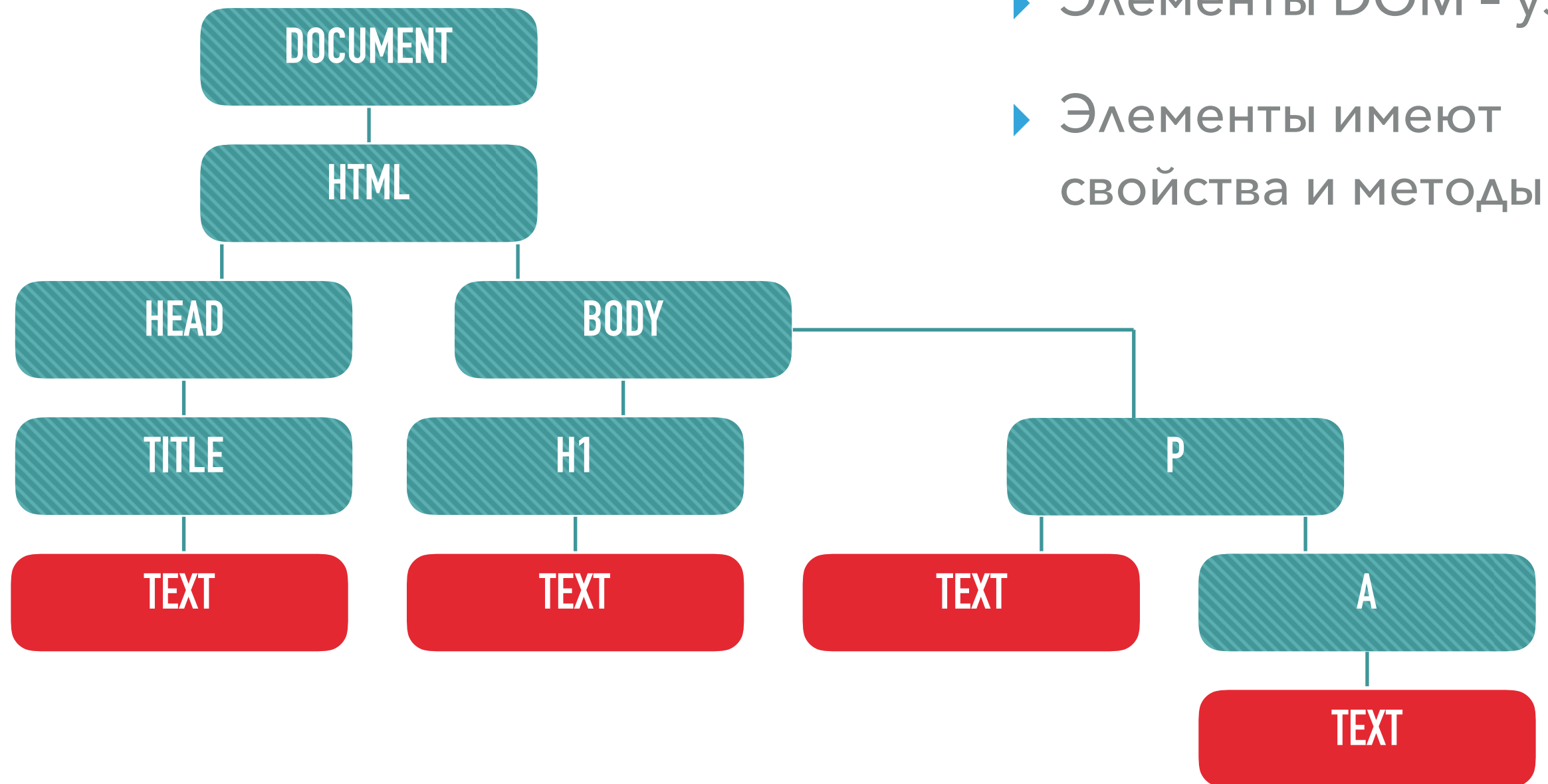


Автоматизация тестирования:
поиск элементов, составление
селекторов, ожидания,
навигация

ПОИСК ЭЛЕМЕНТОВ В SELENIUM

DOCUMENT OBJECT MODEL (DOM)



- ▶ Элементы DOM – узлы
- ▶ Элементы имеют свойства и методы

ПОИСК ЭЛЕМЕНТОВ В SELENIUM

- ▶ `driver.findElement()` возвращает первый элемент, удовлетворяющий запросу
- ▶ `driver.findElements()` находит все элементы по запросу и возвращает массив веб-элементов
- ▶ `driver.findElement().findElement()` находит элемент внутри элемента

ТИПЫ ЛОКАТОРОВ (LOCATOR STRATEGIES)

- ▶ `By.id` – по уникальному идентификатору элемента
- ▶ `By.name` – по значению атрибута «name»
- ▶ `By.className` – по названию класса
- ▶ `By.tagName` – по названию тэга
- ▶ `By.linkText` – по тексту ссылки
- ▶ `By.partialLinkText` – по части текста ссылки
- ▶ `By.css` – по CSS
- ▶ `By.xpath` – по XPATH

.FINDELEMENT(BY.LOCATOR_STRATEGY(SELECTOR))

```
describe("Проверка поиска элементов с разными типами локаторов", () => {
  before(async function() {
    driver = await new Builder().forBrowser('chrome').build();
  });

  it('Find element by id', async function() {
    await driver.get('https://google.com');
    let role = await driver.findElement(By.id("tsf")).getAttribute("role");
    assert.equal(role, 'search');
  });

  it('Find element by name', async function() {
    await driver.get('https://google.com');
    let role = await driver.findElement(By.name("f")).getAttribute("role");
    assert.equal(role, 'search');
  });

  it('Find element by className', async function() {
    await driver.get('https://google.com');
    let role = await driver.findElement(By.className("tsf nj")).getAttribute("role");
    assert.equal(role, 'search');
  });

  it('Find element by tagName', async function() {
```

.FINDELEMENTS(BY.LOCATOR_STRATEGY(SELECTOR))

```
1  require("chromedriver");
2  const assert = require("assert");
3  const { Builder, Key, By, until } = require("selenium-webdriver");
4
5  describe("Проверка поиска элементов средствами Selenium WD", () => {
6      before(async function () {
7          driver = await new Builder().forBrowser("chrome").build();
8      });
9
10     it("findElements: получен массив веб-элементов", async () => {
11         await driver.get("https://google.com");
12         const refs = await driver.findElements(By.tagName("a"));
13         const refText = await refs[21].getText();
14         assert.equal(refText, "Как работает Google Поиск");
15     });
16
17     after(() => driver.quit());
18 });
```

ПОИСК ЭЛЕМЕНТА ВНУТРИ ЭЛЕМЕНТА

```
1  require("chromedriver");
2  const assert = require("assert");
3  const { Builder, Key, By, until } = require("selenium-webdriver");
4  const query = "Beeline";
5
6  describe("Проверка поиска элементов средствами Selenium WD", () => {
7      before(async function () {
8          driver = await new Builder().forBrowser("chrome").build();
9      });
10
11     it("Поиск элемента внутри элемента", async () => {
12         await driver.get("https://google.com");
13         const searchDiv = await driver.findElement(By.xpath("//div[contains(@jsaction, 'input')]"));
14         const searchInput = await searchDiv.findElement(By.tagName("input"));
15         await searchInput.click();
16         await searchInput.sendKeys(query, Key.RETURN);
17         await driver.wait(until.elementLocated(By.id('rcnt')), 10000);
18         const title = await driver.getTitle();
19         assert.equal(title, `${query} - Поиск в Google`);
20     })
21
22     after(() => driver.quit());
23 });
```




СЕЛЕКТОРЫ. CSS И XPATH

CSS-СЕЛЕКТОРЫ

Поиск по	Элемент	Селектор
ID Тэг Значение атрибута Класс Составной селектор	<code><button id="toggleButton-regions-desktop">...</button></code>	<code>button#toggleButton-regions-desktop</code>
	<code><button id="toggleButton-regions-desktop">...</button></code>	<code>button</code>
	<code><button aria-label="Сменить регион">...</button></code>	<code>[aria-label='Сменить регион']</code>
	<code><button class="TopBar_button_2wXZ">...</button></code>	<code>.TopBar_button_2wXZ</code>
	<code><button id="toggleButton-regions-desktop" class="TopBar_button_2wXZ">...</button></code>	<code>button#toggleButton-regions-desktop.TopBar_button_2wXZ</code>

ХРАТН-СЕЛЕКТОРЫ

Поиск по	Элемент	Селектор
ID	<code><button id="toggleButton-regions-desktop">...</button></code>	<code>//button[@id='toggleButton-regions-desktop']</code>
Тэг	<code><button id="toggleButton-regions-desktop">...</button></code>	<code>//button</code>
Значение атрибута	<code><button aria-label="Сменить регион">...</button></code>	<code>//button[@aria-label='Сменить регион']</code>
Класс	<code><button class="TopBar_button_2wXZ">...</button></code>	<code>//button[@class='TopBar_button_2wXZ']</code>
Составной селектор	<code><button id="toggleButton-regions-desktop" class="TopBar_button_2wXZ">...</button></code>	<code>//button[@id='toggleButton-regions-desktop' and @class='TopBar_button_2wXZ']</code>

ПОИСК ДОЧЕРНИХ ЭЛЕМЕНТОВ

```
▼ <div class="TopBar_componentInner_1hLT">
  ▶ <div>...</div>
  ▶ <div class="TopBar_segments_1g8f">...</div> == $0
  ▶ <div>...</div>
</div>
```

	CSS	XPATH
Селектор	▶ div.TopBar_componentInner_1hLT > div.TopBar_segments_1g8f	//div[@class='TopBar_componentInner_1hLT']/div[@class='TopBar_segments_1g8f']
	▶ div.TopBar_componentInner_1hLT>div.TopBar_segments_1g8f	
	▶ .TopBar_componentInner_1hLT > .TopBar_segments_1g8f	

ИСПОЛЬЗОВАНИЕ ПОРЯДКОВОГО НОМЕРА ДОЧЕРНЕГО ЭЛЕМЕНТА

```
▼ <div class="TopBar_componentInner_1hLT">
  ▶ <div>...</div>
  ▶ <div class="TopBar_segments_1g8f">...</div> == $0
  ▶ <div>...</div>
</div>
```

	CSS	XPATH
Селектор	▶ div.TopBar_componentInner_1hLT > div:nth-child(2)	//div[@class='TopBar_componentInner_1hLT']/div[2]
	▶ div.TopBar_componentInner_1hLT>div:nth-child(2)	
	▶ .TopBar_componentInner_1hLT > nth-child(2)	

ПСЕВДОКЛАССЫ ДЛЯ ФИЛЬТРАЦИИ ПО МЕСТУ СРЕДИ СОСЕДЕЙ

- ▶ :first-child – первый потомок своего родителя
- ▶ :last-child – последний потомок своего родителя.
- ▶ :only-child – единственный потомок своего родителя, соседних элементов нет
- ▶ :nth-child(a) – потомок номер a своего родителя.
Нумерация начинается с 1

ФИЛЬТРАЦИЯ ПО МЕСТУ СРЕДИ СОСЕДЕЙ С ТЕМ ЖЕ ТЭГОМ

- ▶ :first-of-type
- ▶ :last-of-type
- ▶ :only-of-type
- ▶ :nth-of-type
- ▶ :nth-last-of-type

СЕЛЕКТОРЫ АТТРИБУТОВ: CSS

```
▼<div>
  <button class="TopBar_button_2wXZ" aria-label="Сменить язык">RU
  </button>
  <button class="TopBar_region_2jDj TopBar_button_2wXZ" id=
    "toggleButton-regions-desktop" aria-label="Сменить регион">Москва
  </button> == $0
</div>
```

- ▶ [aria-label] – атрибут установлен
- ▶ [aria-label='Сменить регион'] – значение атрибута равно
- ▶ [aria-label^='Смен'] – атрибут начинается с «Смен»
- ▶ [aria-label|='Смен'] – атрибут равен или начинается со «Смен»
- ▶ [aria-label*='Смен'] – атрибут содержит подстроку «Смен»
- ▶ [aria-label~='Смен'] – атрибут содержит «Смен», как одно из значений через пробел
- ▶ [aria-label\$='гион'] – атрибут заканчивается подстрокой «гион»

ПОЛЕЗНЫЕ ССЫЛКИ

- ▶ https://developer.mozilla.org/ru/docs/Web/CSS/CSS_Селекторы – про составление CSS-селекторов
- ▶ <https://docs.google.com/document/d/1PdfKMDfoqFIIF4tN1jKrOf1iZ1rqESy2xVMIj3uuV3g/pub> – про составление XPATH-селекторов



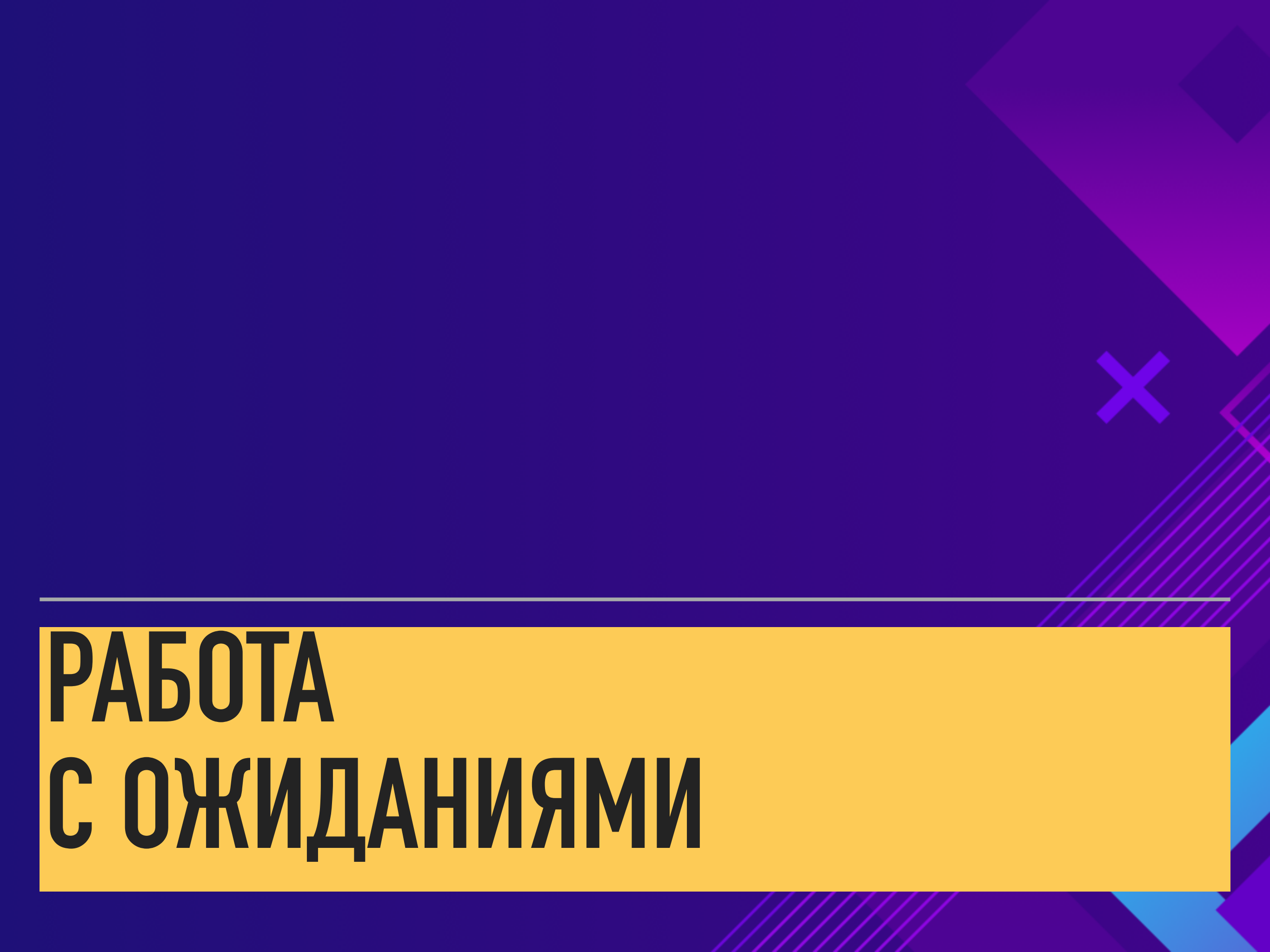
РАБОТА С ВЕБ-ЭЛЕМЕНТАМИ

ВЗАИМОДЕЙСТВИЕ С ВЕБ-ЭЛЕМЕНТАМИ

- ▶ `element.click()` – скроллит к элементу и нажимает на него
- ▶ `element.clear()` – скроллит к редактируемому элементу и очищает контент, которым он заполнен
- ▶ `element.sendKeys()` – скроллит к элементу и имитирует взаимодействие с клавиатурой

СОСТОЯНИЯ ВЕБ-ЭЛЕМЕНТОВ

- ▶ isEnabled()
- ▶ isDisplayed()
- ▶ isSelected()
- ▶ getAttribute()
- ▶ getText()



РАБОТА С ОЖИДАНИЯМИ

ПОЛЕЗНЫЕ ССЫЛКИ

- ▶ <https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/support/ui/ExpectedConditions.html>
- ▶ <https://www.selenium.dev/docs/site/en/webdriver/ waits/>

НАВИГАЦИЯ В БРАУЗЕРЕ

ОСНОВНЫЕ МЕТОДЫ НАВИГАЦИИ В БРАУЗЕРЕ

- ▶ `driver.get()`
- ▶ `driver.navigate().refresh()`
- ▶ `driver.navigate().back()`
- ▶ `driver.getTitle()`