
QA
מע

אלכס גורבצ'וב
АЛЕКСЕЙ ГОРБАЧЁВ



HTML

HTML



Введение в HTML

HTML (HyperText Markup Language) - это язык разметки, используемый для создания структуры веб-страниц. Он состоит из элементов, которые обозначают различные части контента. Элементы обычно состоят из открывающего и закрывающего тегов. Например:

```
<p>Это абзац текста.</p>
```

```
<h1>Заголовок первого уровня</h1>
```

```
<h2>Заголовок второго уровня</h2>
```

HTML



Введение в HTML

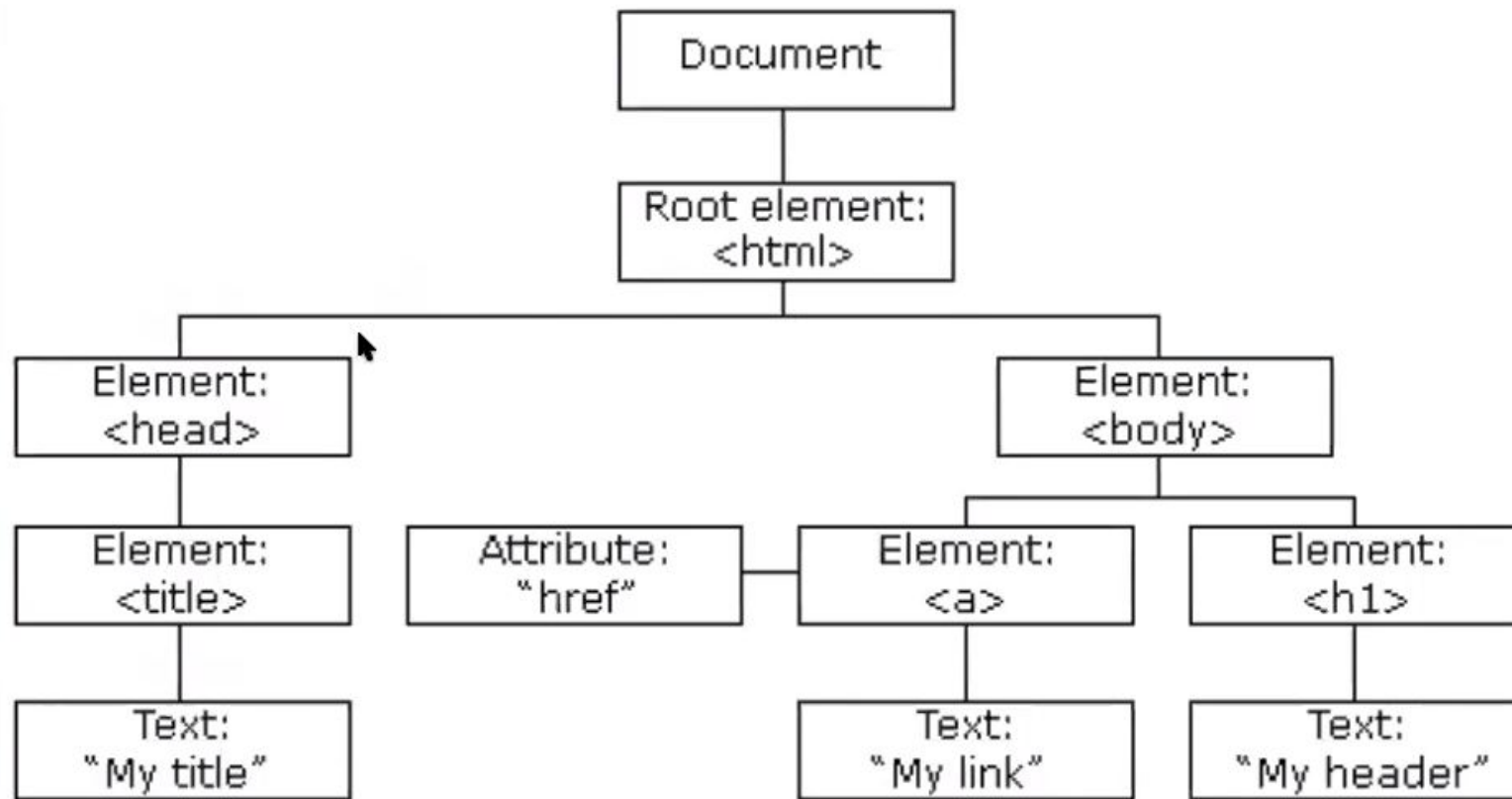
Основная структура HTML:

```
<!DOCTYPE html>
<html>
<head>
  <title>Моя первая веб-страница</title>
</head>
<body>
  <h1>Привет, мир!</h1>
  <p>Это абзац текста.</p>
</body>
</html>
```

Selenium



Web Page



HTML



Введение в HTML

Основная структура HTML:

```
<!DOCTYPE html>
<html>
<head>
  <title>Моя первая веб-страница</title>
</head>
<body>
  <h1>Привет, мир!</h1>
  <p>Это абзац текста.</p>
</body>
</html>
```

Введение в HTML

HTML-документ состоит из различных элементов, таких как заголовки, абзацы, списки, ссылки и многое другое:

```
<h2>Заголовок второго уровня</h2>
```

```
<p>Это другой абзац текста.</p>
```

```
<ul>
```

```
<li>Первый элемент списка</li>
```

```
<li>Второй элемент списка</li>
```

```
</ul>
```

```
<a href="https://www.googlecom">Ссылка на google.com</a>
```

CSS

CSS





Введение в CSS

CSS (Cascading Style Sheets) - это язык стилей, который используется для оформления внешнего вида веб-страниц. Он позволяет изменять цвета, шрифты, расположение элементов и другие аспекты дизайна. Например:

```
p {  
  color: blue;  
  font-size: 16px;  
}
```

```
body {  
  background-color: #f2f2f2;  
}
```

```
h1 {  
  color: green;  
}
```

Введение в CSS

CSS использует понятия "class" и "id" для применения стилей к элементам на веб-странице. Эти понятия позволяют точно управлять стилями и внешним видом элементов, что делает код более гибким и структурированным.

➤ Class (класс):

- Классы - это многократно используемые идентификаторы, которые могут применяться к нескольким элементам на странице.
- Один элемент может иметь несколько классов, разделенных пробелами.
- Для определения класса используется селектор "." перед именем класса.
- Классы полезны, когда вы хотите применить один и тот же стиль к нескольким элементам.

```
.highlight {  
  background-color: yellow;  
  font-weight: bold;  
}
```

```
<p class="highlight">Этот текст выделен классом</p>
```

Введение в CSS

➤ ID (идентификатор):

- Идентификаторы - это уникальные идентификаторы для одного элемента на странице.
- Каждый идентификатор может использоваться только один раз на странице.
- Для определения идентификатора используется селектор "#" перед именем идентификатора.
- Идентификаторы полезны, когда вы хотите применить уникальный стиль или функциональность к конкретному элементу.

```
#header {  
  background-color: #333;  
  color: white;  
  text-align: center;  
}
```

```
<header id="header">Это верхний колонтитул</header>
```

JAVASCRIPT

JAVASCRIPT



JavaScript



Введение в JavaScript

JavaScript - это язык программирования, который позволяет добавлять интерактивность на веб-страницы. Он может реагировать на события, выполнять операции с данными и изменять содержимое страницы динамически. Например:

```
let message = "Привет, мир!";  
console.log(message);
```

JavaScript



Введение в JavaScript

Примеры использования:

```
let age = 20;  
if (age >= 18) {  
  console.log("Вы совершеннолетний");  
} else {  
  console.log("Вы несовершеннолетний");  
}
```

```
for (let i = 0; i < 5; i++) {  
  console.log(i);  
}
```

DOM

DOM



Введение в DOM

DOM (Document Object Model) - это представление веб-страницы в виде древовидной структуры объектов. Он позволяет JavaScript взаимодействовать с элементами страницы, изменять их содержимое и атрибуты. Например:

```
<!DOCTYPE html>
<html>
<head>
  <title>Пример DOM</title>
</head>
<body>
  <p id="myParagraph">Это абзац текста.</p>

  <script>
    // Получаем элемент абзаца по его id
    let paragraph = document.getElementById("myParagraph");

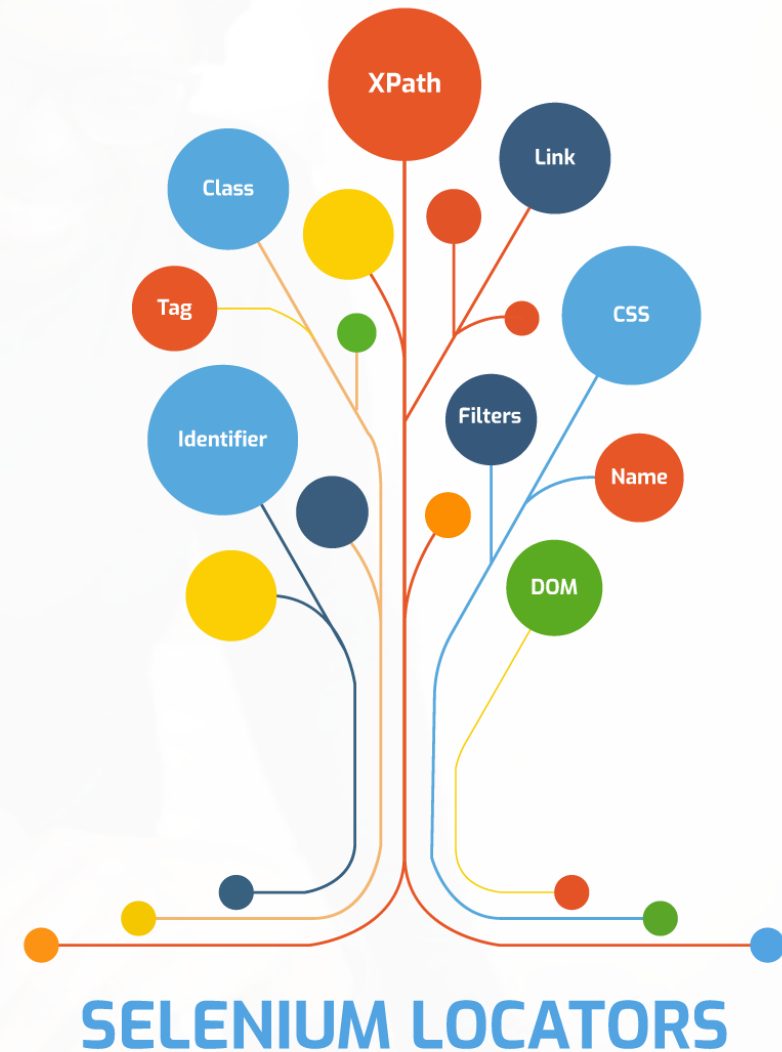
    // Изменяем текст абзаца
    paragraph.innerHTML = "Новый текст абзаца!";
  </script>
</body>
</html>
```


DOM



Введение в DOM. Locators.DOM

При работе с Selenium практически первое, с чем придется столкнуться - это локаторы. Что это такое? Локатор - это строка, уникально идентифицирующая UI-элемент. Когда вы делаете клик мышкой, ввод текста и прочие действия, вы эти действия выполняете над вполне конкретным объектом. Selenium поступает так же. Но поскольку он не умеет читать ваши мысли, то ему надо четко указать объект, для которого надо применить то или иное действие.



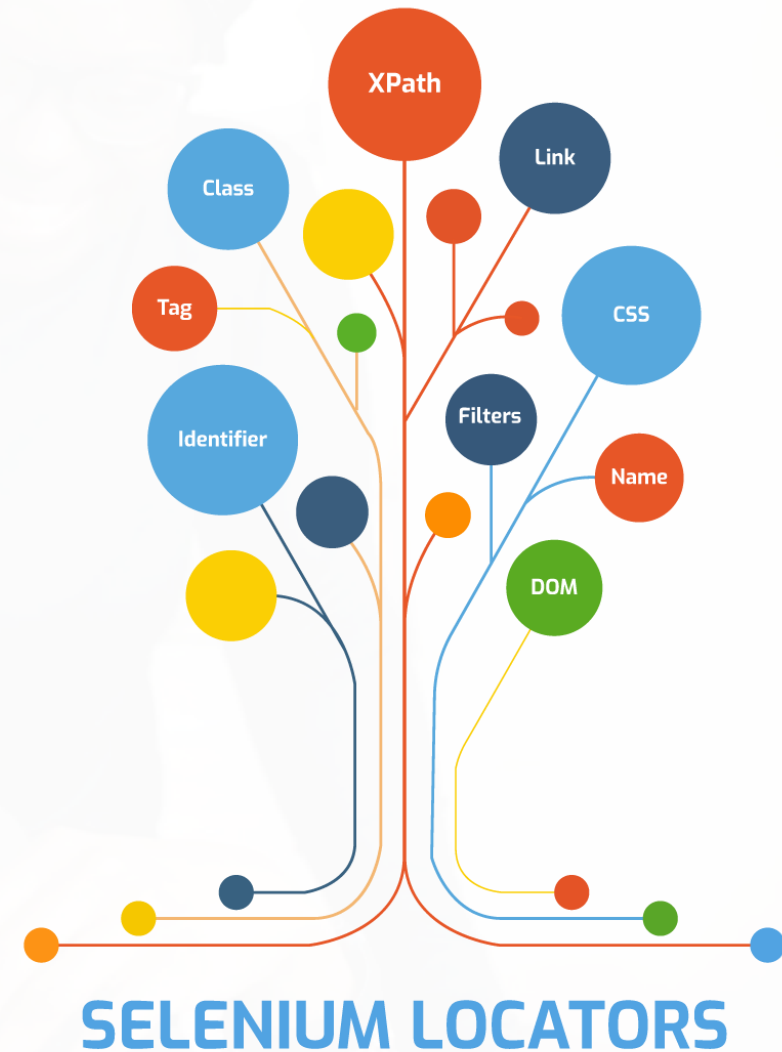
DOM



Введение в DOM. Locators.DOM

Где локаторы используются? Посмотрите на методы, отвечающие за ввод/извлечение данных из элементов управления (это методы click, type, select, isElementPresent, keyPress и т.п.). Первым параметром подобных методов всегда идет локатор.

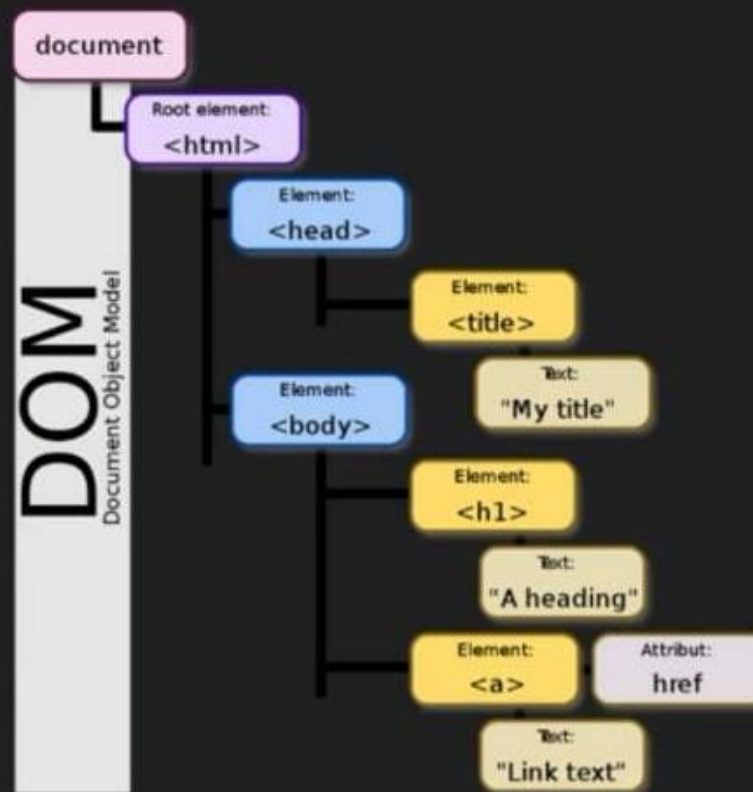
<https://www.browserstack.com/guide/locators-in-selenium>



Введение в DOM. Locators.DOM

The “DOM”

- Document Object Model
- Programming interface for HTML & XML documents
- Used to manipulate the page



Введение в DOM. Locators.DOM

Strategy	HTML	Selenium Locator
ID	<form id="login">	<code>driver.findElement(By.id("login"));</code>
Name	<input name="username" type="text"/>	<code>driver.findElement(By.name("username"));</code>
Tag Name	<form id="login">	<code>driver.findElement(By.tagName("form"));</code>
Class Name	<form class="loginForm">	<code>driver.findElement(By.className("loginForm"));</code>
Link Text	Cancel link	<code>driver.findElement(By.linkText("Cancel link"));</code>
Partial Link Text	Cancel link	<code>driver.findElement(By.partialLinkText("Cancel"));</code>
CSS	<form id="login">	<code>driver.findElement(By.cssSelector("form#login"));</code>
Xpath	<form id="login">	<code>driver.findElement(By.xpath("//form[@id='login']"));</code>

Введение в DOM. Locators.DOM

Примеры:

- Change our elements to find by cssSelector -> [attribute='value']
- Change our elements to find by cssSelector -> #id, .class
- Change out elements to find by xPath -> //*[@attribute='value']

Введение в DOM. Locators.DOM

Примеры

Нахождение по Id

Xpath: `//div[@id='myId']`

CSS: `#myId`

Нахождение по Class

Xpath: `//div[@class='myClass']`

CSS: `.myClass`

Нахождение по атрибуту

Xpath: `//input[@name='username']`

CSS: `input[name='username']`

Нахождение по нескольким атрибутам

Xpath: `//input[@name='username' and @type='submit']`

CSS: `input[name='username'][type='submit']`

SELENIUM
SELENIUM



Selenium



Методы для Python

<https://www.lambdatest.com/blog/selenium-python-cheat-sheet/>

<https://ivantay2003.medium.com/selenium-cheat-sheet-in-python-87221ee06c83>

<http://allselenium.info/python-selenium-commands-cheat-sheet-frequently-used/>

Selenium



Useful Methods

- `sendKeys()` - пишем текст
- `clear()` - стираем текст
- `click()` - нажимаем на элемент
- `getText()` - получаем текст
- `getAttribute('attribute')` – получаем значение атрибута

Selenium



Задание.

- Откройте <https://devexpress.github.io/testcafe/example/>
- Введите своё имя в поле Your Name
- Выберите все checkboxes
- Выберите checkbox
- Напишите что-нибудь в текстовом поле
- Нажмите на кнопку
- На новой странице проверяем
 - Текст элемента
 - Адрес страницы
 - Заголовок страницы



Thanks for your time 😊