Javascript – Course Outline

# Duration

2 days

# Objectives

Web client-side Programming in Javascript (along with Object oriented and functional programming constructs)

# Audience

This course is for Software Engineers with 1 – 4 years of experience in web application development

# Pre-requisite

* Prior experience developing web applications and HTML
* Awareness of Cascading Style Sheets

# Outline

## Day 1 - Session 1 – Javascript as a functional and structural programming Language

### Javscript Programming Language Constructs

**Introduction to JavaScript**

* What is Javascript?
* Power of Javascript – Eagle’s view of what is happening in Javascript world

**Building blocks of JavaScript**

* Variables, functions and comments
* Null, Undefined, Objects, Equality
* Finding Help, Javascript Reference (<http://developer.mozilla.org>)

**Control Flow**

* Conditional Control Flow – If, else and switch
* Iteration – for, for … in, while, do … while
* Error handling – try, catch, finally

**Built-in types and libraries**

* String, Number, Object and Array (Underscore.js)
* Regular Expression, Date (Datejs)
* Library Functions – isNaN, eval, parseFloat, etc
* Math object, JSON

**Debugging Javascript**

* Inspecting HTML
* The Javascript Console
* Inspecting the CSS, Network traffic and Javascript scripts

### Javscript Functional Programming

**JavaScript Functions**

* Invocation, Overloading, The Arguments Object, Recursion
* Different ways of declaring the functions
* Nested / inner functions
* Named functions
* Anonymous functions
* Function as singleton object
* Assigning functions to variables (named and anonymous)
* Event Driven Programming
* Callbacks and Closures

**Functions as Object**

* A blank generic object //var obj = {};
* Similarities between objects and functions
* Caching return results from function
* Function as an object property
* Global object
* Changing the context of a function
* Different ways of changing the context – call, apply
* Implement looping with a callback

**Closures**

* Basic Closure
* Callback functions, timers, event listeners using closures
* Private properties
* Self-executing, temporary function
* Closures and looping
* Anonymous wrapper functions for wrapping libraries

## Day 1 - Session 2 – Javascript as Object Oriented programming Language

### Javascript Objects

**Instantiating Objects**

* Use “new” operator
* Use “this” operator within the object
* Add a new property and method to the object
* Using instanceof
* Using arguments.callee
* Variable arguments – using arguments.length
* Creating (Array().slice) array objects and working with arrays methods
* Modifying Array.prototype, add methods to array (Array.method)
* Exception handling

### Inheritance, Overriding, Overloading, Access Control using Javascript

**Function Prototypes and OOPs**

* Augmenting Types (Function.prototype)
* Adding prototyped method to a function
* Overriding prototyped properties
* Chainable methods
* Using typeof, instanceof, constructor
* Writing default constructor
* Understand prototypal inheritance
* Modifying built-in prototypes
* Potential pitfalls that occur in extension
* Associate context to the original object
* Enforce context to all functions
* Use .bind method from prototype.js
* Determine function’s argument length
* Method overloading
* Declaring private members
* Javascript Object Notation: JSON
* Marshalling and Unmarshalling JSON Data

## Day 2 – Session 1 - Modular programming using Javascript

**Modular javascript**

* Use functions to create modules
* Remove the use of global variables
* Method cascade (returning same object by many methods)
* Method currying to manipulate function values using Function.prototype
* Automatically adding “on”, “fire” and event registry
* Creating namespaces / packages
* Setting up custom MVC architecture from scratch

**Coding practices**

* Avoid global variables, Block scope, Reserved words
* Don’t depend on semicolon insertion
* Inline comments, block comments for documentation
* Typeof, parseInt, + operator, floating point, NaN
* Phony Arrays, Falsy values, hasOwnProperty
* Equality (===, !== vs ==, !=)
* With statement, Continue statement, Blockless statements
* Eval function
* Bitwise operators, Typed wrappers
* Function statement vs Function expression
* Using JSLint

**Asynchronous Communication with Server**

* AJAX
* Making GET and POST calls
* Processing Text and HTML Response
* Processing XML Response
* Handling JSON in AJAX
* Handling JSON request/response in REST environment

## Day 2 – Session 2 – Working with HTML DOM and Browser Object Model using Javascript

**Accessing and manipulating HTML DOM and Browser Object Model (BOM)**

* BOM: Windows methods and events
* Window objects – navigator, frames, location, history, document, screen
* Document API
* Accessing nodes hierarchically, by type, name and value
* Attaching and Detaching events
* Event propagation: capturing and bubbling
* Accessing attribute nodes
* Creating, deleting nodes in the DOM
* Identifying the target of an event
* Javascript – cross browser support in a matrix

**Browser Caching**

* Understand caching in web applications – Browser caches and Proxy caches
* Controlling caches – meta tags, pragma http headers, Expires, Cache-control http headers, validators and validation
* Building a cache aware site
* Javascript versioning and caching
* HTML 5 Caching

**Javascript in Production**

* Coding practices (specifically for production code)
* Javascript minimizers
* Minimize http requests
* Use Content Delivery Network
* Add Expires / Cache-control header
* Gzip components
* Remove duplicate scripts

# Hardware & Network Requirements

* All participants to have individual desktops with at least dual core or higher CPU with 2GB RAM
* All participants’ system to be connected to internet

# Software Requirements

* Windows XP/7 /8 Linux Operating System
* Text editor – SublimeText / Notepad++
* Web browser – Mozilla Firefox with firebug, Google Chrome