

JavaScript

Understanding the basics

Goals



- ✓ To understand JavaScript function
- ✓ Introduction to in-built JavaScript object and functions
- ✓ JS Callbacks and their use
- ✓ Asynchronous v/s Synchronous programming
- ✓ Timer functions in JS
- ✓ Practice all of the above in hands-on

Some differentiating factors of JavaScript are :

- JS is **weakly typed language**, no data type declaration i.e. All variables are declared as “var”.
- JS functions requires **no return type** to mentioned
- JS functions can also be defined as variables/objects
- Client Side JS is interpreted by all modern browsers

JS Objects



JS objects have two main components:

- **Properties** : State of object
- **Methods** : Behaviors of Object

- Properties are defined as variables (int, string, array, objects)
- Methods are defined as functions

JS objects are generally called as **JSON objects**.

JSON



JSON objects are collection of **key – value** pairs :

Key - name of Property or Method

Value – value of property or method

```
var Fan = {}    //empty object

var Car1 = { "speed" : 100 }

var Car2 =
{
"speed" : 90,
"accelerate" : accelerate()
}
```

javascript

JS Functions



There can be 3 methods of declaration

Normal declaration

Variable style declaration

Anonymous

```
function accelerate(){  
    console.log("speed up");  
}
```

```
var accelerate = function(){  
    console.log("speed up");  
}
```

```
function(){  
    console.log("speed up");  
}
```

javascript

JS Scope



scope is the set of variables, objects, and functions you have access to.

```
a = 20;  
function(){  
  var a = 11;  
}
```

javascript

Common Objects

Date object has to be initialized via constructor

```
var d = new Date();  
  
var d = new Date(milliseconds);  
  
var d = new  
Date(year, month, day, hr, min, sec, ms);
```

javascript

JS Array



JS Array are simple to declare

```
var fruits = ["apples", "bananas", "apricots"];
```

Array functions :

- push
- pop
- shift
- unshift
- join
- splice
- concat
- length

JS Strings



Most used String functions are :

- split
- indexOf
- charAt
- slice
- substr

Many Math functions are available :

- abs
- round
- floor
- ceil
- max
- min
- random

```
var random = Math.random();
```

Timed functions are used rarely, but can be useful sometime:

- setTimeout
- setInterval

```
var random = setTimeout(  
function(){console.log("hello")}  
,2000);
```

JS Events



There can be 3 types of events

Browser Events – onload ,onblur, onchange, onfocus

Keyboard Events – keyup, keydown

Mouse Events – mousedown, mouseup, mouseclick, mouseover, mouseout

```
<input onblur="checkPasswordStrength()">
```

```
<input onkeyup="callAds()">
```

```
<button onclick="popup()">
```

Browser Functionality



Main object of browser is called – window

- **window.location** : locational parameters
- **window.innerWidth** : viewport width
- **window.open()** - open a new window
- **window.close()** - close the current window
- **window.moveTo()** -move the current window
- **window.resizeTo()** -resize the current window

Asynchronous JS

JS Callbacks



In JS, functions are first-class objects

- functions are of the type Object
- they can be used as like any other objects
- we can pass them as arguments also

```
function getSpeed(){  
  
}  
  
function accelerate(getSpeed){  
  
}
```

Why callbacks are required

- To control the flow of execution
- To change execution flow according to previous outputs

```
function getSpeed(){
    return speed ;
}

function accelerate(getSpeed){
    if(getSpeed()>100){
        console.log("stop");
    }
}
```

JS Asynchronous calls



Asynchronous call/functions do not block the execution of subsequent call/functions

```
function postData(data){  
}  
  
function getData(data){  
}  
  
postData();  
  
getData();
```

JS Asynchronous calls



Callbacks are mostly used in

- Network Calls or Delayed calls (success or error callbacks)

```
function postData(data, success, error)
{

}
```

URL and Location

URL



Uniform Resource Locator follow a structure :

<http://www.google.com/mail/index.html?query=google>

Protocol + **Site Address** + **File location** + **Params**

JS location object can be used to get and set locational parameters (**window.location**)

Main properties and functions of Location object are:

- href
- pathname
- hostname
- protocol


```
console.log(window.location.href);
```

```
window.location.href = http://www.google.com
```

```
console.log(window.location.path)
```

HTML forms

Variable can be passed from one page to another using these methods.

- 1. Passing Through URL**
- 2. Passing through HTML Forms**

<form>

input elements



There can be multiple input elements in a form

</form>

HTML : Forms - Input Elements



- Textbox
 - for entering short text (Name)
- Password box (Asterisks)
 - for entering Password
- Radio buttons among various choices ()
 - for providing single option
- Check Box
 - for choosing a preference
- Text Area (Suggestions)
 - for entering detailed info
- Drop down boxes choices(Country)
 - for providing a list of
- Submit button
 - for submitting the form

HTML : Forms (Client Side)



name : it acts as **variable** ;

value : it is stored as **information** in variable (comes from user input)

```
<form method="post" action="submit.php">
```

Score:

```
<input type="text" size="10" maxlength="40" name="score">
```

```
<br/>
```

```
<input type="submit" value="Email Yourself">
```

```
</form>
```

score = user input

form.html

New types



color
date
datetime
datetime-local
email
month
number
range
search
tel
time
url
week

form.html

HTML : Forms



If you use GET as method it will be same as **passing through URL**

```
<form method="post" action=submit.php >  
</form>
```

Method = Method of posting form data (POST/GET)

Action = destination of form data (PHP file)

HTTP Request - Response

**Thank
You**