HTML5 and Javascript Training

Initial Sessions

General Notes

Day 1 & 2

Research

HTTP protocol

Fiddler

Session

Debugging Skills

Task (Javascript nature)

Single-threaded

Asynchronous event model

Day 3 & 4

Research

Understanding DOM

Event model

Task / Assignment

Event handling

Session

jsDoc commenting

jsLint pass

Day 5 & 6

Research

AJAX / XHR

DOMParser

Task / Assignment

RegEx

Dynamic UI

Day 7 & 8

Research

CSS3 Transform, Transition and Animation

Task / Assignment

CSS3 Transform, Transition and Animation

Day 9-12

Research

Object Oriented Javascript patterns

Task / Assignment

Modularize previous assignments

Dynamic UI-Object initialization

Initial Sessions

* Introduction to version control - SVN and GIT
* Introduction to Single-page-applications

General Notes

* All programs unless explicitly stated are to be tested on common desktop browsers (eg. Chrome, Firefox, IE ). Moreover the stated assignments should be functional on Touch device browsers( Safari on IOs, Chrome on Android, etc ). A program that only runs within one of the browsers shall be marked as incomplete.
* Copy-pasting code from the Internet is incorrect. This is a training exercise. Code from the Internet should be studied. You should always roll out your own implementation.
* Be more familiar with the development environment (IDE)

Day 1 & 2

**Objective**: To grasp the concepts of web architecture and HTML/5 & CSS.

**Research**

* **HTTP protocol**
* Headers
* Methods(GET,POST,etc)
* **Fiddler**
* Introduction to Fiddler for analyzing HTTP request/response

**Session**

* **Debugging Skills**
* A session explaining and introducing developer tools

**Task (Javascript nature)**

* **Single-threaded**
* Provide one example demonstrating the above concept
* **Asynchronous event model**
* Provide one example demonstrating the above concept

Day 3 & 4

Objective:

**Research**

* **Understanding DOM**
* Analysing core objects interfaces (window, document, iframe, etc.)
* **Event model**
* DOM events and general propagation model
* Single-touch and Multi-touch events (touch devices - iPad, Nexus, Surface)

**Task / Assignment**

* **Event handling**
* Understanding event propagation model and their default handling using some basic browser events
* Create a slider control and bound its value 0 to 100
* Create a textbox and use it to show slider value
* Sync both controls (update slider on textbox value change and vice versa)
* Restrict textbox value to numeric and upto 3 digit (Validate while typing)
* Touch event handling and capture in touch-devices
* *Addition to the above exercise:* Provide touch support in slider control of above problem
* Multi-Touch events - pinching and webkit-scale
* *Addition to the above exercise:* Create a container div of size:600x600px, containing a child div element of size:200x200px. Implement following features
* Child div element should be draggable
* Change size of div on pinching
* Move div to the center of container, on double tap

**Session**

* **jsDoc commenting**
* A session explaining and introducing syntax of commenting the code to support JSDoc.
* <http://usejsdoc.org/#Getting_Started>
* **jsLint pass**
* Javascript file should be validated with jsLint
* <http://www.jslint.com/lint.html>
* [http://jslinterrors.com](http://jslinterrors.com/)

Day 5 & 6

Objective:

**Research**

* **AJAX / XHR**
* Example for understanding AJAX
* XHR2 and FormData
* Same-origin policy
* **DOMParser**
* Parsing for XML file

**Task / Assignment**

* **RegEx**
* Generate regular expression to validate a formula (Math equation)
* Regular expression to validate “Cell-Reference”
* It may contain sheet reference(‘*SheetName*’!) or Absolute sign($)
* Cell reference is combination of Column (represented in alphabet) and Row (represented in digit), e.g. A1(Column A, Row 1) or AC32(Column AC, Row 32). A1 can be $A$1 or $A1 or A$1.
* Sheet Name is valid if it contains digit, alphabet, space or this symbols: *~@#$%^&-\_*
* Validate given formula. Formula must start with ‘=’ sign. It may contain operators **+**(plus),**-**(minus),**\***(Multiplication),**/**(Division). It must contain valid cell reference as described above. It may contain ‘(‘ open bracket and ‘)’ close bracket and each open bracket must has a close bracket.
* **Dynamic UI**
* Download XML runtime using ajax, and generate a dynamic UI after parsing
* Example XML  
  <ui>  
   <button rect=”0,0,20,10”>Hello</button>  
   <lable rect=”25,0,30,10”>Welcome</button>  
  </ui>
* Position of control must be relative to its parent. Following controls are part of UI:
* Panel: control that can contain other control
* Label: control that represents simple text label
* Button: control that represents simple text button
* Image: control that represents any image
* TextBox: control that represents textbox (contenteditable div)
* Add localization functionality. Create and Use a string table to set text of control. For example, take text of control from “en\_US.json” to support english language and from “hi\_IN.json” to support hindi language. Support english and any other two languages. Put a way to select language.

Day 7 & 8

Objective:

**Research**

* **CSS3 Transform, Transition and Animation**
* Learn features of CSS3 transform, transition and animation and learn how to use it
* Accelerated Transformation/Transition

**Task / Assignment**

* **CSS3 Transform, Transition and Animation**
* Generate bouncing-ball animation effect
* Using animation key-frame model
* Using requestAnimationFrame

Day 9-12

Objective: Understand Object oriented programming patterns in Javascript

**Research**

* **Object Oriented Javascript patterns**
* Understanding Closure
* Prototypal Inheritance
* Javascript coding pattern: Module pattern (namespace and class mechanism)

**Task / Assignment**

* **Modularize previous assignments**
* Convert previous assignments (slider, dynamic-ui, bouncing-ball) into module-pattern
* The web-application should have following classes:
* Ajax file downloader: to download XML-UI file
* Dynamic-UI generator: accepts XML string, and create HTML DOM objects dynamically and place them in some given parent element
* Bouncing-ball animation container: white board should export play, pause and stop features to control bouncing-ball animation
* **Dynamic UI-Object initialization**
* Dynamically generate UI from XML something like follow,
* <ui>  
   <app rect=”....” title=”app1”>  
   <button .. > Play </button>  
   <button .. > Pause </button>  
   <slider .. > Speed controller </slider>  
   <bb\_board> Bouncing Ball - Board </bb\_board>  
   </app>  
   <app rect=”....” title=”app2”>  
   < Similar to above - second instance of app >  
   </app>  
  </ui>

Example output:  
