Институт открытых программ дополнительного образования ВШЭ

Python: Инструментальные средства для автоматизации и анализа данных. Часть 2.

Автор курса: Дмитрий Румянцев

Лекция 12

Сбор данных с Selenium

конспект

Москва 2023 г.

Краткая справка

Формы, использующиеся в HTML, создаются при помощи парного тега

<FORM></FORM>

Тег <FORM> имеет два основных аргумента:

action="script" – служит для указания скрипта, который должен обработать введенные в форме данные.

method="get/post" – устанавливает метод отправки данных формой.

Метод Get предполагает прикрепление введенных данных к адресной строке. Метод Post реализует скрытую отправку данных.

Интерфейсные элементы внутри тега <FORM> задаются следующими тегами:

<input>, атрибутом **type=""** которого можно задавать следующие элементы:

button – Кнопка.

checkbox – Флажки.

file – Поле для ввода имени файла.

hidden – Скрытое поле. Не отображается на веб-странице.

Image – Поле с изображением. Ведет себя как кнопка

password – Поле для ввода пароля.

radio – Переключатели. Выбор одной из нескольких альтернатив.

reset – Кнопка для восстановления значений формы по умолчанию.

submit – Кнопка для отправки данных формы на сервер.

text – Строка ввода данных с клавиатуры.

В HTML5 можно задать дополнительные значения атрибута type:

color – Виджет для выбора цвета.

date – Поле для выбора календарной даты.

datetime – Указание даты и времени.

datetime-local – Указание местной даты и времени.

email – Для адресов электронной почты.

number – Ввод чисел.

range – Ползунок для выбора чисел в указанном диапазоне.

search – Поле для поиска.

tel – Для телефонных номеров.

time – Для времени.

url – Для веб-адресов.

month – Выбор месяца.

week – Выбор недели.

Значение, введенное в элемент, передается через аргумент **value="<значение>"**. Каждый интерфейсный элемент имеет имя (которое можно уподобить имени переменной), которое задаётся атрибутом **name=""**.

<SELECT> – список, каждый элемент которого задаётся тегом <OPTION>.

Выбранный элемент списка маркируется атрибутом SELECTED.

XPATH

Сокращение от XML Path – Путь в стиле XML. Описание пути к любому элементу специальным метаязыком. Метаязык стоит из метсимволов. Используется в функциях поиска элементов в DOM.

Символ / (слеш) в начале пути указывает, что поиск должен начинаться от корневого узла.

Символ // (двойной слеш) указывает, что элемент может начинаться в любом месте, то есть иметь какую угодно вложенность.

Имена тегов указываются явно.

* Звёздочка указывает любой элемент (любой тег).

В квадратных скобках указывают атрибуты, которые должны присутствовать в искомом элементе.

@ символ, стоящий в начале имени атрибута.

Библиотека Selenium WevDriver

Официальная документация:

https://www.selenium.dev/documentation/webdriver/

Предназначена для моделирования поведения человека при работе с веб-браузерами. Функции библиотеки могут контактировать с интерфейсными элементами HTMK-документа (такими как строки ввода, кнопки и пр.).

WebDriver – объектно-ориентированный API, определяющий интерфейс для управления поведением веб-браузеров. Для работы с WebDriver могут понадобиться драйверы.

Страницы загрузки драйверов для браузеров:

Chrome

https://chromedriver.chromium.org/downloads

Opera

https://github.com/operasoftware/operachromiumdriver/releases

FireFox:

https://github.com/mozilla/geckodriver/releases

Microsoft Edge WebDriver

https://developer.microsoft.com/en-us/microsoft-edge/tools/webdriver/#downloads

Safari

https://developer.apple.com/documentation/webkit/about_webdriver_for_safari

Пакет

`selenium`

Библиотека

`WebDriver`

Функции, классы и методы

Драйверы браузера

```
webdriver.Chrome(options)
webdriver.Edge(options)
webdriver.Firefox(options)
webdriver.Safari(options)
```

Инициализируют клон браузера для дальнейшей работы с ним.

Options

Класс, входящий в состав драйвера браузера. Экземпляр этого класса используется для предварительных настроек драйвера браузера перед стартом.

driver.get(URL)

Загружает в браузер документ с адресом URL.

driver.find_element(by, values)

Поиск элемента в загруженном документе. Возвращает объект в виде первого встреченного элемента, удовлетворяющего запросу.

by может принимать следующие значения:

By.CLASS_NAME – поиск по имени класса.

By.CSS_SELECTOR – поиск по селектору.

By.ID – поиск по ID

By.NAME – поиск по имени элемента.

By.LINK_TEXT – поиск по тексту ссылки.

By. PARTIAL_LINK_TEXT – поиск по фрагменту текста ссылки.

By.TAG_NAME – поиск по имени тега.

By.XPATH – поиск по пути XPATH.

values – значение.

Может быть заменен функцией семейства find_element_by_XXX()

driver.find_elements(by, values)

Поиск всех элементов в загруженном документе. Возвращает все объекты, удовлетворяющие запросу. Параметры аналогичны driver.find_element()

element.send_keys(str)

Отправляет в элемент последовательность символов (включая управляющие клавиши). Применяется для автоматического ввода значений в строки ввода.

element.click()

Симулирует щелчок мышки по элементу (обычно по кнопке или ссылке).

driver.implicitly_wait(sec)

Инициирует неявное ожидание. Обычно используется в момент загрузки документа, чтобы дать возможность браузеру загрузить весь документ, прежде чем его элементы будут обработаны программой.

WebDriverWait(driver,timeout).until(process)

Включает ожидание окончания некоторого процесса (обычно загрузки нужного элемента). Timeout указывает время, которой необходимо ожидать окончания, прежде чем выполнить прерывание. Требует подключения:

from selenium.webdriver.support.ui import WebDriverWait