ipython

from selenium import webdriver

driver = webdriver.Chrome()

driver.get("http://localhost/litecart/en/")

login = driver.find\_element\_by\_css\_selector("[name=login]")

login.get\_attribute("value")

login.get\_attribute("textContent")

login.get\_attribute("outerHTML")

driver.find\_element\_by\_css\_selector("[name=query]").get\_attribute("value")

driver.find\_element\_by\_css\_selector("[name=query]").text

driver.find\_element\_by\_css\_selector("#breadcrumbs a").get\_attribute("href")

driver.find\_element\_by\_css\_selector("[name=currency\_code] [value=USD]").get\_attribute("selected")

login = driver.find\_element\_by\_css\_selector("[class=regular-price]")

login.get\_attribute("value")

login.get\_attribute("color")

**Ссылки**

Про свойства и атрибуты -- <http://javascript.ru/tutorial/dom/attributes>

**Примеры кода**

**Java**

String href = link.getAttribute("href");

**C#**

string href = link.GetAttribute("href");

**Python**

href = link.get\_attribute("href")

**Ruby**

href = link.attribute("href")

**JavaScript**

var href = link.getAttribute("href");

#### Ссылки

Правила определения видимости:  
<https://w3c.github.io/webdriver/webdriver-spec.html#element-displayedness>

Примеры невидимых элементов:  
<https://output.jsbin.com/saqoca/2>

#### Примеры кода

##### Java

if (button.isDisplayed()) {  
 button.click();  
}

##### C#

if (button.Displayed)  
{  
 button.Click();  
}

##### Python

if button.is\_displayed():  
 button.click()

##### Ruby

if button.displayed?  
 buton.click()  
end

##### JavaScript

button.isDisplayed().then(function() {  
 button.click();  
});

<http://output.jsbin.com/saqoca>

#### Ссылки

Правила получения текста -- <https://w3c.github.io/webdriver/webdriver-spec.html#get-element-text>

#### Примеры кода

##### Java

String txt = link.getText();

##### C#

string txt = link.Text;

##### Python

txt = link.text

##### Ruby

txt = link.text

##### JavaScript

var txt = link.getText();

#### Примеры кода

##### Java

String color = link.getCssValue("color");

##### C#

string color = link.GetCssValue("color");

##### Python

color = link.value\_of\_css\_property("color")

##### Ruby

color = link.css\_value("color")  
# альтернативный способ  
color = link.style("color")

##### JavaScript

var color = link.getCssValue("color");

ipython

from selenium import webdriver

driver = webdriver.Chrome()

driver.get("http://selenium2.ru")

driver.find\_element\_by\_css\_selector("img.alignleft").value\_of\_css\_property("color")

driver.find\_element\_by\_css\_selector("img.alignleft ").value\_of\_css\_property("background-color")

driver.find\_element\_by\_css\_selector("img.alignleft ").value\_of\_css\_property("border-color")

driver.find\_element\_by\_css\_selector("img.alignleft ").value\_of\_css\_property("background-image")

driver.quit()

driver = webdriver.Firefox() (driver = webdriver.Ie() – does not work)

driver.get("http://selenium2.ru")

driver.find\_element\_by\_css\_selector("img.alignleft").value\_of\_css\_property("color")

driver.find\_element\_by\_css\_selector("img.alignleft ").value\_of\_css\_property("background-color")

driver.find\_element\_by\_css\_selector("img.alignleft ").value\_of\_css\_property("border-color")

driver.find\_element\_by\_css\_selector("img.alignleft ").value\_of\_css\_property("background-image")

driver.quit()

#### Ссылки

Про свойства и атрибуты -- <http://javascript.ru/tutorial/dom/attributes>

#### Примеры кода

##### Java

Point location = link.getLocation();  
Dimension size = link.getSize();  
// "два в одном" -- новый способ, который пока не все драйверы поддерживают  
Rectangle rect = link.getRect();

##### C#

Point location = link.Location;  
Size size = link.Size;

##### Python

location = link.location  
size = link.size

##### Ruby

location = link.location  
size = link.size

##### JavaScript

var location = link.getLocation();  
var size = link.getSize();

ДЗ

## [x] Задание 9. Проверить сортировку стран и геозон в админке

Сделайте сценарии, которые проверяют сортировку стран и геозон (штатов) в учебном приложении litecart.

1) на странице <http://localhost/litecart/admin/?app=countries&doc=countries>  
а) проверить, что страны расположены в алфавитном порядке  
б) для тех стран, у которых количество зон отлично от нуля -- открыть страницу этой страны и там проверить, что зоны расположены в алфавитном порядке

2) на странице <http://localhost/litecart/admin/?app=geo_zones&doc=geo_zones>  
зайти в каждую из стран и проверить, что зоны расположены в алфавитном порядке

Можно оформить сценарии либо как тесты, либо как отдельные исполняемые файлы.

Если возникают проблемы с выбором локаторов для поиска элементов -- обращайтесь в чат за помощью.

-----

Уложите созданный файл, содержащий сценарий, в ранее созданный репозиторий. В качестве ответа на задание отправьте ссылку на свой репозиторий и указание, какой именно файл содержит нужный сценарий.

### Submission status

|  |  |
| --- | --- |
| Attempt number | This is attempt 1. |
| Submission status | No attempt |
| Grading status | Not graded |
| Last modified | - |

## [x] Задание 10. Проверить, что открывается правильная страница товара

Сделайте сценарий, который проверяет, что при клике на товар открывается правильная страница товара в учебном приложении litecart.

Более точно, нужно открыть главную страницу, выбрать первый товар в блоке Campaigns и проверить следующее:

а) на главной странице и на странице товара совпадает текст названия товара  
б) на главной странице и на странице товара совпадают цены (обычная и акционная)  
в) обычная цена зачёркнутая и серая (можно считать, что "серый" цвет это такой, у которого в RGBa представлении одинаковые значения для каналов R, G и B)  
г) акционная жирная и красная (можно считать, что "красный" цвет это такой, у которого в RGBa представлении каналы G и B имеют нулевые значения)  
(цвета надо проверить на каждой странице независимо, при этом цвета на разных страницах могут не совпадать)  
д) акционная цена крупнее, чем обычная (это тоже надо проверить на каждой странице независимо)

Необходимо убедиться, что тесты работают в разных браузерах, желательно проверить во всех трёх ключевых браузерах (Chrome, Firefox, IE).

Можно оформить сценарии либо как тесты, либо как отдельные исполняемые файлы.

Если возникают проблемы с выбором локаторов для поиска элементов -- обращайтесь в чат за помощью.

-----

Уложите созданный файл, содержащий сценарий, в ранее созданный репозиторий. В качестве ответа на задание отправьте ссылку на свой репозиторий и указание, какой именно файл содержит нужный сценарий.

### Submission status

|  |  |
| --- | --- |
| Attempt number | This is attempt 1. |
| Submission status | No attempt |
| Grading status | Not graded |
| Last modified | - |