ROLE OF JAVASCRIPT FUNCTION, METHOD, EVENT AND OBJECT IN WEBSITE DEVELOPMENT

Zobair Ullah
Assistant Professor, Department of CS&IT

DOI: http://dx.doi.org/10.24327/ijrsr.2020.1101.5060

ABSTRACT

This paper is intended to describe and discuss the role of JavaScript in website development. The article lucidly and precisely describes and explains some old but common uses of JavaScript as well as some modern uses of JavaScript, very much essential for web applications. This paper specifically discusses the use of JavaScript function, JavaScript method, JavaScript events and JavaScript object together with the help of suitable example.

INTRODUCTION

JavaScript is considered to be an indispensable part of websites because JavaScript is responsible for interactive and responsive web pages. The vast majority of websites use it and all major browsers (Netscape, Chrome, Firefox, Safari, Opera etc) have a dedicated JavaScript engine to execute it. Nowadays, JavaScript is supported by all modern web browsers and is used on almost every site on the web for more powerful and complex functionality. Alongside HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web (WWW). The usage of JavaScript has been included to a mainstream web development skill over the last few years. Right now, it is difficult to get a job as a web developer without JavaScript. So, this is an interesting time to discuss the role and importance of JavaScript in website development.

Brief history of Java Script

JavaScript was created by Brendan Eich in the year 1995 during his time at Netscape Communications Corporation. Live Script was the first name of JavaScript and made its first appearance in Netscape 2.0. Later on, Live Script was changed with JavaScript by Netscape Communications Corporation. In 1997, JavaScript 1.1 was submitted to the ECMA (European Computer Manufacturers Association). The ECMA – 262 specifications defined a standard version of the core JavaScript language.

Some major developers of JavaScript

1. Netscape Communications Corporation
2. Mozilla Foundation
3. ECMA International

Some important terminologies used in web development are

ECMA script ------ refers to a description, defining all the properties, methods and objects of a scripting language. The specification for JavaScript 2.0 is free on the ECMA script site.
Browser------ refers to a software/program used to navigate the internet.

OR

Web browser ------ (Commonly referred to as a browser) refers to a software application for accessing information on the world wide web (www).

- The first web browser was world wide web. It was invented/created by Tim Berners Lee in the year 1990.
- The purpose of a web browser is to fetch information resources and display them on a user’s device.
Some popular web browsers available for use are Google Chrome, Firefox, Internet Explorer, Safari and Edge.

Webpage

The webpage that appears on a browser is basically a combination of structure (HTML), style (CSS) and interactivity (JavaScript). These jobs are undertaken by three different technologies: HTML, CSS and JavaScript, which the browser knows how to interpret.

HTML, CSS and JavaScript work together in web pages but their code must be kept and dealt separately.

Website —— refers to a collection of publicly accessible, interlinked web pages that share a single domain name.

Client —— refers to a desktop computer or workstation that is capable of obtaining information and applications from a server in a computer network.

Server —— refers to a computer that provides data to other computers, LAN and WAN.

Interpreter —— refers to a program that can analyse and execute a program written in high level language line by line.

Compiler —— refers to a program that transforms a source code from a high level programming language to a lower level language or machine language.

JavaScript compilers actually perform Just-in-time compilation which occurs during run-time.

JavaScript Byte-code —— Spider Monkey bytecodes are the canonical form of code representation used in the Javascript engine.

Virtual machine —— refers to the software-driven emulation of a given computer system.

WWW (world wide web) —— refers to a set of protocols and formats that allow us to visit web pages in a browser.

JavaScript engine —— refers to a program or interpreter which executes JavaScript code.

OR

JavaScript engines —— refer to programs that convert JavaScript code into lower level or machine code.

- JavaScript engines also termed as JavaScript interpreter or JavaScript implementation or a type of virtual machine that basically read and execute JavaScript source code.
- JavaScript engines follow the ECMA Script standards.
- JavaScript engines are embedded in browsers and web servers, such as Node.js to allow run-time compilation and execution of JavaScript code.

Client side scripting (embedded scripts) —— refers to code that exists inside the client’s HTML page.

- This code will be processed on the client machine and the HTML page will not perform a Post Back to the web server.
- Client side scripting is used for page navigation, data validation and formatting.

Server side scripting —— refers to a technique used in web development which involves employing scripts on a web server which produce a response customized for each user’s (client’s) request to the website.

HTML Event —— refers to triggering/clicking a button or a link in a website. Some examples of event are closing a window, resizing a window, mouse hovering on small text.

- The “onclick” event type is most commonly used in JavaScript.
- HTML events are “things” that happen to HTML events.
- An HTML event can be something the browser does or something a user does.

Some common examples of HTML events are

- An HTML webpage has finished loading.
- An HTML input field was changed.
- An HTML button was clicked.

Definition

JavaScript is abbreviated as JS. It is a dynamic, high level, light weight, interpreted, general purpose, prototype-based (object oriented), multi-paradigm (event-driven, functional, imperative, object oriented), and open cross platform computer programming/scripting language.

The three main/core technologies used to create web pages are: HTML, CSS and JavaScript.

HTML —— stands for Hypertext Mark up Language.

- HTML is at the core of every web page regardless the complexity of a site or number of technologies involved.
- HTML is used to define the content and basic structure of a website. It defines and describes different structural types like paragraphs, blocks, lists, images, tables, forms, comments etc.
- HTML tells the browser what to display.

CSS —— stands for Cascade Style sheet.

- CSS helps to style the content of a website.
- CSS is used to control the presentation, formatting and specify the layout of web pages of a website.
- CSS tells the browser how to display the HTML content and how each type of element should be displayed, which may vary for different media (like screen, print or handheld device).
- HTML and CSS are on the front end of every webpage and application.

JavaScript —— is abbreviated as JS.

- JavaScript is used to program and provide most of the dynamic behaviour on a web page.
- JavaScript allows the webpage to perform some action.
- JavaScript is used to allow client-side scripting to interact with the user. For example, JavaScript is used to check whether the user has entered a valid e-mail address in a form field (generally referred as client side mechanism).
- JavaScript is a programming language and used to provide some interactive functionality to a website. It allows the user to interact with the browser.
- JavaScript is an interpreted language and embedded within some useful and most frequently used browsers.
like Internet Explorer, Netscape, Google chrome, Firefox and others.

- JavaScript tells the browser how to change the webpage in response to events that happen (like clicking on something or changing the value in a form input).
- JavaScript augments a browser’s default control and behaviour.
- JavaScript can help fix browser problems or patch holes in browser support.
- JavaScript can fix CSS layout issues in certain browser. Using JavaScript, you can find the position and area of any element on the page and the dimensions of the browser window. Using this information one can prevent overlapping of elements.
- Nowadays, JavaScript is supported by all modern web browsers and is used on almost every site on the web for more powerful and complex functionality.
- JavaScript is used to catch user initiated events such as link navigation, button and other actions.

**JavaScript implementation is made up of three distinct parts:** 1. The Core (ECMA script) 2. The document Object model (DOM) 3. The Browser Object Model (BOM).

**DOM** is an Application Programming Interface (API) for HTML as well as XML.

- The DOM maps out an entire page as a document composed of a hierarchy of nodes.
- **DOM level 1** became a W3C recommendation in October 1998. This level consisted of two modules: 1. The DOM core 2. The DOM HTML.
- **DOM level 2** introduced several new modules of the DOM to deal with new types of interfaces. 1. DOM views 2. DOM Events 3. DOM style 4. DOM Traversal and range.
- **DOM level 3** the DOM core is extended to support all of XML 1.0, including XML Infoset, XPath and XML Base.

**BOM** provides objects that interact with the browser window independent of the content.

- JavaScript is most frequently used as a part of web pages.
- To implement JavaScript code in HTML, JavaScript statements must be placed/inserted within the `<script>` tag.
- JavaScript is a case sensitive language.
- JavaScript can be used to animate elements on a page. For example, to show and hide information or to highlight specific sections of a page.

**Some major JavaScript text editors/editing tools are**

- Notepad
- Microsoft Frontpage
- Macromedia
- Dreamweaver Mx
- Macromedia Homesite5

**JavaScript functions**

- **JavaScript functions** refer to small packages that can be called anywhere in your program.
- **JavaScript function** refers to a block of JavaScript code that can be executed when “called” for. A function can be called when an event occurs. An event occurs when the user clicks a button.
- **JavaScript functions** refer to a block of code designed to perform a particular task.
  - JavaScript functions enable a programmer to partition a big program into a number of small and manageable functions.
  - JavaScript function is used to define code once, and use it many times.
  - One can use the same code many times with different arguments, to produce different results.

**JavaScript “function” keyword**

JavaScript functions are defined with the “function” keyword.

**Syntax to declare a JavaScript function**

```javascript
<script>
function function-name(parameters)
{
Local variables declaration and initialization;
Statement(s);
}
document.getElementById("demo").innerHTML = function-name(values);
</script>
```

**Function invocation** the code inside the function will execute when “something” invokes (calls) the function. Some cases are:

- When an event occurs (when a user clicks a button).
- When it is invoked (called) from JavaScript code.
- Automatically (self invoked).

**Local variables** refer to the variables declared within a JavaScript function.

- Local variables can only be accessed from within the function.
- Local variables are created when a function starts and deleted when the function is completed.
- Local variables with the same name can be used in different functions.
- JavaScript variables are containers for storing data values.

**JavaScript “return” keyword**

- A return statement determines the value the function returns.
- When a return statement is used in a function body, the execution of the function is stopped. If specified, a given value is returned to the function caller.
- The return statement ends function execution and specifies a value to be returned to the function caller.
JavaScript "var" keyword is used to declare a variable, optionally initializing it to a value.

**Syntax to declare a JavaScript variable**

```javascript
var variable-name;
```

or

```javascript
var variable-name = value;
```

**JavaScript "this" keyword** refers to the object it belongs to.

- In a method, this refers to the owner object.
- Methods like call() and apply() can refer this to any object.
- In a function, this refers to the global object.
- In a function, in strict mode, this is undefined.
- In an event, this refers to the element that received the event.

**The get Element ById() method** returns the element that has the ID attribute with the specified value.

- This method is one of the most common methods in the HTMDOM.
- This method is used almost every time you want to manipulate or get information from, an element on your document.
- This method returns null if no elements with the specified ID exists.
- The get Element ById() method use id="demo" to find the element.

**Inner HTML** is a property and it is the easiest way to get and replace the content of HTML elements including `<html>` and `<body>`.

**A property** is a value that you can get or set. For instance, changing the content of an HTML element.

**JavaScript programs based on JavaScript function.**

**JavaScript program to display a simple string using JavaScript function and return keyword.**

```html
<html>
<body>
<p id="demo"></p>
<script>
function f1( ) //function declaration statement
{
var a="JavaScript"; //local variable declaration/initialization
document.getElementById("demo").innerHTML = a ;
}
f1( ); //calling function or function invocation
</script>
</body>
</html>
```

**Output**

JavaScript program to display a simple string using JavaScript function, local variable and return keyword.

```html
<html>
<body>
<p id="demo"/>
<script>
function f1( ) //function declaration statement
{
var a="Introduction to JavaScript";
return a ;
}
var r=f1( ); // function is called, return value will end up in var r
document.getElementById("demo").innerHTML=r;
</script>
</body>
</html>
```

**Output**

Introduction to JavaScript

**JavaScript program to display a simple string using JavaScript function, local variable and onclick event.**

```html
<html>
<body>
<p id="demo"></p>
<script>
function f1() //function declaration statement
{
var a1; // Local variables declaration statement
a1="apple";
//variable initialization statement
document.getElementById("demo").innerHTML="My fruit="+ a1;
}
</script>
<button onclick="document.getElementById('demo').innerHTML=f1()"><b>Result</b></button> //calling function using onclick event
</body>
</html>
```

**Output**

Click on result button to get
My fruit=apple
Java Script program to display sum of any two integers using Java Script function, local variable and return keyword.

```html
<html>
<body>
<p id="demo"></p>
<script>
function f1() // function declaration statement
{
var a1,a2; //local variable declaration statement
a1=5; // variable initialization statement
a2=10;
var sum= a1+a2;
return sum;
}
document.getElementById("demo").innerHTML = "Sum=\n"+f1();
</script>
</body>
</html>
```

Output

Sum=15

Java Script program to display product of any two integers using Java Script parametric function and return keyword.

```html
<html>
<body>
<p id="demo"></p>
<script>
function f1(a1,a2) // parametric function declaration statement
{
return a1*a2; // function returns the product of a & b
}
document.getElementById("demo").innerHTML = f1(4, 3);
</script>
</body>
</html>
```

Output

12

Java Script program to display sum of any two integers using Java Script parametric function, local variable and return keyword.

```html
<html>
<body>
<p id="demo"></p>
<script>
function f1(a1,a2) // parametric function declaration statement
{
var sum= a1+a2; //assignment statement
return sum; //Function returns the value stored in local variable sum
}
document.getElementById("demo").innerHTML = "Sum=\n"+f1(10, 30);
</script>
</body>
</html>
```

Output

30

Java Script Objects

JavaScript objects ------- refer to an unordered collection of related data, of primitive or reference types, in the form of “key: value” pairs.

These keys can be variables or functions and are called properties and methods.

Java Script methods ------- refer to actions that can be performed on objects. For instance, adding or deleting an HTML element.

- A JavaScript method is a property containing a function definition.
A JavaScript methods are functions stored as object properties. For example, property value

<table>
<thead>
<tr>
<th>property</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FirstName</td>
<td>Nick</td>
</tr>
<tr>
<td>Id</td>
<td>678</td>
</tr>
</tbody>
</table>

Basic syntax to access object method

Object.method-name();

JavaScript has three kind of pop up boxes 1. Alert box 2. Confirm box and 3. Prompt box. The paper discusses only below two boxes.

The alert() method displays an alert box with a specified message.

- The alert box takes the focus away from the current window, and forces the browser to read the message.
- The basic syntax is alert("message or text");

The prompt() method displays a dialog box that prompts the visitor for input.

- The basic syntax is prompt(text, default text);
- A prompt box is used if you want the user to input a value before entering a page.
- When a prompt box pops up, the user will have to click either “OK” or “cancel” to proceed after entering an input value. If the user clicks “OK” the box returns the input value. If the user clicks “Cancel” the box returns null.

The sort() method sorts an array alphabetically.
The reverse() method reverses the elements in an array.
The pop() method removes the last element from an array.

The pop() method returns the value that was “popped out”.

The push() method adds a new element to an array (at the end).
The push() method returns the new array length.

The shift() method returns the string that was “shifted out”.

toUpperCase() method of the string object, to convert a text to uppercase.

toLowerCase() method of the string object, to convert a text to lowercase letters.

The JavaScript parseInt() function parses a string and returns an integer.

Java Script program based on JavaScript Objects/methods.

Java Script program to display a simple string using Java Script function and alert() method.

```html
<html>
<body>
<p id="demo1"></p>
<script>
function f1() // function declaration statement
{
alert("Be happy!"); //alert() method declaration
}
document.getElementById('demo1').innerHTML=f1();
</script>
</body>
</html>

Output
Be happy!
```

Java Script program to display a simple string using Java Script function, alert() method and onclick event.

```html
<html>
<body>
<p id="demo"></p>
<script>
function f1() // function declaration statement
{
Alert("server error"); //alert() method declaration
}
</script>
<button onclick="document.getElementById('demo').innerHTML=f1()"><b>Result</b></button>
</body>
</html>

Output
Click on result button to get Server error alert box
```

Java Script program to display a simple string using Java Script function, prompt() method, return keyword and onclick event.

```html
<html>
<body>
<p id="demo"></p>
<script>
function f1() // function declaration statement
{
var a=prompt("Enter your name:"); //prompt() method declared
return a;
}
</script>
<button onclick="document.getElementById('demo').innerHTML=f1()">click here</button>
</body>
</html>

Output
Enter your name: John
John
```

Java Script program to sort strings, reverse strings, remove the last element, add a new element and to show shifted out elements using JavaScript function, JavaScript methods and onclick event.

```html
<html>
<body>
<p id="demo1"></p>
<p id="demo2"></p>
<p id="demo3"></p>
<p id="demo4"></p>
<p id="demo5"></p>
</body>
</html>
```

Output

JavaScript program to add a new element in a list and display the list or array using JavaScript function and JavaScript method.

```html
<html>
<body>
<p id="demo1"></p>
<p id="demo2"></p>
<script>
var fruits=["banana","orange","apple","guava"]  //array elements
document.getElementById('demo1').innerHTML="Fruits:" + fruits+"<br>";
function f1( )     // function declaration statement
    return(fruits.sort());  //sorts the elements of fruits
function f2( )     // function declaration statement
    return(fruits);
function f3( )     // function declaration statement
    var r1=fruits.reverse();   //reverses the elements of fruits
    return r1;
function f4( )     // function declaration statement
    var r2=fruits.pop();     //removes the last element
    return r2;
function f5( )     // function declaration statement
    return fruits;
function f6( )     // function declaration statement
    return fruits.shift();  //returns shifted out string
</script>
<button onclick="document.getElementById('demo1').innerHTML= f1()"><b>click to sort</b></button>
<button onclick="document.getElementById('demo2').innerHTML= f2()"><b>click to see new list1</b></button>
<button onclick="document.getElementById('demo3').innerHTML= f3()"><b>click to pop or remove</b></button>
<button onclick="document.getElementById('demo4').innerHTML= f4()"><b>click to reverse</b></button>
<button onclick="document.getElementById('demo5').innerHTML= f5()"><b>click to see new list2</b></button>
<button onclick="document.getElementById('demo6').innerHTML= f6()"><b>click to shift</b></button>
</body>
</html>

Output
banana, orange, apple, guava

JavaScript program to convert/change a string into lowercase and uppercase using JavaScript function, JavaScript method and onclick event.

```html
<html>
<body>
<p id="demo1"></p>
<p id="demo2"></p>
<script>
var a="welcome";                  //string declaration statement
var b="JAVASCRIPT";
var c=prompt ("Enter a string in uppercase:");
function f1( ) // function declaration statement
    return (a.toUpper Case());     //changes string to uppercase
function f2 ( ) // function declaration statement
</script>
</body>
</html>

Output
banana, orange, apple, guava

JavaScript program to convert/change a string into lowercase and uppercase using JavaScript function, JavaScript method and onclick event.
```javascript

{ Return (b.toLowerCase()); // changes string to lowercase
}

function f3 () // function declaration statement
{
var r=c.toLowerCase(); // changes string to lowercase
return r;
}
</script>

<button onclick="document.getElementById('demo1').innerHTML=f1()"><b>click for uppercase</b></button>
<button onclick="document.getElementById('demo2').innerHTML=f2()"><b>click for lowercase</b></button>
<button onclick="document.getElementById('demo3').innerHTML=f3()"><b>click for lowercase</b></button>
</body>
</html>

Output
WELCOME
javascript
click for uppercase
click for lowercase
click for lowercase
javascritp program to input two numbers globally and make a simple calculator using JavaScript function, JavaScript method and onclick event.

<html>
<body>
<p id="demo1"></p>
<p id="demo2"></p>
<p id="demo3"></p>
<p id="demo4"></p>
<script>
var a=parseInt(prompt("Enter first number:"));
var b=parseInt(prompt("Enter second number:"));

function f1 () // function declaration statement
{
  r1=a+b;
  return r1;
}

function f2 () // function declaration statement
{
  r2=a-b;
  return r2;
}

function f3 () // function declaration statement
{
  r3=a*b;
  return r3;
}

function f4 () // function declaration statement
{
  r4=a/b;
  return r4;
}
</script>
<button onclick="document.getElementById('demo1').innerHTML=f1()"><b>click to add</b></button>
<button onclick="document.getElementById('demo2').innerHTML=f2()"><b>click to subtract</b></button>
<button onclick="document.getElementById('demo3').innerHTML=f3()"><b>click to multiply</b></button>
<button onclick="document.getElementById('demo4').innerHTML=f4()"><b>click to divide</b></button>
</body>
</html>

Output
15
5
50
2

click to add click to subtract click to multiply click to divide

JavaScript program to display a simple string using JavaScript object.

<html>
<body>
<p id="demo1"></p>
<script>
var p={ // object declaration statement
  name: "Nick",
};
document.getElementById('demo1').innerHTML="Name:"+p.name;
</script>
</body>
</html>

Output
Name:Nick

JavaScript program to display a string using JavaScript object, JavaScript function and this keyword.

<html>
<body>
<p id="demo1"></p>
<script>
var p={ // object declaration statement
  name: "apple",
  id:100,
  abc:function()
  {
    return this.name=""+"id:"+this.id;
  }
};
document.getElementById('demo1').innerHTML="Name:"+p.abc();
</script>
</body>
</html>

Output
Name:apple id:100

JavaScript program to create JavaScript objects and display them using this keyword.

<html>
<body>
<p id="demo1"></p>
<script>
</script>
</body>
</html>

Output
Name:apple id:100

JavaScript program to create JavaScript objects and display them using this keyword.

<html>
<body>
<p id="demo1"></p>
<script>
</script>
</body>
</html>

Output
Name:apple id:100

JavaScript program to create JavaScript objects and display them using this keyword.

<html>
<body>
<p id="demo1"></p>
<script>
</script>
</body>
</html>

Output
Name:apple id:100

JavaScript program to create JavaScript objects and display them using this keyword.
var student = { //object declaration statement
fname: "Anu",
age: 10,
id: 24,
abc: function () { return this.fname + " " + this.age + " " + this.id; }
};
document.getElementById('demo1').innerHTML = student.abc();
</script>
</body>
</html>

Output
Anu 10  24

JavaScript program to display a string using JavaScript function and JavaScript click event.

<html>
<body>
string:<input type="text" id="demo" value=" ">
<p id="demo"></p>
<script>
function f1() //function declaration statement
{
document.getElementById("demo").value="apple";
}
</script>
<button onclick="f1()">click here</button>
</body>
</html>

Output
string: apple
click here

JavaScript program to input any string and display it using JavaScript function and JavaScript onclick event.

<html>
<body>
string:<input type="text" id="demo" value=" ">
<p id="demo"></p>
<script>
function f1() //function declaration statement
{
var str = prompt("Enter a string:");
document.getElementById("demo").value = str;
}
</script>
<button onclick="f1()">click here</button>
</body>
</html>

Output
string: click here

JavaScript program to display a number using JavaScript object.

<html>
<body>
<p id="demo"></p>
<script>
var a = new Object(); //object declaration statement
a.n = 10;
document.getElementById("demo").innerHTML = a.n;
</script>
</body>
</html>

Output
10

JavaScript program to find sum of two numbers using JavaScript object.

<html>
<body>
<p id="demo"></p>
<script>
var a = new Object(); //object declaration statement
a.n1 = 10;
a.n2 = 20;
a.s = a.n1 + a.n2
document.getElementById("demo").innerHTML = "sum=\n\" + a.s;
</script>
</body>
</html>

Output
Sum=30

JavaScript program to find sum of two numbers using JavaScript function JavaScript click event and JavaScript object.

<html>
<body>
<p id="demo"></p>
<script>
function f1() //function declaration statement
{
var a = new Object(); //object declaration statement
a.n1 = 10;
a.n2 = 20;
a.s = a.n1 + a.n2
document.getElementById("demo").innerHTML = "Sum=\n\" + a.s;
}
</script>
<button onclick="f1()">click</button>
</body>
</html>

Output
Sum=30
Click

JavaScript program to display a simple string and a number using JavaScript object.

<html>
<body>
<p id="demo"></p>
<script>
var aa = { //object declaration statement
fn: "John",
</script>
</body>
</html>
age:10,
});
document.getElementById("demo").innerHTML="aa.fn+"+aa.age;
</script>
</body>
</html>

Output

John 10

Advantages of JavaScript

- Less server interaction.
- Immediate feedback to the visitors.
- Increased interactivity.
- Richer interfaces (drag and drop).

JavaScript limitations

- JavaScript cannot be considered as a full fledged programming language.
- Reading or writing of files is not allowed by client side JavaScript.
- JavaScript support is not available for networking applications.

CONCLUSION

- JavaScript is a wonderful technology to use on the web. It plays nicely with other web technologies such as HTML and CSS and can even interact with plugins such as Flash.
- JavaScript allows us to built highly responsive user interfaces, prevent frustrating page reloads, and even fix support issues for CSS.
- JavaScript can be used to make online systems available offline and sync automatically once the computer goes online by using the right browser addons (such as Google Gears or Yahoo browser plus).
- JavaScript is also not restricted to browsers. The speed and small memory footprint of JavaScript in comparison to other languages brings up more and more uses for it -- from automating repetitive tasks in programs like illustrator, up to using it as a server-side language with a standalone parser.
- On the basis of above discussion, one can easily say the future of JavaScript is wide open.

Acknowledgment

I would like to express my deep gratitude to the vast intellectual community of the world who convey their modern and scientific ideas, new theory and emerging trends and description of new inventions and discoveries in various disciplines of study through internet. These are rich source of information and greatly support research activities. Classroom teaching, student’s problem and situation based learning also help and motivate for exploring, searching and finding new ideas and make things simple and improve existing technology to a larger extent. At last, I would like to thank the great almighty who has given wisdom, strength, knowledge and such a wonderful mind to visualise, analyse and explore things from grass root level and put on papers for the benefit of mankind and to promote safe and green technology inventions and new discoveries.

References

2. JavaScript Tutorial - W3Schools
4. Katz, Yehuda. "Understanding "Prototype"
7. "Chapter 4. How JavaScript Was Created".
9. The Modern JavaScript Tutorial
10. www.javascript.com

How to cite this article:

*******