

Введение в автоматизацию тестирования

Основы работы с Selenium IDE

Введение в автоматизацию

Автоматизация тестирования (test automation) - набор техник, подходов и инструментальных средств, позволяющий исключить человека из выполнения **НЕКОТОРЫХ** задач в процессе тестирования.

Инструментальное средство автоматизированного тестирования (test automation tool) - программа (или набор программ), позволяющая создавать, редактировать, отлаживать и выполнять автоматизированные тесты, а также **собирать статистику** их выполнения.

Технология Record & Playback

Одной из наиболее **распространённых и простых** для понимания технологий автоматизации тестирования является технология **Record & Playback** («Записать и воспроизвести»).

Суть её заключается в том, что средство автоматизации тестирования позволяет **выполнить с тестируемым приложением некоторый набор действий, которые будут записаны на специальном языке программирования, а затем могут быть воспроизведены.**

Технология Record & Playback

Преимущества:

- создание «скелета» теста ускоряется;
- средство автоматизации само собирает техническую информацию о приложении;
- это просто для понимания новичками.

Недостатки:

- записанные тесты содержат т.н. «hard-coded» («жёстко закодированные») значения, которые приходится потом заменять вручную.
- средство автоматизации записывает ВСЁ, в т.ч. много-много лишнего;
- если приложение достаточно сильно изменилась, тест придётся перезаписывать.

Вывод: технология R&P хороша в качестве помощника, но она не выполнит за человека ВСЕ необходимые для автоматизации тестирования действия.

Data-Driven и Keyword-Driven testing тестирование

В автоматизации тестирования существует проблема создания достаточно универсальных и используемых повторно тестов.

Решить эти задачи помогают два подхода:

- **тестирование под управлением данными (Data-Driven testing)** - вынесение данных теста из самого теста;

и

- **тестирование под управлением ключевыми словами (Keyword-Driven testing)** - вынесение логики теста из самого теста.

Что такое Selenium

Selenium – это инструмент для автоматизированного управления браузерами.

Selenium – это драйвер браузера.

Драйвер — компьютерное программное обеспечение, с помощью которого другое программное обеспечение получает доступ к аппаратному обеспечению некоторого устройства.

Официальный сайт Selenium:

<http://www.seleniumhq.org/>

Русскоязычный сайт про Selenium:

<http://selenium2.ru/>



Selenium – позволяет другим программам взаимодействовать с браузером.

Чаще всего этими программами являются – тесты.

Selenium – не содержит каких-то специальных средств для тестирования, он позволяет только управлять браузером.



Что должен уметь делать драйвер для браузера:

- Запускать браузер;
- Открывать URL;
- Находить ссылки;
- Выполнять действия;
- Останавливать браузер.

В Selenium реализованы интерфейсы для языков программирования:

- Java;
- Python;
- Ruby;
- JavaScript;
- C#.



Инструментальное средство Selenium IDE



<https://www.selenium.dev/downloads/>

Selenium IDE (- инструментальная среда для разработки и выполнения скриптов, представленная в виде плагина для браузера **Firefox**).

Свежая версия всегда доступна по адресу:
<https://www.selenium.dev/downloads/>

Selenium IDE

Selenium IDE is a Chrome and Firefox plugin which records and plays back user interactions with the browser. Use this to either create simple scripts or assist in exploratory testing.

Download latest released version [for Chrome](#) or for [Firefox](#) or view the [Release Notes](#).

Download previous [IDE versions here](#).

Selenium IDE поддерживает

Операционные системы:

Windows,



Linux,



OS X,



Solaris



Языки программирования:

Java,



C#,



Selenium IDE
Python,



Ruby,



JavaScript



Браузеры:

Firefox,



IE ,



Safari ,



Opera,

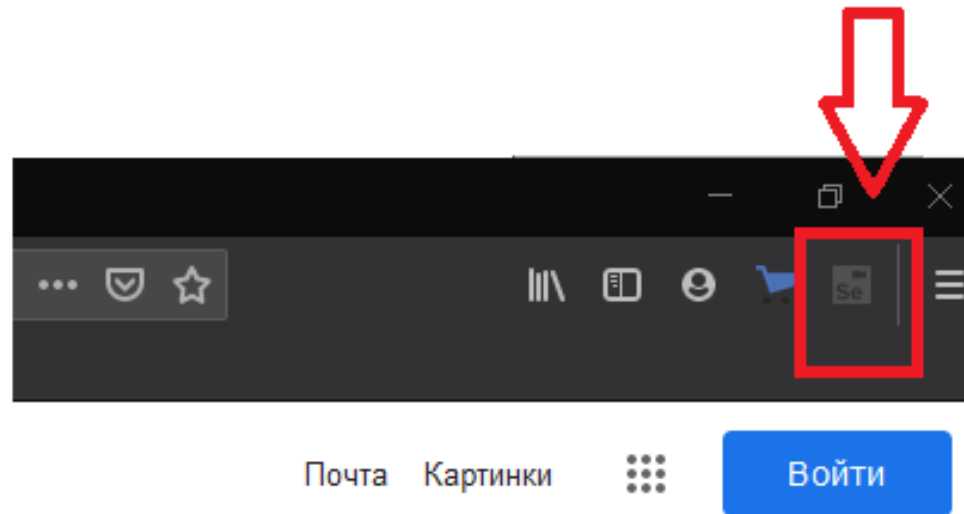


Chrome



Запуск Selenium IDE

После установки плагина Selenium IDE доступен в браузере Firefox



Ключевые поля Selenium IDE

Поле «**Command**» содержит команду (что необходимо сделать)

Поле «**Target**» указывает целевой элемент (с чем это сделать)

Поле «**Value**» содержит значение, с которым выполняется некоторая операция.



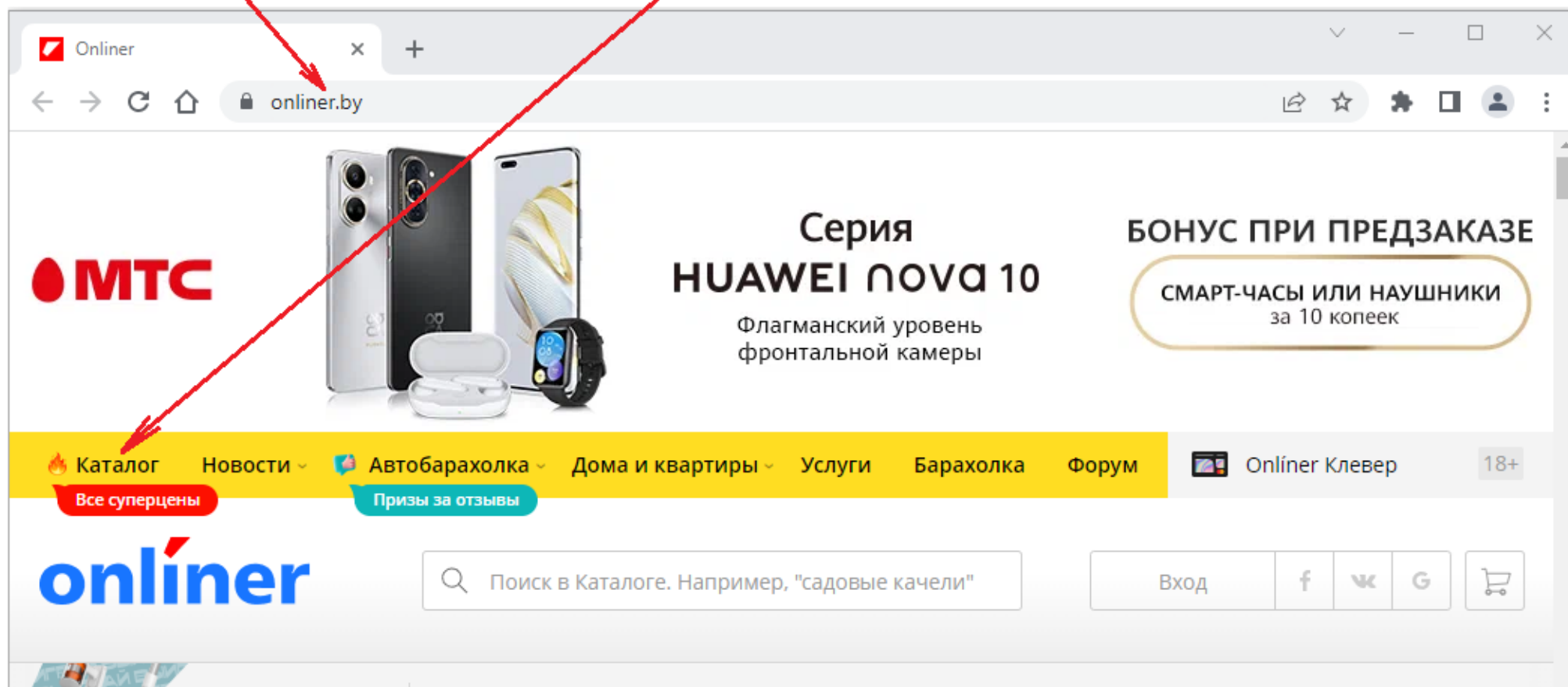
The image shows a screenshot of the Selenium IDE interface. It features three main input fields: 'Command', 'Target', and 'Value'. Each field is highlighted with a red underline. The 'Command' field has a dropdown arrow and a small icon to its right. The 'Target' field has a small icon to its right. The 'Value' field is a simple text input. The background of the slide shows a faint image of a modern building with large windows.

Пример

Открыть

<https://www.onliner.by>

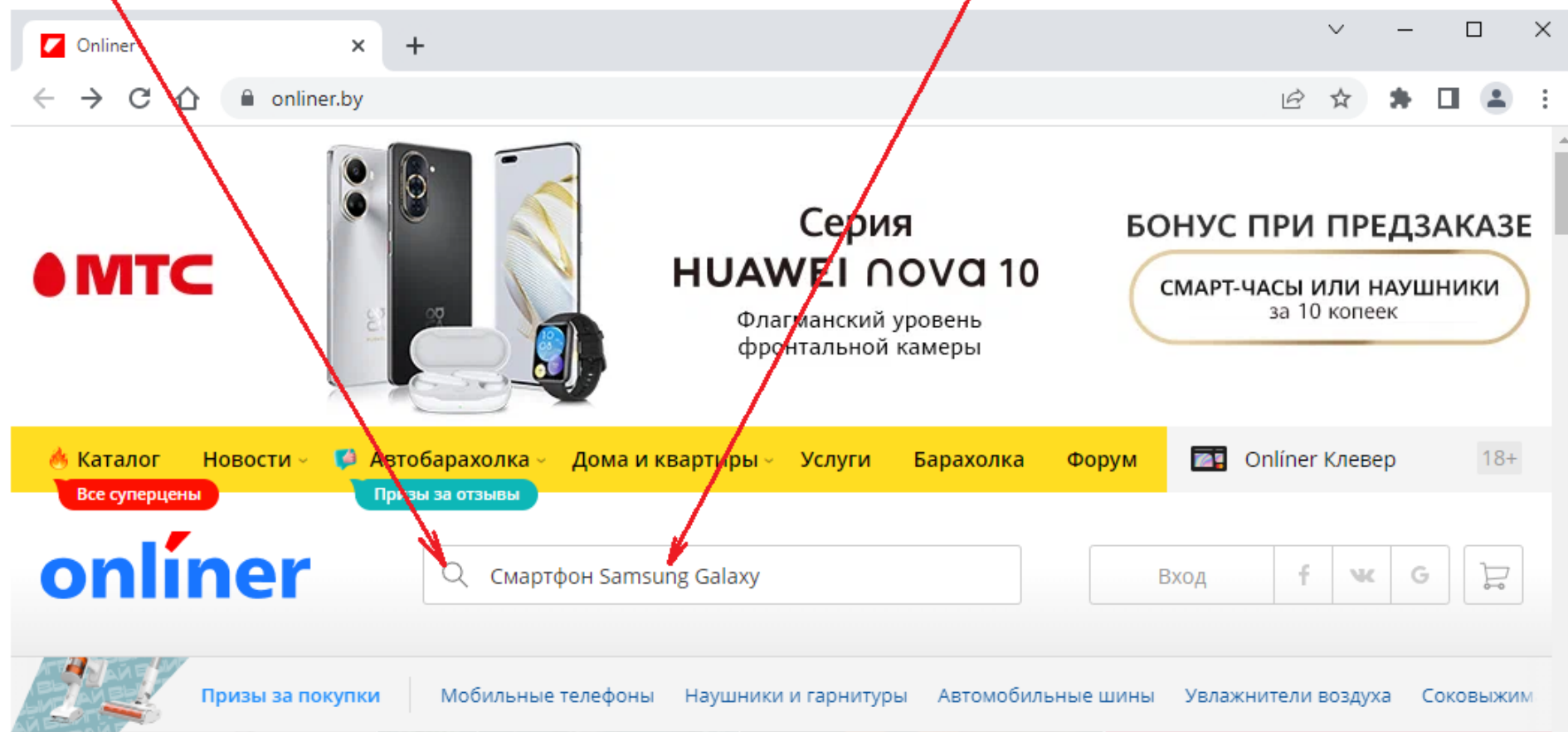
Кликнуть "Каталог"



Пример

Кликнуть "Поиск"

Ввести "Смартфон Samsung Galaxy"



Пример

Selenium IDE - ццццц

Проект:

Тесты +

Поисковые тесты...

https://www.onliner.by

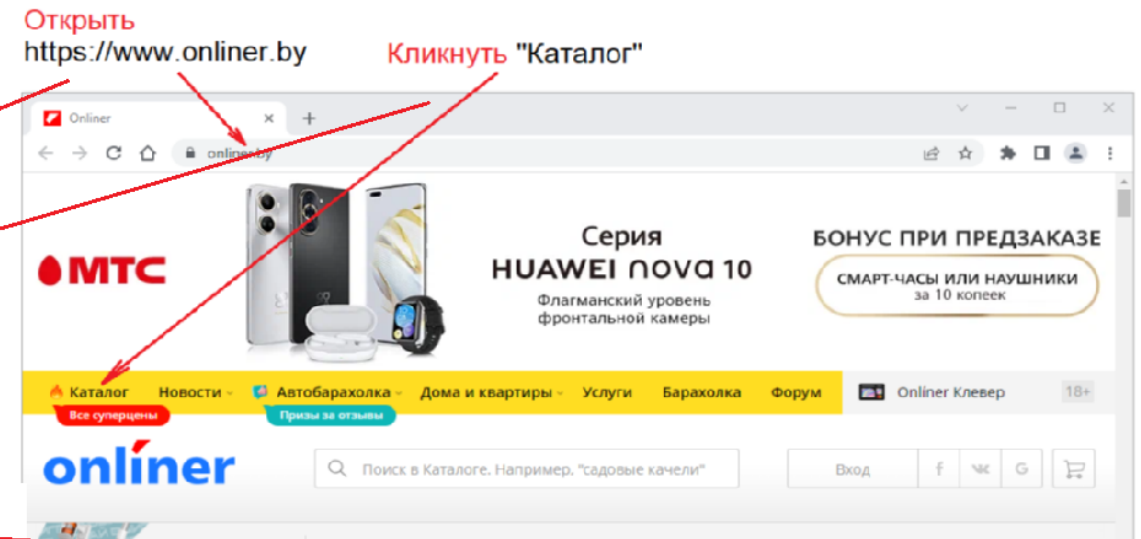
	Команда	Цель	Ценность
1	открытым	https://www.onliner.by/	
2	нажмите	LinkText=Kamalog	
3	тип	name=str	Смартфон Samsung Galaxy
4	нажмите	css=search__input	
5			
6			

Команда

Цель

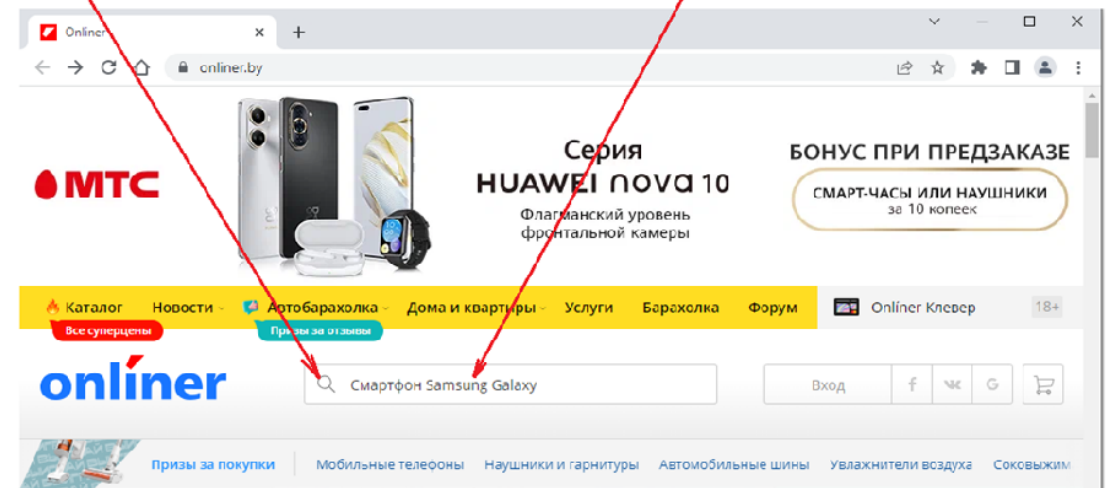
Ценность

Описание



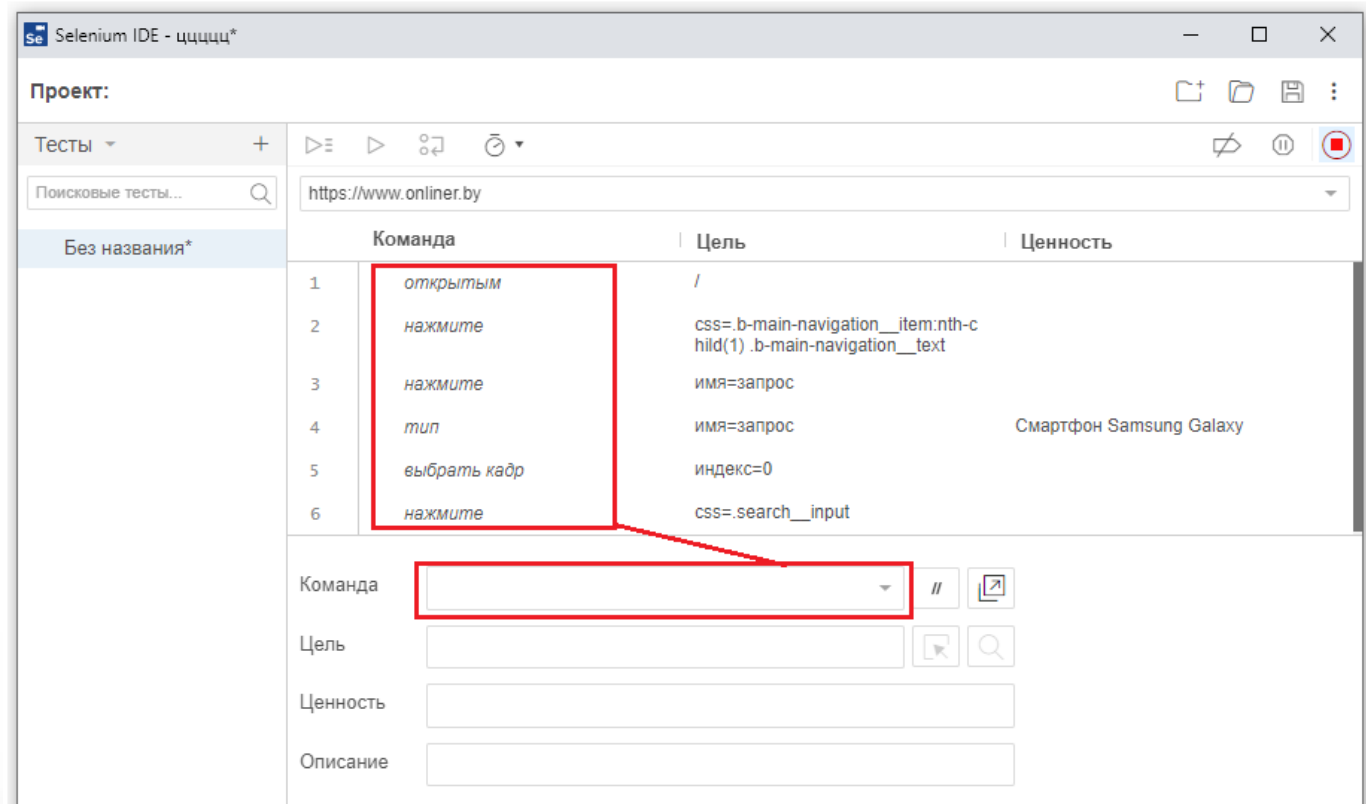
Кликнуть "Поиск"

Ввести "Смартфон Samsung Galaxy"



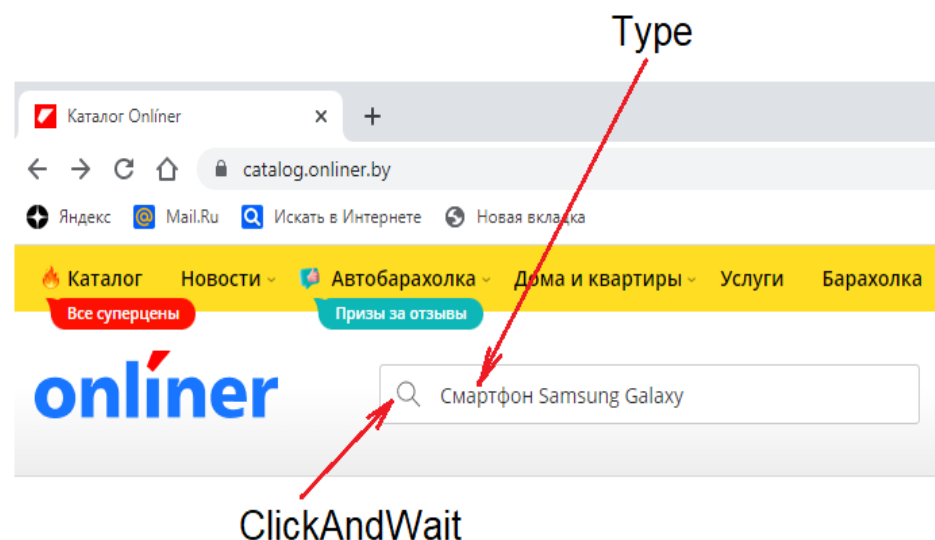
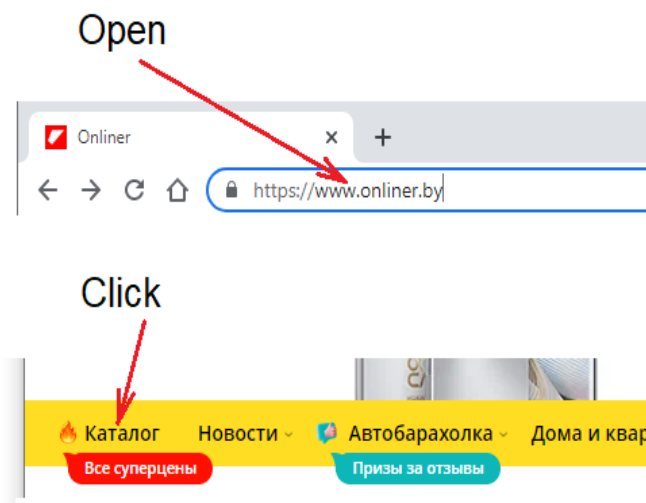
Ключевые поля: Command

Поле «**Command**» содержит указание того, что необходимо выполнить на данном шаге теста.



Ключевые поля: Command, действия

Примеры **действий** (actions):



Ошибка (невозможность) выполнения любой из этих команд приводит к остановке теста!

Ключевые поля: Command, проверки

Проверки используются для **анализа состояния** веб-ориентированного приложения.

Например, можно проверить **наличие** того или иного **элемента** страницы, **значение** того или иного **поля** и т.п.

Command	<input type="text" value="type"/>	<input type="button" value="//"/>	<input type="button" value="🔗"/>
Target	<input type="text" value="name=str"/>	<input type="button" value="📌"/>	<input type="button" value="🔍"/>
Value	<input type="text" value="Смартфон Samsung Galaxy"/>		

Ключевые поля: Command, проверки

Команды, выполняющие проверку, делятся на два больших класса.

- Начинающиеся с `assert`;
- Начинающиеся с `verify`.

Ошибка `assert` =
тест
останавливается.

Ошибка `verify` =
протоколируется факт
наличия проблемы и
тест продолжается.

Два микропримера

Table	Source	
Command	Target	Value
open	/	
click	css=li.topb...	
clickAndWait	link=Новос...	
type	name=str	ISSOFT
clickAndWait	css=input.b...	
assertValue	str	abcde
clickAndWait	link=Мисс ...	

Command: open
Target: /
Value:

Log Reference

[info] Executing: |clickAndWait | css=input.button.big | |
[info] Executing: |assertValue | str | abcde |
[error] Actual value 'ISSOFT' did not match 'abcde'

Table	Source	
Command	Target	Value
open	/	
click	css=li.topb...	
clickAndWait	link=Новос...	
type	name=str	ISSOFT
clickAndWait	css=input.b...	
verifyValue	str	abcde
clickAndWait	link=Мисс ...	

Command: open
Target: /
Value:

Log Reference

[info] Executing: |verifyValue | str | abcde |
[error] Actual value 'ISSOFT' did not match 'abcde'
[info] Executing: |clickAndWait | link=Мисс ИТ-2012 стала Марина Горбачевич | |

Ключевые поля: Command, ожидание

Существует еще один класс команд, (начинающихся со слов **waitFor** или заканчивающихся словами **andWait**).

Эти команды предназначены для работы с элементами, появление или изменение состояния которых **требует некоторого времени**.

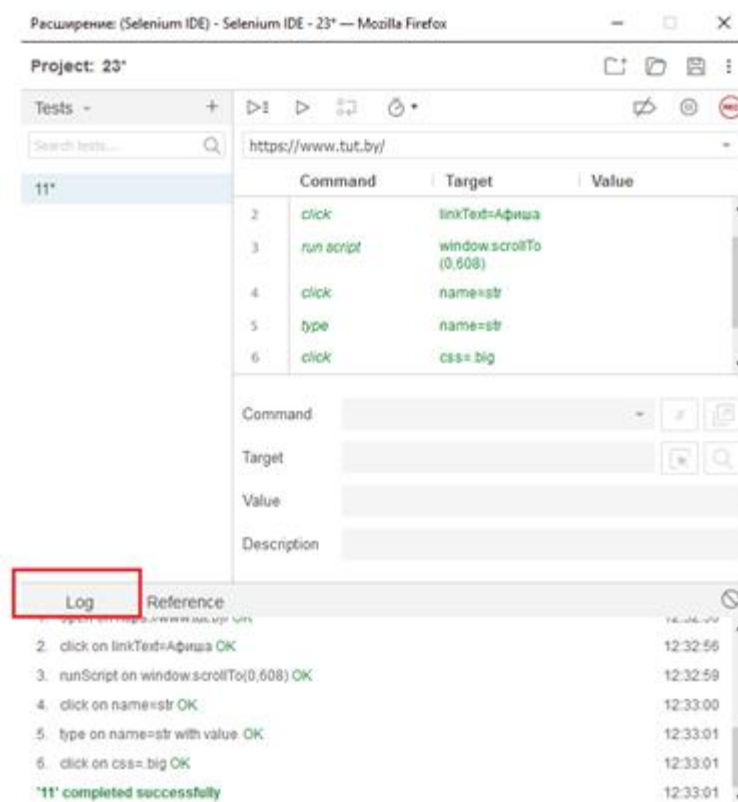
Ошибка (невозможность) выполнения такой команды приводит к остановке теста!

type	name=str	ISSUE 1
clickAndWait	css=input.b...	
clickAndWait	link=Мисс ...	

Command	<input type="text" value="clickAndWait"/>
Target	<input type="text" value="css=input.button."/> ▼
Value	<input type="text"/>

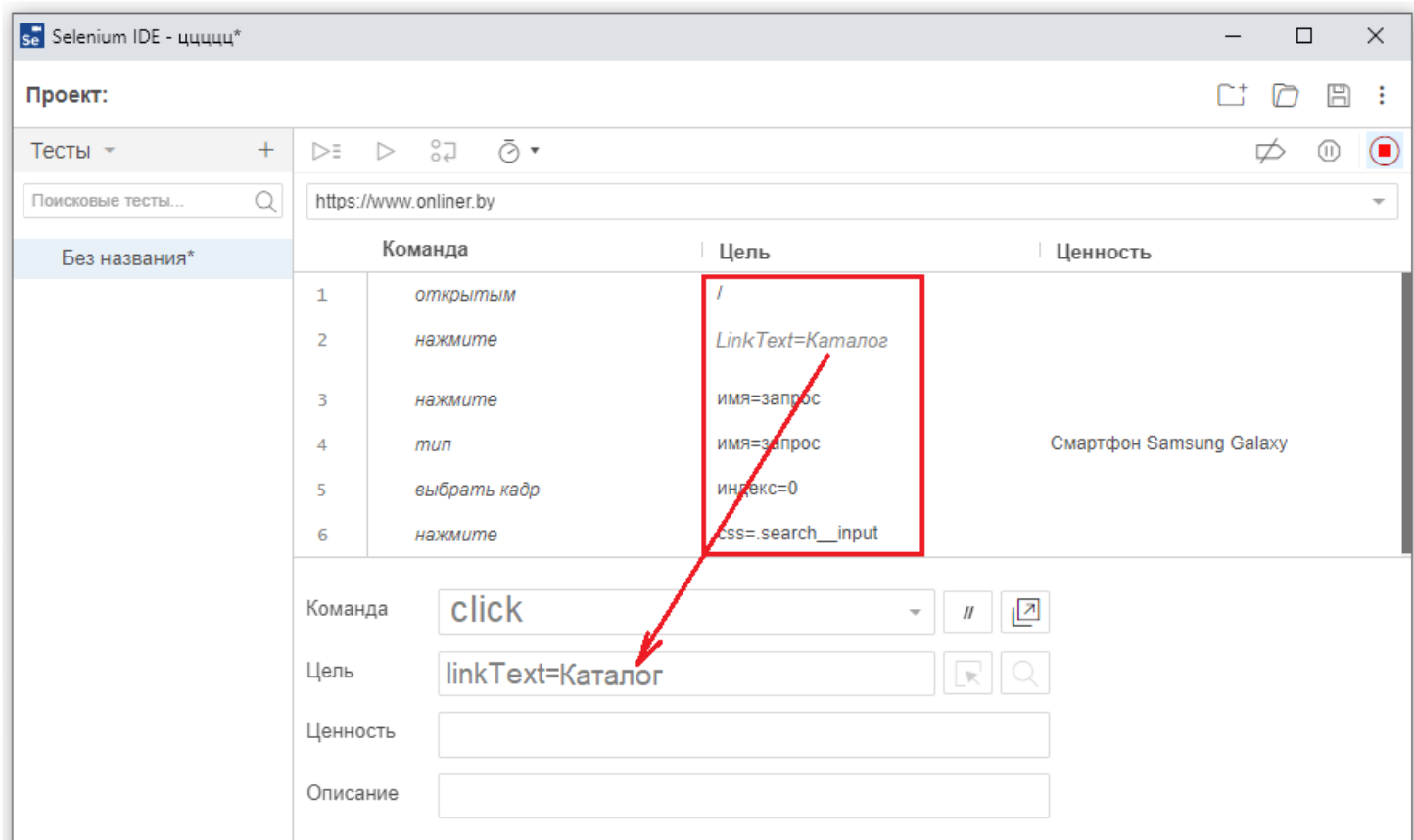
Ключевые поля: Log

За ходом и результатом выполнения тестов можно следить с помощью поля **log**, в котором отражаются все выполняемые Selenium IDE действия.



Ключевые поля: Target

Поле **target** указывает, с каким элементом следует выполнять действие.



Ключевые поля: Target

Указание на элемент, с которым необходимо выполнять действие, производится при помощи т.н. «локаторов» (locator).

Формат локатора таков:

LocatorType = Argument



Локаторы бывают следующих типов:

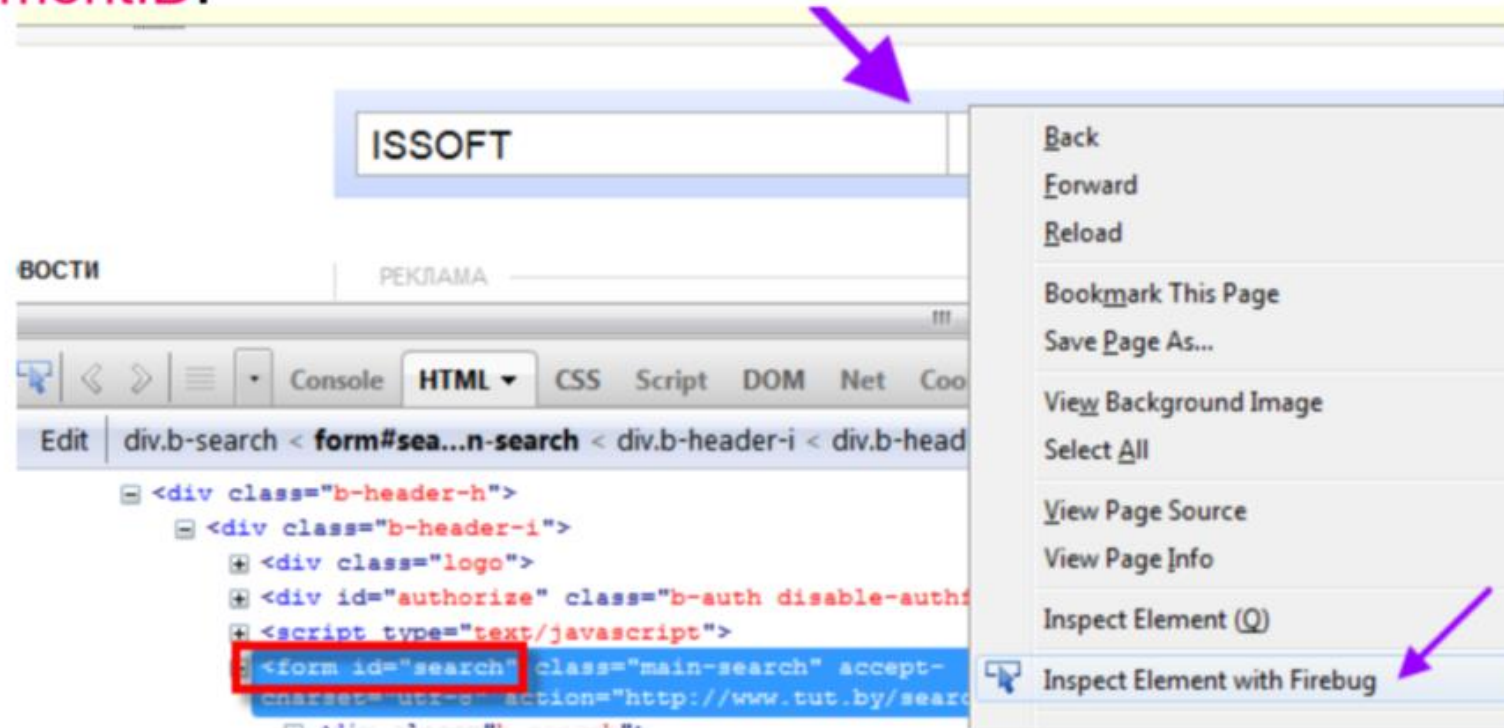
- `id = ElementID`
- `name = ElementName`
- `link = LinkText`
- `xpath = XPath`



Локатор вида id = ElementID

можно (хоть и НЕЖЕЛАТЕЛЬНО) записывать просто в виде ElementID.

Идентификатор элемента нужно смотреть в HTML-коде страницы.

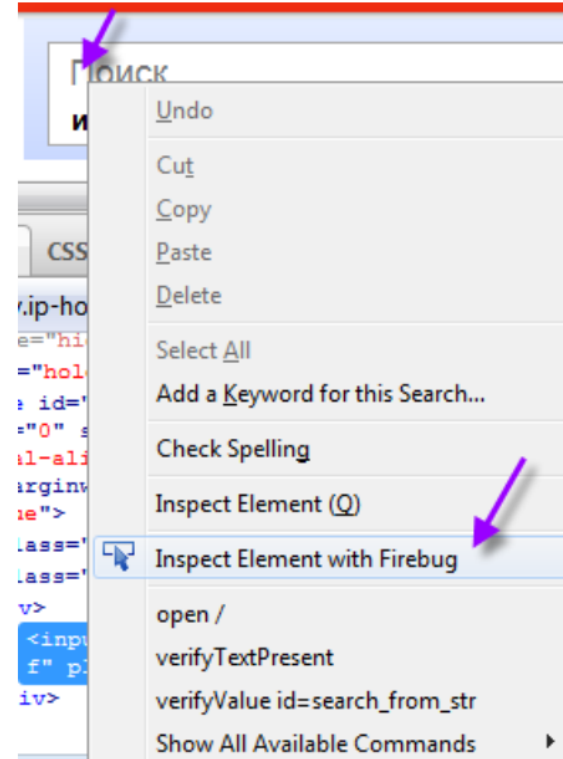


Локатор вида name = ElementName

тоже можно (хоть и ТОЖЕ НЕЖЕЛАТЕЛЬНО)
записывать просто в виде
ElementName.

Имя элемента тоже нужно
смотреть в HTML-коде
страницы.

```
class="ip-holder">  
<div>  
  <input id="search_from_str" class="ip" type="text" autocomplete="off"  
    f" placeholder="Поиск" value="" name="str">  
</div>
```

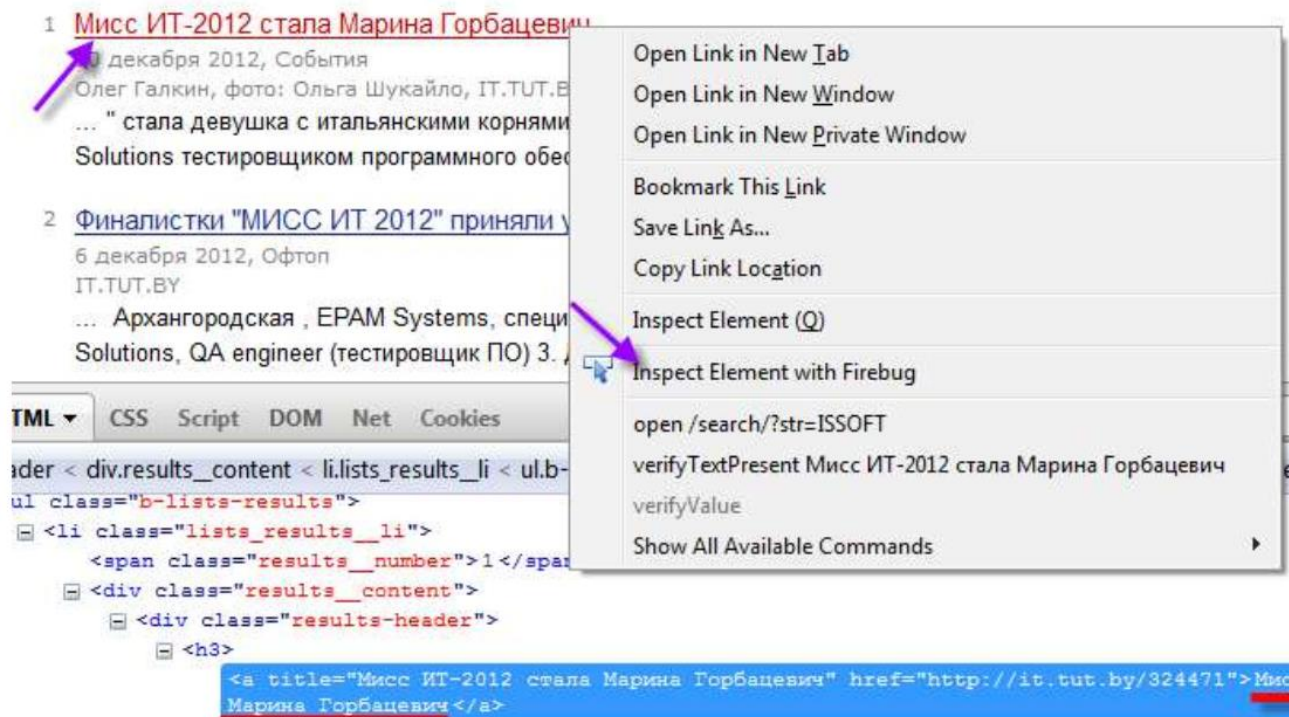


```
<input id="search_from_str" class="ip" type="text" autocomplete="of  
f" placeholder="Поиск" value="" name="str">
```


Локатор вида link = LinkText

НЕЛЬЗЯ записывать
просто в виде LinkText.

Текст ссылки чаще всего
виден «невооружённым
глазом», но если посмотреть
его в коде – хуже не будет.



Напоминание по локаторам

Лучше **НЕ** использовать сокращенную форму записи локаторов (**это может привести к неоднозначности определения элемента**), но если очень хочется...

Можно сокращать локаторы типа `id` или `name`, но **НЕЛЬЗЯ** сокращать локаторы типа `link`.

Command	<input type="text"/>	↔	Command	<input type="text"/>
Target	<input type="text" value="id=search"/>		Target	<input type="text" value="search"/>
Value	<input type="text"/>		Value	<input type="text"/>

Command	<input type="text"/>	↔	Command	<input type="text"/>
Target	<input type="text" value="name=str"/>		Target	<input type="text" value="str"/>
Value	<input type="text"/>		Value	<input type="text"/>

Target	<input type="text" value="link=Новости IT"/>	<input type="button" value="Find"/>
Value	<input type="text"/>	

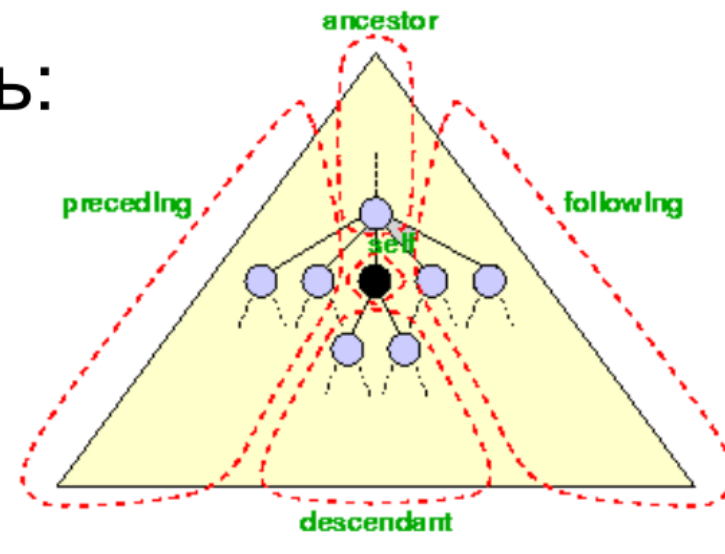
Локатор вида xpath = XPath

является **самым универсальным**, но требует понимания XML и непосредственно XPath.

Почитать можно обо всём этом здесь:

<http://www.w3schools.com/xpath/>

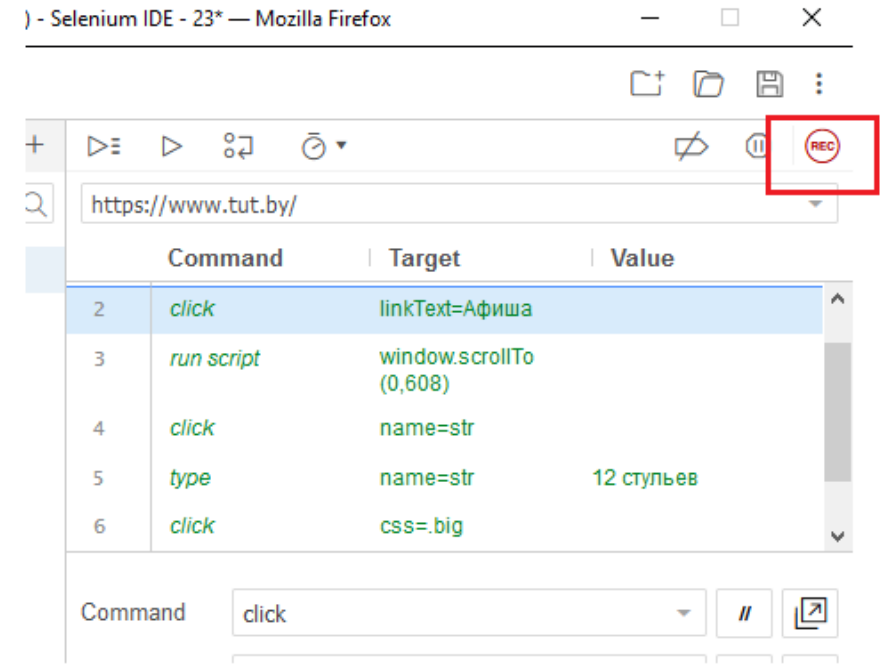
<http://www.w3.org/TR/xpath/>



Запись и воспроизведение теста

Сразу же после запуска Selenium IDE находится в режиме записи теста, выключать и повторно включать который можно кнопкой в правом верхнем углу окна Selenium IDE.

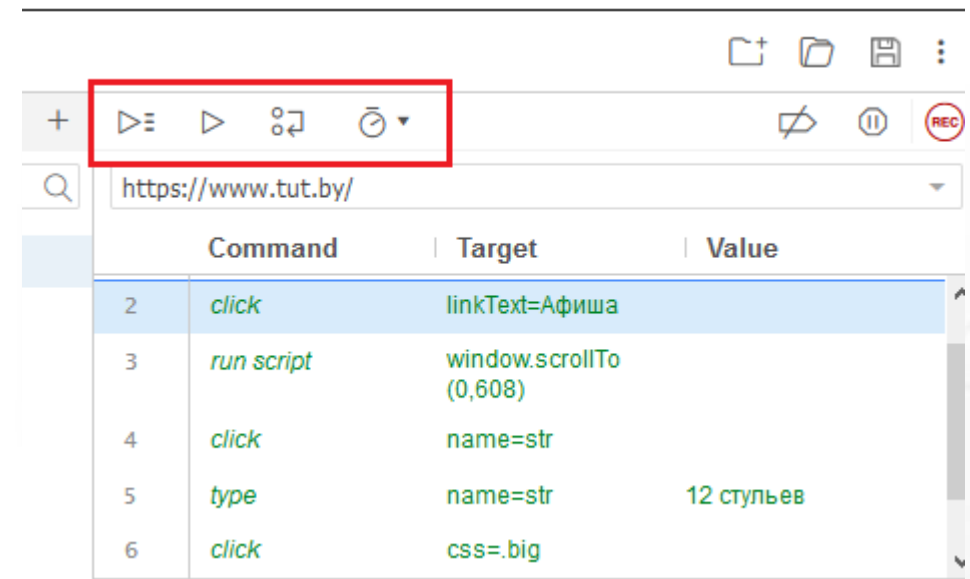
Запись имеет смысл приостанавливать, если вы выполняете действия, не относящиеся к тесту.



Запись и воспроизведение теста

Для выполнения записанного теста есть несколько элементов в левом верхнем углу экрана:

- Кнопка выполнения всех тестов;
- Кнопка выполнения текущего теста;
- Кнопка пошагового выполнения тестов;
- Регулятор скорости выполнения.



Запись и воспроизведение теста

В процессе воспроизведения теста Selenium IDE в реальном времени показывает, успешно ли выполнена та или иная команда:

- Подсветкой команды зеленым или розовым в списке команд;
- Сообщениями в логе.

The screenshot displays the Selenium IDE interface. At the top, the URL bar shows 'https://www.tut.by/'. Below it is a table of commands:

	Command	Target	Value
2	click	linkText=Афиша	
3	run script	window.scrollTo	(0,608)
4	click	name=str	
5	type	name=str	12 ступьев
6	click	css=.big	

Below the table, the 'Command' field is set to 'click' and the 'Target' field is set to 'linkText=Афиша'. A red arrow points to the 'css=.big' target in the table.

At the bottom, the 'Log' section shows the following entries:

- 1. open on https://www.tut.by/ OK
- 2. click on linkText=Афиша OK
- 3. runScript on window.scrollTo(0,608) OK
- 4. click on name=str OK
- 5. type on name=str with value 12 ступьев OK
- 6. click on css=.big OK
- '11' completed successfully

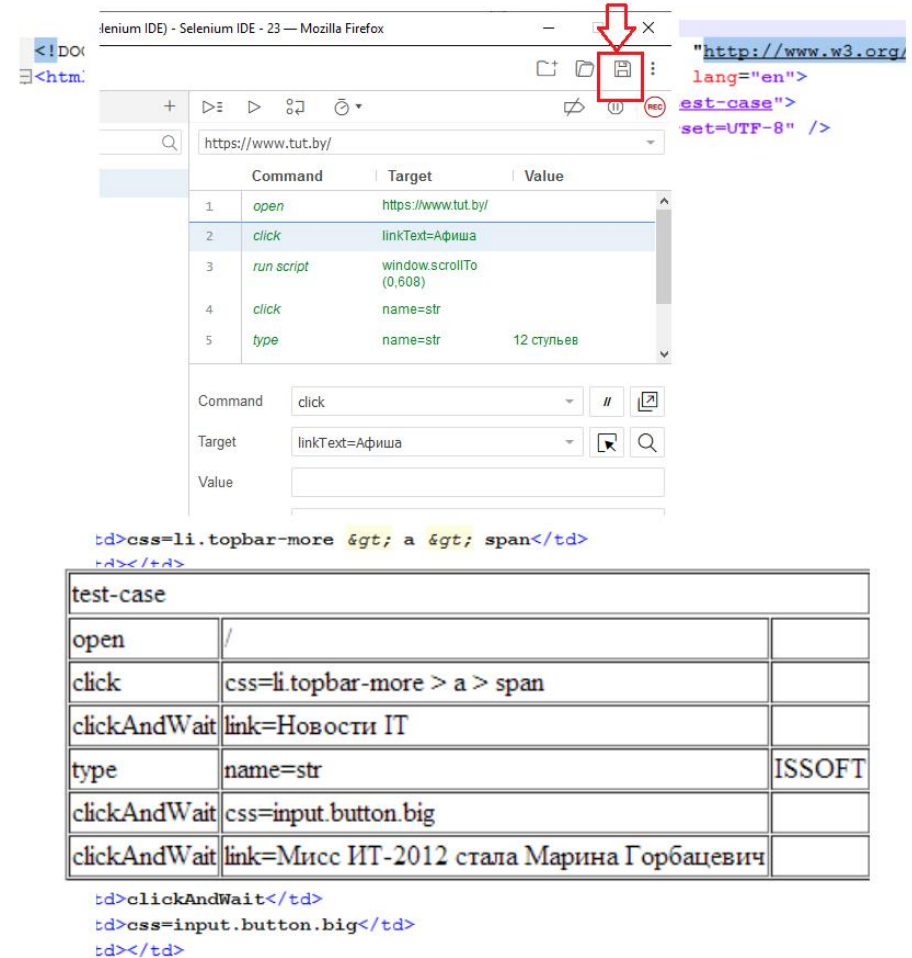
A red arrow points to the final log entry, indicating successful completion.

Сохранение теста

Тесты с которыми в
последствии можно будет
продолжить работу в IDE,
сохраняются в виде обычной
HTML-страницы с определенной
структурой.

Для ускорения просмотра их
можно открывать в браузере как
обычные страницы.

Несколько тестов (сценариев)
можно сохранять в виде «тест-
сюита» (test-suite).



The screenshot displays the Selenium IDE interface within a Mozilla Firefox browser window. The main panel shows a test suite with five commands:

Command	Target	Value
1 open	https://www.tut.by/	
2 click	linkText=Афиша	
3 run script	window.scrollTo	(0,608)
4 click	name=str	
5 type	name=str	12 ступень

Below the command list, the 'click' command is selected, showing its configuration: Command: click, Target: linkText=Афиша, Value: (empty).

At the bottom, the HTML export of the test suite is shown, including the XML structure and the generated HTML code:

```
test-suite
open /
click css=li.topbar-more > a > span
clickAndWait link=Новости IT
type name=str ISSOFT
clickAndWait css=input.button.big
clickAndWait link=Мисс ИТ-2012 стала Марина Горбачевич
clickAndWait
css=input.button.big

```

Благодарю за внимание