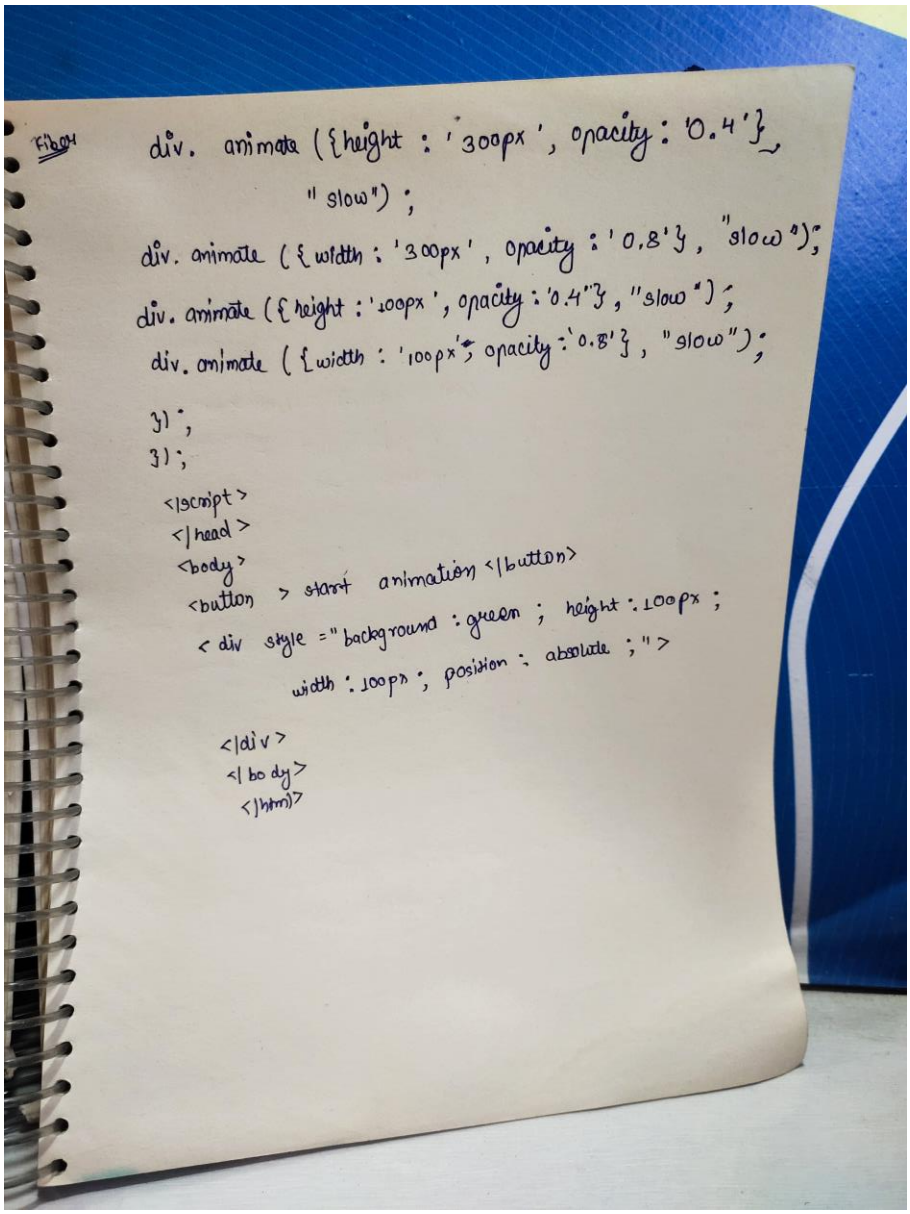


Fiber

```
div. animate ({height : '300px', opacity : '0.4'},
                "slow");
div. animate ({width : '300px', opacity : '0.8'}, "slow");
div. animate ({height : '100px', opacity : '0.4'}, "slow");
div. animate ({width : '100px', opacity : '0.8'}, "slow");
});
});
<|script>
 <| head >
 <body>
  <button > start animation <|button>
  < div style = " background : green ; height : 100px ;
          width : 100px ; position : absolute ; " >

  <|div >
  <| bo dy >
   <|html>
```

27/09/2023

① <script>
$(document).ready(function(){
$("button").click(function(){
$("div").animate({
left:'250px',
height:'+=150px',
width:'+=150px'
});
});
});
</script>
<head>
<body>
<button> start Animation </button>
<div style="background:green; height:100px;
width:100px; position:absolute;">

</body>
</html>

② height='toggle'

③ <script>
$(document).ready(function(){
$("button").click(function(){
var div=$("div");

```
width : '150px'
});
});
});
</script>
</head>
<body>
<button > start Animation </button>
<div style = "background : red ; height : 200px ; width : 300px ;
position : absolute ; "> </div>
```

23/09

Q:- Write a Javascript function to split a string and convert it into an array of words.

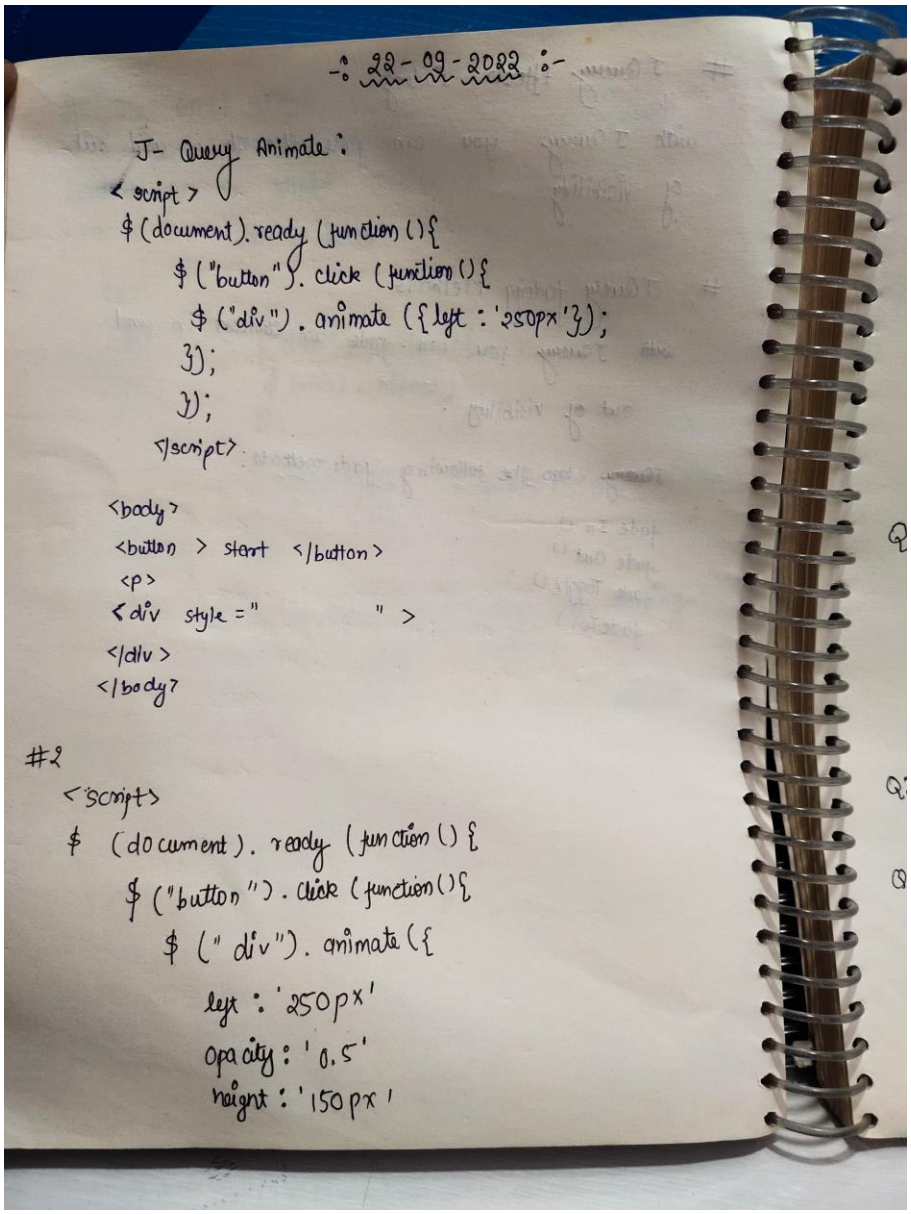
Test data:
console.log(string_to_array("Robin Singh"));

output = ["Robin", "Singh"]

Q:- Write a Javascript function to extract a specified number of characters from a string.

Q:- Write Javascript function to capitalize the first letter of a string.

Q Abbreviate the string



-: 22-09-2022 :-

J- Query Animate :

```
<script>
$(document).ready(function(){
    $("button").click(function(){
     $("div").animate({left:'250px'});
    });
    });
</script>

<body>
<button > start </button>
<p>
<div style="          " >
</div>
</body>
```

#2

```
<script>
$ (document).ready(function(){
    $("button").click(function(){
      $("div").animate({
        left:'250px'
        opacity:'0.5'
        height:'150px'
```

JQuery Effects - Fading

With JQuery you can fade elements in and out of visibility.

JQuery fading Methods

With JQuery you can fade an element in and out of visibility.

JQuery has the following fade methods:

fadeIn()
fadeOut()
fadeToggle()
fadeTo()

The on() method

- The on() method attaches one or more event handlers for the selected elements
- Attach a click eleverit to a <p> element:

Ex:-

```
$("p").on("click", function(){
    $(this).hide();
});
```

20/09/22 swastik printing

JQuery toggle()

You can also toggle b/w hiding and showing an element with the toggle() method.

shown elements are hidden and hidden element are shown.

Ex:-

```
$("button").click(function(){
    $("p").toggle();
});
```

blur()

- The blur() method attaches an event handler function to an HTML form field.
- The function is executed when the form field loses focus:

Example :-

```
$("input").blur(function(){
    $(this).css("background-color", "#ffffff");
});
```

jQuery Effects - Hide and Show

Hide, Show, Toggle, Slide, Fade, and Animate.

JQuery hide() and show()

With JQuery, you can hide and show HTML elements with the hide() and show() methods:

Example:

```
$("#hide").click(function(){
    $("p").hide();
});

$("#show").click(function(){
    $("p").show();
});
```

16/09/2022

Mousedown()

. The mousedown() method attaches an event handler function to an HTML element.

. The function is executed, when the left, middle or right mouse button is pressed down, while the mouse is over the HTML element:

Example:-

```
$(document).ready
    $("#p1").mousedown(function(){
      alert("Mouse down over p1!");
    });
```

Mouse up()

The mouseup() method attaches an event handler function to an HTML element.

The function is executed, when the left, middle or right mouse button is released, while the mouse is over the HTML element:

Example -

```
$("#p1").mouseup(function(){
    alert("Mouse up over p1!");
});
```

Example -

moving a mouse over an element

selecting a radio button

click on an element.

JQuery syntax for Event method

In JQuery, most DOM events have an equivalent JQuery method.

→ To Assign a click event to all paragraph on a page, you can do this:

```
$("p").click();
```

→ The Next step is to define what should happen the event fires. You must pass a function to the event :

```
$("p").click();
```

The Class selector :-

→ The JQuery class selector finds elements with a specific class.

→ To find elements with a specific class, write a period character followed by the name of the class.

$(".test")

→ When a user clicks on a button, the elements with class = "test" will be hidden.

Example :-

```
$(document).ready(function(){
$(":button").click(function(){
$(".test").hide();
});
});
```

JQuery Event method

J Query is tailor-made to respond to event in an HTML Page.

→ What are Events

• All the different visitors actions that a webpage can respond to are called Events

→ An event represents the precise moment when something happens.

Example :-

$(this).hide() — hides the current element

$("p").hide() - hides the all element with class = "text"

The #id selector :-

- The jQuery #id selector uses the id attribute of an HTML tag to find the specific element.
- An id should be unique within a page so you should use the #id selector when you want to find a single unique element.
- To find an element with a specific id, write a hash character, followed by the id of the HTML element.

  $("#test")

  Example

  When a user clicks on a button, the element with id = "test" will be hidden.

Ex:-

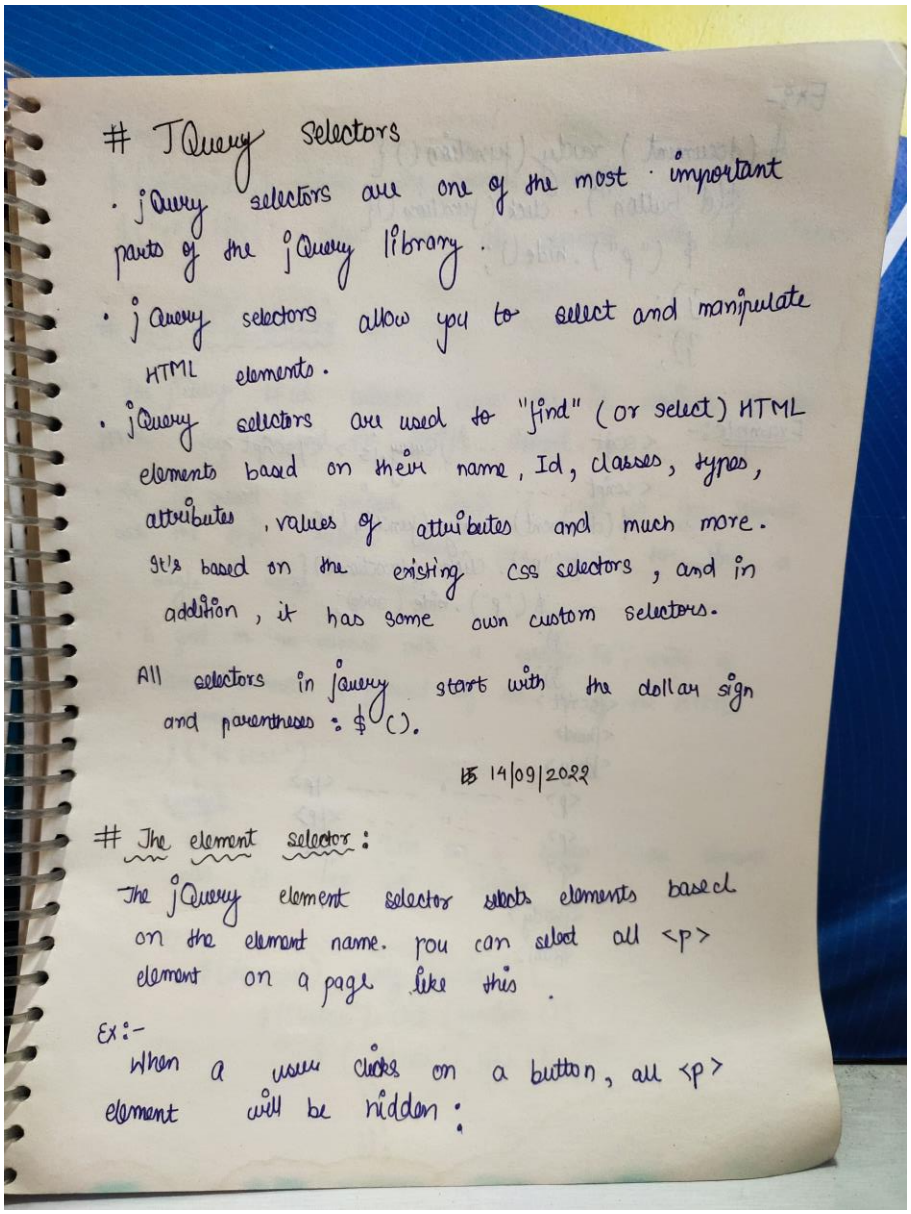
```
$(document).ready(function(){
    $("button").click(function(){
        $("#test").hide()
    });
});
```

Ex:-

```
$(document).ready(function(){
  $("button").click(function(){
    $("p").hide();
  });
});
```

Example:-

```
<script ... "jquery.js"></script>
<script ...
  $(document).ready(function(){
    $("p").click(function(){
      $("p").hide(3000);
    });
  });
</script>
</head>
<body>
  <p> ... </p>
  <p> ... </p>
  <p> ... </p>
</body>
</html>
```



JQuery Selectors

- jQuery selectors are one of the most important parts of the jQuery library.
- jQuery selectors allow you to select and manipulate HTML elements.
- jQuery selectors are used to "find" (or select) HTML elements based on their name, Id, classes, types, attributes, values of attributes and much more. It's based on the existing CSS selectors, and in addition, it has some own custom selectors.

All selectors in jQuery start with the dollar sign and parentheses : $().

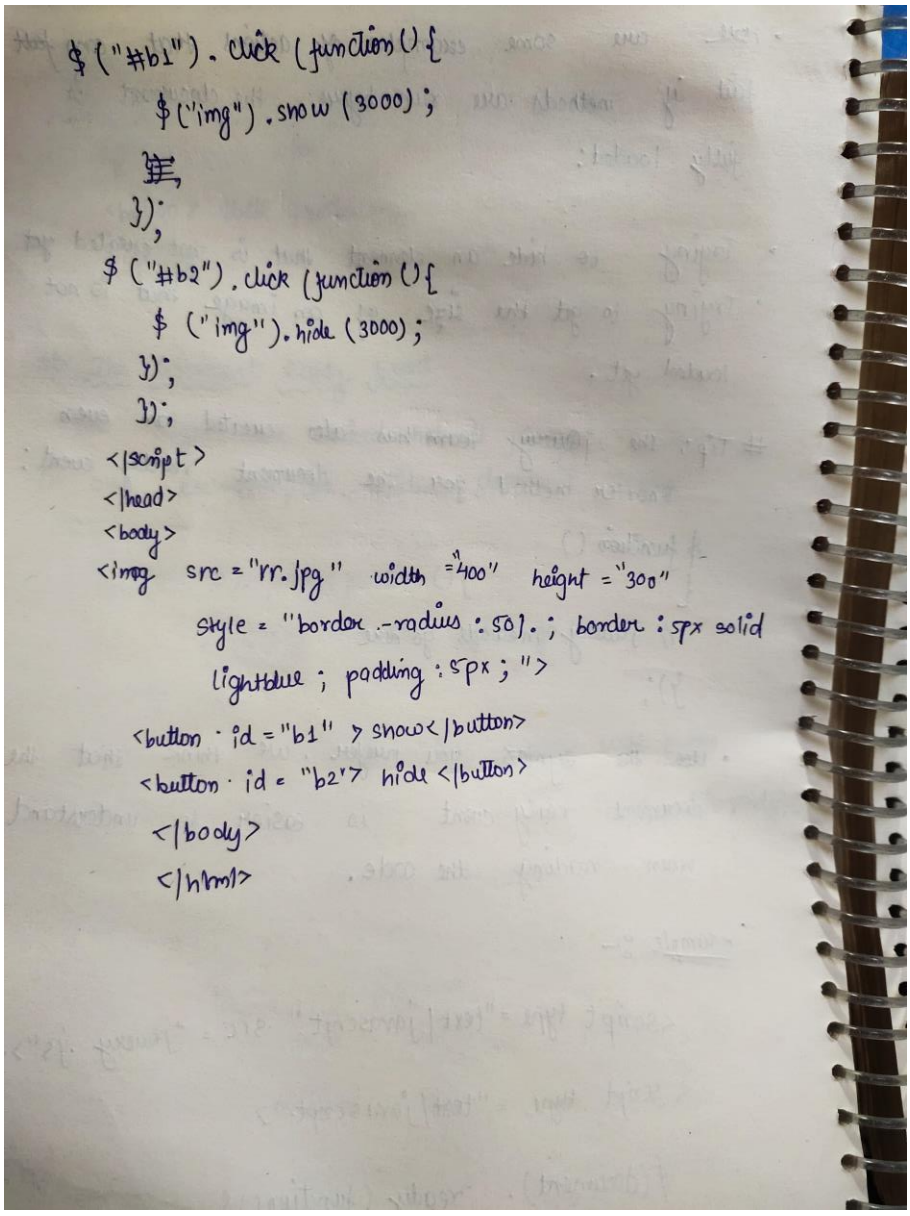
14/09/2022

The element selector :

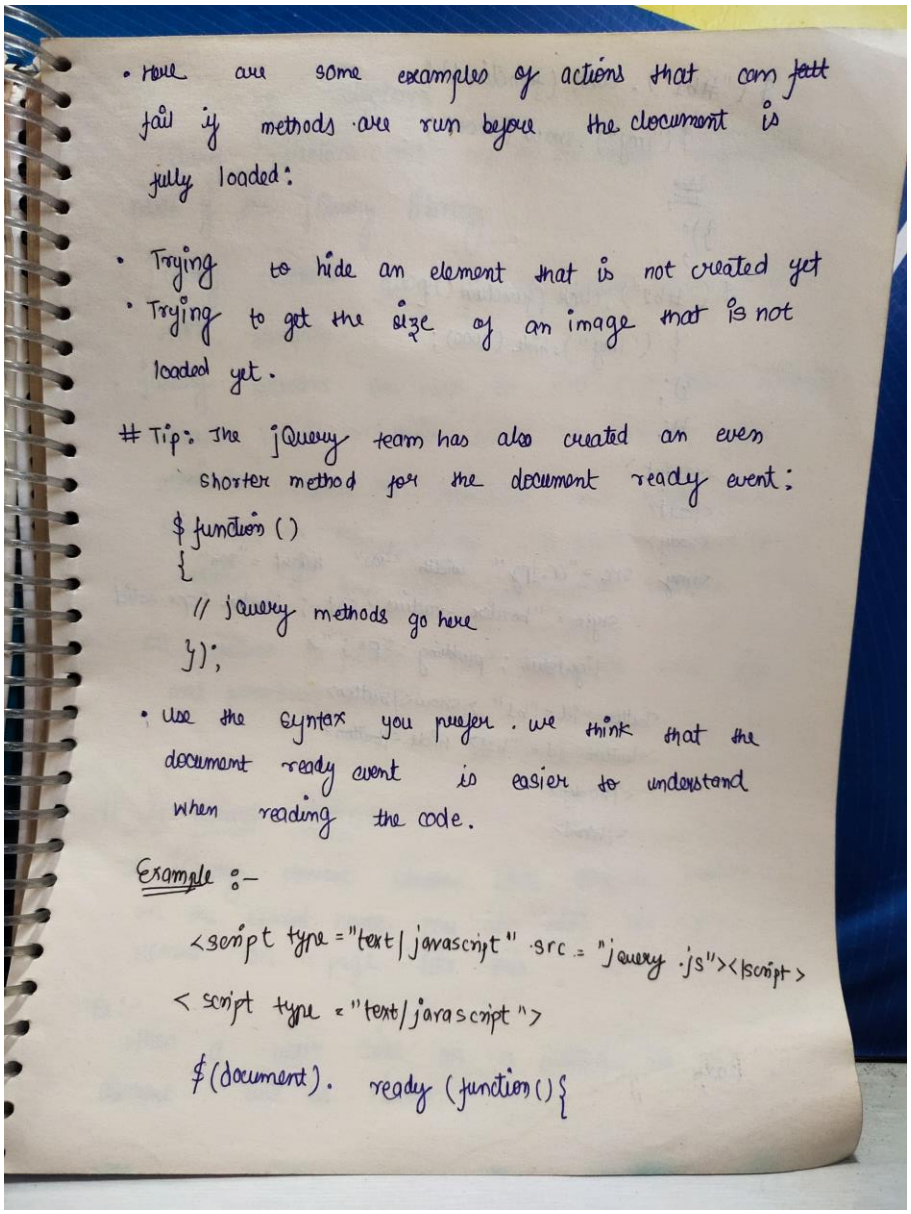
The jQuery element selector selects elements based on the element name. You can select all <p> element on a page like this.

Ex:-

When a user clicks on a button, all <p> element will be hidden :



```
$("#b1").click(function(){
    $("img").show(3000);
});
$("#b2").click(function(){
    $("img").hide(3000);
});
</script>
</head>
<body>
<img src="rr.jpg" width="400" height="300"
    style="border-radius:50%; border:5px solid
    lightblue; padding:5px;">
<button id="b1">show</button>
<button id="b2">hide</button>
</body>
</html>
```



- Here are some examples of actions that can fail if methods are run before the document is fully loaded:

- Trying to hide an element that is not created yet
- Trying to get the size of an image that is not loaded yet.

Tip: The jQuery team has also created an even shorter method for the document ready event:

```
$ function ()
{
 // jquery methods go here
});
```

- Use the syntax you prefer. We think that the document ready event is easier to understand when reading the code.

Example :-

```
<script type = "text/javascript" src = "jquery.js"></script>
<script type = "text/javascript">
$(document). ready (function(){
```

```
</head>
<body>
  <p> ..... </p>
  <button> Click </button>
 </body>
</html>
```

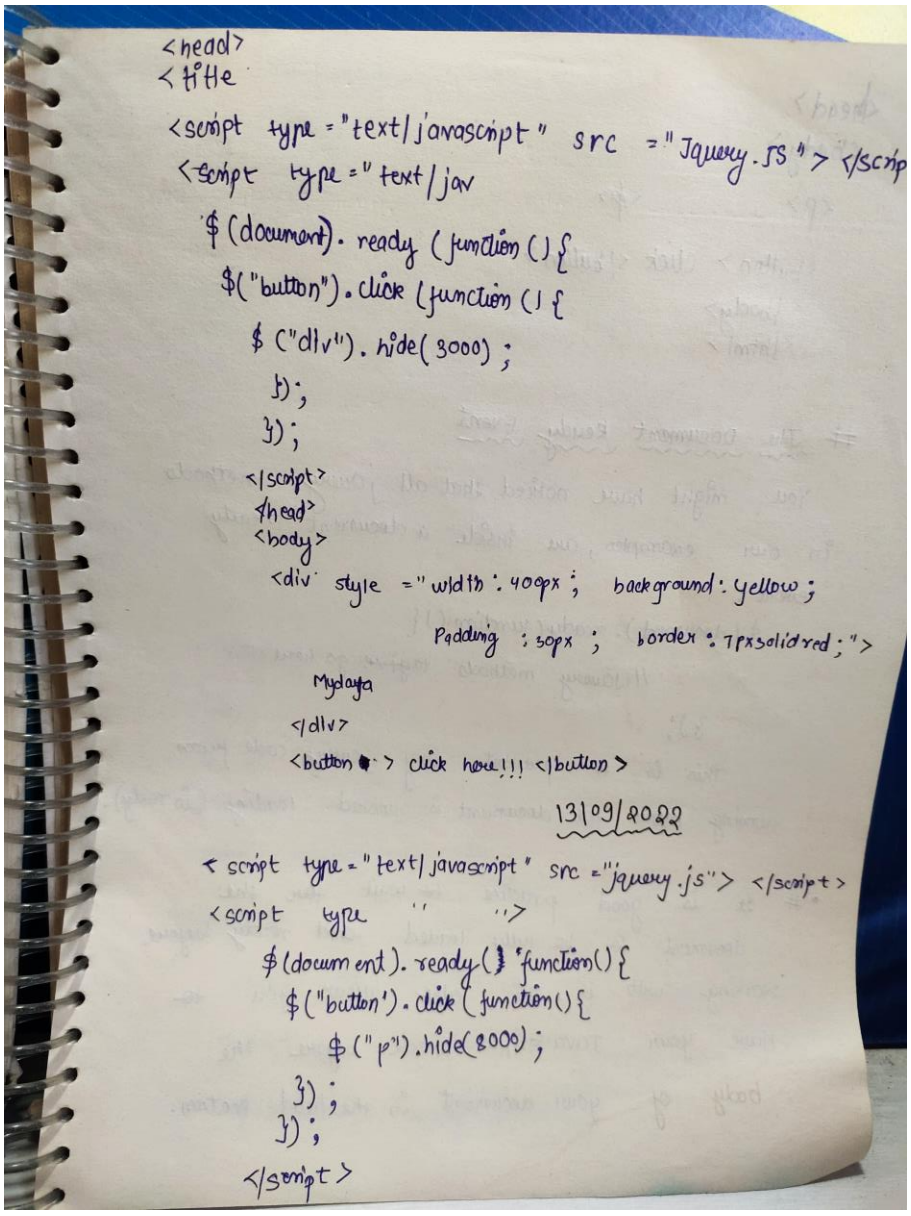
The Document Ready Event

You might have noticed that all jQuery methods in our examples, are inside a document ready event:

```
$(document).ready(function(){
    // jQuery methods to give go here...
});
```

This is to prevent any jQuery code from running before the document is finished loading (is ready).

It is good practice to wait for the document to be fully loaded and ready before working with it. This also allows you to have your Javascripts code before the body of your document, in the head section.



```
<head>
<title
<script type="text/javascript" src="Jquery.js"></scrip
<script type="text/jav
$(document).ready(function(){
$("button").click(function(){
$("div").hide(3000);
});
});
</script>
</head>
<body>
<div style="width:400px; background:yellow;
Padding:30px; border:7px solid red;">
Mydata
</div>
<button> click here!!! </button>
```

13/09/2022

```
<script type="text/javascript" src="jquery.js"></script>
<script type="">
$(document).ready(function(){
$("button").click(function(){
$("p").hide(8000);
});
});
</script>
```

→ The JQuery is a single Javascript file, and you reference it with the HTML <script> tag (notice that the <script> tag should be inside the <head> section)

```
<head>
<script src="jquery-3.5.1.min.js"></script>
```

JQuery Syntax

The JQuery syntax is tailor-made for selecting HTML elements and performing some action of the element(s).

- Basic syntax:

```
$(selector).action();
```

A $ sign to define / access JQuery
A (selector) is HTML
A action() is command of JQuery

Why JQuery ?

There are lots of other Javascript libraries out there, but J Query is probably the most popular, and also the most extendable

Many of the biggest companies on the Web use JQuery such as -

- Google
- Microsoft
- IBM
- Netflix

Adding JQuery to your Web Pages

→ There are several ways to start using J Query on your web site. you can:

1. Download the JQuery library from JQuery.com (uncompressed)
2. Include J Query from a CDN, like google
3. Downloading J Query.

→ There are two version of J Query available for downloading

Both versions can be downloaded from JQuery.com

JQuery library of JS

What is JQuery?

- JQuery is a lightweight, "write less, do more", Javascript library.
- The purpose of JQuery is to make it much easier to use Javascript on your website.
- JQuery takes a lot of common tasks that require many lines of Javascript code to accomplish, and wraps them into methods that you can call with a single line of code.
- JQuery also simplifies a lot of the complicated things from Javascript like AJAX calls and DOM Manipulation.

JQuery library contains the following features:

- HTML/DOM manipulation
- CSS

10/09/2022

Q:- Write a Javascript program to find the most frequent item of an array.

simple array : var arr1 = [3, 'a', 'a', 2, 3, 'a', 3, 'a', 2, 4, 9, 3];

sample output : a(5times)

Q WAJP which accept a string as input and swap the case of each character. for example if you input 'The Quick Brown Fox' the output should be 'tHE qUICK bROWN fOX'.

Q WAJP to compute the sum and product of an array of integers.

Q WAJFunction to check whether a string is blank or not.

Example

```
<script
let text = "a,b,c,d,e,f";
document.write(typeof(text));
const myArray = text.split(",");
document.write(type of (Array myArray));
document.write(my Array);
</script>
```

Example :-

```
let text = "Bhopal, Indore, Gwalior, Jabalpur,
            satna, Panna, Ujjain";
let mydata = text.split(",");
document.write("<ul>");
for (let i=0; i<mydata.length; i++)
{
document.write("<li>", mydata[i], "</li>");
}
document.write("</ul");
</script>
```

Q:-

```
a=3;
b="3";
c=a-b   // 0
c=a+b   // 33
<script>
a=3;
b="3$";
c=a-b; // str nts or sts we get conctaenation
document.write("<h1>", c, "</h1>");
</script>
```

Converting a string to an Array -

If you want to work with a string as an array, you can convert it to an array.

Javascript string.split()

A string can be converted to an array with the split() method.

Example -

```
text.split(",")   // split on comas
text.split(" ")   // split on spaces
text.split("!")   // split on pipe
```

#Note

Property access might be a little unpredictable :

- It makes strings look like arrays (but they are not)
- If no character is found, [ ] returns undefined; while charAt() returns an empty string.
- It is read only.

str[0] = "A" gives no error (but does not work!)

Example

```
let text = "Cybrom";
text[0] = "A";  // Gives no error, but does not work
```

-: 09-09-2022 :-

eval() :--

The eval() method ~~evaluation~~ evaluate or executes an argument. If the argument is an expression, eval() evaluates the expression. If the argument is one or more JavaScript statements, eval() executes the statement.

eval() also work in string statement.

Example - S = "2*3+4(9/2)";

q := eval(S)

#Note

Property access might be a little unpredictable :

- It makes strings look like arrays (but they are not)
- If no character is found, [ ] returns undefined ; while charAt() returns an empty string.
- It is read only.

str[0] = "A" gives no error (but does not work !)

Example

```
let text = " Cybrom ";
text [0] = "A";  // Gives no error, but does not work
```

-: 09-09-2022 :-

eval() :--

The eval() method ~~evaluation~~ evaluate or executes an argument. If the argument is an expression, eval() evaluates the expression. If the argument is one or more JavaScript statements, eval() executes the statement.

eval() also work in string statement.

Example - S = " 2*3+4(9/2) " ;

q := eval(S)

Question — any string taking from user and put *** in middle (hint m/2)

-: 8/09/2022 :-

Javascript string charCodeAt()

The charCodeAt() method returns the unicode of the character at a specified index in a string :

The method returns a UTF-16 code (an integer between 0 and 65535)

Example-:

```
<script
let txt = "Welcome!!";
mycode = txt.charCodeAt(0);
document.write("Mycode: ", mycode);
</script>
```

Property Access

ECMAScript 5 allows property access [ ] on string :

```
let msg = "Cybrom";
let mydata = msg[0];
document.write("<h1>", mydata, "</h1>");
</script>
```

7/09/2022

- pad end()

The padEnd() method pads a string with another string.

```
let str = "India";
let mystr = str.padEnd(9,"*");
document.write("<h1>", mystr, "</h1>");
</script>
```

Extracting string characters

There are 3 methods for extracting string characters:

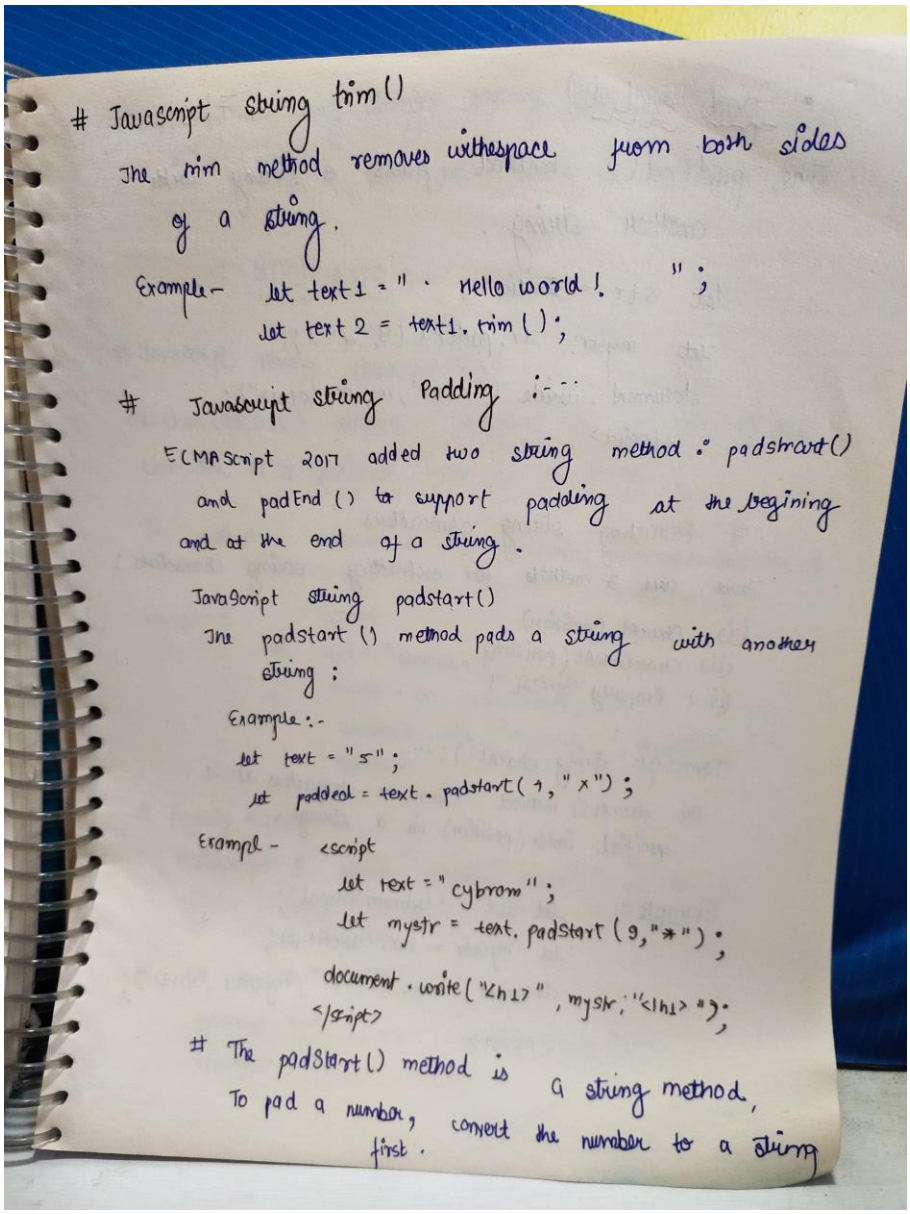
(i) charAt(position)
(ii) charCodeAt(position)
(iii) Property access[ ]

Javascript string charAt():---

The charAt() method returns the characters at a specified index (position) in a string.

Example :-

```
let str = "Cybrom Bhopal";
let mychar = str.charAt(0);
document.write("<h1>", mychar, "</h1>");
</script>
```



Javascript string trim()

The trim method removes withespace from both sides of a string.

Example- let text1 = " . Hello world ! " ;
let text2 = text1. trim ( ) ;

Javascript string Padding :- -

ECMAScript 2017 added two string method : padstart() and padEnd () to support padding at the begining and at the end of a string .

JavaScript string padstart()

The padstart () method pads a string with another string :

Example :-

let text = "5" ;
let paddedd = text . padstart (4 , " x ") ;

Exampl - <script
let text = " cybrom " ;
let mystr = text. padStart (9, " * ") ;
document . write (" <h1> " , mystr, " </h1> ") ;
</script>

The padStart() method is a string method,
To pad a number, convert the number to a string first .

Example:-

```
text = "Hello" + " " + "world!";
text1 = "Hello".concat(" ", "world!");
```

or

```
(" ", "software", text1);
```

(or)

```
let text1 = "Hello world";
let text2 = "we are students from cybrom !!!";
let text3 = text1 + " " + "software" + text2;
document.write(text3);
</script>
```

#Note:

→ All string methods return a new string. They don't modify the original string.

formally said:

→ Strings are immutable: strings cannot be changed, only replaced.

Converting to upper and lower case # 6/05

. A string is converted to upper case with toupper case(

. A string is converted to lower case with toLowerCase():

Example-

```
<script
  let text1 = "Hello world!";
  let text2 = text1.toUpperCase();
  document.write ("Upper case:", text2);
  </script>
```

Example :-

```
let text1 = "Hello World";
let text2 = text1.toLowerCase();
document.write ("upper case:", text2);
```

JavaScript String concat()

Concat() joins two or more strings :

Example :-

```
let txt1 = "Hello";
let txt2 = "World";
let text3 = text1.concat (" ", text2);
```

The concat() method can be used instead of the plus operator. These two lines do the same.

let myans = txt.replace ("microsoft", "cybrom");
document.write ("<h1>", myans, "</h1>");
</script>

If you want to replace all matches, use a regular expression with the /g flag set.

Examples- <script>
let txt = "Welcome to microsoft ! we are microsoft students microsoft is a good company";
document.write <"<h1>", txt, "</h1>");
let myans = txt.replace (/microsoft/g, "cybrom");
document.write ("<h1>", myans, "</h1>");
</script>

By default, the replace() method is case sensitive. Writing Microsoft (with uppercase) will not work.

To replace case insensitive, use a regular expression with an /i flag (insensitive):

Example let txt = "welcome microsoft ! we are Microsoft students MiCrosoft is a god company";
document.write (txt);
let myans = txt.replace (/microsoft/gi, "cybrom");
document.write (myans);

3/09/2022

Replacing string content

The replace() method Replaces a specified value with another value in a string.

```
<script
let txt = " welcome we are microsoft students!!! ";
document.write ("<h1>", txt, "</h1>");
let myans = txt.replace("microsoft", "cybrom");
document.write ("<h1>", myans, "</h1>");
</script
```

- The replace() method does not change the string it is called on.
- The replace() method returns a new string.
- The replace() method replaces only the first match.
- If you want to replace all matches, use a regular expression with the /g flag set.

* By default, the replace() method replaces only the first matches:

```
<script
let txt =" welcome microsoft we are microsoft
students microsoft is a good company!!!";
document. write (" <h1> ,txt, "</h1>");
```

Example :

```
<script
  let str = "abcdefg nac q yznop";
  let myans = str.substring (4);
  document.write ("<h1>", myans, "</h1>");
</script>
```

Javas string substr()

Java Substr() is similar to slice().

The difference is that the second parameter specifies the length of the extracted part.

```
<script
  let str1 = "Hello we are learn web development program";
  let ans = str1.substr(6, 12);
  document.write ("<h1>", ans, "</h1>");
</script>
```

2/09/2022

Extracting string parts#

There are 3 methods for extracting a part of a string:

1. slice (start, end)
2. substring (start, end)
3. substr (start, length)

Javascript string slice ()#

slice () extracts a part of a string and returns the ~~extra~~ extracted part in a new string.

The method takes 2 parameters: the start position, and the end position (end not included).

```
<script                    "0135973100121319 04"
                                    or
let str = "We are students of cybrom technology Bhopal!!!";
let myans = str.slice (4,12);
document.write ("<h1>", myans, "</h1>");
</script
```

Javas string substring()#

substring () is similar to slice ().

The difference is that start and end values less than 0 are treated as 0 in substring ().

If you omit the second parameter, substring () slice out the rest of the string.

Question :- Write a Javascript program to sum the multiples of 3 and 5 under 1000.

Question :- Write a Javascript function to check whether as 'input' is an array or not.

Question. Write a simple Javascript program to join all elements of the following array into a string. Go to the editor.

Sample array : my color = ["Red", "Green", "White", "Black"];

Expected Output :

"Red, Green, White, Black"

"Red, Green, White, Black"

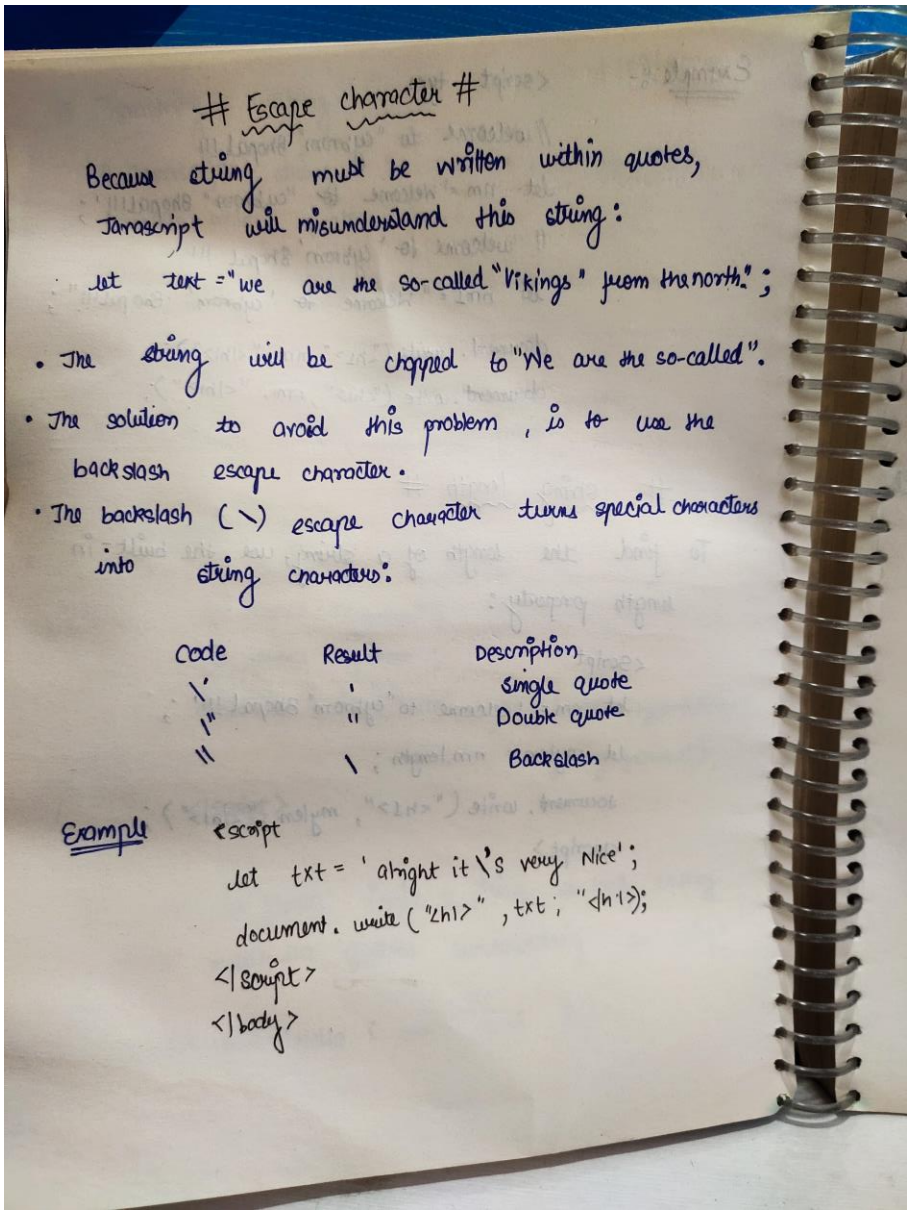
"Red + Green + White + Black"

```
<script
// Welcom to 'cybrom' Bhopal
let txt ='Welcome To \'cybrom\' Bhopal' ;
document . write ("<h1>", txt, "</h1>");
</script>

<script
// Welcome to cybro"m Bhopal!!!
let txt= " Welcome to ~~cybrom"~~ cybrom\"m Bhopal;
document. write ("<h1>", txt, "</h1>");
</script>

<script
// Welcome to " cybrom" Bhopal
let txt =" Welcome to\" Cybrom\" Bhopal ";
document .write ("<h1>", txt, "</h1>");
</script>

<Script
// Welcome to cybrom\we are Student
let txt = "Welcome to cybrom \\ we are students';
document.write ("<h1>", txt, "</h1>");
</Script>
```



Escape character

Because string must be written within quotes, Javascript will misunderstand this string:

let text = "we are the so-called "Vikings" from the north.";

- The string will be chopped to "We are the so-called".
- The solution to avoid this problem, is to use the backslash escape character.
- The backslash (\) escape character turns special characters into string characters:

Code	Result	Description
\'	'	Single quote
\"	"	Double quote
\\	\	Backslash

Example

```
<script
let txt = 'alright it\'s very Nice';
document.write("<h1>", txt, "</h1>");
</script>
</body>
```

Example :-

```
<script type
// welcome to "cybrom" Bhopal !!!
let nm = 'Welcome to "cybrom" Bhopal!!!';
// welcome to 'cybrom' Bhopal !!!
let nm1 = " Welcome to 'cybrom' Bhopal!!!";
document.write("<h1>", nm, "</h1>");
document.write("<h1>", nm1, "</h1>");
```

string length

To find the length of a string, use the built-in length property:

```
<script
let nm = 'welcome to "cybrom" Bhopal !!!';
let mylen = nm.length;
document.write("<h1>", mylen, "</h1>");
</script>
```

Javastring Strings:- 1/09/2022

A Javascript string is zero or more characters written inside quotes.

You can use single or double quotes:

Example

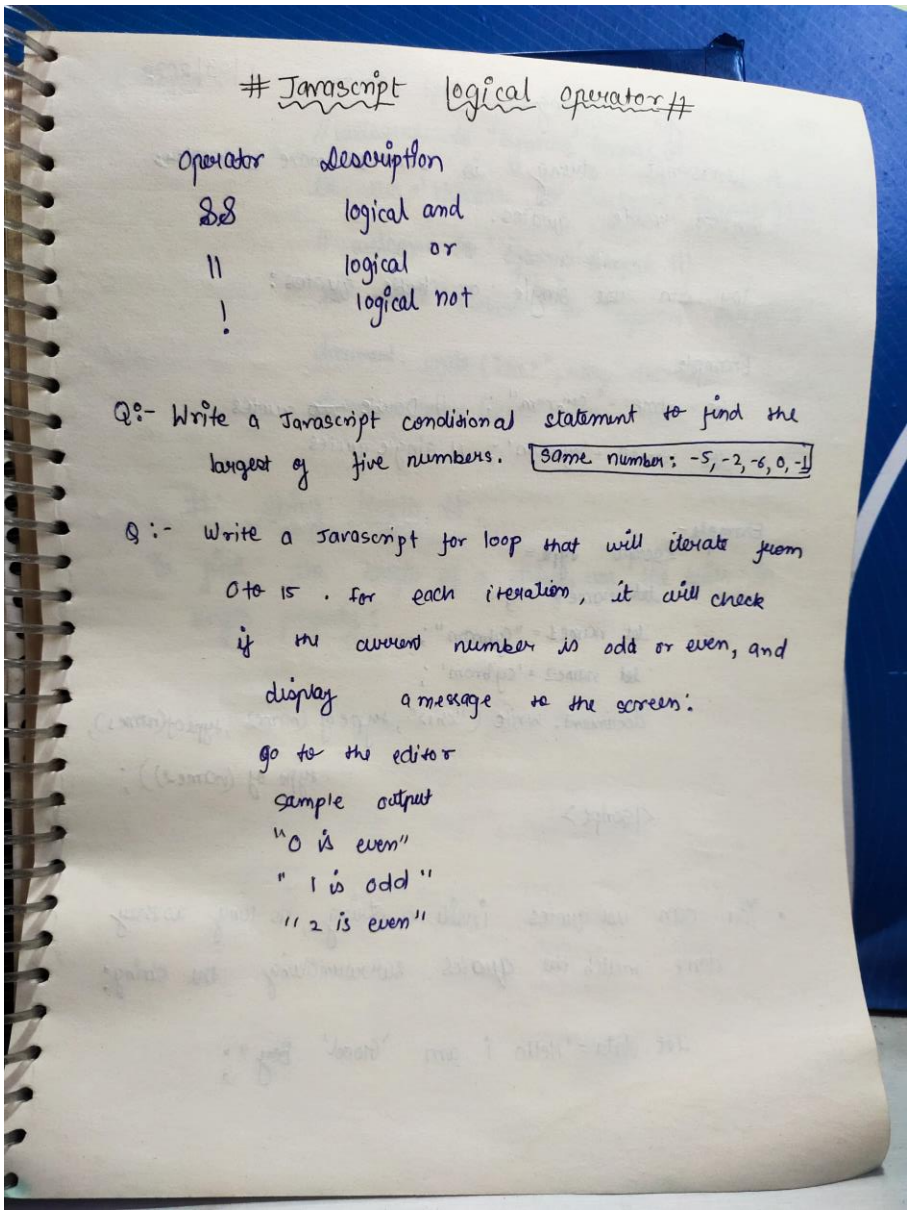
```
let name = "cybrom" ;  // Double quotes
let name2 = 'cybrom' ; // single quotes
```

Example-

```
<script type=
let name = " ";
let name1 = "cybrom";
let name2 = 'cybrom';
document. write ("<h1>", type of (name) , type of(name1),
                 type of (name2)) ;
</script>
```

- You can use quotes inside a string, as long as they don't match the quotes surrounding the string:

```
let data = "Hello i am 'Good' Boy";
```



Javascript logical operator

operator	description
&&	logical and
||	logical or
!	logical not

Q:- Write a Javascript conditional statement to find the largest of five numbers. Same number: -5, -2, -6, 0, -1

Q:- Write a Javascript for loop that will iterate from 0 to 15. For each iteration, it will check if the current number is odd or even, and display a message to the screen.

go to the editor

Sample output

"0 is even"

" 1 is odd "

" 2 is even"

Example - 4.

```
<script
    let a=10
    let b=10;
    if (a==b)
    {
    document.write ("both are equal!!!")
    }
    else
    {
    document.write ("<h1> are not equal
                                </h1>");
    }
</script>
```

Example :-

```
<script
    let a=10;
    let b="10";
    document.write ("<h1> a :", typeof a," b:",
                        typeof(b));
    if (a==b)
    {
     document.write ("<h1>both are equal!!! </h1>")
    }
    else
    {
     document.write ("<h1> both are not equal!!!")
    }
</script>
```

Example –

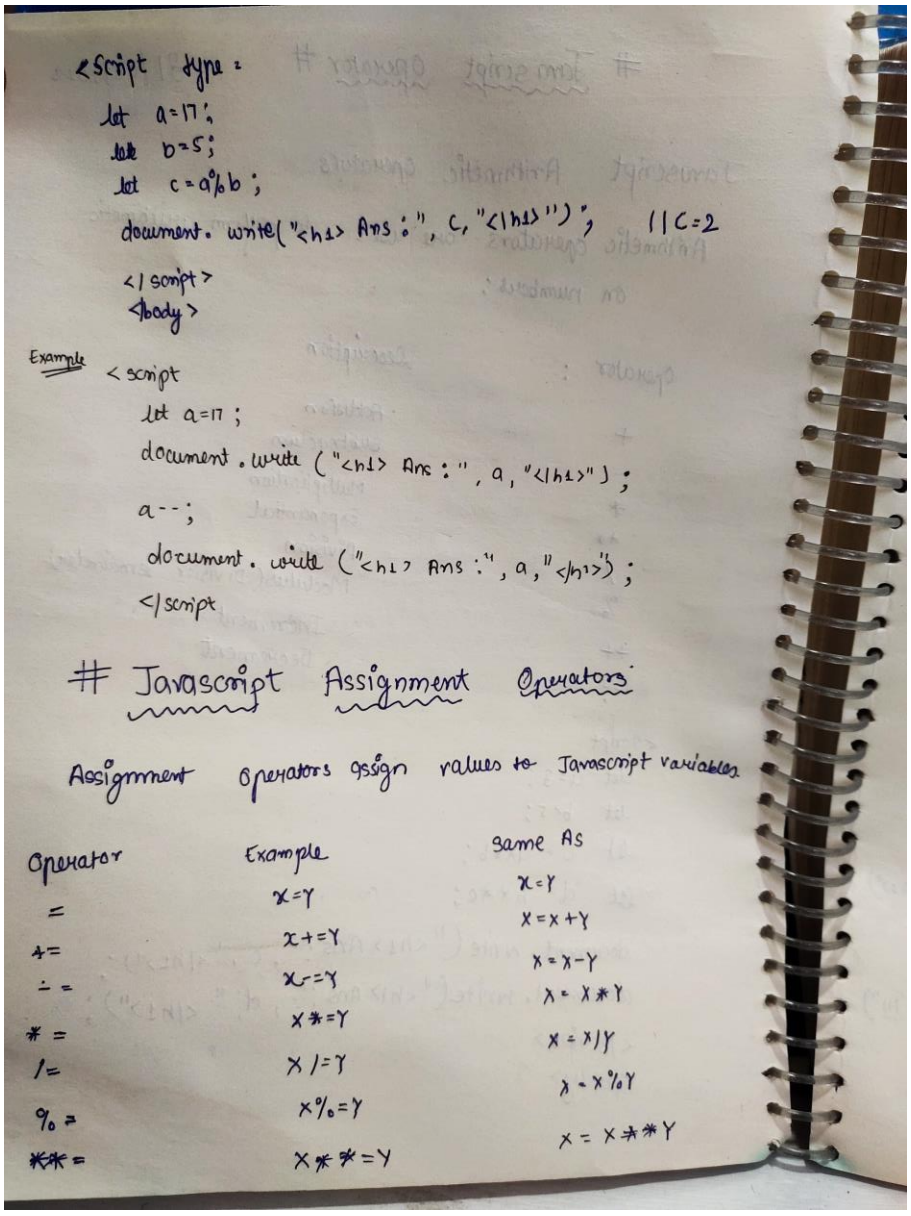
```
<script
    let a=5
    let b=7
    a+=b;      // a = a+b
    document.write("<h1> Ans : ", a, "</h1>");
</script>
```

Example

```
<script
    let a=5
    let b=7
    a**=b;     // a = a**b
    document.write ( "<h1> Ans : "; a , "</h1>");
    </script>
```

Javascript Comparison operators

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



```
<Script type =
  let a=17;
  let b=5;
  let c=a%b;
  document. write("<h1> Ans :", c, "</h1>");    // c=2
</ script>
<body>
```

Example

```
< script
  let a=17;
  document. write ("<h1> Ans :", a, "</h1>");
  a--;
  document. write ("<h1> Ans :", a, "</h1>");
</ script
```

Javascript Assignment Operators

Assignment operators assign values to Javascript variables

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

Javascript operator # 31/08/2022

Javascript Arithmetic operators

Arithmetic operators are used to perform arithmetic on numbers:

operator :	Description
+	Addition
-	Subtraction
*	Multiplication
**	Exponential
/	Division
%	Modulus (Division remainder)
++	Increment
--	Decrement

```
<script
  let a=3;
  let b=5;
  let c= a**b;
  let d= b**c;
  document.write("<h1> Ans :", c, "</h1>");
  document.write("<h1> Ans :", d, "</h1>");
</script>
</body>
```

30/08/2022

The multiple else if statement

Use the else if statement to specify a new condition if the first condition is false.

Syntax:

```
if (condition1)
  {
    //
  }
else if (condition2)
  {
    //
  }
```

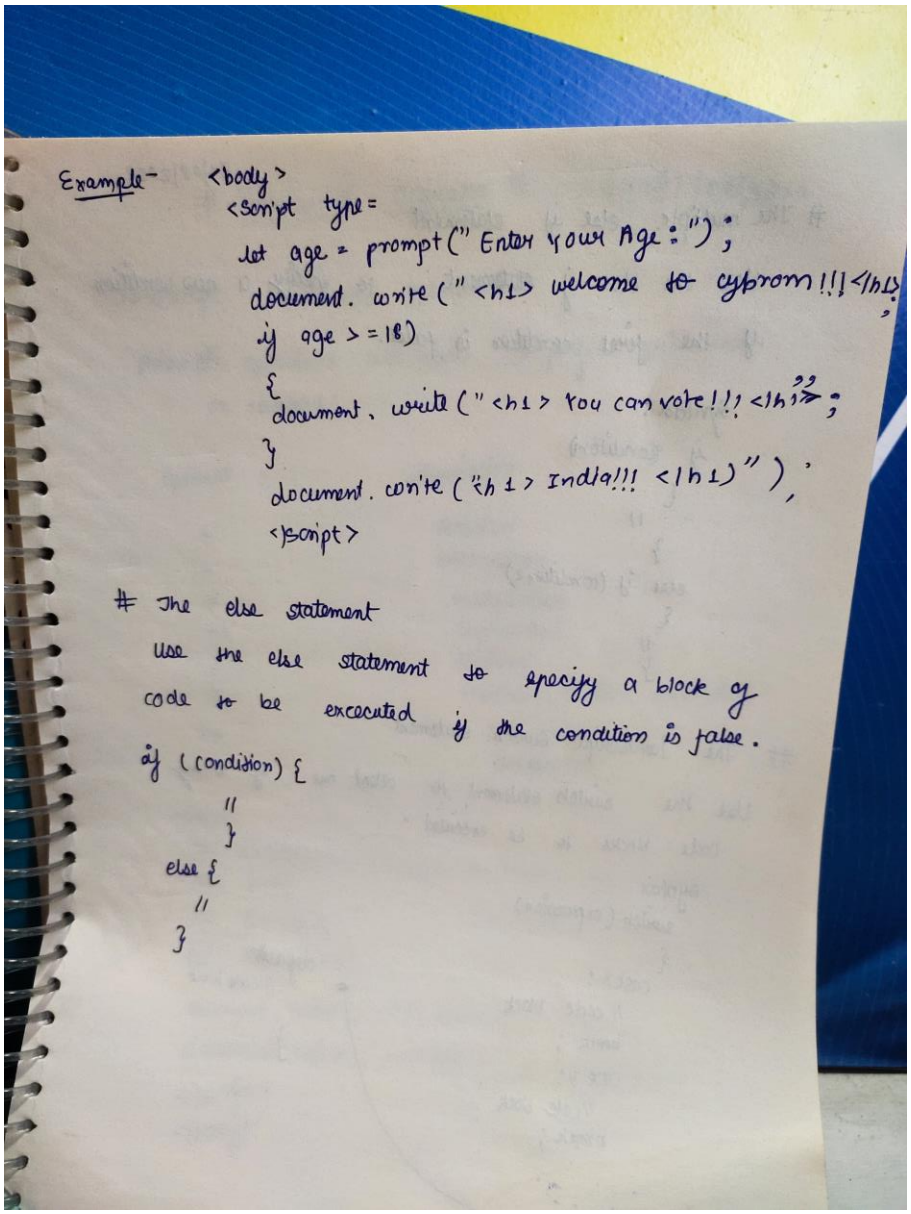
The javascript switch statement

Use the switch statement to select one of many code blocks to be executed.

Syntax

```
switch (expression)
{
  case x:
    // code block
    break;
  case y:
    // code block
    break;
  . . . .
  default:
    // code block
}
```

default
// code block
}



Example-

```
<body>
<script type=
let age = prompt("Enter your Age:");
document.write("<h1> welcome to cybrom!!!</h1>");
if age >= 18)
{
document.write("<h1> You can vote!!!</h1>");
}
document.write("<h1> India!!! </h1>");
</script>
```

The else statement

Use the else statement to specify a block of code to be executed if the condition is false.

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

Conditional statements

Very often when you write code, you want to perform different action for different decisions.

You can use conditional statement in your code to do this.

In Javascript we have the following conditional statement

1. Simple if statement.
2. if else statement.
3. ~~Peter~~ multiple if else statement.
4. Switch statement.

The if statement

Use the if statement to specify a block of Javascript code to be executed if a condition is true.

Syntax.

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

→ Using splice () to remove elements

With clever parameter setting ; you can use splice () to remove elements without leaving "holes" in the array:

codes - same as previous
but name. splice (0,1)

Java script Array slice

The slice () method slices out a piece of an array into a new array.

```
{
const name = ["raju", "sanju", "rohan", "Pradeep"];
const myname = name . slice(2);
document. getElementById ('demo'). innerHTML = name ;
document. getElementById ('demo1') . innerHTML = myname ;
}
```

another Example

```
{
Const name = [ "raju", "sanju", "rohan", "Pradeep",
               "ranju", "anju", "manju" ];
const myname = name . slice (2,4);
document. getElementById ('demo'). innerHTML = name
          "                ('demo1').     "     = myname
}
```

splicing and slicing Arrays

The splice() method adds new item in an array.

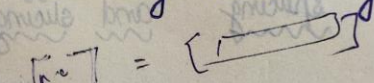
The slice() method slices out a piece of an array

Javascript Array splice()

The splice() method can be used to add new items in an array.

```
<script
function Display()
{
const name = ["raju", "sanju", "rohan", "Pradeep"];
document.getElementById('demo').innerHTML = name;
name.splice(2, 0, "sonu", "monu");
                 overlap or not
          index number where we want to add
document.getElementById('demo1').innerHTML = name;
}
</script
<body>
<h1 id="demo"> Welcome to aybrom </h1>
<h1 id="demo1"> we are developer </h1>
<h2 id="demo2"> learn Javascript !! </h2>
<h1 id="demo3"> web development </h1>
<button onclick="Display();"> click here !! </button>
</body>
```

* The concat() method can take any number of array arguments:



```
const name = ["raju", "sanju", "rohan", "Pradeep"]
const fname = ["mohit", "seema", "deepika"]
const sname = ["sandeep", "Ravi", "sujan"];

document.getElementById('demo').innerHTML = name;
document.getElementById('demo1').innerHTML = fname;
document.getElementById('demo2').innerHTML = sname;
const friends = name.concat(fname, sname);
document.getElementById('demo3').innerHTML = friends;
}
</script>
</head>
<body>
<h1 id='demo'> Welcome to cybrom </h1>
<h1 id="demo1"> We are developer </h1>
<h1 id='demo2"> Learn Java script!!! </h1>
<h1 id="demo3"> Web development </h1>
<button onclick="display();"> click </button>
```

Javascript Array delete()

Array elements can be delete using the Javascript operator delete.

Previous syntax:

```
delete name[2];
```

Merging (concatenating) Arrays

The concat() method creates a new array by merging (concatenating) existing arrays.

```
function display ()
{
  const name = ["raju", "sanju", "rohan", "Pradeep"];
  const fname = ["mohit", "seema", "deepika"];
  document.getElementById("demo").innerHTML = name;
  document.getElementById("demo1").innerHTML = fname;
  const friends = name.concat(fname);
  document.getElementById('demo2').innerHTML = friends;
}
</script>
<body>
<h1 id="demo"> welcome to cybrom </h1>
<h1 id="demo1"> We Deep</h1>
<h1 id="demo2"> Patel</h1>
```

Shifting Elements

shifting is equivalent to popping, but working on the first element instead of the last.

Javascript Array shift() → no argument

The shift() method removes the first Array element and "shifts" all other element to a lower index.

Javascript Array Unshift() → argument pass

The unshift() method adds a new elements to an array (at the beginning), and "unshifts" older elements.

Previous syntax:-

```
name.unshift("Manju");
```

Changing Elements

Array elements are accessed using their index number:

```
name[1] = "Mayank";
```

Previous syntax:-

Javascript Array push( ) argument

The push() method adds a new element to an array (at the end):

Example :-

```
function display ()
{
 const  name = ["raju", "sanju", "rohan", "Pradeep"];
document . getElement By Id ('demo
```

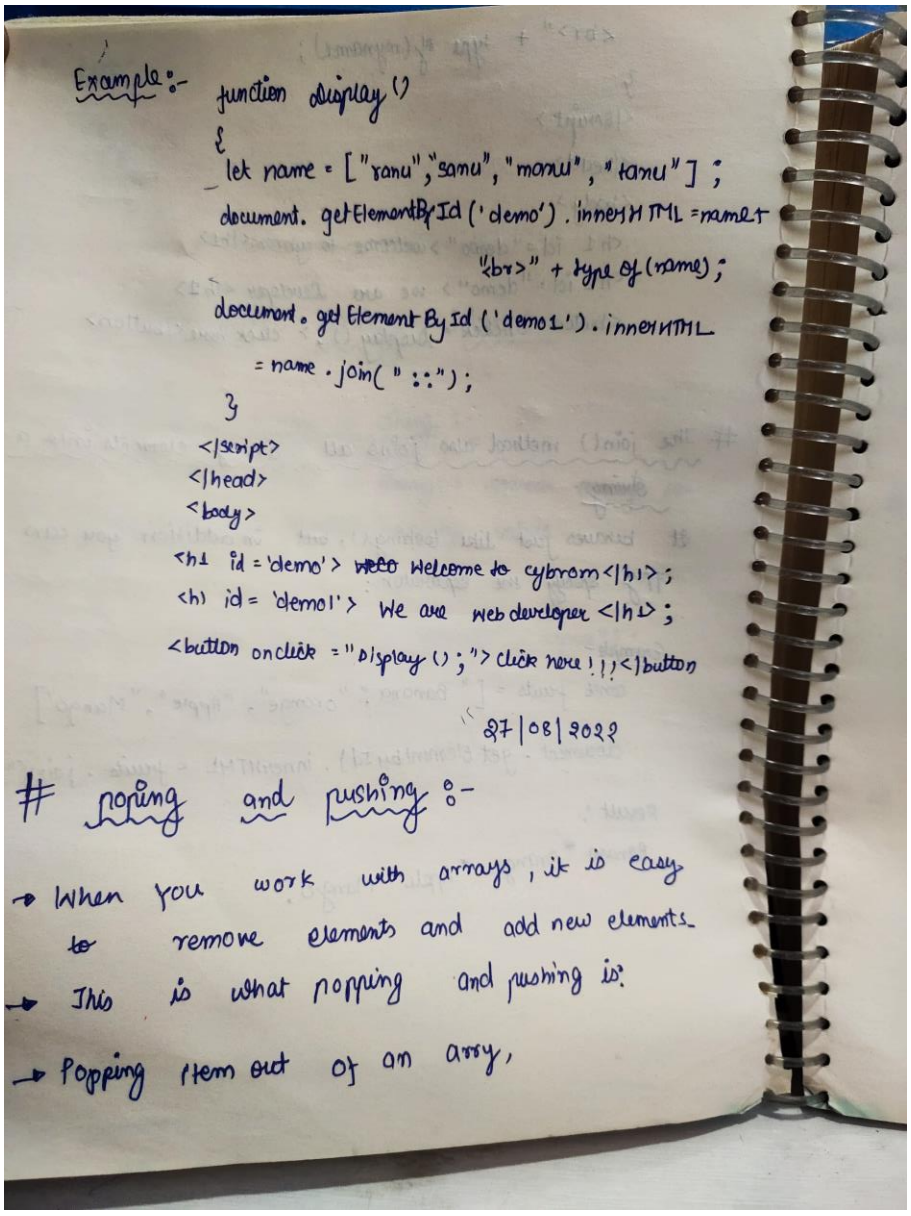
Javascript Array pop()

The pop() method removes the last element from array

Example:-

```
<script type
function display()
{
 const name = ["raju", "sanju", "rohan",
                "pradeep"];
 document.getElementById('demo').innerHTML=name;
 name.pop()
 document.getElementById('demo1').innerHTML=name;
}
</script>

<body>
<h1 id='demo"> Welcome to cybrom</h1>
<h1 id="demo1" > we are developer</h1>
<button onclick="display();"
```



Example :-

```
function display()
{
let name = ["ranu","sanu","manu","tanu"];
document.getElementById('demo').innerHTML=name+
                    "<br>" + typeof(name);
document.getElementById('demo1').innerHTML
    = name.join("::");
}
</script>
</head>
<body>
<h1 id='demo'> Welcome to cybrom</h1>;
<h1 id='demo1'> We are web developer</h1>;
<button onclick="display();"> Click here!!!</button>
```

27|08|2022

Popping and pushing :-

→ When you work with arrays, it is easy to remove elements and add new elements.

→ This is what popping and pushing is.

→ Popping item out of an array,

```
<br>" + typeof(myname);
}
</script>
<head>
<body>
<h1 id="demo">welcome to cybrom</h1>
<h1 id="demo">we are Developer</h1>
<button onclick="Display();">click here</button>
```

The join() method also joins all array elements into a string.

It behaves just like tostring(), but in addition you can ~~apply~~ specify the separator:

Example-

```
const fruits = ["Banana", "orange", "Apple", "Mango"];
document.getElementById).innerHTML = fruits.join("*");
```

Result:

Banana * Orange * Apple * Mango.

Q. Print :

1
1 2
1 2 3
1 2 3 4

x x x

Javascript Array Method

Converting Arrays to string :-

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

Exp:-

```
const name = ["ram", "mohan", "sanjay", "sapna"];
Const txt = name.toString();
document.getElementById('data').innerHTML = txt;
```

Example:-

```
<script type="text/
function display()
{
let name = ["ranu", "sanu", "manu", "tanu"];
document.getElementById('demo').innerHTML = name +
        "<br>" + type of(name);
let myname = name.toString();
document.getElementById('demo1').innerHTML = myname+
```

```
<script
document.write("<h2> While : </h2>");
var a = 1;
while (a >= 10)
{
document.write (a + "<br>");
a = a+1;
}
document.write ("<h2> Do while : </h2>");
var b = 10;
}
while (b >= 10);
</script>
```

ऊपर वाला नहीं चलेगा

लेकिन do while condition गलत होने पर भी 1 बार चलेगा।

Assignment :- WAJSP to print fibbonaci series up to given no. of term

enter digit : 7

Output

0 1 1 2 3 5 8

- WAP to print factorial of any given number
- WAP to print table of any given number.
- Print :

```
* * * *
* * *
* * *
* *
*
```

→ # Java script While loop

loops can execute a block of code as long as a specified condition is true.

The while loop

syntax:-

```
while(condition)
 {
   // code block to be executed
 }
```

⟹ # The Do while loop

The do while loop is a variant of the while loop. This loop will ~~exp~~ execute the code block once, before checking if the condition is true, then it will repeat the loop as long as the condition is true.

syntax:-

```
do
{
  // code block
}
while (condition)
```

Javascript for of

The Javascript for of statement loop through the values of an iterable object.

→ It lets you loop over iterable data structure such as Arrays, strings, maps, Nodelists, and more :

Syntax :-

```
for (variable of iterable)
{
  // code block to be executed
}
```

variable - For every iteration the value of the next property is assigned to the variable. Variable can be declared with const, let, or var.

Iterable - An object that has iterables properties.

```
<script
name = ["sanjit",
let student = {
rollno : 120,
name
```

```
<script
let student = "welcome to cybrom";
for (a of student)
{
document.write("Name : "+a+"<br>");
}
</script>
</body>
</html>
```

Javascript for In

The for In loop

The javascript for in statement loops through the properties of an object:

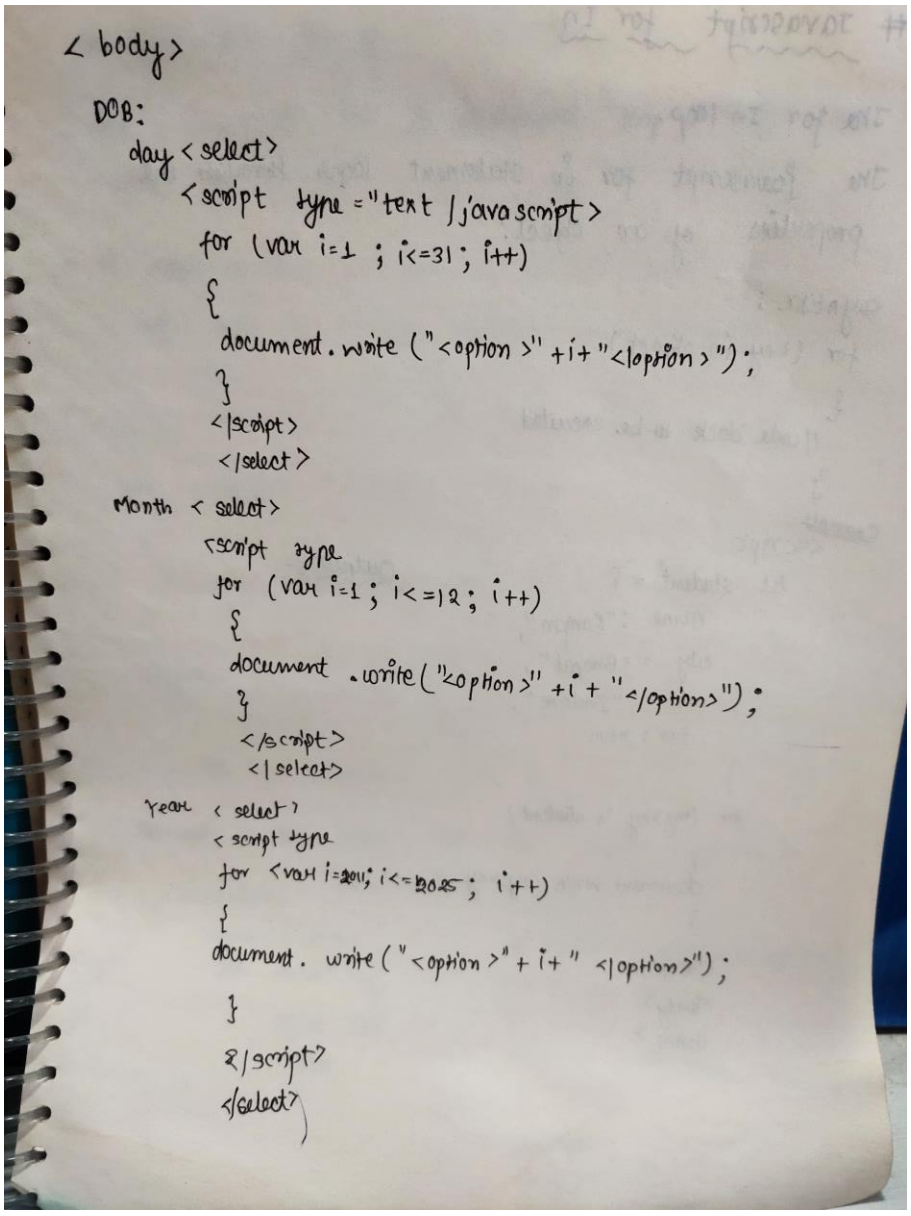
Syntax :--

```
for (key in object)
{
 //code block to be executed
}
```

Example

```
<script
  let student = {
        name :"Raman",
        city = "Bhopal",
        add = "Indore",
        fees : 3670
   }
  for (my key in student)
     {
     document.write (my key, "<br>");
     }
  </script>
  <body>
  <html>
```

Output :-



```
< body >

DOB:
  day < select >
      < script type = "text / javascript >
       for (var i=1 ; i<=31 ; i++)
       {
        document . write ("<option >" +i+ "<|option >");
       }
       <|script>
       <|select>
  month < select >
       <script type
       for (var i=1 ; i<=12 ; i++)
        {
        document . write ("<option >" +i + "</option>");
        }
        </script>
        <|select>
  Year < select >
       < script type
       for <var i=2011; i<=2025 ; i++)
       {
       document . write ("<option >" + i+ " <|option>");
        }
       &|script>
       </select>
```

The for loop

The for loop has the following syntax:

```
for (statement 1 ; statement 2 ; statement 3)
 //
 {
 }
```

Example :-

```
<script type = " text/javascript ">
for (var i=1 , i<=10; i++)
{
document.write (i+ " <br> ");
}
</script>
```

Example 2:-

```
<script type = "text/javascript">
var myno = prompt ("Enter any no: ");
var ans;
for (var i=1 ; i<=10; i++)
{
  ans = myno * i;
  document. write (myno, " * ", i, " = ", ans "<br>");
}
</script>
```

25/08/2022

Javascript loops

→ loops can execute a block of code a number of time.

→ loops are handly, if you want to run the same code over and over again, each time with a different value.

→ often this is the case when working with arrays:

Different kinds of loops.

Javascripts supports different kinds of loops:

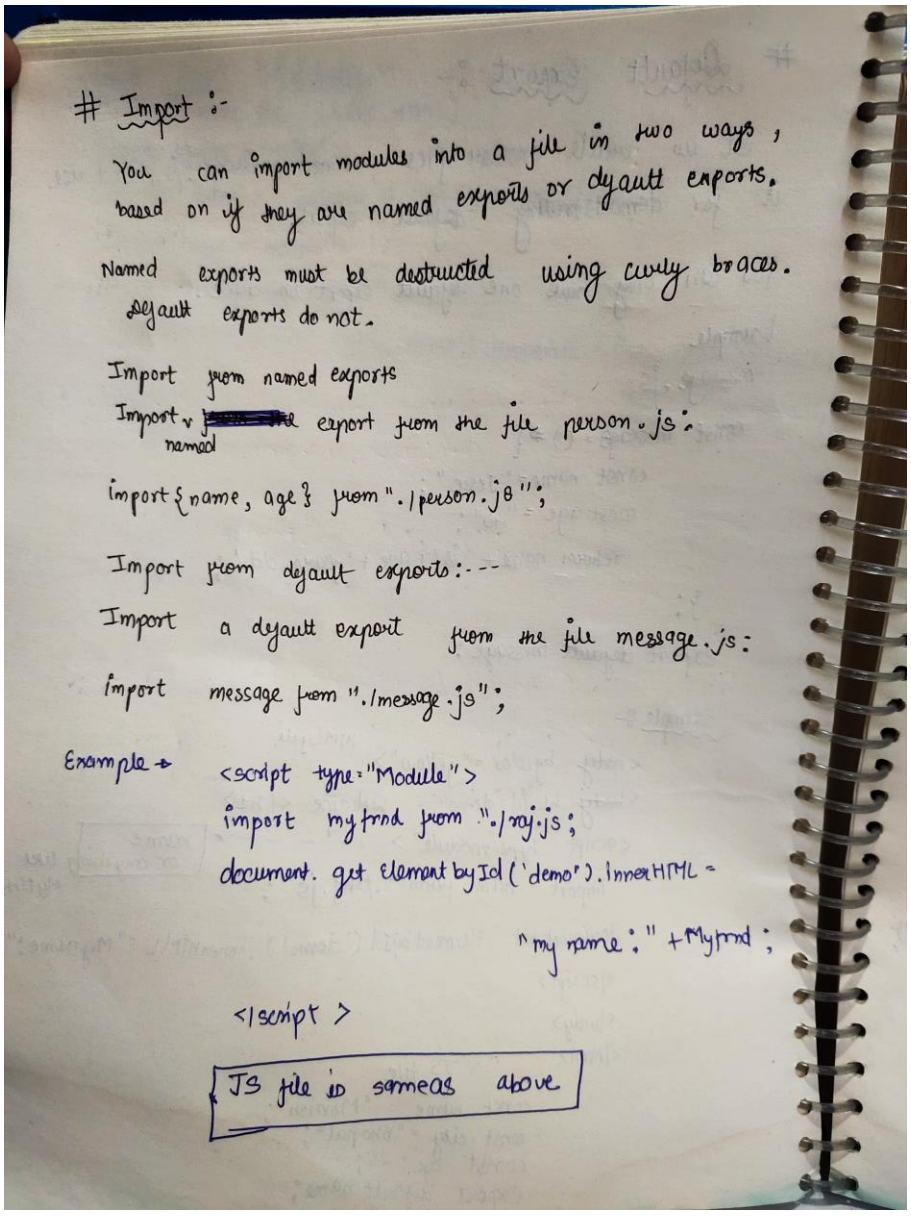
for - loops through a block of code a number of times.

for/in - loops through the properties of an object.

for/of - loops through the values of an iterable object.

while - loops through a block of code while a specified condition is true.

do/while - also loops through a block of code while a specified condition is true.



Import :-

You can import modules into a file in two ways, based on if they are named exports or dyault exports.

Named exports must be destructed using curly braces. dyault exports do not.

Import from named exports

Import named export from the file person.js:

```
import {name, age} from "./person.js";
```

Import from dyault exports:---

Import a dyault export from the file message.js:

```
import message from "./message.js";
```

Example →

```
<script type="Module">
import myfrnd from "./raj.js";
document.getElementById('demo').innerHTML =
                "my name:" +Myfrnd;
</script>
```

JS file is sameas above

Default Export :-

let us create another files, named message.js, and use it for demonstrating default export.

You can only have one default export in a file.

Example

message.js

```
const message = () => {
    const name = "Jesse";
    const age = "40";
    return name + " is " + age + " years old.";
};
export default message;
```

Example :-

HTML file

```
<body bgcolor = "yellow">
<body> <h1 id = "demo"> Welcome </h1>
<script type = "module" >
  import name from "./raj.js";
  document.getElementById('demo').innerHTML = "My name :" + name;
</script>
</body>
</html>
```

name or anything like My Nm

JS file

```
const name = "Manish";
const city = "Bhopal";
const age = 25;
export default name;
```

named export individually -

HTML file (index.HTML)

```
<body bgcolor = "yellow">
<h1 id = "demo"> welcome to cybrom Bhopal </h1>
<script type = " module ">
import {name, age } from "./raj.js";
document.get element byId('demo').innerHTML = " My name" +name+
                                              "Mage: " +age;
</script>
</body>
</html>
```

JS file (raj.js)

```
export const name = " sourabh";
export const age = " 24 ";
```

Export all at the one

HTML file is same as upper

JS file (raj.js)

```
const name = " sourabh";
const age = 22;
export {name, age};
```

```
</script>
    <h1 id="demo"> welcome to cybrom Bhopal </h1>

</body>
</html>
```

JS file -

```
const name = "Mohnish";
const age = 23;
const city = "Bhopal";

export {name, age};
```

24/08/2022

Named Exports

You can create named exports two ways. In-line individually or all at once at the bottom.

In-line individually :

```
Person.js
  export const name = "Jesse"
  export const age = "40"
```

All at once at the bottom :

```
person.js
Const name = "Jesse"
Const age = "40"
export {name, age}
```

ES6 Module

XAMP
apache web server
+
htdocs
+

Modules :-

- Javascript modules allow you to break up your code into separate files.
- This makes it easier to maintain the code-base.
- ES6 Modules rely on the import and export statements.

Export :

You can export a function or variable from any file.

There are two types of exports :

1. Named and
2. Default.

Named Exports

You can create named exports two ways. In-line individually, or all at once at the bottom.

Example :-

```
<!head>
< body bgcolor = "yellow">
< script type = "module">
import { name, age } from "./raj.js";
document.getElementById('demo').innerHTML = "My name"
+name+"My age" + age + "!";
```

If you have parameters, you pass them inside the parentheses:

Arrow functions with Parameters:

```
hello = (val) => "Hello" + val;
```

Infact, if you have only one parameter, you can skip the parenthesis as well

Arrow function without Parentheses:

```
hello = val => "Hello" + val;
```

Example

```
function Display()
{
cybrom = a => a* a*a ;
document.getElementById('demo').innerHTML = "addition:" +cybrom
                                              (5);
}
```

It gets shorter if the function has only one statement and the statement returns a value, you can remove the brackets and the return keyword:

Arrow function Return value by Default:

```
hello = () => "Hello World";
```

Note: This works only if the function have only one statement.

Example-

```
<script
function display ()
{
  const cybrom = (a, b) => a+b;
  document.getElementById('demo').innerHTML
    = "addition:" + cybrom(40,40);
}
</script>
```

Example.

```
const cybrom=() => "we are web developer!!";
```

ES6

Ex1 With Arrow function :

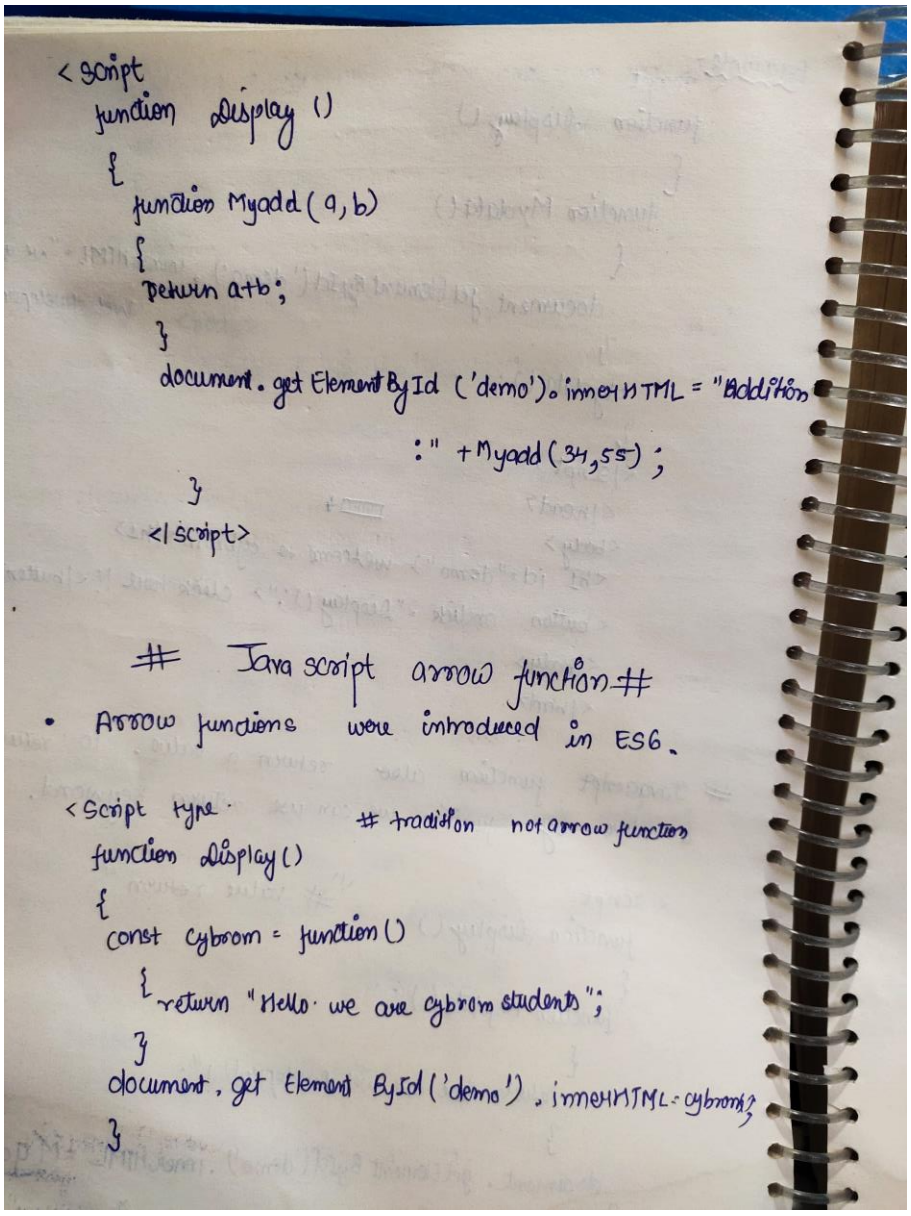
```
hello =() =>{
    return "Hello world";
}
```

Example 2 :-

```
<script
function Display()
{
const cybrom =()=>
 {
  return "we are web designer !!! ";
 }
document.getElementById('demo').innerHTML
                                =cybrom();
}
</script>
<body>
  ''''
</body>
```

Example 3 :-

```
<script
function Display()
{
 const cybrom=(a,b)=>
  {
   return a+b;
  }
document.getElementById('demo').innerHTML =
                                  ~~cybrom~~()
"addition"+cybrom(3,5);
```



```
<script
  function display ()
  {
    function Myadd (a, b)
    {
     return a+b;
    }
    document.getElementById ('demo').innerHTML = "Addition
                                  :" + Myadd (34, 55);
  }
</script>
```

Java script arrow function

- Arrow functions were introduced in ES6.

```
<script type·                    # tradition  not arrow function
  function display()
  {
   const cybroom = function()
    {return "Hello· we are cybrom students";
    }
   document.getElementById('demo').innerHTML=cybroom();
  }
```

Example-

```
<script
    function display()
    {
      function Mydata()
        {
          document.getElementById('demo').innerHTML = "we are
                                                    web developer!!!";
        }
      Mydata();
    }
</script>
<head>
<body>
<h1 id="demo"> welcome to cybrom </h1>
<button onclick="display();"> click here !! </button>
</body>
</html>
```

Javascript function also return a value. to return a value by function we can use return keyword.

```
<script
    function display()
    {
      function Mydata()
        {
          return "we are web developer!!!";
        }
      document.getElementById('demo').innerHTML = Mydata();
    }
</script>
```

"# value return".

```
<script>
  let x = this ;
  document. write (x)
</script>
```

Example:-

```
<body>
  <marquee onmouseover = "this. stop();"
           onmouseout = "this. start();">
    <h1> welcome to Cyprom Bhopal </h1>
  </marquee>
</body>
```

19/08/2020

Function Syntax :-

Javascript function :-

- A Javascript function is a block of code designed to perform a particular task.
- A Javascript function is executed when "something" invokes it (calls it)

Synatax :

```
function demo (para1, para2, ...)
{
  // ...
}
```

The values are written as ~~he~~ name : value pairs (name and value separated by a colon.

Accessing object properties :---

You can access object properties in two ways :

objectName . propertyName
or
Object Name ["propertyName"]

Javascript this keyword :---

this in a Method

~~In an object method, this refers to the owner.~~

What is this ?

The Javascript this keyword refers to the object it belongs to.

this Alone.

- When used Alone, the owner is the Global object, so this refers to the Gibbal object.
- In browser window the Gibbal object is [object window]:

Example -

```
<p id = "demo" ></p>
<script>
let x = this;
document. ~~write~~ get element by I d
```

```
document.write("<br>")
document.write("Name:", student["Name"]); //sachin
document.write("<br>")
document.write("fees:", student["fees"]); //35780
</script
```

other :-

```
<script type="text/javascript">
const student = {
    rollno :120,
    name :"sachin",
    myclass :"BCA",
    fees :30000
    };
    document.write("Roll number :", student.rollno); //120
    document.write("<br>");
    document.write("Name :", student.name); //sachin
    document.write("<br>");
    document.write("fees :", student.fees);
```

Example -

```
<script type="text/javascript">
    let a = 30;
    document.write(a);
    document.write("<br>");
    document.write(typeof(a));
    document.write("<br>");
    const student = {
        rollno: 120,
        name: "Sachin",
        add: "Bhopal",
        fees: 34900
        };
        document.write(typeof(student));
        </script>
```

Example :-

```
<script type "text/javascript">
const student = {
    roll no: 120,
    name: "Sachin",
    my class: "BCA",
    fees: 35780
};

document.write("Roll number:", student["roll no"]); //120
```

Javascript Output :- 18/08/2022

→ Javascript Display Possibilities :-

→ Javavascript can "display" data in different ways:

Writing into an HTML element, using innerHTML.
Writing into the HTML output using document.write().
Writing into an alert box, using window.alert().
Writing into the browser console, using console.log().

ctrl+shift+i → mostly wide used to detect error. because javascript run on browser. It has no built in.

Javascript objects :-

Javascript objects

- We know that Javascript variables are containers for data values.
- This code assigns a simple value (cybrom) to a variable named nm.

```
let nm = "cybrom";
```

objects are variables too. But objects can contain many values.

This code assigns many values (fiat, 500, white) to a variable named car:

```
const car = {type: "fiat", model: "500", color: "white"};
```

JS Object

output

Using the Javascript Keyword new :-

```
<p id="demo"></p>
<script>
const cars = new Array ("saab", "volvo", "BMW");
document.getElementById("demo").innerHTML = cars;
</script>
```

Example-

```
<script type="text/javascript">
function display()
{
const sub= new Array ("PHP", "oracle", "Asp", "css",
                      "HTML");
document.getElementById('demo').innerHTML = "first val:"
                +sub[0]+ "last value", +sub[4]+, "/";
}
```

Example-

```
<script type="text/javascript">
function display()
{
const sub=[];
sub[0]="oracle";
sub[1]="java";
sub[2]="php";
sub[3]="html";
document.getElementById("demo").innerHTML=sub;
}
</script>
<head>
<body>
<p id="demo">My data</p>
<button onclick="display();">click here!!</button>
</body>
</html>
```

```
<script
  function display()
  {
  const sub = ["oracle", "java", "php", "html"];
  document.getElementById('demo').innerHTML = sub;
  }
  </script>
  </head>
  <body>
  <p id="demo">Mydata</p>
  <button onclick="display();">click here!!!</button>
  </body>
  </html>
```

You can also create an array, and then provide the elements:

```
<p id="demo"></p>
<script>
  const cars = [];
  cars[0] = "saab";
  cars[1] = "volvo";
  cars[2] = "BMW";
  document.getElementById("demo").innerHTML = cars;
  </script>
```

—: Javascript Array :-

An array is a special variable, which store value.

→ Using an array literal is the easiest way to create a Javascript Array.

Syntax:

```
const array_name = [item1, item2, ....];
```

Example ①

```
<script type="text/javascript">
const name = ["raj", "mohan", "sanjay", "Pankaj"];
document.write(name[0]);    // raj
document.write("<br>");
document.write(name[3]);    // Pankaj → output
</script>
```

②

```
<script
const name = ["raj", "mohan", sanjay, 23, "Pankaj", 3.67];
document.write(name[0]);    # raj
document.write("<br>");
document.write(name[3]*3;   // 69
document.write("<br>");
document.write(name[5]+10);  //13.67
</script>
```

17/08/2022

Java Script Array

Number Methods

In the chapter Number Methods, you will find more methods that can be used to convert strings to number:

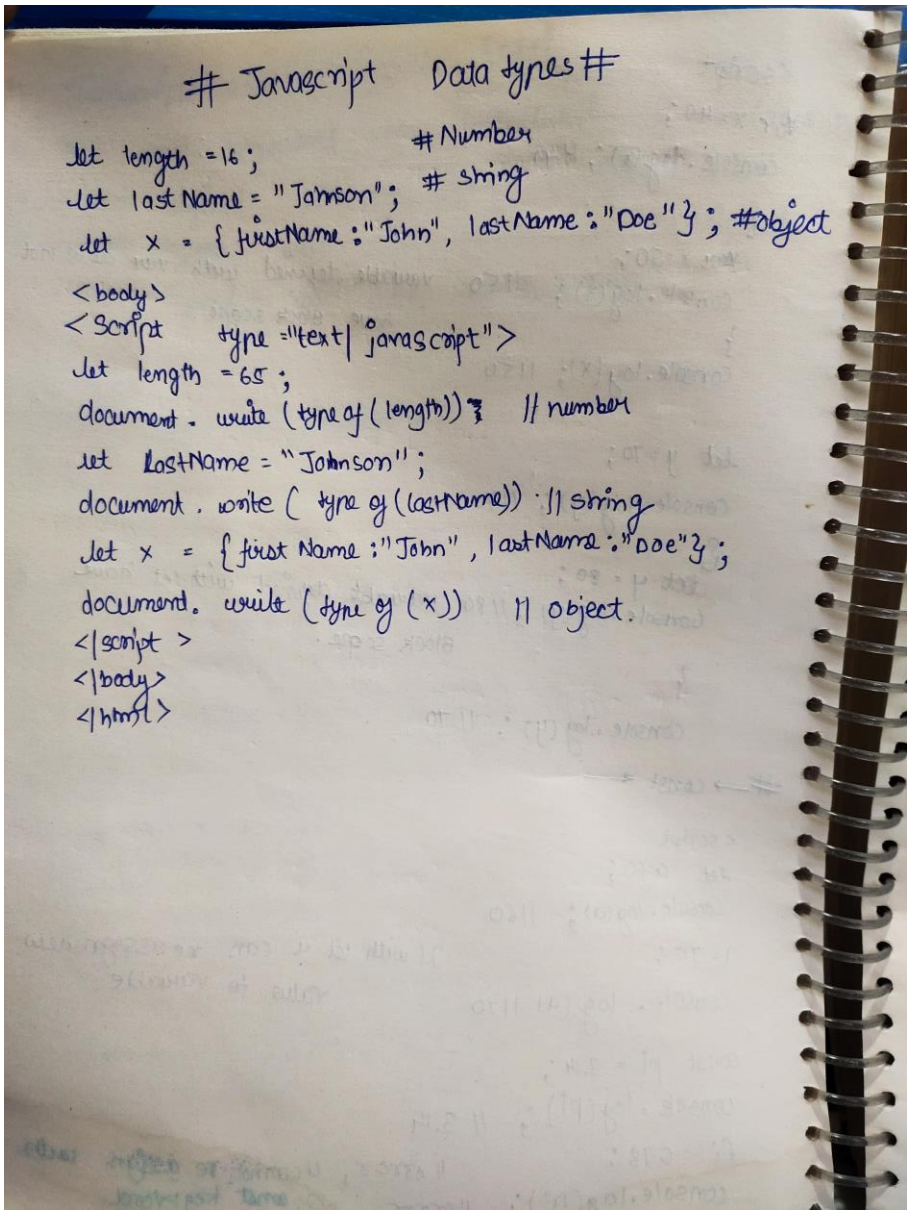
Method	Description
Number()	Returns a number, converted from its argument
parseFloat()	Parses a string and returns a floating point number
ParseInt()	

Converting Numbers to Strings

- The global method String() can convert numbers to string
- It can be used on any type of numbers, literals, variables, or expressions:

Example -

```
String(x)        // returns a string from a number variable
String(123)      // returns a string from a number literal 123
String(100+23)   // returns a string from a number from an expression.
```



Javascript Data types

```
let length = 16;                    # Number
let lastName = "Johnson";           # string
let x = {firstName:"John", lastName:"Doe"};  #object

<body>
<script type="text/javascript">
let length = 65;
document.write(typeof(length));  // number
let LastName = "Johnson";
document.write(typeof(lastName)); // string
let x = {firstName:"John", lastName:"Doe"};
document.write(typeof(x))  // object.
</script>
</body>
</html>
```

```
<script
var x=40;
console.log(x); //40
{
  var x=50;
  console.log(x); //50
}
console.log(x); //50

let y=70;
console.log(y); //70
{
  let y=80;
  console.log(y); //80
}
console.log(y); //70
```

variable defined with var does n have Block scope.

variable defined with let have Block scope.

→ const ←

```
<script
let a=60;
console.log(a); //60
a=70;
console.log(a) //70
const pi=3.14;
console.log(pi); //3.14
Pi:=6.78;
console.log(Pi); //error
</script>
```

// with let u can reassign new value to variable.

// error, u cannot re assign with const keyword

C+S+i

Javascript const

- The const keyword was introduced in ES6 (2015)
- Variables defined with const cannot be Redeclared.
- Variables defined with const cannot be Reassigned
- Variables defined with const have Block scope.

```
<body>
<script type="text/javascript">
  var a=30;
  var a=50;
  Console.log(a); // right
  let b=40;
  let b=70;   // wrong
  Console.log(b);
  </script>
  </body>
```

```
<body >
<script >
a=40;
var a;          // right
console.log(a);
b=50;
let b;          // wrong
console.log(b);
</script>
```

Javascript Variable

3 ways to Declare a Javascript Variable:

Using var.
Using let
Using const

When to Javascript Var?

Always declare Javascript variables with var, let or const.

The var keyword is used in all Javascript code from 1995 to 2015.

The let and const keywords were added to Javascript in 2015.

If you want your code to run in older browser, you must use hover.

Javascript let (ECMA)

- The let keyword was introduced in ES6 (2015).
- Variables defined with let cannot be Redeclared.
- Variables defined with let must be Declared before use.
- Variables defined with let have Block scope.

Example :-

```
<script type="text/javascript">
  function MyColor()
  {
   document.getElementById('demo').style.color="red";
  }
</script>
<head>
<body>
  <p id='demo'> ------ </p>
<button onclick="MyColor();"> Click here !!! </button>
</body>
</html>
```

6/08/2022

```
<script type="text/javascript">
 function
 {
 document.getElementById('nm').style.background
```

```
document.getElementById('tprice').value = mytotal;
document.getElementById('cgst').value = mycgst;
document.getElementById('sgst').value = mysgst;
document.getElementById('gst').value = mygst;
document.getElementById('nprice').value = mynetprice;
}
```

5/08/2022

changing HTML By style

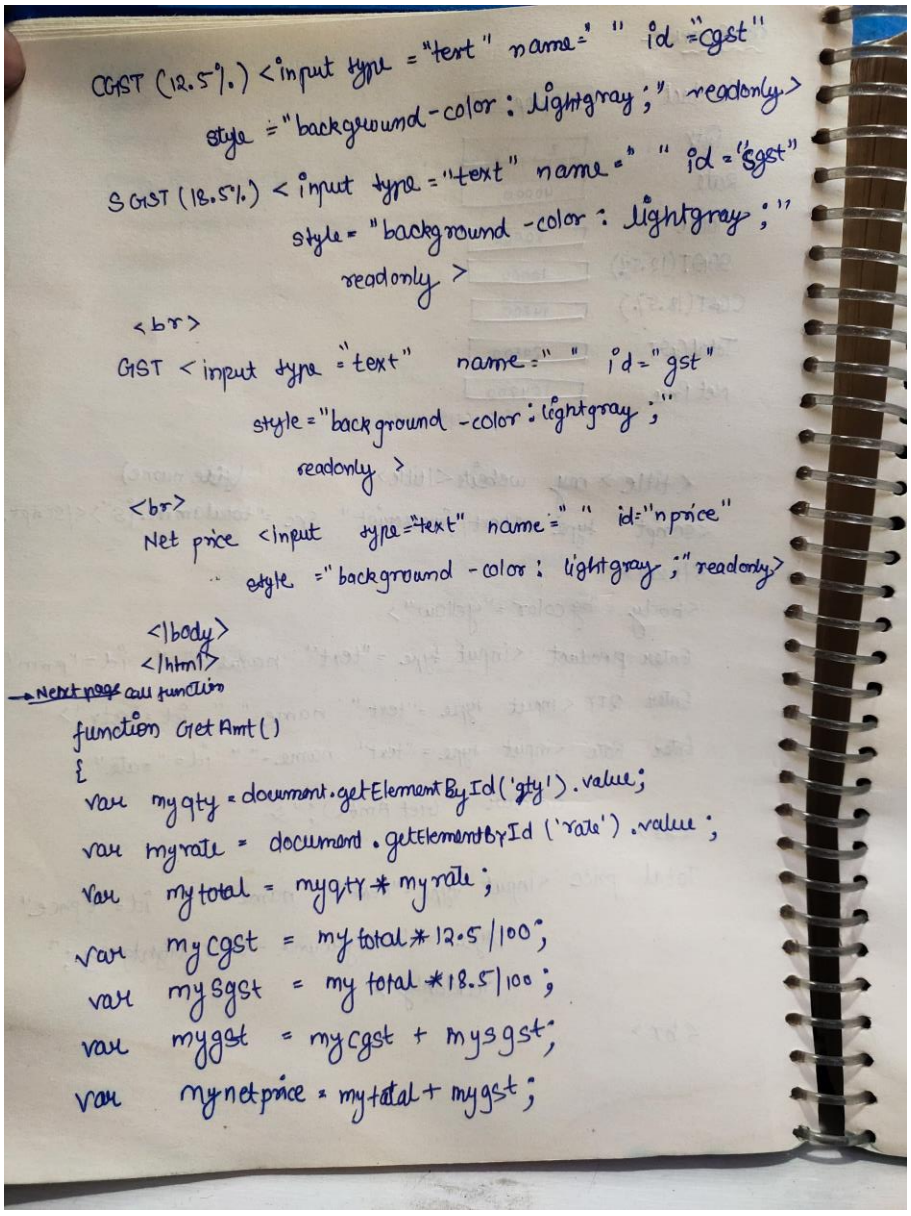
To change the style of an HTML element, use this syntax:

Syntax:--

```
document.getElementById(id).property = newStyle
```

Exp:--

```
<p id="p1">Hello World!</p>
<p id="p2">Hello world!</p>
<script>
document.getElementById("p2").style.color = "blue";
document.getElementById("p2").style.fontfamily = "Arial";
document.getElementById("p2").style.fontsize = "larger";
</script>
```



CGST (12.5%) <input type = "text" name = " " id = "cgst"
style = "background-color : lightgray ;" readonly>
SGST (18.5%) <input type = "text" name = " " id = "sgst"
style = "background-color : lightgray ;"
readonly >

GST <input type = "text" name = " " id = "gst"
style = "background-color : lightgray ;"
readonly >

Net price <input type = "text" name = " " id = "nprice"
style = "background-color : lightgray ;" readonly>
</body>
</html>

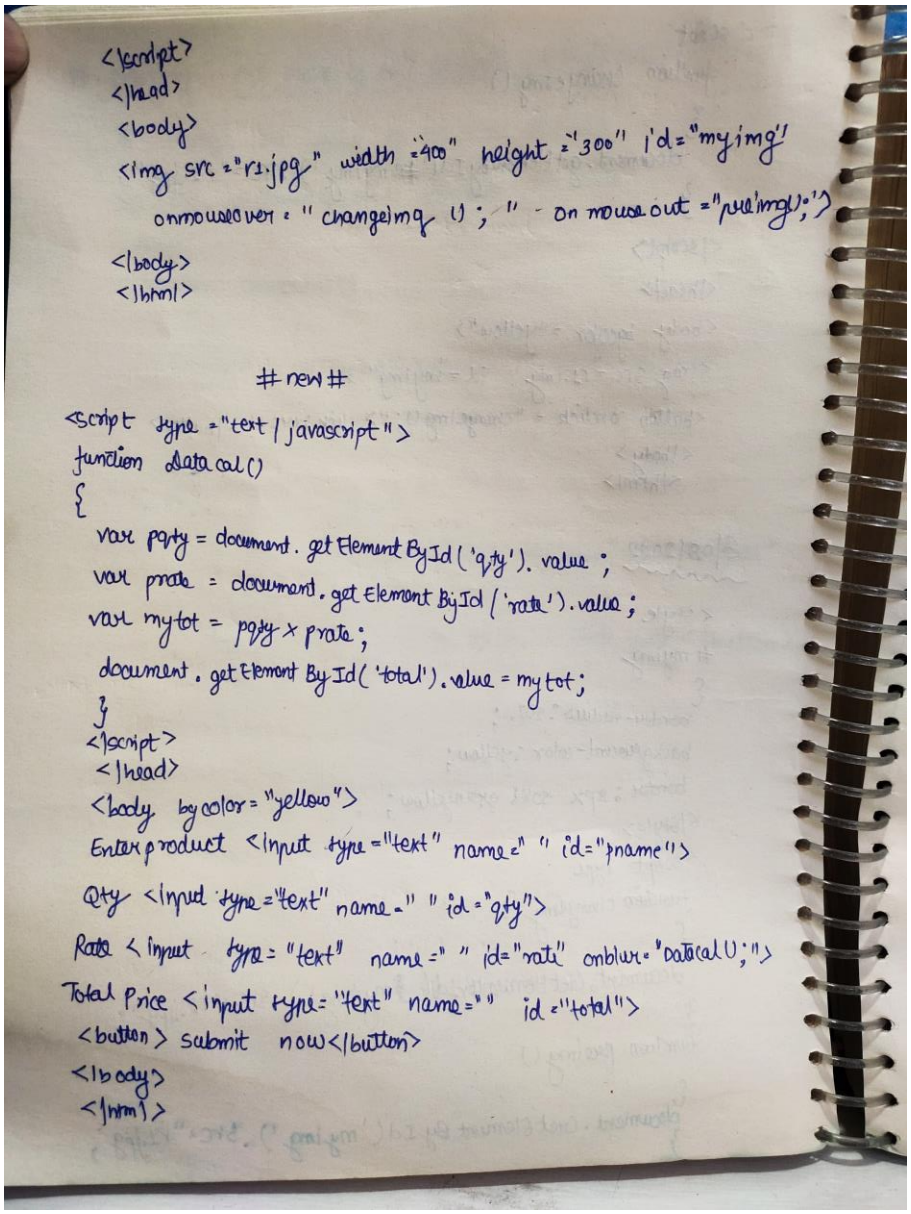
→ Next page call function

```
function GetAmt()
{
var myqty = document.getElementById('qty').value;
var myrate = document.getElementById('rate').value;
var mytotal = myqty * myrate;
var mycgst = mytotal * 12.5 / 100;
var mysgst = mytotal * 18.5 / 100;
var mygst = mycgst + mysgst;
var mynetprice = mytotal + mygst;
```

Homework :-

Product	Laptop
Qty	2
Rate	40000
Total	80000
SGST(12.5%)	10000
CGST(18.5%)	14800
TotalGST	24800
Net Price	104800

```
<title> my website</title>                    (file name)
<script type ="text/javascript" src ="totalamnt.js"></script>
</head>
<body bg color ="yellow">
Enter product <input type ="text" name =" " id ="pnm">
Enter QTY <input type ="text" name=" " id ="qty">
Enter Rate <input type = "text" name =" " id = "rate"
        on blur ="Get Amt();">
<br>
Total price <input type ="text" name =" " id ="tprice"
        style ="background-color: lightgray;"
        readonly >
<br>
```



```
</script>
</head>
<body>
<img src="r1.jpg" width="400" height="300" id="myimg"
onmouseover="changeimg();" onmouseout="previmg();">
</body>
</html>
```

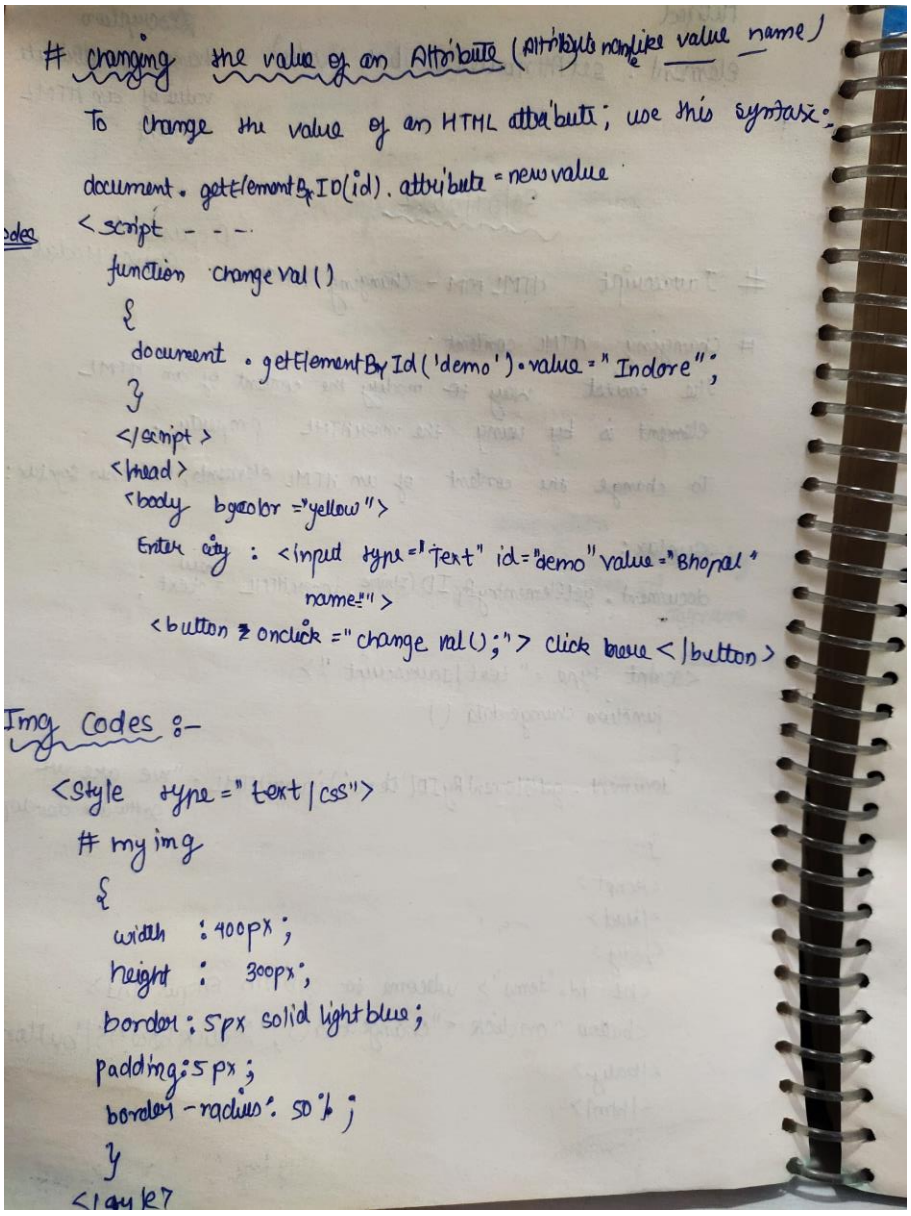
#new#

```
<script type="text/javascript">
function datacal()
{
var qty = document.getElementById('qty').value;
var prate = document.getElementById('rate').value;
var mytot = qty * prate;
document.getElementById('total').value = mytot;
}
</script>
</head>
<body bgcolor="yellow">
Enter product <input type="text" name="" id="pname"><br>
Qty <input type="text" name="" id="qty"><br>
Rate <input type="text" name="" id="rate" onblur="datacal();"><br>
Total Price <input type="text" name="" id="total"><br>
<button> submit now</button>
</body>
</html>
```

```
<script
  function changeImg()
  {
    document.getElementById('myimg').src="r2.jpg";
  }
</script>
</head>
<body bgcolor="yellow">
<img src="r1.img" id="myimg">
<button onclick="changeImg();">click here!!</button>
</body>
</html>
```

3/08/2022

```
<style
#myimg
{
  border-radius:50%;
  background-color:yellow;
  border:8px solid greenyellow;
</style>
<script type
function changeimg()
{
  document.GetElementById('myimg').src="r2.jpg";
}
function preimg()
{
  document.GetElement By Id('myimg').src="r1.jpg";
}
```



changing the value of an Attribute (Attributes name like value name)

To change the value of an HTML attribute, use this syntax:

document.getElementById(id).attribute = new value

Codes

```
<script - - -
  function changeval()
  {
   document.getElementById('demo').value = "Indore";
  }
</script>
<head>
<body bgcolor="yellow">
 Enter city : <input type="Text" id="demo" value="Bhopal"
              name="">
 <button onclick="change val();"> Click here </button>
```

Img Codes :-

```
<style type="text/css">
 # myimg
  {
   width : 400px;
   height : 300px;
   border : 5px solid light blue;
   padding : 5px;
   border-radius : 50%;
  }
</style>
```

Method	Description
element . setAttribute (attribute , value)	change the attribute value of an HTML element

1

30/07/2022

Javascript HTML DOM - Changing HTML (Document Object Modal)

Changing HTML content.

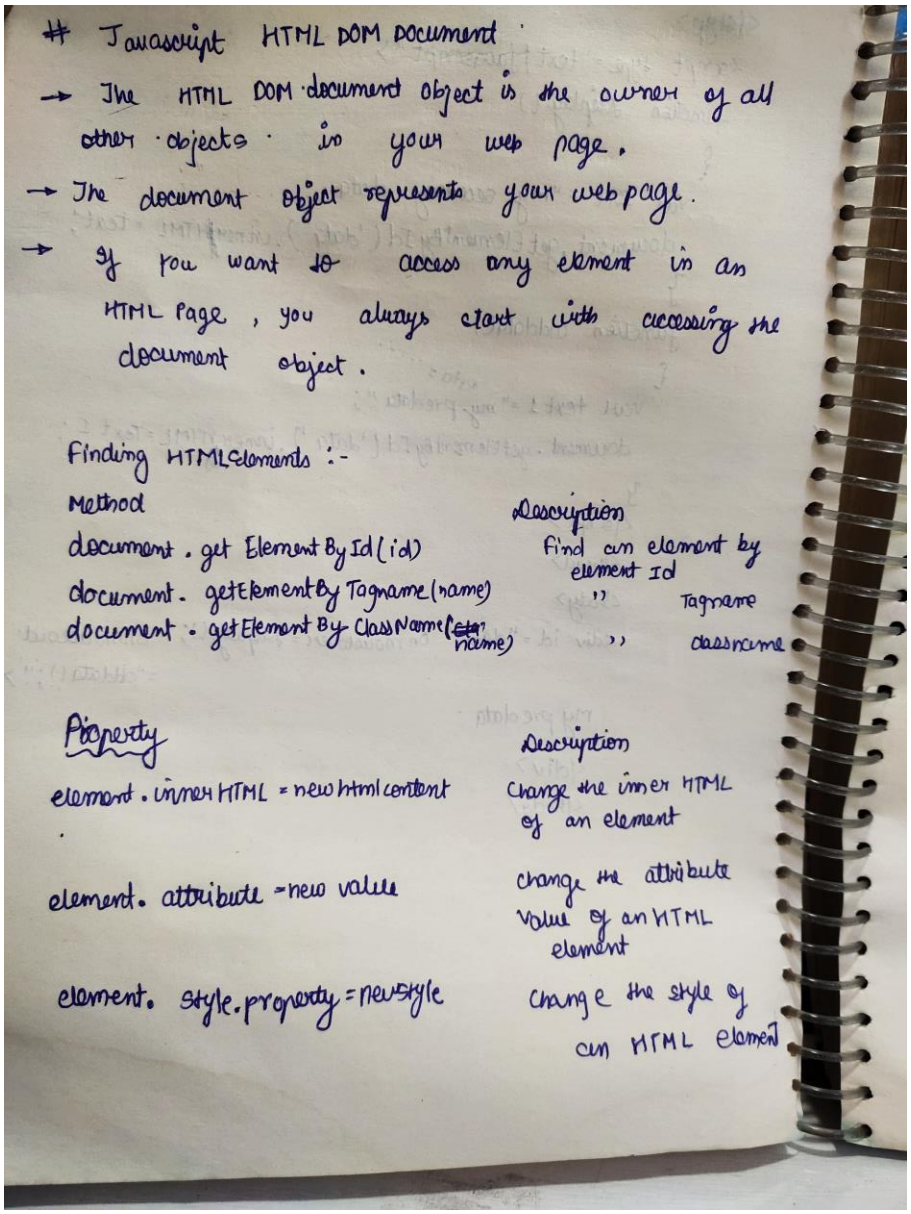
The easiest way to modify the content of an HTML element is by using the innerHTML property.

To change the content of an HTML elements, use this syntax:

Syntax:

document . getElementById (id name) . innerHTML = new text ;

```
<script type = " text/javascript ">
  function changedata ()
  {
  document . getElementById('demo'). innerHTML = "we are we
                                                  software developer ";
  }
  </script>
  </head>
  <body>
   <h1 id="demo"> welcome to cybrom Bhopal </h1>
   <button onclick ="changedata();"> click here! </button>
  </body>
  </html>
```



Javascript HTML DOM document

→ The HTML DOM document object is the owner of all other objects in your web page.

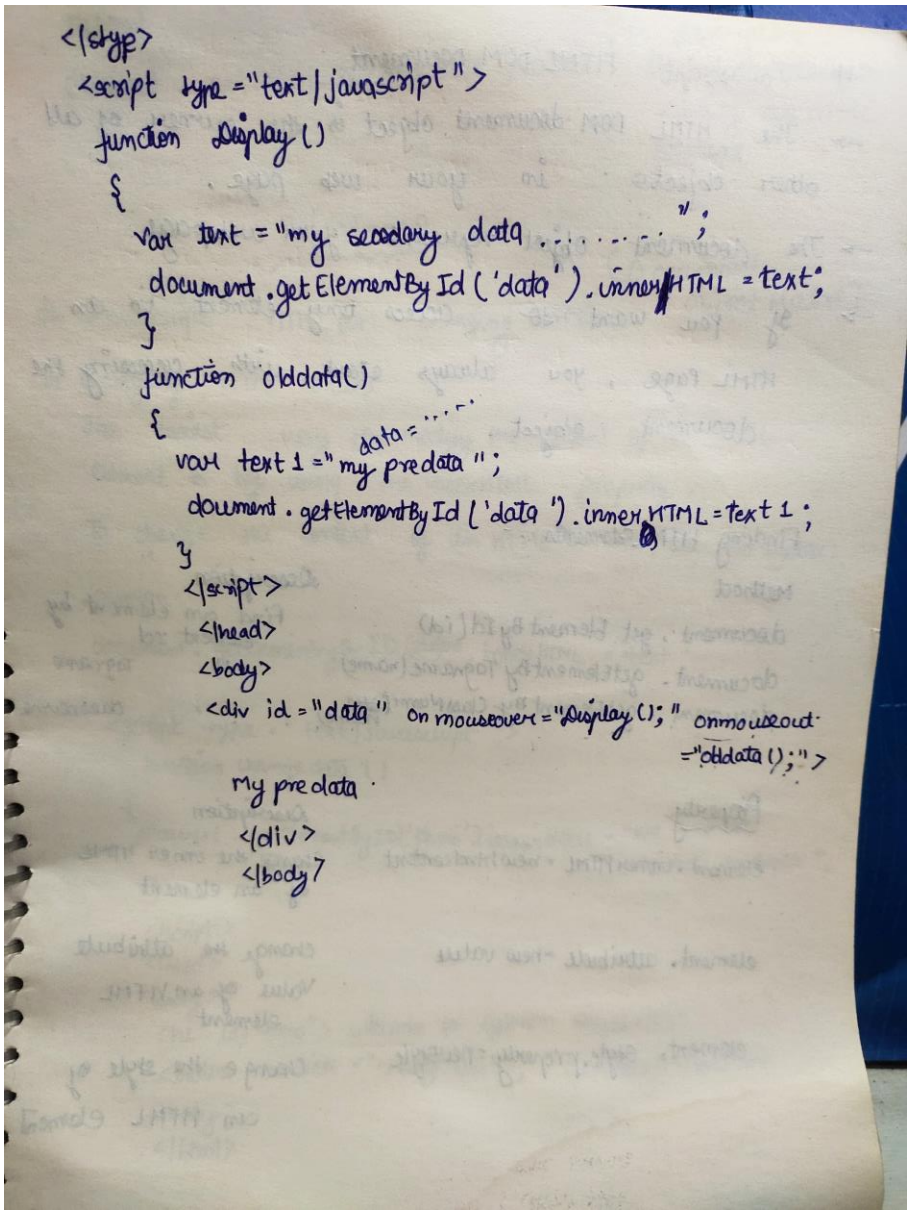
→ The document object represents your web page.

→ If you want to access any element in an HTML page, you always start with accessing the document object.

Finding HTML elements :-

Method	Description
document.getElementById(id)	Find an element by element Id
document.getElementByTagname(name)	" Tagname
document.getElementByClassName(name)	" classname

Property	Description
element.innerHTML = new html content	Change the inner HTML of an element
element.attribute = new value	Change the attribute value of an HTML element
element.style.property = new style	Change the style of an HTML element



```
</styp>
<script type="text/javascript">
function display()
{
var text = "my secodary data ..... ";
document.getElementById('data').innerHTML = text;
}
function olddata()
{
var text1 = "my predata ";
document.getElementById('data').innerHTML = text1;
}
</script>
<head>
<body>
<div id = "data" onmouseover = "display();" onmouseout
                                    ="olddata();">
my predata
</div>
</body>
```

get element by id

DOM

29/07/2022

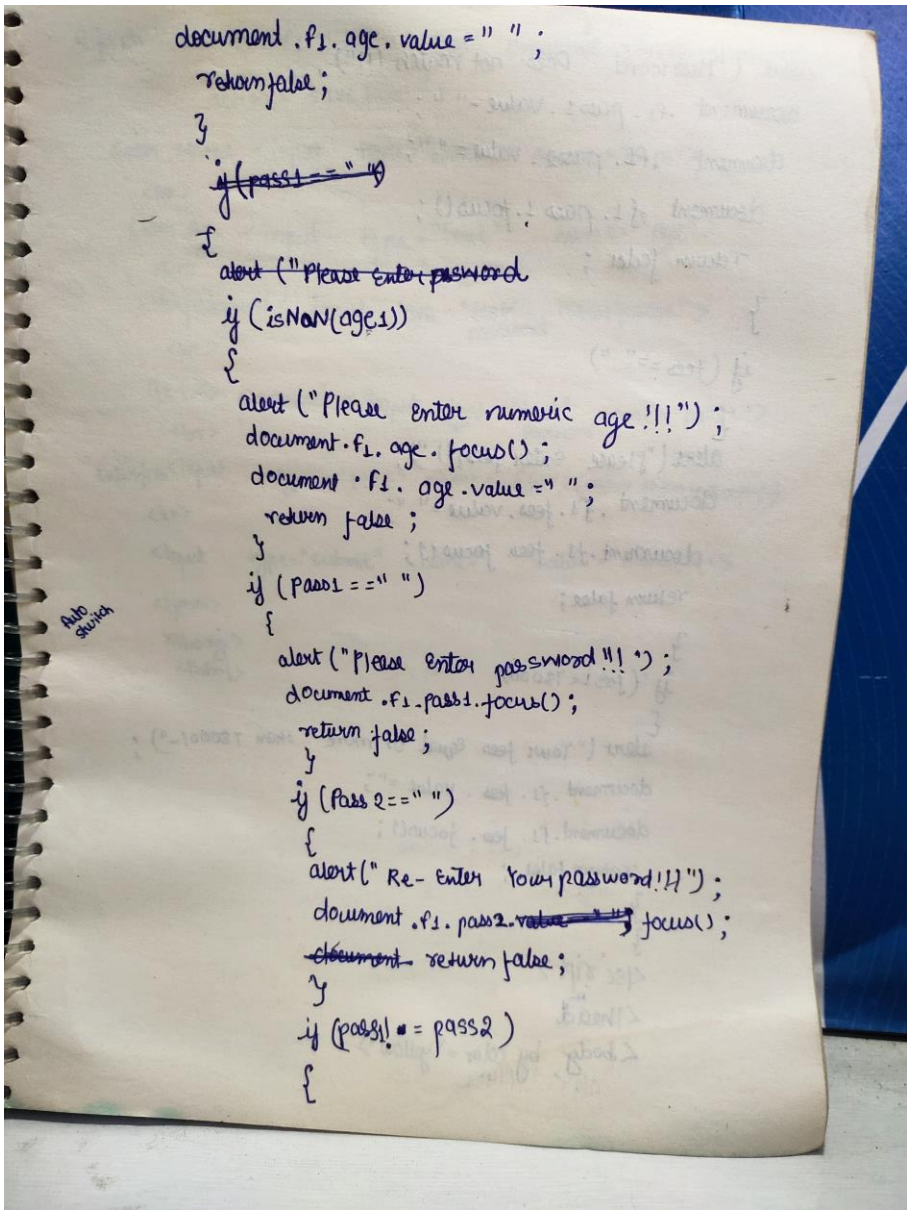
```
<script type="text/javascript">
function Mydata()
{
document.getElementById('demo').innerHTML = "We are
                                              React Developer!!!";
}
</script>
</head>
<body>
<h1 id="demo"> welcome To cybroom Bhopal!!!</h1>
<button onclick="Mydata();"> click here !!!</button>
</body>
</html>
```

→ → → → →

```
<title> my website </title>
<style type="text/css">
#data
{
* font-family: verdana;
  font-size: 15px
  Background color: "yellow;
  color: dark red;
  width: 300px;
  heign: 250 px;
  Padding: 20px;
  border: 3px solid red
  border radius: 30px
  text align: justify;
}
```

```
<form  name="f1" method "Post" onsubmit = " return validation()
       action = "save.html" >
Enter Name <input  type = "text"  name = "nm" >
<br>
Enter Age < input  type = "text"  name = "age" >
<br>
Enterpassword < input  type = "text"  name = "pass1" >
                      password
<br>
Re-enter password < input  type = "text"  name = "pass2" >
<br>                      Password
Enterfees <input  type = "text"  name = "Fees" > (more than 48000/-
<br>
<Input  type = "submit"  name = " "  value = "save!!!" >
</form>
</body>
</html>
```

```
alert ("Password Does not match !!!");
document.f1.pass1.value="";
document.f1.pass2.value="";
document.f1.pass1.focus();
return false;
}
if (fees=="")
{
alert("Please enter fees!!!");
document.f1.fees.value="";
document.f1.fees.focus();
return false;
}
if (fees1<78000)
{
alert("Your fees Equal or more than 78000/-");
document.f1.fees.value="";
document.f1.fees.focus();
return false;
}
}
</script>
<head
<body bg color="yellow">
```



```
document.f1.age.value = " ";
return false;
}
if(pass1 == "")
{
alert ("Please Enter password
if (isNaN(age1))
{
alert ("Please enter numeric age !!!");
document.f1.age.focus();
document.f1.age.value = " ";
return false;
}
if (pass1 == " ")
{
alert ("Please enter password !!!");
document.f1.pass1.focus();
return false;
}
if (pass2 == " ")
{
alert ("Re-Enter your password !!");
document.f1.pass2.value = " "; focus();
document return false;
}
if (pass1 != pass2)
{
```

Auto Switch

```
<input type="submit" name="" value="Data save!!!">
</form>
</body>
</html>
```

28/07/22

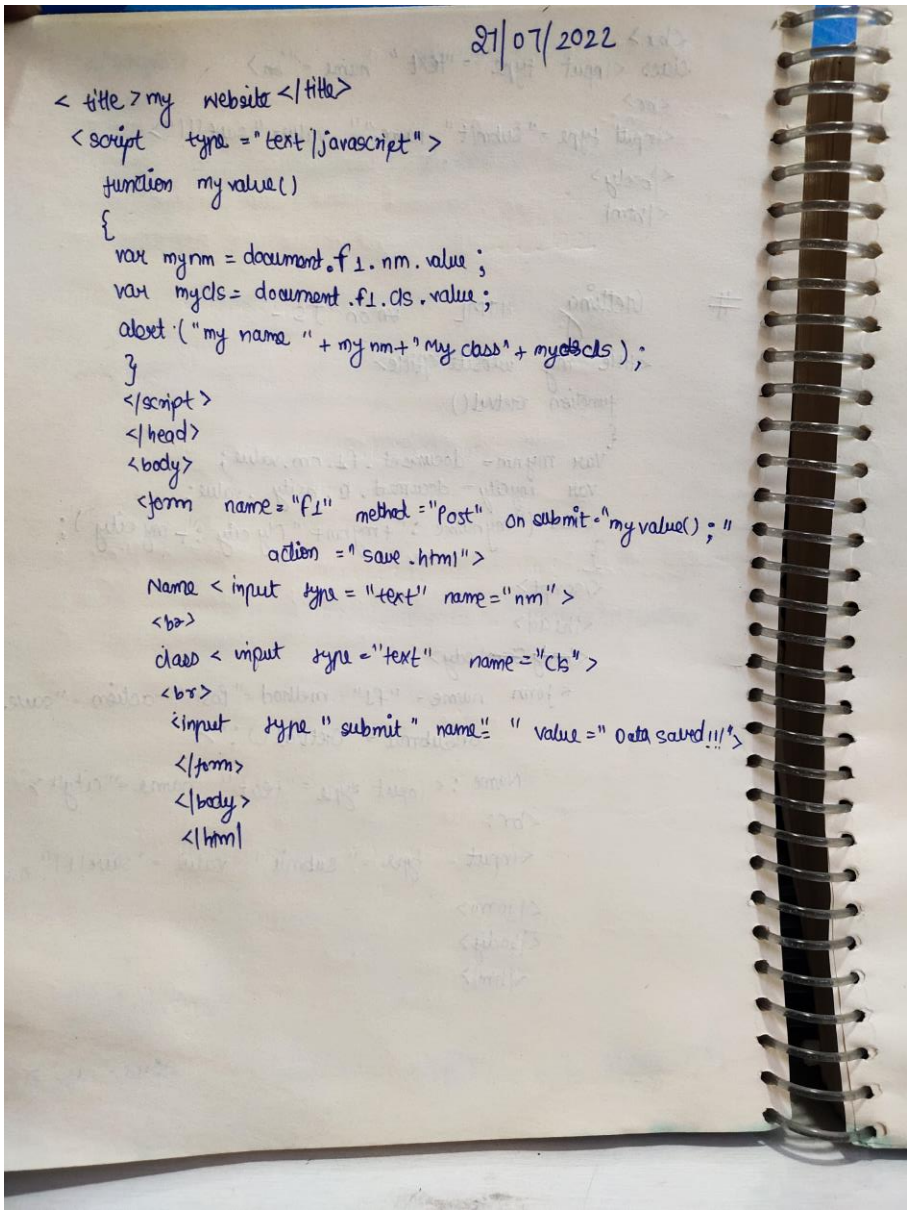
```
<title>my website</title>
<script type="text/javascript">
function validation()
{
var mynm = document.f1.nm.value;
var age = document.f1.age.value;
var age1 = parseInt(age);
var pass1 = document.f1.pass1.value;
var pass2 = document.f1.pass2.value;
var fees = document.f1.fees.value;
var fees1 = parseInt(fees);
if (mynm=="")
{
alert("Please Enter Your name!!!");
document.f1.nm.focus();
return false;
}
if (age=="")
{
alert("Please enter Your Age!!!");
```

Client - end validation —

```
<title> my website </title>
<script type = " text/javascript ">
    function validation ()
    {
    var myname = document.f1.nm.value ;
    var mycity = document.f1.city.value ;
    if (myname == " ")
    {
     alert ("Please Enter your name !!!");
     document.f1.nm.focus();
     return false;
    }
     if (mycity == " ").
     {
      alert ("Please Enter your city !!!");
      document.f1.city.focus();
       return false;
      }
     }
     </script>
     </head>
     <body bg color = "yellow">
      <form name = "f1"
           method = " post "    onsubmit = " return validation();"
           action = " save.html) ">
     Enter name <input type = "text"  name = "nm">
     <br>
     Enter city <input type = "text"  name = "city">
      <br>
```

Client - end validation —

```
<title> my website </title>
<script type = " text/javascript ">
    function validation ()
    {
    var myname = document.f1.nm.value ;
    var mycity = document.f1.city.value ;
    if (myname == " ")
    {
     alert ("Please Enter your name !!!");
     document.f1.nm.focus();
     return false;
    }
     if (mycity == " ").
     {
      alert ("Please Enter your city !!!");
      document.f1.city.focus();
       return false;
      }
     }
     </script>
     </head>
     <body bg color = "yellow">
      <form name = "f1"
           method = " post "    onsubmit = " return validation();"
           action = " save.html) ">
     Enter name <input type = "text"  name = "nm">
     <br>
     Enter city <input type = "text"  name = "city">
      <br>
```



27/07/2022

```
<title>my website</title>
<script type="text/javascript">
function myvalue()
{
var mynm = document.f1.nm.value;
var mycls = document.f1.cls.value;
alert("my name " + mynm + "my class" + mycls);
}
</script>
</head>
<body>
<form name="f1" method="post" onsubmit="myvalue();"
action="save.html">
Name <input type="text" name="nm">
<br>
class <input type="text" name="cls">
<br>
<input type="submit" name="" value="Data saved!!!">
</form>
</body>
</html>
```

```
<br>
class <input type = "text" name = "" on >
<br>
<input type = "submit" name = "" value = "save!!!">
</body>
</html>
```

Getting HTML on JS -

```
<title> my website </title>
function Getval()
{
var mynm = document.f1.nm.value;
var mycity = document.f1.city.value;
alert ("myname :" + mynm + "My city :" + my city );
}
</script>
<head>
<body>
<form name = "f1" method = "post" action = "save.html"
onsubmit = "Getval();">
Name : <input type = "text" name = "city" >
<br>
<input type = "submit" value = "save!!!" name="">
</form>
</body>
</html>
```

```
</script>
</head>
<body>
<button onclick ="Display();"> Click here!!!</button>
</body>
</html>
```

codes-: onmouseover , onmouseout

```
<title>...</title>
<script type =" text/javascript">
function Display()
{
alert ("This event example!!!");
}
</script>
</head>
<body>
<button onmouseout = "Display();"> Click here!!!</button>
</body>
</html>
```

#(N)

```
<title> ..</title>
<script type =" text/javascript ">
function Display ()
{
alert ("Please enter your name!!!");
}
</script>
</head>
<body>
Name<input type ="text" name="" onfocus ="Display();">
```

Function and event in javascript

→ **Function :-** Function are small program or sub program, we can create function in java script using function . keyword follow with function name, and call anywhere in the ~~prog~~program.

Syntax :- -

```
function functionname()
{
//body of the function
}
```

Event in Js :- Javascript is a Event Handelling programming language, which support various type of event. Events are some specific situation and this situation we can perform some specific task.

In javascript there are various event, some basic events are : onclick, onmouseover, onmouseout, onsubmit, onblur, onfocus, onchange, onload, onunload, ... etc

Example codes

```
<title> my website </title>
<script type = "text/javascript">
function display()
{
alert ("This Event example !!!");
}
```

26/07/2022

Prompt () :-- this function is used to input value by user at run time.

- this function can input by using prompt box. the prompt function has two parameteres. first is message and second is default value, the default value is an optional parameter.

syntax :--

```
prompt ( Message , Default value) ;
```

code-

```
<body>
<script type ="text/javascript">
var age = prompt("Enter your age", 18);
if (age >=18)
{
 alert ("You can vote!!!");
}
else
{
 alert ("sorry ! you can not vote!!! ");
}
</script>
</body>
```

syntax: --->

confirm (message to be display)

```
<body>
<script type ="text/javascript">
  var myval = confirm ("This is conjumation Message");
  if (myval == True)
  {
    alert(You click on ok button !!!");
  }
  else
  {
    alert (" you click on ok cancle button !!!");
  }
</script>
</body>
</html>
```

alert(), confirm() and prompt() in JS :-

• alert() :-

The alert() function is used to display alert() messages in Box (formal the alert box has only on button that ok button. when we click Ok button, this box does not return any value.

Syntax: ---

alert(message to be display)

#

```
<body>
<script type = "text/javascript">
var myval = alert(' This is alert Box message!!!");
document. write ("<h1>", my val, "</h1>");
</script>
</body>
```

• confirm() : --- This function is used to display conformation message in Box format.

Conform box has two button Ok and cancel. when we click ok button this box return True when we click cancle button. This Box return false.

If we want to assign string value to the variable we use single or double quotes for numeric type of value we don't use any quotes.

```
<body>
<script type="text/javascript">
    var phy = 55;
    var che = 45;
    var math = 41;
    var eng = 78;
    var hnd = 64;
    var total = phy + che + math + eng + hnd;
    var per = total *100/500 ;
    document. write ("<h1> total marks : ", total, "</h1>");
    document. write ("<br>");
    document. write ("<h2> percentage :", per, "</h2>");
</script>
</body>
</html>
```

```
<body>
<script type="text/javascript">
    var no1 = 45;
    var no2 = 33;
    var ans = no1 + no2;
    document.write(ans);
</script>
</body>
</html>
```

Variable in Javascript :-

- In java script we can declare variable using var keyword.
- we can also assign the value at the time of declaration. To declare a variable we can use var keyword follow with variable name.

Syntax :- var variable name;
var variable name = value;

Example- var name = "sachin";
var sal = 34590;
var age;
age = 23;

25/07/2022

Javascript

- It is a client/server side scripting language: which is used to design a web application.

- Using javascript we can create client side validation, drop down menus, DOM, and many more it is light weighted, do more, event handeling programming language.

- we can write javascript inside the body section and inside the head section or we can call external javascript inside the head section.

```
<html>
  <head>
  <title> my website </title>
  <script type = " text / javascript" src = "filename.js" >
  </script>
  <script type = " text / javascript">
        //JS statements
   </script>
  </head>
  <body>
   <script type = "text/ javascript" >
      //JS statement .
    </script>
  </body>
  </HTML>
```

25/07/2022

Javascript

- It is a client/server side scripting language: which is used to design a web application.

- Using javascript we can create client side validation, drop down menus, DOM, and many more it is light weighted, do more, event handeling programming language.

- we can write javascript inside the body section and inside the head section or we can call external javascript inside the head section.

```
<html>
  <head>
  <title> my website </title>
  <script type = " text / javascript" src = "filename.js" >
  </script>
  <script type = " text / javascript">
        //JS statements
   </script>
  </head>
  <body>
   <script type = "text/ javascript" >
      //JS statement .
    </script>
  </body>
  </HTML>
```