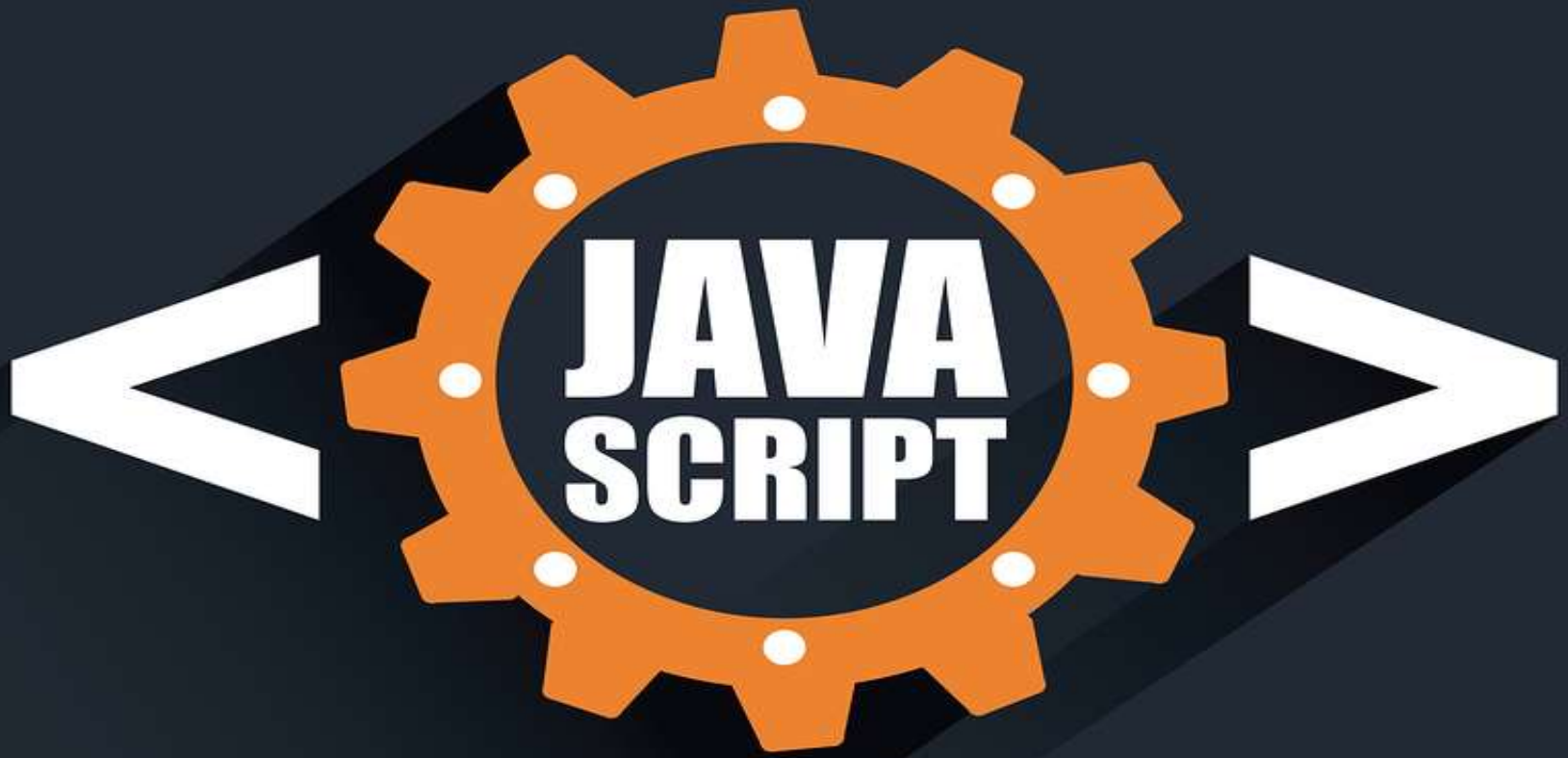


Learn JavaScript From Scratch

...

Mohd Manzoor Ahmed

(Microsoft Certified Trainer | 15+yrs Exp)

What Are We Going To Learn?

- Introduction To JavaScript
- Introduction To Visual Studio
- Modes Of Execution
- Data Types In JavaScript
- Conditional Statements
- UnConditional Statements
- Iterative or Loop
- Types Of Variables
- Events
- OOPs Through JavaScript
- Object
- Class
- Properties
- Methods

Enroll Here

At 50% OFF - Use Coupon Code as ‘HALF-OFF’

Why JavaScript?

- We need responsive HTML pages.
  - React to events.
  - Read and write HTML elements.
  - Validate data.
  - Detect the visitor's browser.
  - Lot more these days....

What is JavaScript?

- JavaScript is a scripting and lightweight programming language.
- It is an interpreted language and requires a browser to run it.
- Everyone can use JavaScript without purchasing any license.
- It is usually embedded directly into HTML pages in script tag inside head tag.
- You can use any text editor to write javascript including HTML
    - <head>
    - <script type=”text/javascript”>

Display Hello World

- Mostly used methods to display any content at runtime on web page are
  - document.write()
  - console.log()
  - window.alert()

Demo

For “Hello World”

What is JavaScript?

- But these days it is being written in separate file and it's extension is **.js** say **myScript.js**
- It is being referred in the web pages using **script** tag at the bottom of the page, i.e., after **body** tag.
    - <html>
    - <head>
    - </head>
    - <body>

Demo

For .js file

IDE - Introduction To Visual Studio

- https://www.visualstudio.com/en-us/products/visual-studio-community-vs.aspx

Any Version

Demo

Getting Friendly with Visual Studio 2015

Modes Of Execution

- **Immediate mode** of script execution is writing script inside body tag and gets executed as soon as page loads.
- **Deferred mode** of script execution is writing script in head tag and gets executed on any event.

Demo

For both immediate and deferred modes

Data Types In JavaScript

- numbers
- strings
- arrays
- objects
- array of objects

Enroll Here

At 50% OFF - Use Coupon Code as ‘HALF-OFF’

Single Variable Declaration

- var x;
- x=10;
- or
- var x=10;
- or
- x=10;

- var x=10;
- x=’Peter’;
- x=”John”;
- x=12.6;

Performing Operations - Require Operators

- **Arithmetic** : +,-,*,/,%
- **Assignment** : =, += (x+=y => x=x+y),-=,*=,%=
- **Comparison** : ==,===,!==,!===,<,<=,>,>=,

Demo

For All Operations

Control Structure

- Controlling the flow of our structure is achieved through the following statements
- Conditional
    - if, if else, else..if ladder and switch..case
- UnConditional
    - break, continue, goto and return. (Covered in functions)
- Iterative or Loop
    - for, while, do while, for/in

Conditional Statements - if else

```
<script type="text/javascript">
  var t=6;
      if (t<10) {
            document.write("<b>Good morning</b>");
            }
      else {
            document.write("Good day!");
            }
 </script >
```

Conditional Statements - else if Ladder

```
<script type="text/javascript">
  var t=16;
            if (t<10)  {
            document.write("<b>Good morning</b>");
            }
            else if (t>10 && t<16)  {
            document.write("<b>Good day</b>");
            }
            else {
            document.write("<b>Hello World!</b>");
            }
</script>
```

Demo

For if, if else and else if ladder

Conditional Statements - switch case default

```
<script type="text/javascript">
        theDay=5;
        switch (theDay){
        case 5:  document.write("Finally Friday");
        break;
        case 6:  document.write("Super Saturday");
        break;
        case 0:  document.write("Sleepy Sunday");
        break;
        default:  document.write("I'm looking forward to this weekend!");
        }
</script>
```

Enroll Here

At 50% OFF - Use Coupon Code as ‘HALF-OFF’

Demo

For switch case default

Iterative Statements - for

Syntax

for (statement 1; statement 2; statement 3) {

____

}

Ex:

```
for(i=0;i<10;i++){
document.write("MTT"+<br/>);
}
```

Demo

For for loop

Iterative Statements - while

Syntax:

while (condition) {

_____

}

Eg:

```
i=0;
while(i<10){
document.write("MTT"+<br/>);
i++;
}
```

Demo

For while loop

Iterative Statements - do while

Syntax:

do {

_____

}

while (condition);

Eg:

```
i=0;
do{
    document.write("MTT"+<br/>);
i++;}
    while(i<10);
```

Enroll Here

At 50% OFF - Use Coupon Code as ‘HALF-OFF’

Demo

For do while

Array Variable Declaration

- An Array is collection of similar type of elements.
- It can be defined in three ways
- **Literal Array: var** x=[‘Peter’,’John’,’Bob’]; or var x=[]; //best approach
- **Condensed Array: var** x=**new Array**(‘Peter’,’John’,’Bob’);
- **Regular Array: var** x=**new Array**();
    - x[0]=’Peter’;

Demo

For arrays

Iterative Statements - for/in

Syntax:

for(item in list)
{
____
}

Eg: var people=[‘Peter’,’John’,’Bob’];
for (i in people) {
document.write(people[i] +”
”);
}

Demo

For for/in

Functions

- It is a set of instructions to perform a specific task.
- It can be classified into four categories.
    - No Parameter - No Return Value
    - With Parameter - No Return Value
    - No Parameter - With Return Value
    - With Parameter - With Return Value
- Syntax: *function name(parameter1, parameter2, parameter3) {*

Demo

Implementing functions

Types Of Variables or Scope Of Variables

- Local : Declared inside the function.
- Global : Declared outside the function.

Enroll Here

At 50% OFF - Use Coupon Code as ‘HALF-OFF’

Demo

For Variables

Events

- Windows
- Form
- Keyboard
- Mouse

Windows Events

Major windows events

- onload

Eg:

<element onload="script">

Note:

- It is mostly used on <body> tag.
- Introduction to navigator object.

Demo

For Windows Events

Forms Events

Major events used with form elements

- onfocus : textbox
- onblur : textbox
- onchange : textbox and dropdown list
- onsubmit : form

Note:

- Introduction to getElementById object.
- Introduction to test() method of Regular Expression

Demo

For Form Events

Keyboard Events

Major Keyboard events are

- onkeydown //textbox
- onkeyup //textbox
- onkeypress //textbox

Note:

- Count number of characters in the textarea, like in twitter.

Enroll Here

At 50% OFF - Use Coupon Code as ‘HALF-OFF’

Demo

For Keyboard Events

Mouse Events

Major mouse events

- onclick : button
- onmouseover : link
- onmouseout : link

Note:

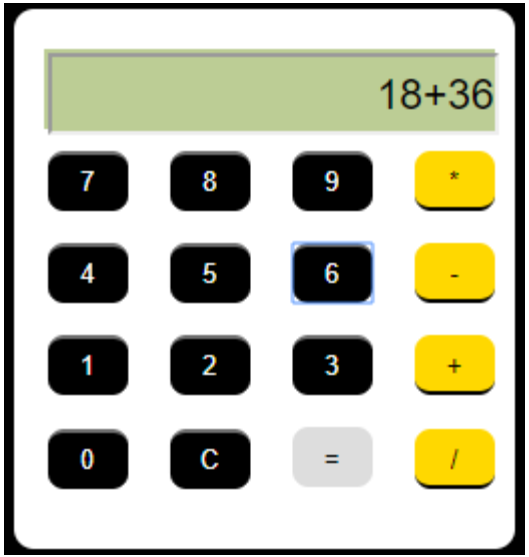
- Introducing <marquee> tag

Demo

For Mouse Events

Creating Calculator

- Basic Operations
- Introducing eval function



Demo

For Calculator

Stand Alone Object/s

- Single Object
  - var obj={“RollNo”:1, “Name”:"Peter", “Marks”:75};
  - obj.RollNo
- List Of Object
  - var lstObj=[{“RollNo”:1, “Name”:"Peter", “Marks”:75},
    {“RollNo”:2, “Name”:"John", “Marks”:25},
    {“RollNo”:3, “Name”:"Bob", “Marks”:35}]

Enroll Here

At 50% OFF - Use Coupon Code as ‘HALF-OFF’

Demo

For Stand Alone Object\s

Why OOPs?

- No strong binding between data and functions.

Demo

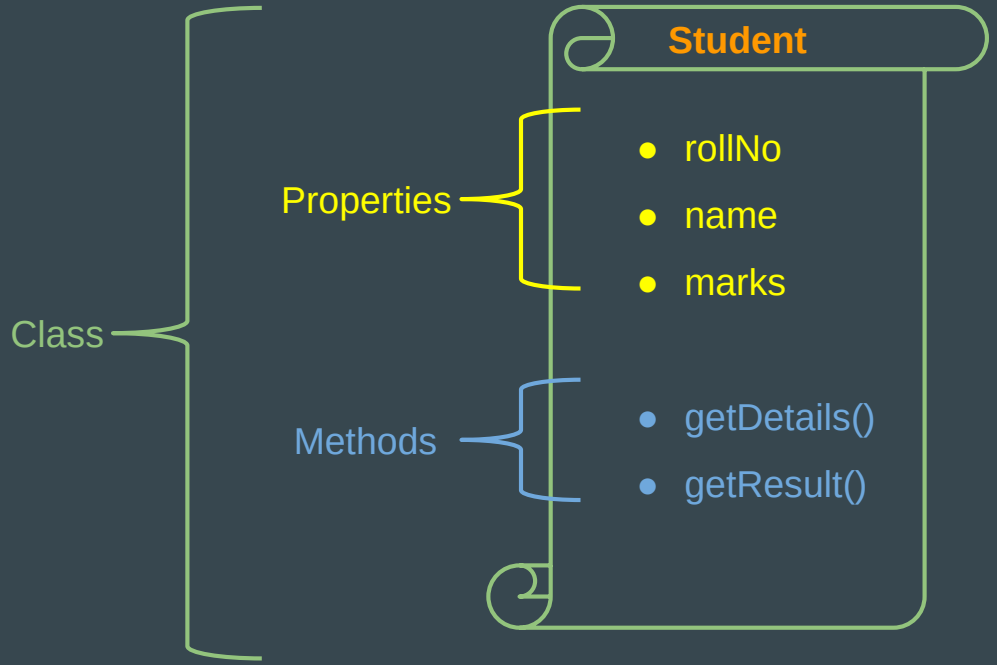
For Why OOPs

OOPs Through JavaScript

- Object
- Class
- Properties
- Methods

Analogy

- Objects
  - **Jack**
  - **Peter**
  - **John**



Object & Class

- **Object**
  - Any real time entity is called as an object.
  - Every object consist of state(look and feel) and behaviour(what is does).
  - States are called as **properties** and behaviors are called as **methods.**
- **Classes**
  - Class is a blueprint of an object.

Enroll Here

At 50% OFF - Use Coupon Code as ‘HALF-OFF’

Constructor

- Constructor is invoked automatically when we create an object.
- You cannot invoke constructor explicitly.
- It is used to initialize an object.
- Here in javascript, it is slightly different.

Defining A Class

- Javascript is prototype-oriented, classless, or instance-based programming.
- But we can simulate the class concept using JavaScript functions.

```
function Student(){
        this.rollNo=123;
        this.name="ManzoorThetrainer";
        this.marks=75;
        this.getResult=function(){
            if(this.marks>=35)
            return “Pass”;
            else
            return “fail”;
        }
}
```

Demo

For Defining A Class

Object Creation

- Creating Object
- Accessing Properties
- Accessing Method

```
var obj=new Student();
alert(obj.rollNo + “ ” obj.name);
alert(obj.getResult());

obj.rollNo=111;
obj.name=”Peter”;
obj.marks=23;
alert(obj.rollNo + “ ” obj.name);
alert(obj.getResult());
```

Demo

For Object Creation

Constructor

- Creating Class
- Creating Object
- Accessing Properties
- Accessing Method

```
function Student(r,n,m){
        this.rollNo=r;
        this.name=n;
        this.marks=m;
       ...
       ...
}
var obj=new Student(1,”Jack”,78);
alert(obj.rollNo + “ ” obj.name);
alert(obj.getResult());
```

Demo

For Constructor

Enroll Here

At 50% OFF - Use Coupon Code as ‘HALF-OFF’

List Of Objects

- Creating List Of Object
- Accessing Object's Properties
- Accessing Object's Method

```
function Student(r,n,m){
        this.rollNo=r;
        this.name=n;
        this.marks=m;
       ...
       ...
}
var lstObj=new Array(); / [];
lstObj.push(new Student(1,"Jack",78));
lstObj.push(new Student(2,"Peter",81));
```

Demo

For List Of Objects

Enroll Here

At 50% OFF - Use Coupon Code as ‘HALF-OFF’

Thanks