£	£	Pound
		Non-breaking space
©	©	Copyright symbol

Objects

<object>	Object
<param />	Parameter

Empty Elements

<area />	<img />
<base />	<input />
 	<link />
<col />	<meta />
<hr />	<param />

Core Attributes

class	style
id	title

Note: Core Attributes may not be used in base, head, html, meta, param, script, style or title elements.

Language Attributes

dir	lang

Note: Language Attributes may not be used in base, br, frame, frameset, hr, iframe, param or script elements.

Keyboard Attributes

accesskey	tabindex

Window Events

onLoad	onUnload

Form Events

onBlur	onReset
onChange	onSelect
onFocus	onSubmit

Keyboard Events

onKeydown	onKeyup
onKeypress	

Mouse Events

onClick	onMouseout
onDblclick	onMouseover
onMousedown	onMouseup
onMousemove	

Available free from AddedBytes.com

HTML Cheat Sheet [TAGS]

New [tags added in HTML5]

<article>	self-contained composition that is independently distributable
<aside>	section of page that consists of content tangentially related to content around it
<audio>	sound content
<bdi>	span of text to be isolated from surroundings for bidirectional formatting purposes
<canvas>	area that can be used to draw graphics via JavaScript
<command>	user invokable command
<datalist>	dropdown list
<datatemplate>	data template
<details>	details of an element
<embed>	embedded content
<figcaption>	caption of figure element
<figure>	group of media content
<footer>	footer for section or page
<header>	header for section or page
<hgroup>	group of headings for section
<keygen>	generated key in a form
<mark>	marked text
<meter>	measurement in defined range
<nav>	navigation links
<output>	represents results of calculation
<progress>	progress of any kind of task
<rp>	parenthesized ruby text
<rt>	ruby text
<ruby>	ruby annotations
<section>	section in a document
<source>	media resources
<summary>	header of a detail element
<time>	date/time
<video>	video
<wbr>	possible line break

Old [unsupported tags]

<acronym>	acronym
<applet>	applet
<basefont>	base font
<bgsound>	background sound
<big>	big text
	centered text
<fn>	footnotes
<font>	text font, size, and color
<frame>	sub window
<frameset>	set of frames
<isindex>	provides searchable index related to current document
<dir>	directory list
<noembed>	no embed section
<noframes>	no frame section
<s>	strikethrough text
<strike>	strikethrough text
<tt>	teletype text
<u>	underlined text
<xmp>	preformatted text

Existing [tags in HTML4 & 5]

<!--...-->	comment
<!doctype>	document type
<a>	hyperlink
<abbr>	abbreviation
<address>	address element
<area>	image map area
<b>	bold text
<base>	base URL for all links in page relative to document root
<bdo>	text direction
<blockquote>	long quotation
<body>	body element
 	single line break
<button>	push button
<caption>	table caption
<cite>	citation
<code>	code text
<col>	attributes for columns
<colgroup>	groups of columns
<dd>	definition description
<del>	deleted text
	generic block-level element
<dfn>	defining instance of a term
<dl>	definition list
<dt>	definition term
	emphasized text
<fieldset>	logically group items in a form
<form>	defines a form
<h1> to <h6>	header 1 to header 6
<head>	document information
<hr>	horizontal rule
<html>	html document
<i>	italic text
<iframe>	inline sub window
<img>	image
<input>	input field
<ins>	inserted text
<kbd>	keyboard text
<label>	label for a form control
<legend>	title in a fieldset
	list item
<link>	resource reference
<map>	image map
<menu>	menu list
<meta>	meta information
<noscript>	no script section
<object>	embedded object
	ordered list
<optgroup>	option group
<option>	option in a drop-down list
<p>	paragraph
<param>	parameter for an object
<pre>	preformatted object
<q>	short quotation
<samp>	sample computer code
<script>	script
<select>	selectable list
<small>	small text
	inline generic container
	strong text
<style>	style definition
<sub>	subscripted text
<sup>	superscripted text
<table>	table
<tbody>	table body
<td>	table cell
<textarea>	text area
<tfoot>	table footer
<th>	table header
<thead>	wraps row containing table headers
<title>	document title
<tr>	table row
	unordered list
<var>	variable

Brought to you by: inmotion hosting

HTML 5 NEW TAG

TAG NOT SUPPORTED IN HTML 5

<!--...-->	*Define a comment*
<!DOCTYPE>	*Defines the document type*
<a>	*Defines a hyperlink* href, hreflang, media, ping , rel, target, type
<abbr>	*Defines an abbreviation*
<acronym>	*Used to define an embedded acronyms*
<address>	*Defines an address element*
<applet>	*Used to define an embedded applet*
<area>	*Defines an area inside an image map* alt, coords, href, hreflang, media, ping, rel, shape, target, type
<article>	*Defines an article* cite, pubdate
<aside>	*Defines content aside from the page content*
<audio>	*Defines sound content* autobuffer, autoplay, controls, src
<b>	*Defines bold text*
<base>	*Defines a base URL for all the links in a page* href, target
<basefont>	*Used to define a default font-color, font-size, or font-family for all the document*
<bdo>	*Defines the direction of text display* dir
<big>	*Used to make text bigger*
<blockquote>	*Defines a long quotation* cite
<body>	*Defines the body element*
 	Inserts a single line break
<button>	*Defines a push button* autofocus, disabled, form, formaction, formenctype, formmethod, formnovalidate, formtarget, name, type, value
<canvas>	*Defines graphics* height, width
<caption>	*Defines a table caption*
	Used to center align text and content
<cite>	*Defines a citation*
<code>	*Defines computer code text* autobuffer, autoplay, controls, src
<col>	*Defines attributes for table columns*
<colgroup>	*Defines groups of table columns* span
<command>	*Defines a command button* checked, disabled, icon, label, radiogroup, type
<datalist>	*Defines a dropdown list*
<dd>	*Defines a definition description*
<del>	*Defines deleted text* cite, datetime
<details>	*Defines details of an element* open
<dialog>	*Defines a dialog (conversation)*
<dfn>	*Defines a definition term*
<dir>	*Used to define a directory list*
	Defines a section in a document
<dl>	*Defines a definition list*
<dt>	*Defines a definition term*
	Defines emphasized text
<embed>	*Defines external interactive content or plugin* height, src, type, width
<fieldset>	*Defines a fieldset* disabled, form, name
<figure>	*Defines a group of media content, and their caption*
<font>	*Used to define font face, font size, and font color of text*
<footer>	*Defines a footer for a section or page*
<form>	*Defines a form* accept-charset, action, autocomplete, enctype, method, name, novalidate, target
<frame>	*Used to define one particular window (frame) within a frameset*
<frameset>	*Used to define a frameset, which organized multiple windows (frames)*
<h1> to <h6>	*Defines header 1 to header 6*
<head>	*Defines information about the document*
<header>	*Defines a header for a section or page*
<hgroup>	*Defines information about a section in a document*
<hr>	*Defines a horizontal rule*
<html>	*Defines an html document* manifest, xmlns
<i>	*Defines italic text*
<iframe>	*Defines an inline sub window* height, name, sandbox, seamless, src, width
<img>	*Defines an image* alt, src, height, ismap, usemap, width
<input>	*Defines an input field* accept, alt, autocomplete, autofocus, checked, disabled, form, formaction, formenctype, formmethod, formnovalidate, formtarget, height, list, max, maxlength, min, multiple, name, pattern, placeholder, readonly, required, size, src, step, type, value, width
<ins>	*Defines inserted text* cite, datetime
<keygen>	*Defines a generated key in a form* autofocus, challenge, disabled, form, keytype, name
<kbd>	*Defines keyboard text*
<label>	*Defines an inline sub window* for, form
<legend>	*Defines a title in a fieldset*
	Defines a list item value
<link>	*Defines a resource reference* href, hreflang, media, rel, sizes, type
<map>	*Defines an image map* name
<mark>	*Defines marked text*
<menu>	*Defines a menu list* label, type
<meta>	*Defines meta information* charset, content, http-equiv, name
<meter>	*Defines measurement within a predefined range* high, low, max, min, optimum, value
<nav>	*Defines navigation links*
<noframes>	*Used to display text for browsers that do not handle frames*
<noscript>	*Defines a noscript section*
<object>	*Defines an embedded object* data, form, height, name, type, usemap, width
	Defines an ordered list reversed, start
<optgroup>	*Defines an option group* label, disabled
<option>	*Defines an option in a drop-down list* disabled, label, selected, value
<output>	*Defines some types of output* for, form, name
<p>	*Defines a paragraph*
<param>	*Defines a parameter for an object* name, value
<pre>	*Defines preformatted text*
<progress>	*Defines progress of a task of any kind* max, value
<q>	*Defines a short quotation* cite
<rp>	*Used in ruby annotations to define what to show browsers that to not support the ruby element*
<rt>	*Defines explanation to ruby annotations*
<ruby>	*Defines ruby annotations*
<s>, <strike>	*Used to define strikethrough text.*
<samp>	*Defines sample computer code*
<script>	*Defines a definition list* async, type charset defer, src
<section>	*Defines a section* cite
<select>	*Defines a selectable list* autofocus, disabled, form, multiple, name, size
<small>	*Defines small text*
<source>	*Defines media resources* media, src, type
	Defines a section in a document
	Defines strong text
<style>	*Defines a style definition* type, media, scoped
<sub>, <sup>	*Defines sub/super-scripted text*
<table>	*Defines a table* summary
<tbody>	*Defines a table body* summary
<td>	*Defines a table cell* colspan, headers, rowspan
<textarea>	*Defines a text area* autofocus, cols, disabled, form, maxlength, name, placeholder, readonly, readonly, required, rows, wrap
<tfoot>, <thead>	*Defines a table footer / head*
<th>	*Defines a table header* colspan, headers, rowspan, scope
<time>	*Defines a date/tim* datetime
<title>	*Defines the document title*
<tr>	*Defines a table row* datetime
<tt>	*Used to define teletype text*
<u>	*Used to define underlined text*
	Defines an unordered list
<var>	*Defines a variable*
<video>	*Defines a video* autobuffer, autoplay, controls, height, loop, src, width

HTML5 TAG CHEAT SHEET

Created by WebsiteSetup.org

HTML5 CHEAT SHEET

TAGS	MEANING
<a>	hyperlink
<abbr>	abbreviation
<address>	address element
<area>	area inside an image map
<article>	article
<aside>	content aside from the page content
<audio>	sound content
<b>	bold text
<base>	base URL for all the links in a page
<bdi>	bi-directional text formatting
<bdo>	direction of text display
<blockquote>	long quotation
<body>	body element
 	single line break
<button>	push button
<canvas>	define graphics
<caption>	table caption
<cite>	citation
<code>	computer code text
<col>	table columns
<colgroup>	groups of table columns
<data>	Allows machine-readable data to be provided
<datalist>	"autocomplete" dropdown list
<dd>	definition description
<del>	deleted text
<details>	details of an element
<dfn>	definition term
<dialog>	part of an application is interactive.
	section in a document
<dl>	definition list
<dt>	definition term
	emphasized text
<embed>	external application or interactive content
<fieldset>	fieldset
<figcaption>	caption for the figure element.
<figure>	group of media content, and their caption
<footer>	footer section or page
<form>	specifies a form

TAGS	MEANING
<h1>	heading level 1
<h2>	heading level 2
<h2>	heading level 2
<h3>	heading level 3
<h4>	heading level 4
<h5>	heading level 5
<h6>	heading level 6
<head>	information about the document
<header>	group of introductory or navigational aids
<hr>	horizontal rule
<html>	html document
<i>	italic text
<iframe>	inline sub window (frame)
<img>	image
<input>	input field
<ins>	inserted text
<kbd>	keyboard text
<keygen>	generates a key pair
<label>	label for form control
<legend>	title in a fieldset
	list item
<link>	resource reference
<main>	main content area of an HTML document.
<map>	image map
<mark>	marked text
<menu>	menu list
<menuitem>	command that user can invoke from popup menu
<meta>	meta information
<meter>	measurement within a predefined range
<nav>	navigation links
<noscript>	noscript section
<object>	embedded object
	ordered list
<optgroup>	option group
<option>	option in a drop-down list
<output>	types of output
<p>	paragraph
<param>	parameter for an object
<pre>	preformatted text

TAGS	MEANING
<progress>	progress of a task
<q>	short quotation
<rb>	marks the base text component of a ruby annotation.
<rp>	used for the benefit of browsers that don't support ruby annotations
<rt>	ruby text component
<rtc>	marks a ruby text container for ruby text components in a ruby annotation.
<ruby>	ruby annotation
<s>	Indicates text that's no longer accurate or relevant.
<samp>	sample computer code
<script>	script
<section>	section
<select>	selectable list
<small>	small text
<source>	media resources
	section in a document
	strong text
<style>	style definition
<sub>	subscripted text
<summary>	summary / caption for the <details> element
<sup>	superscripted text
<table>	table
<tbody>	table body
<td>	table cell
<textarea>	text area
<tfoot>	table footer
<th>	table header
<thead>	table header
<time>	date/time
<title>	document title
<tr>	table row
<track>	text track for media such as video and audio
<u>	text with a non-textual annotation.
	unordered list
<var>	variable
<video>	video
<wbr>	line break opportunity for very long words and strings of text with no spaces.

CREATED FOR YOU BY ReviewBox HOSTINGREVIEWBOX.COM

5) What are the advantages of CSS?

- Bandwidth
- Site-wide consistency
- Page reformatting
- Accessibility
- Content separated from presentation

6) What are the limitations of CSS?

- Ascending by selectors is not possible
- Limitations of vertical control
- No expressions
- No column declaration
- Pseudo-class not controlled by dynamic behavior
- Rules, styles, targeting specific text not possible

10) What are the advantages of Embedded Style Sheets?

- You can create classes for use on multiple tag types in the document.
- You can use selector and grouping methods to apply styles in complex situations.
- No extra download is required to import the information.

11) What is a CSS selector?

It is a string that identifies the elements to which a particular declaration apply. It is also referred as a link between the HTML document and the style sheet. It is equivalent of HTML elements. There are several different types of selectors in CSS: -

- CSS Element Selector
- CSS Id Selector
- CSS Class Selector
- CSS Universal Selector

- CSS Group Selector

24) What is the difference between inline, embedded and external style sheets?

28) What is the CSS Box model and what are its elements?

The CSS box model is used to define the design and layout of elements of CSS.

The elements are:

- Margin - It removes the area around the border. It is transparent.
- Border - It represents the area around the padding
- Padding - It removes the area around the content. It is transparent.
- Content - It represents the content like text, images, etc.

31) What is the purpose of the z-index and how is it used?

The z-index helps to specify the stack order of positioned elements that may overlap one another. The z-index default value is zero and can take on either a positive or negative number.

An element with a higher z-index is always stacked above than a lower index.

Z-Index can take the following values:

- **Auto:** Sets the stack order equal to its parents.
- **Number:** Orders the stack order.
- **Initial:** Sets this property to its default value (0).
- **Inherit:** Inherits this property from its parent element.

32) Explain the difference between visibility: hidden and display: none?

visibility: hidden hides the element, but it occupies space and affects the layout of the document.

display: none also hides the element but not occupy space. It will not affect the layout of the document.

33) What do you understand by W3C?

W3C stands for World Wide Web Consortium. Its purpose is to deliver the information of the World Wide Web. It also develops rules and guidelines for the Web.

34) What is tweening?

It is the process of generating intermediate frames between two images.

It gives the impression that the first image has smoothly evolved into the second one.

It is an important method used in all types of animations.

In CSS3, Transforms (matrix, translate, rotate, scale) module can be used to achieve tweening.

7) Which HTML tag is used to display the data in the tabular form?

The **HTML table tag** is used to display data in tabular form (row * column). It also manages the layout of the page, e.g., header section, navigation bar, body content, footer section. Here is the list of tags used while displaying the data in the tabular form:

Tag	Description
<table>	It defines a table.
<tr>	It defines a row in a table.
<th>	It defines a header cell in a table.
<td>	It defines a cell in a table.
<caption>	It defines the table caption.

<colgroup>	It specifies a group of one or more columns in a table for formatting.
<col>	It is used with <colgroup> element to specify column properties for each column.
<tbody>	It is used to group the body content in a table.
<thead>	It is used to group the header content in a table.
<tfooter>	It is used to group the footer content in a table.

10) What is semantic HTML?

Semantic HTML is a coding style. It is the use of HTML markup to reinforce the semantics or meaning of the content. For example: In semantic HTML   tag is not used for bold statement as well as   tag is used for italic. Instead of these we use and tags.

11) What is an image map?

Image map facilitates you to link many different web pages using a single image. It is represented by <map> tag. You can define shapes in images that you want to make part of an image mapping.

13) How to create a nested webpage in HTML?

The HTML iframe tag is used to display a nested webpage. In other words, it represents a webpage within a webpage. The HTML <iframe> tag defines an inline frame. For example:

22) How to make a picture of a background image of a web page?

To make a picture a background image on a web page, you should put the following tag code after the </head> tag.

```
1. <body background = "image.gif">
```

Here, replace the "image.gif" with the name of your image file which you want to display on your web page.

23) What are empty elements?

HTML elements with no content are called empty elements. For example:
, <hr> etc.

```
<iframe src="demo_iframe.html" width="200px" height="200px"></iframe>
```

26) What are the entities in HTML?

The HTML character entities are used as a replacement for reserved characters in HTML. You can also replace characters that are not present on your keyboard by entities. These characters are replaced because some characters are reserved in HTML.

27) Why is a URL encoded in HTML?

An URL is encoded to convert non-ASCII characters into a format that can be used over the Internet because a URL is sent over the Internet by using the ASCII character-set only. If a URL contains characters outside the ASCII set, the URL has to be converted. The non-ASCII characters are replaced with a "%" followed by hexadecimal digits.

29) What is the canvas element in HTML5?

The <canvas> element is a container that is used to draw graphics on the web page using scripting language like JavaScript. It allows for dynamic and scriptable rendering of 2D shapes and bitmap images. There are several methods in canvas to draw paths, boxes, circles, text and add images. For Example:

```
<canvas id="myCanvas1" width="300" height="100" style="border:2px solid;">
Your browser does not support the HTML5 canvas tag.
</canvas>
```

30) What is SVG?

HTML SVG is used to describe the two-dimensional vector and vector/raster graphics. SVG images and their behaviors are defined in XML text files. So as XML files, you can create and edit an SVG image with the text editor. It is mostly used for vector type diagrams like pie charts, 2-Dimensional graphs in an X, Y coordinate system.

```
<svg width="100" height="100">
 <circle cx="50" cy="50" r="40" stroke="yellow" stroke-width="4" fill="red" />
</svg>
```

36) What is the use of figure tag in HTML 5?

The figure tag is used to add a photo in the document on the web page. It is used to handle the group of diagrams, photos, code listing with some embedded content.

```
<p>The Taj Mahal is widely recognized as "the jewel of Muslim art in India and one of t
he universally admired masterpieces of the world's heritage."</p>
<figure>
  <img src="htmlpages/images/tajmahal.jpg" alt="Taj Mahal"/>
</figure>
```

37) What is the use of figcaption tag in HTML 5?

The <figcaption> element is used to provide a caption to an image. It is an optional tag and can appear before or after the content within the <figure> tag. The <figcaption> element is used with <figure> element and it can be placed as the first or last child of the <figure> element.

```
<figure>
  <img src="htmlpages/images/tajmahal.jpg" alt="Taj Mahal"/>
<figcaption>Fig.1.1 - A front view of the great Taj Mahal in Agra.</figcaption>
</figure>
```

42) If I do not put <!DOCTYPE html> will HTML 5 work?

No, the browser will not be able to identify that it is an HTML document and HTML 5 tags do not function properly..