Selenium Class 21 - Introduction to Selenium WebDriver

i) Introduction to Selenium WebDriver

ii) Automated Testing using Selenium WebDriver

...........................

Test Level: System Testing

Test Type: Functional Testing

Test Method: Automated Testing

Test Tool: Selenium

Note: We conduct Automated Functional Testing using Selenium at System Testing Level....

...........................

i) Introduction to Selenium WebDriver

Selenium Components or Selenium Tool's Suite

1) Selenium IDE
2) Selenium RC (*Out Dated)
3) Selenium WebDriver
4) Selenium Grid

...........................
History of Selenium WebDriver

- In 2006 Selenium WebDriver was launched at Google,

- In 2008, the whole Selenium team decided to merge Selenium WebDriver with Selenium RC in order to form a more powerful tool called selenium 2.0

Versions of Selenium

Selenium 1.0
(Selenium IDE, Selenium RC, Selenium Grid)

Selenium 2.0
(Selenium IDE, Selenium RC, Selenium WebDriver, Selenium Grid)

Selenium 3.0 (Sep 2016)
(Selenium IDE, Selenium WebDriver, Selenium Grid)

Now they are planning to release Selenium 4.0

Selenium WebDriver Features

- Selenium WebDriver is a most important tool in Selenium's Tool Suite

- Selenium WebDriver has Programming Interface (API), no IDE

- Selenium IDE - has IDE (Integrated Development Environment) - Tool features
G C Reddy Technologies (www.gcreddy.com)

(No Programming)

- Selenium WebDriver - has Programming Interface but no IDE (No tool features)

- UFT/QTP - has both IDE as well as programming Interface

- Selenium WebDriver supports various Operating Environments to write & execute Test cases

   MS Windows
   Linux
   Macintosh etc...

- Selenium supports various Browsers to write and execute Test Cases

   Google Chrome,
   Mozilla Firefox,
   MS Edge,
   Safari,
   Opera etc...

- Selenium WebDriver supports various Programming languages to write Test Scripts/Test Cases

   Java,
   Python,
   C#.NET,
Perl, 
Ruby, 
PHP,

- Selenium WebDriver supports Batch Testing, Data Driven Testing, Cross Browser Testing, and Database Testing

- Selenium WebDriver is faster in Test Execution over other tools in Selenium suite

- Selenium WebDriver supports Parallel Testing with the help of Testing Framework

Drawbacks of Selenium WebDriver

1) Selenium WebDriver has no IDE, only Programming Interface
   (Takes more time and efforts to create Test cases)

2) It doesn't have built-in Result Reporting facility
   (Using Programming features, Using Testing Framework Assert Methods we can create Result/s)

3) No Other Tool Integration for Test management
   (We can use Test Management tool separately)

4) No Centralized maintenance of Elements/Objects
   (Using POM - Page Object Model we can create Object Repositories)

5) It supports only Web based Applications
6) No reliable Technical support from anybody

7) It doesn’t recognize Flash Objects, Windows Objects in web pages
(Using Sikuli tool integration we can handle Flash Objects and Windows objects, Using Programming features (Ex: Alert Class/Object in Java) we can handle windows objects)

8) New Features may not work properly

9) Environment Setup is some difficult when it compares to commercial tools like UFT, RFT etc...
Etc...

**ii) Automated Testing using Selenium WebDriver**

*How to create Test Cases / Test Scripts*

**Selenium IDE:**

Selenium IDE Test Cases can be created using Element Locators and Selenese commands / Selenium IDE Commands

**Selenium WebDriver:**

Selenium WebDriver Test Cases can be created using Element Locators and WebDriver API Commands
UFT/QTP:

UFT Test Cases can be created using Test Objects and Test Methods

............................

Elements / Objects / Software Objects

Software Objects in Web Applications,

Edit Box, Text Box, Link, Button, Drop down Box, List Box, Combo Box, Image, Image Button, Image Link, Radio Button, Checkbox, Web Table, Frame etc...

Element in Selenium Webdriver is equals to Object in UFT/QTP and Command in Selenium Webdriver is equals to Test Method in UFT/QTP

............................

Using Element Locator and WebDriver Command we can create Selenium Test Cases,

- Element Locator - to identify / recognize / locate an element in Web Pages
- WebDriver Command - to perform Operation on the Element

Example 1:
Test Step: Click "Gmail" link in Google Home Page

Gmail (Link) Element/Object -Element Locator
Click - Operation on the Element -WebDriver Command

Example 2:
Test Step: Enter a value into Username edit box in gcrShop Admin interface
Username (Edit box) - Element / Object - Element locator
Enter a Value - Operation on the Element - WebDriver Command
Parameter - Username value - abcdef@gmail.com

Test Step: Click "Gmail" Link in Google Home page

**UFT Test Script:**

Browser("Google").Page("Google").Link("Gmail").Click

**Selenium WebDriver Test Step:**

driverObject.findElement(By.linkText("Gmail")).Click

driverObject - Browser driver (user defined),
findElement - WebDriver Command,
By - Selenium predefined Class,
linkText - Element Locator (HTML),
"Gmail" - Locator value provided by Web Developer
Click - WebDriver Command,

**UFT/QTP**

<table>
<thead>
<tr>
<th>Object</th>
<th>Property</th>
<th>Value</th>
<th>Operation</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Button</td>
<td>text</td>
<td>OK</td>
<td>Click</td>
<td>NA</td>
</tr>
<tr>
<td>Edit Box</td>
<td>attached text</td>
<td>Agent Name:</td>
<td>Set</td>
<td>ABCDE</td>
</tr>
</tbody>
</table>

**Selenium WebDriver**

<table>
<thead>
<tr>
<th>Element</th>
<th>Locator</th>
<th>Value</th>
<th>Operation</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Button</td>
<td>id</td>
<td>&quot;tdb1&quot;</td>
<td>click</td>
<td>NA</td>
</tr>
<tr>
<td>Edit Box</td>
<td>name</td>
<td>&quot;username&quot;</td>
<td>sendKeys</td>
<td>ABCDE</td>
</tr>
</tbody>
</table>
### General Object

<table>
<thead>
<tr>
<th>Object/Element</th>
<th>Property/Locator</th>
<th>Value</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dog</td>
<td>Color</td>
<td>black</td>
<td>jump</td>
</tr>
<tr>
<td>Man</td>
<td>name</td>
<td>Venkat</td>
<td>read</td>
</tr>
</tbody>
</table>