**Module 6 :- JavaScript Basic & DOM**

**1). What is JavaScript?**

JavaScript is a Dynamic Programing Language that’s used for web development, in web applications, for game development, and lots more. It allows you to implement dynamic features on web pages that cannot be done with only HTML and CSS.

Many browsers use JavaScript as a scripting language for doing dynamic things on the web. Any time you see a click-to-show dropdown menu, extra content added to a page, and dynamically changing element colors on a page, to name a few features, you’re seeing the effects of JavaScript.

**2). What is the use of isNaN function?**

NaN values are generated when arithmetic operations result is undefined or unrepresented values. Such value do not necessarily represent overflow conditions.

For example, dividing zero by zero results in a NaN — but dividing other numbers by zero does not.

**Examples :-**

**Using NaN() :-**

Number.isNaN(NaN); // true

Number.isNaN(Number.NaN); // true

Number.isNaN(0 / 0); // true

Number.isNaN(37); // false

**Difference between Number.isNaN () and global isNaN () :-**

Number.isNaN() doesn’t attempt to parameter to a number, So non-numbers always return false. The following are all false :-

Number.isNaN ("NaN"); Number.isNaN (undefined);

Number.isNaN({}); Number.isNaN ("blabla");

Number.isNaN (true); Number.isNaN (null);

Number.isNaN ("37"); Number.isNaN ("37.37");

Number.isNaN (""); Number.isNaN (" ");

**The global isNaN () coerces its parameter to a number :-**

isNaN ("NaN"); // true

isNaN (undefined); // true

isNaN({}); // true

isNaN("blabla"); // true

isNaN(true); // false, this is coerced to 1

isNaN(null); // false, this is coerced to 0

isNaN("37"); // false, this is coerced to 37

isNaN("37.37"); // false, this is coerced to 37.37

isNaN(""); // false, this is coerced to 0

isNaN(" "); // false, this is coerced to 0

**3). What is negative infinity?**

No other number is lesser than this value. It can be generated using a self-made function or by arithmetic operation.

**Note :- JavaScript shows the NEGATIVE\_ INFINITY value as – Infinity.**

**Negative Infinity** is different from mathematical infinity the following ways:

1. Negative infinity results in 10 when divided by any other number.
2. When divided by itself or positive infinity, negative return NaN
3. Negative infinity when divided by any positive number (apart from positive infinity ) is negative infinity.
4. Negative infinity divided by any negative number ( apart form negative infinity ) is positive infinity.
5. If we multiply negative infinity with NaN, we will get NaN as a result.
6. The product of NaN and negative infinity is 0.
7. The product of two negative infinities is always a positive infinity.
8. The product of both positive and negative infinity is always negative infinity.

**4). Which company developed JavaScript?**

JavaScript was invented by Brendan Eich in 1995. It was developed for Netscape 2, and became the ECMA-262 standard in 1997. After Netscape handed JavaScript over to ECMA, the Mozilla foundation continued to develop JavaScript for the Firefox browser.

**5). What are undeclared and undefined variables?**

**Undefined** variables means a variable has been declared but it does not have a value.

Example :-

var car;

console.log(car);

**Undeclared** variable means that the variable does not exist in the program at all.

Example :-

console.log(bike);

**6). Write the code for adding new elements dynamically?**

HTML defines the structure of your web document and the content therein. CSS declares various styles for the contents provided on the web document.

HTML and CSS are often called markup languages rather than programming languages, because they, at their core, provide markups for documents with very little dynamism.

JavaScript, on the other hand, is dynamic programming language that supports Math calculations, allows you to dynamically add HTML contents to the DOM, creates dynamic style declarations, fetches contents from another website, and lots more

Example :-

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Nested Loop</title>

</head>

<body>

<h2>Nested Loop</h2>

<p>First Work Outer Loop Then Work Inner Loop</p>

<script>

// 1 2 3 4 5 6 7 8 9 10

// 11 12 13 14 15 16 17 18 19 20

// 21 22 23 24 25 26 27 28 29 30

// 31 32 33 34 35 36 37 38 39 40

// 41 42 43 44 45 46 47 48 49 50

// 51 52 53 54 55 56 57 58 59 60

// 61 62 63 64 65 66 67 68 69 70

// 71 72 73 74 75 76 77 78 79 80

// 81 82 83 84 85 86 87 88 89 90

// 91 92 93 94 95 96 97 98 99 100

for (var a=1;a<=100;a+=10)

{

for (var b=a;b<a+10;b++)

{

document.write(b+" ");

}

document.write("<br>");

}

document.write("<br>");

// 1

// 1 2

// 1 2 3

// 1 2 3 4

// 1 2 3 4 5

for (var a=1;a<=5;a++)

{

for (b=1;b<=a;b++)

{

document.write(b+" ");

}

document.write("<br>");

}

document.write("<br>");

// 1 2 3 4 5

// 1 2 3 4

// 1 2 3

// 1 2

// 1

for (var a=5;a>=1;a--)

{

for (b=1;b<=a;b++)

{

document.write(b+" ");

}

document.write("<br>");

}

document.write("<br>");

// 1

// 2 2

// 3 3 3

// 4 4 4 4

// 5 5 5 5 5

for (var x = 1; x <= 5; x++) {

for (y = 1; y <= x; y++) {

document.write(x + " ");

}

document.write("<br>");

}

document.write("<br>");

// 5 5 5 5 5

// 4 4 4 4

// 3 3 3

// 2 2

// 1

for (var x = 5; x >= 1; x--) {

for (y = 1; y <= x; y++) {

document.write(x + " ");

}

document.write("<br>");

}

document.write("<br>");

</script>

</body>

</html>

**7). What is the difference between ViewState and SessionState?**

The ViewState is to manage state at the client’s end, making state management easy for end-user while SessionState manages state at the server’s end, making it easy to manage content from this end too.

Usage

**SessionState :-** It can be used to store information that you wish to access on different web pages.

**ViewState :-** It can be used to store information that you wish to access from same web pages.

**8). What is === operator?**

When we compare two variables of different type e.g. a Boolean with a string or a number with string using === operator, it automatically converts one type into another and return value based upon content equality.

While === operator is strict equality and only return true if both variable of same type and also contains same value.

Example :-

**0**==**false** // true, because false is equivalent of 0

**0**===**false** // false, because both operands are of different type

**2**=="2" // true, auto type coercion, string converted into number

**2**==="2" // false, since both operands are not of same type

**9). How can the style / class of an element be changed?**

**10). How to read and write a file using JavaScript?**

**fs.readfile ()** and **rs.readfile ()** methods are used to read and write of a file using JavaScript. The file is read using the fs.readFile() function, which is an inbuilt method.

**Syntax :-**

fs.readFile ( file\_name, encoding, callback\_function )

**Parameters :-**

Filename :- It contains the filename to be read, or the whole path if the is saved elsewhere.

Encoding :- It stores the file’s encoding. ‘utf8’ is the default setting.

Callback Function :- This is a function that is invoked after the file has been read. It requires two inputs.

Err :- If there was an error.

Data :- The file’s content.

Return Value :- It returns the contents contained in the file, as well as any errors that may have occurred.

**fs.writeFile() function is used to write data to a file in an asynchronous manner. If the file already exists, it will be replaced.**

**Syntax :-**

fs.writeFile ( file\_name, data, options, callback)

**Parameters :-**

file\_name :- It’s a string, a buffer, a URL, or a file description integer that specifies the location of the file to be written. When you use a file descriptor, it will function similarly to the fs.writeFile() method.

Data :- The data that will be sent to the file is a string, Buffer, Typed Array, or DataView.

Options :- It’s a string or object that may be used to indicate optional output options. It include three more parameters that may be selected.

Encoding :- It’s a string value that indicates the file’s encoding. ‘utf8’ is the default setting.

Mode :- The file mode is specified by an integer number called mode. 0o666 is the default value.

Flag :- This is a string that indicates the file-writing flag. ‘w’ is the default value.

Callback :- This function gets invoked when the method is run.

Err :- If the process fails, this is the error that will be thrown.

**11). What are all the looping structures in JavaScript?**

If you want to run the same code over and over again, each time with a different value.

**Different Types of loops :-**

**For :-** loops through a block of code a number of times

**For/In :-** loops through the properties of an object

**For/of :-** loops through the values of an iterable object

**While :-** loops through a block of code while a specified condition is true

**Do/While :-** Also loops through a block of code while a specified condition is true

**Examples :-**

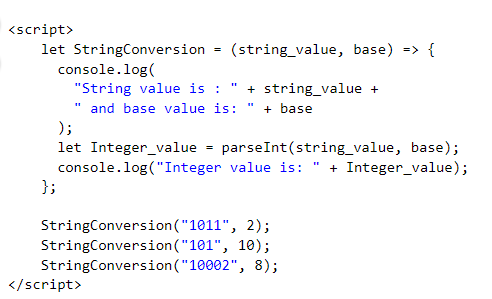
<https://github.com/jayminnai>

**12). How can you convert the string of any base to an integer in JavaScript?**

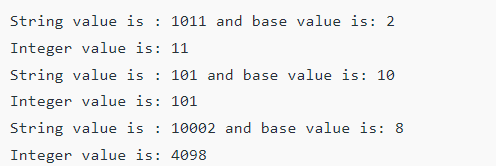
**parseInt ()** function (or a method) is used to convert the passed in string parameter or value to an integer value itself. This function returns an **integer** of base which is specified in second argument of **parseInt () function.**

parseInt () function returns NaN(not a number) when the string doesn’t contain number.

**Example :-**



**Output :-**



**13). What are all the types of Popup boxes available in JavaScript?**

**Alert Box :-**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Alert Box</title>

</head>

<body>

<h2>Alert Box</h2>

<script>

var a=30;

var b=20;

if (a>b)

{

alert("A is greater than B");

}

else

{

alert("B is greater than A");

}

</script>

</body>

</html>

**Confirm Box :-**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Confirm Box</title>

</head>

<body>

<h2>Confirm Box</h2>

<script>

var a= confirm("can we proceed?")

//alert(a);

if (a)

{

// if you click on ok button(true)

alert("Success");

}

else

{

// if you click on cancel button(false)

alert("fail");

}

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

**Prompt Box :-**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Prompt Box</title>

</head>

<body>

<h2>Prompt Box</h2>

<!-- <script>

var a= prompt("enter your name");

document.write("Welcome" + a);

</script> -->

<script>

var a=prompt("enter number of month");

switch(true)

{

case(a==1): document.write("january");

break;

case(a==2): document.write("february");

break;

case(a==3): document.write("march");

break;

case(a==4): document.write("april");

break;

case(a==5): document.write("may");

break;

case(a==6): document.write("june");

break;

case(a==7): document.write("july");

break;

case(a==8): document.write("august");

break;

case(a==9): document.write("september");

break;

case(a==10): document.write("october");

break;

case(a==11): document.write("november");

break;

case(a==12): document.write("december");

break;

default: document.write("please enter valid number 1 to 12");

}

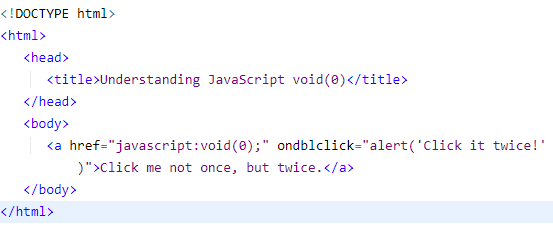
</script>

</body>

</html>

**14). What is the use of Void (0)?**

If inserting an expression into a web page results in an unwanted effects, then use JavaScript void to remove it. Adding “JavaScript: void(0)”, returns the undefined primitive value.



**15). How can a page be forced to load another page in JavaScript?**

We can use **window.location** property inside the script tag to forcefully load another page in JavaScript. It is a reference to a location object that is it represents the current location of the document. We can change the URL of a window by accessing it.

**Syntax :-**

<script>

window.location = <Path / URL>

</script>

****

**New Page :-**

****

**16). What are the disadvantages of using innerHTML in JavaScript?**

**The use of innerHTML very slow :-** The process of using innerHTML is much slower as it’s contents as slowly built, also already parsed contents and elelments are also re-parsed which takes time.

**Content is replaced everywhere :-** Either you add, append, delete or modify contents on a webpage using innerHTML, all contents is replaced, also all the DOM nodes inside that elements are re-parsed and recreated.

**Appending to innerHTML is not supported :-** Usually, += is used for appending in JavaScript. But on appending to an HTML tag using innerHTML, the whole tag is re-parsed.

**Can break the document :-** There is no proper validation provided by innerHTML, so any valid HTML code can be used. This may break the document of JavaScript. Even broken HTML can be used, which may lead to unexpected problems.