

Introduction to Programming and JavaScript

Learning Goals

- Introduction to Basic programming in general (applicable to JavaScript in particular)
- Applications of JavaScript
- JavaScript Events
- JS HTML DOM
- Idea for the project

Contents

- Examples of what JavaScript can do.
- What/Why JavaScript
- Introduction to Programming (data types, if else ,loop, functions)
- Buttons and forms in HTML.
- JavaScript events
- Introduction to DOM
- Implementing all the above concepts in JavaScript through a calculator program
- JavaScript Project

What/why JavaScript

JavaScript is now a de facto standard for adding interactivity to web sites:

- General purpose programming language
- Platform-independent
- Runs in every (major) browser
- Has built-in HTML DOM interface

It is also:

- “world’s most misunderstood programming language”
- source to confusion (**JavaScript, ECMAScript**)

Examples

- Slideshows in websites
- Animation Effects
- Form Validation
- Add interactivity

etc.

Interactive websites - Technologies

Client-side (require special browser plugins):

- Adobe Flash - support dropped in Google Chrome
- Java applets - security flaws, high resource demands
- Silverlight - deprecated

Server-side (require special sw on web server):

- ASP, JSP, PHP etc.

The trend is to move more and more processing in the client - web browser.

JS in this course

- Interactive website (project 2): we'll change behaviour of an existing website dynamically (on the go)
- Game in JS (project 3):
we'll create a *separate* application that runs in the browser
- Make web application (project 4):
dynamic website connected to a database, unique experience for every user

Programming in JS

We use JS for *web programming*.

But JS *programs* are just a bunch of text files.

Web browser *executes* JS programs - follows our instructions to change its behaviour.

Web programming - first of all, changing behaviour of the *client* (can be used to make changes in the server too).

Programming and programs

A *program* - set of instructions/steps to solve a problem. Also called *algorithm*.

Algorithms in real life:

- cooking recipe
- getting from home to work by train/bus/boat
- ...

Programming - creating instructions so that anyone can solve same problem.

Programming and programs (2)

Humans can often understand even unclear instructions.

Computers need precise instructions.

Computer programs are written in a *programming language*.

Every programming language has rules for writing programs - *syntax*.

- Basic Concepts that will be covered in this course are

- 1)Data Types
- 2)Conditional Statements
- 3)Loops
- 4)Functions

Data Types

- var is the only data type that can represent any type of data be it Integers, Real numbers or Strings or objects or booleans.
- var price1 = 5;
var price2 = 6;
var total = price1 + price2;
- var person = "John Doe", carName = "Volvo", price = 200;

- Note: The following slides have been made Referring from W3schools.com

Conditional Statements

- Used to execute code based on conditions
- Two types of conditional statements.

1) if else

2) switch case

Syntax

```
if(condition)
```

```
{
```

```
//code to run
```

```
}
```

```
Else{ //Code to run }
```

- Switch(variable)

```
{
```

```
case 0://code to run; break;
```

```
case 1://code to run; break;
```

```
etc.....
```

```
}
```

Loops

- Used for performing a repetitive tasks. There are 3 types of loops
 - 1)for loop
 - 2)while loop
 - 3)do-while loop

We will look only into ‘for loop’

- Syntax

```
for (statement 1; statement 2; statement 3) {  
    code block to be executed  
}
```

eg.

```
for (var i = 0; i < 5; i++)  
{  
    document.write(i+"<br>")  
}
```

Functions

- A piece of code that can be used repetitively.
- Syntax

```
function name(parameter1, parameter2, parameter3) {  
    //code to be executed  
}
```

Example

```
function myFunction(a, b)  
{  
    return a * b;           // Function returns the product of a and b  
}  
  
var x = myFunction(4, 3);
```

HTML Forms and Buttons

- HTML forms are used to collect user input.
- <form>
 - .
 - form elements*
 - .
 - </form>
- The <input> element is the most important **form element**.
- The <input> element has many variations, depending on the **type** attribute.

- eg.

```
<form Name="calc">  
Enter Number 1:<br>  
<input type="text" id="no1" name='no1' ><br>  
</form>
```

JavaScript Events

- An HTML event can be something which the code executes
- Here are some examples of HTML events:
 - 1)An HTML input field was changed
 - 2)An HTML button was clicked
 - 3)Specific keys on the keyboard were typed
 - 4)An image was dragged

- A list of some common HTML events:

onchange()

onclick()

onmouseover()

ondrag()

etc..

- For a complete list follow http://www.w3schools.com/jsref/dom_obj_event.asp

JavaScript HTML DOM

- JavaScript needs to access the HTML elements to modify it or animate it, or to do any job.
- The “document” object of the JavaScript accesses the HTML elements through its various methods available.
- With the help of “document” object the following actions can be performed

1)change the content of HTML elements

2)change the style (CSS) of HTML elements

eg. `document.getElementById("result").innerHTML=table;`

This section will be elaborated with the Calculator Web App that will be shown in the next section

Calculator Web App

- The calculator web App will implement all the concepts taught to you in the previous slides. i.e Implementation of data types, conditional statements, for loops, functions, JavaScript Events and JavaScript DOM.

Idea for Project on JavaScript

- Need to create a simple slideshow.
- Hints
 - 1) just use `window.setTimeout()`; to change the image (if you have any other idea in mind, please feel free to use).
 - 2) Load your function that displays the slideshow along with the page.