

Selenium

Selenium이란 무엇인가?

- Selenium is an open-source tool that is used for test automation
- Apache License 2.0
- 웹 응용 어플리케이션을 테스트하기 위한 오픈 소스 자동화 테스트 툴임
- 제품군
 - Selenium IDE
 - Selenium RC(Remote Control)
 - Selenium WebDriver

Selenium 장점

Selenium	QTP
Selenium은 오픈 소스 도구입니다.	QTP는 상용 도구이며 각 라이선스에 비용이 소요됩니다.
DOM을 폭로하는 다양한 기술을 확장 할 수 있습니다.	제한된 추가 기능 및 각 기술에 대한 추가 기능이 필요합니다.
여러 브라우저에서 스크립트를 실행할 수 있는 기능이 있습니다.	Firefox, IE 및 Chrome의 특정 버전에서 테스트를 실행할 수 있습니다.
다양한 운영 체제에서 스크립트를 실행할 수 있습니다.	Windows에서만 작동합니다.
모바일 장치를 지원합니다.	타사 도구를 사용하여 모바일 장치를 지원합니다.
브라우저 내에서 테스트를 실행하므로 스크립트 실행이 진행되는 동안 포커스가 필요하지 않습니다.	요구 사항 도구가 브라우저에서 작동하므로 스크립트 실행 중에 초점을 맞 춥니 다 (사용자 동작 모방).
Selenium Grid를 사용하여 테스트를 병렬로 실행할 수 있습니다.	QTP는 테스트를 병렬로 실행할 수 없지만 QTP를 QC와 통합하면 테스터가 병렬로 실행할 수 있습니다. QC는 상업적 도구이기도합니다.

Selenium 단점

Selenium	QTP
웹 기반 응용 프로그램 만 지원합니다.	웹 및 데스크톱 응용 프로그램을 모두 테스트 할 수 있습니다.
Object Repository / Recovery Scenario와 같은 기능이 없습니다.	QTP에는 개체 저장소 및 복구 시나리오가 내장되어 있습니다.
IDE가 없으므로 스크립트 개발은 QTP만큼 빠르지 않습니다.	더 직관적 인 IDE; 자동화가 더 빨리 이루어질 수 있습니다.
브라우저 내의 컨트롤에 액세스 할 수 없습니다.	즐거 찾기 모음, 뒤로 및 앞으로 단추와 같은 브라우저 내의 컨트롤에 액세스 할 수 있습니다.
기본 테스트 보고서 생성이 없습니다.	도구 내에서 기본 테스트 결과 생성.
매개 변수화를 위해 사용자는 프로그래밍 언어에 의존해야 합니다.	매개 변수화는 기본 제공되어 구현하기 쉽습니다.

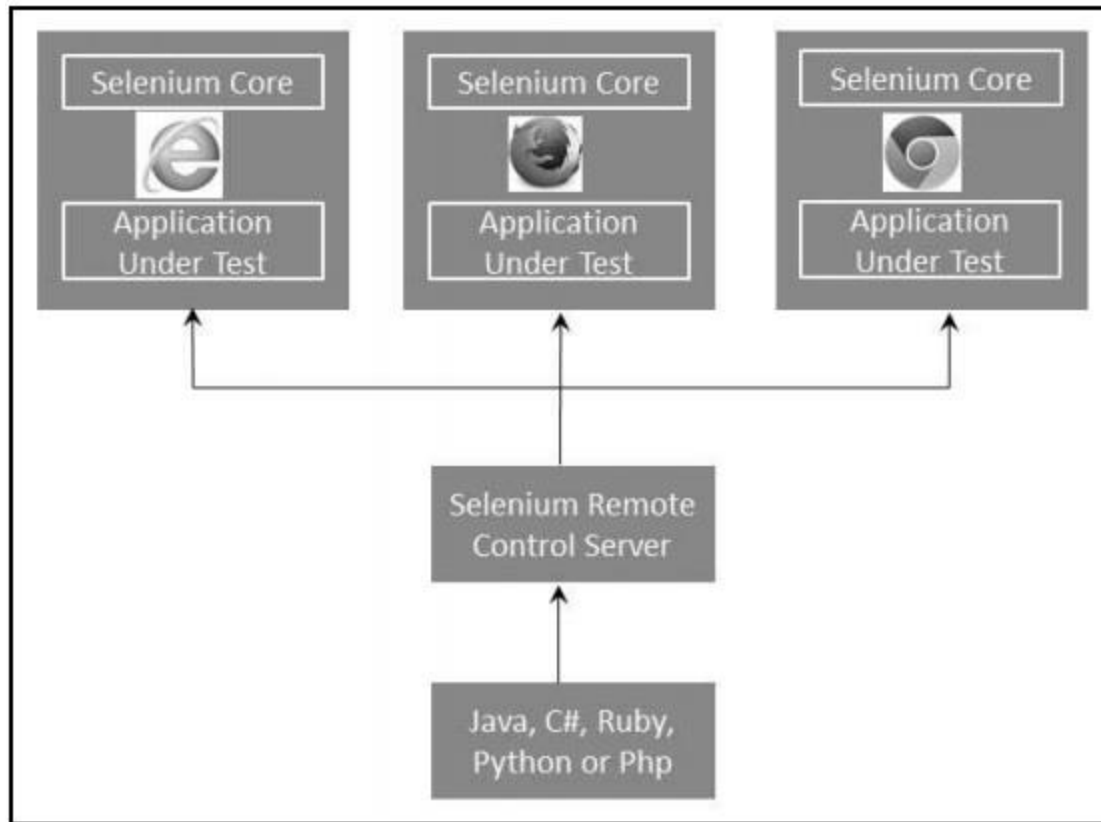
Selenium 환경 설정

- Java 다운로드 및 설치
- Eclipse 다운로드 및 설치
- FireBug 및 FirePath 구성
- Selenium RC 구성
- Selenium WebDriver 구성

- 파이썬은
 - `pip install selenium`

Selenium 아키텍처

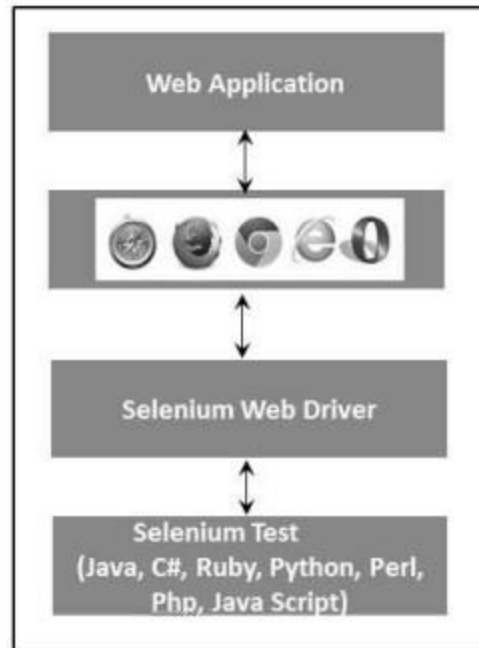
- 웹브라우저는 Javascript인터프리터를 사용하여 Selenium 명령을 실행함



- Selenium Server가 실행되어 브라우저가 종료됨. 그 외에도 Selenese명령을 해석하고 실행함. 웹브라우저와 테스트중인 응용 프로그램 간에 전달되는 http메세지를 가로 채서 확인하는 http프록시 역할을 함

Selenium WebDriver

- WebDriver는 웹 응용 어플리케이션을 위한 자동화 도구임.
- WebDriver는 서버에 의존하는 Selenium RS와 달리 중개자가 없이 웹브라우저와 직접 상호 작용을함.
 - Selenium RC에서 잘 지원되지 않는 웹브라우저를 지원하고 다중 웹브라우저 테스트 기능을 제공함
 - 여러가지 프레임 워크, 다중 웹브라우저창, 팝업창, 경고창을 제어 할 수 있음
 - 고급 탐색기능과 드래그 앤 드롭 기능을 제공함
 - AJAX기반 UI 요소 지원



Selenium WebDriver

	명령 및 설명
1	driver.get ("URL") 애플리케이션으로 이동함
2	element.sendKeys ("inputtext") 입력 상자에 텍스트를 입력함.
삼	element.clear () 입력 상자에서 내용을 지움
4	select.deselectAll () 페이지의 첫 번째 SELECT에서 모든 OPTION을 선택함
5	select.selectByVisibleText ("some text") 사용자가 지정한 입력으로 OPTION을 선택함
6	driver.switchTo (). window ("windowName") 한 창에서 다른 창으로 초점을 이동함
7	driver.switchTo (). frame ("frameName") 프레임간에 스위칭함
8	driver.switchTo (). alert () 경고 처리
9	driver.navigate (). to ("URL") URL로 이동함
10	driver.navigate (). forward () 앞으로 탐색함
11	driver.navigate (). back () 뒤로 이동함
12	driver.close () 드라이버와 관련된 현재 브라우저를 닫음
13	driver.quit () 드라이버를 종료하고 해당 드라이버의 모든 관련 창을 닫음
14	driver.refresh () 현재 페이지를 새로 고침

Locators

- Selenium WebDriver의 요소 찾기는 WebDriver 및 WebElement 클래스에서 제공하는 findElement() 및 findElements() 메소드를 사용하여 수행함

Method	Syntax	설명
By ID	driver.findElement (By.id (<요소 ID>))	ID 속성을 사용하여 요소 찾기
By name	driver.findElement (By.name (<요소 이름>))	Name 속성을 사용하여 요소 찾기
By class name	driver.findElement (By.className (<element class>))	Class 특성을 사용하여 요소 찾기
By tag name	driver.findElement (By.tagName (<htmltagname>))	HTML 태그를 사용하여 요소 찾기
By link text	driver.findElement (By.linkText (<linktext>))	링크 텍스트를 사용하여 링크 찾기
By partial link text	driver.findElement (By.partialLinkText (<linktext>))	링크의 부분 텍스트를 사용하여 링크 찾기
By CSS	driver.findElement (By.cssSelector (<css selector>))	CSS 선택기를 사용하여 요소 찾기
By XPath	driver.findElement (By.xpath (<xpath>))	XPath 쿼리를 사용하여 요소 찾기