India's Most Popular
Online IT Championship

28 States, Hundreds of School,
Thousands of Students, One Platform

●●●●

Preparation & Reference Book

Category - Senior

Class : XII

www.itolympiad.in

CITIS Infotech : Knowledge Partner

JavaScript

Chapter 1 Introduction

JavaScript or ‘JS’ in short is an interpreted or just in time programming language is also known as scripting language.

JavaScript is used to program the behaviour of web pages.

JavaScript can be used with many desktop and server program. Some databases like mongoDB and couch DB also use JavaScript as their programming language

I) Javascript in HTML File

```
<!doctype html>

<html>

<head>

<title>My Web Page</title>

<script>

  alert('hello world!');

</script>

</head>

</html>
```

Insert first image

II) HTML File and JavaScript File Linking

First program in JavaScript i.e. Hello world & How to add JS in HTML file

Steps to follow

1. First create a folder in your project directory name it as a JS
2. Create new file and name it as a Hello world. JS and save it in your JS folder.
3. We need to connect JavaScript to html file so we need to add following line in our <body> tag before closing </body> tag

```
< script src = “JS/Helloworld.js”> </script>// JS is the folder which we created
```

4. After that add following code in helloworld.js

```
var myText=document.querySelector (‘p’);

  myText.textContent = “Hello world!’;
```

5. Done! Now load Helloworld.html in the browser.

HTML code reference

```
<! DOCTYPE html>
<html>
<head>
<title> Hello word! </title>
</head>
<body>
<P> </ P>
< script src = “JS/Helloworld.js’> </script>
</ body>
</ html>
```

Do It your self

Try to develop simple script using above code.

Chapter 2 Basics of JavaScript

While scripting code we have to clear with few basic concepts. Then it will be easy for you to write code in JavaScript.

Variable

Variable is used to store the value in it. In JS we can declare variable by using var keyword. Followed by any name you want.

Syntax

var variablename;

For example

var myvariable;

After declaring a variable, you can give value to it.

For example

var myvariable = ‘Radar’;

You can declare many variables in one line one statement. Start variable with ‘var’ and separate it with comma.

For example

Var name = ‘Ramesh’, age = 29;

Surname = ‘Jadhav’;

Rule for constructing name for variable –

1. use letters, digits, underscores and dollar signs.
2. Name must begin with letter also begin with $(Doller), &(ampersand), _(underscore)
3. In JS names are case sensitive myvariable and myVariable are different.

Data types in JavaScript

Data types are used to specify type of data. There are following data types available in JavaScript.

1. String
   In JavaScript value written in double or single quotes is treated as a string.

2. Number

Numbers are written without quotes. But if you put number in quotes it is treated as a text string.

3. Boolean – This is the true and False are special keywords in JS and don't need quotes.
   For example:
   var myVariable = true;

Operator in Javascript

Operators are used to operate variables and generate output.

1. **Assignment operator <=>**

   Assignment operator is used to assign value to the variable.

   For example:

   Var name = "Rahul";

   Var x = 2; // assign the value 2 to X

2. **Addition operator (+)**

   Addition operator are used to add two value or concatenate to string together

   For example:

   a. Adding two numners

   var x = 2;

   var y = 3;

   var z = x + y;
   b. Concatenate two strings.
   var name = "Rahul";
   var surname = "Rai";
   var fullName= name + '" +surname;

So, the answer will be that is variable fullName Rahul Rai

3. **Subtraction operator (-)**
   Used to subtract one value from another value

4. **(*) multiplication**
   Used to find multiplication of the table.
5. **Division (/)**
   Used to divide one value by another

6. **Increment operator (++)**
   This operator is used to increment value by one.
   Example:
   Var x = 5;
   X + +;

7. Decrement operator (- -)

This operator is used to decrement value by one.
Example:
Var x = 5
X - -;

JavaScript operator precedence

Multiplication and division (/) have higher precedence than addition (+) and subtraction (-).

8. Typeof ( ) operator :

Typeof ( ) operator is used find the which type of variable is.
Typeof " Rahul" //Returns string
Typeof 2 // returns number.

9. Logical operator:

Logical operator works on the Boolean values.
&& - AND operator – Both sides need to be true.
|| - OR operator – one side need to true.

Conditional block:

If ---- else block is used to check condition and returns answer when condition written in "if" is true otherwise control proceeds to the else part.

Syntax:

```
if(condition)
{
Statements;
}
Else
{
Statements;
}
```

For example :

```
var carName = "Maruti";
If (carName = = "Maruti")
{
Alert ("I like Maruti cars');
}
else
{
Alert ("I don't like Maruti cars');
}
```

In above code break in if statement checks carName if it is true then first block of code is run but if condition is false then else block is run instead.

Practical of logical operators using conditional block
For example

```
Var x = 5;
Var y = 6;
Var z = 7;


if ( x = = 5 && y = = 6 )
 {
        alert (“ First block is true”);
        if ( x>5 || z == 7 )
        {
                alert (“ second block is true “);
        }
        Else
         {
                alert ( “All statements ‘are wrong’),
        }
}
```

Comments
Every programming language allows you to add notes or explanation of code, which called comment. Comments have no effect on the layout of the document. Commented code does not execute.

1. To put single line comment in JS we can use ‘//’

For example:
// This single line comments.

2. To put multiline comment, we use /* -----*/

To put multiline comment, we use /* this is multiline comment */

Chapter 3. Loops

Loop is also known as iteration. Loop is used to repeat the same task again again.

While loop :-

While loop is to execute a code block repeatedly till expression is true. If the expression Become false, while loop terminates.

Syntax :-

```
While (condition/expression)
{
Statement is execute till the expression is true.
}
```

For example :

```
var counter = 10;
document. Write (“while loop example : “)
While (counter <20)
{
Document. Write (“ counter “ + counter + <br>
Counter + + ;
}
Document .write (“while loop terminates”)
```

Do… while loop :

The do … while loop is same as while loop but in this code block is executed at least ones, even if the condition is false. In do … while loop condition check happens at loop ends.

Syntax :

```
Do {
Expression/statement to be executed.
}while (condition/expression)’
```

Note: Don’t forget to give semi – colon at the end of do … while loop

For example

```
var counter = 10;
Document.write (‘do…while example’);
do {
document.write (“counter:” + counter “</br>”);
counter ++;
}while (counter <10);
document.write(“while loop terminates”);
```

For loop: -

For loop is divided into three parts initialization, test condition, iteration.

Initialization – is a starting point of loop. The initialization execute before loop begins.

Test condition – Test condition, will test give condition is true of false. If the condition is true. It runs the statement given in the loop. If condition is false the control goes out of the loop.

Iteration – Iteration is used to increase or decrease counter.

Syntax of for loop –

```
For (initialization; test condition; iteration)
{
Statement is executed if test condition true;
}
```

Example:

```
var counter;
For (counter = 0; counter <10; counter ++)
{
document.write (“counter” + counter + “</br>”);
}
document.write (“for loop terminates”);
```

Do it your self

1. Write a code for addition of any two numbers.
2. Write code to check whether the number is odd or even.
3. Find the given number is prime or not.
4. Write code for addition of only even numbers from 1 to50.

Introductions to Object

Java Script is an object-oriented programming language. Means the language offers all object-oriented facilities to the developer like Encapsulation, Polymorphism, Inheritance etc.

An object is an instance of a particular class or subclass. objects are run time entity. Objects may be a person, a car, an account, a bicycle, a fruit anything or any item.

Basically, objects are the variables of type class. Objects interact with each other by sending messages or requests. For any class we will be able to create any number of objects.

e.g. Tomato, spinach, cucumber, cauliflower are members of class vegetables

If 'student' and 'teacher' are two objects of class student and class teacher respectively. Then student object may request to object teacher for the notes of the particular subject.

Each object is having some properties and some methods

For example

Object is vegetables	Properties	Methods
	Vegetable.name = Cabbage	Vegetable. wash()
	Vegetable.Colour = Green	Vegetable. cut()
	Vegetable. Weight = 1Kg	Vegetable.cook()

Objects in Java Script.

In JavaScript if you understand an object that means you understand JavaScript. In JS the data types like Boolean, Numbers, Strings can be objects. With the help of ‘New’ keyword we will be able to create the objects.

Date, Math, Arrays, Strings, Functions are always object, each object is having its own attributes and functions. Java Script objects are collection of the name - value pair

For Example, Object Girl

Properties	Value
Name	Aarti
Hair colour	Black
Eyes colour	Brown

Creating objects in JavaScript

We will be able to create objects in JS by different ways

- By using ‘new’ keyword

```
var employee = new Object();
employee.firstName = "ajay";
employee.lastName = "Dev";
employee.age = 40;
employee.hairColor = "black";
```

- By using literals that is defining name value pairs for example

```
var employee = {firstName:"ajay”, lastName:"Dev”,age:40, hairColor:"black"};
```

- By using constructor

The constructor is the method for creating an object with new keyword. They are called once in a lifetime. In java script special object() function is provided to create the object.

```
<html>
  <head>
    <title>Objects creation</title>
    <script type = "text/javascript">
      var employee= new Object(); // Create the object
      employee.name = "Prasad";     // Assign properties to the object
      employee.age = "29";

var employee1 = new Object();   // Create the object
      employee1.name = "Ram";     // Assign properties to the object
      employee1.age = "38";
    </script>
  </head>

  <body>
    <script type = "text/javascript">
      document.write("employee name is : " + employee.name + "<br>");
      document.write("employee age is : " + employee.age + "<br>");

       document.write("employee1 name is : " + employee1.name + "<br>");
      document.write("employee1 age is : " + employee1.age + "<br>");
    </script>
  </body>
</html>
```

Result is

```
employee name is :Prasad
employee age is:29
employee1 name is :Ram
employee1 age is: 38
```

Chapter 5. JavaScript Array

JavaScript array is an object that represents a collection of similar type of elements.

In JavaScript we can construct array in 3 ways.

1. By array literal
2. By creating object of Array directly (using new keyword)
3. By using an Array constructor (using new keyword)

```
var arrayname=[value1,value2.....valueN];
```

As you can see, values are contained inside [ ] and separated by , (comma).

example of creating and using arra

```
<script>

var stud=["Varad","Naman","Pradnya",”Tanuja”];

for (i=0;i<stud.length;i++){

document.write(stud[i] + "<br/>");

}

</script>
```

Note. The “.length” is aproperty of array object and this property returns the length of an array.

Output of the above example

```
Varad
Naman
Pradnya
Tanuja
```

Using New Keyword

Using new keyword also we can construct an array

syntax

```
var arrayname=new Array();
```

Here, **new keyword** is used to create an object of array.

Example:

```
<script>

var i;

var stud = new Array();

stud[0] = "Ram";

stud[1] = "Sham";

stud[2] = "Param";

 stud[3] = "pooja";


for (i=0;i<stud.length;i++){

document.write(stud[i] + "<br>");

}

</script>
```

Output:

```
Ram

Sham

Param
```

Pooja

Create array using constructor

Here, you need to create instance of array by passing arguments in constructor so that we don't have to provide value explicitly.

Example:

```
<script>

var i;

var stud = new Array("Ram", "Sham", "Param", "Pooja");

for (i=0;i<stud.length;i++){

document.write(stud[i] + "<br>");
}
</script>
```

Output:

```
Ram
Sham
Param
Pooja
```

Methods	Description
concat()	It returns a new array object that contains two or more merged arrays.
copywithin()	It copies the part of the given array with its own elements and returns the modified array.
find()	It returns the value of the first element in the given array that satisfies the specified condition.
indexOf()	It searches the specified element in the given array and returns the index of the first match.
join()	It joins the elements of an array as a string.
lastIndexOf()	It searches the specified element in the given array and returns the index of the last match.
sort()	It returns the element of the given array in a sorted order.

Chapter 4.Function in JavaScript

Introduction

A function is a group f reusable code. This can be eliminating writing same code again and again. Main use of Function in JavaScript is reusability of code that is defining the code once ad use it many times. We can call a function is a subprogram or a block of code which is used to perform particular task.

In earlier our chapter you have seen alert () and write() functions but they been written in care of JavaScript only once.

JavaScript allows us to write our own function as well.

Defining function

In the case of own function, before using we have to define it. In JavaScript "function" Keyword is used to define function (), followed by function_name and statements surrounded by curly braces.

Syntax:

```
function function-name(Parameter list)
{
//statements;
}
```

For example:

```
<Script type="text/JavaScript">
function Mymsg()
{
Alert("Hello, This is Function);
}
</script>
```

Function Call

To use the function actually we have to call the function.

See the following code to call fuction,In Html body write following code To Display message Click the button.

```
<button onClick="MyMsg()"> Click button</Buttton>
```

And load the Html Page in the browser.

Function Parameter

We have seen function without parameter but sometimes functions may have parameters. We can pass different parameters separated by comma to function. When function get call these parameters captured inside function and processed them and generate output, which further returns to the function call statement.

Function example

```
<html>
  <head>

    <script type = "text/javascript">
      function myName(name, surname)
      {
        document.write ( " first name is " + name + " Surname is "+ surname);
      }
    </script>

  </head>
  <body>
    <p>Click the following button to call the function</p>

    <form>
      <input type = "button" onclick = "myName('Ajay','Patil')" value = "Click me">
    </form>

    <p>Use different parameters inside the function and then try...</p>
  </body>
</html>
```

Do It your self Run the above code with different parameters.

Return statement

The return statement is used to specify the value that is returned from the function. This function is a last statement of function body and used only when function returns a value otherwise it is optional statement.

```
<script type="text/JavaScript">
```

```
function myMulti(x,y)
{
return x*y;
}
document.write(myMulti(10,10));
</script>
```

In above example we have passed two numbers in a function and the function returns multiplication of those numbers.

DO It your self

Create an addition function and call that function

Using function write a code for addition of numbers within a range.

Using function check even odd numbers between 100 to 400

Introductions to Object

Java Script is an object-oriented programming language. Means the language offers all object-oriented facilities to the developer like Encapsulation, Polymorphism, Inheritance etc.

An object is an instance of a particular class or subclass. objects are run time entity. Objects may be a person, a car, an account, a bicycle, a fruit anything or any item.

Basically, objects are the variables of type class. Objects interact with each other by sending messages or requests. For any class we will be able to create any number of objects.

e.g. Tomato, spinach, cucumber, cauliflower are members of class vegetables

If 'student' and 'teacher' are two objects of class student and class teacher respectively. Then student object may request to object teacher for the notes of the particular subject.

Each object is having some properties and some methods

For example

Object is vegetables	Properties	Methods

Vegetable.name = Cabbage	Vegetable. wash()
Vegetable.Colour = Green	Vegetable. cut()
Vegetable. Weight = 1Kg	Vegetable.cook()

Concepts of creating objects in Java Script.

In JavaScript if you understand objects that means you understand JavaScript. In JS the data types like Boolean, Numbers, Strings can be objects. With the help of 'New' keyword we will be able to create the objects.

Date, Math, Arrays, Strings, Functions are always object, each object is having its own attributes and functions. Java Script objects are collection of the name - value pair

For Example, Object Girl

Properties	Value
Name	Aarti
Hair colour	Black
Eyes colour	Brown

Creating objects in JavaScript

We will be able to create objects in JS by different ways

- By using ‘new’ keyword

```
var                                   employee                                  = new Object();
employee.firstName = "ajay";
employee.lastName = "Dev";
employee.age = 40;
employee.hairColor = "black";
```

- By using literals that is defining name value pairs for example

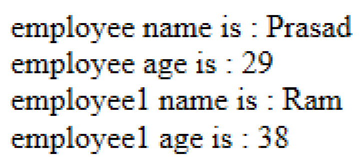
```
var employee = {firstName:"ajay”, lastName:"Dev”,age:40, hairColor:"black"};
```

- By using constructor

  The constructor is the method for creating an object with new keyword. They are called once in a lifetime. In java script special object() function is provided to create the object.

```
<html>
  <head>
    <title>Objects creation</title>
    <script type = "text/javascript">
      var employee= new Object(); // Create the object
      employee.name = "Prasad";    // Assign properties to the object
      employee.age = "29";

var employee1 = new Object();   // Create the object
      employee1.name = "Ram";    // Assign properties to the object
      employee1.age = "38";
    </script>
  </head>

  <body>
  <script type = "text/javascript">
   document.write("employee name is : " + employee.name + "<br>");
   document.write("employee age is : " + employee.age + "<br>");


     document.write("employee1 name is : " + employee1.name + "<br>");
   document.write("employee1 age is : " + employee1.age + "<br>");
  </script>
 </body>
</html>
```

Result is



```
employee name is : Prasad
employee age is : 29
employee1 name is : Ram
employee1 age is : 38
```

Chapter 5 Events in JavaScript

Introduction

Events are action that can be performed. The JavaScript events are normally used in combination of function.

See the following example of OnClick() event.

This event occurs when user click on an element.

Syntax:

<element onClick=”JavaScipt code”>

Example

```
<html>

  <head>

    <script type = "text/javascript">

      <!--

        function myFirstEvent() {

          alert("Hi First Event")

        }

      //-->

    </script>

  </head>


  <body>

    <p>Click the following button and see result</p>

    <form>

      <input type = "button" onclick = "myFirstEvent()" value = "say Hi" />

    </form>

  </body>

</html>
```

JavaScript Code is in alertmsg.js:

```
function AlertMsg()

{

Alert("This is on Click Event");

}
```

Likewise we can use several available events from JavaScript.

Some Events From JavaScript

1. **onDoubleClick event:**
   This event triggers when user double clicks the element
2. **onload event:**
   This event triggers when object is loaded.
3. **onMousedOver event:**
   This event triggers when we bring mouse cursor over any element.
4. **onMouseOut event:**
   This event triggers when we bring mouse cursor over any element.
5. **Unload**
   This event triggers when the user leaves the page.
6. **Reset**
   This event triggers when the user leavers the form using the reset button.
7. **Submit**
   This event triggers when the form is submitted.
8. **Error**
   This event triggers when the error occurs during loading of document /image.
9. **Focus**
   This event triggers when input focus is given to a form element.

Chapter6

Java Script Page redirect

Page redirection is also called as URL redirection or URL forwarding.

Using this way, we will be able to redirect a web page to another web page. The redirected page is mostly from the same website, or from different web site or may be on different web server.

JavaScript URL redirection

In JavaScript, we have different methods which are related to window object, that is a property of the Window object is window.location. Using this we will be able to make redirection by getting current URL and by redirecting it to new web page.

Page redirection means to open a desired web page by clicking on the current web page URL

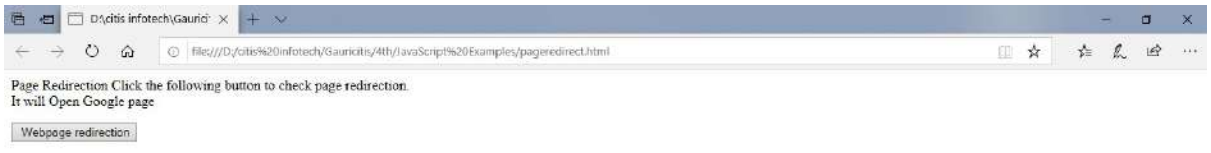
URL Redirection Practical's

1 Page Redirect In this example we will try to open the www.google.com page

```
<html>
  <head>
        <script type="text/javascript">
      <!--
        function Redirect() {
          window.location="https://www.google.com";
        }
      //-->
    </script>
      </head>
    <body>
    <p>Page Redirection Click the following button to check page redirection.<br>
It will Open Google page</p>
    <form>
      <input type="button" value=" Webpage redirection" onclick="Redirect();" />
    </form>


  </body>
</html>
```

Ans Screen

Insert pageredirect image

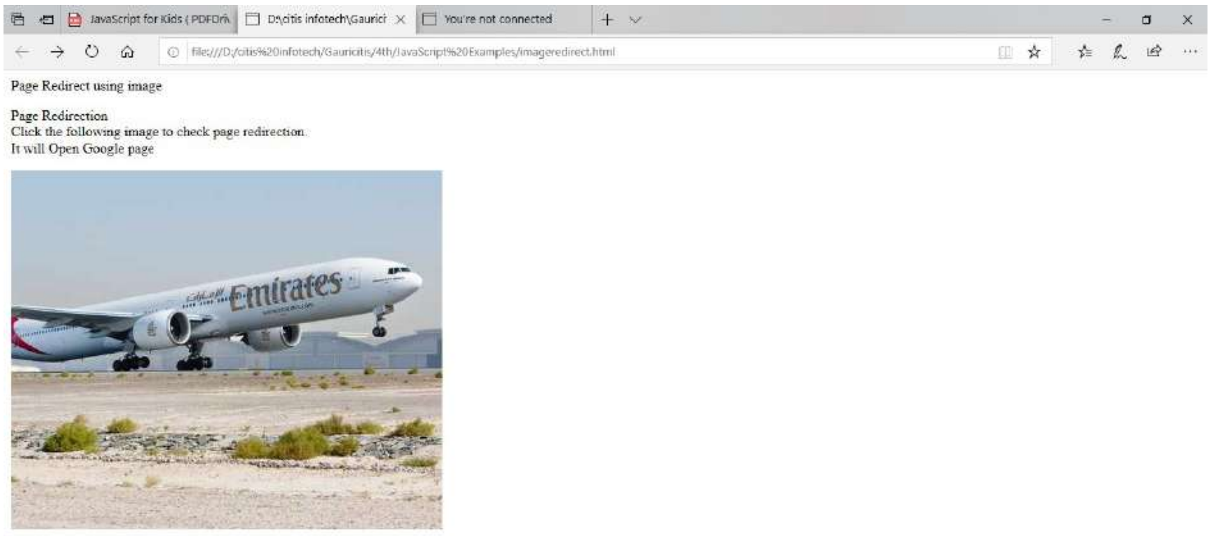

D:\citis infotech\Gauric
file:///D:/citis%20infotech/Gauricitis/4th/JavaScript%20Examples/pageredirect.html
Webpage redirection

Page Redirection Click the following button to check page redirection.
It will Open Google page

Following is the example of page redirection using image

```
<html>
  <head>

    Page Redirect using image

  </head>

  <body>
    <p>Page Redirection <br>Click the following image to check page redirection. <br>
It will Open Google page</p>

    <form>

<a href="https://www.google.com">
```

```
<img src="BOEING 777.JPG"></img>
    </form>


  </body>
</html>
```


JavaScript for Kids (PDFDri
D:\citis infotech\Gauricit
You're not connected
file:///D:/citis%20infotech/Gauricitis/4th/JavaScript%20Examples/imageredirect.html

Page Redirect using image

Page Redirection
Click the following image to check page redirection
It will Open Google page

Tip → href stands for hypertext reference, it's a stylish way to say "web address."

Do It your self

1. Try to run above code using different images and different destination address.
2. Try to develop the web pages and open another web page from current page.
3. What is URL Stands For?

Chapter 7.

Error and Exceptions

Issues that arises unexpectedly that cause a software to not function properly is called as an error.

There are three types of error

Logical Errors

Syntax Errors

And Runtime Errors

When Programmer violate in rules while writing code in any programming language are called as syntax errors.

Following is an example of a syntax error

<script type = "text/javascript → The syntax error occurs here as anchor '>'tag and double quotes (")are missing

```
  <!--
    window.print() → here also semicolon is missing(;)
  //-->
</script>
```

This type of errors is also called as Parsing errors.

Logical errors

These errors are very difficult to catch. These are due to logical errors in the code and we due to which we will not get the expected result.

e.g. If we want to add two numbers like 5+3 the result will be 8

But by mistake we wrote 5-3 then the result will be 2 which is not as per expectation.

Same mistake user may do while writing text, numbers, code etc.

Runtime Errors

This type of errors is also called as exception errors which occurs at the runtime that is at the execution of program. May be at compile time or runtime.

To catch exception, we have exception handling in the Java Script.

JavaScript provides the **try...catch...finally** construct and the **throw** operator to deal with exceptions.

In exception, handling we will be able to catch the runtime errors but not syntax error.

Try block should always followed by at least one catch block or finally block or both.

We write our code in try block where exception may occur and we write catch block to catch particular exception. Finally block gets always executed.

```
try{
   Somefunction()
   alert('error may occur')
}
catch(e){
   alert('An error has occurred: '+e.message)
}
```

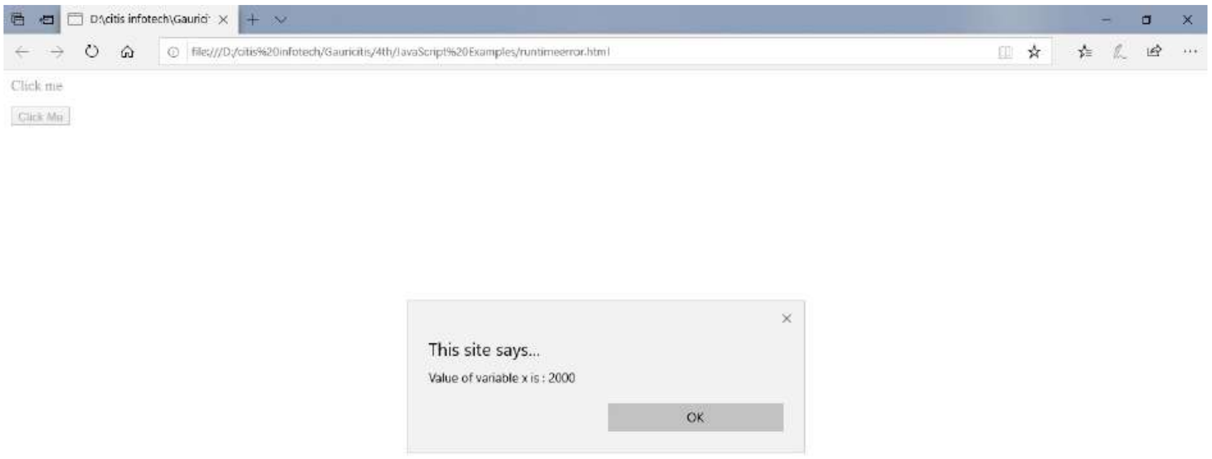
Example of exception handling

```
<html>

  <head>

    <script type = "text/javascript">

      <!--

        function testfunction() {

          var x = 2000;

                try{

          alert("Value of variable x is : " + x );


                }

                catch (e)

                alert("Error"+e.description);

        }

      //-->

    </script>

  </head>


  <body>

    <p>Click me</p>


    <form>
```

```
         <input type = "button" value = "Click Me" onclick = "testfunction();" />
      </form>
   </body>
</html>
```

Result of

Insert runtimeerror image



This site says...

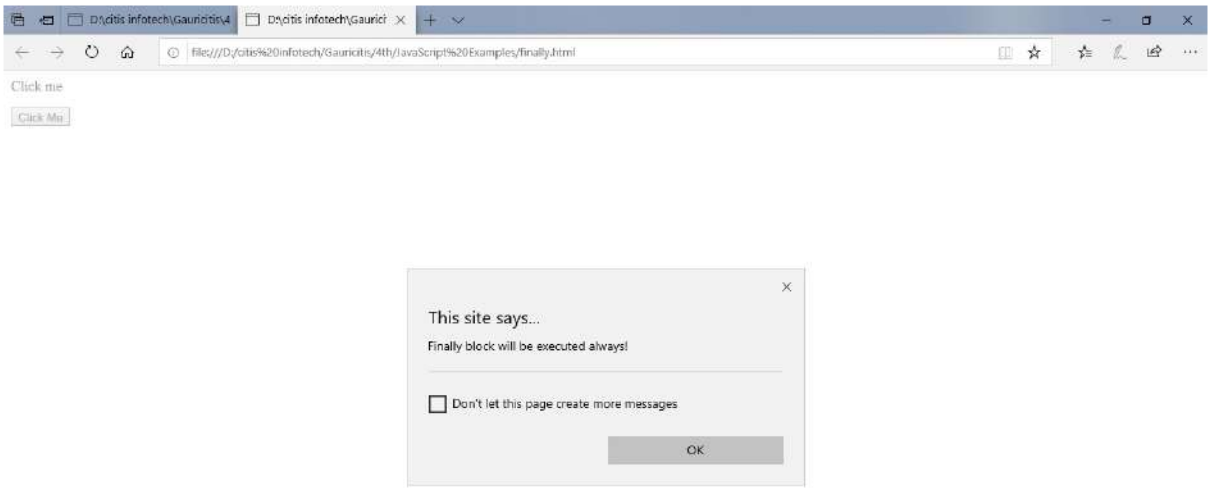
Value of variable x is : 2000

Now in the above code and insert finally block in it and execute the code

```
<html>
   <head>
      <script type = "text/javascript">
         <!--
            function testfunction() {
               var x = 2000;
                        try{
               alert("Value of variable x is : " + x );


                        }
```

```
                catch(e)
                {
                alert("Error"+e.description);
                }
                finally {
            alert("Finally block will be executed always!" );
          }
        }

      //-->
    </script>
  </head>

  <body>
    <p>Click me</p>

    <form>
      <input type = "button" value = "Click Me" onclick = "testfunction();" />
    </form>
  </body>
</html>
```

Insert finally image


file:///D:/citis%20infotech/Gauricitis/4th/JavaScript%20Examples/finally.html
Click me
Click Me
OK

This site says...

Finally block will be executed always!

Don't let this page create more messages

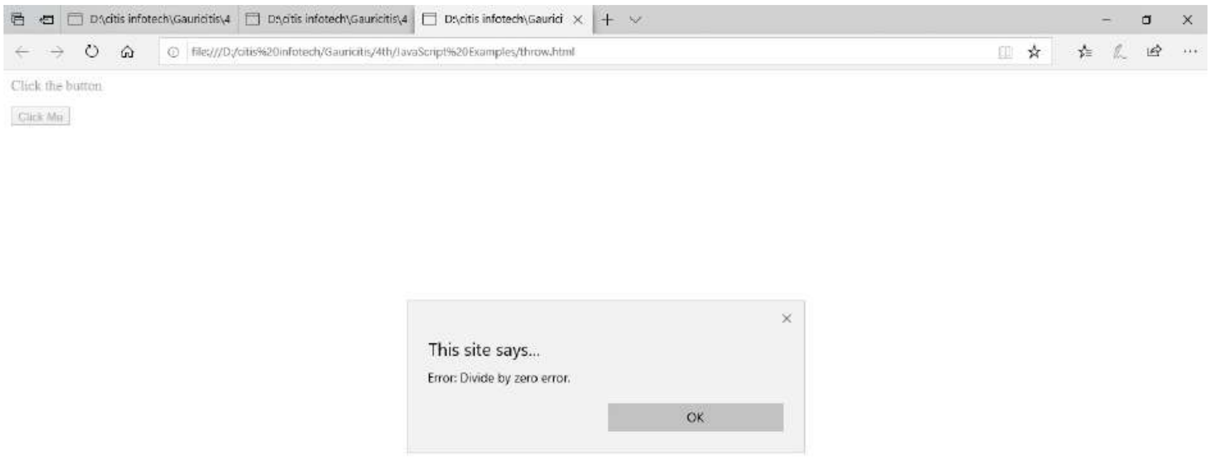
We can use throw statement for built in or customise exceptions.

Example of throw

Code

```
<html>
   <head>

      <script type = "text/javascript">
         <!--
            function testFunction() {
               var x = 50;
               var y = 0;
               try {
                  if ( y == 0 ) {
                     throw( "Divide by zero error." );
                  }

                  else {
                     var z= x / y;
                  }
               }

               catch ( e ) {
                  alert("Error: " + e );
               }
            }
         //-->
      </script>
    </head>

    <body>
      <p>Click the button</p>

      <form>
         <input type = "button" value = "Click Me" onclick = "testFunction();" />
      </form>

    </body>
</html>
```

Insert throw image


Click the button

Click Me

This site says...

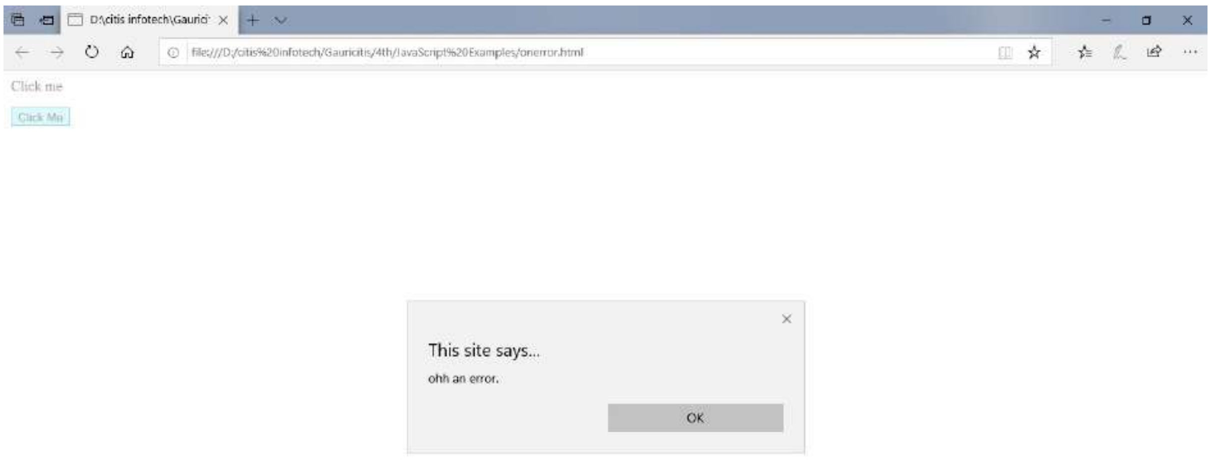
Error: Divide by zero error.

In Java Script we have function called onerror().It's called as the event handlers.

When exception occurs   exception is fired on window object

```
<html>
  <head>

    <script type = "text/javascript">
      <!--
        window.onerror = function () {
          alert("ohh an error.");
        }
      //-->
    </script>

  </head>
  <body>
    <p>Click me </p>

    <form>
      <input type = "button" value = "Click Me" onclick = "myFunc();" />
    </form>

  </body>
</html>
```

Result Insert onerror


D:\citis infotech\Gauric
file:///D:/citis%20infotech/Gauricitis/4th/JavaScript%20Examples/onerror.html
Click me
Click Me

This site says...

ohh an error.

Do it your self

In above code try to find why error occurred.

Execute addition of numbers program with exception handling.

Chapter 8

Form Validation

Validation means checking for inappropriate values.

In software it is always important to validate the elements

In short validating means to authenticate user.

In JavaScript we have facility for client-side validation that means before submitting the data to the server.

In Java Script we can validate fields like password, email id, date, numbers, name and many more.

For example.

Using Java Script, we will be able to check that

Form field data is filled or not that is empty field?

On the Numeric field it will check for numbers entered or text?

In side date it will check for valid date format.

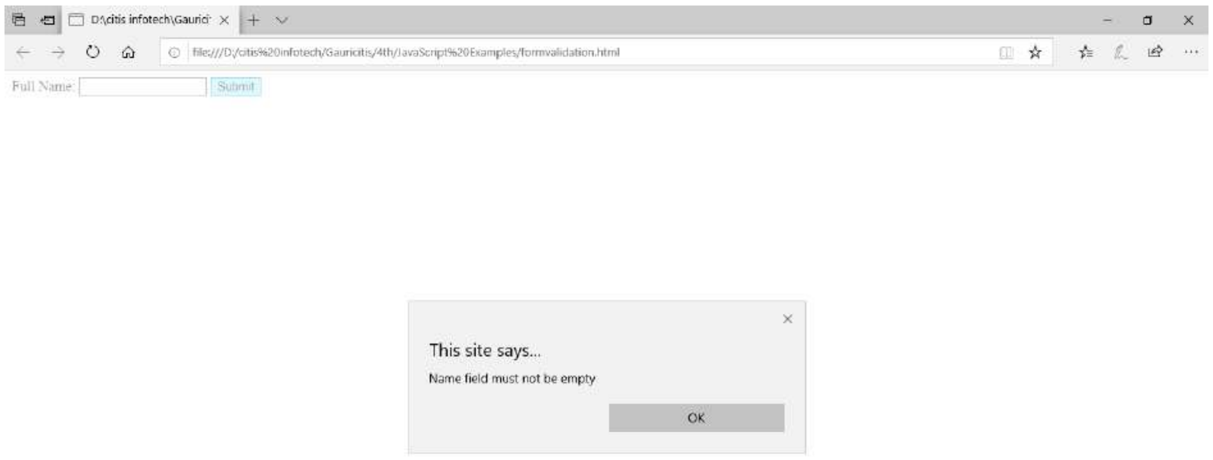
Java Script Form Validation Example.

Here we are checking that Full name field is filled or empty.

```
<html>
<head>
<script>
function formValidate() {
  var a = document.forms["form1"]["fullname"].value;
  if (a == "") {
    alert("Name field must not be empty");
    return false;
  }
}
</script>
</head>
<body>

<form name="form1" action="/action_page.php" onsubmit="return formValidate()"
method="post">
  Full Name: <input type="text" name="fullname">
  <input type="submit" value="Submit">
</form>

</body>
</html>
```

Insert fromvalidation image

Result



This site says...

Name field must not be empty

Example 2. Validate number fields

Enter numbers from 10 to 50

```
<!DOCTYPE html>
<html>
<body>

<h2>Validation for numbers</h2>

<p>Enter number from 10 and 50:</p>

<input id="number1">

<button type="button" onclick="numValidate()">Submit</button>

<p id="demo1"></p>

<script>
function numValidate() {
  var a, txt;

  // Get the value of the input field with id="number1"
  a = document.getElementById("number1").value;

  // If a is Not a Number or less than 10 or greater than 50
  if (isNaN(a) || a < 10 || a > 50) {
    txt = "enter valid number";
  } else {
    txt = " OK";
  }
  document.getElementById("demo1").innerHTML = txt;
}
</script>
</body>
</html>
```

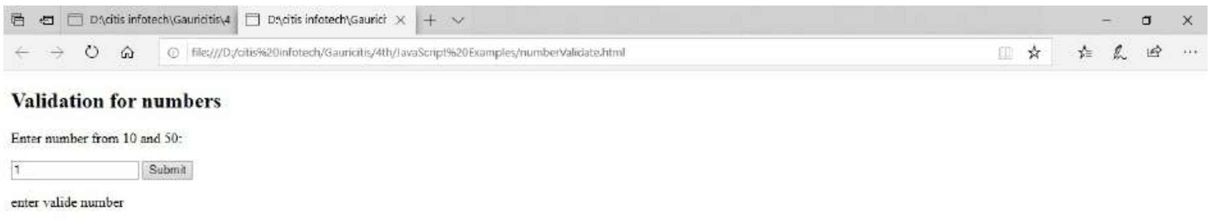
Above code contains

Tip→

The library function isNan() that is Not-a-Number. This function will retun true value if the input is number else it will return false. We have (||) or operator. This operator will return true if both or single condition is true.

```
OR operation
true || true; implies true
true || false; implies true
false || true; implies true
false || false; implies false
```

Result Insert numbervalidate image


Validation for numbers

Enter number from 10 and 50:

1 Submit

enter valide number

Do It Your self

Develop a simple form containing name and age field and validate it.

Chapter 9

Debugging in JavaScript:

While writing program developer made mistake in code. This mistake can be a logical or syntax error. This error is referred to as a bug. Many errors in programming are difficult to diagnose. To find that errors in programming code is called debugging. Modern web browsers have built-in JavaScript debugger. It can turn on or off.

Syntax:

Debugger;

The debugger keyword is used to stops the execution of JavaScript code. And calls debugging function. With debugging we can also set breakpoint where we can stop execution of JavaScript code and examine the code and after examining the code we can resume the execution of code.

- **Console.log()** – This method is used to display values in the debugger window.

Example:

```
X = “I’m debugging”;
Y=”I’m finding errors from the JS code”

console.log(X);
console.log(Y);
```

To activate the debugging window of your browser by pressing F12 and select console in debugger menu.

Chapter 10 Navigator Object

We will get information about visistor’s browser using window.navigator object

---

Window Navigator

We will be able to write window.navigator object, without the window prefix.

Some examples:

navigator.appName

navigator.appCodeName

navigator.platform

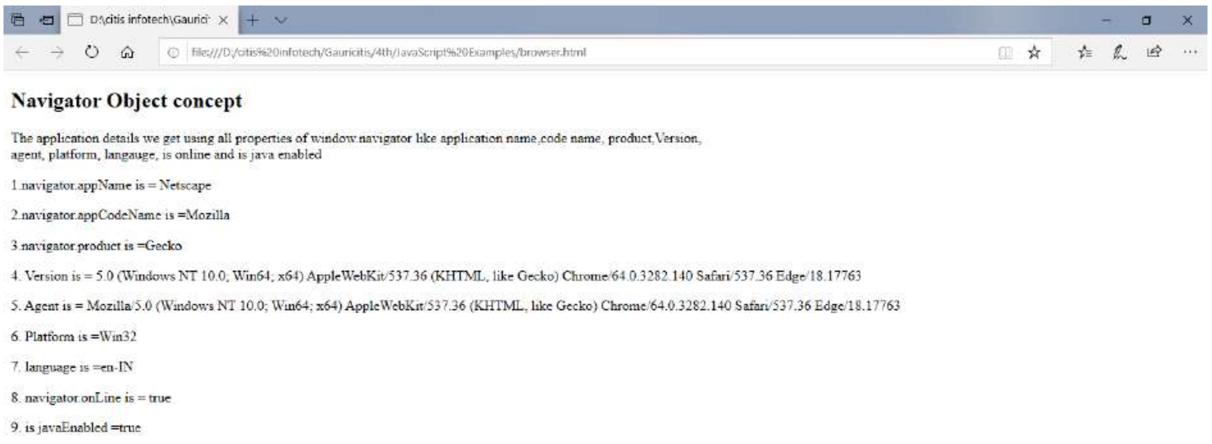
Example

```
<!DOCTYPE html>
<html>
<body>

<h2>Navigator Object concept</h2>

<p>The application details we get using all properties of window.navigator like application name,code name, product,Version,<br>
agent, platform, langauge, is online and is java enabled
 </p>

<p id="demo"></p>
<p id="demo1"></p>
<p id="demo2"></p>
<p id="demo3"></p>
<p id="demo4"></p>
<p id="demo5"></p>
<p id="demo6"></p>
<p id="demo7"></p>
<p id="demo8"></p>
<script>
document.getElementById("demo").innerHTML =
"1.navigator.appName is = " + navigator.appName;
document.getElementById("demo1").innerHTML =
"2.navigator.appCodeName is =" + navigator.appCodeName;
document.getElementById("demo2").innerHTML =
"3.navigator.product is  =" + navigator.product;
```

```
document.getElementById("demo3").innerHTML = "4. Version is = "+navigator.appVersion;


document.getElementById("demo4").innerHTML = "5. Agent is = "+navigator.userAgent;

document.getElementById("demo5").innerHTML = " 6. Platform is ="+navigator.platform;

document.getElementById("demo6").innerHTML = "7. language is ="+ navigator.language;

document.getElementById("demo7").innerHTML =

"8. navigator.onLine is = " + navigator.onLine;

document.getElementById("demo8").innerHTML = " 9. is javaEnabled
="+navigator.javaEnabled();

</script>


</body>

</html>
```

Result

Insert navigatorobject


D:\citis infotech\Gauric
file:///D:/citis%20infotech/Gauricitis/4th/JavaScript%20Examples/browser.html

Navigator Object concept

The application details we get using all properties of window.navigator like application name,code name, product,Version, agent, platform, langauge, is online and is java enabled

```
1.navigator.appName is = Netscape

2.navigator.appCodeName is =Mozilla

3.navigator.product is =Gecko

4. Version is = 5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.140 Safari/537.36 Edge/18.17763

5. Agent is = Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.140 Safari/537.36 Edge/18.17763

6. Platform is =Win32

7. language is =en-IN

8. navigator.onLine is = true

9. is javaEnabled =true
```

Do it your self

Try to practice above code in different browsers

Chapter 11.Cookies in JavaScript

Cookies are data, stored in small text files, on your computer. These cookies are sent by web server to client browser. It used to keep track of client's previous activities. This data got stored in name value pair format.

JavaScript code allows us cookies to be created, retrieved, and deleted through a simple and intuitive interface.

MCE Society's
P. A. Inamdar College of Visual Effects, Design & Art

Global Certifications

unity
Certification

CERTIPORT
A PEARSON VUE BUSINESS
AUTHORISED TESTING CENTER

Microsoft
Office Specialist

intuit quickbooks. Online
Certified User

ESB
Entrepreneurship and Small Business

IC3
DIGITAL LITERACY
CERTIFICATION

AUTODESK.
Certified User

Adobe
CERTIFIED ASSOCIATE

EC-Council
Associate

Do it from anywhere - anytime
As per your convenience

+91 888 880 8544/ 855 096 6911

www.itolympiad.in

CITIS
knowledge integrated

CITIS Infotech : Knowledge Partner