**Protractor Framework – TOC**

**Prerequisites:**

* Knowledge/Experience of Testing Applications
* Logical Mind to understand Programming
* Knowledge of Selenium

**Performance Outcome:**

* The participants will be able to write test cases for angular application using jasmine framework

**Hardware & Software:**

* They need any IDE like sublime or netbeans , and other software are open source can be downloaded from web
* Download and Install Permissions For Attendees
* MS Office
* Required Browsers - Chrome, Firefox
* Required OS - Windows 8.1/10 / MAC / Ubuntu

**Duration: 5 Days**

**Day 1**

**JavaScript Fundamentals**

Let’s learn the fundamentals of script building.

* Hello, world!
* Code structure
* The modern mode, "use strict"
* Variables
* Data types
* Type Conversions
* Operators
* Comparisons
* Interaction: alert, prompt, confirm
* Conditional operators: if, '?'
* Logical operators
* Loops: while and for
* The "switch" statement
* Functions
* Function expressions and arrows
* JavaScript specials

**Code quality**

This chapter explains coding practices that we’ll use further in the development.

* Debugging in Chrome
* Coding style
* Comments
* Ninja code
* Automated testing with mocha
* Polyfills

**Objects: the basics**

* Objects
* Garbage collection
* Symbol type
* Object methods, "this"
* Object to primitive conversion
* Constructor, operator "new"

**Data types**

More data structures and more in-depth study of the types.

* Methods of primitives
* Numbers
* Strings
* Arrays
* Array methods
* Iterables
* Map, Set, WeakMap and WeakSet
* Object.keys, values, entries
* Destructuring assignment
* Date and time
* JSON methods, toJSON

**Day 2**

**Advanced working with functions**

* Recursion and stack
* Rest parameters and spread operator
* Closure
* The old "var"
* Global object
* Function object, NFE
* The "new Function" syntax
* Scheduling: setTimeout and setInterval
* Decorators and forwarding, call/apply
* Function binding
* Currying and partials
* Arrow functions revisited

**Objects, classes, inheritance**

In this section we return to objects and learn them even more in-depth.

* Property flags and descriptors
* Property getters and setters
* Prototypal inheritance
* F.prototype
* Native prototypes
* Methods for prototypes
* Class patterns
* Classes
* Class inheritance, super
* Class checking: "instanceof"
* Mixins

**Error handling**

* Error handling, "try..catch"
* Custom errors, extending Error

**Day 3**

**Protractor course content**

**Introduction**

* Understand the importance of Protractor
* Why Node.js? Install Node
* Install Protractor with Web driver plug-in
* Install commands List

**Jasmine framework with Protractor**

* Introduction to Jasmine and its terminology
* Jasmine test configuration and execution
* Jasmine test options
* Parallel execution

**[Getting started with Protractor](javascript:void(0))**

* Introduction to Jasmine and its terminology
* Importance of Configuration file in protractor framework
* File reference
* Creating Protractor TestRunner file to run the scripts
* Install Angular JS plugin and convert to intern project

**Deeper Understanding of Protractor Promises and its working mechanism**

* Important Player Tips for Best learning Experience:
* Javascript Asynchronus and its problem with Protractor
* Importance of Promises and how its solves Asynchronous issue
* Links to read more on promises
* Understand how protractor inbuilt takes care of promise
* Project code download

**Getting Started with Protractor Tests**

* Understand Global variables of Protractor
* Usage of Protractor locators - 1
* Usage of Protractor locators - 2
* Jasmine assertions to validate Protractor tests
* Running Protractor tests on Firefox and Internet Explorer
* Browser configuration file
* Running Protractor tests on Non Angular Sites

**Day 4**

**Deep dive into Protractor Core AP**

* Web applications to Practise Protractor Automation
* Importance of chain locators with example
* Chain locators to identify Child elements-Example
* Getting the list of Elements with All method in Protractor
* Practise exercises on All and chain locators
* Handling list of dropdown options with example

**Handling Mouseactions, Windows, Alerts with Protractor**

* Actions and its importance in the Angular UI
* Examples of handling Autocomplete dropdowns with Actions
* Handling child windows with Protractor
* Examples on validating child windows with Switch Mechanism
* Handling Java Alerts with Protractor

**Frames and Synchronization practical usage**

* What are Frames? Methods to handle in protrator
* Handling synchronization with Protractor

**Framework Part 1- Understand the Javascript objects and their accessibility**

* Framework Learning Plan for JavaScript
* What are JavaScript objects and importance of it
* Accessing the java objects from another JS file

**Day 5**

**Framework Part 2- Page Object Mechanism to drive the objects**

* Page object Mechanism for Protractor tests
* Driving the Page objects with setup&TearDown methods

**Framework Part 3- Package.json and Node basics to trigger tests from Framework**

* OnPrepare functionality in Configuration file for preloading the data
* Understanding the importance of Package.json file
* Package.json configuration properties -1
* Defining Scripts configuration in Package.json to eliminate eclipse testRunner
* npm commands to trigger Test Suites

**Framework Part 4: Generating Client reports&Tagging for Protractor tests**

* Generating excellent client reports for Protractor tests
* Tags configuration to control test scripts execution

**Framework Part 5: Integrating Protractor tests with Jenkins CI too**

* Introduction to Jenkins
* Configuring Jenkins job with protractor tests
* Scheduling Protractor Jenkin job for continuous Integration

**Understanding Version Control System GIT**

* Introduction to GIT
* Importance of Github and its uses
* Creating Git config and repositories
* How to push code to remote repository
* Understanding Staging and commit in git
* Add remote repository and push the committed code
* End to end working example on Git commands -1
* End to end working example on Git commands -2
* Importance of Branching in GIT
* How to resolve Merge conflicts with GIT

**Understanding WebdriverJS**

* Installation of Selenium Webdriver
* Overview of WebDriverJS
* Communication between Chrome Driver and Selenium WebdriverJS
* One hands-on workflow simulation project for example login flow