**Java Script**

* **Different ways to execute JS:**
  + Inside <script>...</script>
  + Loading External Js file with 'src' attribute
  + Using Event handlers
  + Browser console using F12
* **JavaScript Comments:**
  + Single Line Comment : //This is a Comment
  + Another Single Line comment <!--This is commented
  + Multi Line Comment : /\*Multi line comment\*/
  + Lesser Known Single Line Comment : --> This is also a Comment
* **Concatenation in JavaScript:**
  + Since JavaScript is a loosely typed language, We can Concatenate Strings with Strings as well as with Other Data Type like number (integer/floats)/boolean with "+" Operator
  + For example:
    - var x="aaaaaa"+"bbbb"; //console.log(x) would return aaaaaabbbb
    - var x="aaa"+1234;//console.log(x) would return aaa1234
* **Defining Variables**
  + **Var x=1** 
    - This is only used for function scope (local scope inside function) but when it comes to block scope like {for, if} it can be accessed outside a block scope (global scope inside block scope) so this was a problem that why they made ‘let’ in es6
    - Example
      * For (i=0 ; i<7;i++){
        + Var x = 10 // local scope
      * }
      * Console.log(x) //accessed outside the block scope
  + **Let x=1** 
    - This will define a variable (local scope) only inside a block scope like {for , if} and can’t be accessed outside the block scope
    - Only local scope
    - Example
      * For (i=0; i<7;i++){
        + Let x = 10 // local scope
      * }
      * Console.log(x) //error: can’t be accessed outside the block scope
  + **Const pi= 3.14**
    - Constant variable within a block scope and can’t be reassigned
* **Arrow function**
  + **Callback function:**
    - Is a function that has another function as parameter
  + Syntax
    - Let function\_name () => {return 5+60}
    - Let fun\_name (param1)=> {return param1\*2}
  + This
* **Synchronous vs asynchronous programing**
  + Callback funtion
  + Promise
    - Then catch
    - Async/await
  + Ajax
    - <script>
    - var xhrobj=new XMLHttpRequest();
    - xhrobj.onload=function()
    - {
    - if(this.status===200)
    - {
    - var resp=this.responseText;//this.responseText contains the server response, there's also responseXML property Which can be used to parse XML only
    - alert("Server Response: "+resp);//this will alert the HTML content of server response
    - }
    - };
    - xhrobj.open("GET","/readme.html");
    - xhrobj.send();
    - </script>
* **Built-in Functions**
  + **alert(1);prompt(1);confirm(1) -** 
    - alert() is used to popup an alert message box with the Argument String written in it, it doesn't return anything
    - prompt("Argument Here"): ( is used to popup a prompt to take user input with a message supplied in Argument String, it returns the value entered by the user
      * var x=prompt("Enter Value: ");// after User enters value it will be stored in x for further operations
    - confirm("Message"): is used to popup a confirm box to confirm(OK/Cancel) Something, Pressing OK returns true and Cancel returns false.
      * var conf=confirm("Are You Sure To blablabla?");
  + setTimeout() - The setTimeout() method calls a function after a specified number of milliseconds.
    - setTimeout(function(){ alert("Hello"); }, 3000); // will call anonymous function after 3 seconds
  + setInterval() - The setInterval() method calls a function after specified intervals endlessly
    - setInterval(alert(1),3000); // will call alert(1) after every 3 seconds
  + eval("JavaScript here") - used to evaluate string in its argument as JavaScript.
    - eg.eval('var x=1;alert(x)');
* **Document Object Model**
  + Window and Document Object in javascript:
  + Window is the main JavaScript global object, accessed by "window" object it contains all the information about the opened window (height, width, name of the window, DOM etc) and also its opener window (if its there). Document is a property of "window" object only but since "window" object is global it could be accessed by "document" or even as "window.document".
  + So basically, DOM (Document Object Model) loads all the objects inside "WINDOW" and then the "DOCUMENT" object gets loaded inside the window object. Some of the Important Properties of "window" & "document" objects are:
    - window.location.host : "www.host.com"
    - window.location.hostname : "www.hostname.com"
    - window.location.href : "https://www.hostname.com/path/to/file.php"
    - window.location.origin : "https://www.hostname.com"
    - window.location.pathname :"/path/to/file.php"
    - window.location.port :""
    - window.location.protocol : "https:"
  + window.location.hash or location.hash –
    - This property returns the part of URI Fragment ie. Whatever is written in the URL after "#" (hash symbol)
    - For example a URL https://www.securityidiots.com/#blablalabla in that case "location.hash;" would return "#blablalabla" and location.hash.slice(1); would return blablalabla ie. It will slice 1st character from the string, which is #.
    - This part after hash in the URL is for client side usage and hence it couldn't be accessed by Server Side Languages which is why is very helpful in bypassing Server Side WAF.
  + window.location.search or location.search - This property returns the query string or the "GET" parameter. For example:
    - https://www.securityidiots.com/?xyz=1&abc=zen
    - "location.search;" would return "?xyz=1&abc=zen"
  + document.domain - This property is used to return the hostname of where javascript is executing. Similar to location.hostname. This is mainly used to confirm that XSS is executing on the right domain.
  + document.cookie - This property is used to get all Cookies as a String, but if there is a "HttpOnly" flag set in the cookies, then cookies couldn't be accessed via JavaScript.
  + document.getElementsById('123') - This is a Method (a Function) which is used to get all elements (nodes) having 123 value of "id" attribute provided in the argument. We could then even modify/remove the selected nodes
  + There are many other similar methods too to manipulate the DOM like:
    - document.getElementsByName('Name');
    - document.getElementsByTagName('TagName');
    - document.getElementsByClassName('ClassName');
  + document.innerHTML - This method is used to write HTML content within a selected node.

javascript

1]javascript can be written in a separate file

or in the <script></script> tag in

head or body

Placing scripts at the bottom of the

<body> element improves the display

speed

2] JavaScript files have the file extension .js

3] JavaScript uses the var keyword to declare variables.

4] // or between /\* and \*/ is as a comment.

5] function function\_name(parameter1, parameter2,

parameterx) {

// code to be executed

return

}

6] to declare an object //

var object\_name = {firstName:"John",

lastName:"Doe", age:50, eyeColor:"blue"};

7] to make an outpout :

//alert()

//document.write() "for testing only"

//innerhtml()

//console.log() "output is in the console"

8]HTML events are "things" that happen to HTML elements

Here are some examples of HTML events:

a )An HTML web page has finished loading

b)An HTML input field was changed

c)An HTML button was clicked

"onchange" // An HTML element has been

changed

"onclick" // The user clicks an HTML

element

"ondblclick" // The user double clicks an HTML

element

"onmouseover" // The user moves the

mouse over an HTML element

"onmouseout" // The user moves the

mouse away from an HTML element

"onkeydown" // The user pushes a

keyboard key

"onload" // The browser has finished

loading the page

"onblur" // when a user focus on a field then

unfocus

"onkeyup" // when a user

"onsubmit" // when a user submit a form

-element.preventDeault beyemna3 el submit

-Writing into an HTML element, using

" innerHTML."

Writing into the HTML output using

" document.write()" //for testing only

Writing into the browser console, using

" console.log()."

window.alert() //to make an alert box

Html Dom { document object model}

1]control html elements by javascript

-how to find elements in a html page :

document.getelementbyid()

document.getelementsbyclassname()

document.getelementsbytagname()

document.queryselectorall()

when using by class name and tag name we must use an array because it sotre values as an array

2] you can access any element in a page Ex:

document.title

document.images

document.body

document.forms

document.links

document.anchors //without href

3]you "can get and set" the content of any element by :

innerhtml // deal with text and if any element inside the element selected

textcontent // deal with text only

outerhtml

4] you can change and select any attribute by

{ ELEMENT.ATTRIBUTE = "change value" }

Ex: div.width

element.getAttibute ('att\_name')

element.setAttribue('att\_name' . 'att\_val')

element.hasattribute('attribute name')

// used in conditions

element.removeattribute('attribute name')

// remove attribute

5] you can deal with child element by :

a> { ELEMENT.CHILDREN // get all html elements under that element }

b> { ELEMENT.CHILDNODE // get all html elements and text element , (text without a tag) under that element }

c>{ ELEMENT.FIRSTCHILD // get the first child under that element ; can be a text node or element node}

d> { ELEMENT.FIRSTELEMENTCHILD // get the first child under that element ; an element between tags}

e> { lastchild / lastelementchild }

f> { ELEMENT1.appendchild(ELEMENT2) }

// ba7ot element2 in element1

g> { ELEMENT.REMOVECILD () }

// remove a child in that element

6] document.CreateElement()

document.createTextNode ()

document.createcomment ()

document.createAttribute () // create an attribue

and then set that attribute by " setattributeNode () "

7] Element.nodename // print name of anything

including text without tags and comments

Element.tagname // print name of tag elements

only Ex : p , div ,..ect

8] to change the style of an element Ex:

Element.style.color =' ';

9] document.inputEncoding // show the encoding of

the page Ex: utf-8

document.url // show url of the page

10]addEventListener('event\_name' , function())

11]Element1.contains(element2) //check if ele2 is in

element1

12]Element.clientheight //get the height of that elem

Element.clientwidth //get the width of that elem

{doesnt include the padding , border , scroll}

Element.offsetheight //get the height of that elem

Element.offsetwidth //get the width of that elem

{include the padding , border , scroll}

HTML BOOM [browser object model]

control with the browser

1] { window } is the defualt object of the browser

any code i write window object is inserted

automaticly before it ex: window.alert()

window.document.getelementbyid()

2] window.confirm('') // print an ok or cancel message

if (confirm("Press a button!")) {

txt = "You pressed OK!";

} else {

txt = "You pressed Cancel!"; }

window.alert('') // print a message on screen

window.prompt ('') // user will a enter a value on

screen

3] settimeout (function() , millisecond) // do this

function after that tim e one time only

cleartimeout ((idofsettimeout) // stop the

settimeout

clearsetinterval ((id) // stop the setinterval

setinterval(function() , 2000) // same like settimeout by can do the function every nb of sec entered here

2s

4]window.open ("url" , "attribute" , "specefication") // open that url and

the attributes can be \_self or \_blank or \_target

and the specefication of the window opened like

its width and height and left ....

5] window.scrollby(fom left , from top) // in px

window.scrollto(0 , 0) // beyrou7 lel makan elli enta

katbo unlike scrollby elli btscroll bel input kol mara

window.stop // stop loading of the page

window.close // to close the window opened by

window.open

6] window.innerwidth

window.innerheight

//show the width & height of the browser

window and if you rezise it will change

window.outerwidth

window.outerheight

//show the width & height of the browser

window and if you rezise it wont change

7] window.pagexoffset // show how much the page

is scrolled to the right

window.pageyoffset // show how much the page

is scrolled to the bottom

8]

window.location.href = www.facebook.com //will redirect to tfacebook

window.location.href can open protocols

a)window.location.href // returns the href (URL) of

the current page

b )window.location.hostname // returns the domain

name of the web host

c)window.location.pathname // returns the path and

filename of the current page

d)window.location.protocol // returns the web

protocol used (http: or https:)

e)window.location.hash = " id" // will go to the # in a page

a function can be called when an event occurs or self invoked by