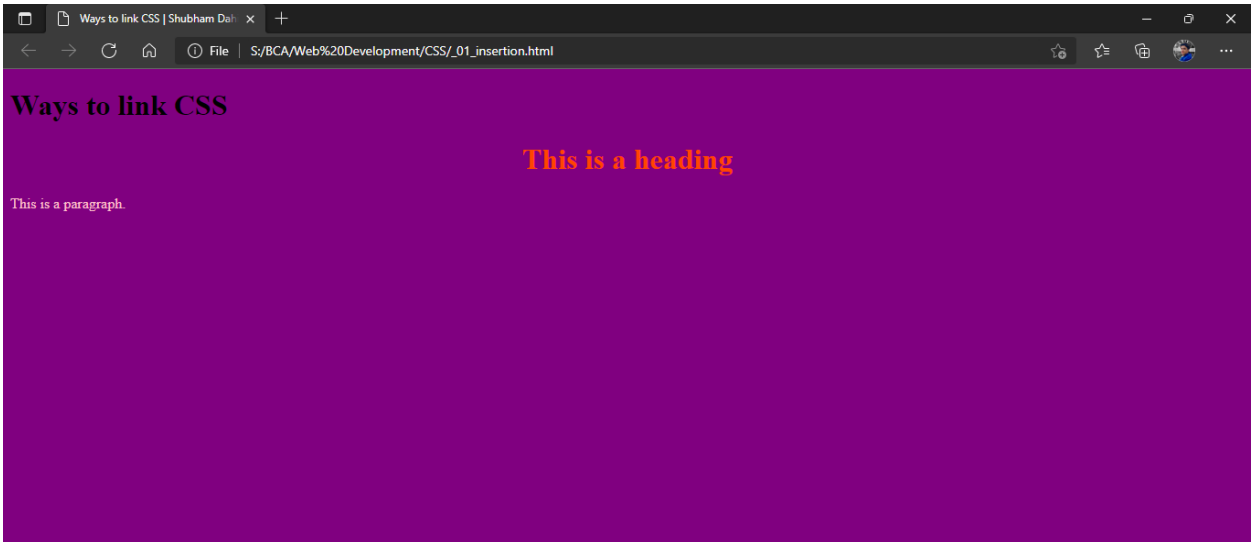
Cascading Style Sheets

1. Ways to link CSS

```
<!DOCTYPE html>
<html>
<head>
<title>Ways to link CSS | Shubham Dahiya</title>

<!-- linking external css -->
<link rel="stylesheet" href="mystyle.css">

<!-- Internal CSS -->
<style>
body {
background-color: lightblue;
}
</style>
</head>
<body>
<h1>Ways to link CSS</h1>

<!-- Inline CSS -->
<h1 style="color:orangered; text-align:center">This is a heading</h1>
<p style="color: pink">This is a paragraph.</p>

</body>
</html>
```

OUTPUT:

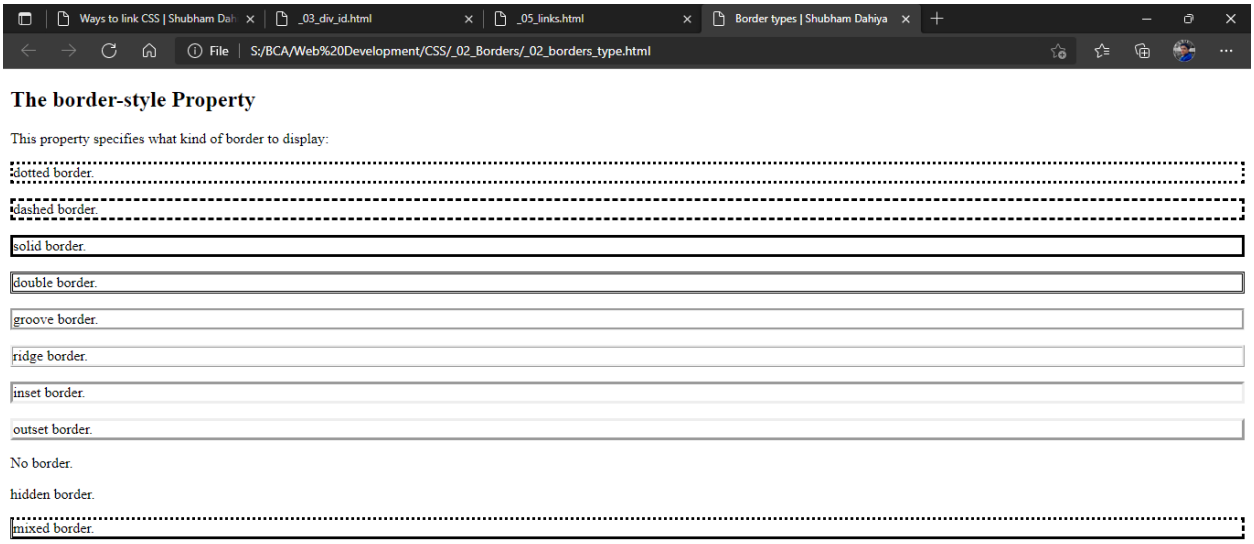


2. Border types

```
<!DOCTYPE html>
<html>
<head>
<title>Border types | Shubham Dahiya</title>
<style>
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.none {border-style: none;}
```

```
p.hidden {border-style: hidden;}
p.mix {border-style: dotted dashed solid double;}
</style>
</head>

<body>

<h2>The border-style Property</h2>
<p>This property specifies what kind of border to display:</p>

<p class="dotted"> dotted border.</p>
<p class="dashed"> dashed border.</p>
<p class="solid"> solid border.</p>
<p class="double"> double border.</p>
<p class="groove"> groove border.</p>
<p class="ridge"> ridge border.</p>
<p class="inset"> inset border.</p>
<p class="outset"> outset border.</p>
<p class="none">No border.</p>
<p class="hidden"> hidden border.</p>
<p class="mix"> mixed border.</p>

</body>
</html>
```

OUPTUT:



3. Border width

```
<!DOCTYPE html>
<html>
<head>
<title>Border width | Shubham</title>
<style>
p.one {
  border-style: solid;
  border-width: 5px;
}

p.two {
  border-style: solid;
  border-width: medium;
}

p.three {
```

```
  border-style: dotted;
  border-width: 2px;
}

p.four {
  border-style: dotted;
  border-width: thick;
}

p.five {
  border-style: double;
  border-width: 15px;
}

p.six {
  border-style: double;
  border-width: thick;
}
/*Specic side widths*/
p.seven {
  border-style: solid;
  border-width: 25px 10px 4px 35px; /* 25px top, 10px right, 4px bottom and 35px left */
}
</style>
</head>
<body>

<h2>The border-width Property</h2>
<p>This property specifies the width of the four borders:</p>
```

```
<p class="one">Shubham Dahiya</p>
<p class="two">Shubham Dahiya</p>
<p class="three">Shubham Dahiya</p>
<p class="four">Shubham Dahiya</p>
<p class="five">Shubham Dahiya</p>
<p class="six">Shubham Dahiya</p>
<p class="seven">Shubham Dahiya</p>

<p><b>Note:</b> The "border-width" property does not work if it is used alone.
Always specify the "border-style" property to set the borders first.</p>

</body>
</html>
```

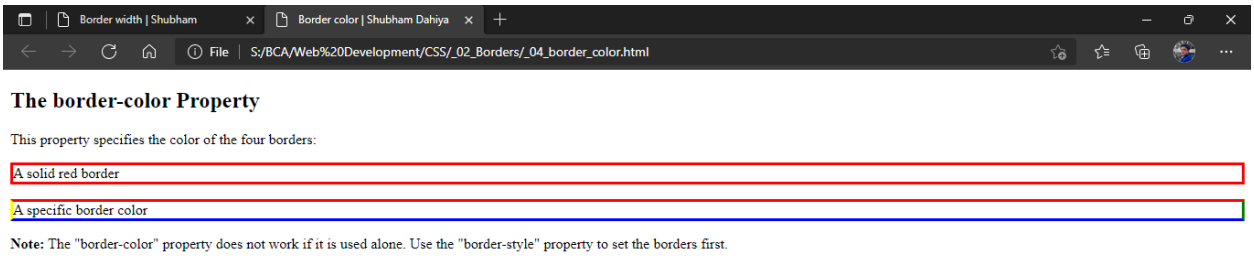
OUTPUT:

4. Border color

```
<!DOCTYPE html>
<html>
<head>
<title>Border color | Shubham Dahiya</title>
<style>
p.one {
  border-style: solid;
  border-color: red;
}

p.two {
  border-style: solid;
  border-color: red green blue yellow;
}

</style>
</head>
<body>

<h2>The border-color Property</h2>
<p>This property specifies the color of the four borders:</p>

<p class="one">A solid red border</p>
<p class="two">A specific border color</p>

<p><b>Note:</b> The "border-color" property does not work if it is used alone. Use the "border-style" property to set the borders first.</p>
```

```
</body>

</html>
```

OUTPUT:

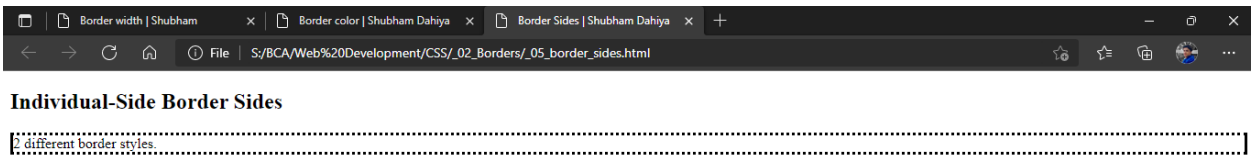


5. Border sides

```
<!DOCTYPE html>

<html>

<head>

<title>Border Sides | Shubham Dahiya</title>


<style>

p {

  border-top-style: dotted;

  border-right-style: solid;

  border-bottom-style: dotted;

  border-left-style: solid;

}
```

```
</style>
</head>
<body>

<h2>Individual-Side Border Sides</h2>
<p>2 different border styles.</p>

</body>
</html>
```

OUTPUT:

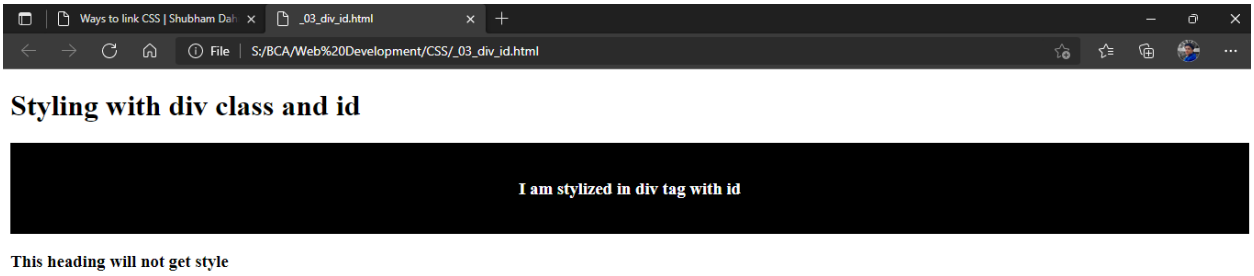


6. Using division tag and id attributes

```
<!DOCTYPE html>
<html>
<head>
<style>
#abc {
```

```
  background-color: black;
  color: white;
  padding: 20px;
  text-align: center;
}

#abcd {
  background-color: black;
  color: white;
  padding: 20px;
  text-align: center;
}
</style>
</head>
<body>

<h1>Styling with div class and id</h1>
<div id="abc">
   <h3> I am stylized in div tag with id</h3>
</div>

<h3 >This heading will not get style</h3>

</body>
</html>
```

OUTPUT:



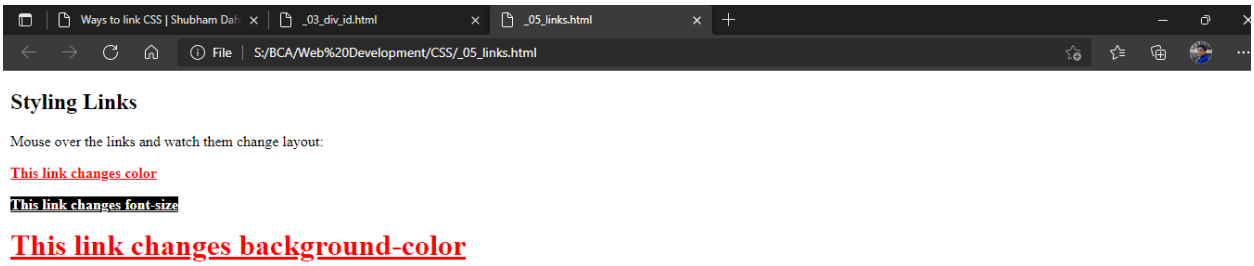
7. Styling Links

```
<!DOCTYPE html>
<html>
<head>
<style>
a.one:link {
   color: red;
}
a.one:visited {
   color: green;
}
a.one:hover {
color: hotpink;
text-decoration: underline;
}

a.two:link {
color:white; background-color:black;
```

```
}
a.two:visited {
color:cyan;
}
a.two:hover {
font-size:150%;
}

a.three:link {
color:#ff0000;
font-size:200%;
}
a.three:visited
{
color:#0000ff;
text-decoration:none;
}
a.three:hover {
background:#66ff66;
}

</style>
</head>
<body>

<h2>Styling Links</h2>

<p>Mouse over the links and watch them change layout:</p>
```

```
<p><b><a class="one" href="default.asp" target="_blank">This link changes color</a></b></p>

<p><b><a class="two" href="default.asp" target="_blank">This link changes font-size</a></b></p>

<p><b><a class="three" href="default.asp" target="_blank">This link changes background-color</a></b></p>


</body>

</html>
```

OUTPUT:



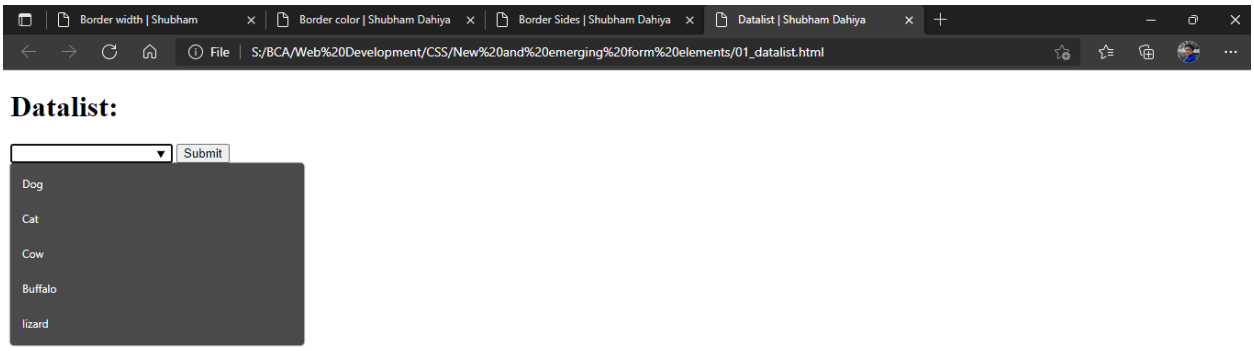
New and emerging form elements:

8. Datalist

```
<!DOCTYPE html>

<html>

<head>

<title>Datalist | Shubham Dahiya</title>

</head>
```

```
<body>
<h1>Datalist:</h1>
<form action="#" method="get">
<input list="animal" name="animal">
<datalist id="animal">
<option value="Dog">
<option value="Cat">
<option value="Cow">
<option value="Buffalo">
<option value="lizard">
</datalist>
<input type="submit">
</form>
</body>
</html>
```

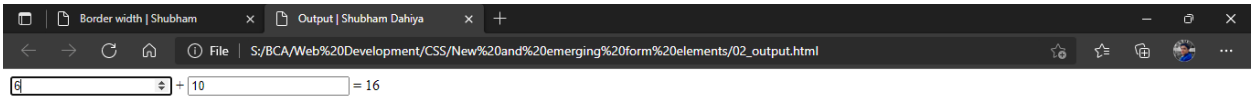
OUTPUT:



9. Output

```
<!DOCTYPE html>
<html>
<head>
<title>Output | Shubham Dahiya</title>
</head>
<body>
<form oninput="result.value = parseInt(first.value) + parseInt(second.value)">
<input name="first" type="number" value="0"> +
<input name="second" type="number" value="10"> =
<output name="result" for="first second">0</output>
</form>
</body>
</html>
```

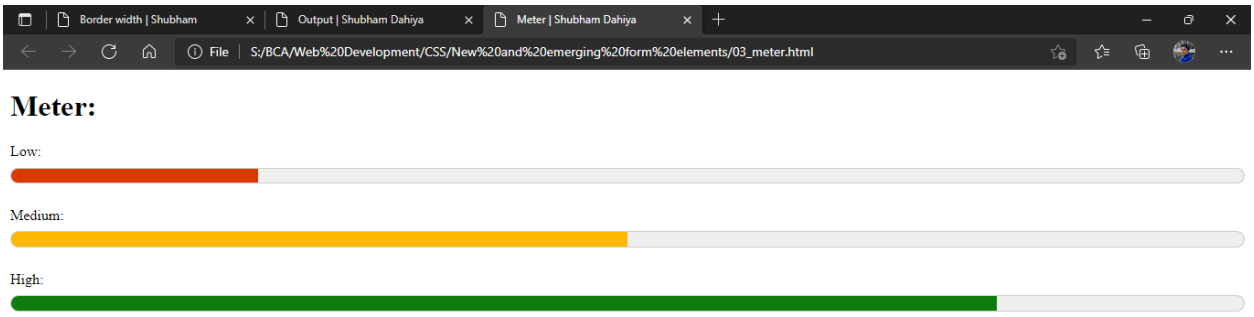
OUTPUT:



10. Meter

```
<!DOCTYPE html>
<html>
<head>
<title>Meter | Shubham Dahiya</title>
<style>
meter
{
width:100%;
height:35px;
}
</style>
</head>
<body>
<h1>Meter: </h1>
<p>Low: <meter value="20" min="0" max="100" low="25" high="75" optimum="100"></meter></p>
<p>Medium: <meter value="50" min="0" max="100" low="25" high="75"
optimum="100"></meter></p>
<p>High: <meter value="80" min="0" max="100" low="25" high="75" optimum="100"></meter></p>
</body>
</html>
```

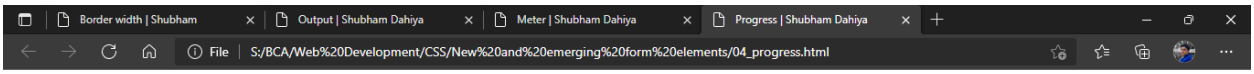
OUTPUT:



11. Progress

```
<!DOCTYPE html>
<html>
<head>
<title>Progress | Shubham Dahiya</title>
</head>
<body>
<h1>Progress: </h1>
<p>Indeterminate: <progress max="100"></progress></p>
<p>Progress 60%: <progress value="60" max="100"></progress></p>
</body>
</html>
```

OUTPUT:



Progress:

Indeterminate:

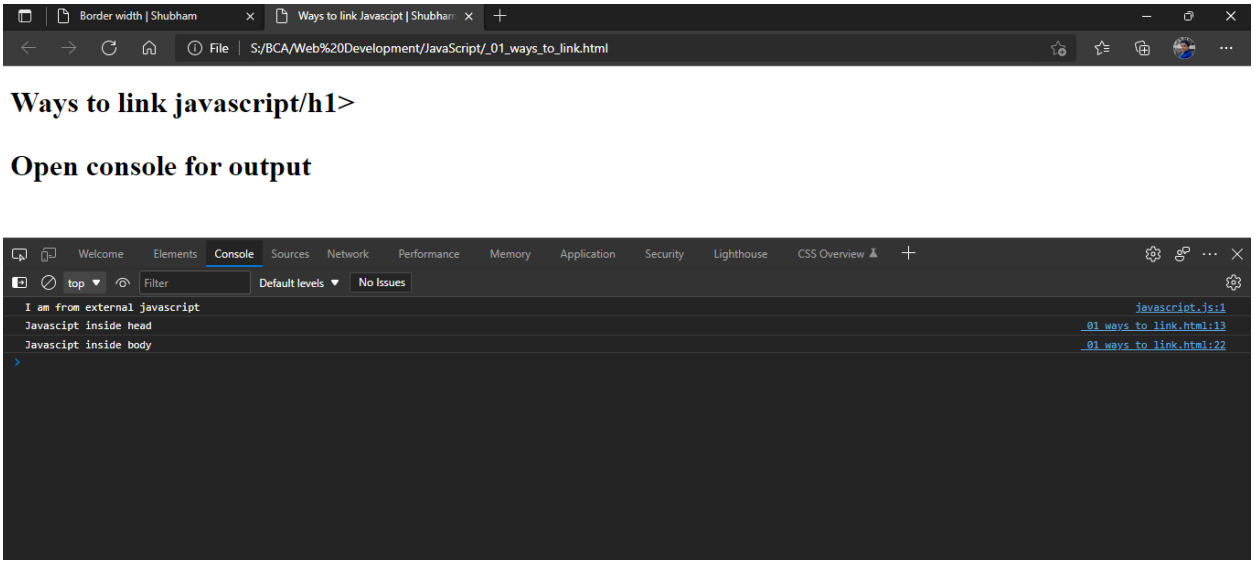
Progress 60%:

JavaScript

12. Ways to link

```
<!DOCTYPE html>
<html>
<head>

<title>Ways to link Javascipt | Shubham Dahiya</title>

<!-- linking javascript via external file -->
<script src="javascript.js"> </script>

<!-- Javascript inside body -->
<script>

console.log("Javascipt inside head")
</script>
</head>
<body>
<h1>Ways to link javascript/h1>
<p >Open console for output</p>

<script>

console.log("Javascipt inside body")
</script>
</body>
</html>
```

OUTPUT:



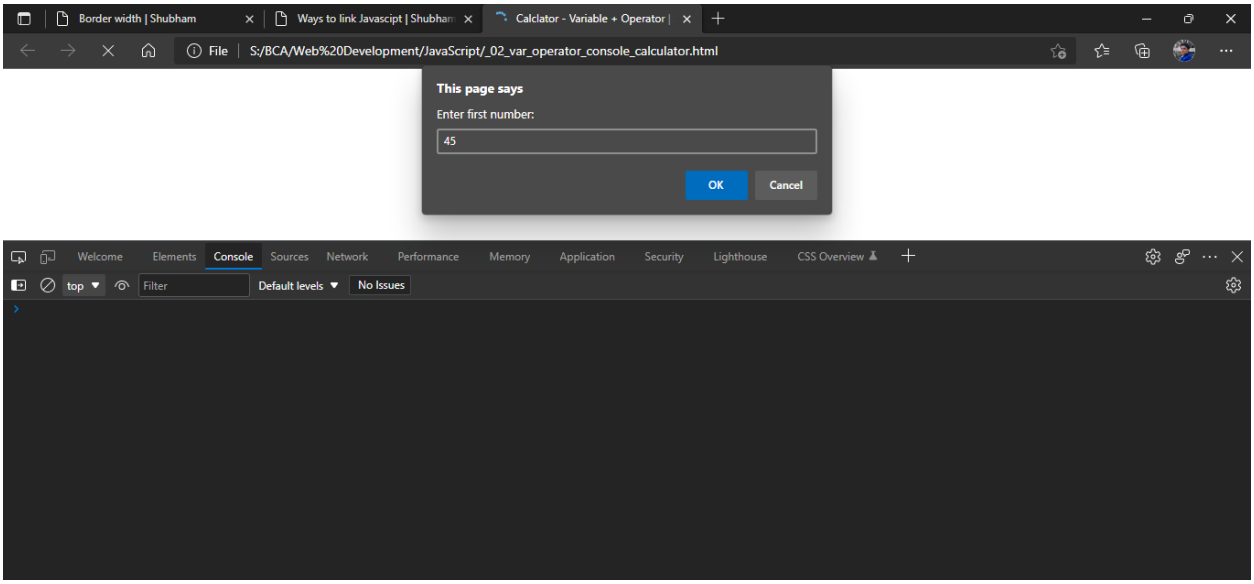
13. Variable & Operator in JavaScript – Calculator Program

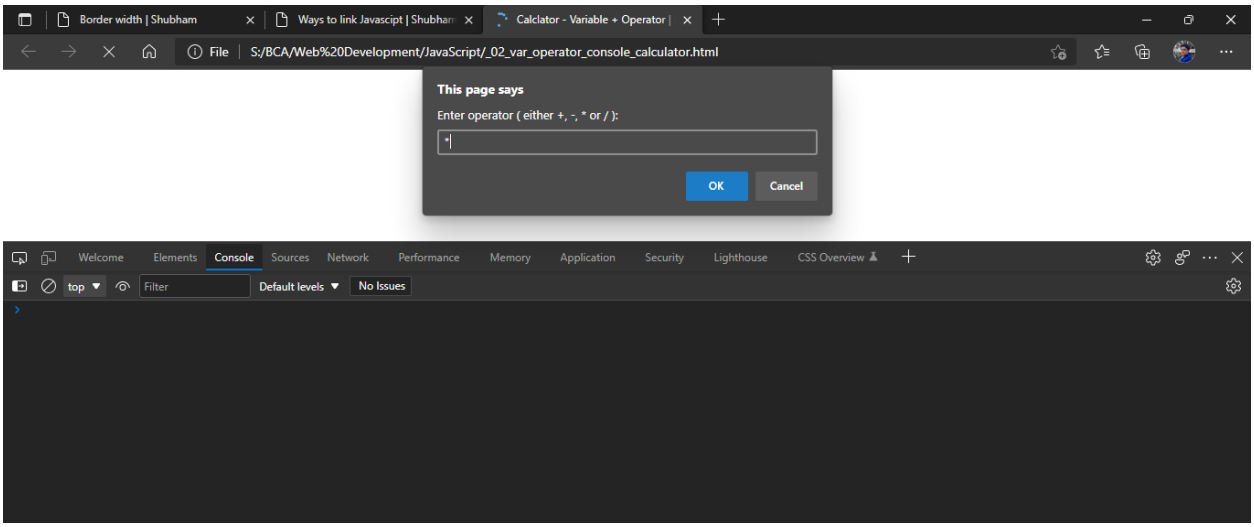
```
<!DOCTYPE html>

<html>

<head>


<title>Calclator - Variable + Operator | Shubham Dahiya</title>



<script>


// take number 1

var number1 = parseFloat(prompt('Enter first number: '));
```

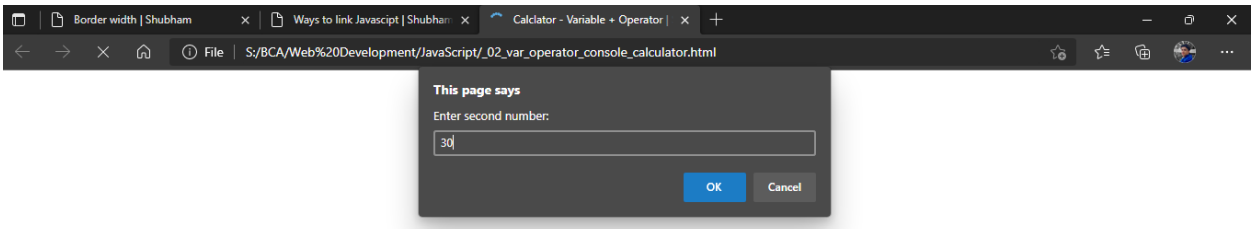
```
// take the operator input
var operator = prompt('Enter operator ( either +, -, * or / ): ');
//taking number 2
var number2 = parseFloat(prompt('Enter second number: '));

var result;

// using if...else if... else
if (operator == '+') {
result = number1 + number2;
}
else if (operator == '-') {
result = number1 - number2;
}
else if (operator == '*') {
result = number1 * number2;
}
else {
result = number1 / number2;
}

// result output
console.log(`${number1} ${operator} ${number2} = ${result}`);
</script>
</head>
<body>
<h1>Variable + Operator in Javascipt </h1>
<h3>Calculator Program</h3>
<p >Open console for output</p>
```

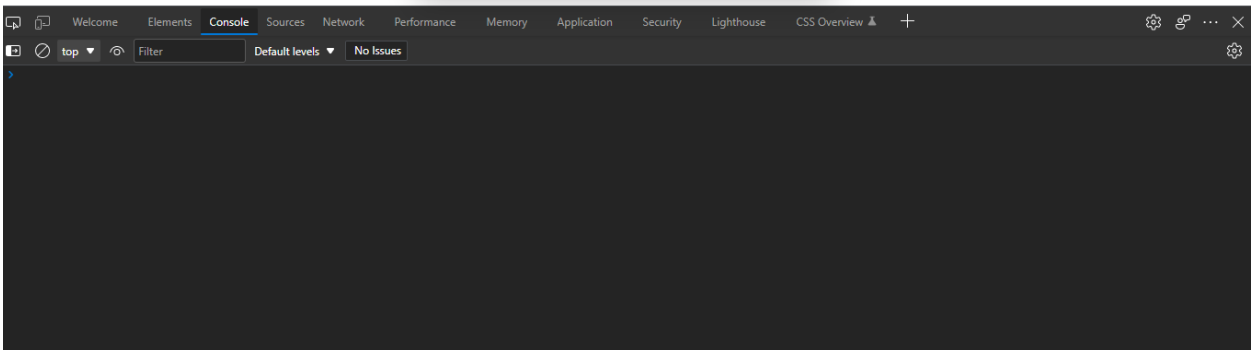
```
</body>

</html>
```

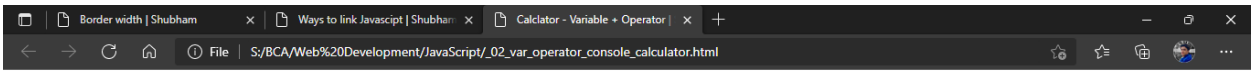
OUTPUT:







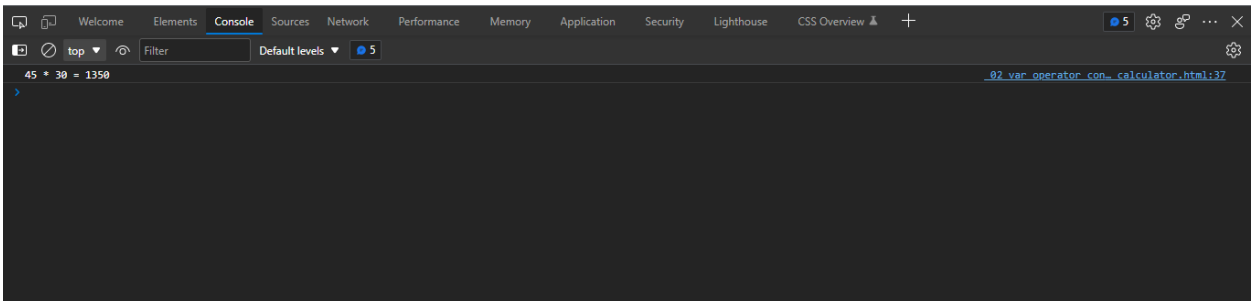




Variable + Operator in Javascipt

Calculator Program

Open console for output



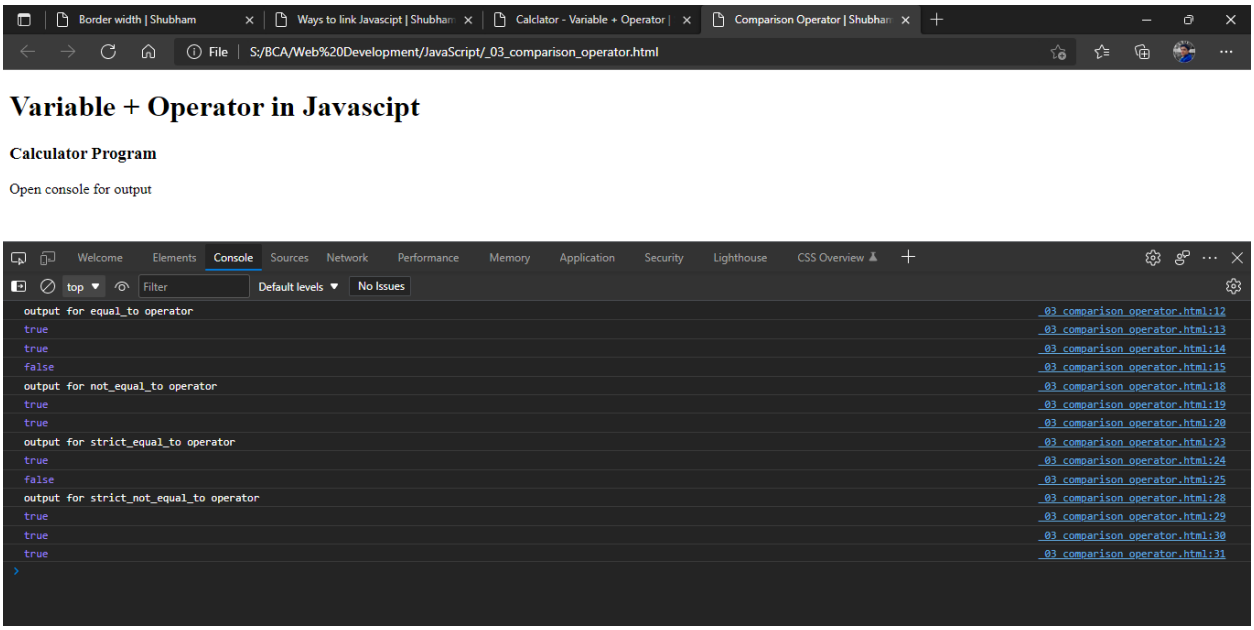
14. Comparison operator

```
<!DOCTYPE html>

<html>

<head>


   <title>Comparison Operator | Shubham Dahiya</title>
```

```
  <script>
      var a = 10, b = 20;
      var c = 'javascript';

      // equal to operator
      console.log("output for equal_to operator")
      console.log(a == 10);     // returns true
      console.log(b == '20');   // returns true
      console.log(c == 'Javascript');  //  returns false

      // not equal operator
      console.log("output for not_equal_to operator")
      console.log(a != 20); // returns true
      console.log(b != 'Javascript'); // returns true

      // strict equal operator
      console.log("output for strict_equal_to operator")
      console.log(a === 10); // true
      console.log(a === '10'); // false

      // strict not equal operator
      console.log("output for strict_not_equal_to operator")
      console.log(a !== 2); // false
      console.log(a !== '2'); // true
      console.log(b !== 'Hello'); // true


  </script>
</head>
```

```
<body>
  <h1>Variable + Operator in Javascipt </h1>
  <h3>Calculator Program</h3>
  <p >Open console for output</p>

</body>
</html>
```

OUTPUT:



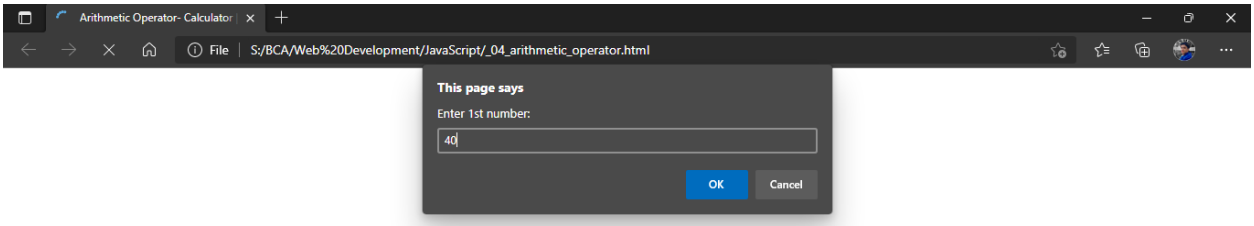
15. Arithmethic operator

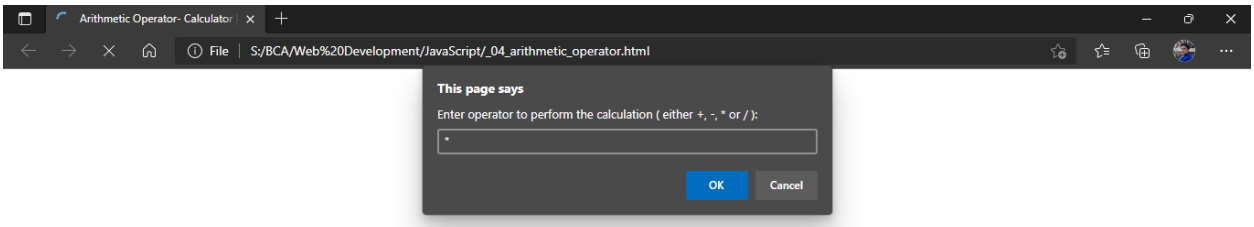
```
<html>
<head>
  <title>Arithmetic Operator- Calculator | Shubham Dahiya</title>
```

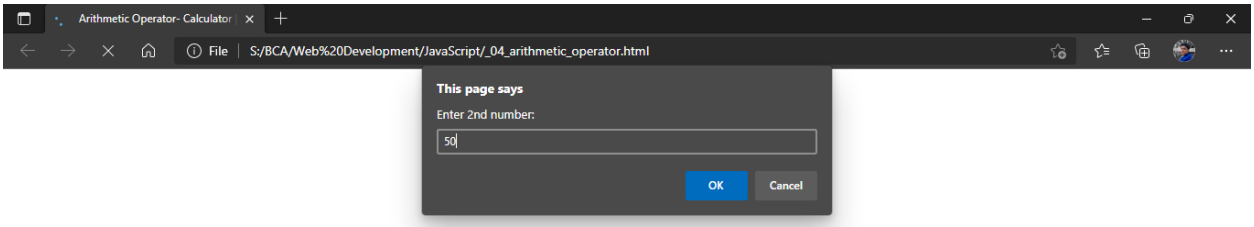
```
<script>

// accept the number from the user through prompt box
const number1 = parseFloat(prompt ('Enter 1st number: '));
const operator = prompt('Enter operator to perform the calculation ( either +, -, * or / ): ');
const number2 = parseFloat(prompt ('Enter 2nd number: '));

let result;

//if-else ladder:
if (operator == '+') {
   result = number1 + number2;
}
else if (operator == '-') {
   result = number1 - number2;
}
else if (operator == '*') {
   result = number1 * number2;
}
else {
   result = number1 / number2;
}

// Result
window.alert(" Result is" + result);
</script>

<body>
   <h1>Calclulator - Arithmetic Operator</h1>
```

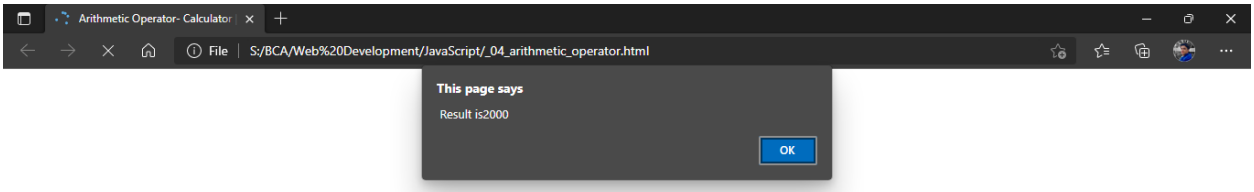
```
</body>
  </html>
```

OUTPUT:







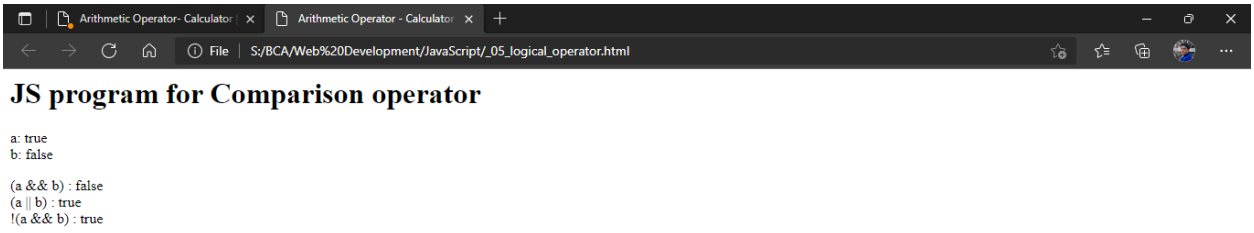


16. Logical operator

```
<html>
  <head>
    <title>Arithmetic Operator - Calculator | Shubham Dahiya</title>
  </head>
  <body>
  <h1>JS program for Comparison operator</h1>
  <p>a: true <br> b: false</p>
  <script type = "text/javascript">


      const a = true;
```

```
        const b = false;
        const linebreak = "<br />";

        document.write("(a && b) : ");
        document.write(a && b); //return false
        document.write(linebreak);

        document.write("(a || b) : ");
        document.write(a || b); //return true
        document.write(linebreak);

        document.write("!(a && b) : ");
        document.write(!(a && b)); //return true
        document.write(linebreak);

    </script>
  </body>
</html>
```

OUTPUT:

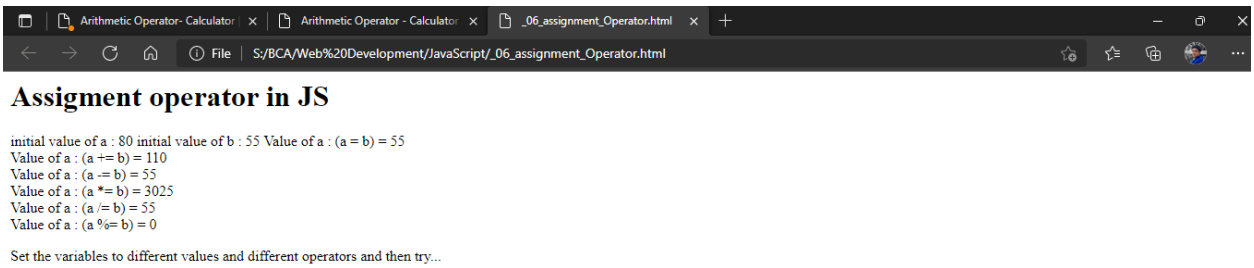


17. Assignment operator

```
<html>
   <body>

   <h1>Assigment operator in JS</h1>
    <script type = "text/javascript">
        var a = 80;
        var b = 55;

        document.write("initial value of a : " + a +"\n initial value of b : " + b );

        document.write("\nValue of a : (a = b) = ");
        document.write( a = b);
        document.write("<br />");

        document.write("Value of a : (a += b) = ");
        document.write(a += b);
        document.write("<br />");

        document.write("Value of a : (a -= b) = ");
        document.write(a -= b);
        document.write("<br />");

        document.write("Value of a : (a *= b) = ");
        document.write(a *= b);
        document.write("<br />");
```

```
        document.write("Value of a : (a /= b) = ");
        document.write(a /= b);
        document.write("<br />");

        document.write("Value of a : (a %= b) = ");
        document.write(a %= b);
        document.write("<br />");
    </script>
    <p>Set the variables to different values and different operators and then try...</p>
  </body>
</html>
```

OUTPUT:

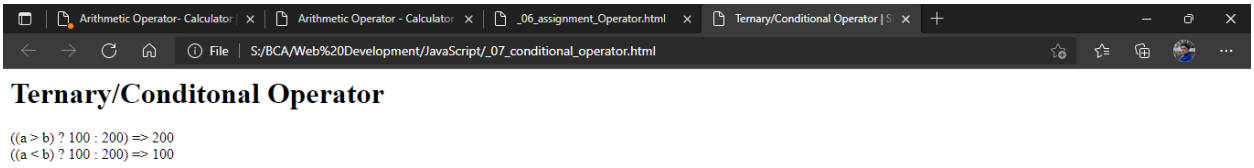


18. Conditional operator

```
<html>
<head>
<title>Ternary/Conditional Operator | Shubham Dahiya</title>
</head>
<body>
```

```
<h1>Ternary/Conditonal Operator</h1>
<script type = "text/javascript">

var a = 10;
var b = 20;

document.write ("((a > b) ? 100 : 200) => ");
result = (a > b) ? 100 : 200;
document.write(result);
document.write("<br />");

document.write ("((a < b) ? 100 : 200) => ");
result = (a < b) ? 100 : 200;
document.write(result);
document.write("<br />");
</script>
</body>
</html>
```

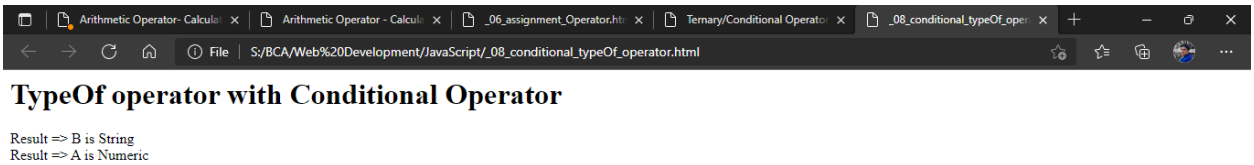
OUTPUT:



19. Conditional typeOf Operator

```
<html>
<body>
<h1>TypeOf operator with Conditional Operator</h1>
<script type = "text/javascript">

var a = 10;
var b = "Shubham";
var linebreak = "<br />";

var result = (typeof b == "string" ? "B is String" : "B is Numeric");
document.write("Result => ");
document.write(result);
document.write(linebreak);

result = (typeof a == "string" ? "A is String" : "A is Numeric");
document.write("Result => ");
document.write(result);
document.write(linebreak);
</script>
</body>
</html>
```

OUTPUT:



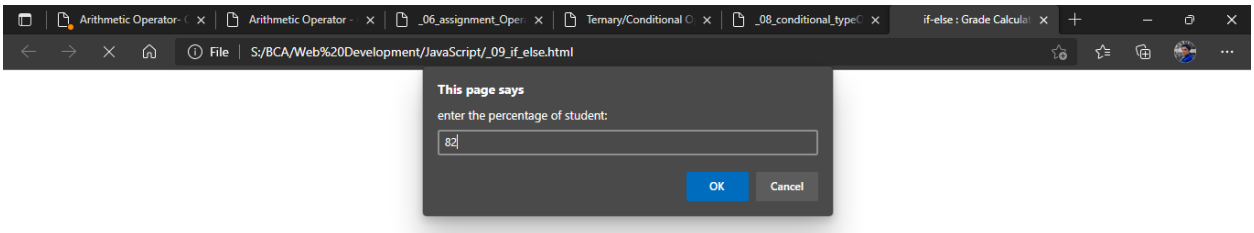
If else program

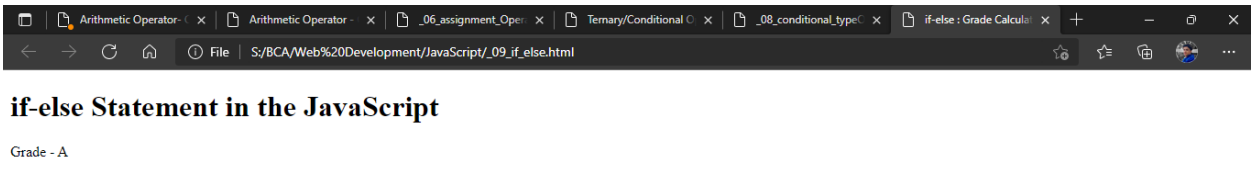
```
<!DOCTYPE HTML>
<html>
<head>
<title>if-else : Grade Calculator | Shubham Dahiya</title>
</head>
<body>
<h1>  if-else Statement in the JavaScript</h1>
<script type="text/javascript">
/*
if percentage >85 ->A
->A- if %<=85 and >80
->B if %<=80 and >70
->C if %<=70 and >60
->D if %<=60 and >40
->E if %<=40 and also should print 'candidate failed' if %<=35
Input
enter the percentage of student:
```

```
86
Output
Grade - 'A'
*/

var msg = 'enter the percentage of student:';
//msg for user input
var a = prompt(msg);
if(a>80)
{
document.write('Grade - A');
}

else if(a>70 && a<=80)
//message variable
{
document.write('Grade - B');
}

else if(a>60 && a<=70)
{
document.write('Grade - C');
}
else if(a<=60 && a>35)
{ document.write('Grade - D'); }
else if(a<=35)
{
document.write('Grade - E Candidate failed');
}
```

```
</script>
</body>
</html>
```

OUTPUT:





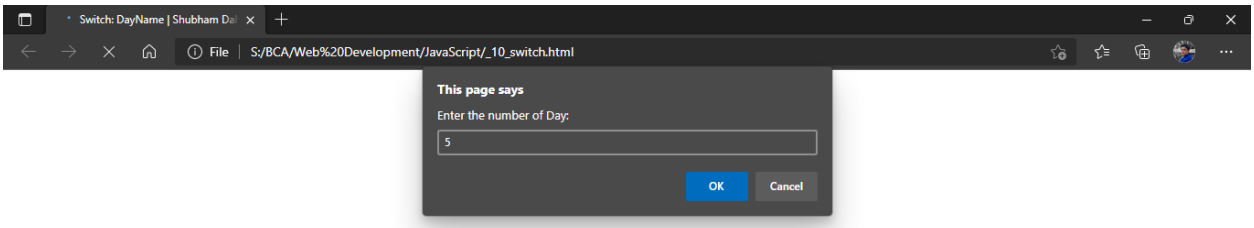
20. Switch statement program

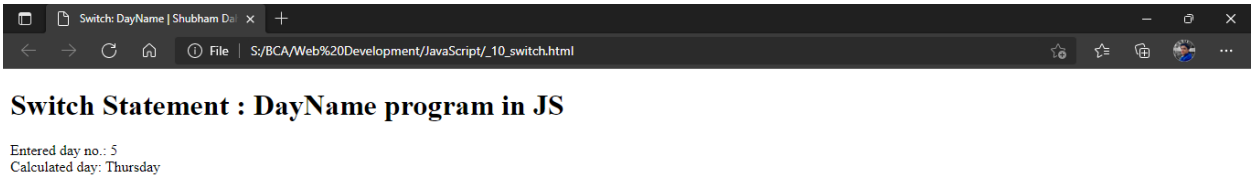
```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
<title>Switch: DayName | Shubham Dahiya </title>
</head>
<body>
<h1>Switch Statement : DayName program in JS</h1>

<script>

let day = prompt("Enter the number of Day: ");
day = parseInt(day)

let dayName;

switch (day) {
case 1:
dayName = 'Sunday';
break;

case 2:
dayName = 'Monday';
break;

case 3:
dayName = 'Tuesday';
break;

case 4:
dayName = 'Wednesday';
break;
```

```
case 5:
dayName = 'Thursday';
break;

case 6:
dayName = 'Friday';
break;

case 7:
dayName = 'Saturday';
break;

default:
dayName = 'Invalid day';
}

document.write("Entered day no.: " + day + "<br /> Calculated day: " + dayName)
</script>
</body>
</html>
```

OUTPUT:



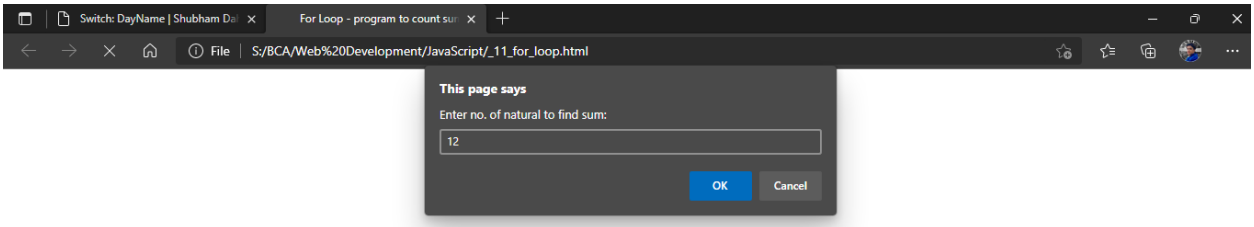


21. For loop

```
<!DOCTYPE html>

<html lang="en">

<head>

<title>For Loop - program to count sum of n natural numbers | Shubham Dahiya</title>

</head>

<body>

<h1>For Loop - program to count sum of n natural numbers</h1>

<script>
```

```
// program to display the sum of n natural numbers
let sum = 0;
let n = prompt("Enter no. of natural to find sum: ");
n = parseInt(n)

for(let i = n; i >= 1; i-- ) {
// adding i to sum in each iteration
sum += i; // sum = sum + i
}

document.write("Sum of " + n +" natural number is: " + sum)
console.log('sum:',sum);
</script>
</body>
</html>
```

OUTPUT:



For Loop - program to count sum of n natural numbers

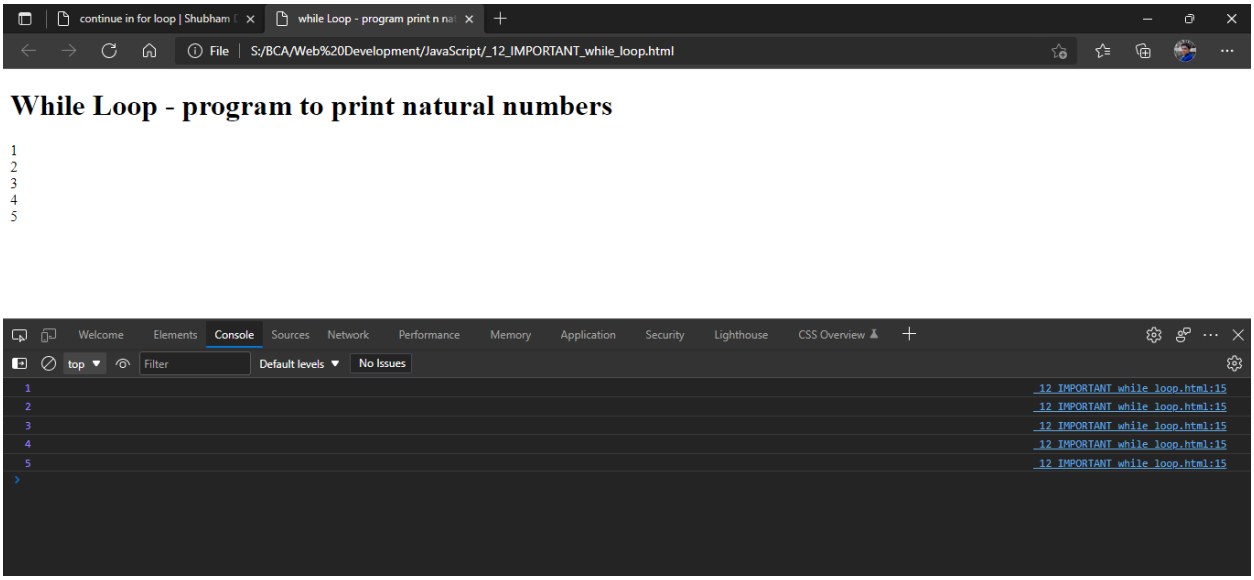
Sum of 12 natural number is: 78

22. While loop

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>while Loop - program print n natural numbers | Shubham Dahiya</title>
</head>
<body>
<h1>While Loop - program to print natural numbers </h1>
<script>
// program to display numbers from 1 to 5
// initialize the variable
let i = 1, n = 5;

// while loop from i = 1 to 5
while (i <= n) {
console.log(i);
document.write(i + "<br />")
i += 1;
```

```
}


</script>
</body>
</html>
```

OUTPUT:

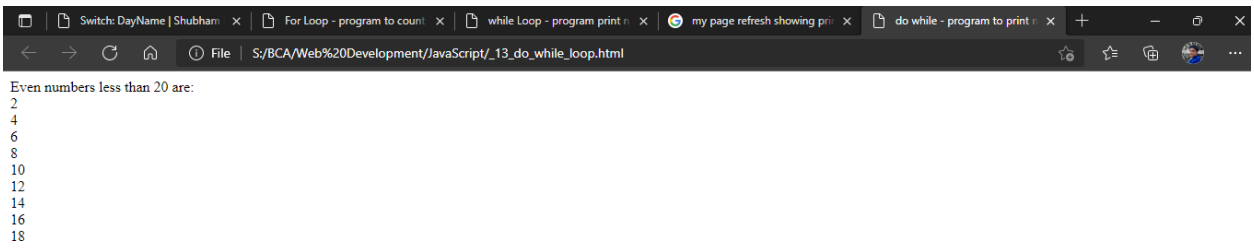


23. Do while loop

```
<html>
<head>
<title>do while - program to print number less than 20 | Shubham Dahiya</title>
</head>
<body>
<h1></h1>
<script type="text/javascript">
```

```
var i = 2;
document.write("Even numbers less than 20 are: <br />");
do
{
document.write(i + "<br />");
i = i + 2;
}while(i<20)
</script>
</body>
</html>
```

OUTPUT:

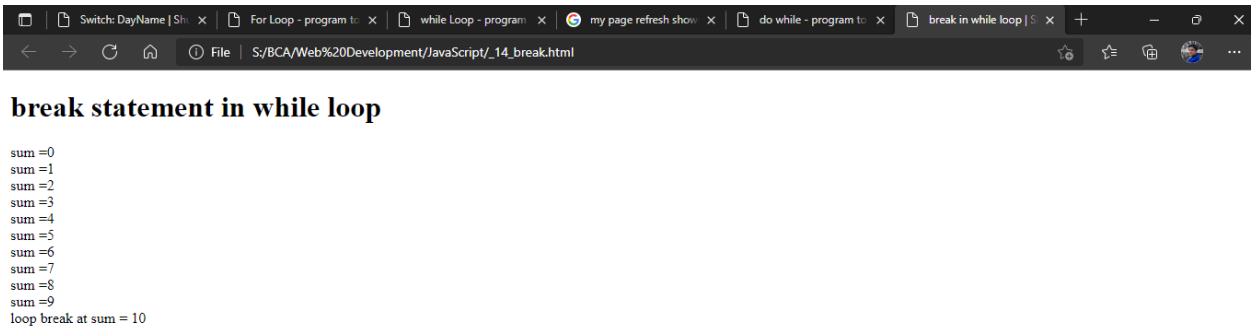


24. Break Statement

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>break in while loop | Shubham Dahiya</title>
```

```
</head>
<body>
<h1>break statement in while loop</h1>
<script>
let sum = 0;
while(sum < 20) {
if(sum == 10) {
document.write("loop break at sum =  " + sum);
break;
}
document.write(`sum =${sum} <br/>`)
sum += 1;
}
</script>
</body>
</html>
```

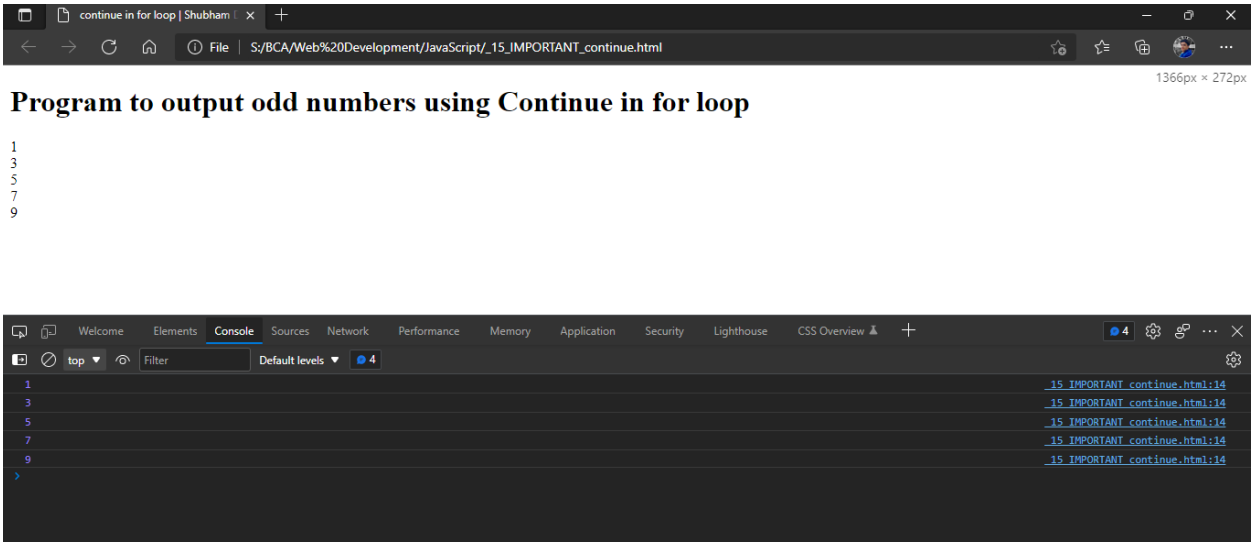
OUTPUT:



25. Continue statement – Program to output odd numbers

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>continue in for loop | Shubham Dahiya</title>
</head>
<body>
<h1>Program to output odd numbers using Continue in for loop </h1>

<script>
for (let i = 0; i < 10; i++) {
if (i % 2 === 0) {
continue;
}
console.log(i);
document.write(i + "<br />")
}

</script>
</body>
</html>
```

OUTPUT:



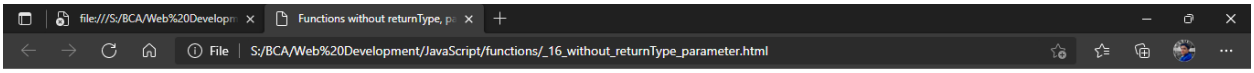
Functions in JS

26. Function without return type, without parameter

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Functions without returnType, parameter | Shubham Dahiya</title>
</head>
<body>
<h1>Functions without returnType, parameter - Program to find area of square</h1>
<script>
//variable declarations
```

```
var side = 20;
//function declaration
document.write("side of square: " + side + "<br/ >")
function area_of_square()
{
var area = side * side;
document.write("Area of the Square =" + " " + area);
}
//function call
area_of_square();
</script>
</body>
</html>
```

OUTPUT:



Functions without returnType, parameter - Program to find area of square

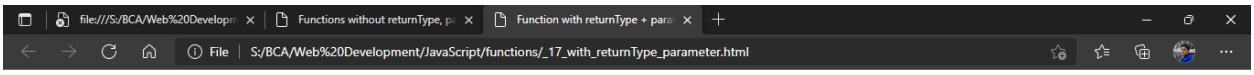
side of square: 20
Area of the Square = 400

27. Function with return type and parameter

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
<title>Function with returnType + parameter | Shubham Dahiya</title>
</head>
<body>
<h1>Function with returnType + parameter</h1>
<script>
function factorial(num) {
if (num == 0)
return 1
let x = num - 1;
while (x > 1) {
num *= x;
x--;
}
return num;
}

let n = 8

document.write(`Factorial of ${n} is: ${factorial(n)}`)
</script>
</body>
</html>
```

OUTPUT:



Function with returnType + parameter

Factorial of 8 is: 40320

28. Date Methods

```
<!DOCTYPE html>
<html lang="en">
<head>

<title>Date - time class + methods | Shubham Dahiya</title>
</head>
<body>
<h2>Date and time Methods</h2>
<script>
var date= new Date();
document.write(date.getDate() +"<br>")
document.write(date.getFullYear()+"<br>")
document.write(date.getMonth()+"<br>")
document.write(date.getHours()+"<br>")
document.write(date.getMilliseconds()+"<br>")
document.write("apart of getting we can also set the values for the date class."+"<br>")
```

```
date.setDate("12")
document.write("after setting the date date is now"+date.getDate()+"<br>")
date.setMonth("9")
document.write( "after setting the month, the month is "+date.getMonth()+"<br>")

</script>
</body>
</html>
```

OUTPUT: