

< Teach
Me
Skills />

Selenium WebDriver - основы



Важное отступление

HTML-теги — основа языка HTML. Теги используются для разграничения начала и конца элементов в разметке. Каждый HTML-документ состоит из дерева HTML-элементов и текста. Каждый HTML-элемент обозначается начальным (открывающим) и конечным (закрывающим) тегом. Например `<p>Some Text</p>` - это тег `p`, содержащий текст "Some Text"

HTML-атрибуты сообщают браузеру, каким образом должен отображаться тот или иной элемент страницы. Атрибуты позволяют сделать более разнообразными внешний вид информации, добавляемой с помощью одинаковых тегов. Например `<p href="value">` - это тег `p` имеющий атрибут `href`, значение атрибута `href` - "value"

Из чего состоит Selenium?



1

Selenium WebDriver - библиотека для различных языков программирования, позволяющая взаимодействовать с браузерами, используя код и драйвер браузера

2

Selenium IDE - приложение, позволяющее записывать последовательность действий в браузере и запускать их

3

Selenium GRID - это приложение, позволяющее распараллеливать тесты на разных конфигурациях операционных систем и браузеров

WebDriver - типы

```
WebDriver chrome = new ChromeDriver();  
WebDriver firefox = new FirefoxDriver();  
WebDriver edge = new EdgeDriver();  
WebDriver opera = new OperaDriver();  
WebDriver safari = new SafariDriver();  
WebDriver ie = new InternetExplorerDriver();  
WebDriver remote = new RemoteWebDriver(new URL(""), new  
ChromeOptions());
```

Все браузеры и их системные переменные



WebDriver - действия













```
driver.get("http://onliner.by");  
driver.close();  
driver.quit();
```



- ▶ Открыть Браузер/URL
- ▶ Закрывать вкладку браузера
- ▶ Закрывать браузер

WebDriver - действия

```
driver.manage().window().
```







 	<code>setSize(Dimension dimension)</code>	<code>void</code>
 	<code>maximize()</code>	<code>void</code>
 	<code>fullscreen()</code>	<code>void</code>
 	<code>getPosition()</code>	<code>Point</code>
 	<code>getSize()</code>	<code>Dimension</code>
 	<code>setPosition(Point point)</code>	<code>void</code>

Работа с размерами окна



WebDriver - действия

```
driver.manage().timeouts().
```

 	<code>implicitlyWait(long l, TimeUnit timeUnit)</code>	Timeouts
 	<code>pageLoadTimeout(long l, TimeUnit timeUnit)</code>	Timeouts
 	<code>setScriptTimeout(long l, TimeUnit timeUnit)</code>	Timeouts

```
driver.manage().timeouts().implicitlyWait(20,  
TimeUnit.SECONDS);
```



Работа с таймаутами

Cookie











Ку́ки (англ. cookie, буквально — печенье) — небольшой фрагмент данных, отправленный веб-сервером и хранимый на компьютере пользователя. Веб-клиент (обычно веб-браузер) всякий раз при попытке открыть страницу соответствующего сайта пересылает этот фрагмент данных веб-серверу в составе HTTP-запроса.

○ Применяется для сохранения данных на стороне пользователя, на практике обычно используется для:

- аутентификации пользователя;
- хранения персональных предпочтений и настроек пользователя;
- отслеживания состояния сеанса доступа пользователя;

WebDriver - действия

```
driver.manage().
```


 timeouts()	Timeouts
 window()	Window
 addCookie (Cookie cookie)	void
 deleteAllCookies()	void
 deleteCookie (Cookie cookie)	void
 deleteCookieNamed (String s)	void
 getCookieNamed (String s)	Cookie
 getCookies()	Set<Cookie>
 ime()	ImeHandler
 logs()	Logs

Работа с cookies



WebDriver - действия

```
driver.|
```

 	<code>getWindowHandle()</code>	<code>String</code>
 	<code>getWindowHandles()</code>	<code>Set<String></code>
 	<code>navigate()</code>	<code>Navigation</code>
 	<code>switchTo()</code>	<code>TargetLocator</code>

Навигация назад/вперед




Переключение в/из фрейма

☐ Переключение между
вкладками браузера



WebDriver - действия

```
driver.|
```

 <code>getCurrentUrl()</code>	<code>String</code>
 <code>getPageSource()</code>	<code>String</code>
 <code>getTitle()</code>	<code>String</code>

Получить текущий URL

Получить код страницы


☐ Получить title текущей вкладки

(отображается в имени вкладки в
браузере)



WebDriver - действия

driver.find


 **findElement**(By by) WebElement

 **findElements**(By by) List<WebElement>

^↓ and ^↑ will move caret down and up in the editor >>>



Поиск одного элемента. Если не найден - NoSuchElementException

 Поиск нескольких элементов. На выходе - коллекция, если не найдены - пустая коллекция

А как ищем? Класс By

- `driver.findElement(By.id(""));`
- `driver.findElement(By.name(""));`
- `driver.findElement(By.cssSelector(""));`
- `driver.findElement(By.className(""));`
- `driver.findElement(By.tagName(""));`
- `driver.findElement(By.linkText(""));`
- `driver.findElement(By.partialLinkText(""));`
- `driver.findElement(By.xpath(""));`



WebElement - действия

```
WebElement element = driver.findElement(By.id("some-id"));  
element.getAttribute("value");  
element.getAttribute("innerText");  
element.getText();  
element.getCssValue("font-size");  
element.getTagName();
```

- Получение атрибута, любого, в том числе текста, если он не виден
- Получение текста из всех вложенных элементов
- Получение текста, css-свойста и имени тега



WebElement - действия

```
WebElement element = driver.findElement(By.id("some-id"));  
element.click();  
element.sendKeys("text");  
element.submit();  
element.clear();
```

- Клик
- Введение текста
- Подтверждение формы
- Очистка текста внутри элемента



WebElement - проверки

```
WebElement element = driver.findElement(By.id("some-id"));  
element.isSelected();  
element.isDisplayed();  
element.isEnabled();
```

- Выбран ли элемент из списка или включен ли checkbox
- Отображается ли элемент на странице (атрибут hidden)
- Включен ли элемент (допустим активна ли кнопка)



Клики бывают разные



1

`element.click()` - обычный клик через Selenium WebDriver

2

`((JavascriptExecutor) driver).executeScript("arguments[0].click();", element);`

Клик с использованием Javascript













3

`new Actions(driver).click(element).perform();`

Клик с захватом действий мышки

1 8 Select - действия

```
Select select = new Select(element);  
select.
```

 selectByVisibleText (String text)	void
 getFirstSelectedOption ()	WebElement
 deselectAll ()	void
 deselectByIndex (int index)	void
 deselectByValue (String value)	void
 deselectByVisibleText (String text)	void
 getAllSelectedOptions ()	List<WebElement>
 getOptions ()	List<WebElement>
 getWrappedElement ()	WebElement
 isMultiple ()	boolean
 selectByIndex (int index)	void
 selectByValue (String value)	void

^↓ and ^↑ will move caret down and up in the editor >>

π

Select - это класс обертка для тега
<select>

Имеет ряд методов для выбора по
тексте/value/номеру опции

