

Test Automation Engineering Fundamentals: Java + Selenium WebDriver

Styles, Navigation
and browsers

**GROW
CONFI
DENTLY**

Selenium WebDriver basics

- **Styles**
- **Browser functionalities**
- **Using different browsers**

Styles

Test Automation Engineering Fundamentals: Java + Selenium WebDriver



Sometimes we need to check the styles of element to check such things as:

- **Correct color of button;**
- **Correct font size;**
- **Correct image in the background.**

If element has attribute “style” then we can get just that, but this is not being used by most people. Instead all styles are saved as CSS. We can get that value by function: “element.getCssValue(“css_value”)”. E.g.:

- `element.getCssValue("background-color")`
- `element.getCssValue("font-size")`

All the styles can be seen in the “Computed” tab in the dev tool in browser.

Computed tab example

Styles	Computed	Event Listeners	DOM Breakpoints	Properties
Filter		<input type="checkbox"/> Show all		
▶ background-color	<input type="checkbox"/> rgb(241, 241, 2...			
▶ box-sizing	border-box			
▶ color	<input checked="" type="checkbox"/> rgb(0, 0, 0)			
▶ display	block			
▶ font-family	Verdana, sans-ser...			
▶ font-size	15px			
height	244px			
▶ line-height	22.5px			
▶ margin-bottom	16px			
▶ padding-bottom	64px			
▶ padding-top	64px			
▶ text-align	center			
▶ text-size-adjust	100%			
width	1583px			
Rendered Fonts				

Sample 8

Let's explore example in Sample8.java

Browser functionalities



**Test Automation Engineering
Fundamentals: Java +
Selenium WebDriver**

Browser functionalities

During other test we may need to simulate user activity, such as refreshing a page or navigating back and forward between pages or resizing window.

Navigation can be done by using:

- `driver.navigate().refresh();`
- `driver.navigate().back();`
- `driver.navigate().forward();`

While windows resizing is done with:

- `driver.manage().window().setSize(new Dimension(800, 600));`
- `driver.manage().window().maximize();`

Using different browsers



**Test Automation Engineering
Fundamentals: Java +
Selenium WebDriver**

Using different browsers

In most cases we do not support just one browser or have some default browser, which is not Firefox, Selenium WebDriver allows us to use other browsers, however additional implementation and drivers are required. Such as “chromedriver” or “IEDriverServer” (for Chrome or Internet Explorer).

We also may run tests on headless browsers.

“A headless browser is a web browser without a graphical user interface. Headless browsers provide automated control of a web page in an environment similar to popular web browsers, but are executed via a command-line interface or using network communication.”

© *Wikipedia*

An example of headless browser is phantomJS, which runs without us seeing actual execution and still providing the test results and. Execution in headless browser is faster.

Declarations for different browsers

✓ **Chrome:**

```
System.setProperty("webdriver.chrome.driver", "path_to\\chromedriver.exe");  
driver = new ChromeDriver();
```

✓ **Internet Explorer:**

```
System.setProperty("webdriver.ie.driver", "path_to\\IEDriverServer.exe");  
driver = new InternetExplorerDriver();
```

✓ **phantomJS:**

```
System.setProperty("phantomjs.binary.path", libWithDriversLocation +  
"phantomjs.exe");  
driver = new PhantomJSDriver();
```

QUESTIONS

