**How do I validate a page in Selenium WebDriver?**

To write a good Automation script, it is important to put checks and validations at every possible point. Validating page involved the checking of elements present on the page.

Some of the Validations are:

* **Page Title Validation** – Verifying page title by a method called getTitle().
* **Page URL Validation** – Verifying page URL by method called getCurrentUrl().
* **Scroll Down Operation** – Scroll Down operation is performed by invoking JavaScript.
* **Hidden Element Verification**– Verifying whether Web element is visible or hidden on a page by method isDisplayed ().
* **Checkbox Validation** – Verifying whether Checkbox is checked or not by method isEnabled().
* **Radio Button Validation** – Verifying whether Radio button is selected or not by method isSelected().

# Which version of Selenium webdriver is stable for Firefox 38?

Use Selenium 2.53.0 or above but below 3.

# How do I learn Selenium Webdrive to clear interview?

his depends on what you inform the interviewer.

Scenario 1 (you are needed to be part of a team developing automation framework)

The question will range from

1. What framework have you used
2. Which language you have knowledge about
3. What was the application you automated
4. How was the reporting features
5. How did you Handel multiple browsers
6. How did you start with framework
7. How much time did you develop and be part in core features of framework

So you can see the interview will try to scrutinise you and try to gain knowledge about you being and useful resource.

Scenario 2 (you are required for maintaince of the framework)

The question will be

1. Have you executed scripts
2. How did you manage failure
3. Did you change scripts if it failed
4. How did you manage locators
5. Did you update locators
6. Explain your roles and responsibilities in execution
7. Did you write scripts or automated test cases

So the question will be a lesser of roles and responsibilities types.

Now for you, what needs to done to clear an interview.

1. If you want to an s/w engineer in test.
2. You need to know one programming knowledge
3. You need to k ow one automation tool(selenium, qtp..etc)
4. You need to know types of framework
5. You need to know reporting functionality
6. You need to know application than can or can't be automated.

Learning is not difficult, start learning from seleniumeasy, toolsq.

Look up YouTube, for people explaining frameworks.

Try automating applications, be confident about the flow of automation frameworks.

# How does the Selenium WebDriver work?

There are 2 ways of explaining how Selenium WebDriver works:

* easy
* technical

Lets see how easy is the first one.

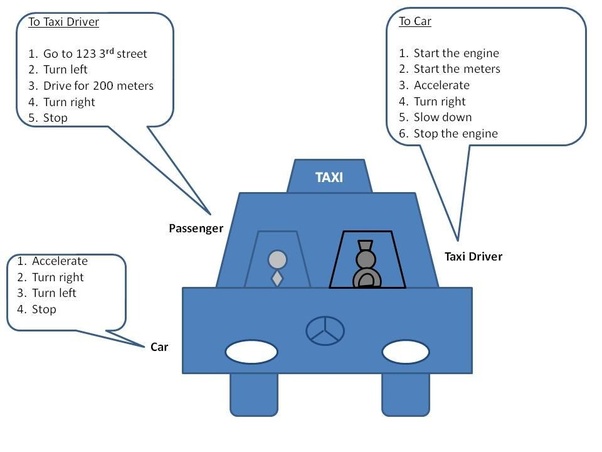
It uses an analogy with taxi driving.

[Selenium WebDriver drives a browser the same way a taxi driver drives a cab](http://test-able.blogspot.com/2016/01/webdriver-test-automation-like-taxi-driving.html).

In taxi driving, there are typically 3 actors:

* **the client;**he tells the taxi driver where he wants to go and how to get there
* **the taxi driver;** he executes the client's requests; the taxi driver sends his own requests to the car
* **the car;**the car executes the taxi driver's requests

The client gets to the destination through dialogues that happen between the client - taxi driver and taxi driver - car.

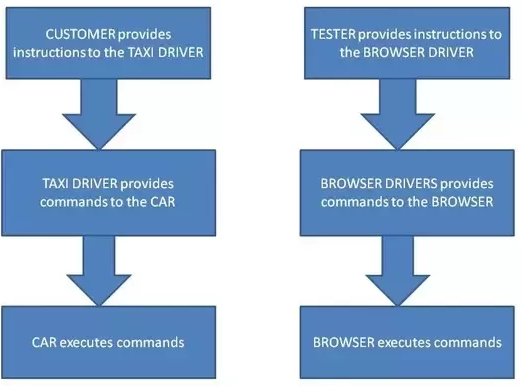


In test automation with Selenium WebDriver (and other tools), there are 3 actors as well:

* **test engineer that writes the automation code**; the automation code sends requests to the browser driver component
* **the browser driver component**; it executes the test engineer requests; it sends its own request to the browser
* **the browser**; it executes the browser driver requests

So this is the analogy:

1. the test engineer is like a taxi client
2. the browser driver is like a taxi driver
3. the browser is like a taxi



Like most technical explanations, there are no photos either :(

When the automation script is executed, the following steps happen:

* **for each Selenium command, a HTTP request is created and sent to the browser driver**
* the browser driver **uses a HTTP server for getting the HTTP requests**
* **the HTTP server**determines the **steps needed for implementing the Selenium command**
* the implementation steps are executed on the browser
* the execution status is sent back to the HTTP server
* the **HTTP server sends the status back to the automation script**