

Selenium Cheat Sheet 2

Driver Initialization

Basic

```
WebDriver driver;  
driver = new ChromeDriver();  
driver = new FirefoxDriver();  
driver = new SafariDriver();
```

Advanced

FirefoxProfile

```
// load firefox from non default location  
System.setProperty("webdriver.firefox.bin",  
    "path/to/firefox/binary");
```

```
FirefoxProfile fp = new FirefoxProfile();  
// load with firefox addons  
File file = new File("path/to/extension.xpi");  
fp.addExtension(file);  
// other settings on the profile...  
DesiredCapabilities dc = DesiredCapabilities.firefox();  
dc.setCapability(FirefoxDriver.PROFILE, fp);  
WebDriver driver = new RemoteWebDriver(dc);
```

Chrome Options

```
System.setProperty("webdriver.chrome.driver",  
    "/path/to/chromedriver.exe");  
ChromeOptions options = new ChromeOptions();  
File extnFile = new File("/path/to/extension.crx");  
options.addExtensions(extnFile);  
DesiredCapabilities dc = DesiredCapabilities.chrome();  
dc.setCapability(ChromeOptions.CAPABILITY, options);  
WebDriver driver = new RemoteWebDriver(dc);
```

Proxy

```
String PROXY = "localhost:8080";  
Proxy proxy = new Proxy();  
proxy.setHttpProxy(PROXY)  
    .setFtpProxy(PROXY)  
    .setSslProxy(PROXY);  
DesiredCapabilities cap = new DesiredCapabilities();  
cap.setCapability(CapabilityType.PROXY, proxy);  
// set for the corresponding browser  
WebDriver driver = new FirefoxDriver(cap);  
WebDriver driver = new ChromeDriver(cap);
```

Navigation

```
// navigate to url  
driver.get("http://example.com");  
driver.navigate().to("http://example.com");  
// refresh page  
driver.navigate().refresh();  
// navigate forwards in browser history  
driver.navigate().forward();  
// navigate backwards in browser history  
driver.navigate().back();
```



Selenium WebDriver Cheat Sheet

Locators

By.id: (Most preferred)

By.name **Single (WebElement)**
driver.findElement()

By.className

By.tagName

By.linkText

By.partialLinkText

By.cssSelector

By.xpath

List (List<WebElement>)
driver.findElements()

Operations

```
WebElement element;  
element.click(); //click item/button  
element.sendKeys("String to type");  
element.submit(); //submit form  
element.clear();  
  
element.getText();  
element.getAttribute("attribute");  
element.isDisplayed();  
element.isSelected();  
element.isEnabled();  
  
// list box  
Select select = new Select(element);  
select.selectByVisibleText("One");  
select.selectByValue("value1");  
select.selectByIndex(1);  
select.deselectAll();  
select.selectAll();  
  
WebDriver driver;  
driver.getTitle();  
driver.getCurrentUrl();  
  
driver.getPageSource();  
driver.close(); //close current window  
driver.quit(); //quite webdriver  
  
driver.manage().window().maximize();  
driver.manage().getCookies();  
driver.manage().deleteAllCookies();  
driver.manage().window().maximize();  
driver.manage().getCookies();  
driver.manage().deleteAllCookies();
```

Windows

```
String handle = driver.getWindowHandle();  
Set<String> handles = getWindowHandles();  
driver.switchTo().window(handle);  
  
/* How to switch to newly created window */  
String curWindow = driver.getWindowHandle();  
// perform action to create new window...  
// get all window handles  
Set<String> handles = getWindowHandles();  
for(String handle : handles) {  
    if(!handle.equals(curWindow) {  
        driver.switchTo().window(handle);  
    }  
}  
}
```

Frames

```
driver.switchTo().frame(1); // using frame index  
driver.switchTo().frame("name") // using name of frame  
driver.switchTo().frame(element) // Web Element object  
  
driver.switchTo().defaultContent(); //back to main document
```

Alerts

```
driver.switchTo().alert().getText();  
driver.switchTo().alert().accept();  
driver.switchTo().alert().dismiss();  
driver.switchTo().alert().sendKeys("Text");
```