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.

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	

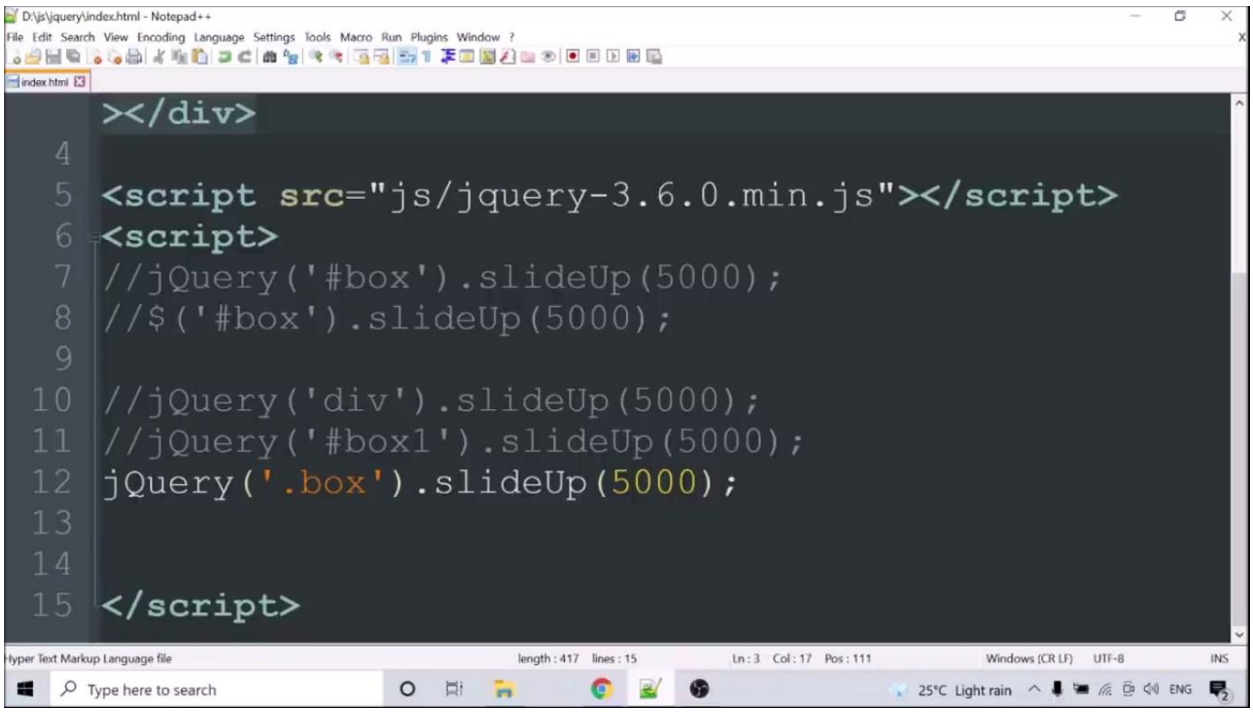
JavaScript Booleans	
toSource()	valueOf()
toString()	

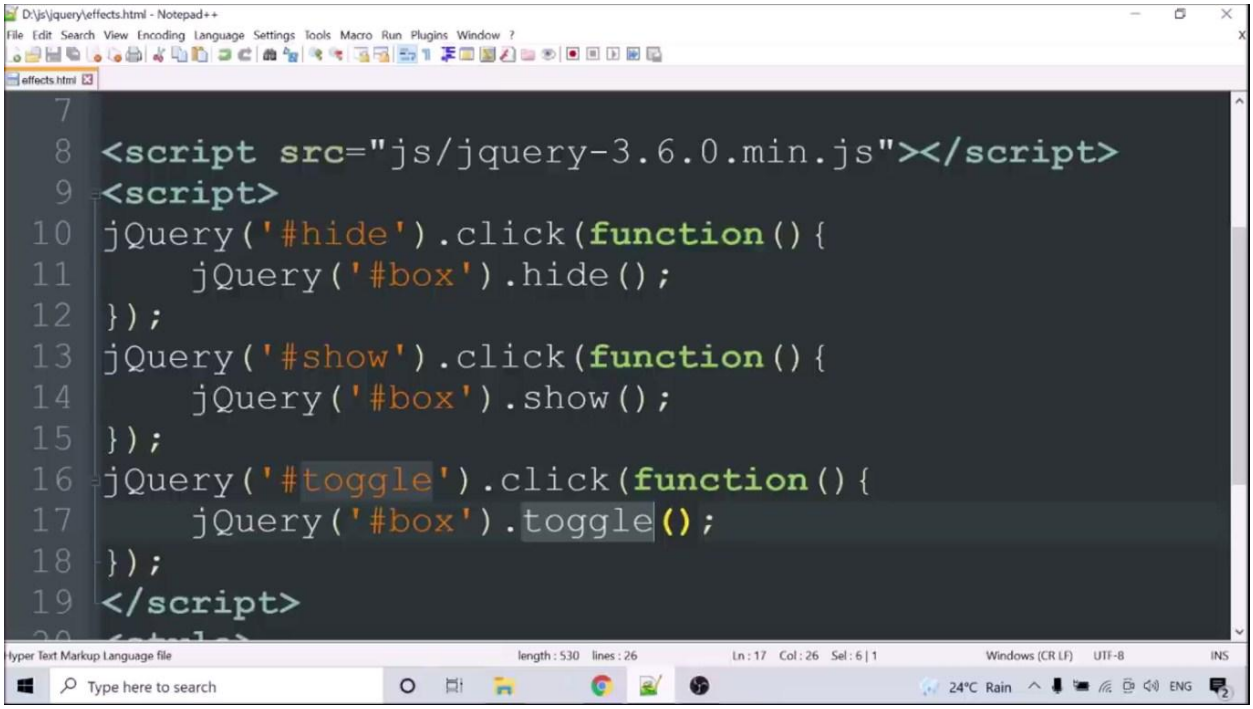
JavaScript Dates	
Date()	setMonth()
getDate()	setFullYear()
getDay()	setHours()
getMonth	setMinutes()
getFullYear	setSeconds()
getYear	setMilliseconds()
getHours	setTime()
getMinutes	setUTCDate()
getSeconds	setUTCDay()
getMilliseconds	setUTCMonth()
getTime	setUTCFullYear()
getTimezoneO-ffset()	setUTCHours()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

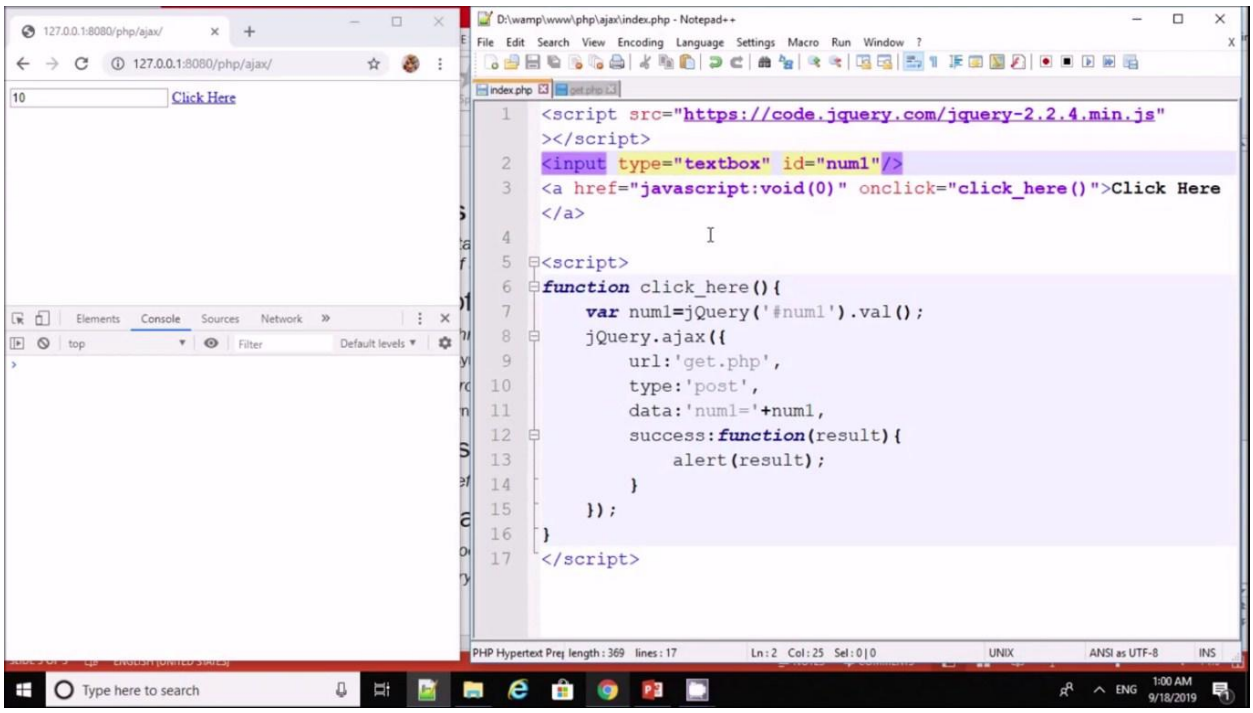
JavaScript Strings	
charAt()	slice()
charCodeAt()	split() x
concat()	substr()
fromCharCode()	substring()
indexOf()	toLowerCase()
lastIndexOf()	toUpperCase()
length	toLocaleLowerCase()
localeCompare()	toLocaleUpperCase()
match() x	toSource()
replace() x	valueOf()
search() x	

String object methods with an x support regular expressions.

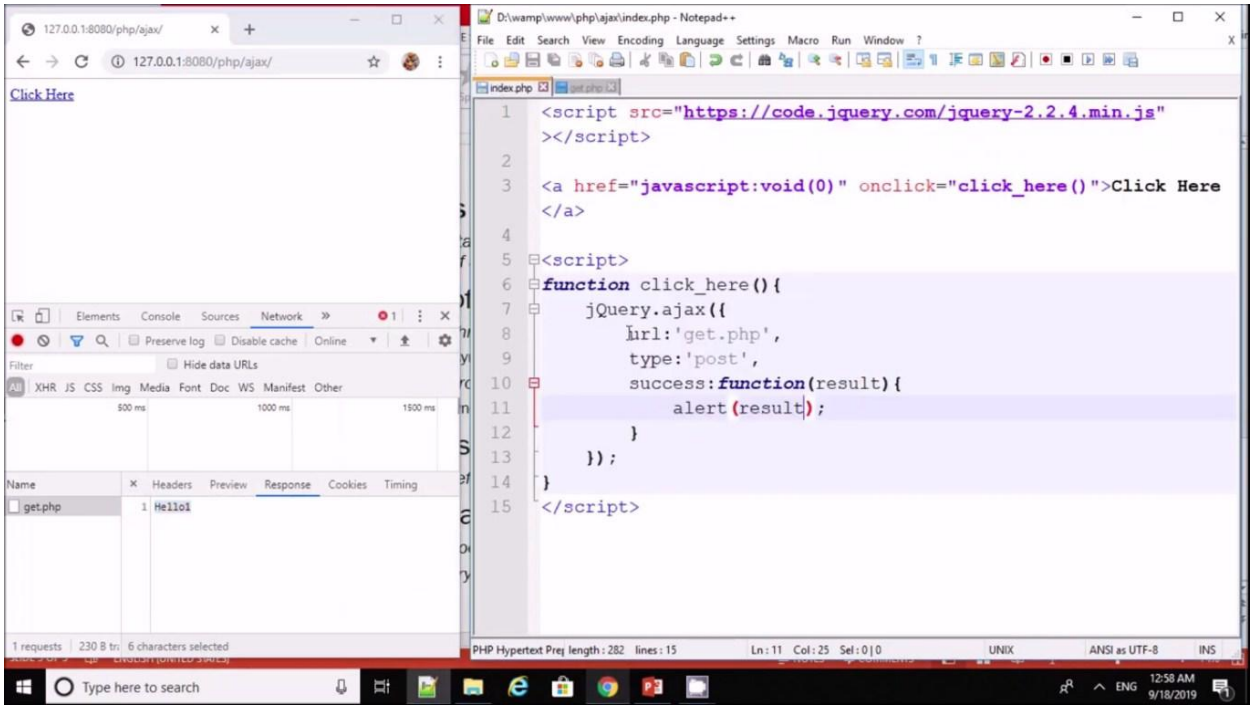
JavaScript Functions	
decodeURI()	isNaN()
decodeURIComponent()	Number()
encodeURI()	parseFloat()
encodeURIComponent()	parseInt()
escape()	String()
eval()	unescape()
isFinite()	


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

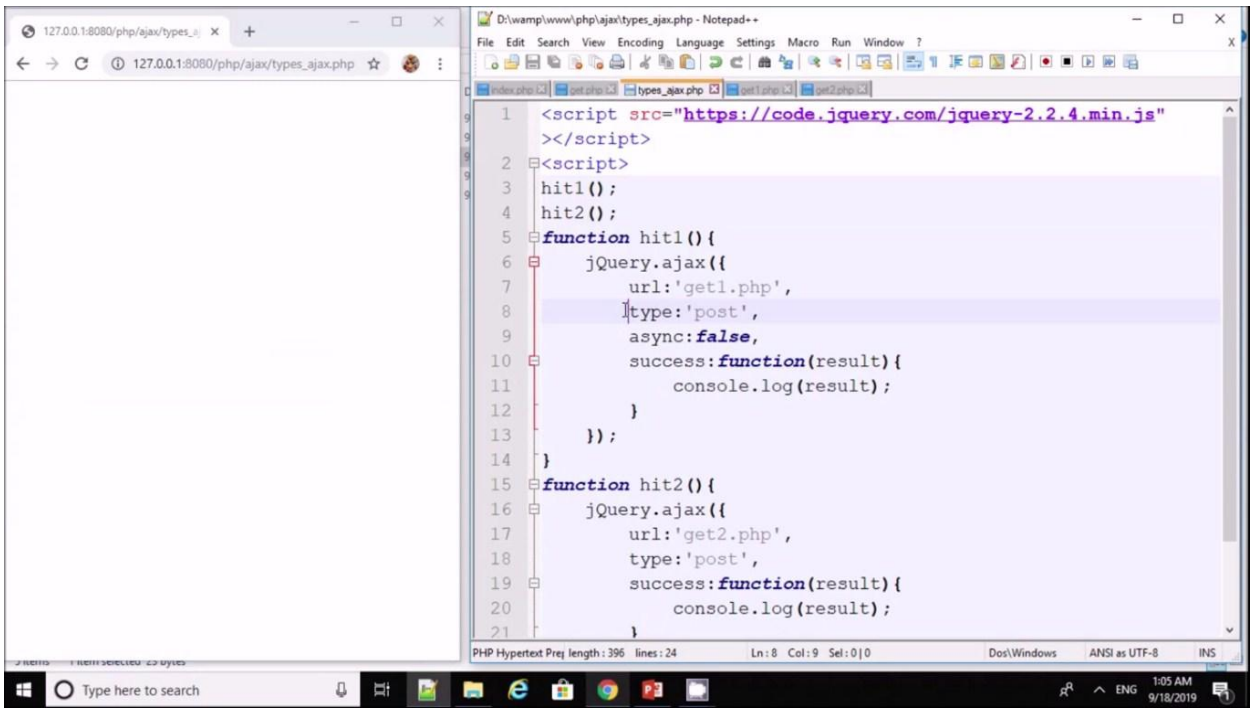

D:\js\jquery\effects.html - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
effects.html
<script src="js/jquery-3.6.0.min.js"></script>
<script>
jQuery('#hide').click(function(){
jQuery('#box').hide();
});
jQuery('#show').click(function(){
jQuery('#box').show();
});
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

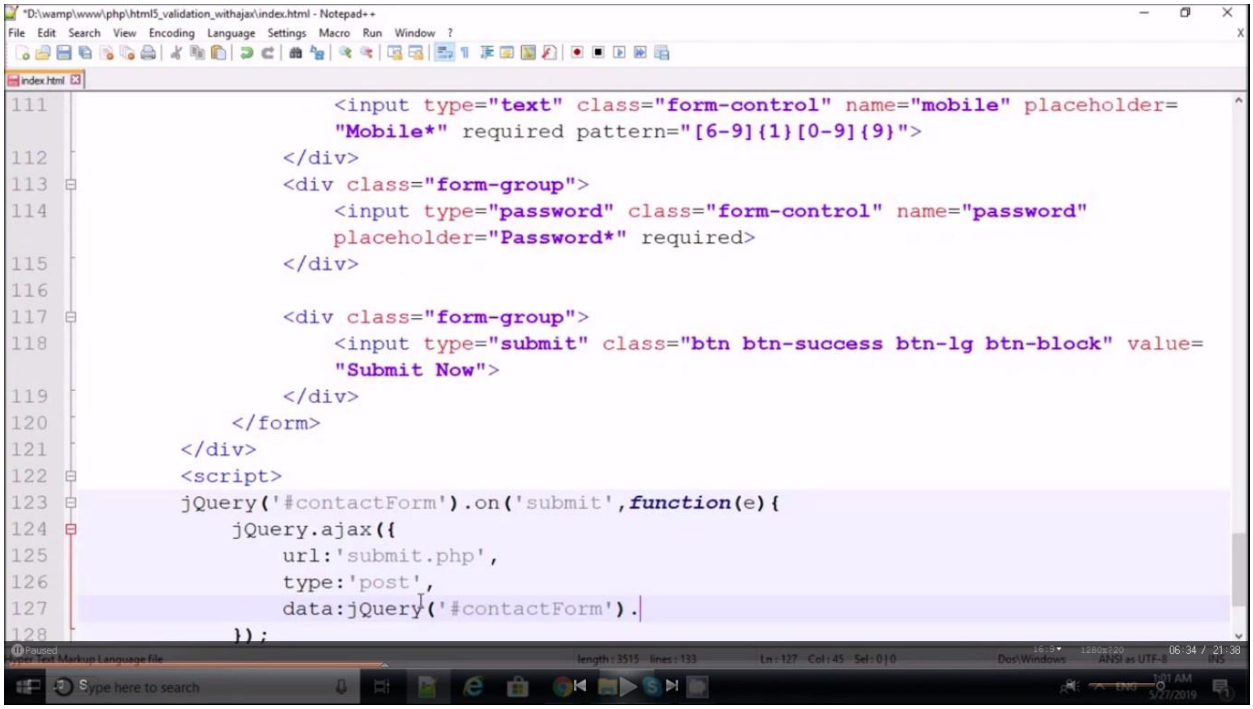

127.0.0.1:8080/php/ajax/
10
Click Here
Elements
Console
Sources
Network
top
Filter
Default levels
D:\wamp\www\php\ajax\index.php - Notepad++
File Edit Search View Encoding Language Settings Macro Run Window ?
index.php
get.php
1 <script src="https://code.jquery.com/jquery-2.2.4.min.js"
></script>
2 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

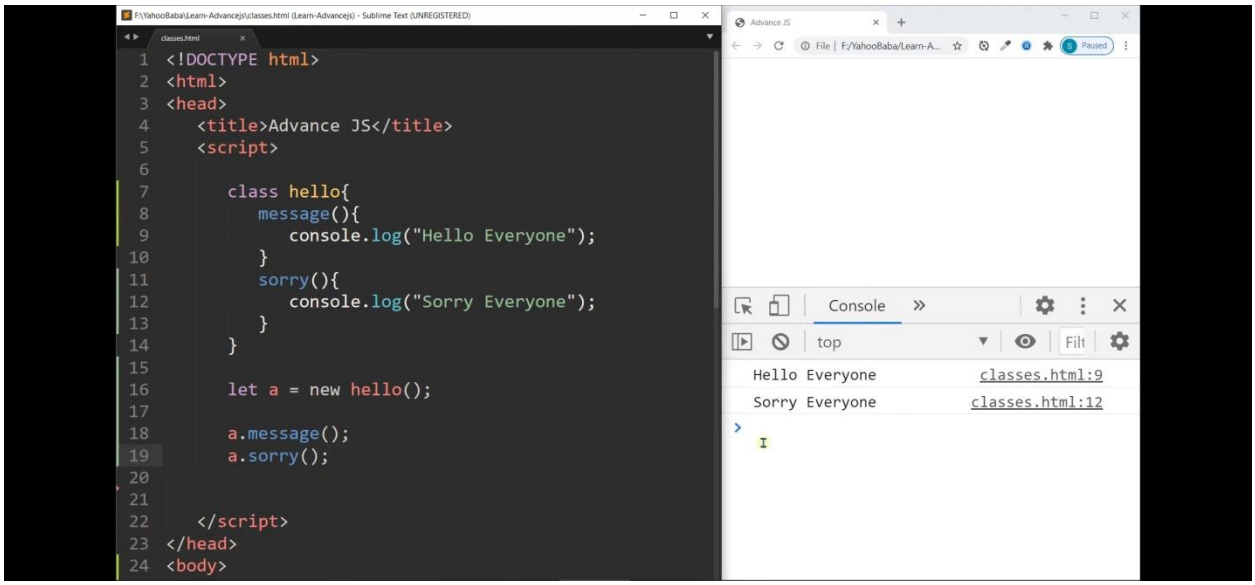

*D:\wamp\www\php\html5_validation_withajax\index.html - Notepad++
File Edit Search View Encoding Language Settings Macro Run Window ?
index.html
<input type="text" class="form-control" name="mobile" placeholder=
"Mobile*" required pattern="[6-9]{1}[0-9]{9}">

<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

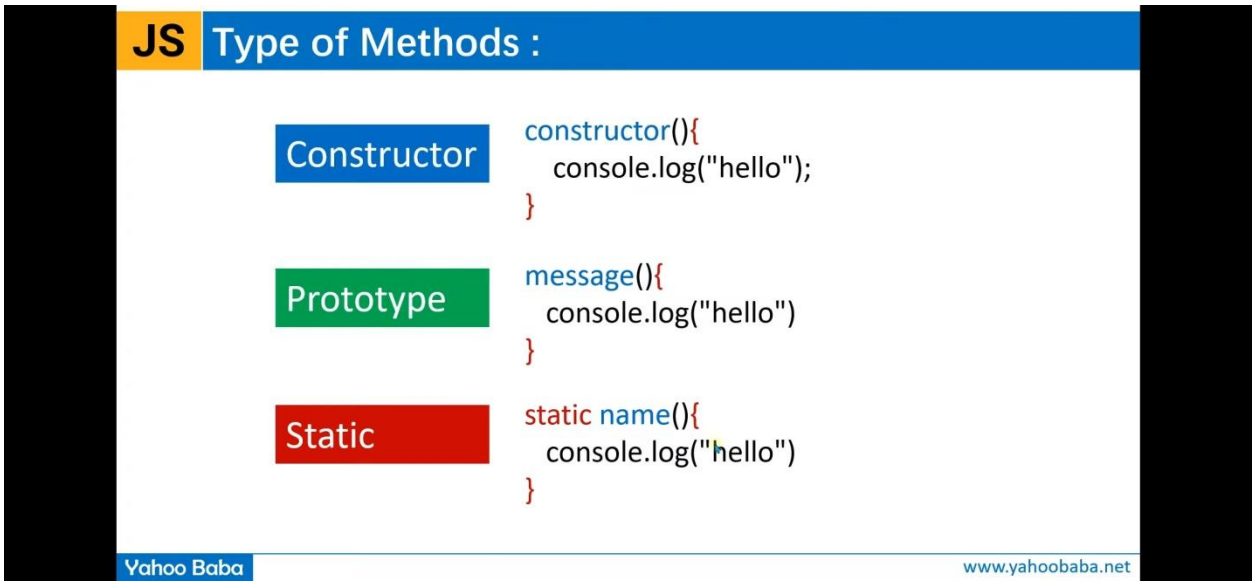
```
constructor(){
   console.log("hello");
}
```

Prototype

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

Static

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



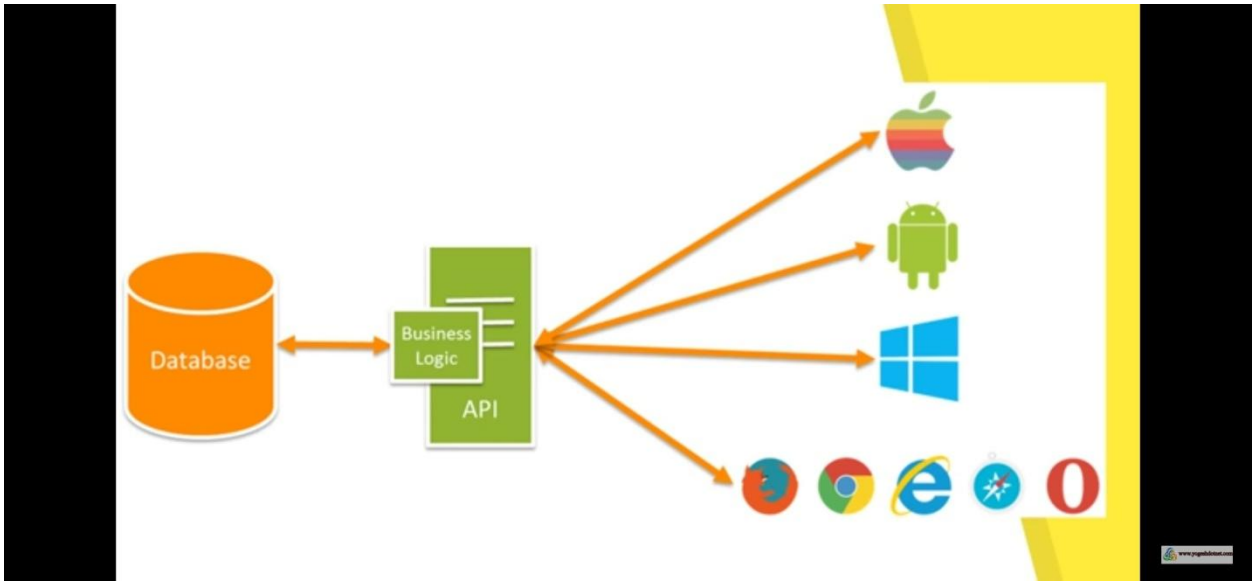
Pillar of OOPS

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

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

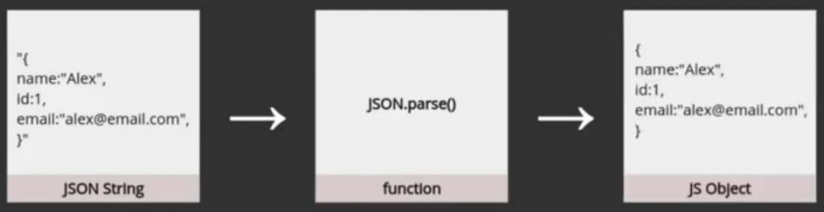


JSON.stringify()

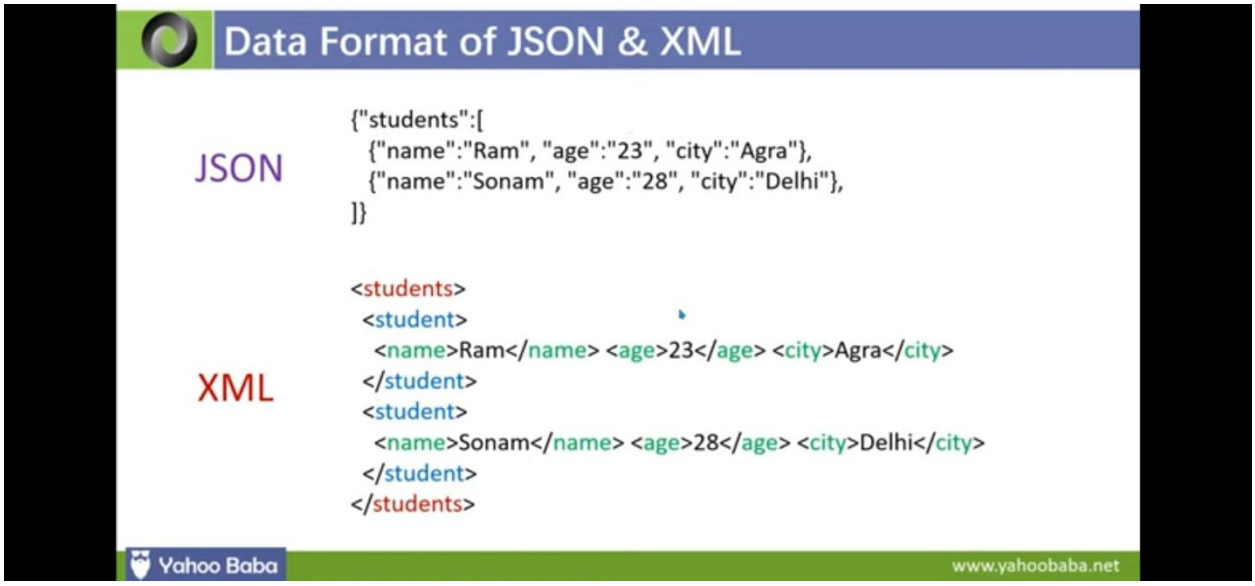
Transforms a JavaScript object into a JSON string.

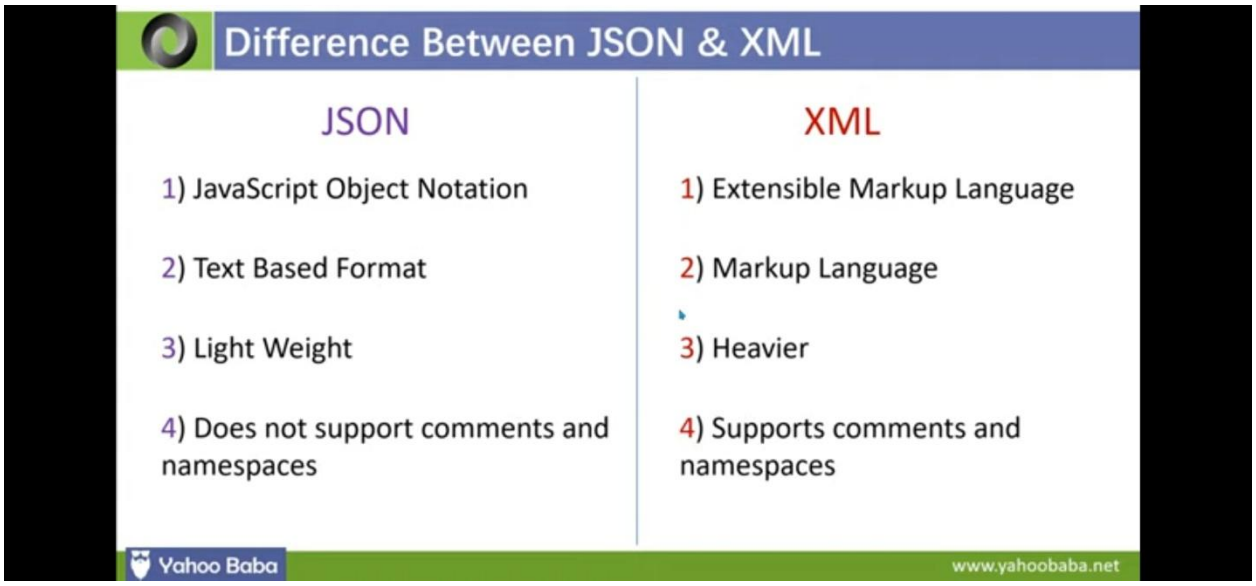
JSON.parse()

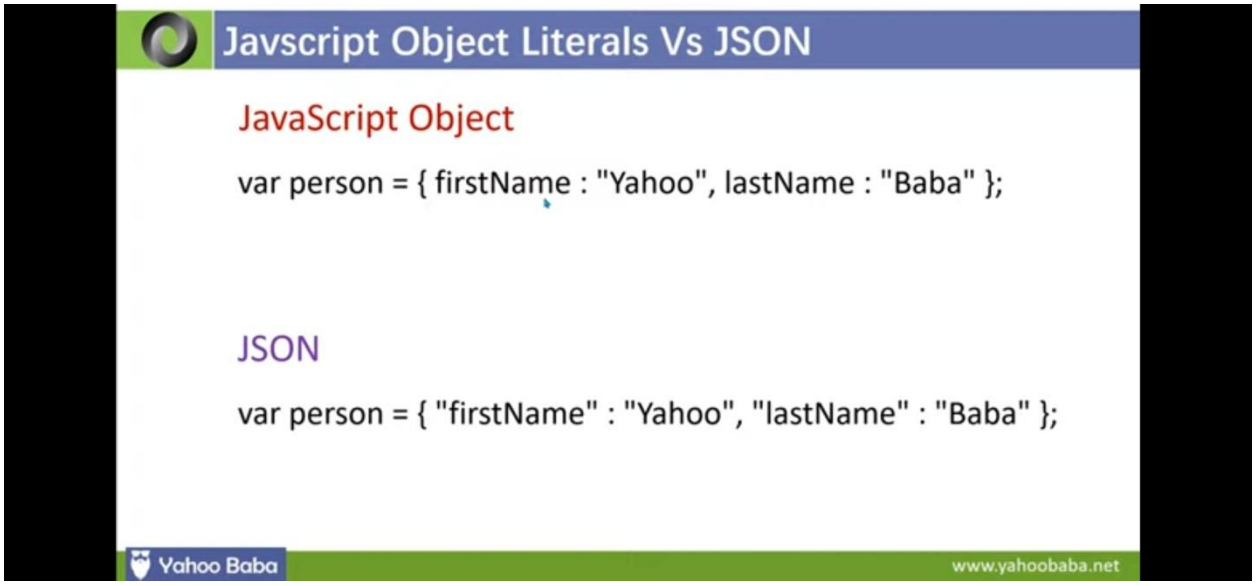
Takes a JSON string and transforms it into a JavaScript object.



```
> 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

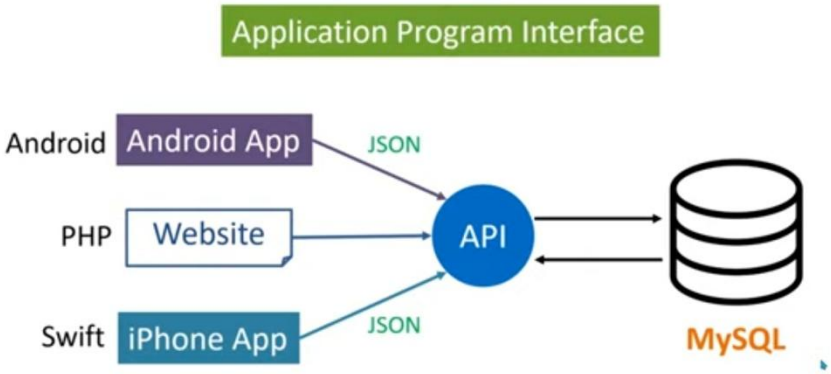
JavaScript Object

```
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

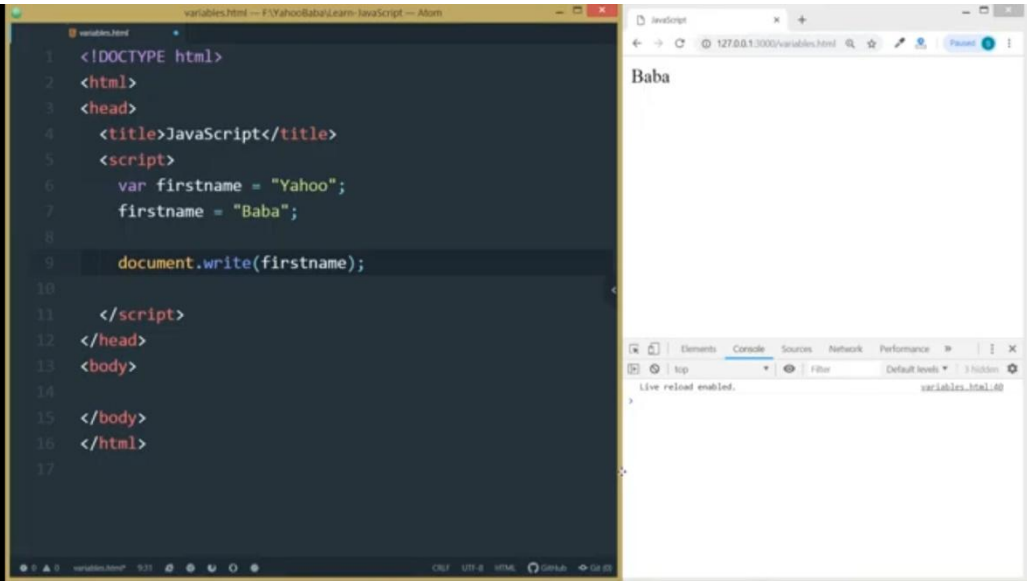
DOM Get Methods :

- innerText
- innerHTML
- getAttribute
- getAttributeNode
- Attributes

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

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 :

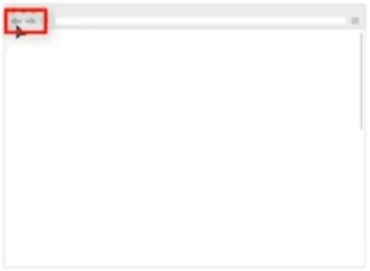
Initialization Condition Increment / Decrement

```
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()

JavaScript Number Methods :

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

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

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)

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

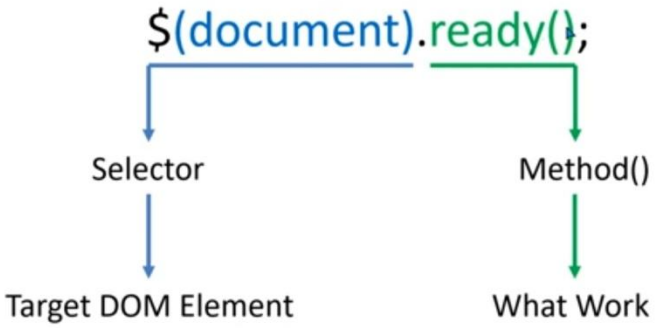
```
<script>
    var a = document.getElementById("idName").innerHTML();
    console.log(a);
</script>
```

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")

Advance Selectors

$("*")

$("ul li")

$(".abc , .xyz")

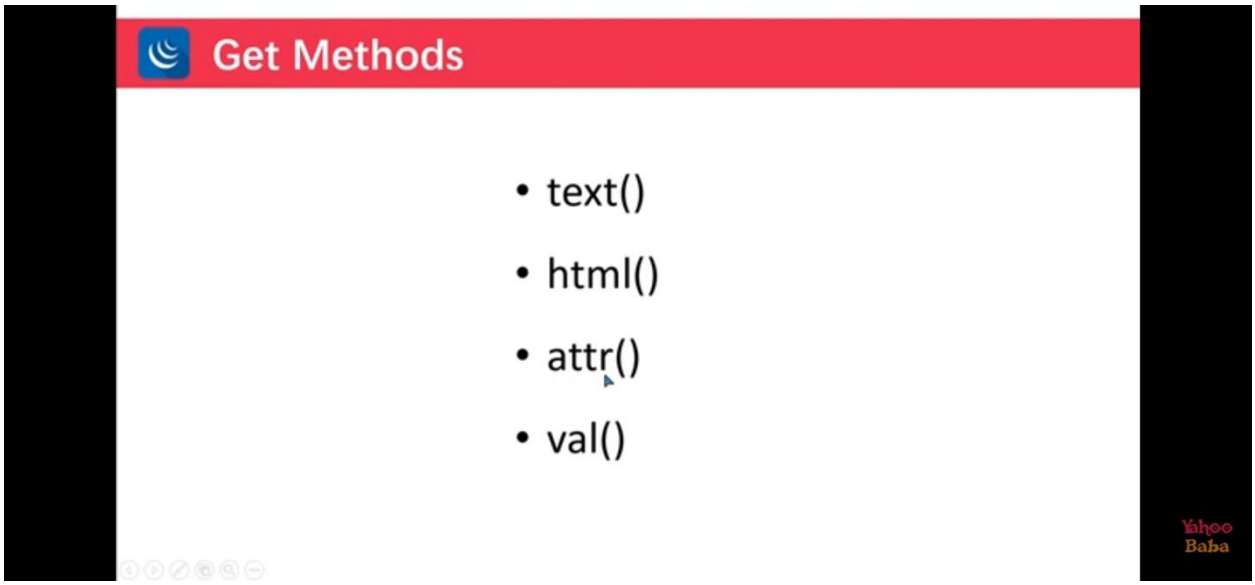
$("h1, div, p")

$("p:first")

$("p:last")

$("li:even")

$("li:odd")


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

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

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!");
});
```

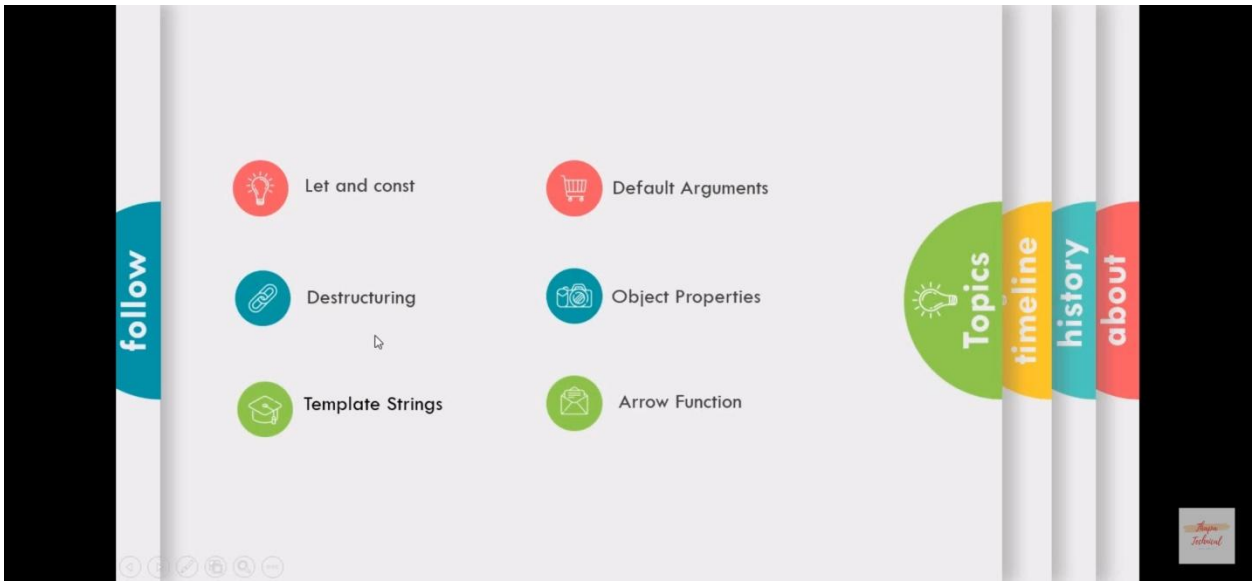
Selectors

```
$("*")                      // 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()

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"; ⊗

JS What is Fetch() Method ?

AJAX

jQuery

$.ajax();

$.get();

$.post();

JavaScript

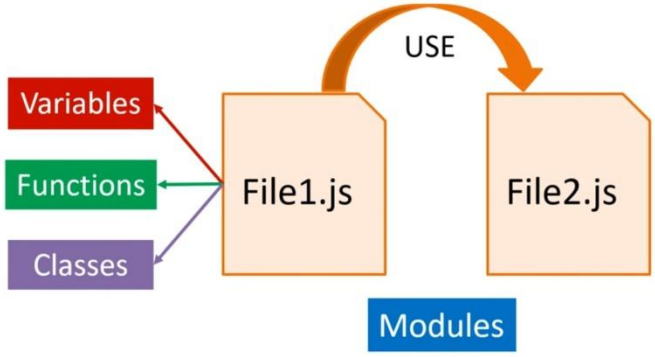
XMLHttpRequest

JavaScript ES6

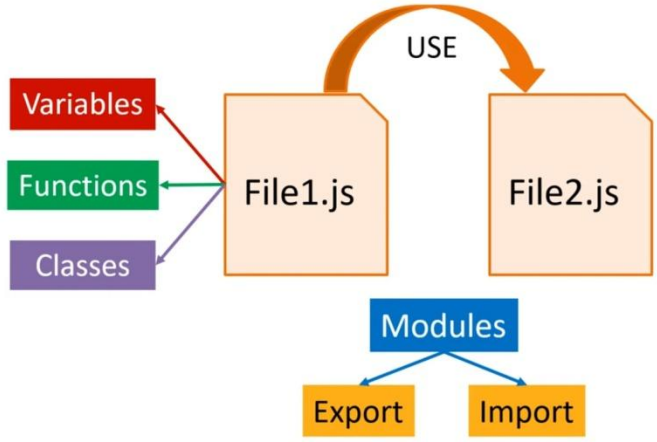
Fetch()

- 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();
```

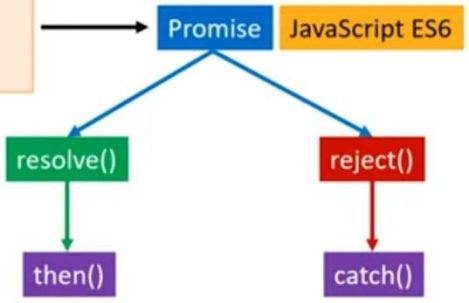
Pause ← yield "First";

g.next(); → First

JS Async Function

```
async function test(){

}
```

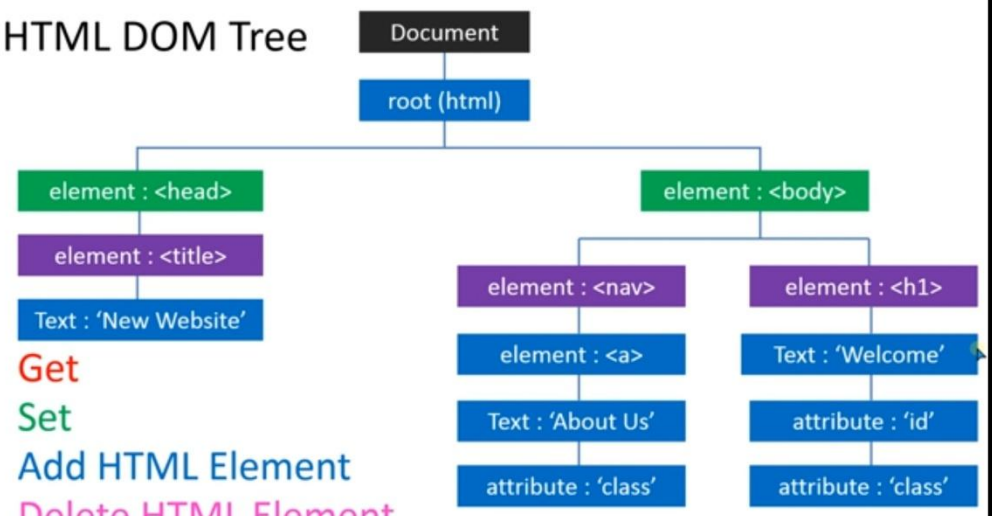
JavaScript ES 2017



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

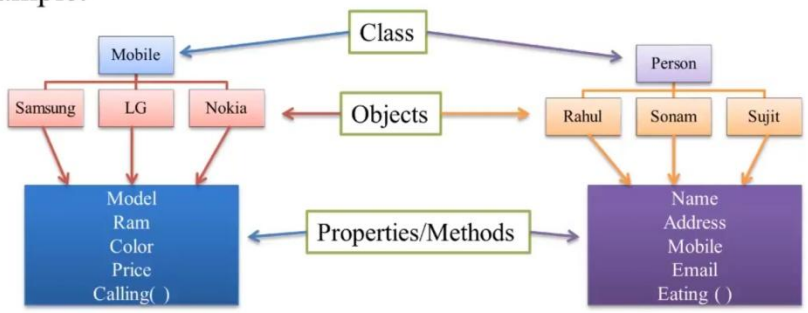
DOM Set Methods :

- innerText
- innerHTML
- setAttribute
- Attribute
- removeAttribute

Class

A specific category can be defined as class.

Example:-



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);
```

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);
```

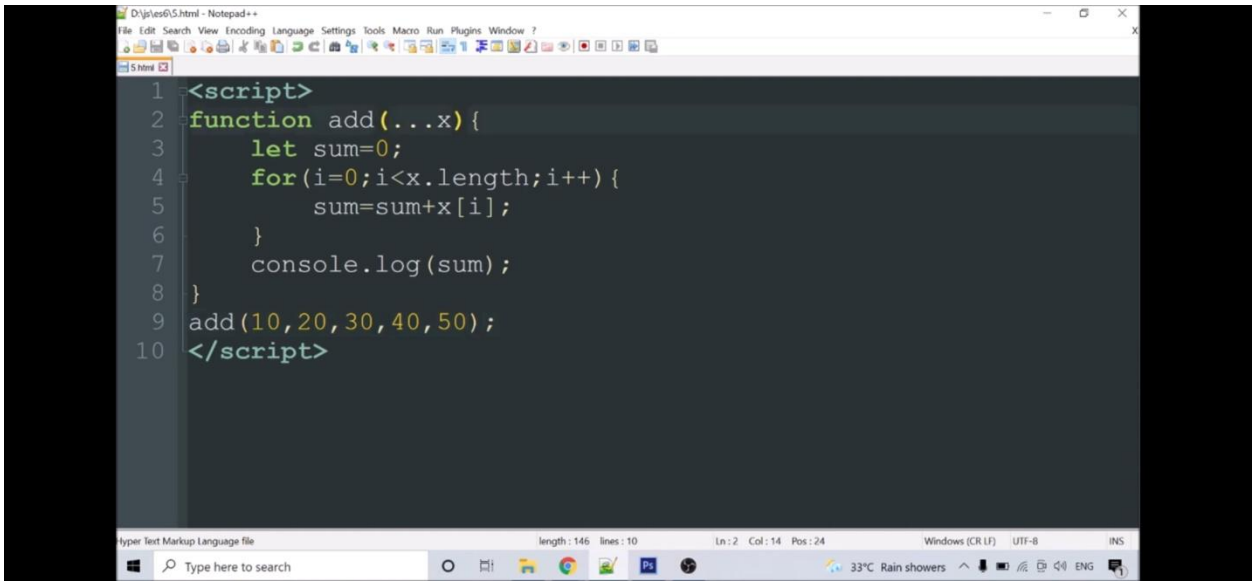

Defining a Class

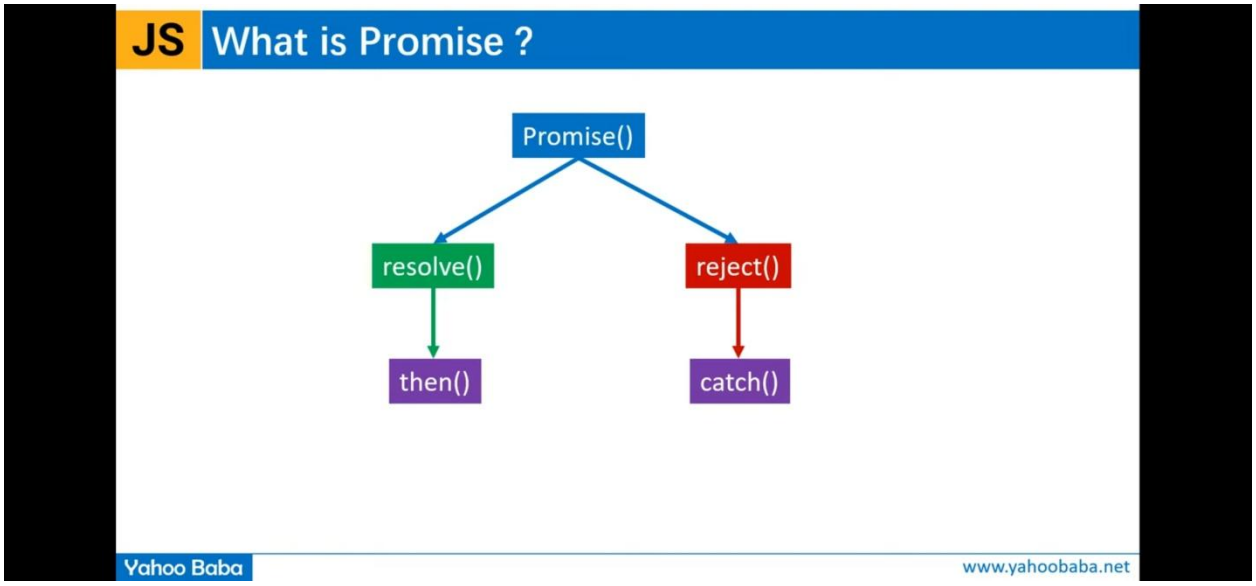
We define class in JavaScripts using custom constructor.

```
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);
```

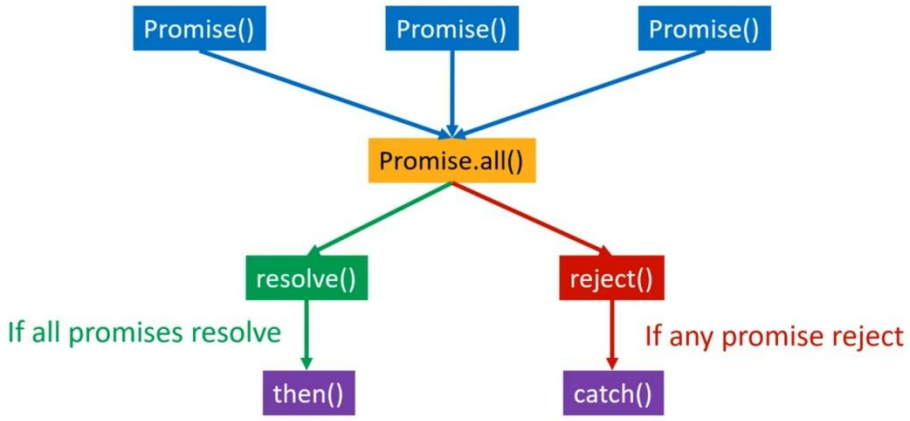
ES6 Class

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


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

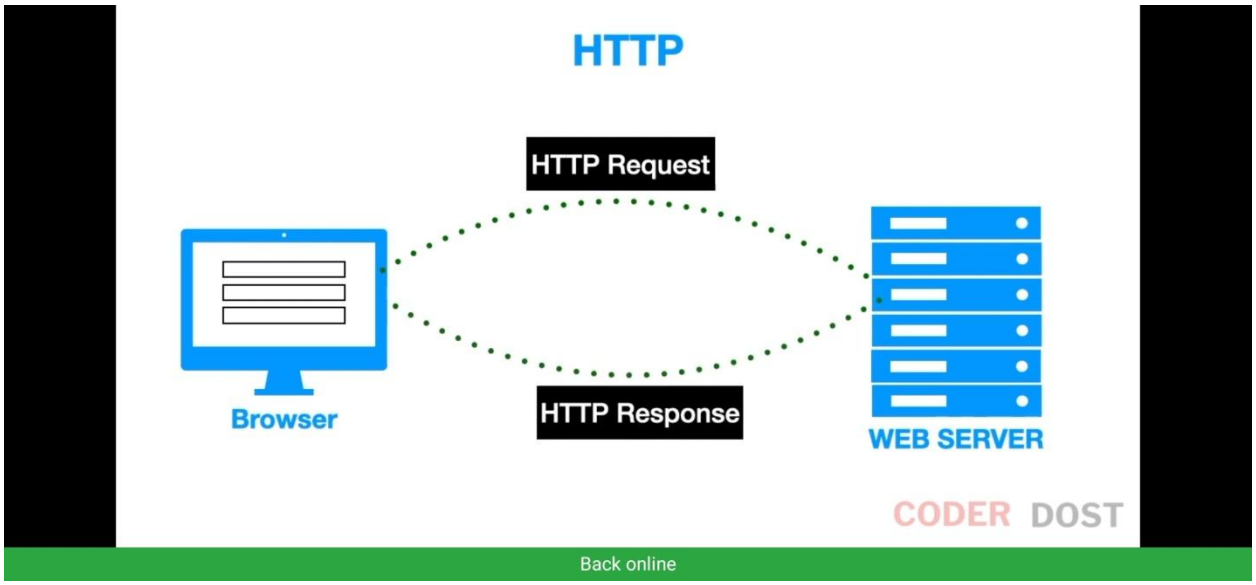

JS
What is Promise ?
Promise()
resolve()
reject()
then()
catch()
Yahoo Baba
www.yahoobaba.net

JS What is Promise.all() ?



Promise : .then() & .catch()

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

})
p.then(function(data){

}).catch(function(data){

})
```



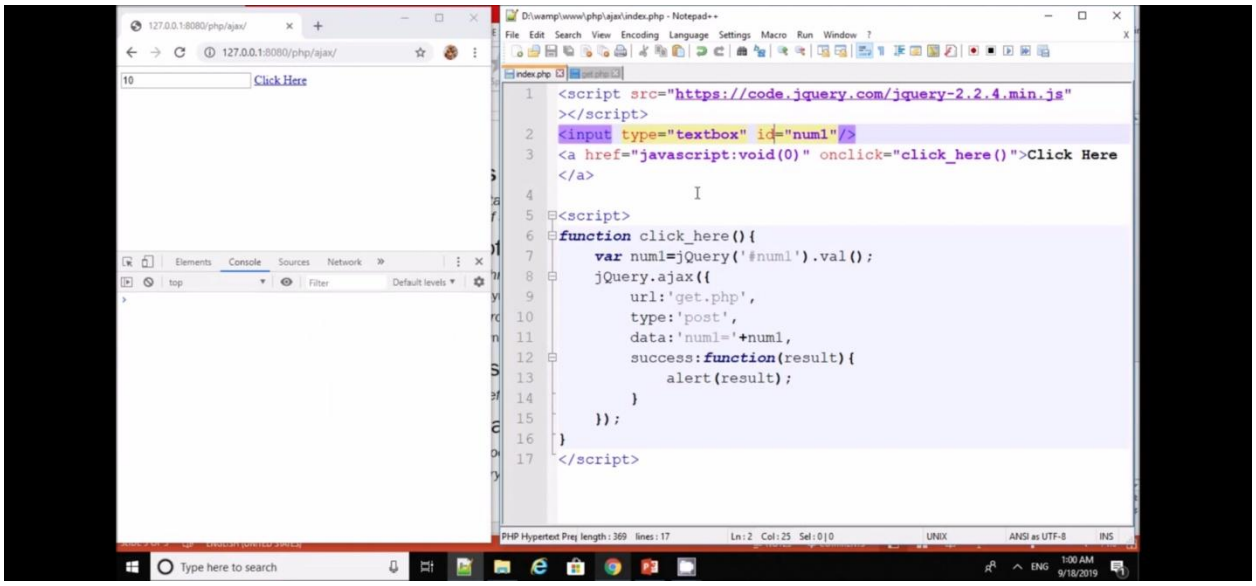
Back online

Recursion in Function

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

For Example

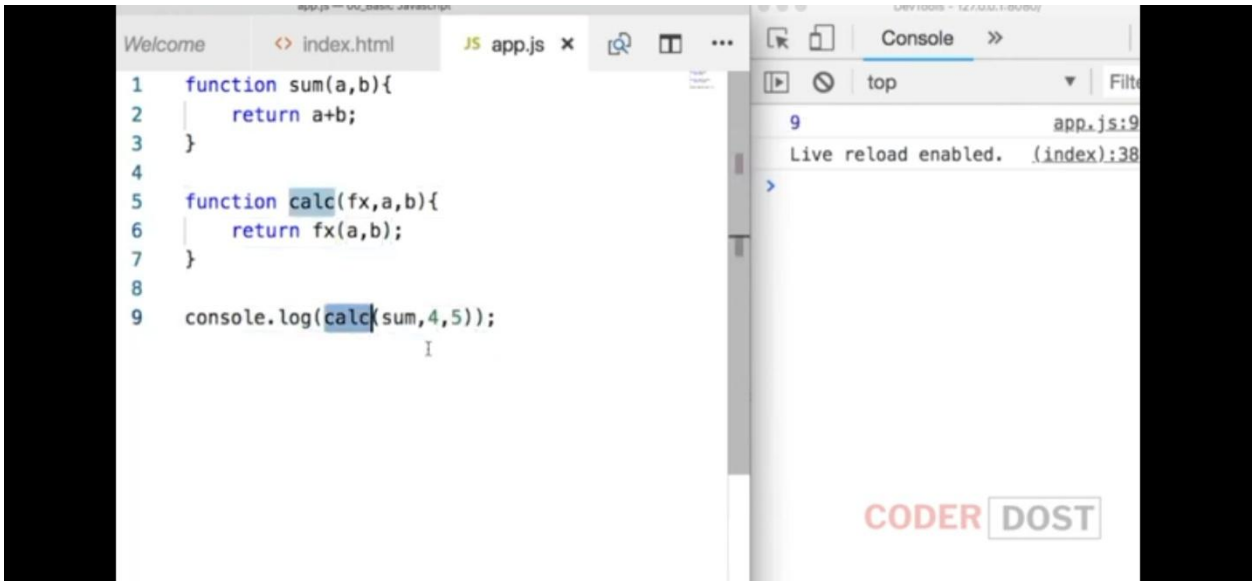
```
function fact(a)
{
If(a==1)
return 1;
else
return a*fact(a-1);
}
```

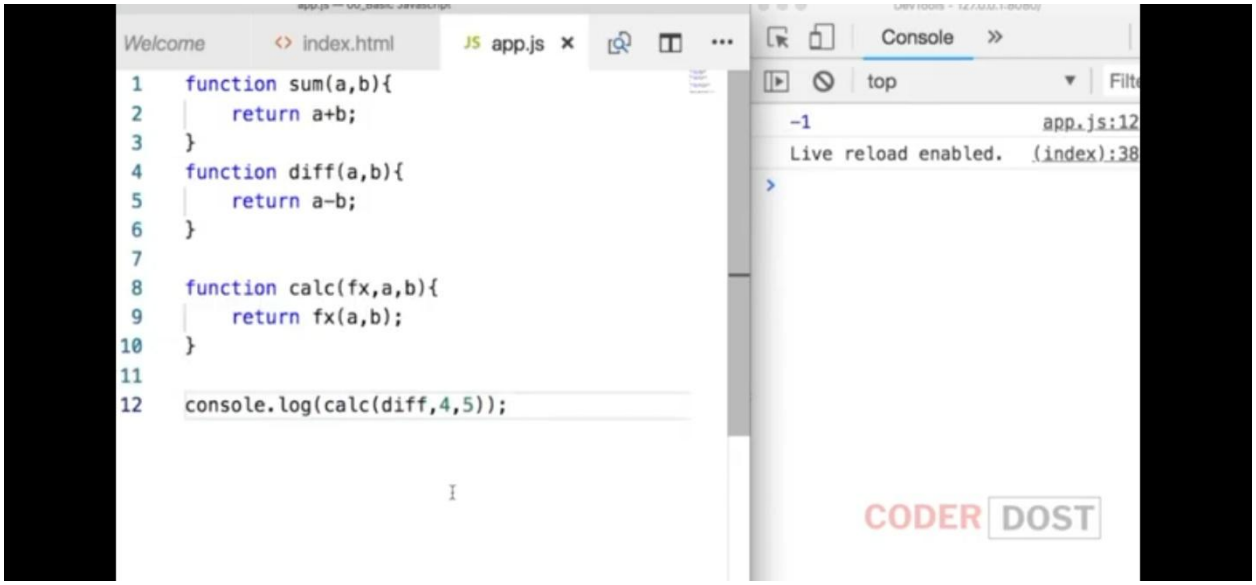


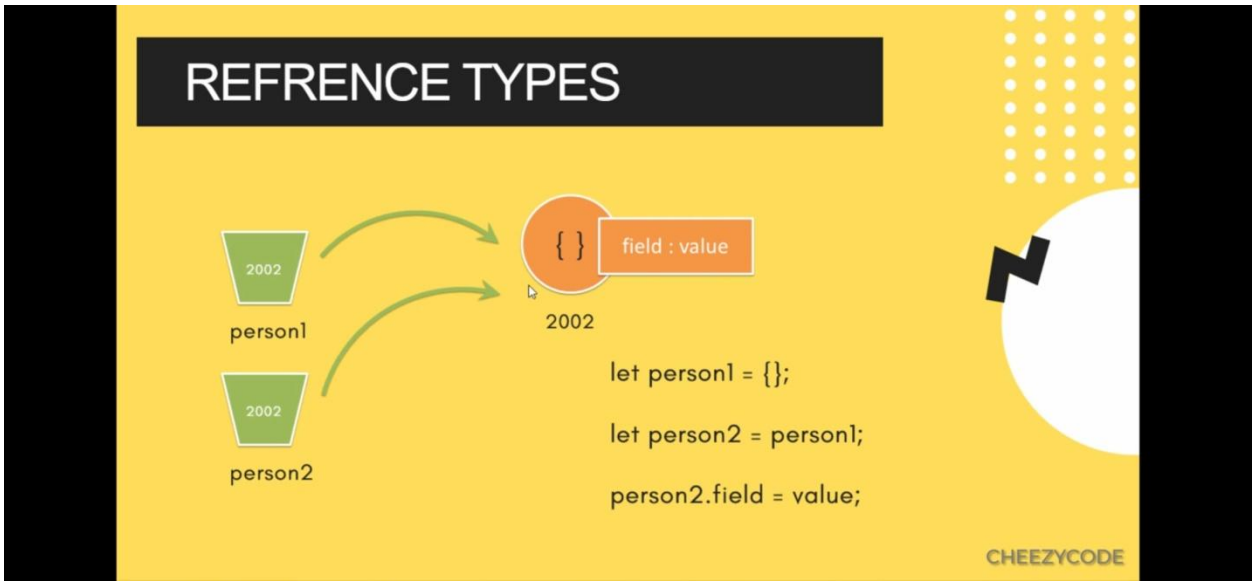
Callback Functions

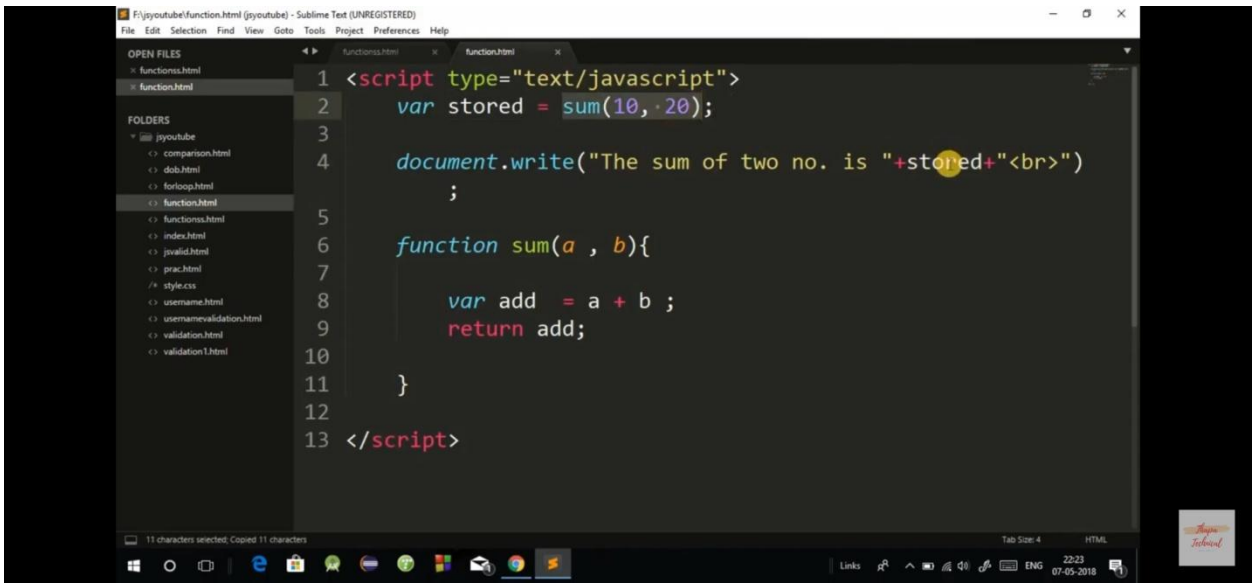
```
var calc = function(fx,a,b){
      return fx(a,b);
}

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

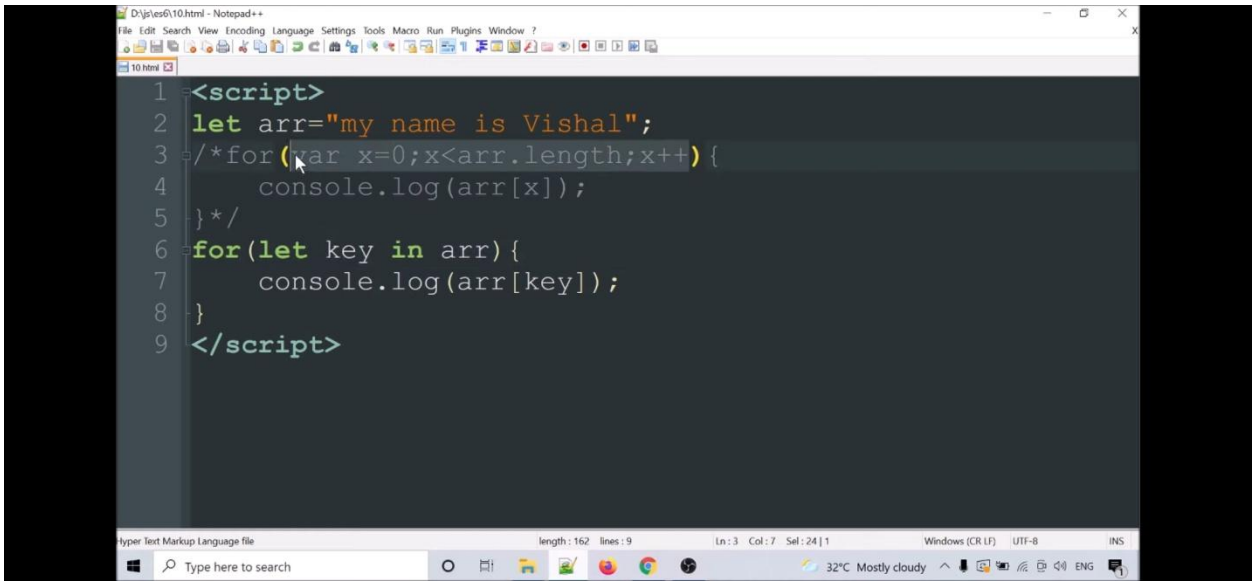
calc(sum,4,5)

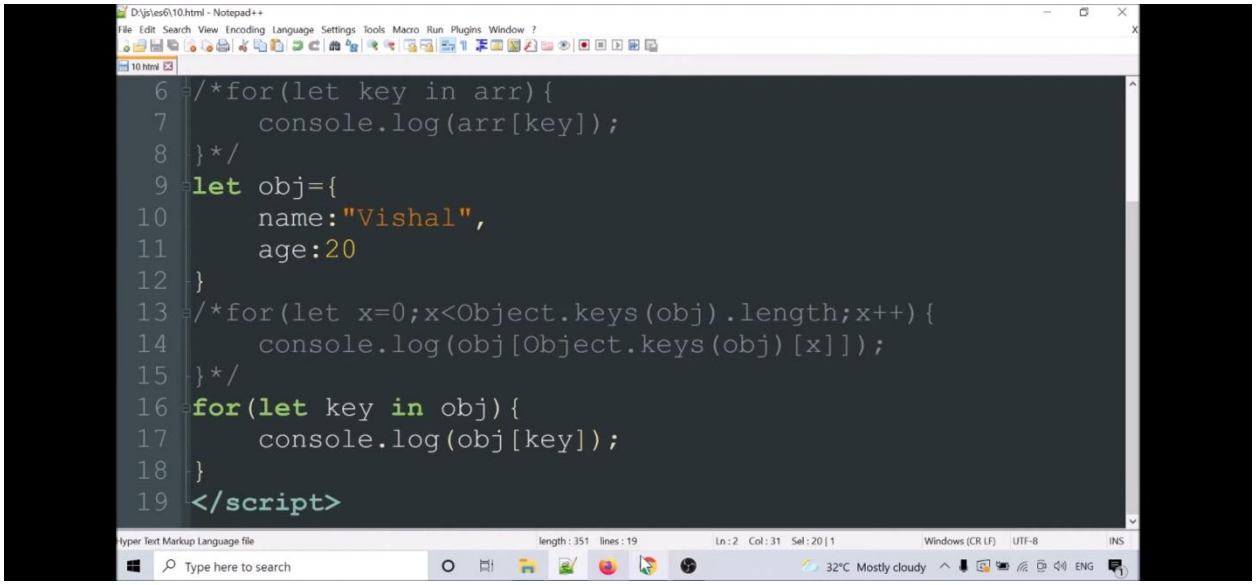

Welcome
index.html
app.js
function sum(a,b){
return a+b;
}
function calc(fx,a,b){
return fx(a,b);
}
console.log(calc(sum,4,5));
Console
top
9
app.js:9
Live reload enabled.
(index):38
CODER DOST


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: -

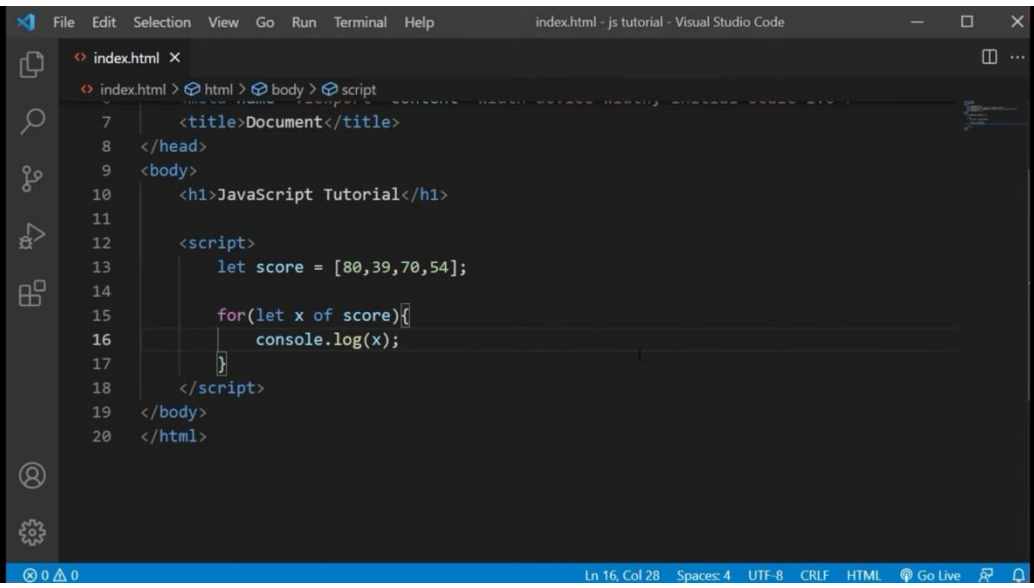
Inheritance allows a class (subclass) to acquire the properties and behavior of another class (super-class). It helps to reuse, customize and enhance the existing code. So it helps to write a code accurately and reduce the development time.

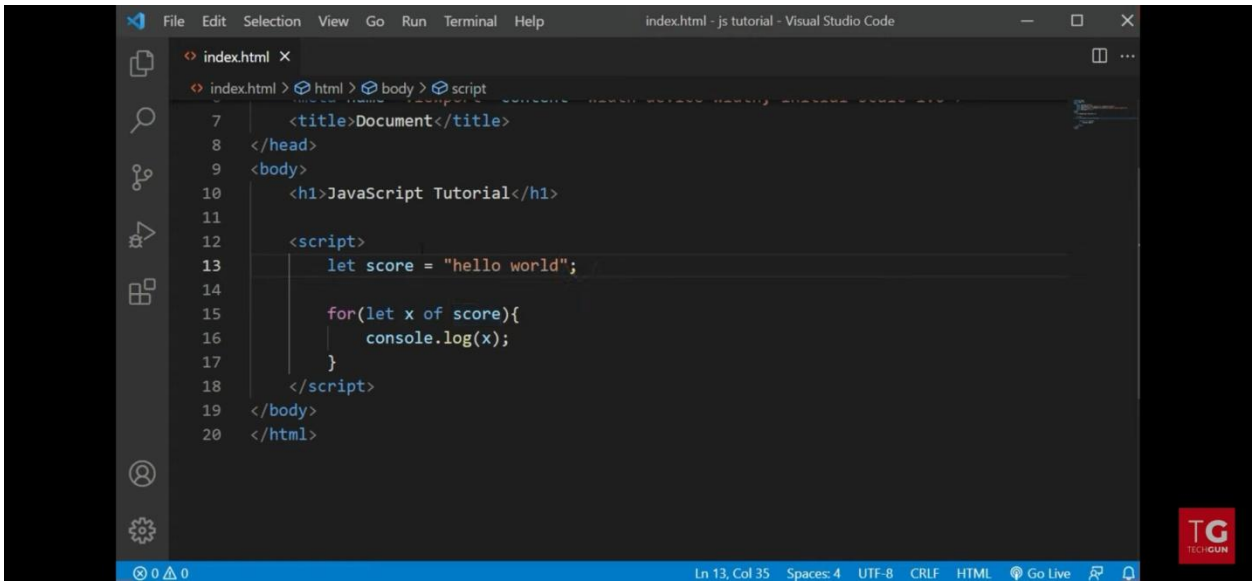
4. Polymorphism: -

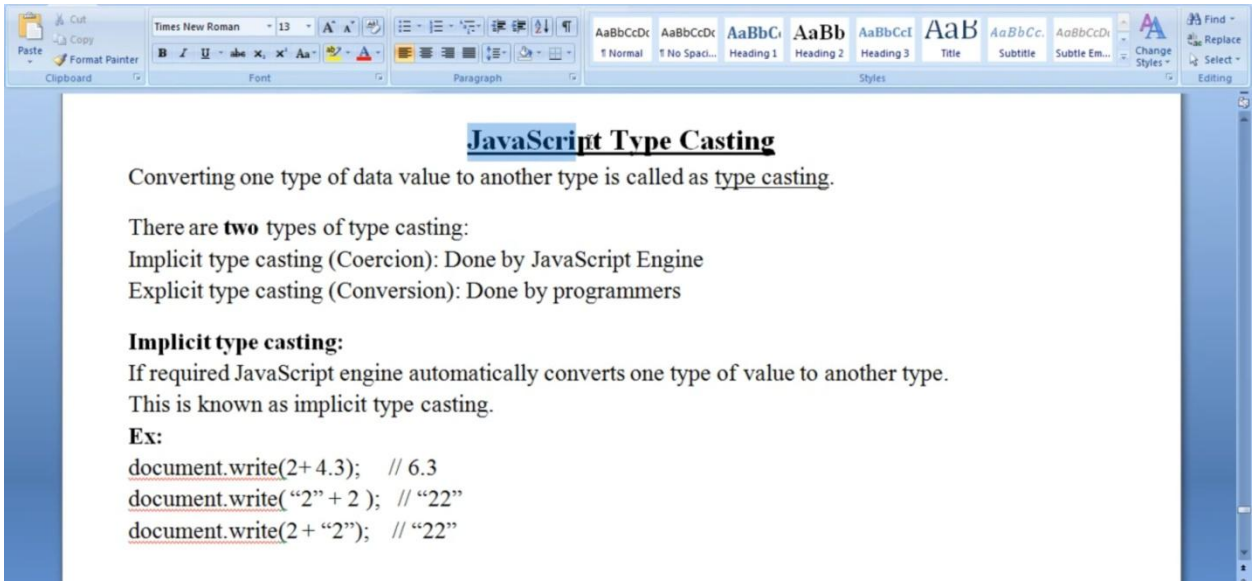
So polymorphism means "many forms". A subclass can define its own unique behavior and still share the same functionalities or behavior of its parent/base class.

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

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

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

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







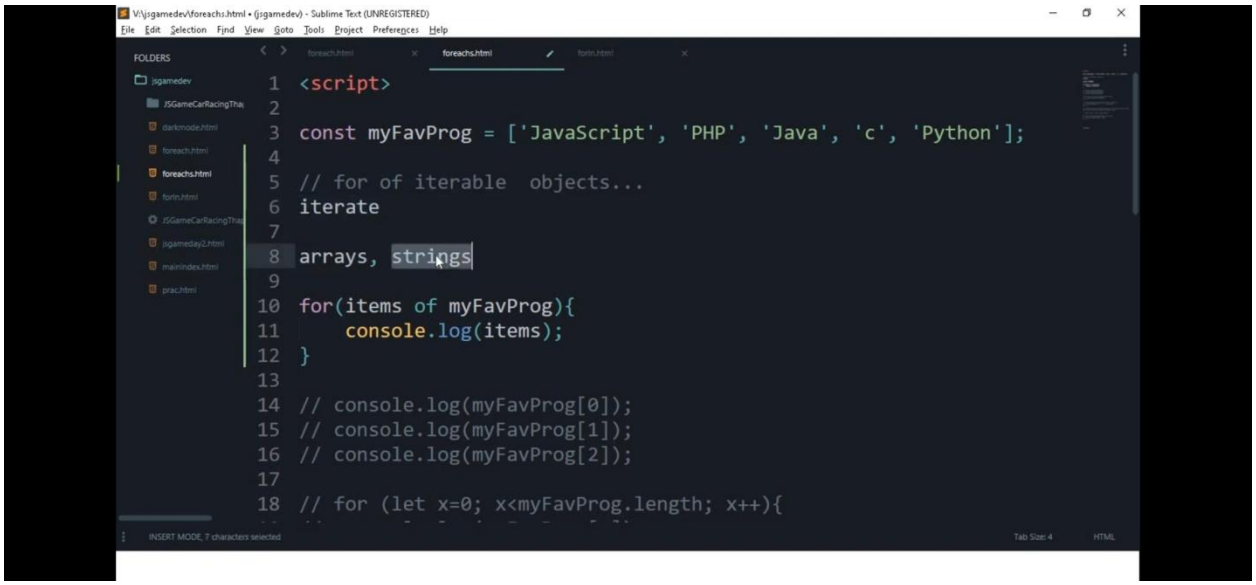
JavaScript Type Casting

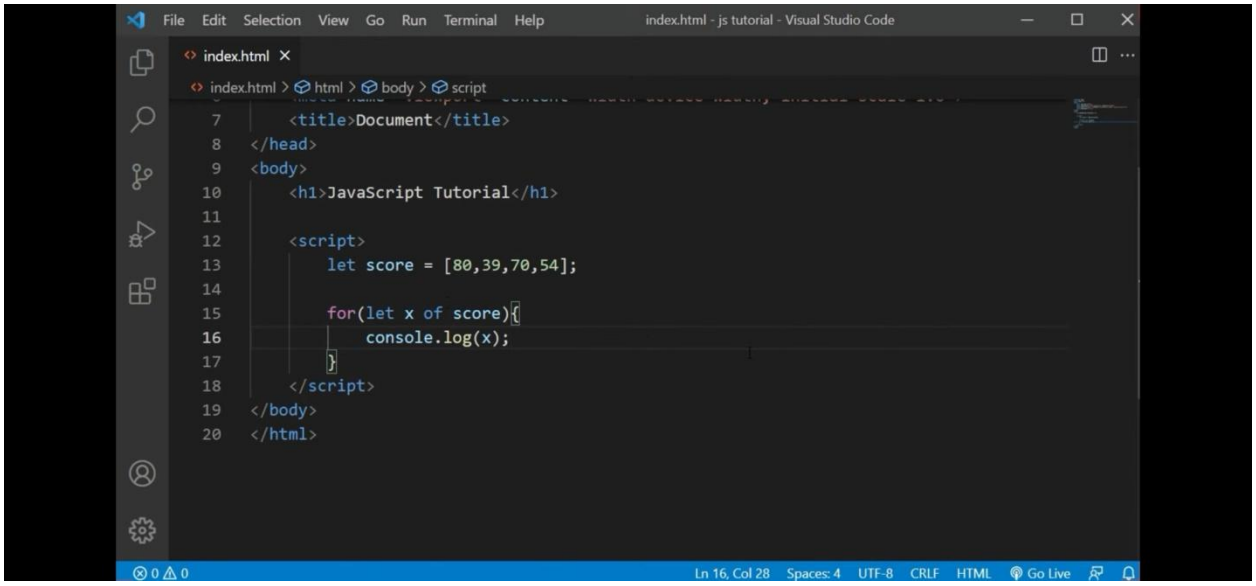
Converting one type of data value to another type is called as type casting.

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

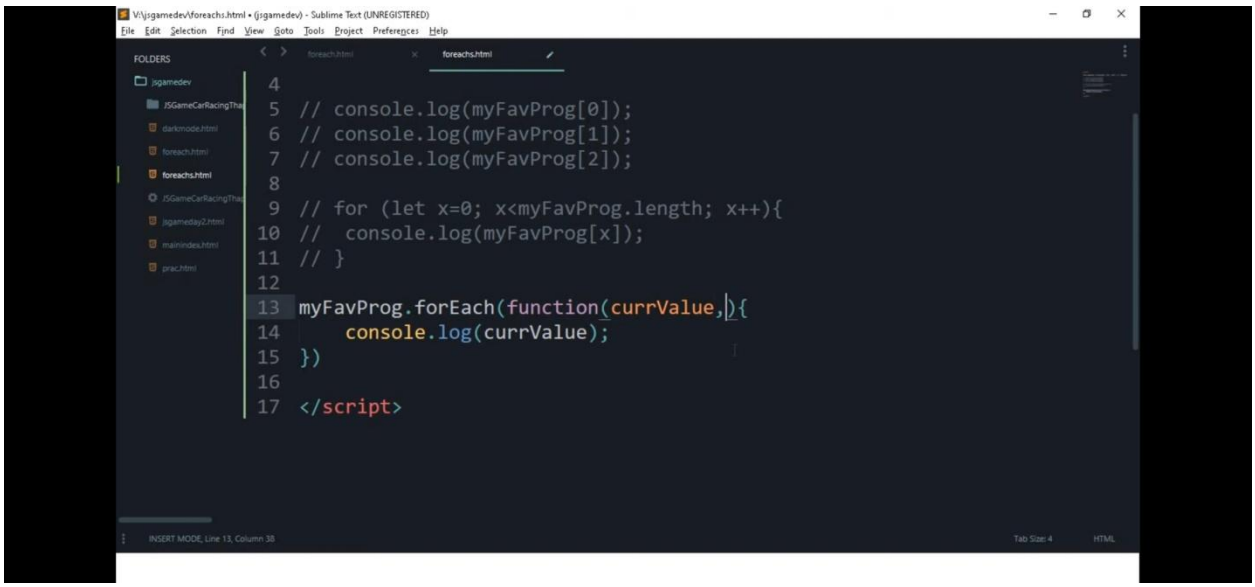

File Edit Selection View Go Run Terminal Help
index.html - js tutorial - Visual Studio Code
index.html
index.html > html > body > script
<title>Document</title>
</head>
<body>
<h1>JavaScript Tutorial</h1>
<script>
let score = [80,39,70,54];
for(let x of score){
console.log(x);
}
</script>
</body>
</html>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

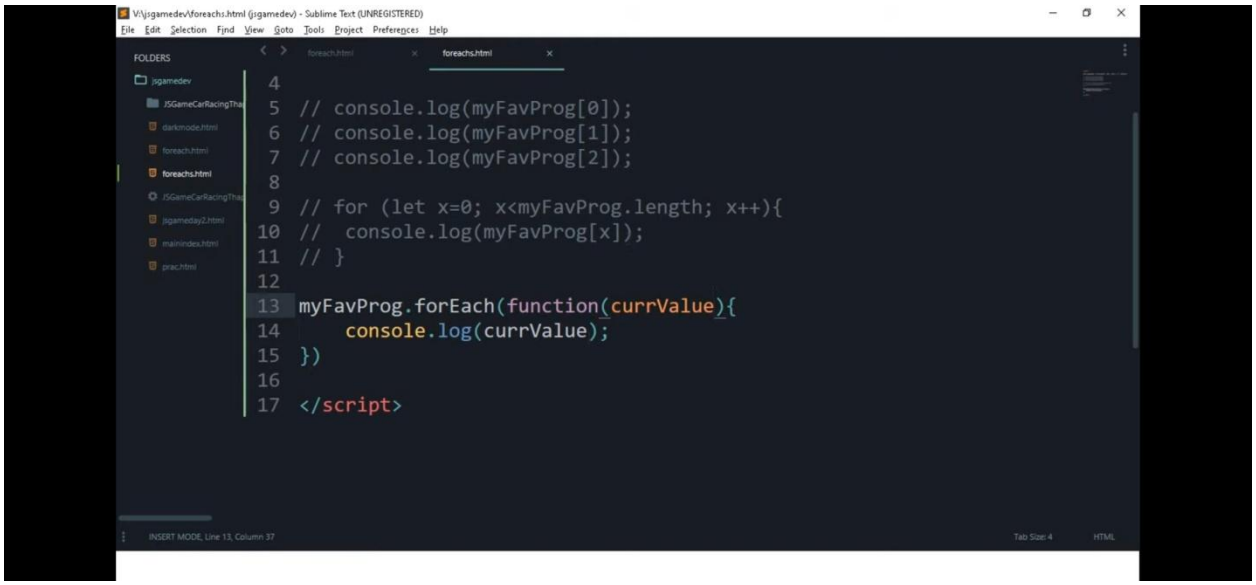
The forEach() method calls a function once for each element in an array, in order.

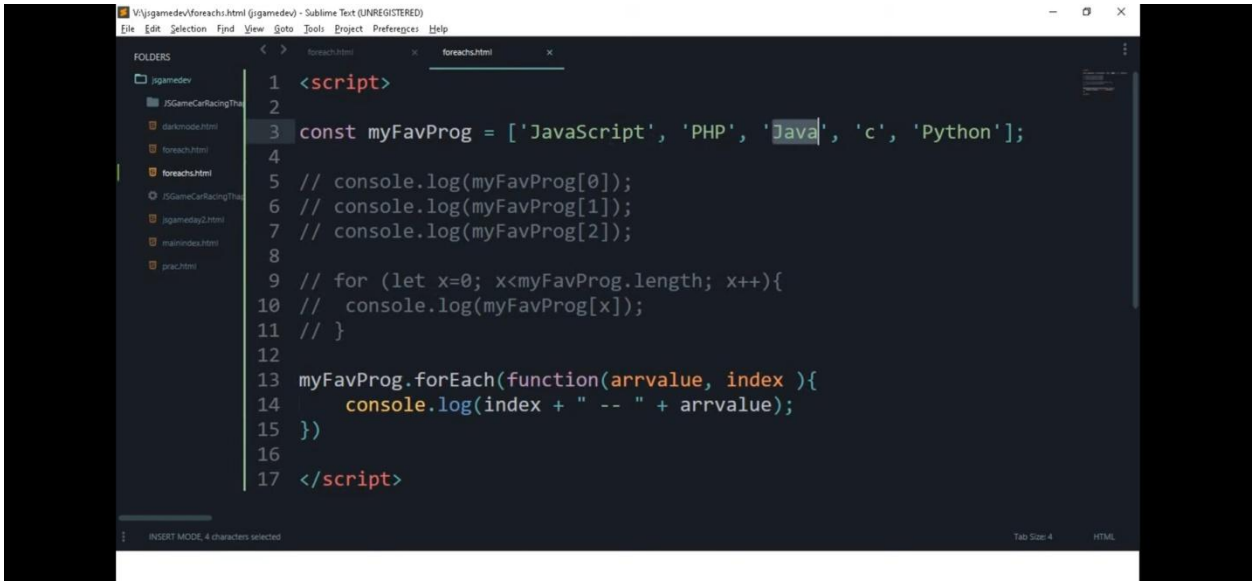
forEach() Method

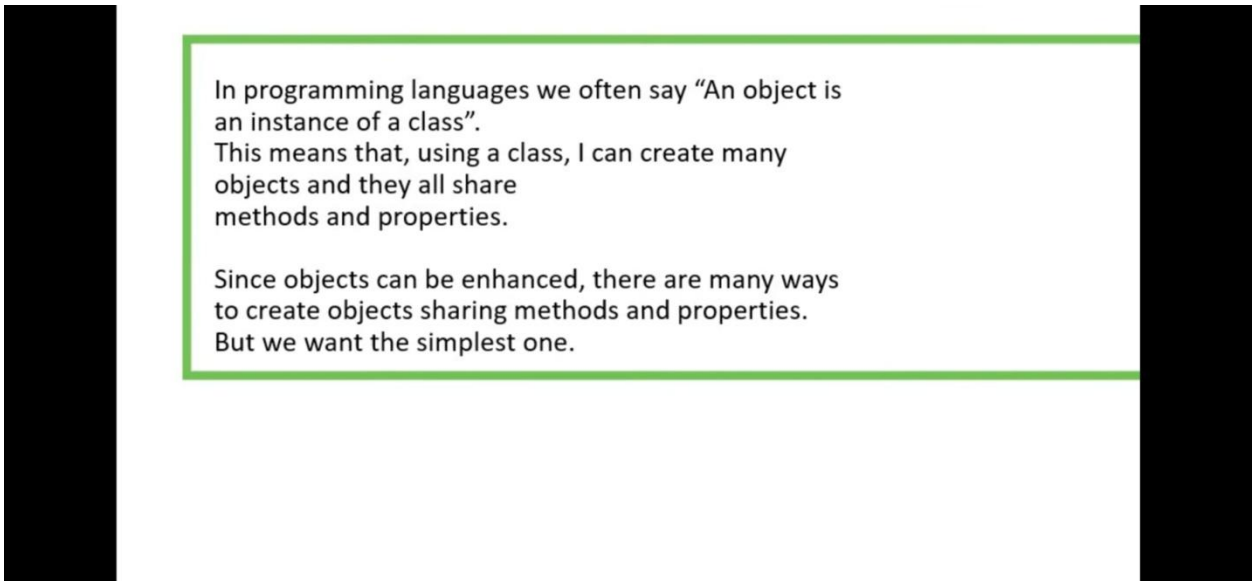
Syntax: arr.forEach(callback(currentValue [, index [, array]])[, thisArg])

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:

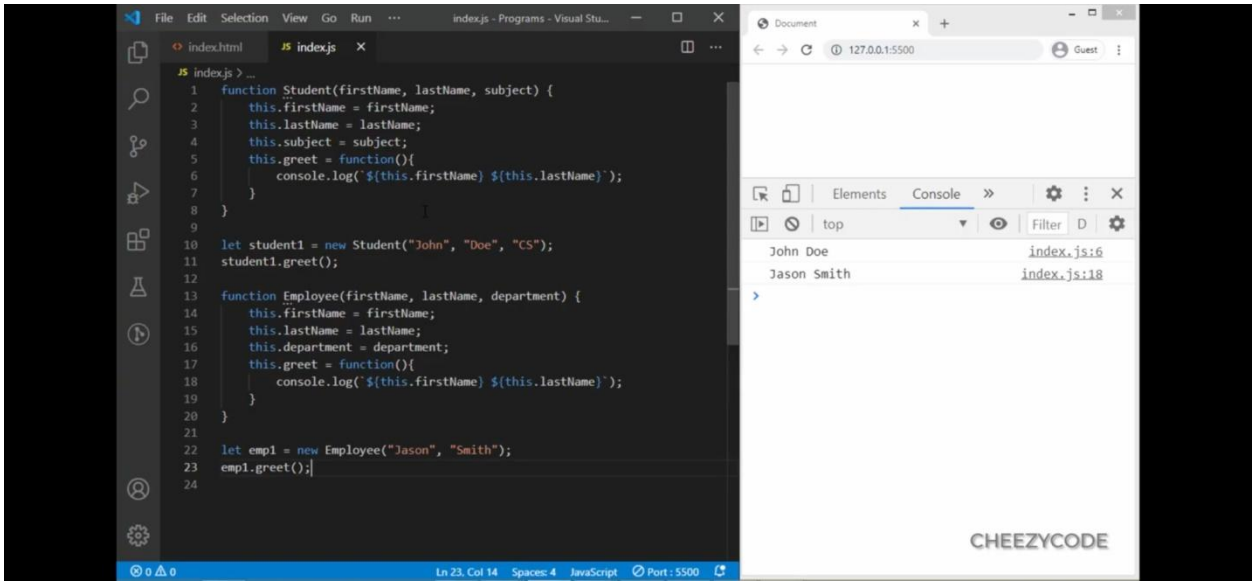
bind() Method

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>
```



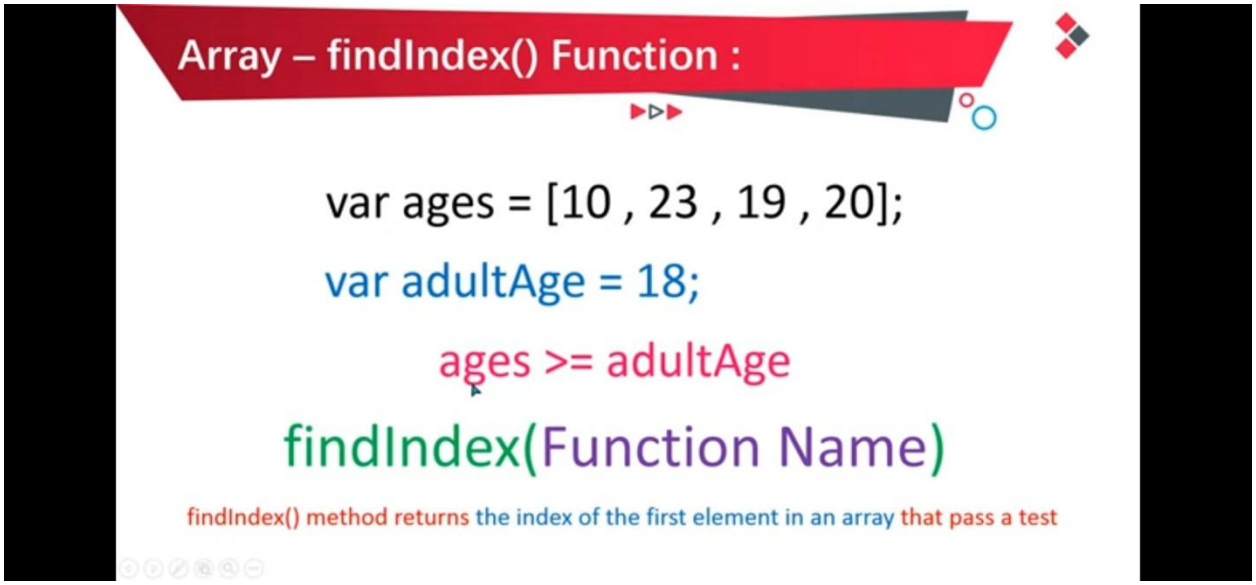

const person = {
firstName:"John",
lastName: "Doe",
fullName: function () {
return this.firstName + " " + this.lastName;
}
}
console.log(person.fullName()); // John Doe
JS
SUBSCRIBE


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

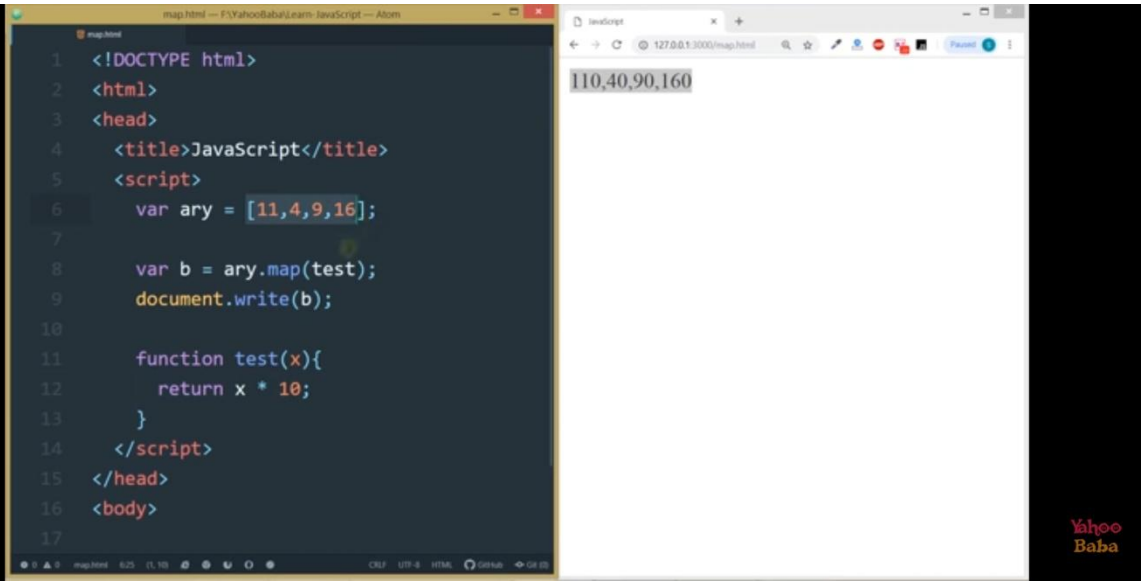

const obj = { name: "John" };
let greeting = function(a,b){
return `${a} ${this.name}. ${b}`;
}
let bound = greeting.bind(obj);
console.log(bound("Hello", "How are you?"));
// returns: Hello John. How are you?


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

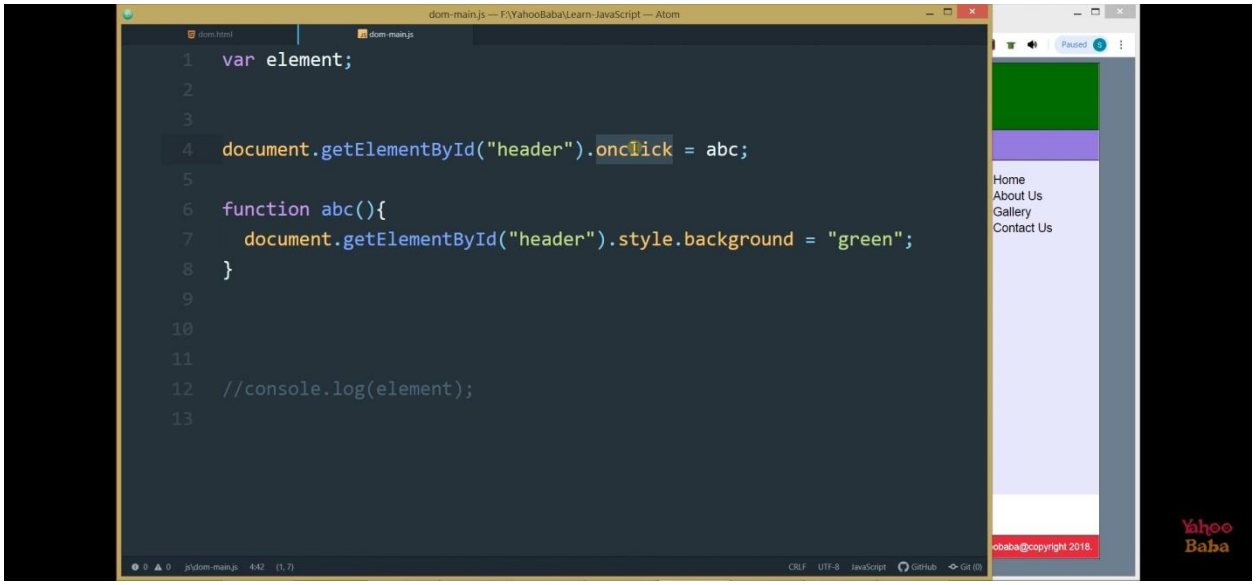
Why do we use ESNext, JSX, Babel & Webpack?

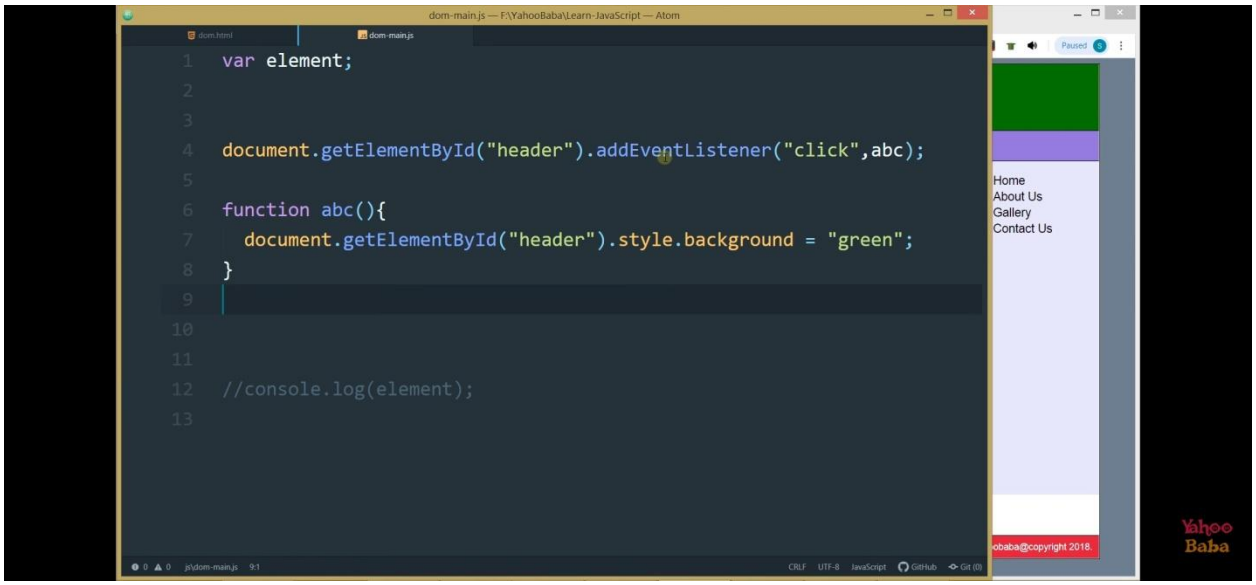
- ESNext makes for simpler code that is easier to read and write.
- Convert ESNext & JSX syntax to code browser can understand
- Using transformation tools help in optimizing our code for widest variety of browsers.
- Organize your code into smaller modules and then bundle them together using Webpack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

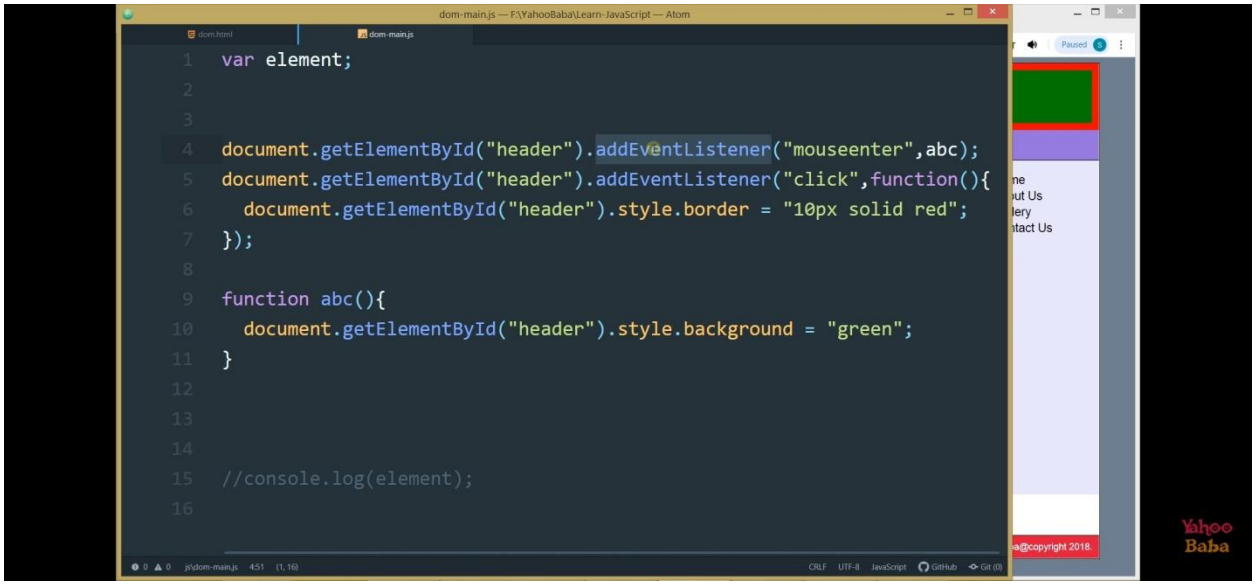
Why do we use ESNext, JSX, Babel & Webpack?

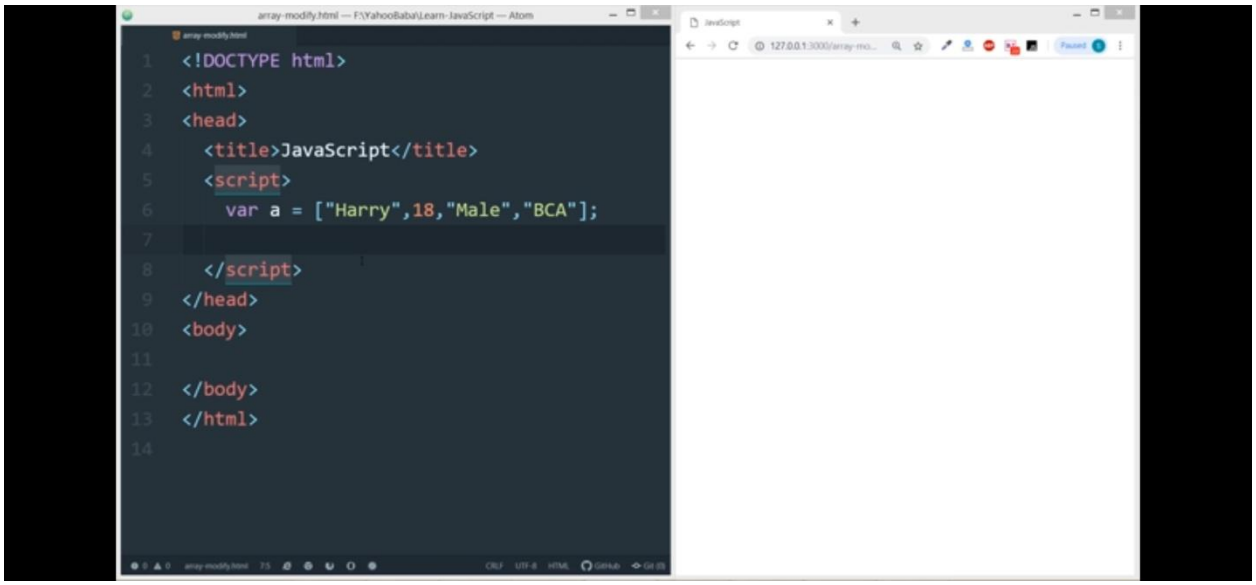
- Both webpack and Babel are tools written in JavaScript and run using Node.js (node)
- Node.js is a runtime environment for JavaScript outside of a browser
- node allows you to run JavaScript code on the command-line..


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

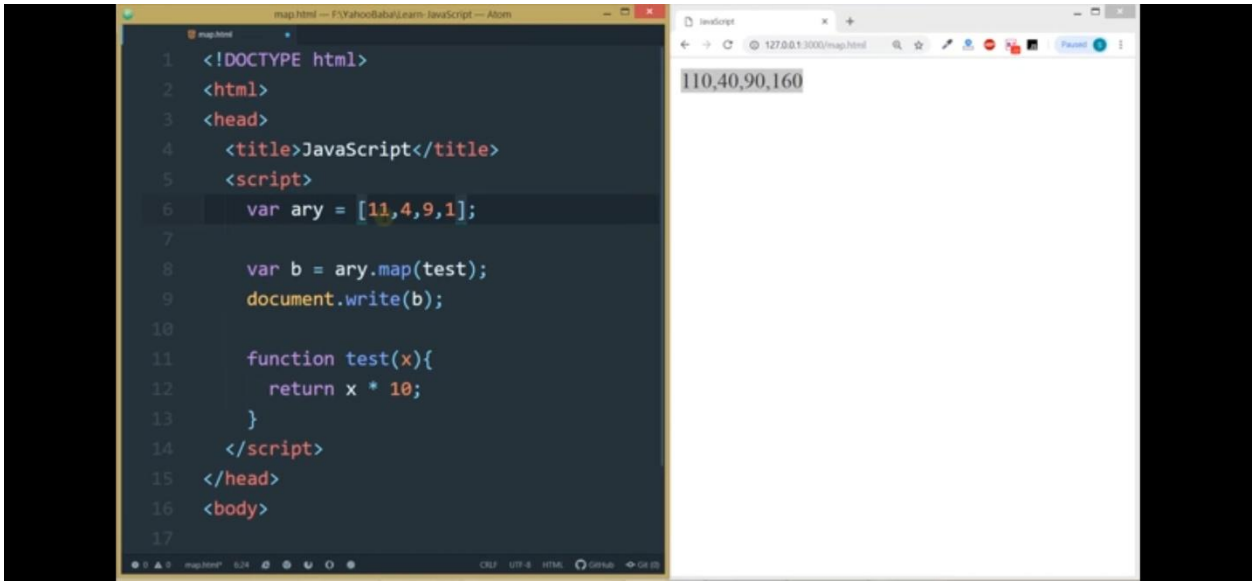
```
var a = [“Harry” , 18 , ”BCA”];
delete a[1];
```

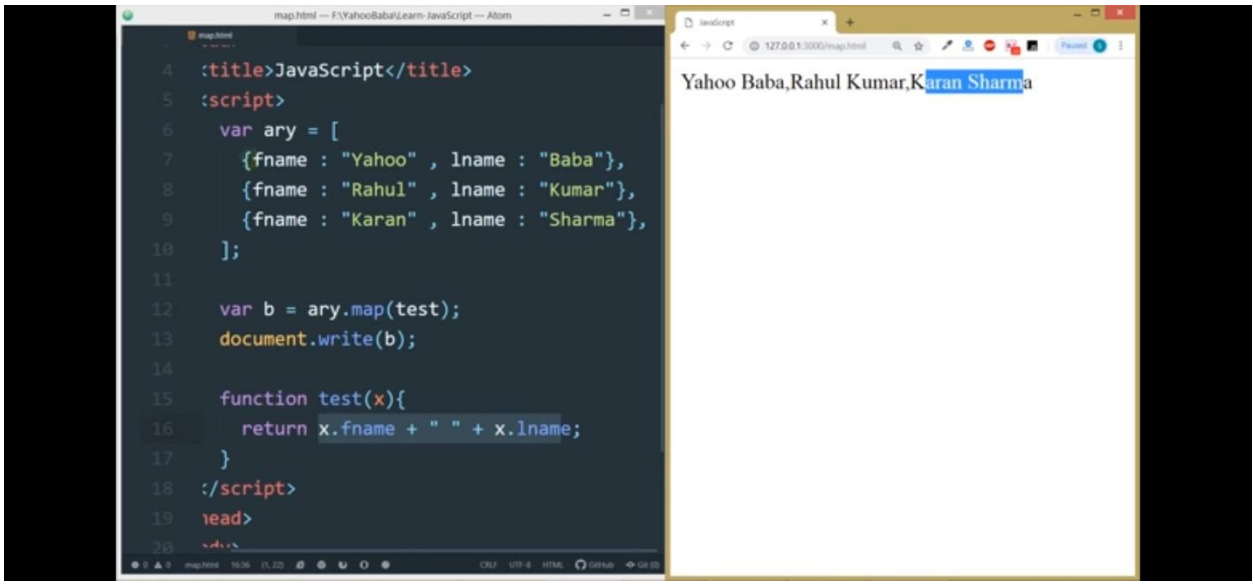
Array – Map() Function :

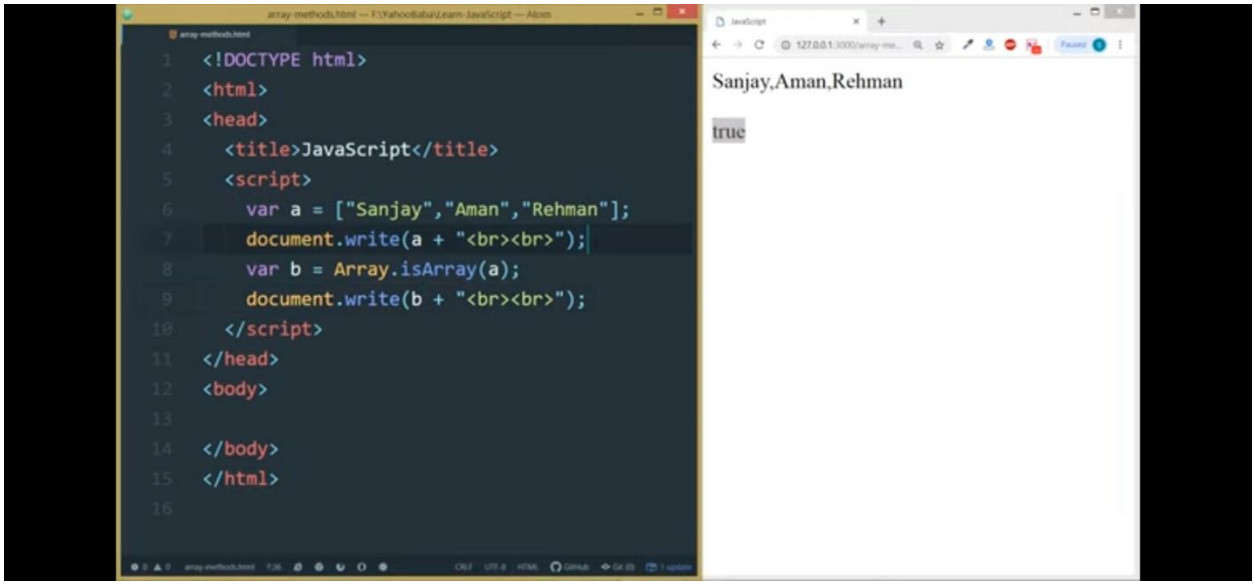
```
var a = [1 , 3 , 5 , 8 , 10];
a.map(function(){
  Statement
});
```

Array – Map() Function :

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

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

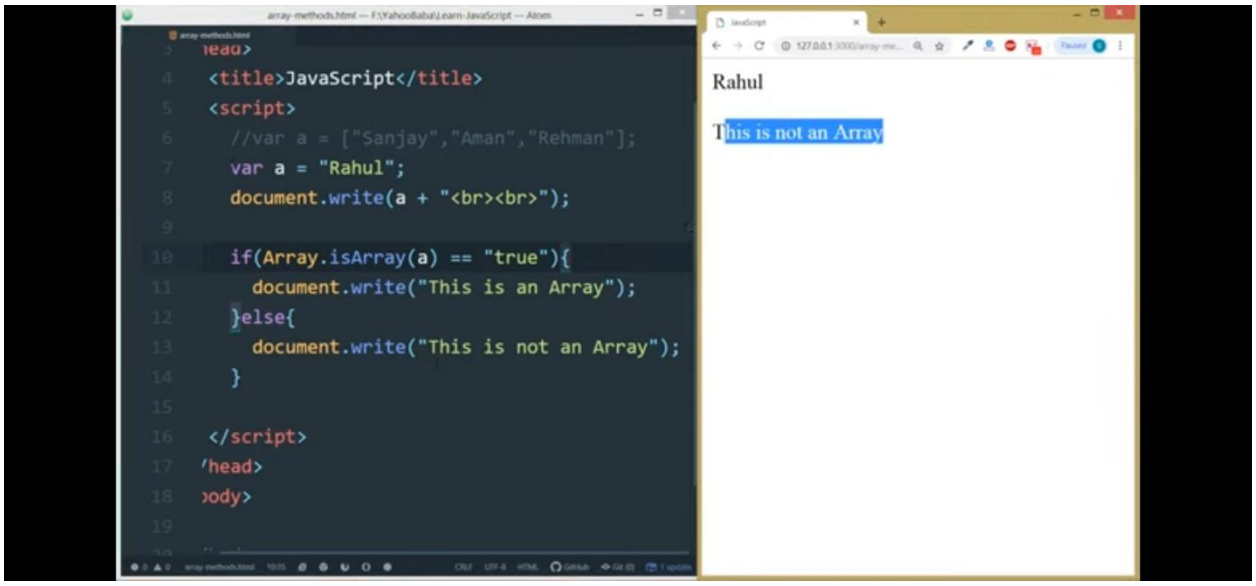



map.html — F:\YahooBaba\Learn-JavaScript — Atom
:title>JavaScript</title>
:script>
var ary = [
{fname : "Yahoo" , lname : "Baba"},
{fname : "Rahul" , lname : "Kumar"},
{fname : "Karan" , lname : "Sharma"},
];
var b = ary.map(test);
document.write(b);
function test(x){
return x.fname + " " + x.lname;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

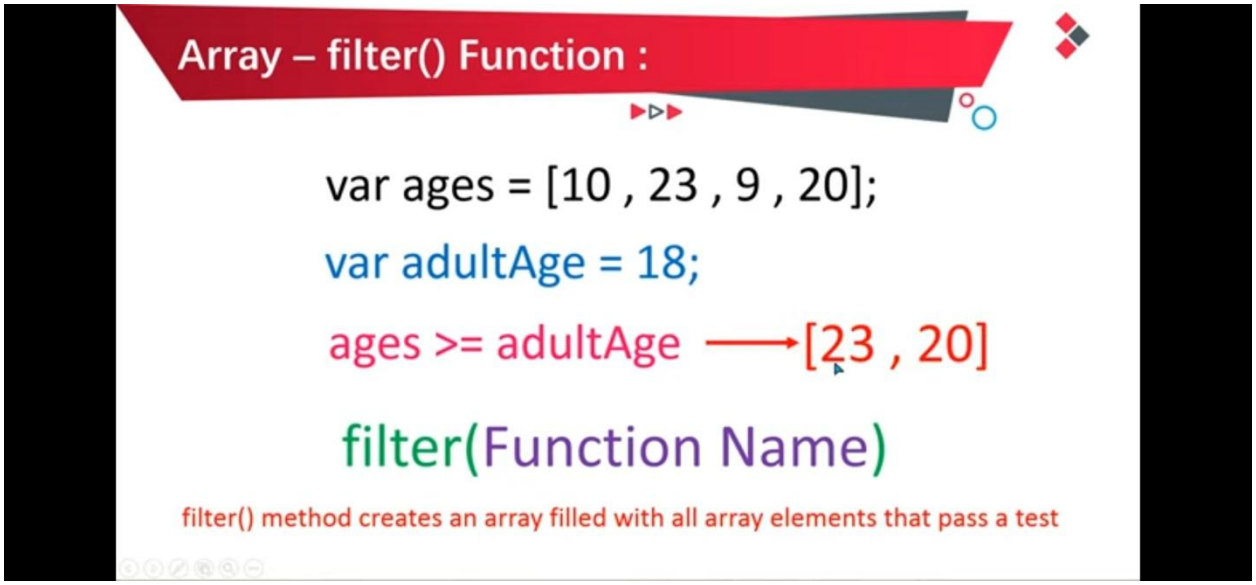

array-methods.html — F:\YahooBaba\Learn-JavaScript — Atom
<!DOCTYPE html>
<html>
<head>
<title>JavaScript</title>
<script>
var a = ["Sanjay","Aman","Rehman"];
document.write(a + "

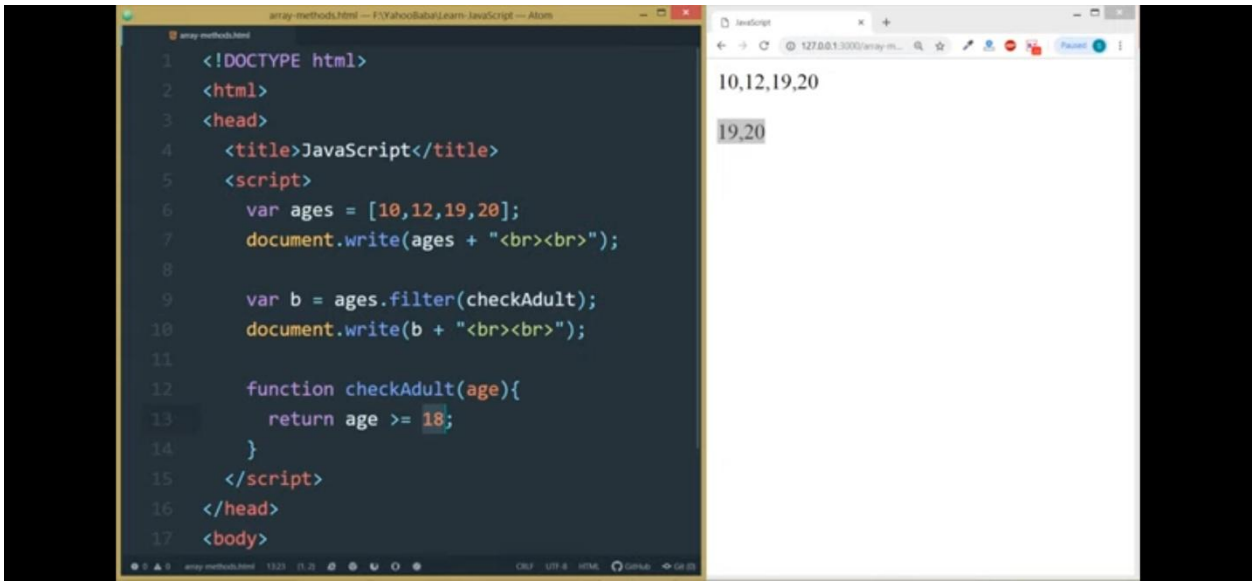
");
var b = Array.isArray(a);
document.write(b + "

");
</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 + "

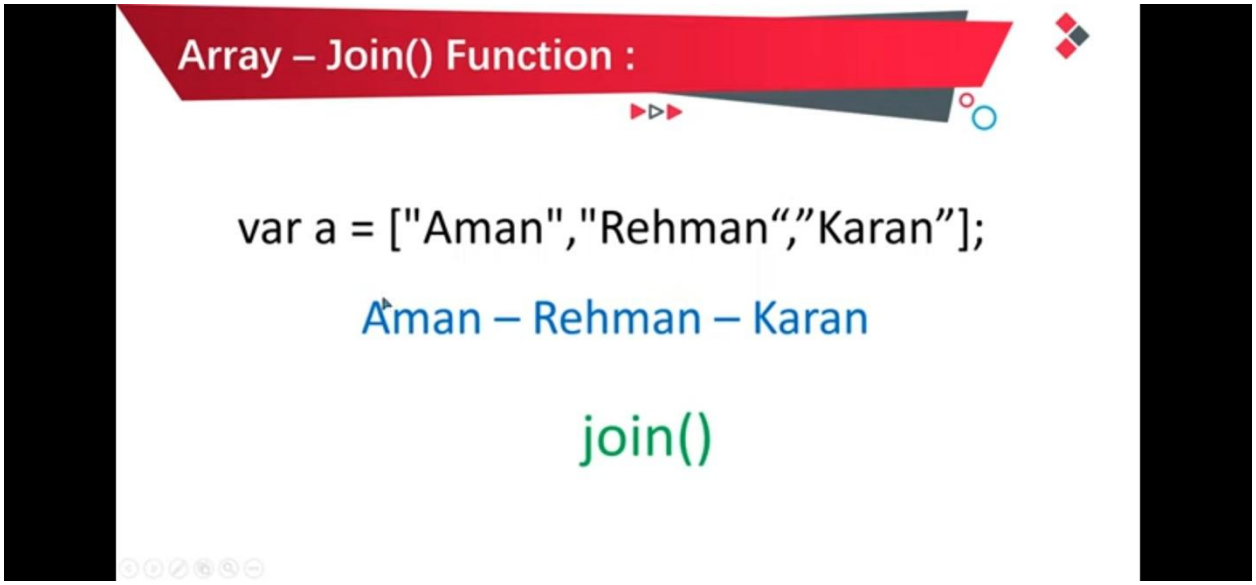
");
if(Array.isArray(a) == "true"){
document.write("This is an Array");
}else{
document.write("This is not an Array");
}
</script>
Rahul
This is not an Array

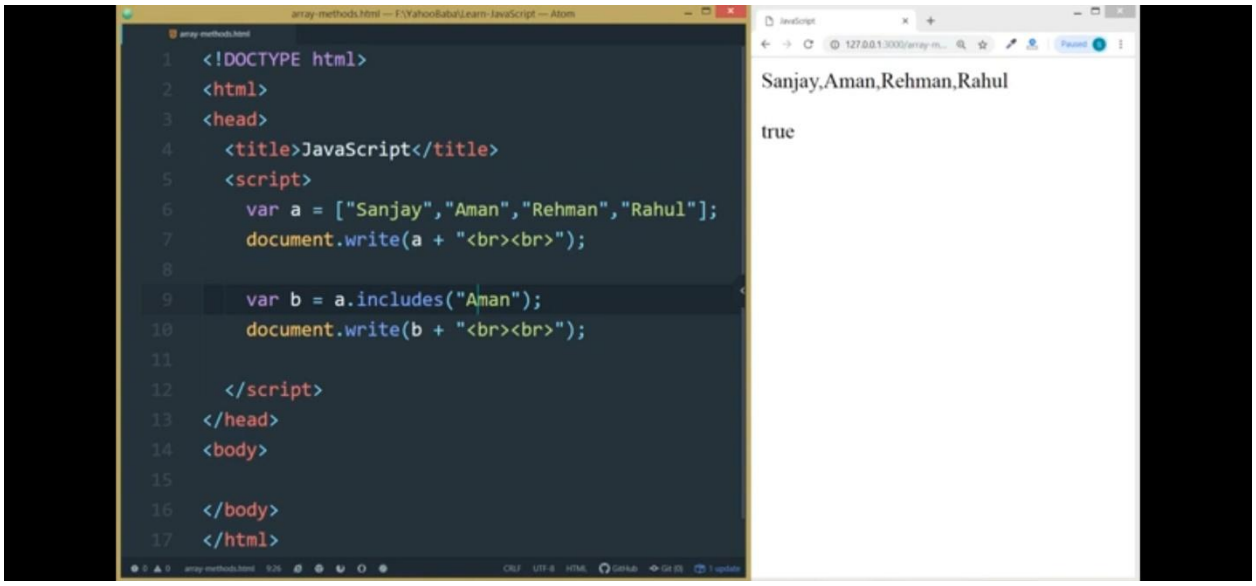

Array – filter() Function :
var ages = [10 , 23 , 9 , 20];
var adultAge = 18;
ages >= adultAge ⟶ [23 , 20]
filter(Function Name)
filter() method creates an array filled with all array elements that pass a test


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

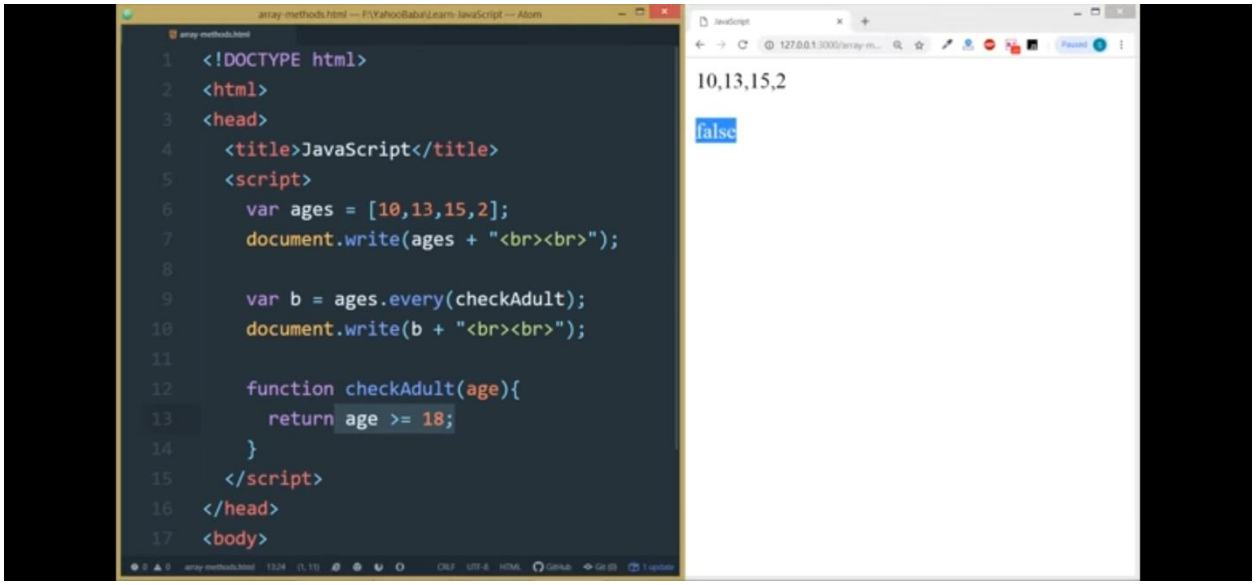

Array – Join() Function :
var a = ["Aman","Rehman","Karan"];
Aman – Rehman – Karan
join()



```
<!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

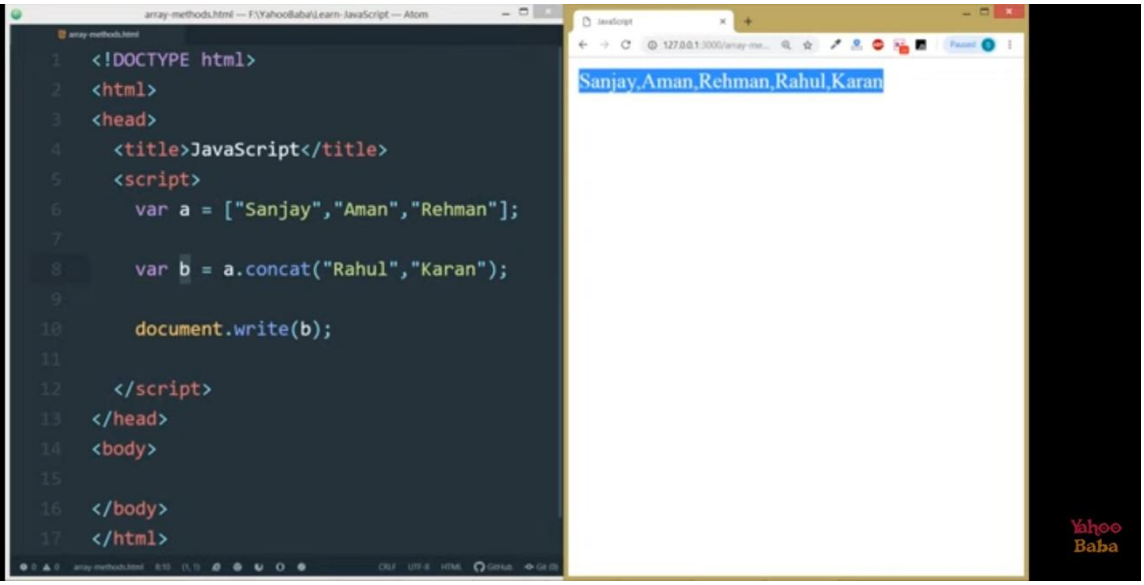
Array – indexOf() Function :

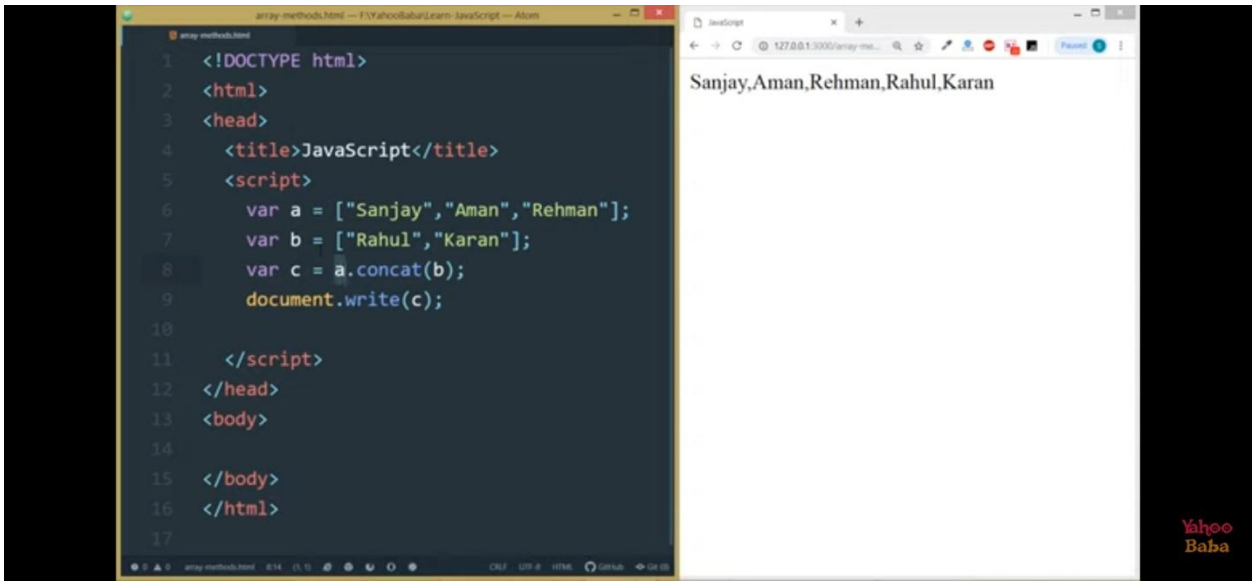
0 1 2 3

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

Aman ⟶ 1

indexOf("Search item" , Start)





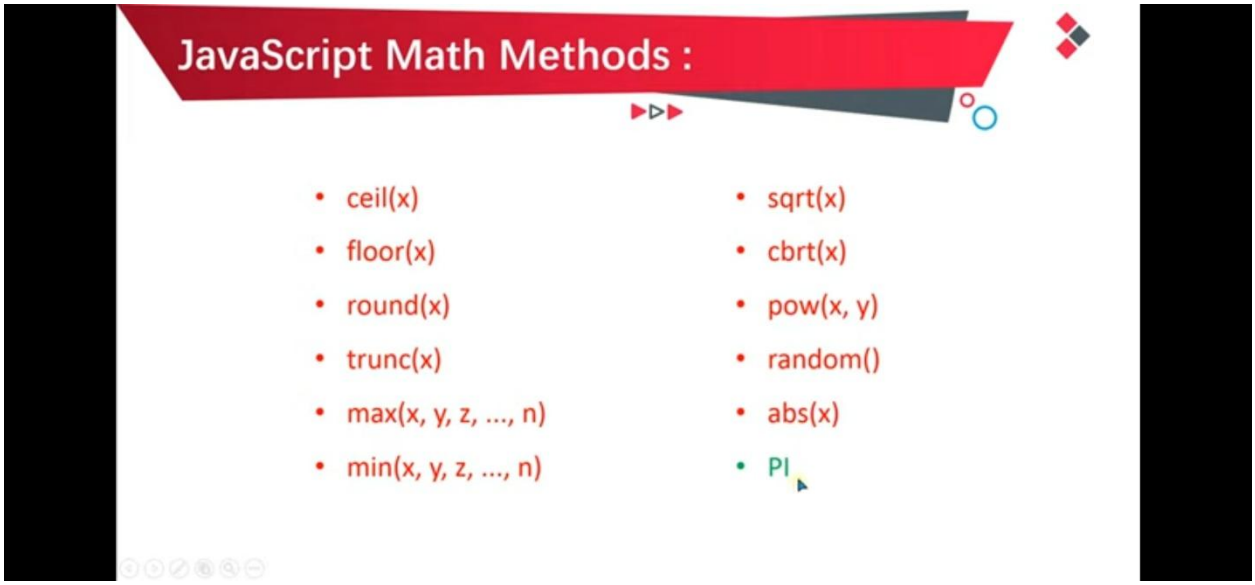
Converting an Arrays to Strings

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.

```
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>
```

JavaScript Number Methods :

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

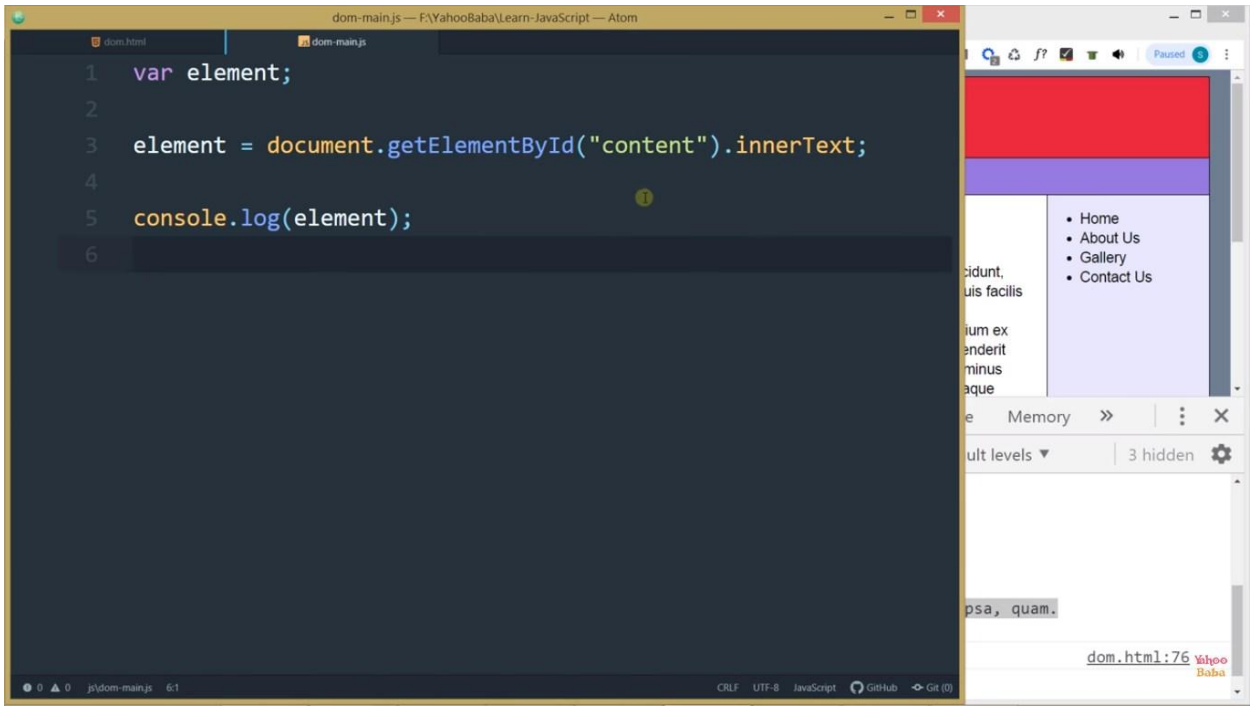
JavaScript Math Methods :

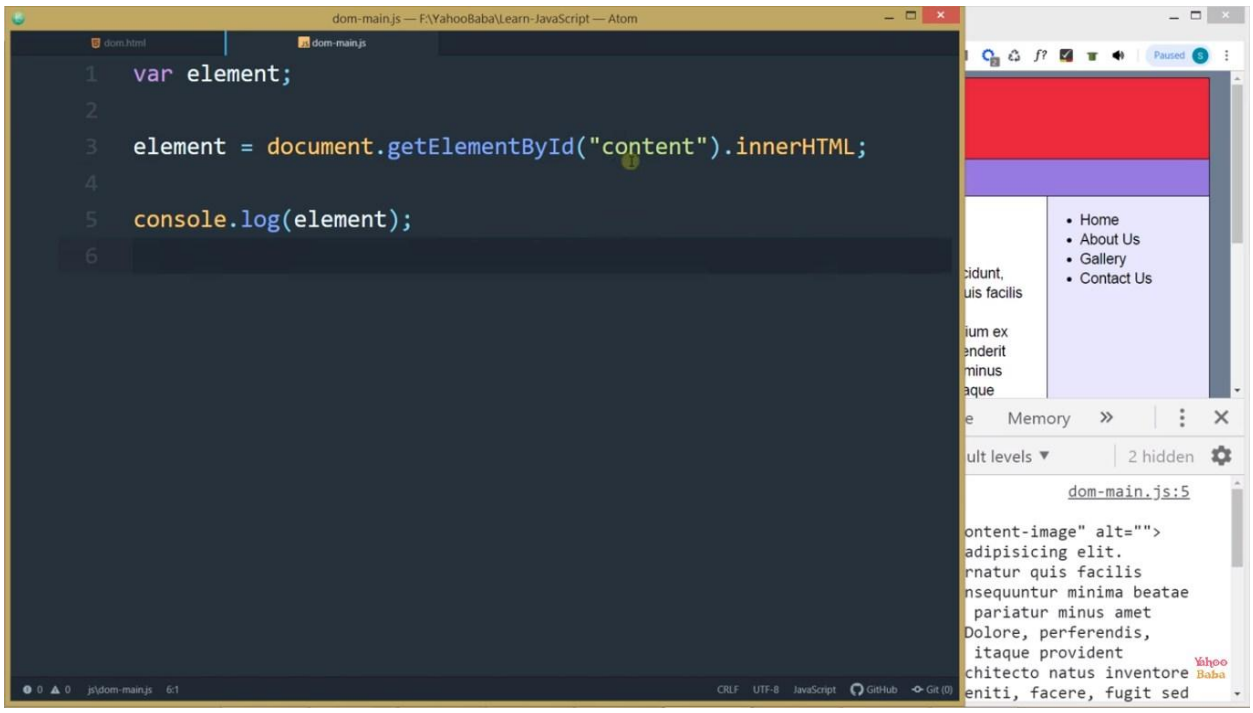
- toDateString()
- getDate()
- getFullYear()
- getMonth()
- getDay()
- getHours()
- getMinutes()
- getSeconds()
- getMilliseconds()
- setDate()
- setFullYear()
- setHours()
- setMilliseconds()
- setMinutes()
- setMonth()
- setSeconds()

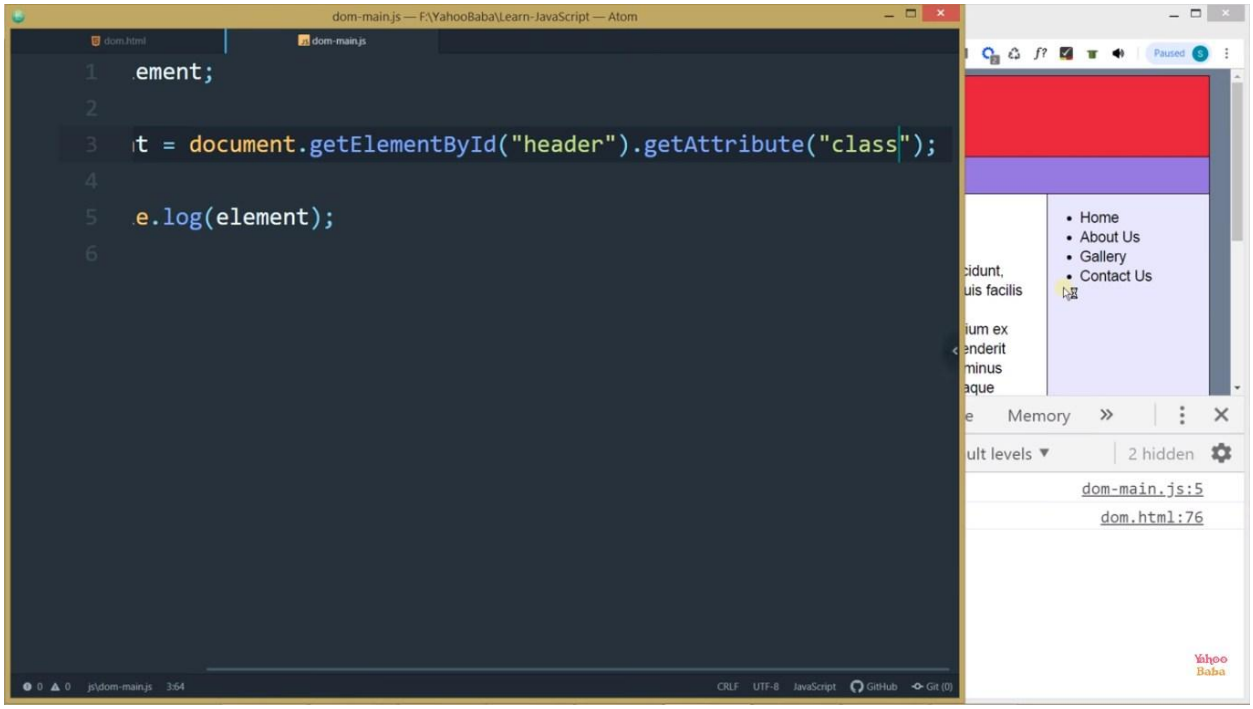
DOM Get Methods :

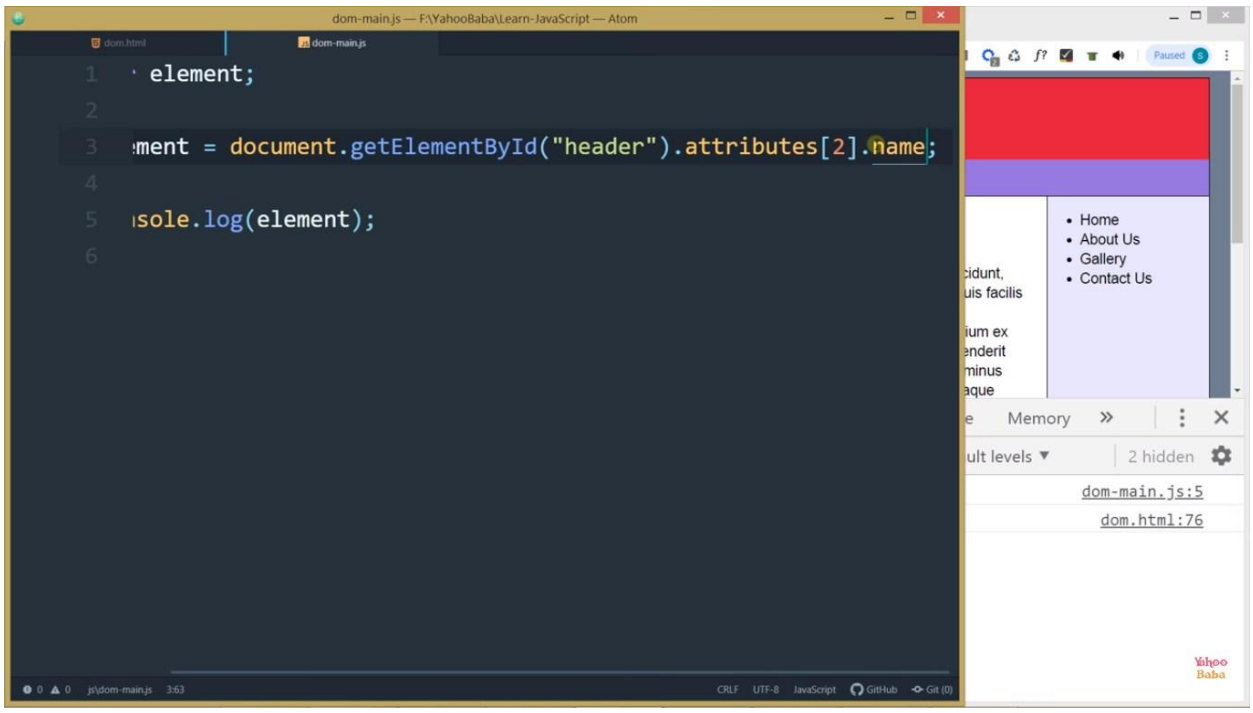
- innerText
- innerHTML
- getAttribute
- getAttributeNode
- Attributes

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

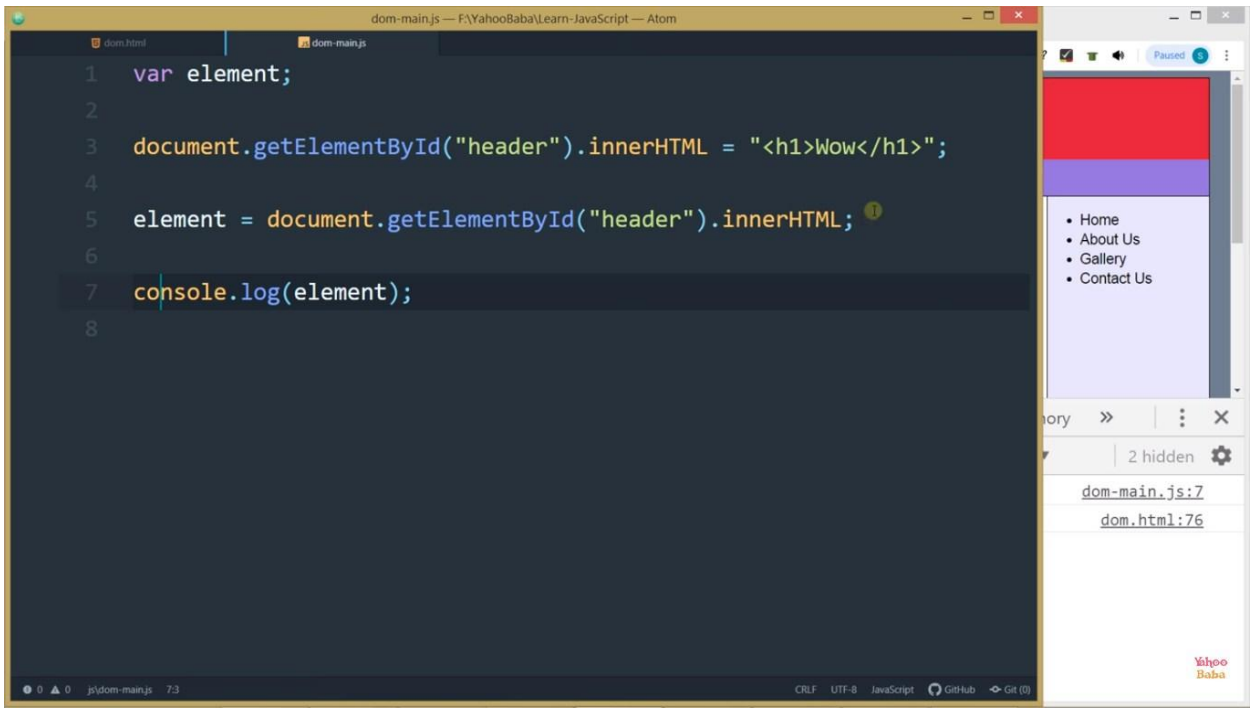

dom-main.js — F:\YahooBaba\Learn-JavaScript — Atom
dom.html
dom-main.js
ement;
t = document.getElementById("header").getAttribute("class");
e.log(element);
js\dom-main.js 3:64
CRLF UTF-8 JavaScript GitHub Git (0)
Home
About Us
Gallery
Contact Us
Memory
2 hidden
dom-main.js:5
dom.html:76

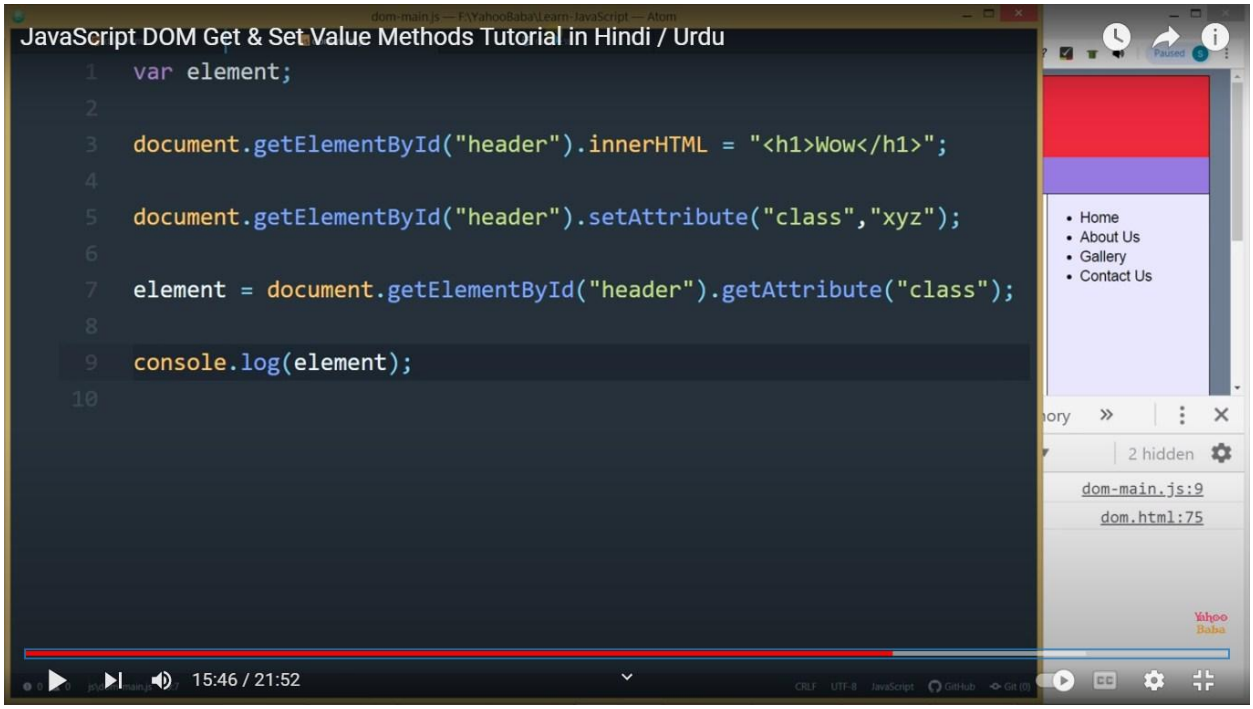


DOM Set Methods :

- innerText
- innerHTML
- setAttribute
- Attribute
- removeAttribute

Yahoo Baba

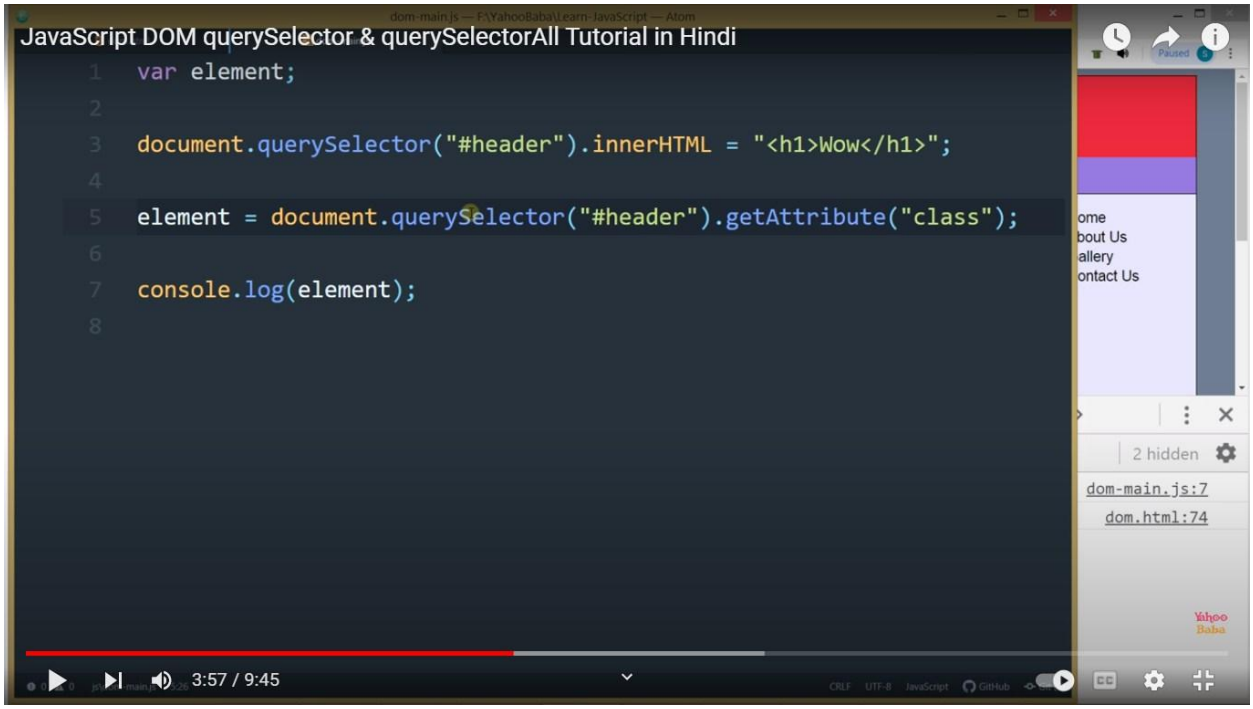

dom-main.js — F:\YahooBaba\Learn-JavaScript — Atom
dom.html
dom-main.js
1 var element;
2
3 document.getElementById("header").innerHTML = "<h1>Wow</h1>";
4
5 element = document.getElementById("header").innerHTML;
6
7 console.log(element);
8
Home
About Us
Gallery
Contact Us
2 hidden
dom-main.js:7
dom.html:76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

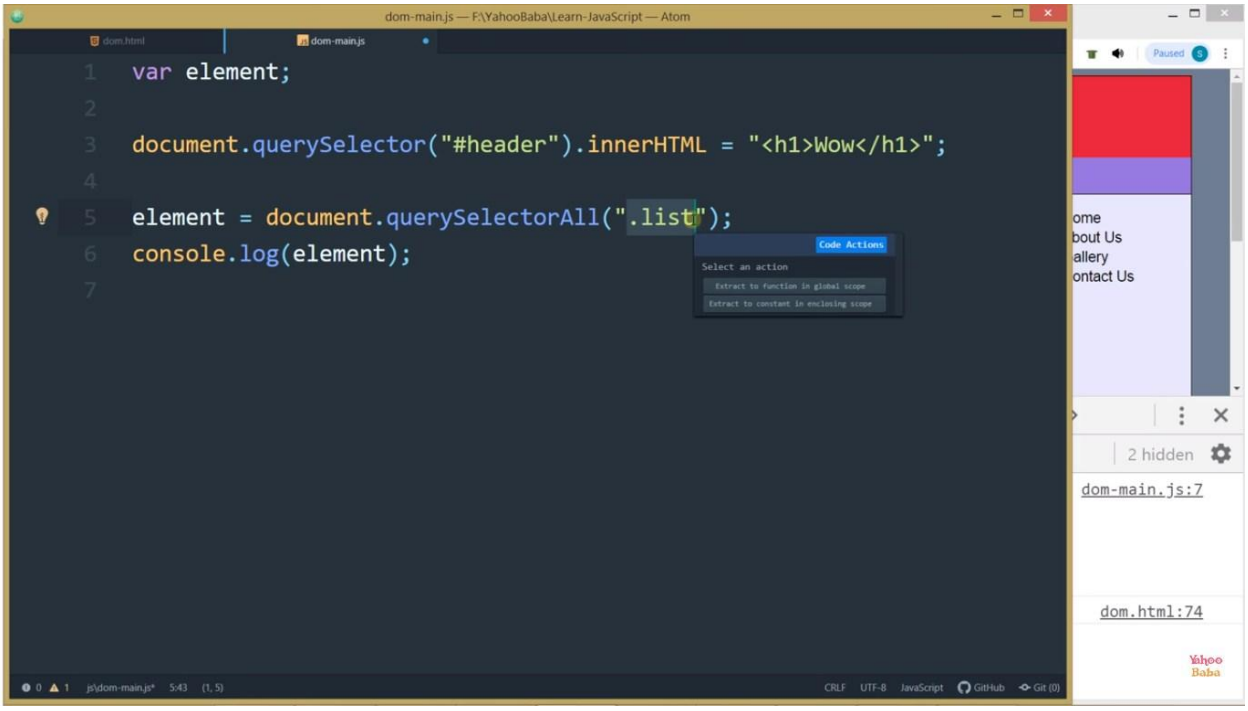

JavaScript DOM Get & Set Value Methods Tutorial in Hindi / Urdu
1 var element;
2
3 document.getElementById("header").innerHTML = "<h1>Wow</h1>";
4
5 document.getElementById("header").setAttribute("class","xyz");
6
7 element = document.getElementById("header").getAttribute("class");
8
9 console.log(element);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

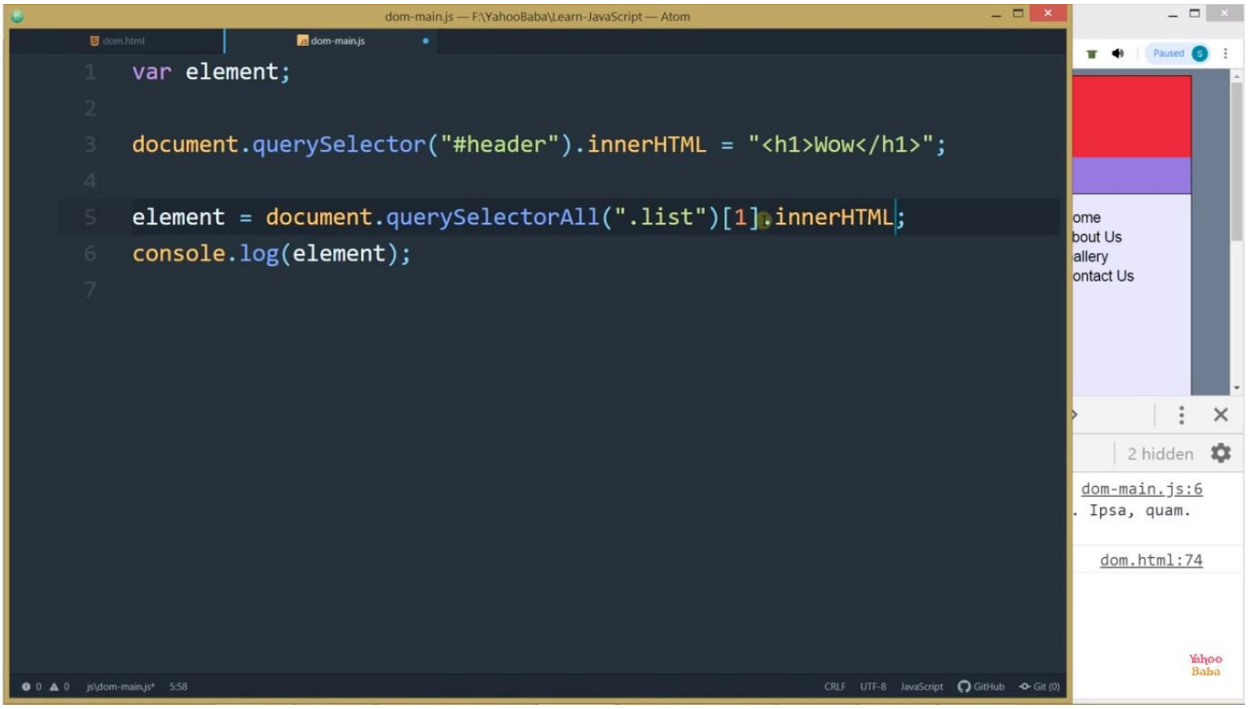
New DOM Targeting Methods :

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

Yahoo Baba



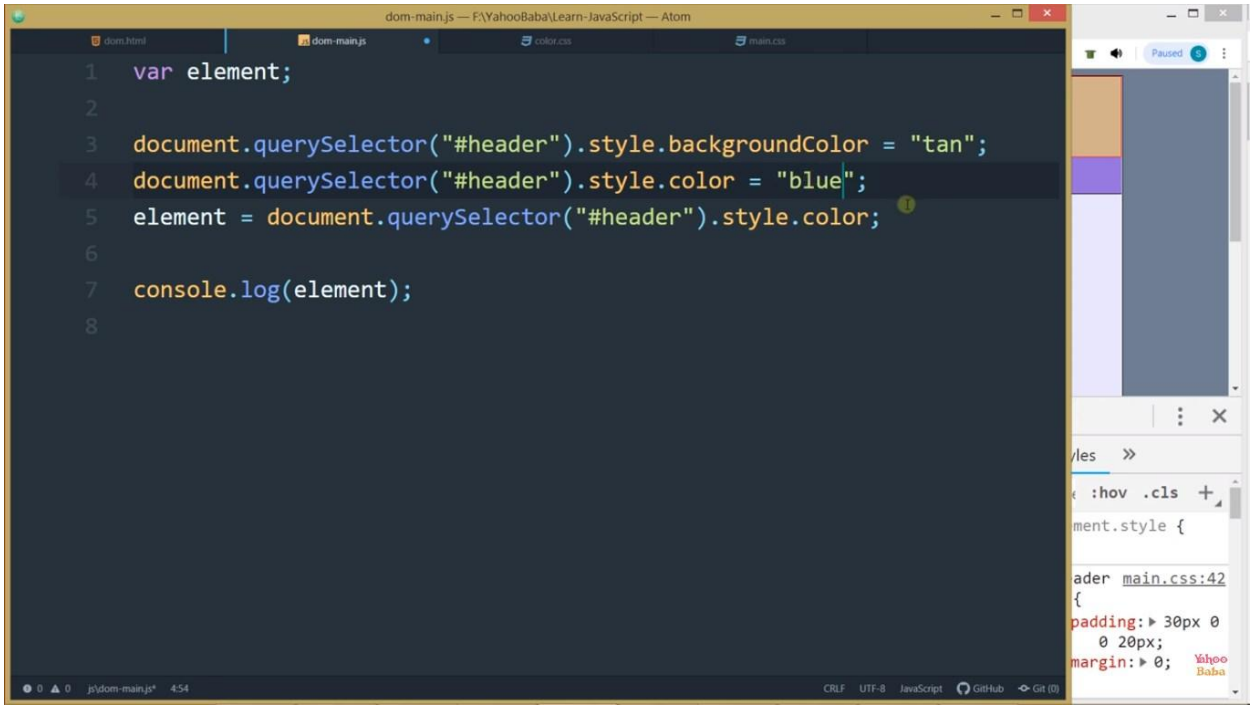

dom-main.js — F:\YahooBaba\Learn-JavaScript — Atom
dom.html
dom-main.js
var element;
document.querySelector("#header").innerHTML = "<h1>Wow</h1>";
element = document.querySelectorAll(".list");
console.log(element);
Code Actions
Select an action
Extract to function in global scope
Extract to constant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

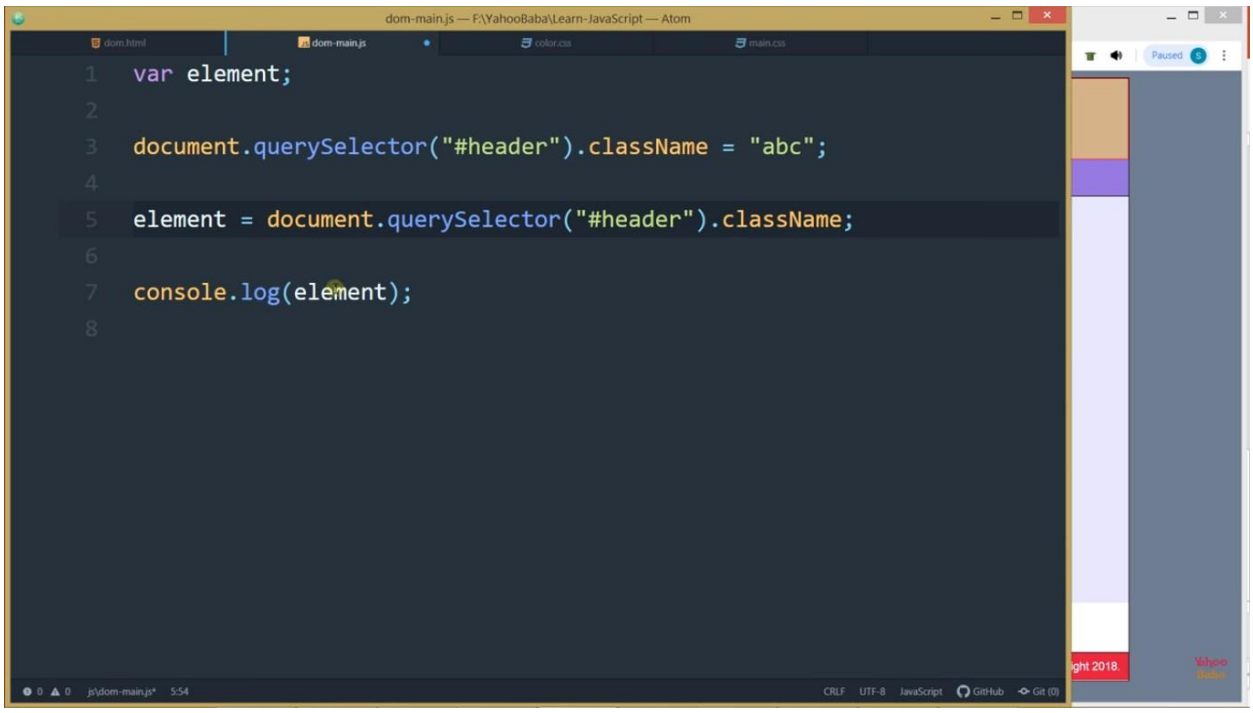

dom-main.js — F:\YahooBaba\Learn-JavaScript — Atom
dom.html
dom-main.js
var element;
document.querySelector("#header").innerHTML = "<h1>Wow</h1>";
element = document.querySelectorAll(".list")[1].innerHTML;
console.log(element);
ome
bout Us
allery
ontact Us
2 hidden
dom-main.js:6
. Ipsa, quam.
dom.html:74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

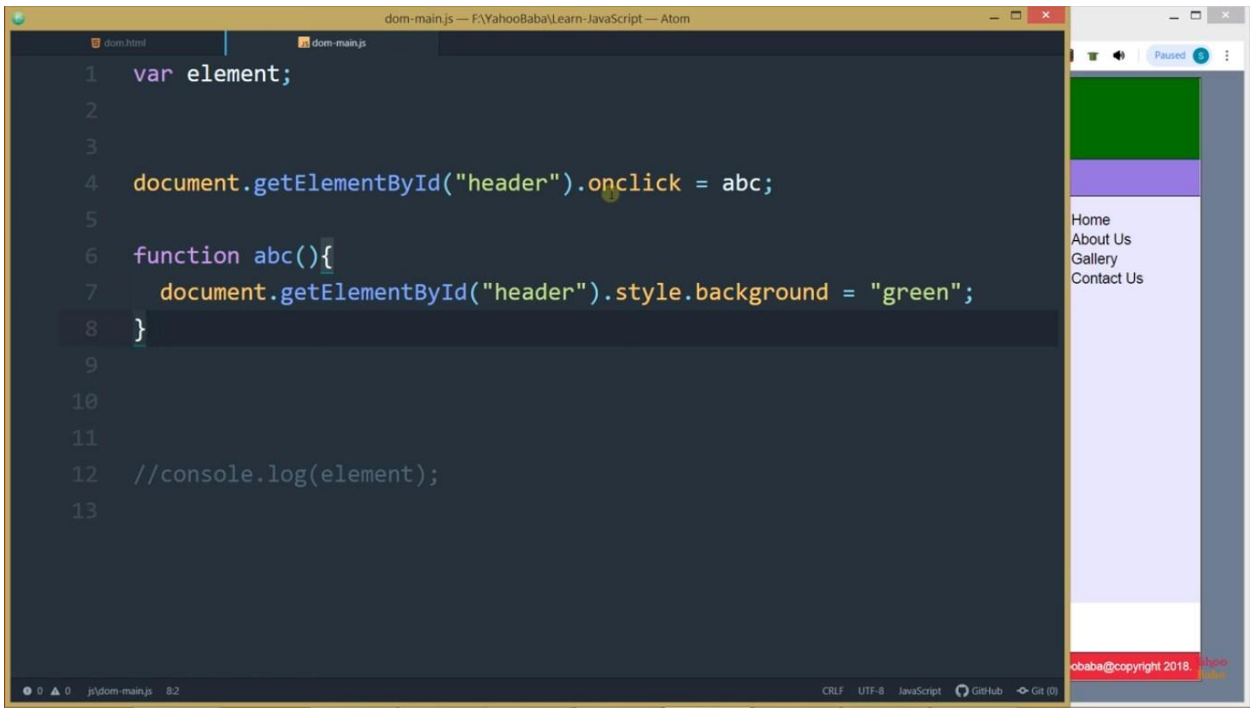
Yahoo Baba

HTML Event Attributes :

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

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


Assign Events Using the HTML DOM :

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



DOM addEventListener() Method :

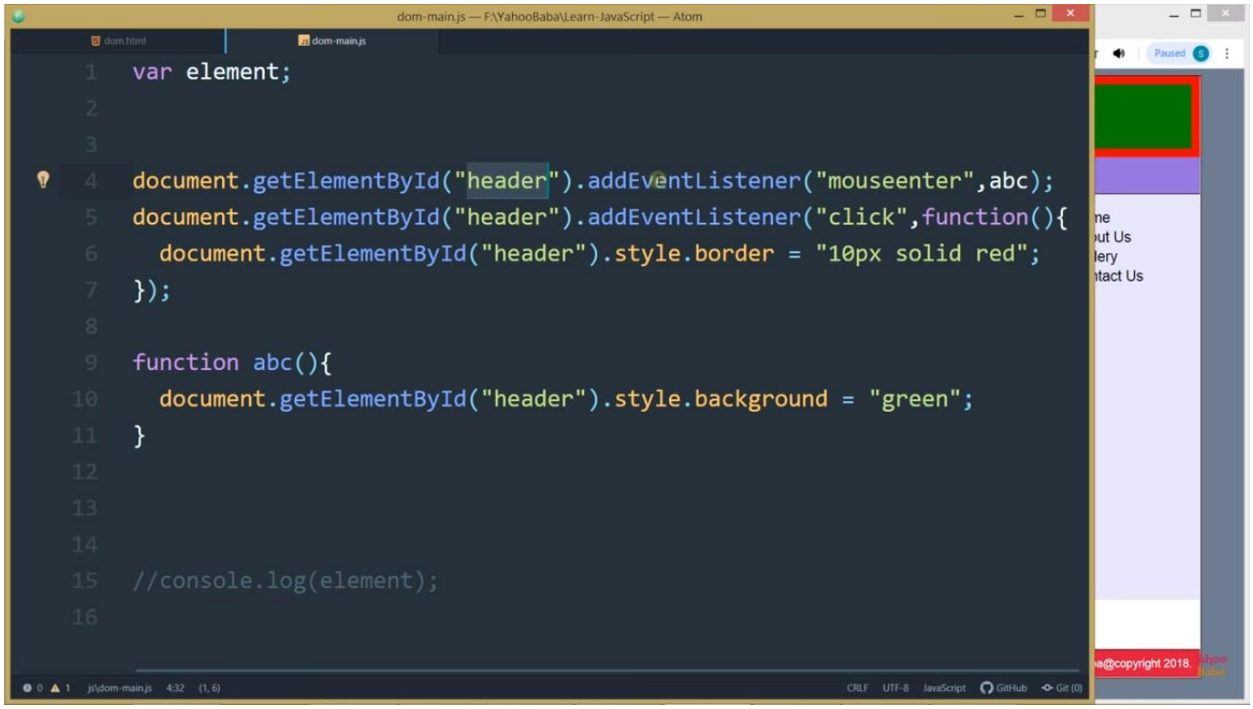
```
document.getElementById(Id). addEventListener("click", functionName);
```

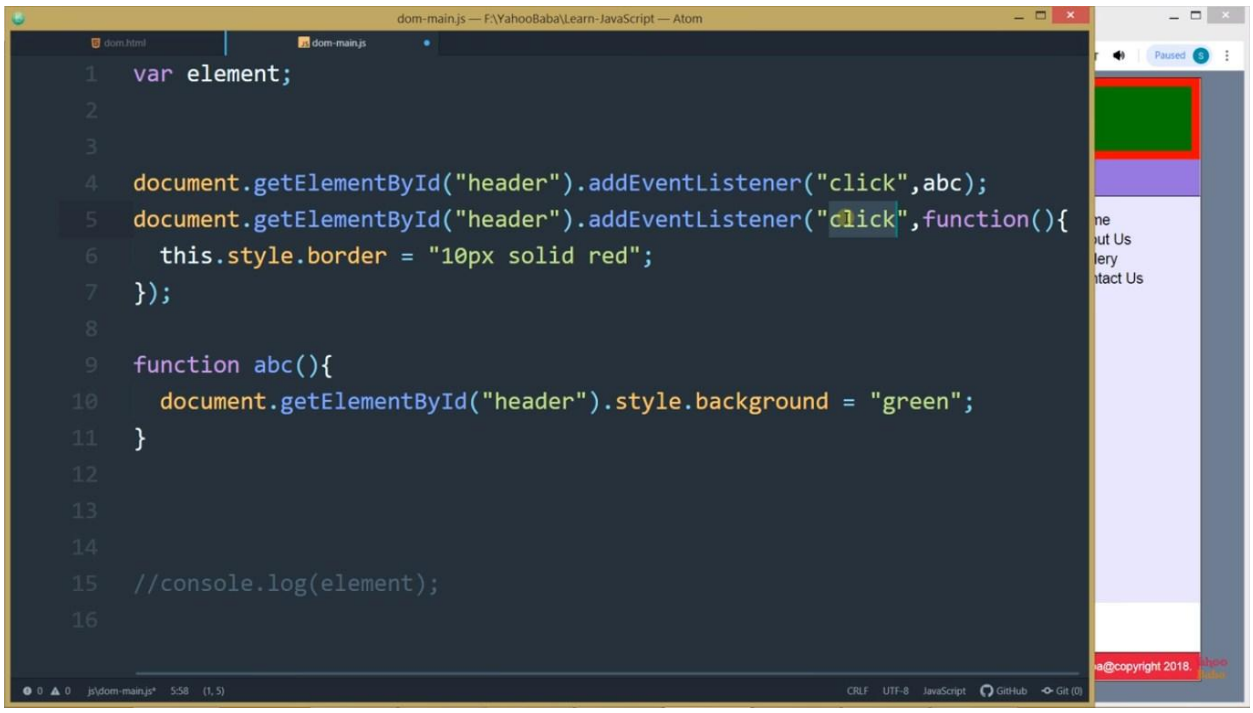
Yahoo Baba

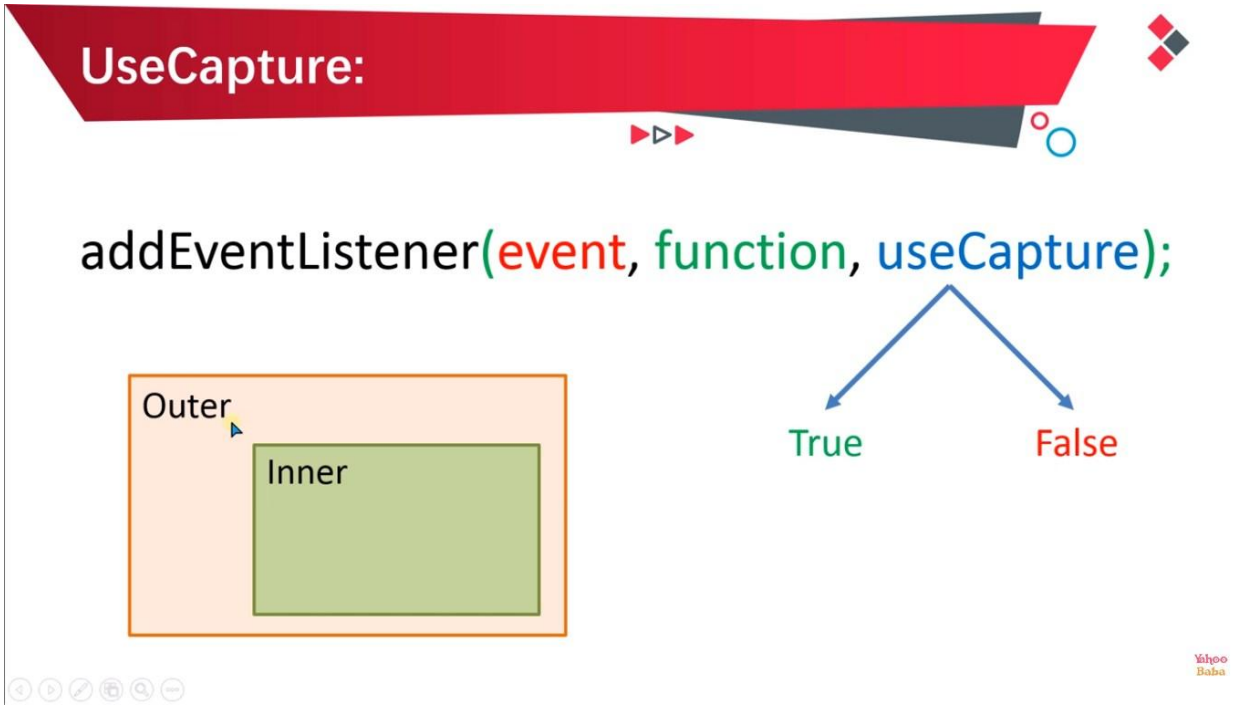
DOM addEventListener() Method :

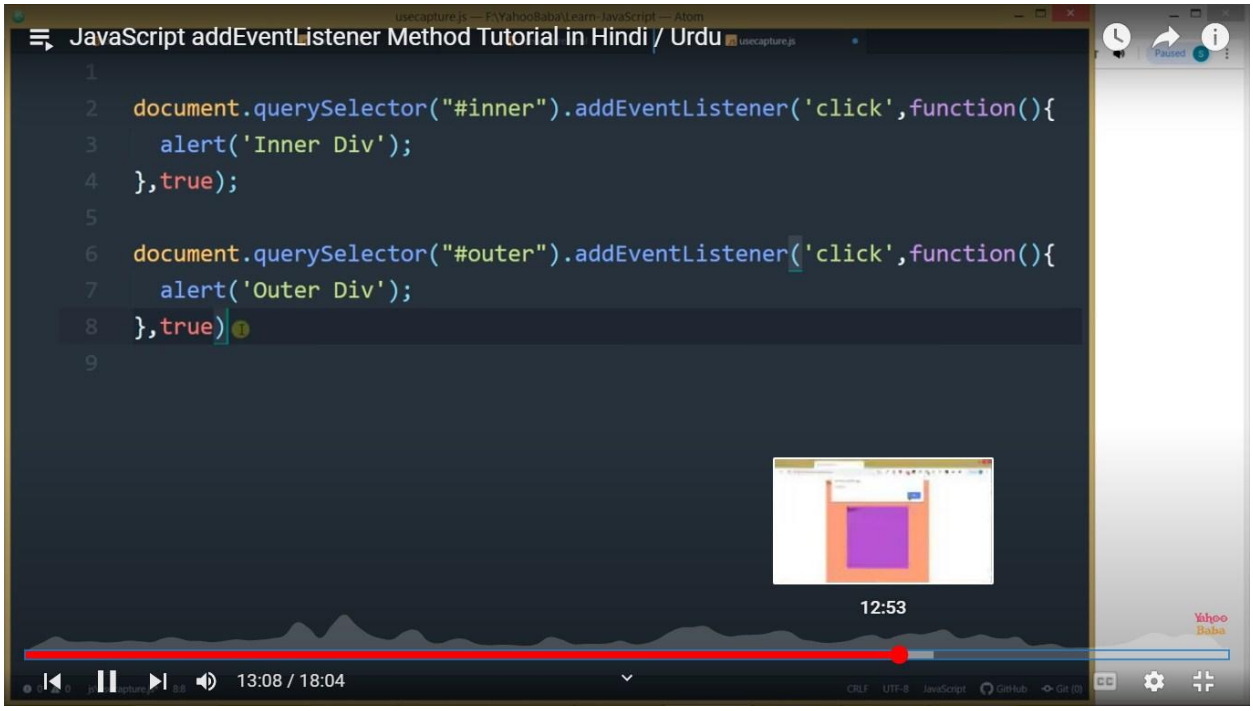
```
document.getElementById(Id). addEventListener("click", functionName);

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

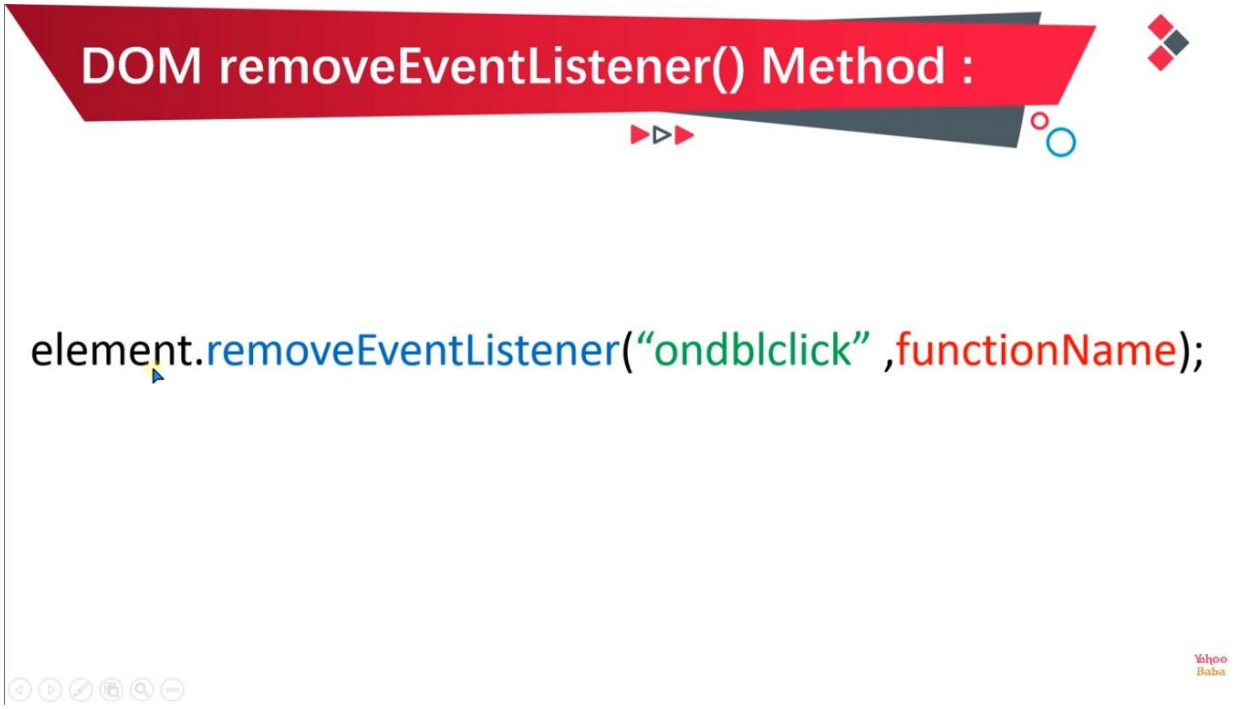
Yahoo Baba

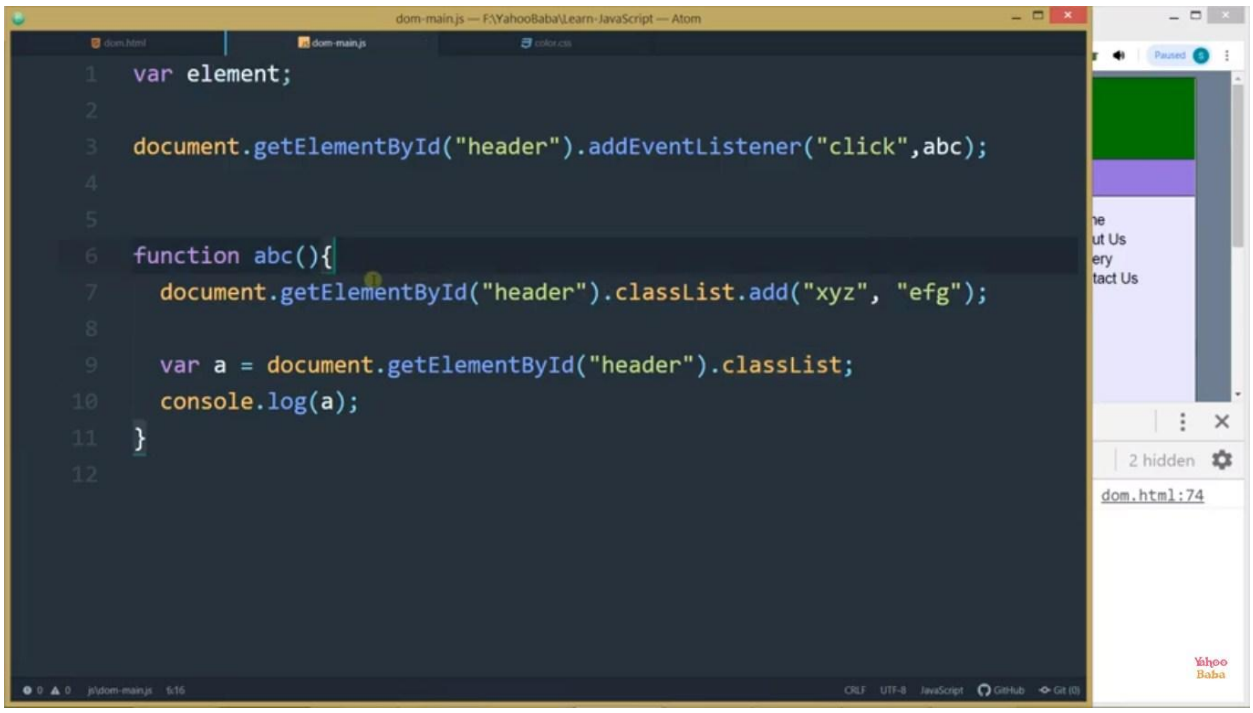


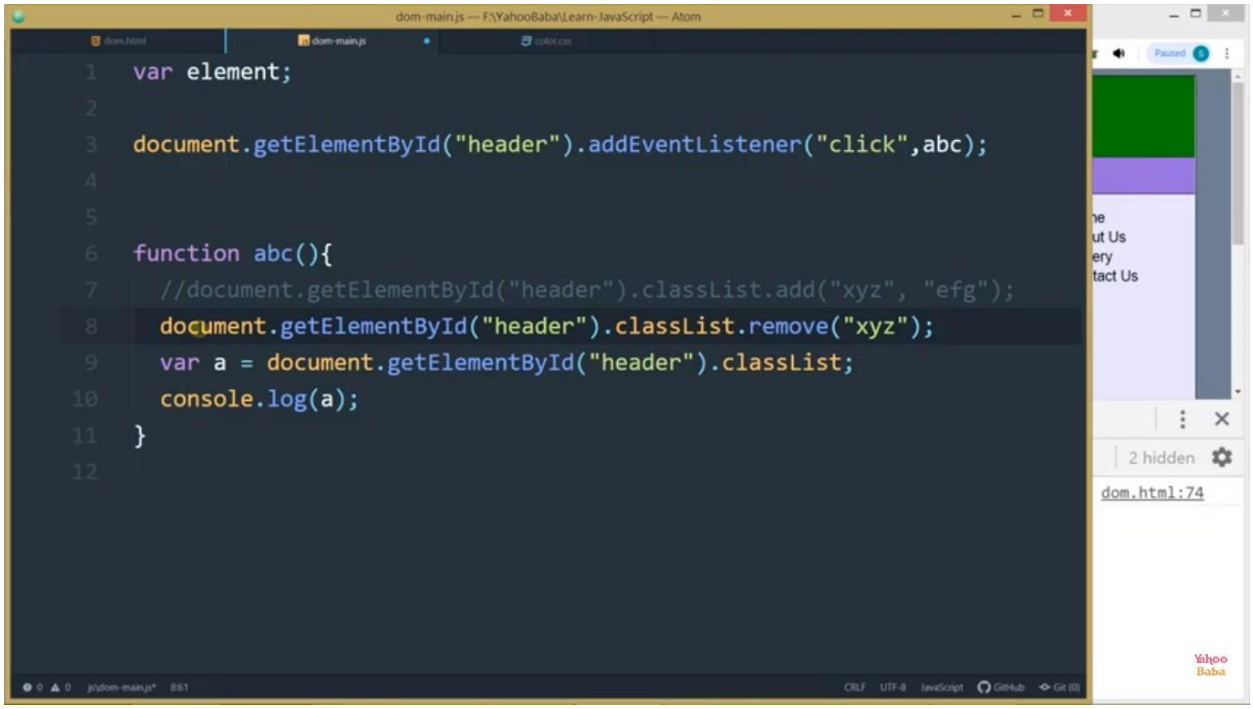

dom-main.js — F:\YahooBaba\Learn-JavaScript — Atom
dom.html
dom-main.js
1 var element;
2
3
4 document.getElementById("header").addEventListener("click",abc);
5 document.getElementById("header").addEventListener("click",function(){
6 this.style.border = "10px solid red";
7 });
8
9 function abc(){
10 document.getElementById("header").style.background = "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

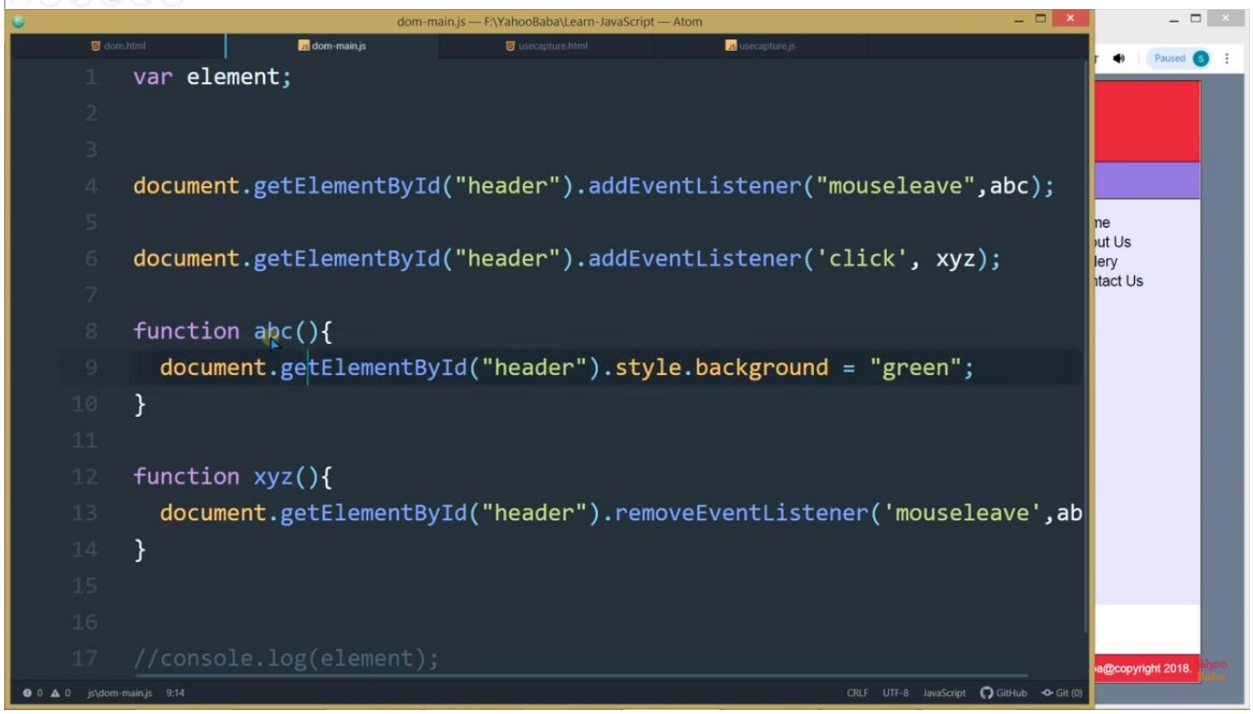

DOM removeEventListener() Method :
element.removeEventListener("ondblclick" ,functionName);


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

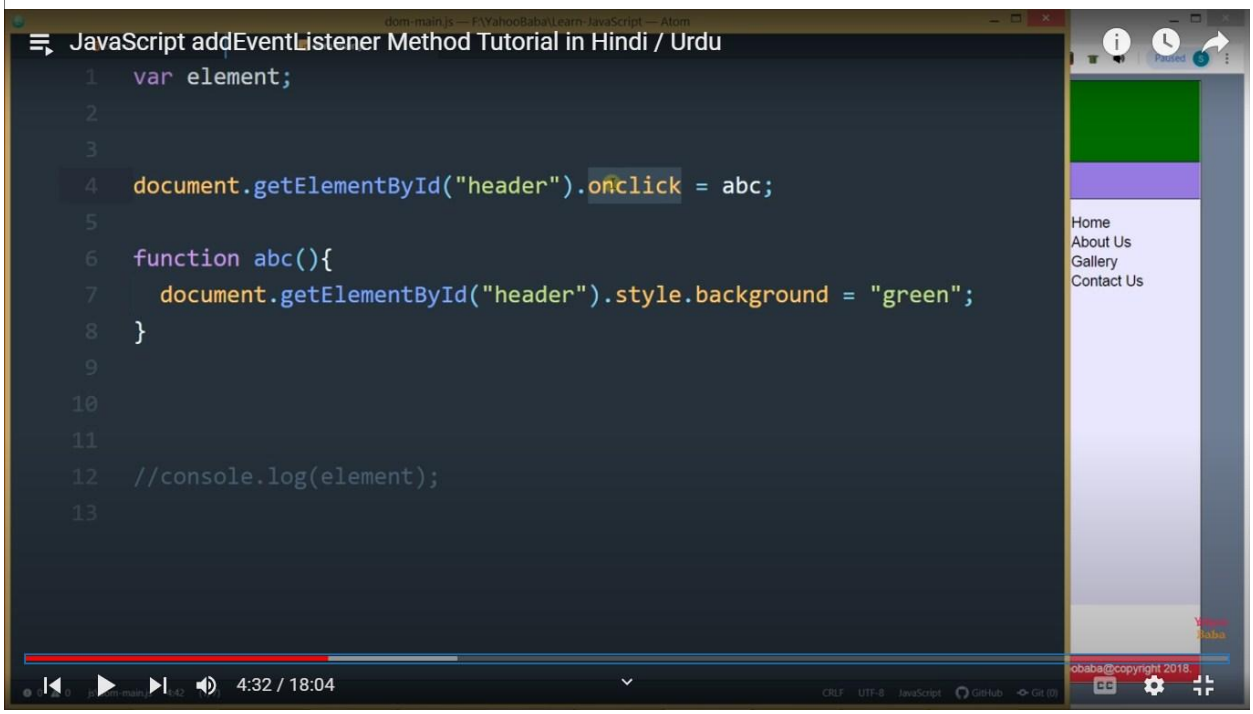
JavaScript classList Methods :

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



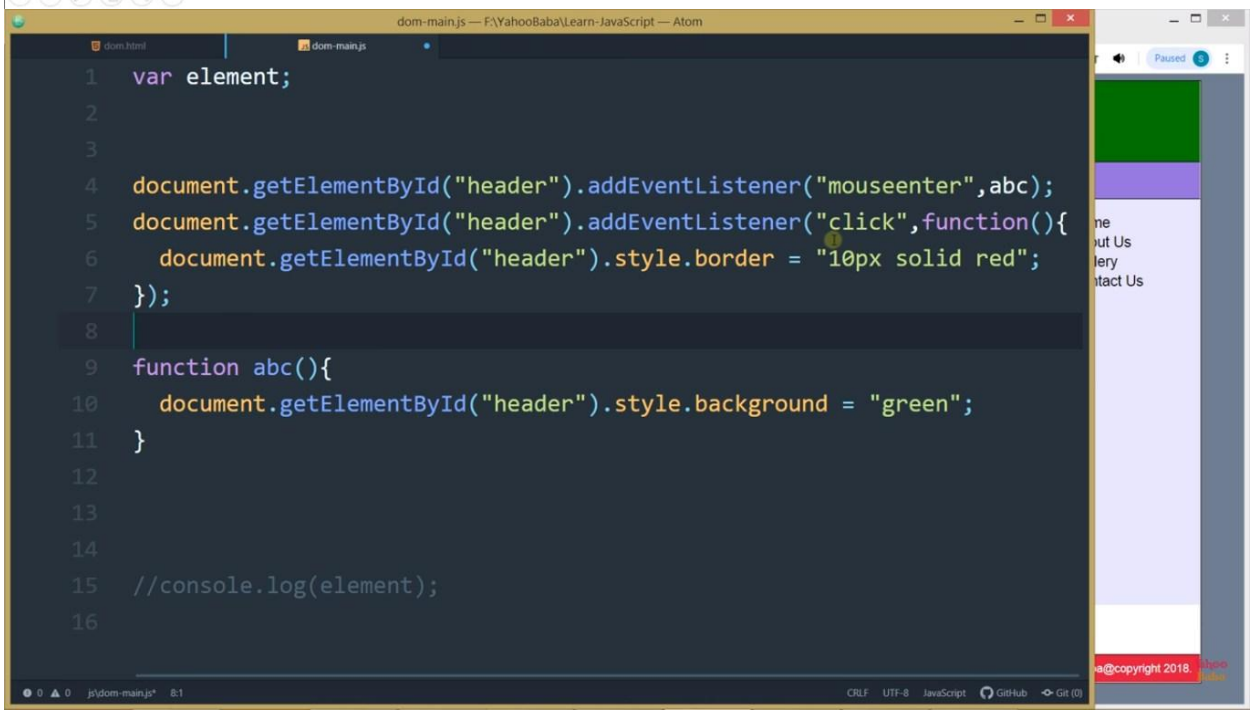
HTML Event Attributes :

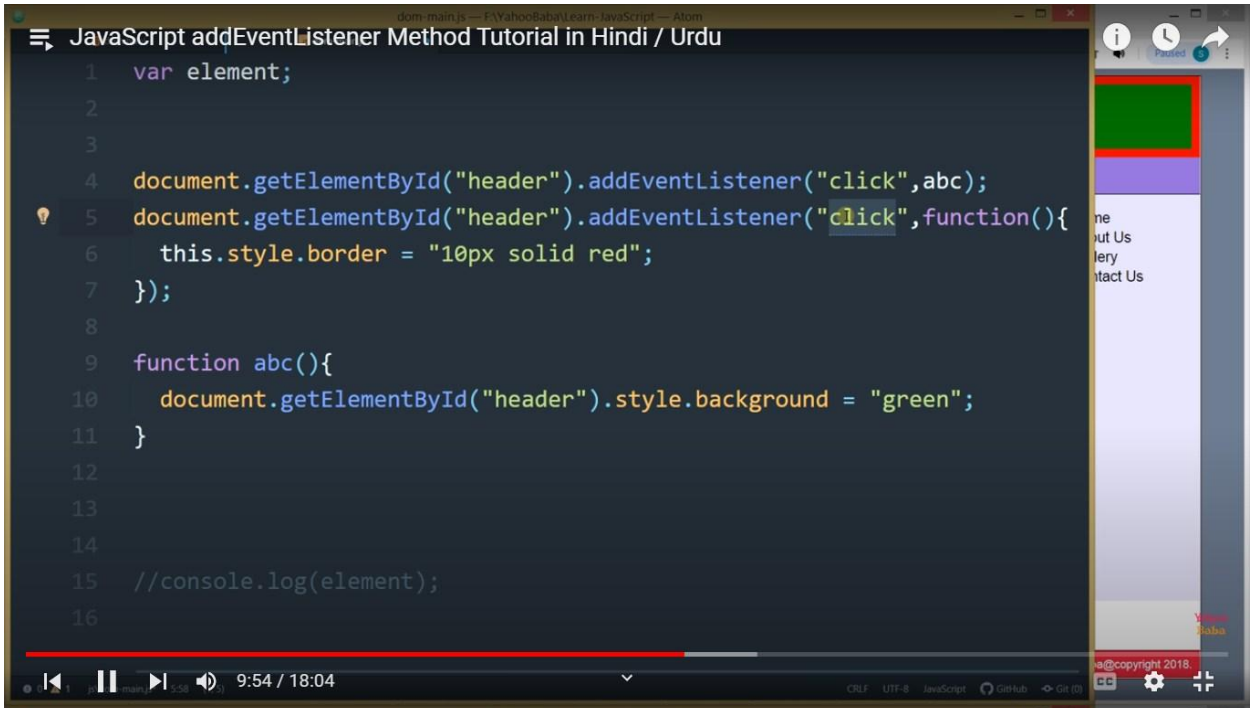
```
<button onClick="abc()"></button>

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

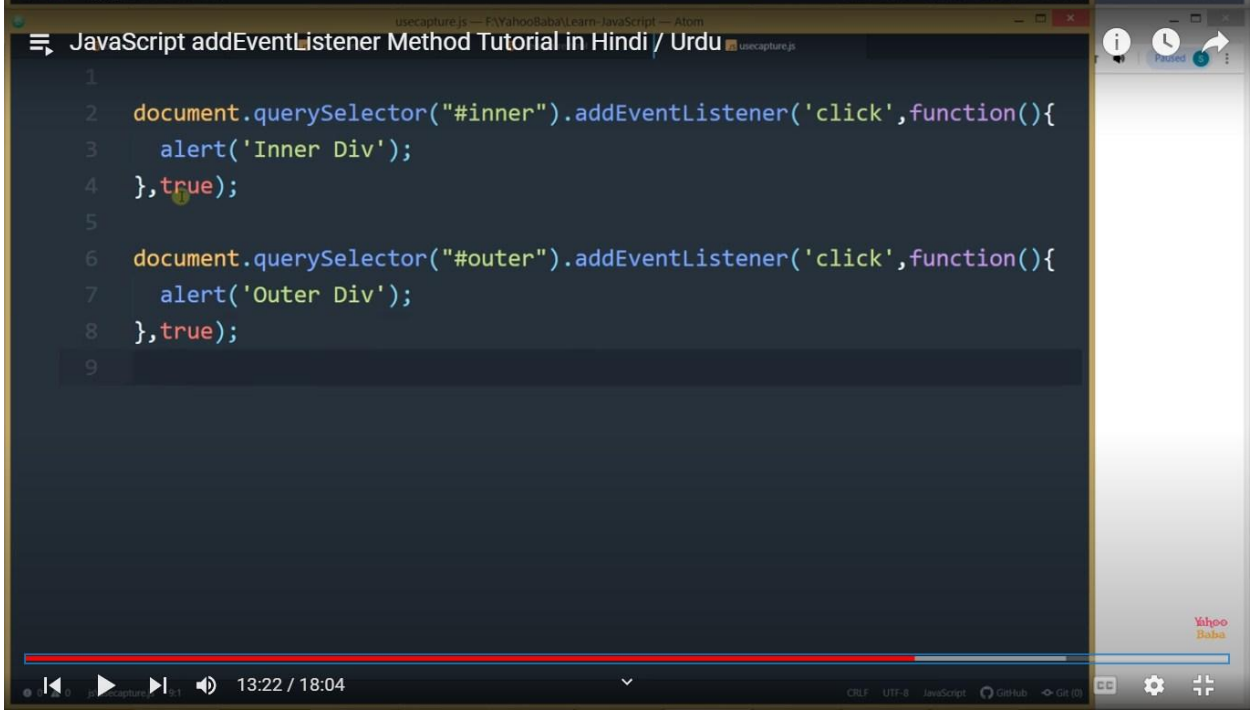


DOM addEventListener() Method :

```
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

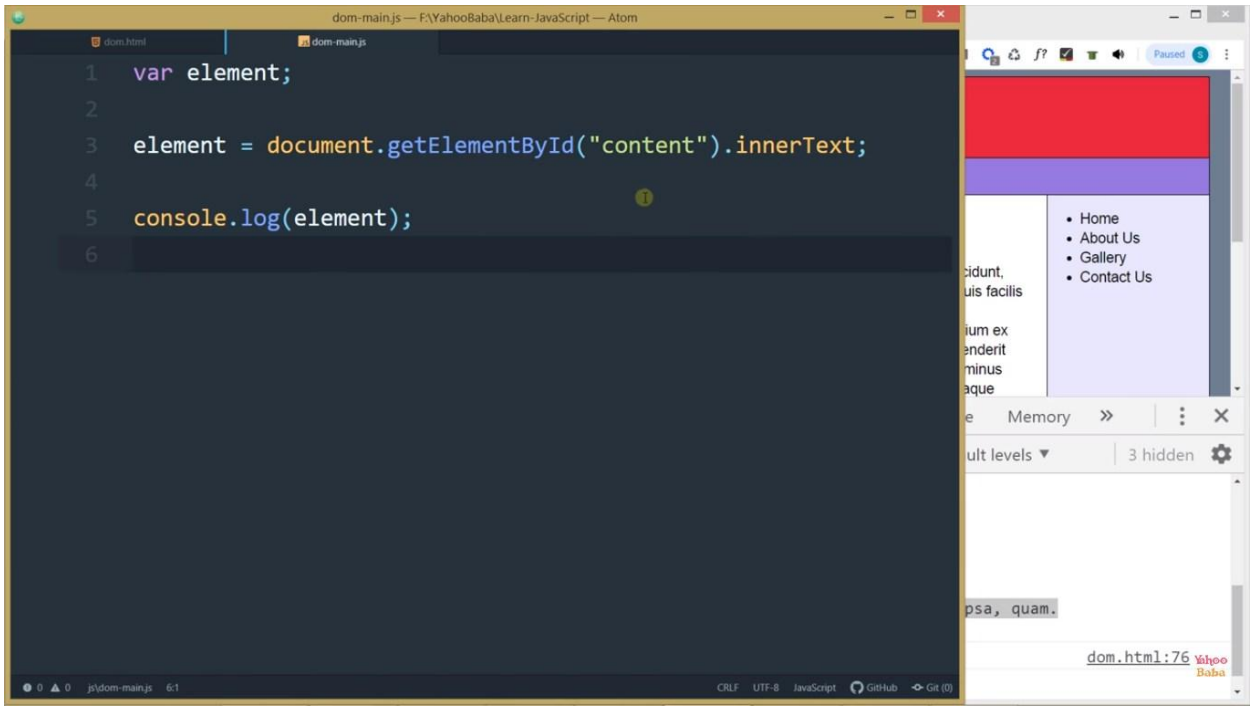
DOM removeEventListener() Method :

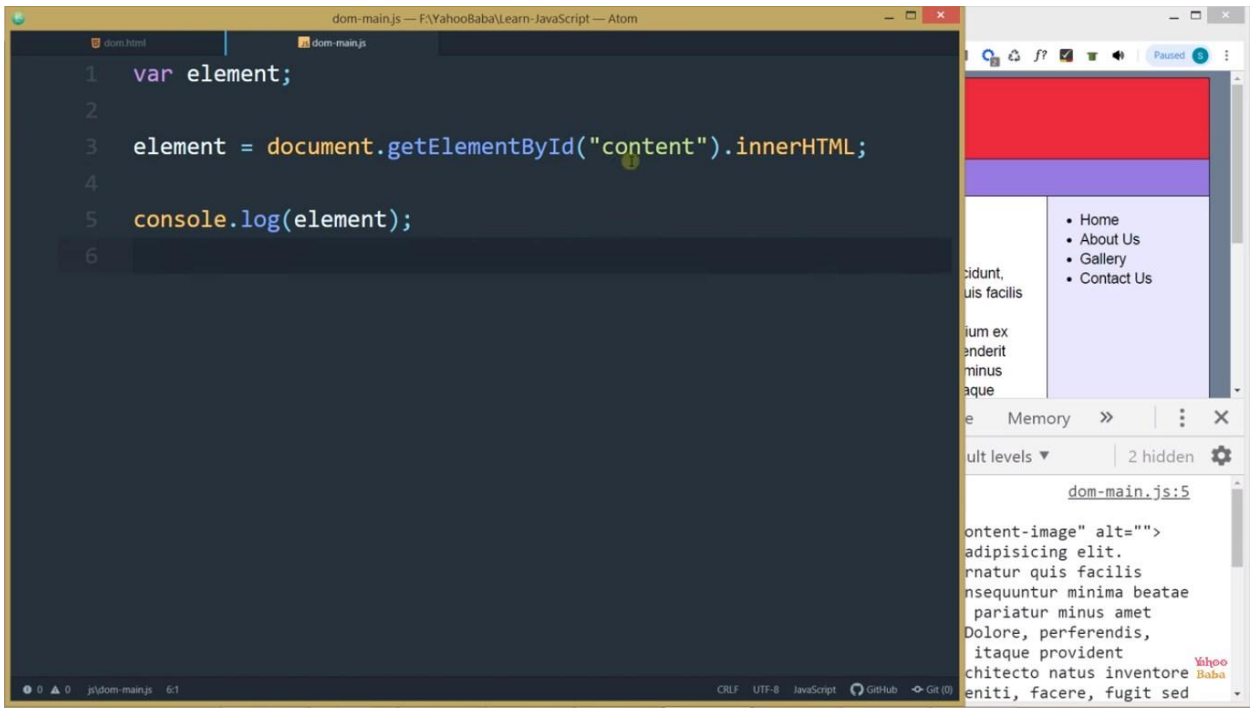
```
element.removeEventListener("ondblclick" ,functionName);
```

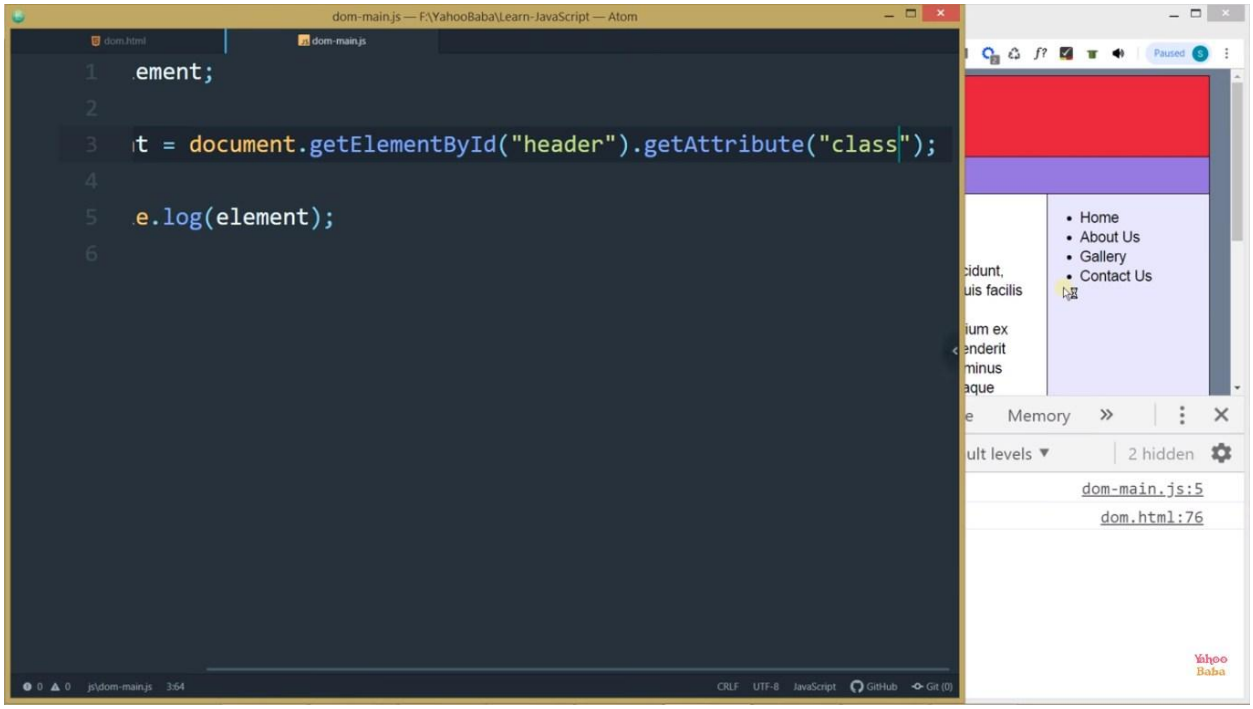
DOM Get Methods :

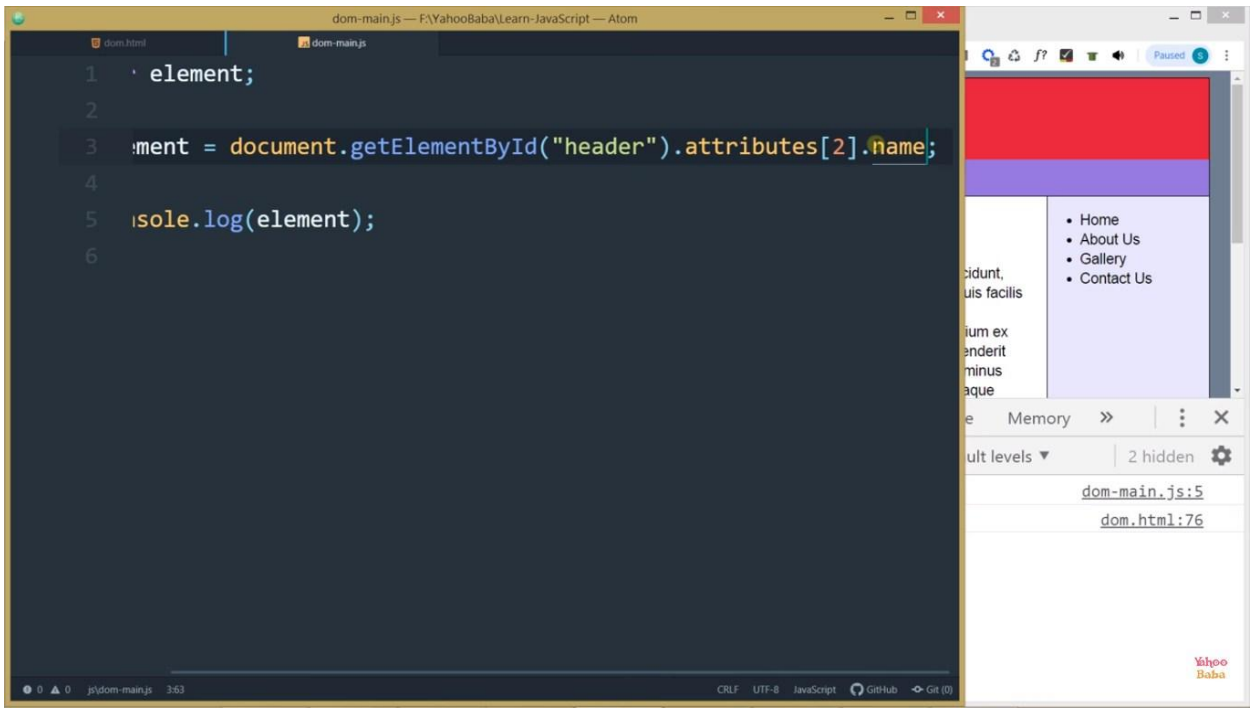
- innerText
- innerHTML
- getAttribute
- getAttributeNode
- Attributes

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

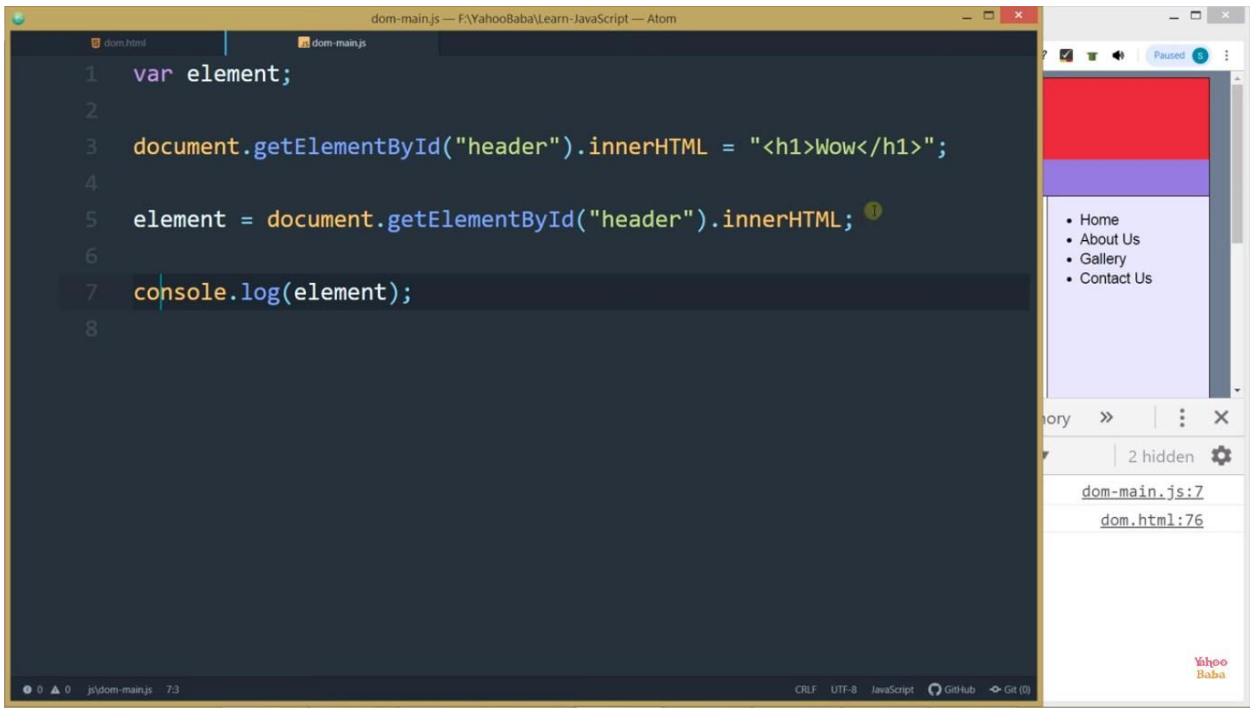

dom-main.js — F:\YahooBaba\Learn-JavaScript — Atom
dom.html
dom-main.js
ement;
t = document.getElementById("header").getAttribute("class");
e.log(element);
js\dom-main.js 3:64
CRLF UTF-8 JavaScript GitHub Git (0)
Home
About Us
Gallery
Contact Us
Memory
2 hidden
dom-main.js:5
dom.html:76

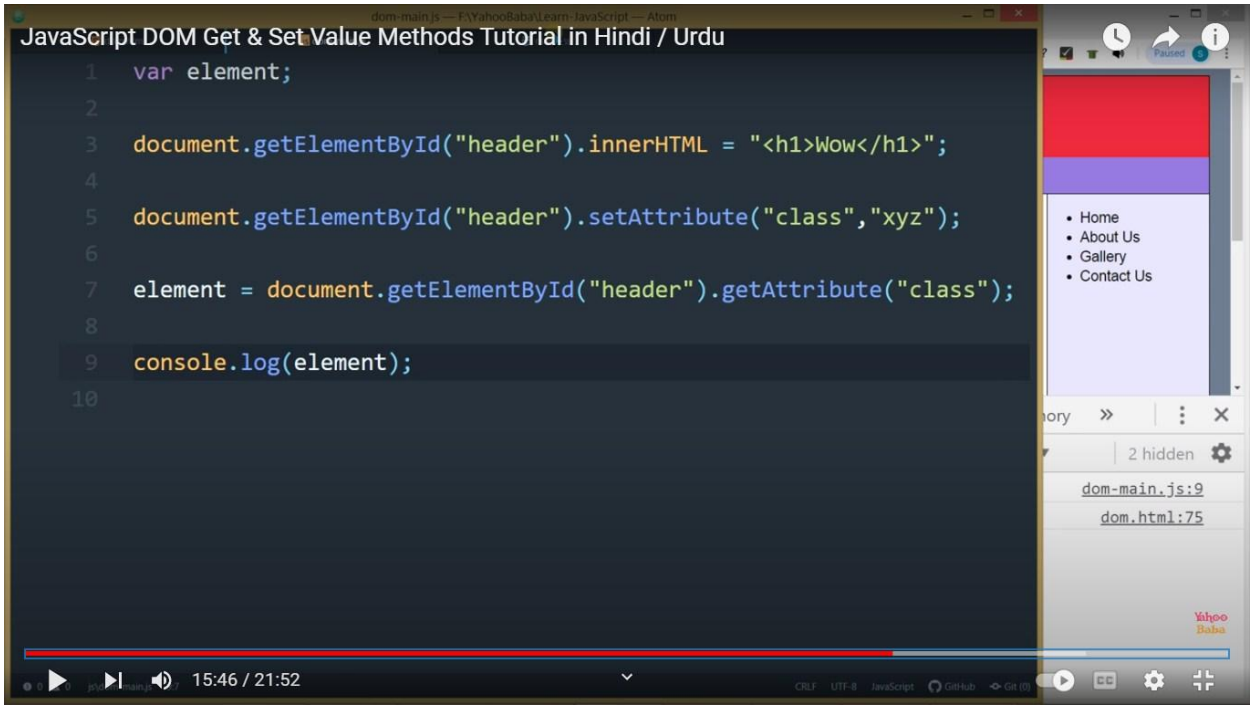


DOM Set Methods :

- innerText
- innerHTML
- setAttribute
- Attribute
- removeAttribute

Yahoo Baba

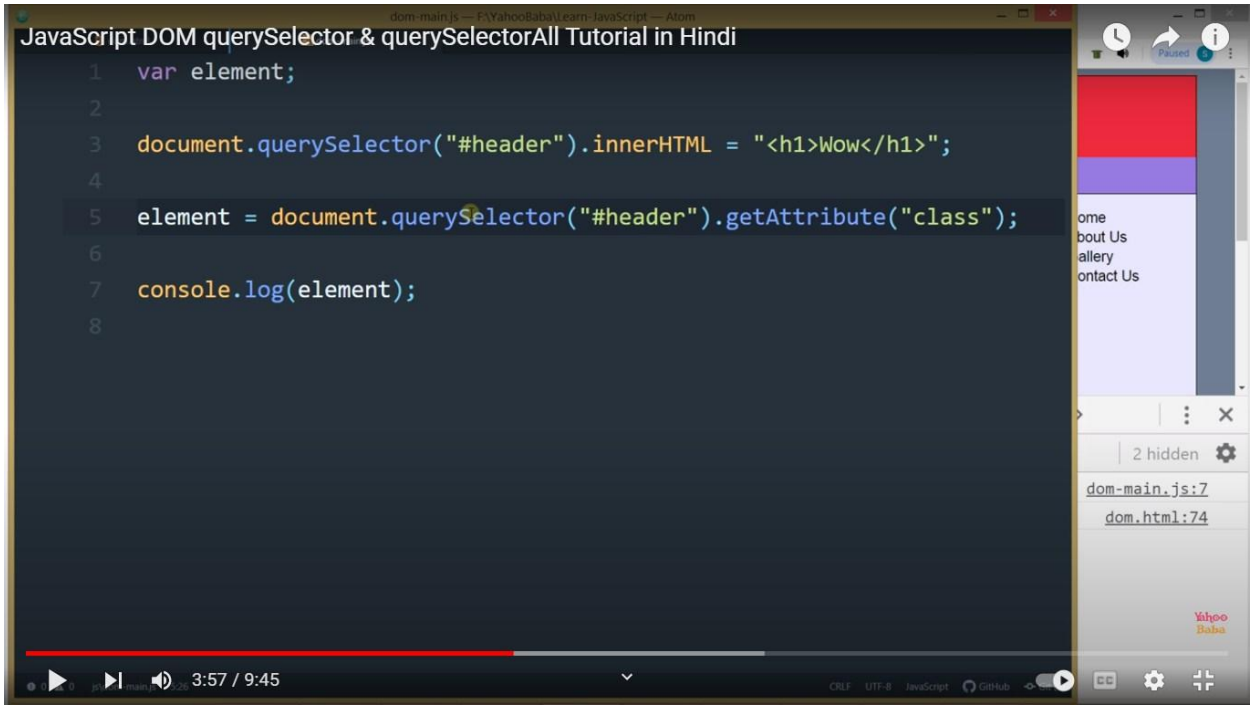

dom-main.js — F:\YahooBaba\Learn-JavaScript — Atom
dom.html
dom-main.js
1 var element;
2
3 document.getElementById("header").innerHTML = "<h1>Wow</h1>";
4
5 element = document.getElementById("header").innerHTML;
6
7 console.log(element);
8
Home
About Us
Gallery
Contact Us
2 hidden
dom-main.js:7
dom.html:76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

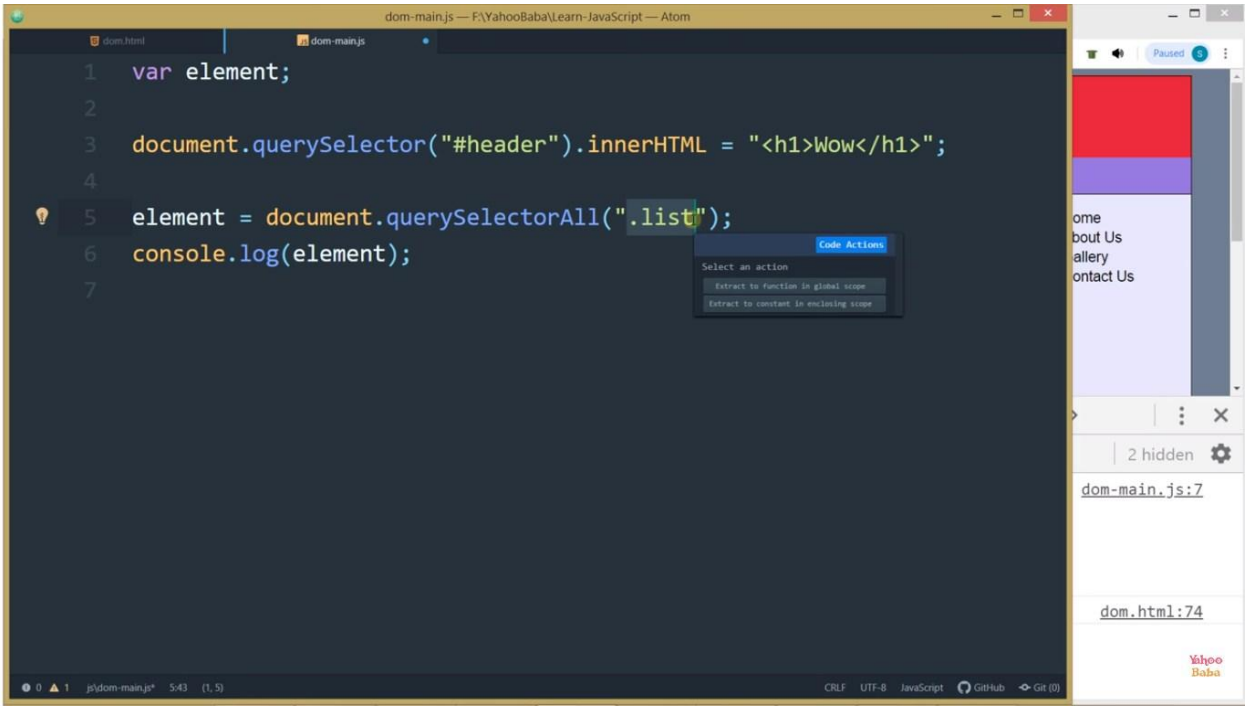

JavaScript DOM Get & Set Value Methods Tutorial in Hindi / Urdu
1 var element;
2
3 document.getElementById("header").innerHTML = "<h1>Wow</h1>";
4
5 document.getElementById("header").setAttribute("class","xyz");
6
7 element = document.getElementById("header").getAttribute("class");
8
9 console.log(element);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

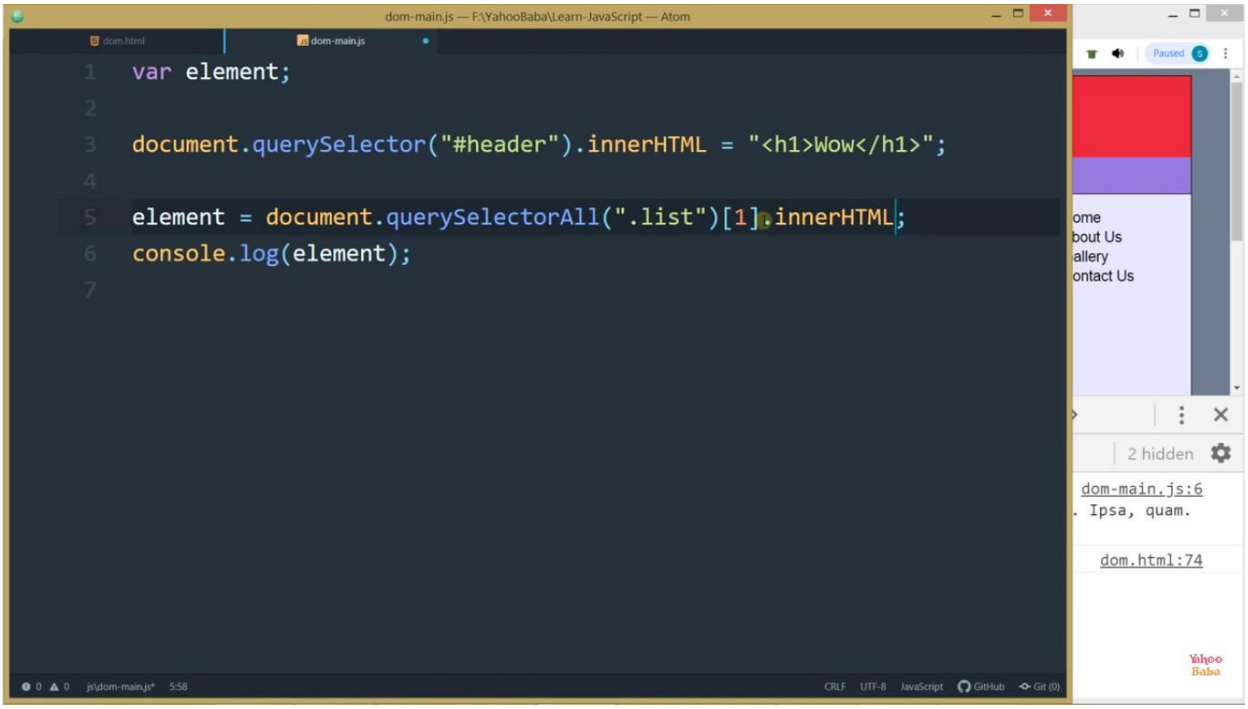
New DOM Targeting Methods :

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

Yahoo Baba



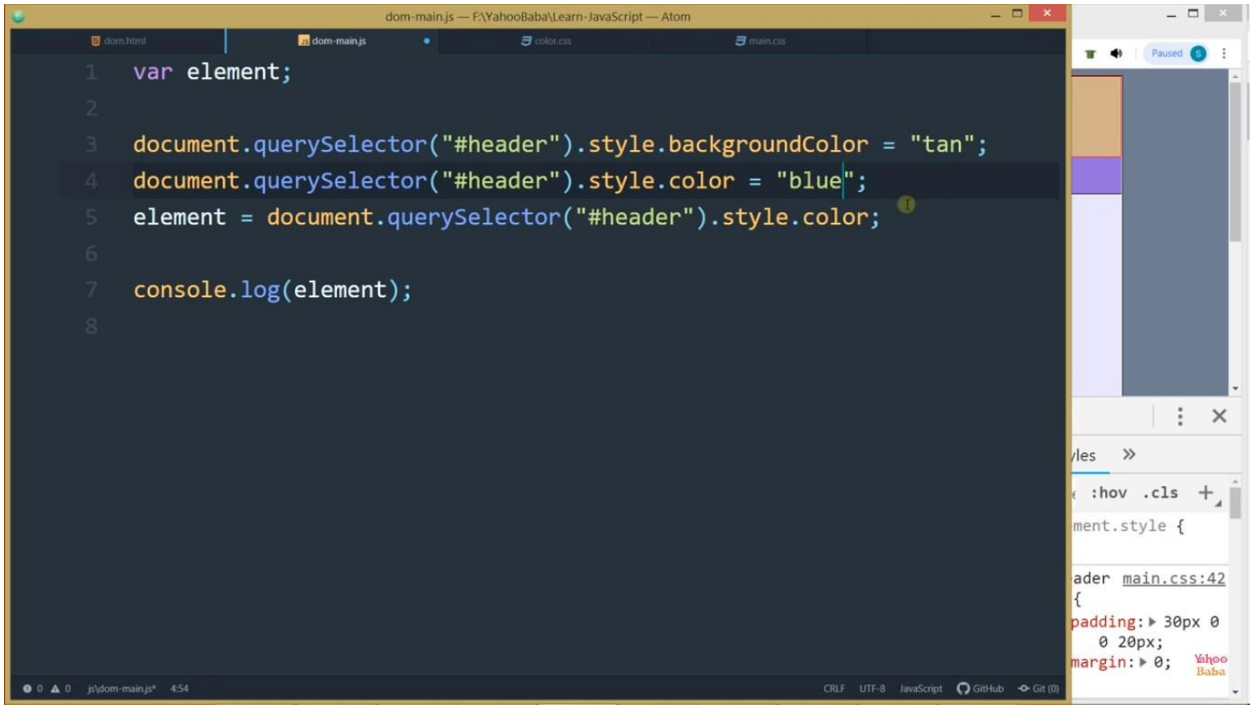

dom-main.js — F:\YahooBaba\Learn-JavaScript — Atom
dom.html
dom-main.js
var element;
document.querySelector("#header").innerHTML = "<h1>Wow</h1>";
element = document.querySelectorAll(".list");
console.log(element);
Code Actions
Select an action
Extract to function in global scope
Extract to constant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

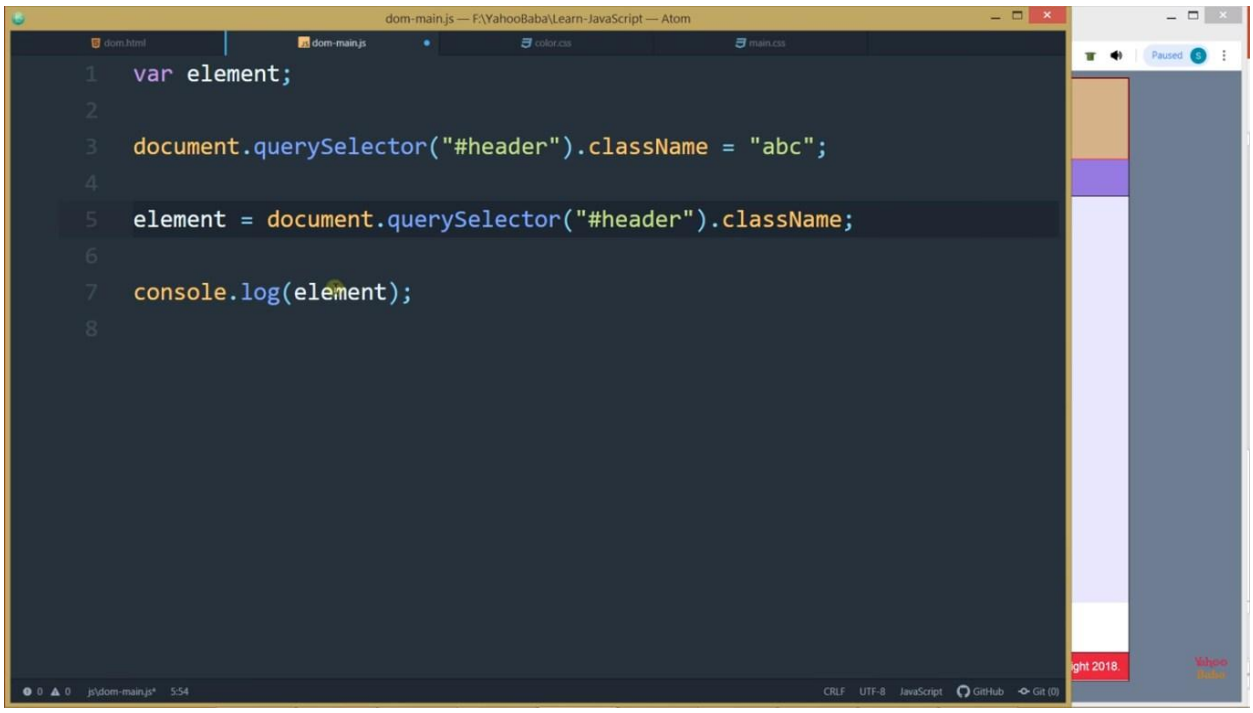

dom-main.js — F:\YahooBaba\Learn-JavaScript — Atom
dom.html
dom-main.js
var element;
document.querySelector("#header").innerHTML = "<h1>Wow</h1>";
element = document.querySelectorAll(".list")[1].innerHTML;
console.log(element);
ome
bout Us
allery
ontact Us
2 hidden
dom-main.js:6
. Ipsa, quam.
dom.html:74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

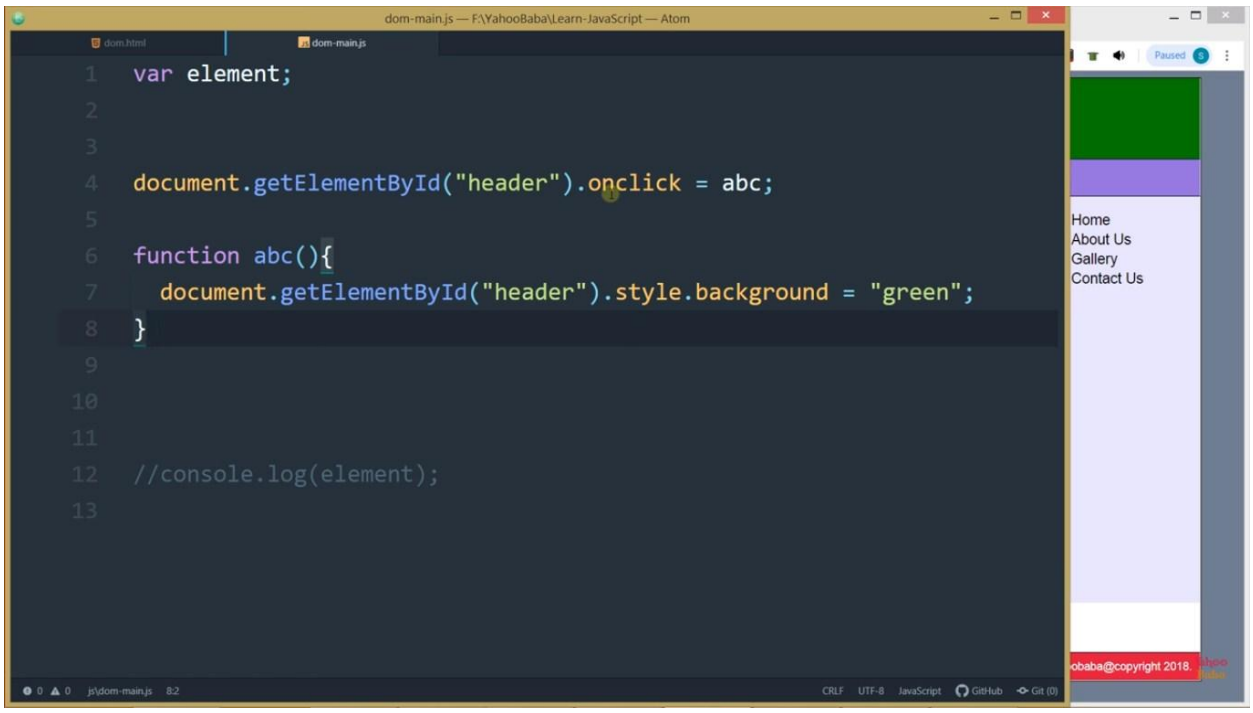
Yahoo Baba

HTML Event Attributes :

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

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


Assign Events Using the HTML DOM :

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



DOM addEventListener() Method :

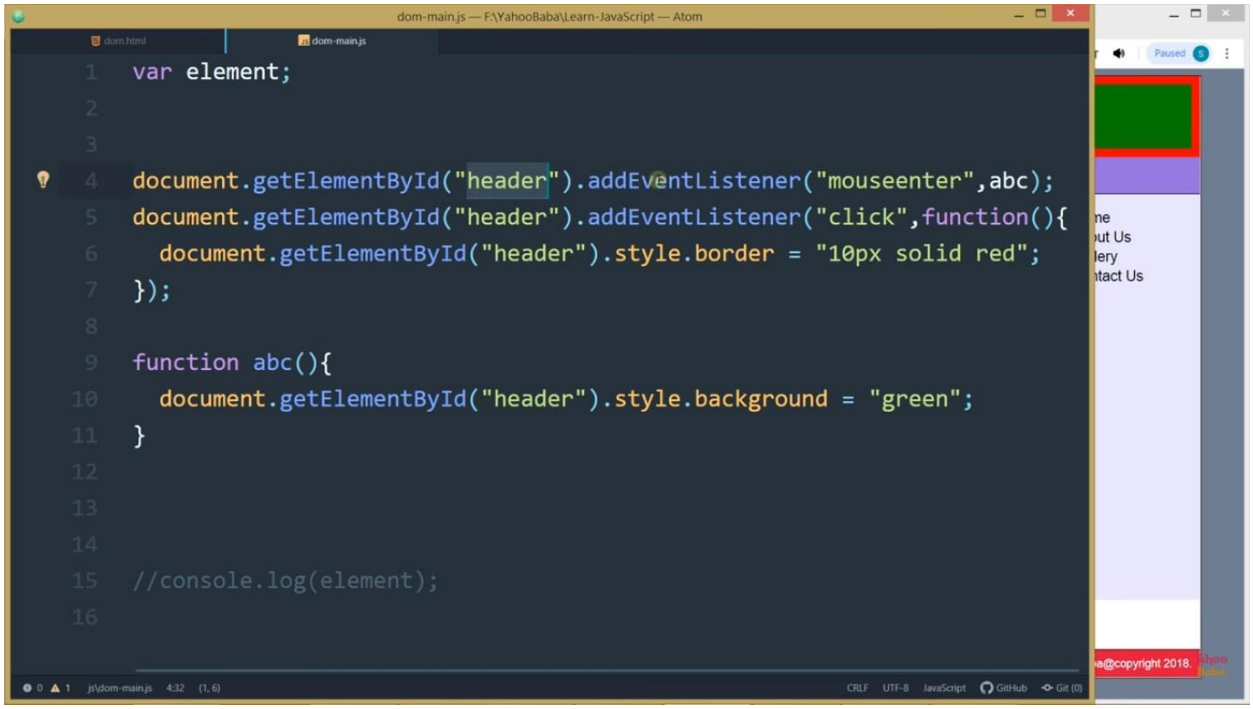
```
document.getElementById(Id). addEventListener("click", functionName);
```

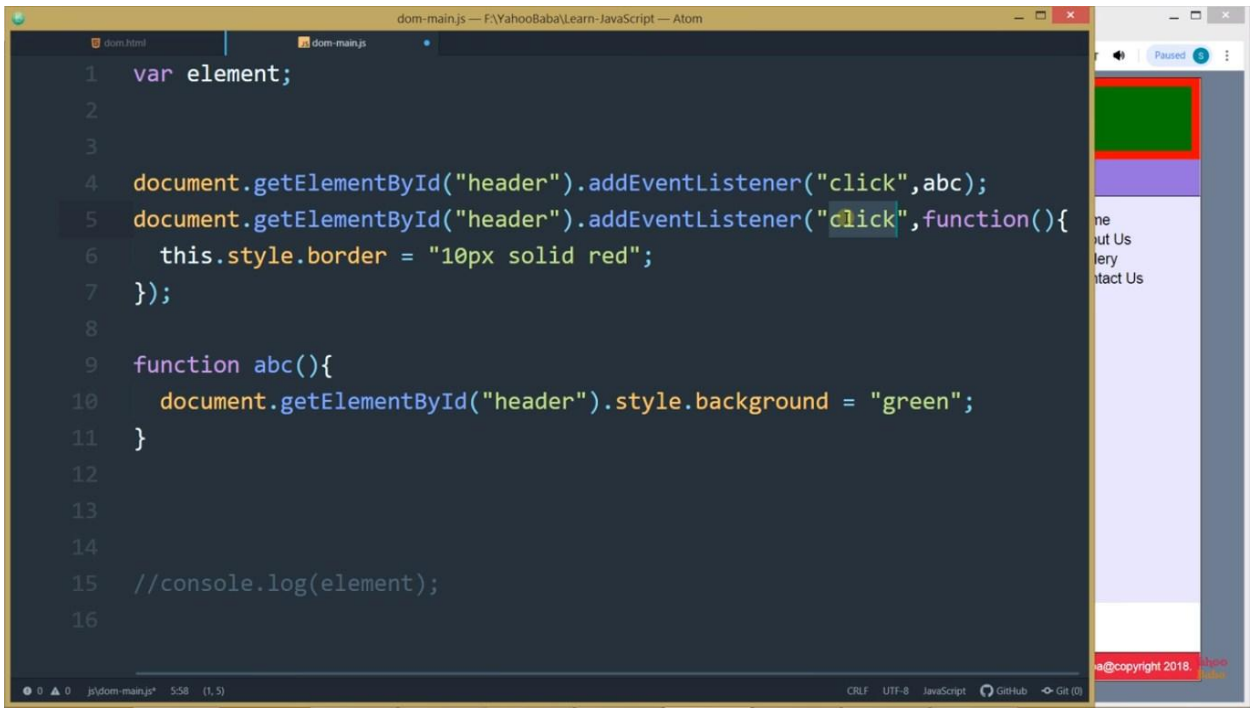
Yahoo Baba

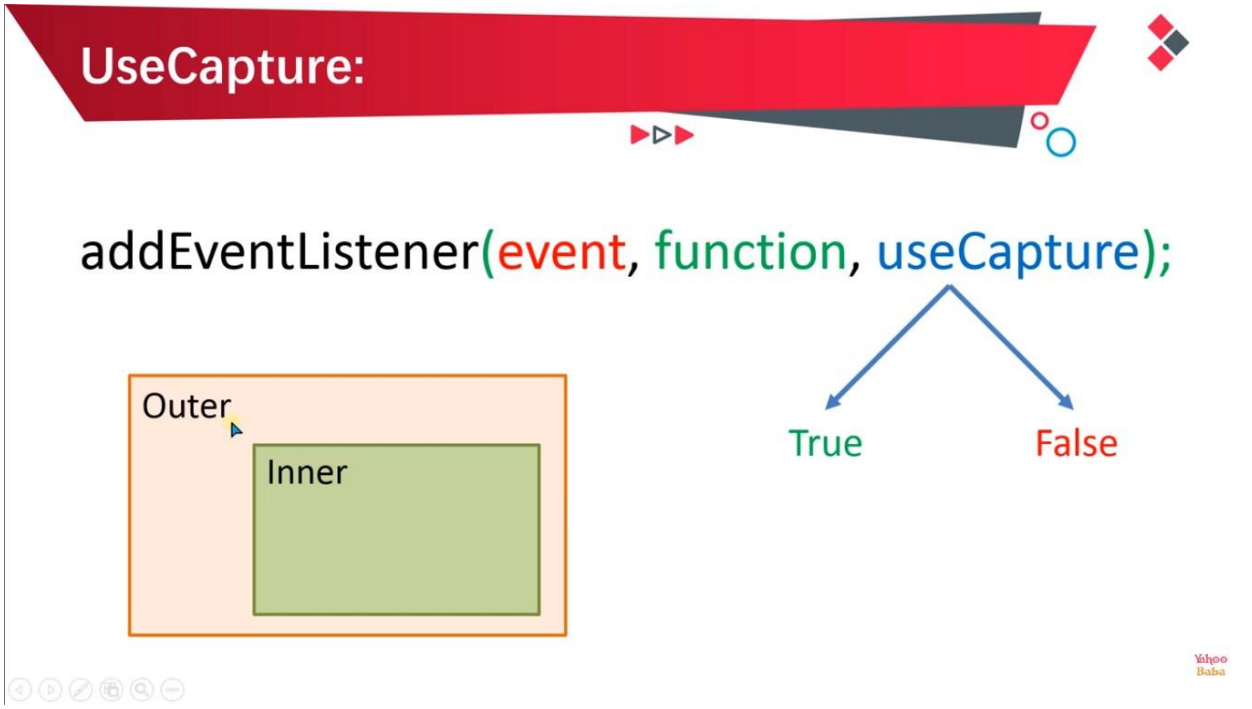
DOM addEventListener() Method :

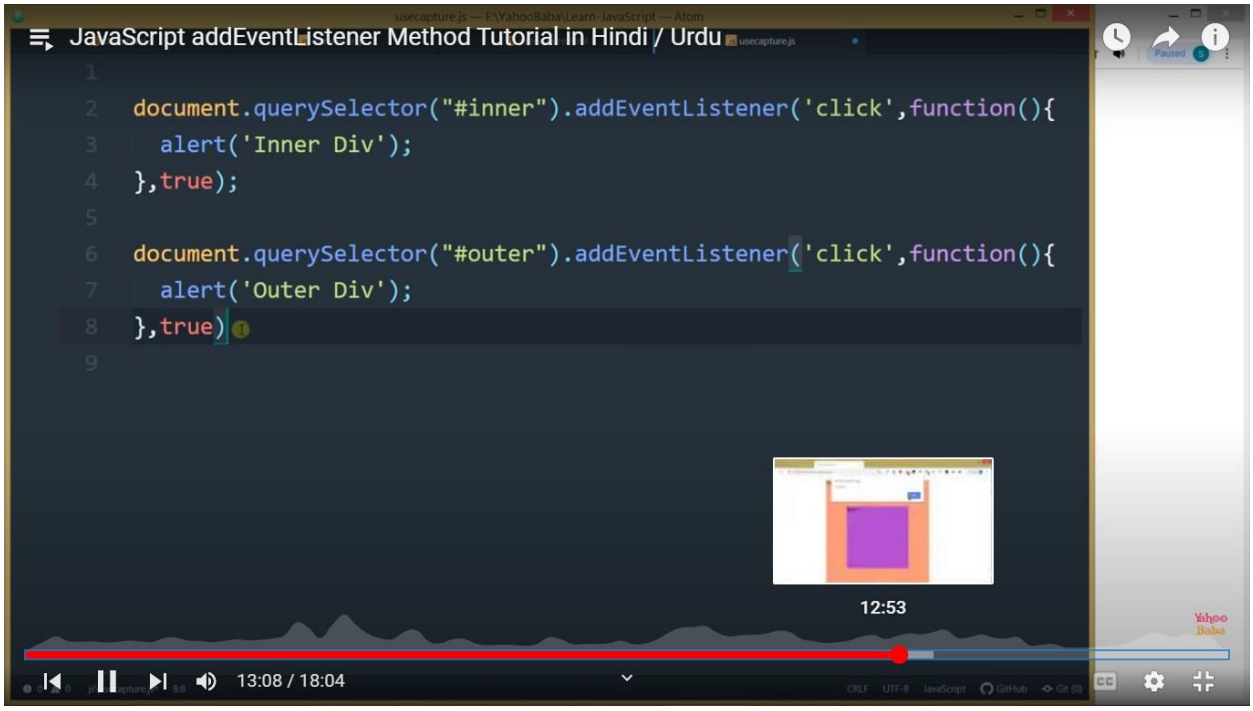
```
document.getElementById(Id). addEventListener("click", functionName);

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

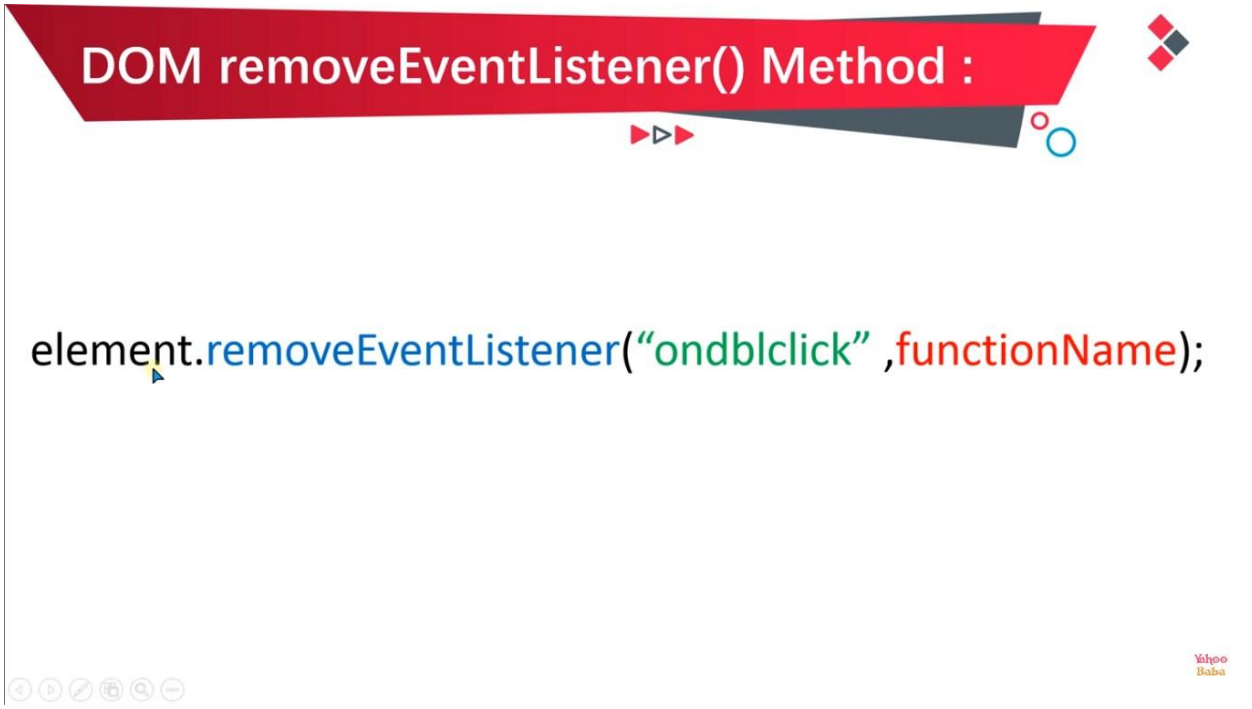
Yahoo Baba

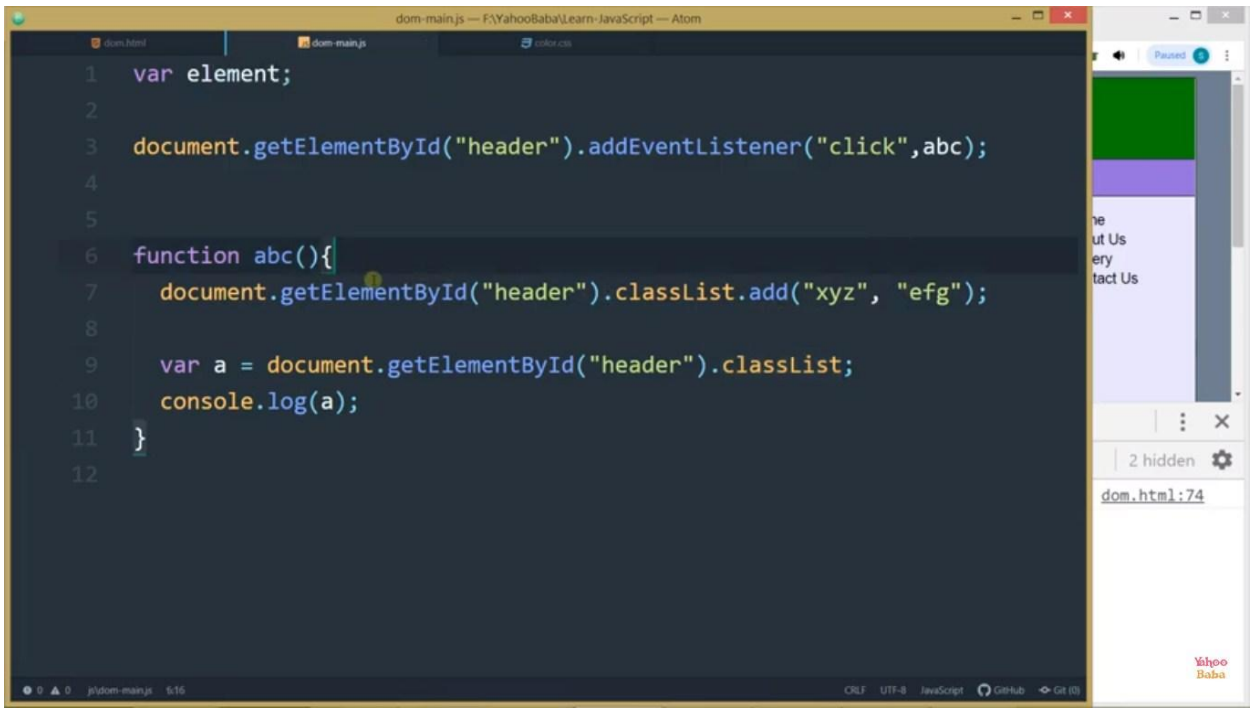


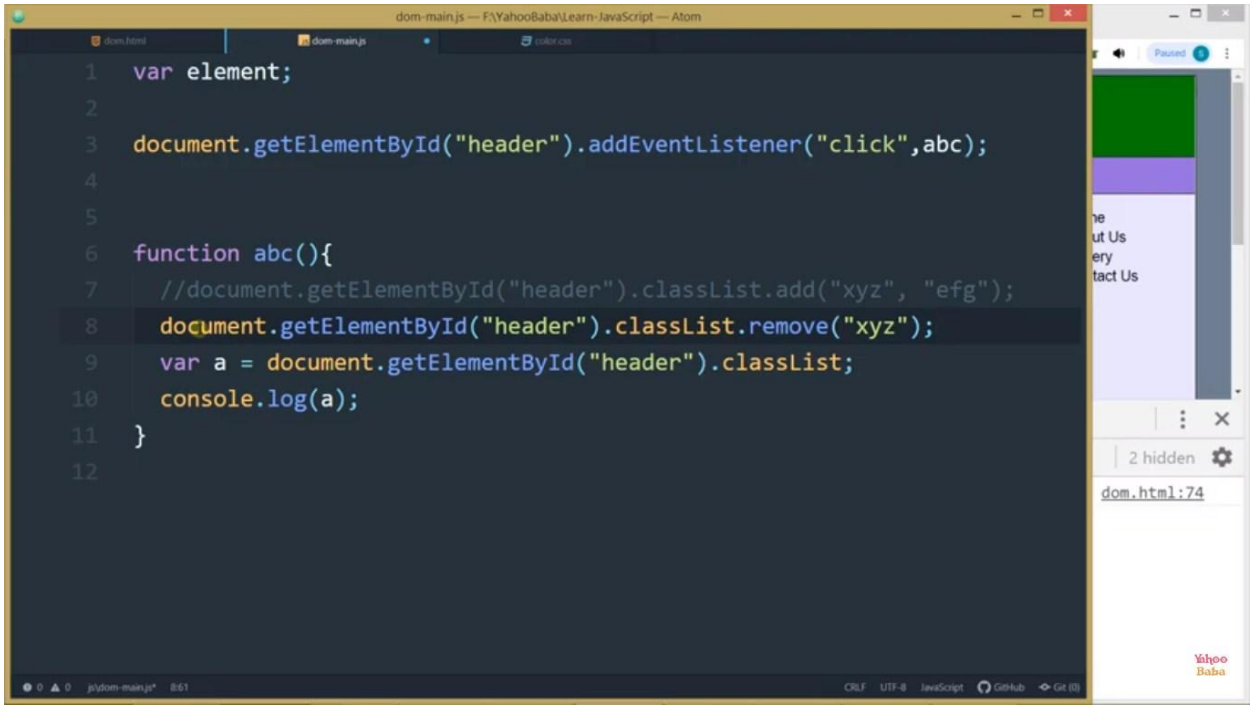

dom-main.js — F:\YahooBaba\Learn-JavaScript — Atom
dom.html
dom-main.js
1 var element;
2
3
4 document.getElementById("header").addEventListener("click",abc);
5 document.getElementById("header").addEventListener("click",function(){
6 this.style.border = "10px solid red";
7 });
8
9 function abc(){
10 document.getElementById("header").style.background = "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

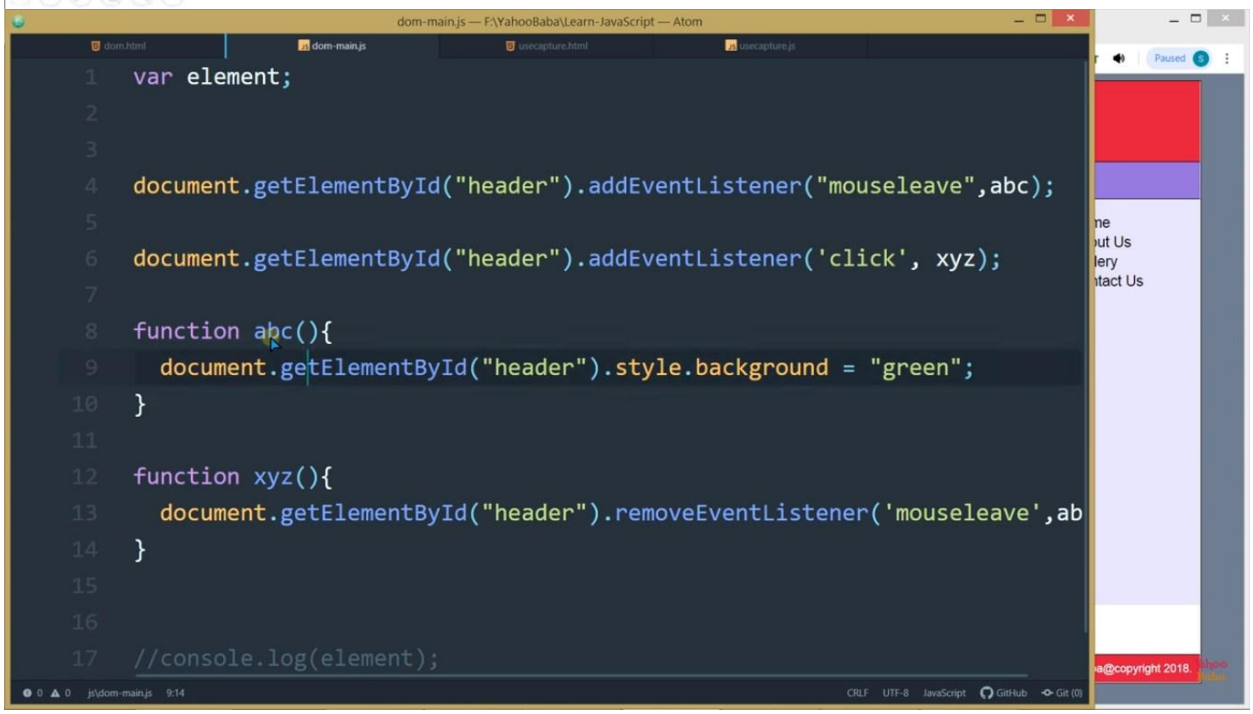

DOM removeEventListener() Method :
element.removeEventListener("ondblclick" ,functionName);


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

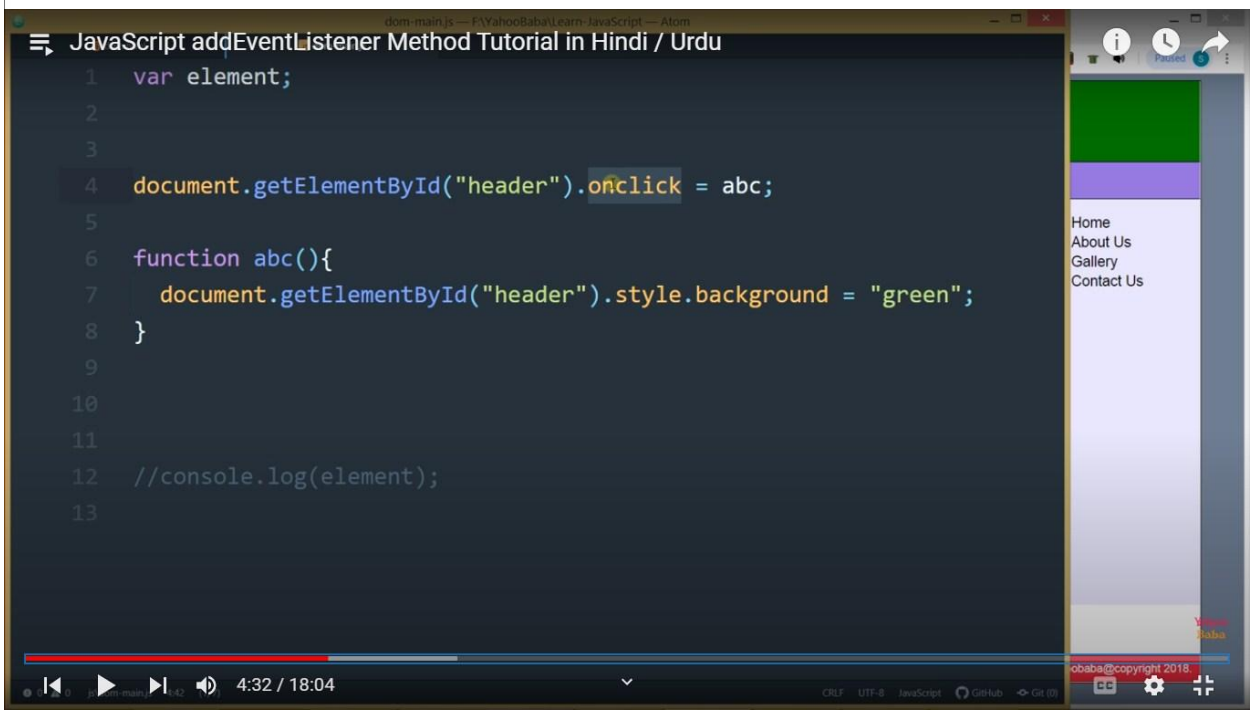
JavaScript classList Methods :

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



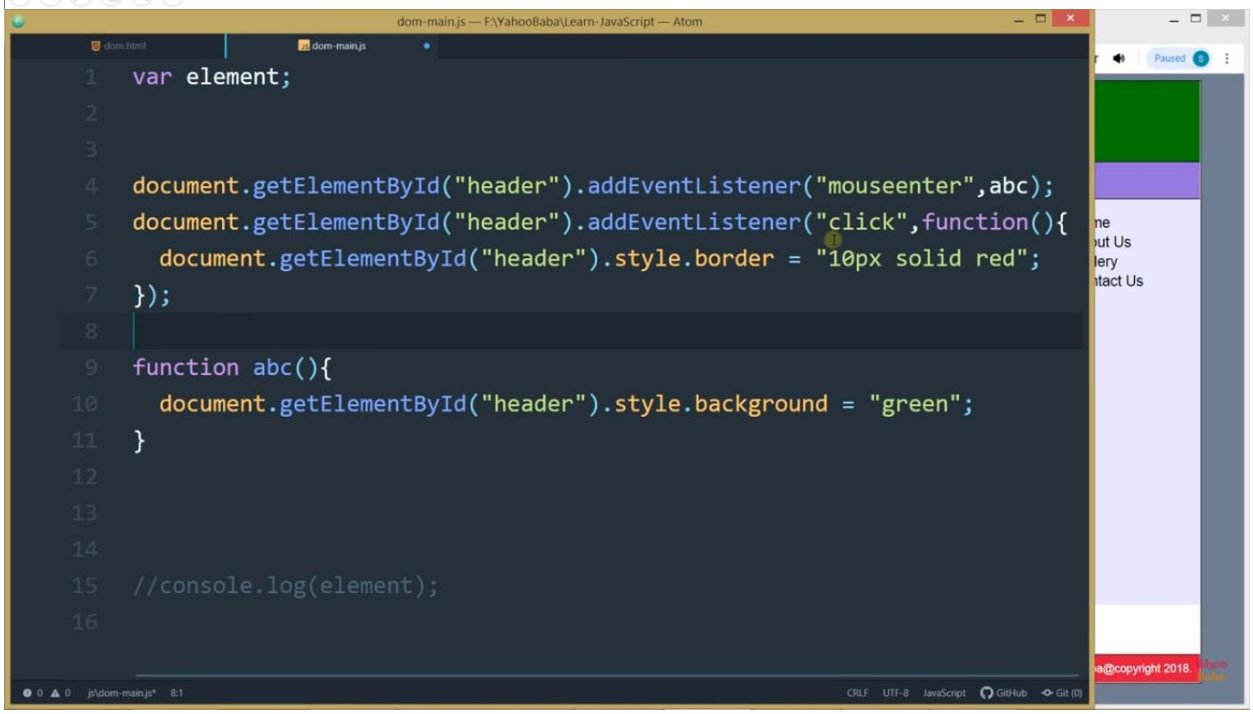
HTML Event Attributes :

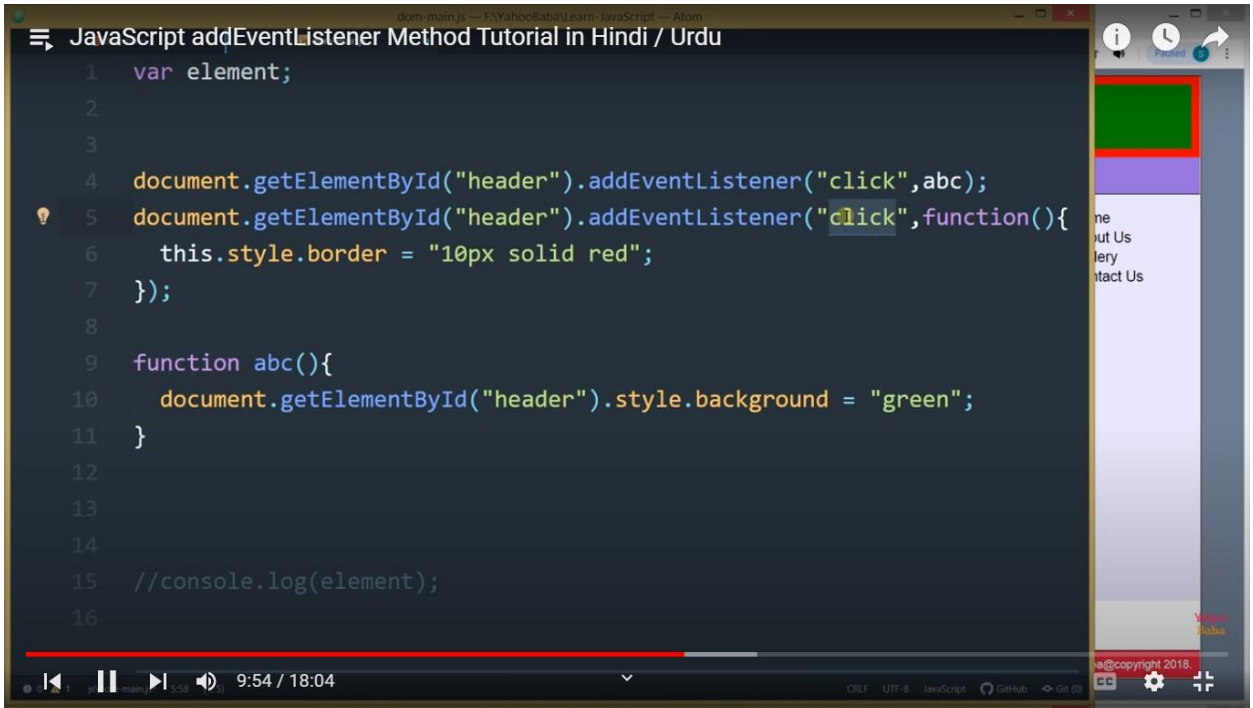
```
<button onClick="abc()"></button>

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

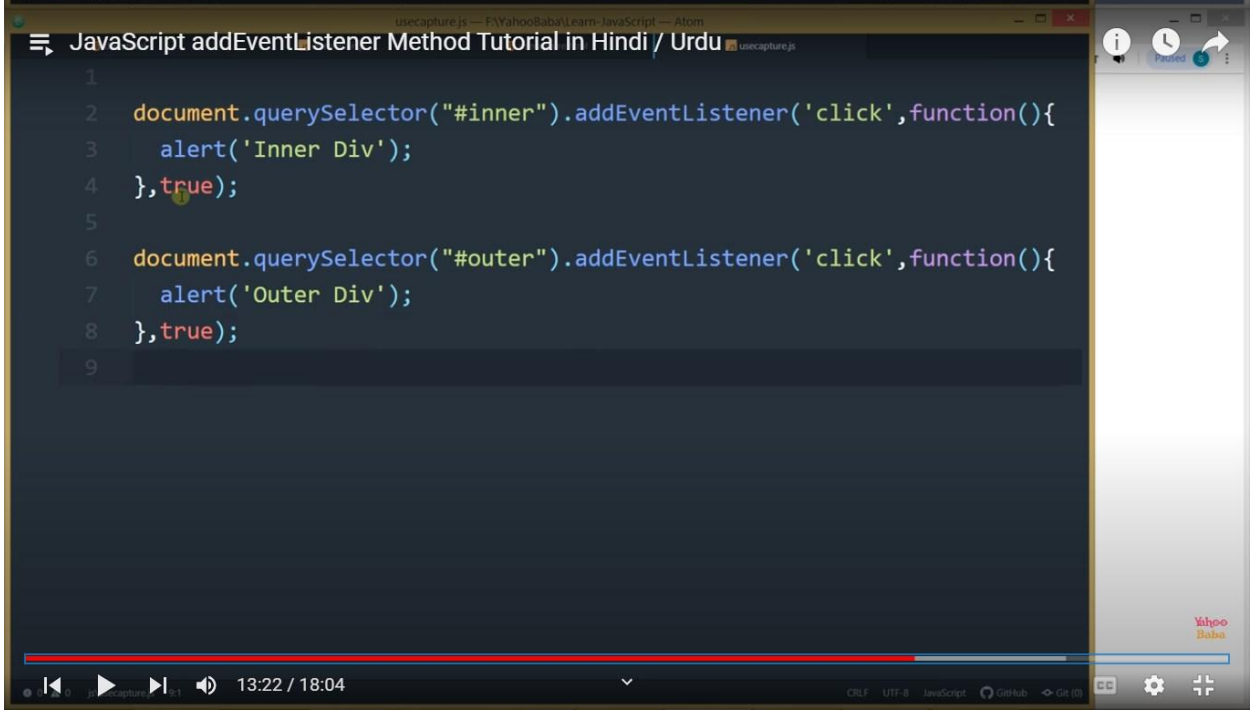


DOM addEventListener() Method :

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

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

});
```

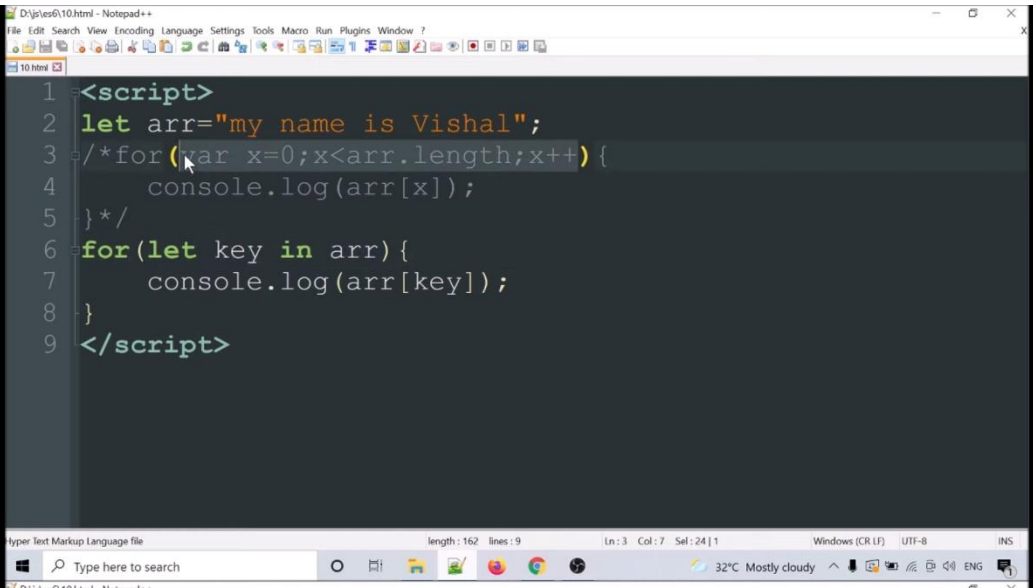


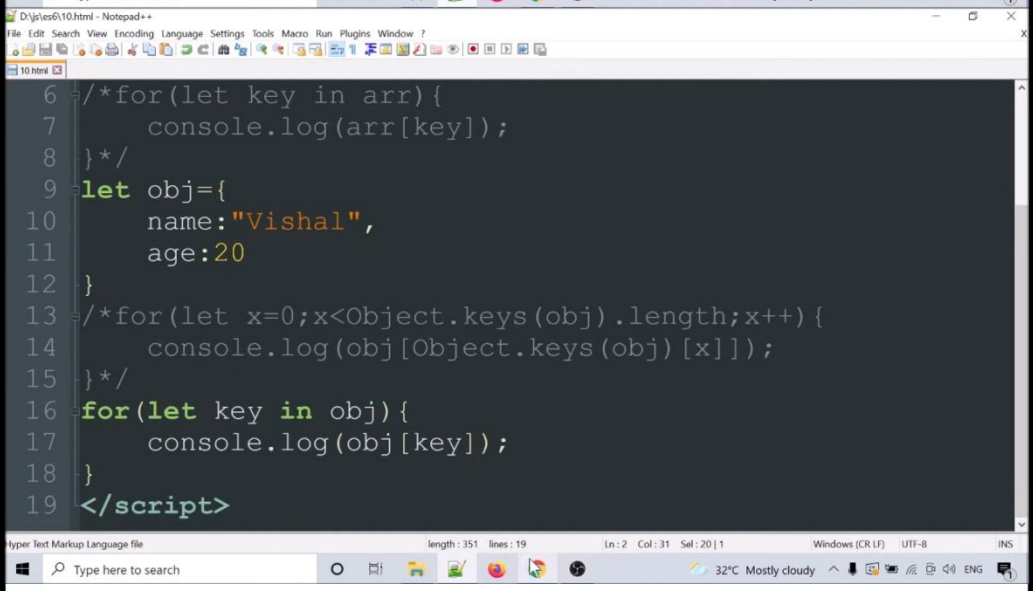

JavaScript addEventListener Method Tutorial in Hindi / Urdu
var element;
document.getElementById("header").addEventListener("click",abc);
document.getElementById("header").addEventListener("click",function(){
this.style.border = "10px solid red";
});
function abc(){
document.getElementById("header").style.background = "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);
```





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: -

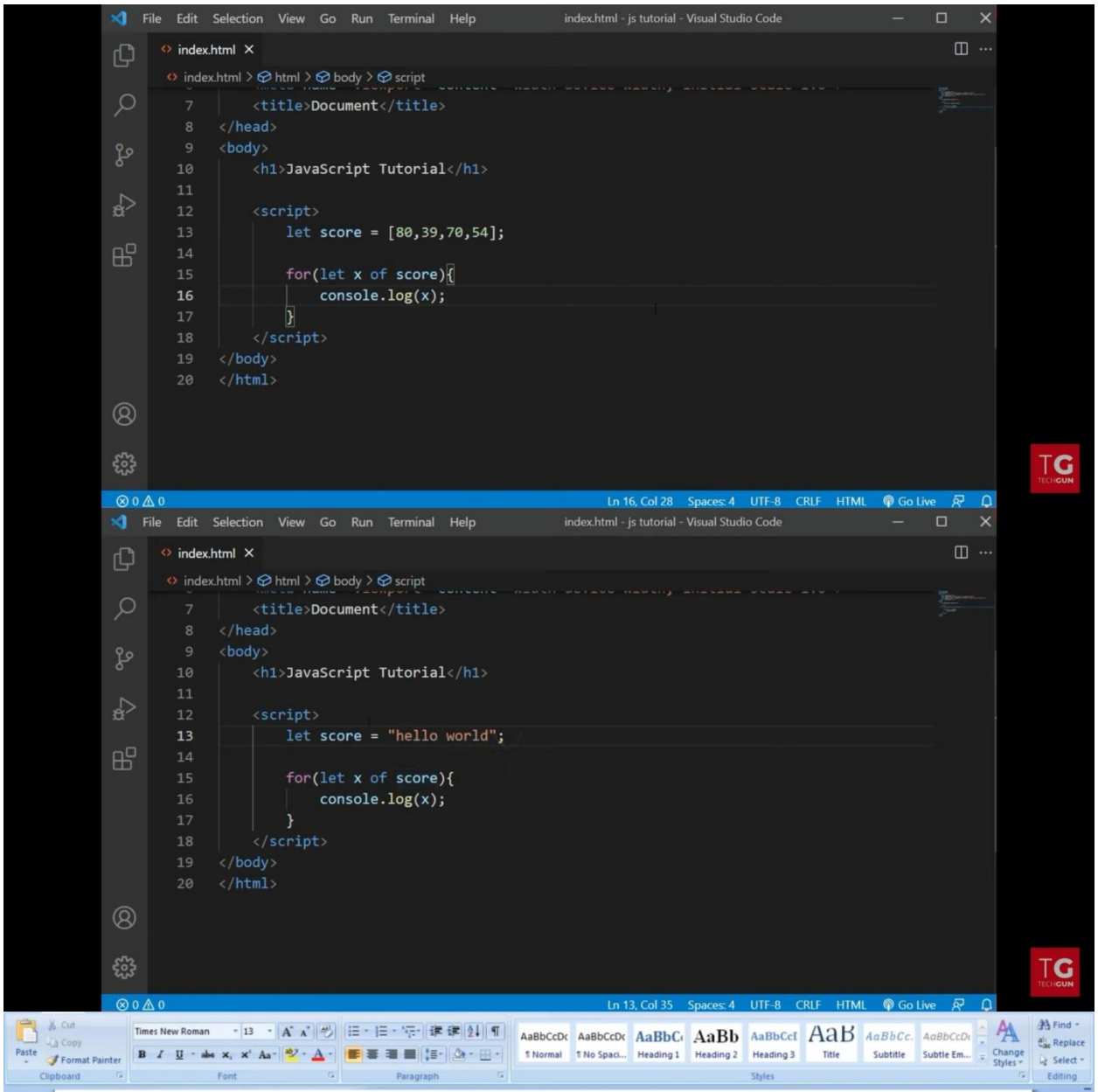
Inheritance allows a class (subclass) to acquire the properties and behavior of another class (super-class). It helps to reuse, customize and enhance the existing code. So it helps to write a code accurately and reduce the development time.

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++){
```

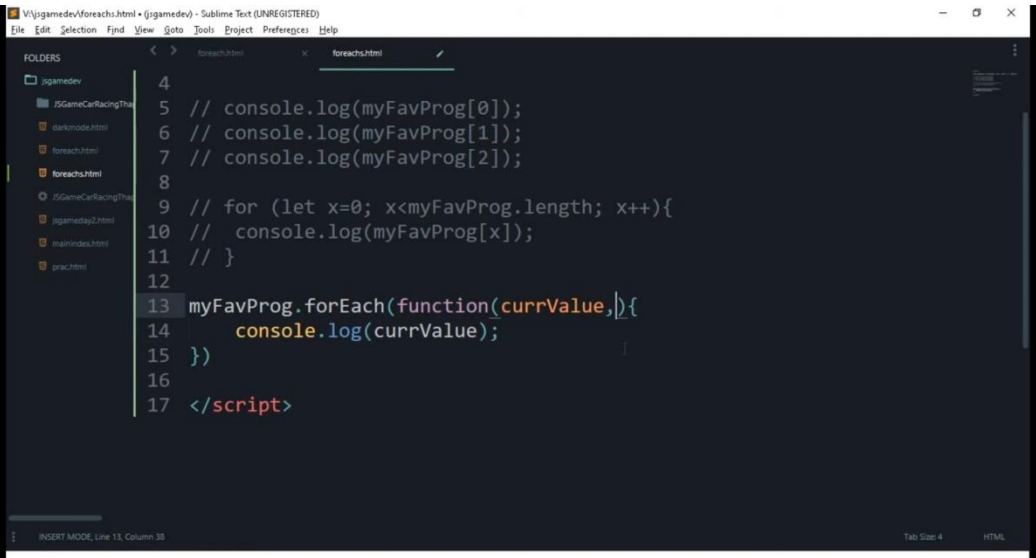
```
    <title>Document</title>
</head>
<body>
    <h1>JavaScript Tutorial</h1>

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

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

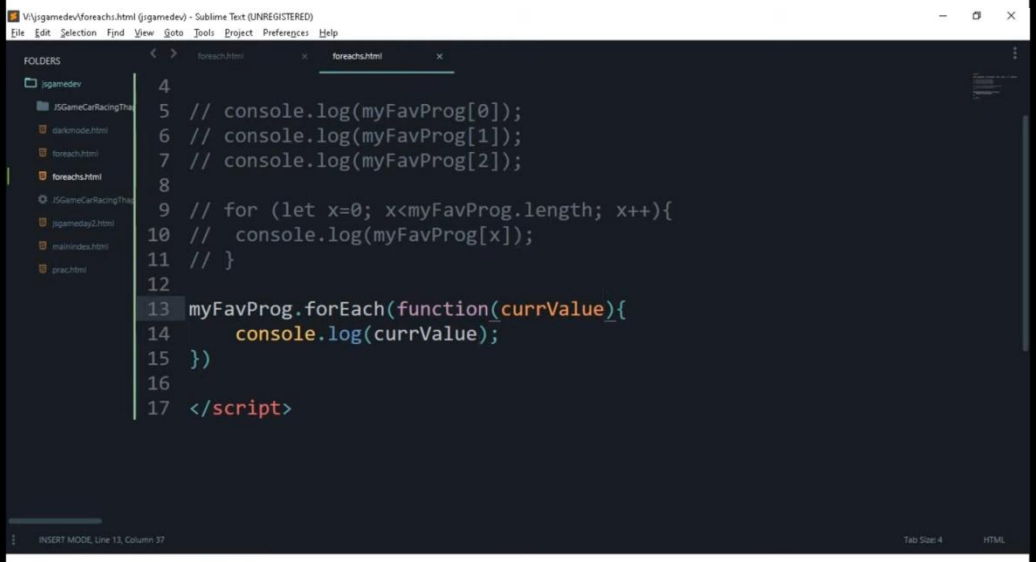
The forEach() method calls a function once for each element in an array, in order.

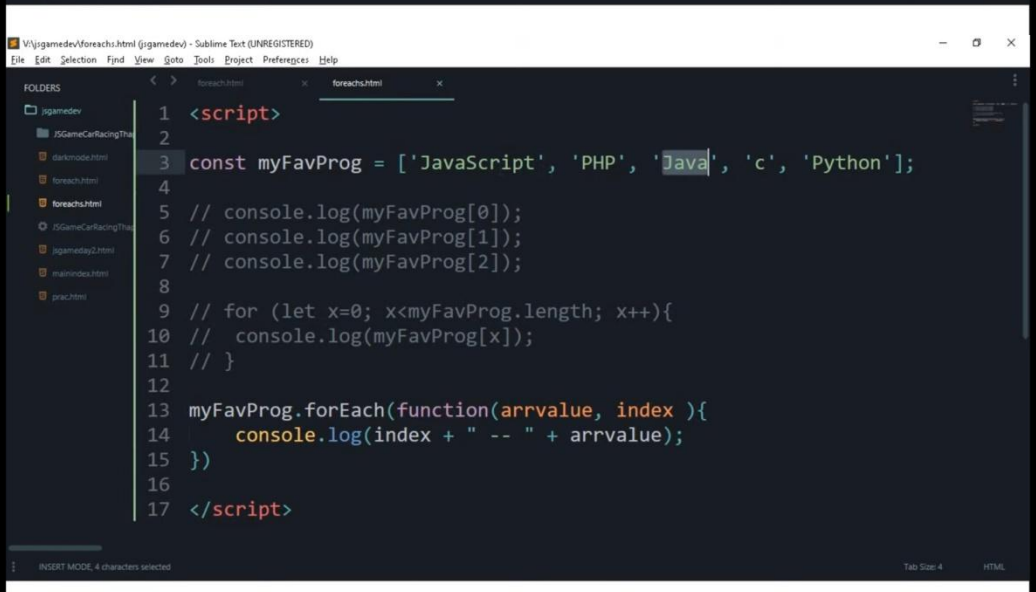
forEach() Method

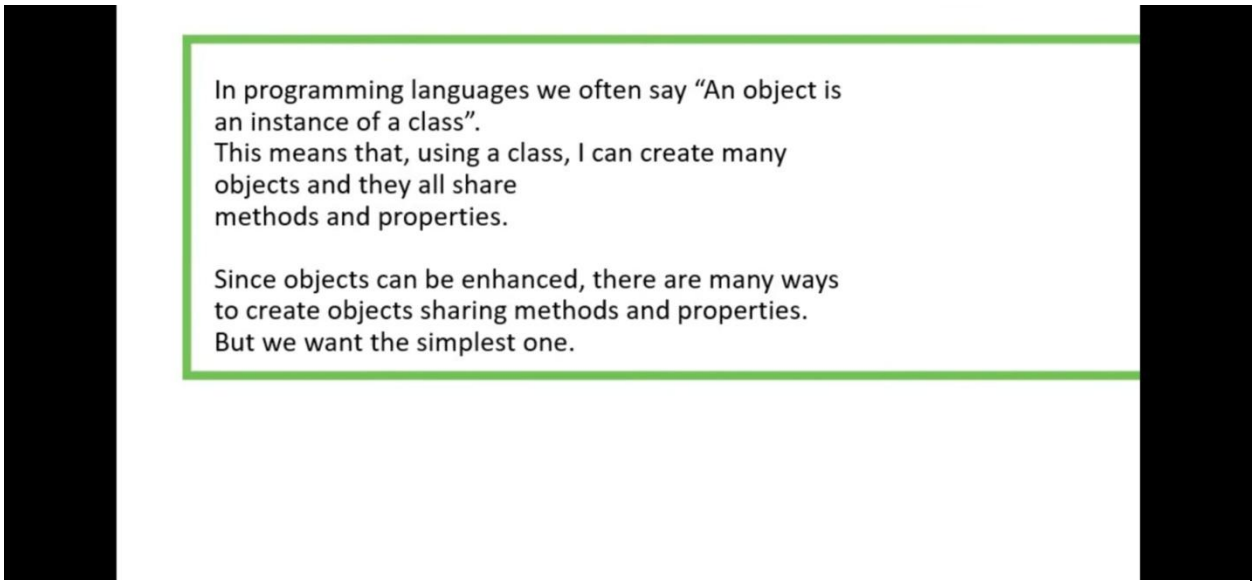
Syntax: arr.forEach(callback(currentValue [, index [, array]])[, thisArg])

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.

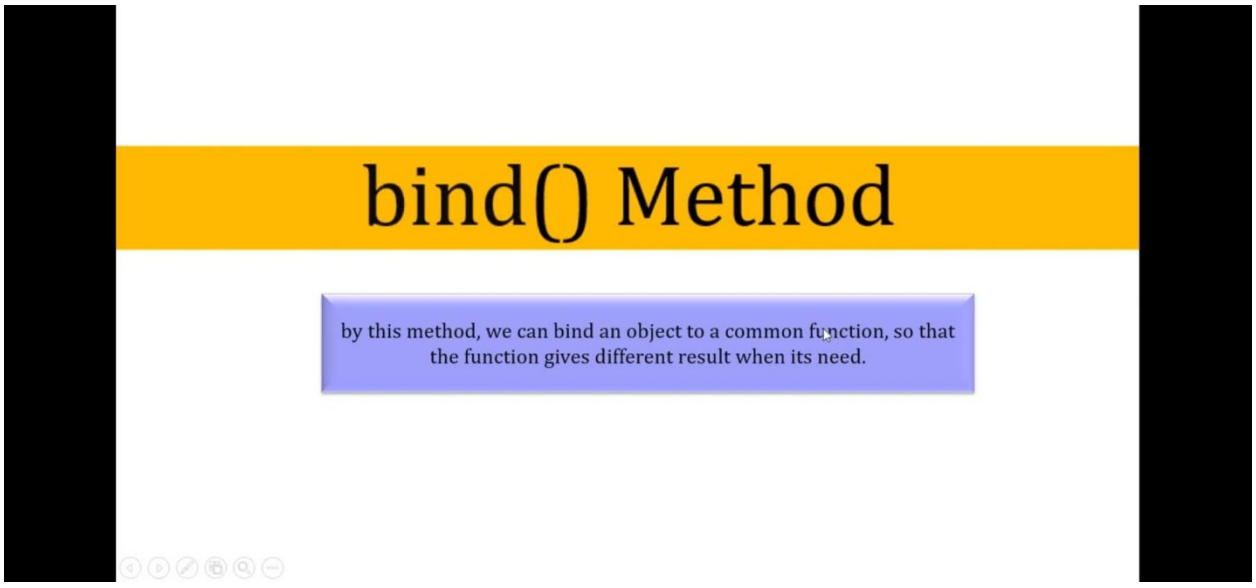
That is why you should use arrow functions. With arrow functions, `this` will always represent the object that defined the arrow function.

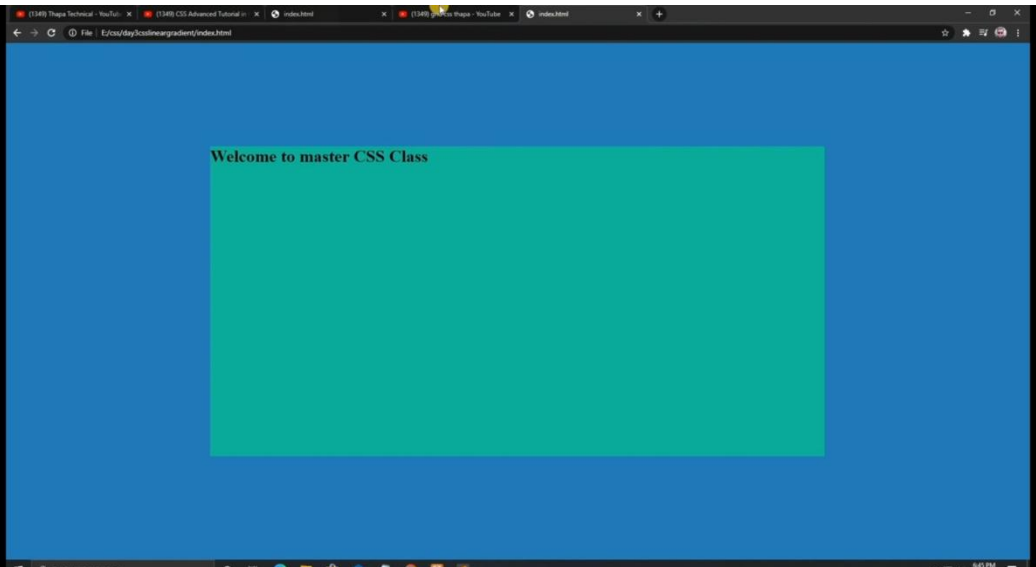
Why Arrow Functions?

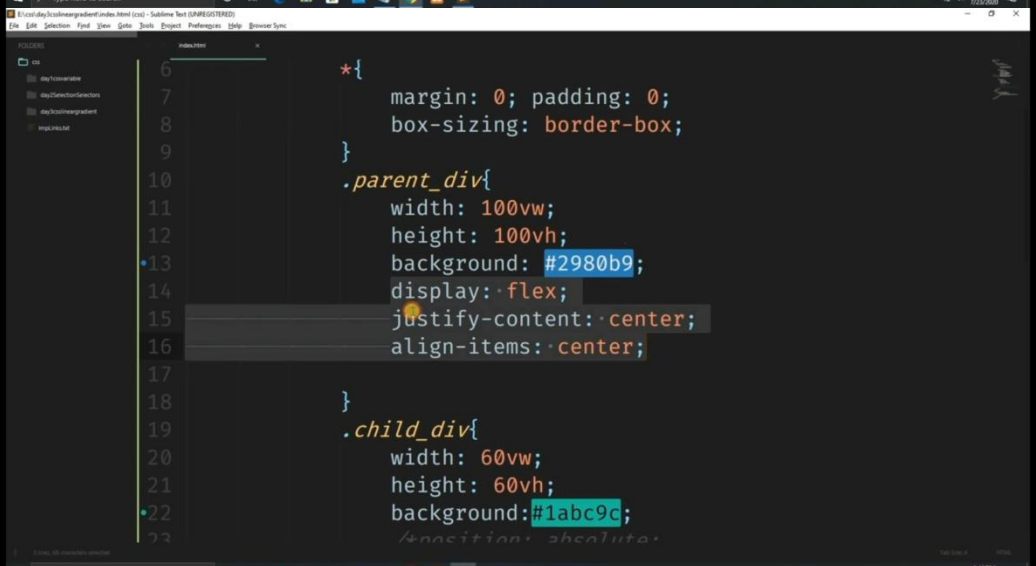
In class components, the `this` keyword is not defined by default, so with regular functions the `this` keyword represents the object that called the method, which can be the global window object, a HTML button, or whatever.

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

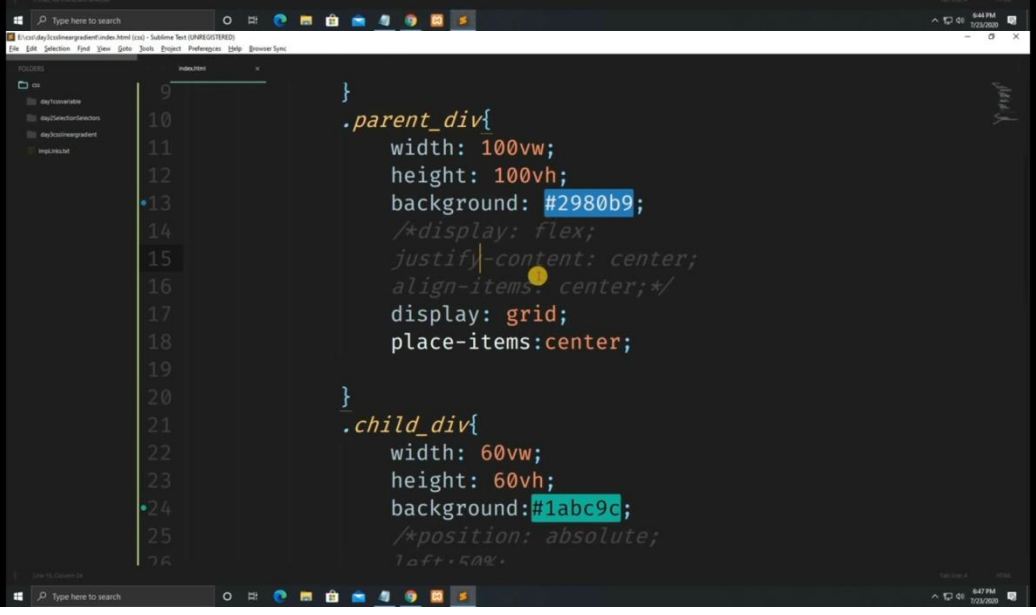
If you *must* use regular functions instead of arrow functions you have to bind `this` to the component instance using the `bind()` method:







```
*{
    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;
```

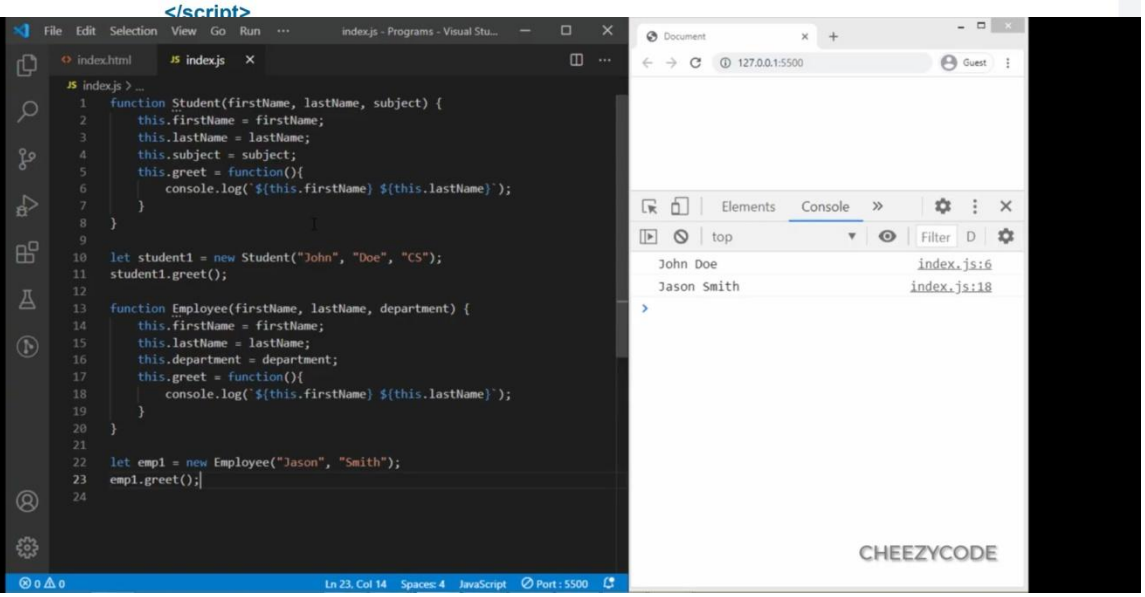


```
}
.parent_div{
    width: 100vw;
    height: 100vh;
    background: #2980b9;
    /*display: flex;
    justify-content: center;
    align-items: center;*/
    display: grid;
    place-items:center;

}
.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


```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

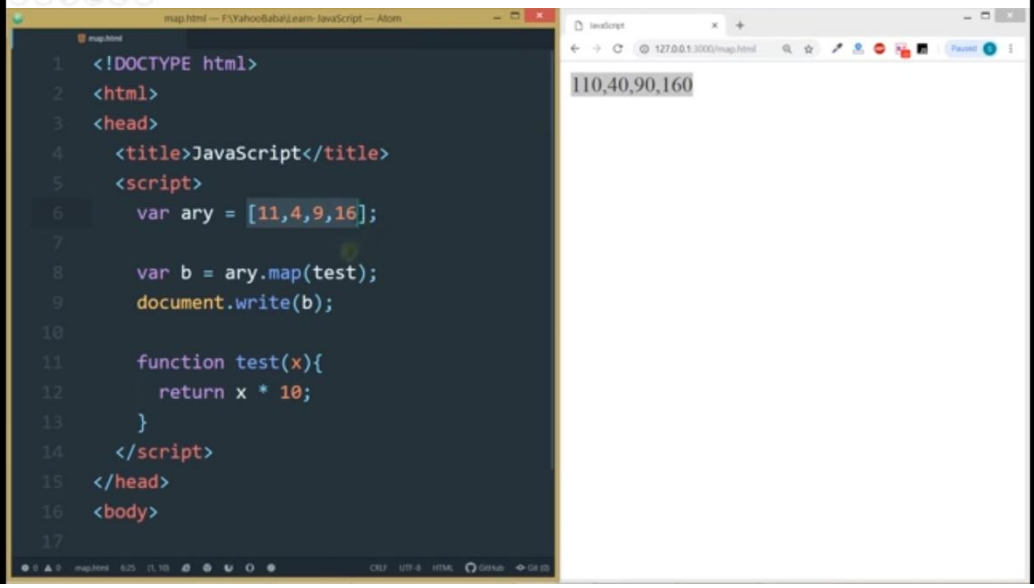
findIndex(Function Name)

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