

Локаторы

Содержание

Локаторы.....	1
Цель изучения HTML и локаторов.....	2
HTML.....	2
Собственно материалы по локаторам.....	2
Вводные.....	2
Шпаргалки.....	4
Тренажеры для начинающих.....	4
Помимо вводных.....	5
Случаи посложнее.....	6
Браузерные инструменты.....	7
Сайты для практики.....	8
Также.....	9

Цель изучения HTML и локаторов

Чтобы, глядя на код сайта который хочется автоматизировать, понимать что именно перед нами — простой случай, "хороший" или сложный-плохой, и за что и как к элементам "цепляться".

HTML

Если знание этого языка уже есть, данную ссылку можно пропустить. Элементы, для которых мы ищем локаторы, ищутся на страницах написанных на языке разметки HTML. Знакомство с ним для работы в веб-автоматизации весьма желательно. Однако отдельно этот язык чаще всего автоматизаторам не преподают, вполне хватает любого учебника, лишь бы нравился.

Первый онлайн учебник из поиска который понравился мне — вот:
https://basicweb.ru/html/html_book.php

Но подойдет, ещё раз, любой — лишь бы пояснял "что есть что": какие элементы есть в HTML страницах, что они делают, как узнавать их в HTML-коде страниц.

Собственно материалы по локаторам

Вводные

Локаторы — "формулы" по которым мы в браузере — или Селениум или Селенид или Джаваскрипт) находим элементы на HTML-страницах.

Локатору желательно быть

1. Точным (чтобы он нашел именно тот самый элемент или первый из них если их несколько)
2. Понятным-наглядным (чтобы глядя на него мы понимали что и как там ищется).
3. Коротким (длинные локаторы могут местами не срабатывать или срабатывать нестабильно).

Бывают разные виды локаторов. Часто говорят о CSS selectors или XPath, но оба этих способа написания локаторов могут обращаться по тегу, атрибутам (популярны атрибуты id, name, class) или специально создаваемые (data-qa , data-testid), а XPath "если других выходов нет" или они менее удобны, ищет ещё и по тексту. Более подробно будет по ссылкам ниже.

<https://www.youtube.com/watch?v=TNh2ydpOow>

Алексей Баранцев: Десять правил построения хороших локаторов
-- видео агитирует за более удобные (короткие, более ясные) CSS селекторы. Для начинающих и тренажёров такие локаторы — то с чего стоит начать.

Можно ли найти чего-то вводного и по Xpath? Да.

https://www.youtube.com/watch?v=EVGH_XxAbXQ

Xpath - локаторы. Уроки для начинающих

Но этим можно не ограничиваться — лишь бы получалось в браузере писать нужные локаторы, начиная с простых.

Шпаргалки

https://www.w3schools.com/cssref/css_selectors.php

W3C CSS selectors: хорошая шпаргалка (пусть на английском) про возможности оных CSS selectors

<https://devhints.io/xpath>

-- хороший пример шпаргалки и по CSS selectors и по XPath

Шпаргалок есть, конечно же, больше — но эти из самых популярных.

Тренажеры для начинающих

<https://flukeout.github.io/>

CSS diner — игровой тренажер по CSS Selectors. Введи правильный селектор, нажми Enter, и если выполнил условие — проходишь на следующий уровень. Уровни-задачи описаны на несложном английском и сопровождаются толстыми подсказками на английском же (можно переводить каким-то переводчиком). Вышеупомянутая шпаргалка от W3C тоже может помочь.

<https://topswagcode.com/xpath/>

XPATH diner — совершенно аналогичный тренажер для XPath

Советы

<https://testerslittlehelper.wordpress.com/2016/07/10/real-xpath/>

“Не так страшен XPath как его незнание”

Статья не для начинающих, но говорит о полезных принципах, которые будут понятны тем кто понял как начать).

1. Никогда не используй плагины или копирование xpath из кода страницы средствами браузера или веб-разработчика. Вот например как показывает одну ссылку плагин к Файрфокс: `//header/div/ul/li[2]/a` . Разве из этой ссылки понятно, о каком элементе речь, что мы ищем? Ведь порой бывает, что взглянув на локатор в коде или в тексте исключения мы должны понять о каком элементе речь. Как это можно понять из такой строки? Я уже не говорю о том, что любой код, основанный на таких локаторах упадет при любом дуновении ветерка. Каждый раз, когда ты пишешь локатор подобный `//div[1]/div[2]/ul/li` (продолжать можно долго) в мире умирает что-то хорошее!!! Это, если хотите, говнокод тестировщика, который нужно выжигать каленым железом.

2. Старайся написать xpath как можно короче и понятнее, используй его возможности и схожест с языком программирования, чтобы и через месяц ты сам мог понять о каком элементе речь и что нужно поправить в случае изменения верстки

3. Xpath’у время и место! Если есть возможность использовать `id`, `name` или попросить разработчиков внести в код `id` то сделай это!

4. Вместо длинной цепочки слешей, как указано выше, используй отношения элементов: предок, потомок, сестринский элемент

5. Можно и нужно использовать логические операции `and`, `not` , `or`

6. Нормальный xpath всегда начинается с `//` и не использует фильтры с номером элемента в стиле `[2]` (например `//div[2]`)

Случаи посложнее

О них тут коротко, "продолжение как-нибудь следует".

XPath по тексту: `//tag[contains(text(),"myText")]` , часто применяется если способа удобнее нет.

Теги `frame` / `iframe` , внутри которых сидит нужный элемент – "в браузере находит, но не находит Селениум / Селенид / Селен". На фреймы в коде надо переключаться, для чего и в Селениуме и в Селениде есть отдельные инструкции.

"Лазанье по дереву" в XPath, от некоего элемента сначала поднимаемся через `../` (наподобие как по дереву папок в операционке), а потом спускаемся.

Shadow DOM: когда элементы в неких особых местах и тегах. "Признак" этого — наличие в HTML-коде чего-то вроде `#shadow root` Встречался с вопросами по такому в чатах, но в работе не видел. Ещё надо будет вложить время в поиск и разобраться.

Браузерные инструменты

Основная работа по поиску локаторов происходит в браузерных инструментах. Сейчас это чаще всего Google Chrome и Developer Tools. Поэтому две ссылки на полезные видео по ним.

<https://www.youtube.com/watch?v=a3vRm9neKhs>

Lesson 14. Chrome Devtools (Артем Русов), 25 минут.

<https://www.youtube.com/watch?v=dytFJdIgwRk>

"Карты, деньги, консоль браузера для QA" (доклад, 42 минуты).

Смотреть их не прямо обязательно, -- но может быть полезно "если не уверены".

Сайты для практики

<https://demoqa.com/>

-- куча страниц-тренажеров по разным аспектам (веб-формы, фреймы)

<https://the-internet.herokuapp.com/>

-- тоже куча страниц-тренажеров по каким-то отдельным аспектам (логин, фреймы, динамические страницы...).

Вообще сайтов намного больше, но какие-то устаревают, какие-то появляются, какие-то перегружаются так что информация может стать неактуальной.

Если вы хотите автоматизировать сайт с достаточно серьёзными фидами но без представительности в топах, то

- можно задействовать [общий поиск automation practice sites](#).
- можно поискать сайты региональных компаний, небольшие магазинчики и пр.

Один из сайтов-тренажеров из поиска выше:

<https://magento.softwaretestingboard.com/>

<https://www.demoblaze.com/>

— Blaze Demo (демо-сайт компании Blazemeter предлагающей услуги тестирования перфоманса)

Octoperf Petstore — аналогично, только от Octoperf

<https://petstore.octoperf.com/>

<https://www.automationexercise.com/> — сайт для практики и даже с описанием готовых возможных кейсов.

Можно взять открытый Яндексовский дипломный сайт:

<https://stellarburgers.nomoreparties.site/>

-- но он ориентирован на "грязный" код, где локаторы выдумывать сложнее (например, цепляться за текст).

Также

<https://www.w3schools.com/html/default.asp>

-- курс по HTML на английском.