Javascript

* + - Developed by Brendan Eich at Netscape Communications as the scripting language for the Netscape Navigator Browser
    - Formerly called Mocha, then LiveScript, then JavaScript
    - Standardized by ECMA International as ECMAScript
    - Latest version : Javascript 1.8.1, ECMAScript 5
    - Common version : Javascript 1.5, Jscript 5.5, ECMAScript v3
    - **JavaScript Frameworks:**
      * Script.aculo.us, JQuery, MooTools, Prototype, Dojo Toolkit..
    - **Linked/Embedded in web pages using the “<script>” element**
      * Linked :
        + <script type=”text/javascript” src=”script.js”> </script>
      * Embedded ( either in the “<head>” or “<body>” element )

<script type=”text/javascript”>

// script goes here

</script>

<noscript> </noscript>

* + - JavaScript + DOM/BOM + CSS + (X)HTML = DHTML
    - JavaScript code in XHTML pages can be executed “on the fly” as the document is rendered.
    - JavaScript code is executed in response to document events
    - **Basic Language features :**
      * **Paradigm** :
        + Object-oriented, functional, imperative scripting language
      * **Java-/C-like syntax**
        + Implicit semicolon insertion for statement termination
        + Identifiers are alphanumeric, \_, ad $ characters
        + Single-line ( // ) or block (/\* \*/) comments
      * **Type system and variable scoping rules :**
        + Dynamic Typing ( loose or weak typing )
        + Global or Local scopes
        + **Data Types :**

**Primitive Types**

Numbers

Booleans

Strings

Undefined and Null

**Composite types ( object )**

Core JavaScript Objects

Object, Number, Boolean, String, Date, Math, Global, RegExp, Error

Arrays

Functions ( Function, Arguments )

DOM Objects

Anchor, Applet, Attr, Comment, DOMException, DOMImplementation, DocumentFragement, Element, Event, Form, Image, Input, Later, Link, Node, Option, Select, Style, Text, TextArea

* + - **Keywords :**
      * Break, case, catch, continue, default, delete, do, else, finally, for, function, if, in, instanceOf, new, return, switch, this, throw, try, typeOf, var, void, with
    - **Reserved words ( unused )**
      * Abstact, Boolean, byte, char, class, const, debugger, float
    - **Statements and control structures**
      * + Var

Used to declare global/local scoped variables

* + - * + If-else

Condition expressions having values of 0, “”, null, and undefined evaluate to false

* + - **Switch-case-default-break**
      * Allows any expression type to be used as the switch expression
      * Case labels may be different types
      * Case labels may be expressions
      * Case execution falls through, unless terminated by a break
    - **While, do-while, for, for-in, break, continue**
      * For while and do-while, false condition expressions similar to if-else
      * for-in used for property enumeration
      * allows labeled break/continue
    - **try-catch-finally, throw**
      * throw and catch can handle any expression type
    - **function, return**
      * JavaScript functions are similar to Java methods except for the following differences :
        + No return value type is specified, and “return” is optional within the function body
        + Functions may return a value on one invocation and not return a value on another invocation
        + Functions may return different types of values on different invocations
        + Functions parameters are dynamically typed
        + Functions can be invoked with an arbitrary number of arguments, regardless of the actual parameters specified in the function definition
        + Functions are first-class objects
      * Functions can be invoked as global functions as methods of specific objects, or as object constructors
    - **with :**
      * used to access object properties without having to explicitly qualify the property with the object name.