



Become QA Auto



Selenium WebDriver. Базові операції

Бутенко Сергій



План лекції



Відкриття/Закриття
браузера



Пошук елементів



Відкриття вебсторінки



Дії над елементами



Selenium WebDriver



Sign in to GitHub

Username or email address

Password

[Forgot password?](#)

Sign in

New to GitHub? [Create an account](#) .

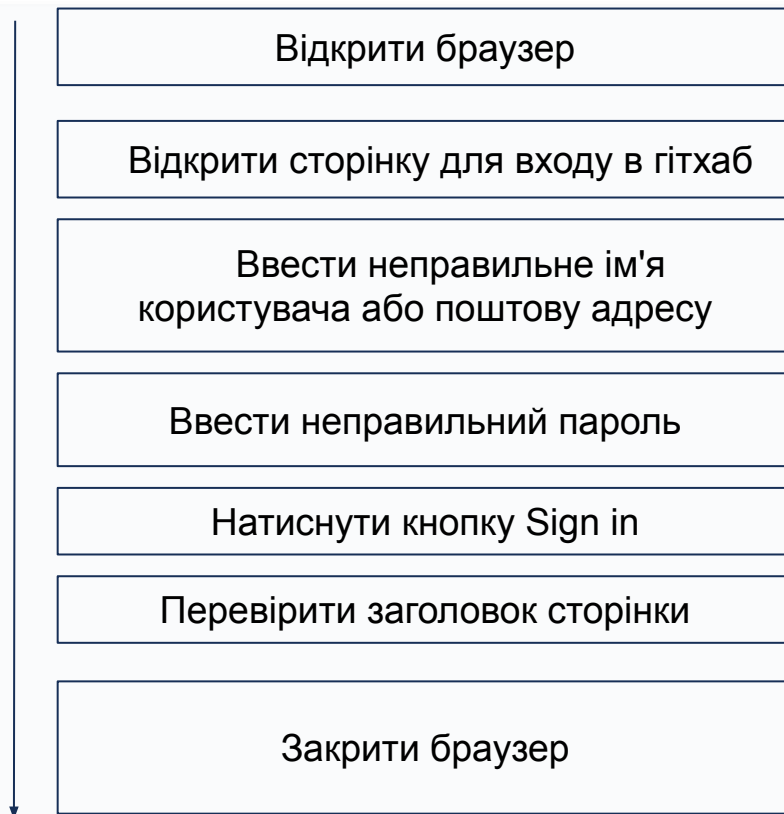
[Terms](#) [Privacy](#) [Security](#) [Contact GitHub](#)

- Неуспішний сценарій - користувач помилився при вводі даних

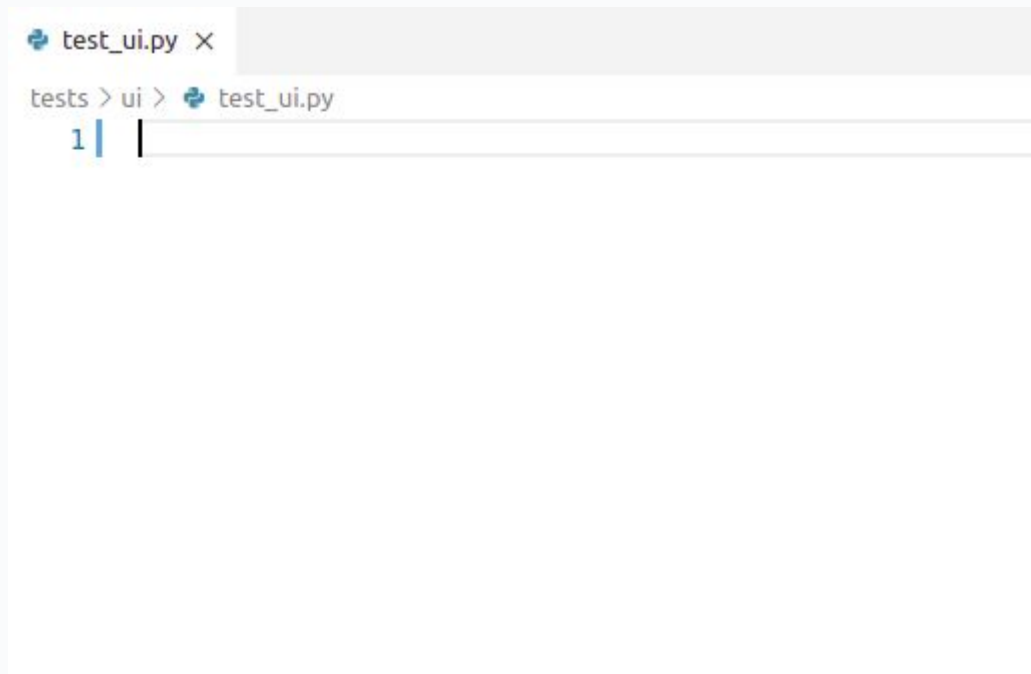
⦿ Selenium WebDriver - алгоритм тесту



Неуспішний сценарій - користувач помилився при вводі даних



⦿ Базові Операції - Відкриття/Закриття браузера

A screenshot of a code editor window. The title bar at the top shows a blue icon, the filename 'test_ui.py', and a close button 'X'. Below the title bar is a breadcrumb navigation path: 'tests > ui > test_ui.py'. The main editing area shows a single line of code starting with the number '1' followed by a vertical cursor bar and a short horizontal line, indicating the current position for typing.

○ Базові Операції - Відкриття/Закриття браузера



test_ui.py X

tests > ui > test_ui.py > ...

You, seconds ago | 1 author (You)

```
1 import pytest
2
3 from selenium import webdriver
4 from selenium.webdriver.chrome.service import Service
5
6
7 @pytest.mark.ui
8 def test_check_incorrect_username():
9     # Створення об'єкту для керування браузером
10    driver = webdriver.Chrome(
11        service=Service(r"/home/sbutenko/repos/LnD/Become QA Auto/" + "chromedriver")
12    )
13
14    # відкриваємо сторінку https://github.com/login
15    driver.get("https://github.com/login")
16
17    # Закриваємо браузер
18    driver.close()
```

Базові Операції - Відкриття/Закриття браузера



test_ui.py x

tests > ui > test_ui.py > test_check_incorrect_username

You, seconds ago | 1 author (You)

```
1 import pytest
2
3 from selenium import webdriver
4 from selenium.webdriver.chrome.service import Service
5
6
7 @pytest.mark.ui
8 def test_check_incorrect_username():
9     # Створення об'єкту для керування браузером
10    driver = webdriver.Chrome(
11        service=Service(r"/home/sbutenko/repos/LnD/Become QA Auto/" + "chromedriver.exe")
12    )
13
14    # відкриваємо сторінку https://github.com/login
15    driver.get("https://github.com/login")
16
17    # Закриваємо браузер
18    driver.close()
```

You, seconds ago • Uncommitted changes

○ Базові Операції - Відкриття/Закриття браузера



```
test_ui.py x
tests > ui > test_ui.py > ...
You, seconds ago | 1 author (You)
1 import pytest
2
3 from selenium import webdriver
4 from selenium.webdriver.chrome.service import
5
6
7 @pytest.mark.ui
8 def test_check_incorrect_username():
9     # Створення об'єкту для керування браузером
10    driver = webdriver.Chrome(
11        service=Service(r"/home/sbutenko/re
12    )
13
14    # Відкриваємо сторінку https://github.com
15    driver.get("https://github.com/login")
16
17    # Закриваємо браузер
18    driver.close()
```

Метод **get** - це операція відкриття сторінки в браузері.

Синтаксис виклику:
driver.get("URL-адреса, яку ви бажаєте відкрити")

○ Базові Операції - Відкриття/Закриття браузера



```
test_ui.py X
tests > ui > test_ui.py > ...
You, seconds ago | 1 author (You)
1 import pytest
2
3 from selenium import webdriver
4 from selenium.webdriver.chrome.service import Service
5
6
7 @pytest.mark.ui
8 def test_check_incorrect_username():
9     # Створення об'єкту для керування браузером
10    driver = webdriver.Chrome(
11        service=Service(r"/home/sbutenko/re
12    )
13
14    # Відкриваємо сторінку https://github.com/login
15    driver.get("https://github.com/login")
16
17    # Закриваємо браузер
18    driver.close()
```

Метод **close** - це операція закриття браузера

Синтаксис виклику:
driver.close()

⦿ Базові Операції - Відкриття/Закриття браузера



```
sbutenko@kbp1-lhp-a00394:~/repos/LnD/Become QA Auto
→ Become QA Auto git:(selenium) X pytest -m ui
===== test session starts =====
platform linux -- Python 3.8.10, pytest-7.2.0, pluggy-0.13.1
rootdir: /home/sbutenko/repos/LnD/Become QA Auto, configfile: pytest.ini
plugins: allure-pytest-2.10.0, rerunfailures-10.2, Faker-14.2.0, forked-1.3.0, a
nyio-3.4.0, xdist-2.5.0, metadata-1.10.0, ordering-0.6, html-3.1.1, instafail-0.
4.2
collected 21 items / 20 deselected / 1 selected

tests/ui/test_ui.py . [100%]

===== 1 passed, 20 deselected in 2.48s =====
→ Become QA Auto git:(selenium) X
```

⦿ Selenium WebDriver - алгоритм тесту



Неуспішний сценарій - користувач помилився при вводі даних



Відкрити браузер



Відкрити сторінку для входу в гітхаб

Ввести неправильне ім'я користувача або поштову адресу

Ввести неправильний пароль

Натиснути кнопку Sign in

Перевірити заголовок сторінки



Закрити браузер



Тестові сценарії



Sign in to GitHub

Username or email address

Password

[Forgot password?](#)

Sign in

New to GitHub? [Create an account](#) .

[Terms](#) [Privacy](#) [Security](#) [Contact GitHub](#)

Базові Операції - Пошук елементів



```
test_ui.py X
tests > ui > test_ui.py > ...
You, seconds ago | 1 author (You)
1 import pytest
2
3 from selenium import webdriver
4 from selenium.webdriver.chrome.service import Service
5 from selenium.webdriver.common.by import By
6
7 | You, seconds ago • Uncommitted changes
8 @pytest.mark.ui
9 def test_check_incorrect_username():
10     # Створення об'єкту для керування браузером
11     driver = webdriver.Chrome(
12         service=Service(r"/home/sbutenko/repos/LnD/Become QA Auto/" + "chromedriver")
13     )
14
15     # відкриваємо сторінку https://github.com/login
16     driver.get("https://github.com/login")
17
18     # Знаходимо поле, в яке будемо вводити неправильне ім'я користувача або поштову адресу
19     login_elem = driver.find_element(By.ID, "login_field")
20
21     # Вводимо неправильне ім'я користувача або поштову адресу
22     login_elem.send_keys("sergiibutenko@mistakeinemail.com")
23
24     # Закриваємо браузер
25     driver.close()
26
```



Тестові сценарії



Sign in to GitHub

Username or email address

Password

[Forgot password?](#)

Sign in

New to GitHub? [Create an account](#) .

[Terms](#) [Privacy](#) [Security](#) [Contact GitHub](#)



Тестові сценарії



The screenshot displays the GitHub login page in a web browser. A context menu is open over the password input field, showing options like Undo, Redo, Cut, Copy, Paste, and AdBlock. The Chrome DevTools window is open, showing the HTML structure of the login form and the CSS styles applied to the input fields. The HTML shows a form with a hidden authenticity token and a login input field. The CSS shows styles for the form control, including focus and visible states.

Sign in to GitHub

Username or email address

Password

Sign in

New to GitHub? [Create an account.](#)

Terms Privacy Security Contact GitHub

Elements Sources Recorder Performance insights Console Network Performance

Styles Computed Layout

Filter: :hov .cls +

element.style { }

.auth-form-body .input-block { margin-top: 4px; margin-bottom: 16px; }

.form-control:focus-visible, .form-select:focus-visible { border-color: var(--color-accent-fg); outline: none; box-shadow: inset 0 0 1px var(--color-accent-fg); }

.form-control:focus, .form-select:focus { border-color: var(--color-accent-fg); outline: none; box-shadow: inset 0 0 1px var(--color-accent-fg); }

... form px-3 div.auth-form-body mt-3 form input#login_field.form-control.input-block.js-login-field ...

Recorder panel updates
New step context menu, option to copy a single step from a script, remove the first navigation step, and more.

Improved JavaScript debugging
Inline preview for WeakRef, correct preview of shadowed inline variable, and more.

Go to symbols for TypeScript
Support view and navigate symbols (Ctrl+Shift+O / Cmd+Shift+O) for TypeScript files.

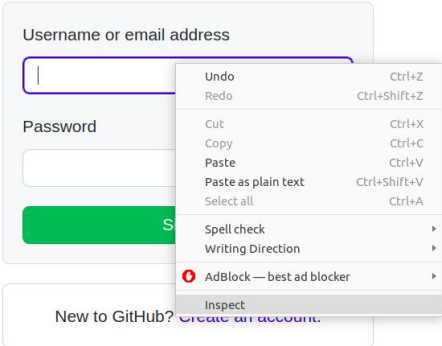
Learn more Close

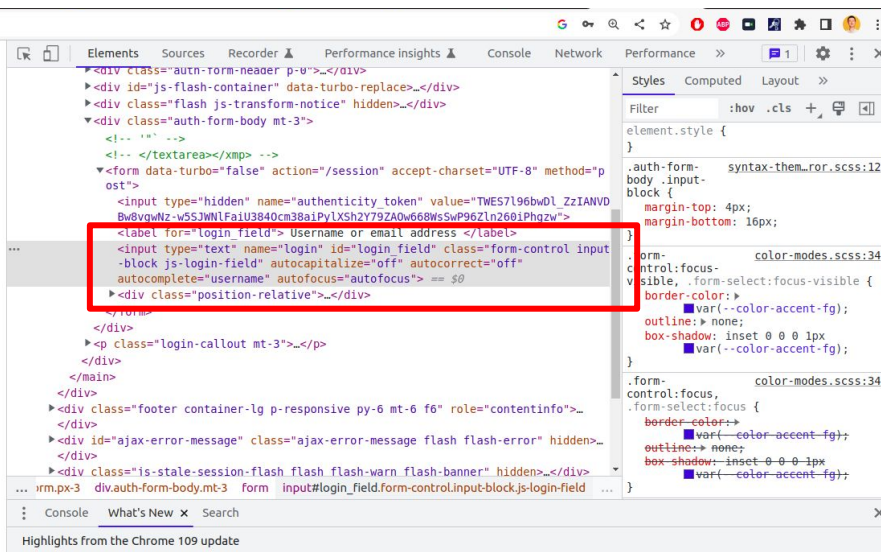
new 109

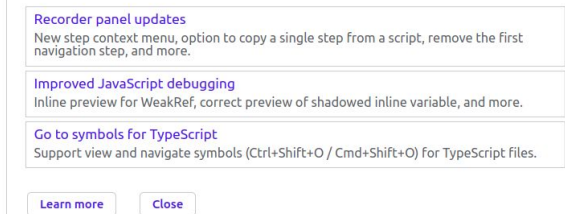



Тестові сценарії











Базові Операції - Пошук елементів



test_ui.py X

tests > ui > test_ui.py > ...

You, seconds ago | 1 author (You)

```
1 import pytest
2
3 from selenium import webdriver
4 from selenium.webdriver.chrome.service import Service
5 from selenium.webdriver.common.by import By
6
7 | You, seconds ago • Uncommitted changes
8 @pytest.mark.ui
9 def test_check_incorrect_username():
10     # Створення об'єкту для керування браузером
11     driver = webdriver.Chrome(
12         service=Service(r"/home/sbutenko/repos/LnD/Become
13     )
14
15     # відкриваємо сторінку https://github.com/login
16     driver.get("https://github.com/login")
17
18     # Знаходимо поле, в яке будемо вводити неправильне ім'я
19     login_elem = driver.find_element(By.ID, "login_field")
20
21     # Вводимо неправильне ім'я користувача або поштову адр
22     login_elem.send_keys("sergiibutenko@mistakeinemail.com")
23
24     # Закриваємо браузер
25     driver.close()
26
```

Метод **find_element** - це операція пошуку потрібного елемента за іменем атрибута та значенням цього атрибута

Синтаксис виклику:
driver.find_element(ІМ'Я АТРИБУТА, ЗНАЧЕННЯ АТРИБУТА)

Базові Операції - Пошук елементів



```
test_ui.py X
tests > ui > test_ui.py > ...
You, seconds ago | 1 author (You)
1 import pytest
2
3 from selenium import webdriver
4 from selenium.webdriver.chrome.service import Service
5 from selenium.webdriver.common.by import By
6
7 | You, seconds ago • Uncommitted changes
8 @pytest.mark.ui
9 def test_check_incorrect_username():
10     # Створення об'єкту для керування браузером
11     driver = webdriver.Chrome(
12         service=Service(r"/home/sbutenko/repos/LnD/Become QA Auto/" + "chromedriver")
13     )
14
15     # відкриваємо сторінку https://github.com/login
16     driver.get("https://github.com/login")
17
18     # Знаходимо поле, в яке будемо вводити неправильне ім'я користувача або поштову адресу
19     login_elem = driver.find_element(By.ID, "login_field")
20
21     # Вводимо неправильне ім'я користувача або поштову адресу
22     login_elem.send_keys("sergiibutenko@mistakeinemail.com")
23
24     # Закриваємо браузер
25     driver.close()
26
```

Базові Операції - Пошук елементів



```
test_ui.py X
tests > ui > test_ui.py > ...
You, seconds ago | 1 author (You)
1 import pytest
2
3 from selenium import webdriver
4 from selenium.webdriver.chrome.service import Service
5 from selenium.webdriver.common.by import By
6
7 | You, seconds ago • Uncommitted changes
8 @pytest.mark.ui
9 def test_check_incorrect_username():
10     # Створення об'єкту для керування браузером
11     driver = webdriver.Chrome(
12         service=Service(r"/home/sbutenko/repos/LnD/Become QA Auto/" + "chromedriver")
13     )
14
15     # відкриваємо сторінку https://github.com/login
16     driver.get("https://github.com/login")
17
18     # Знаходимо поле, в яке будемо вводити неправильне ім'я користувача або поштову адресу
19     login_elem = driver.find_element(By.ID, "login_field")
20
21     # Вводимо неправильне ім'я користувача або поштову адресу
22     login_elem.send_keys("sergiibutenko@mistakeinemail.com")
23
24     # Закриваємо браузер
25     driver.close()
26
```

Базові Операції - Пошук елементів



```
test_ui.py X
tests > ui > test_ui.py > ...
You, seconds ago | 1 author (You)
1 import pytest
2
3 from selenium import webdriver
4 from selenium.webdriver.chrome.service import Service
5 from selenium.webdriver.common.by import By
6
7 | You, seconds ago • Uncommitted changes
8 @pytest.mark.ui
9 def test_check_incorrect_username():
10     # Створення об'єкту для керування браузером
11     driver = webdriver.Chrome(
12         service=Service(r"/home/sbutenko/repos/LnD/Become QA Auto/" + "chromedriver")
13     )
14
15     # відкриваємо сторінку https://github.com/login
16     driver.get("https://github.com/login")
17
18     # Знаходимо поле, в яке будемо вводити неправильне ім'я користувача або поштову адресу
19     login_elem = driver.find_element(By.ID, "login_field")
20
21     # Вводимо неправильне ім'я користувача або поштову адресу
22     login_elem.send_keys("sergiibutenko@mistakeinemail.com")
23
24     # Закриваємо браузер
25     driver.close()
26
```

Базові Операції - Пошук елементів



test_ui.py X

tests > ui > test_ui.py > ...

```
You, seconds ago | 1 author (You)
1  import pytest
2
3  from selenium import webdriver
4  from selenium.webdriver.chrome.service import Service
5  from selenium.webdriver.common.by import By
6
7  You, seconds ago • Uncommitted changes
8  @pytest.mark.ui
9  def test_check_incorrect_username():
10     # Створення об'єкту для керування браузером
11     driver = webdriver.Chrome(
12         service=Service(r"/home/sbutenko/repos/LnD/Become QA Auto/")
13     )
14
15     # відкриваємо сторінку https://github.com/login
16     driver.get("https://github.com/login")
17
18     # Знаходимо поле, в яке будемо вводити неправильне ім'я користувача
19     login_elem = driver.find_element(By.ID, "login_field")
20
21     # Вводимо неправильне ім'я користувача або поштову адресу
22     login_elem.send_keys("sergiibutenko@mistakeinemail.com")
23
24     # Закриваємо браузер
25     driver.close()
26
```

Метод **send_keys** - це операція введення тексту до елемента сторінки

Синтаксис виклику:
element.send_keys(ТЕКСТ, ЯКИЙ ВИ БАЖАЄТЕ ВВЕСТИ)

◎ Базові Операції - Пошук елементів



```
sbutenko@kbp1-lhp-a00394:~/repos/LnD/Become QA Auto
→ Become QA Auto git:(selenium) X pytest -m ui
===== test session starts =====
platform linux -- Python 3.8.10, pytest-7.2.0, pluggy-0.13.1
rootdir: /home/sbutenko/repos/LnD/Become QA Auto, configfile: pytest.ini
plugins: allure-pytest-2.10.0, rerunfailures-10.2, Faker-14.2.0, forked-1.3.0, a
nyio-3.4.0, xdist-2.5.0, metadata-1.10.0, ordering-0.6, html-3.1.1, instafail-0.
4.2
collected 21 items / 20 deselected / 1 selected

tests/ui/test_ui.py . [100%]

===== 1 passed, 20 deselected in 2.48s =====
→ Become QA Auto git:(selenium) X
```


Базові Операції - Пошук елементів



test_ui.py x

tests > ui > test_ui.py > ...

```
You, seconds ago | 1 author (You)
1  import pytest
2
3  from selenium import webdriver
4  from selenium.webdriver.chrome.service import Service
5  from selenium.webdriver.common.by import By
6  import time
7
8  @pytest.mark.ui
9  def test_check_incorrect_username():
10     # Створення об'єкту для керування браузером
11     driver = webdriver.Chrome(
12         service=Service(r"/home/sbutenko/repos/LnD/Become QA Auto/" + "chromedriver")
13     )
14
15     # відкриваємо сторінку https://github.com/login
16     driver.get("https://github.com/login")
17
18     # Знаходимо поле, в яке будемо вводити неправильне ім'я користувача або поштову адресу
19     login_elem = driver.find_element(By.ID, "login_field")
20
21     # Вводимо неправильне ім'я користувача або поштову адресу
22     login_elem.send_keys("sergiibutenko@mistakeinemail.com")
23     time.sleep(3)
24
25     # Закриваємо браузер
26     driver.close()
```

◎ Базові Операції - Пошук елементів



```
sbutenko@kbp1-lhp-a00394:~/repos/LnD/Become QA Auto
→ Become QA Auto git:(selenium) X pytest -m ui
===== test session starts =====
platform linux -- Python 3.8.10, pytest-7.2.0, pluggy-0.13.1
rootdir: /home/sbutenko/repos/LnD/Become QA Auto, configfile: pytest.ini
plugins: allure-pytest-2.10.0, rerunfailures-10.2, Faker-14.2.0, forked-1.3.0, a
nyio-3.4.0, xdist-2.5.0, metadata-1.10.0, ordering-0.6, html-3.1.1, instafail-0.
4.2
collected 21 items / 20 deselected / 1 selected

tests/ui/test_ui.py . [100%]

===== 1 passed, 20 deselected in 2.48s =====
→ Become QA Auto git:(selenium) X
```


⦿ Selenium WebDriver - алгоритм тесту



Неуспішний сценарій - користувач помилився при вводі даних



Відкрити браузер



Відкрити сторінку для входу в гітхаб



Ввести неправильне ім'я користувача або поштову адресу

Ввести неправильний пароль

Натиснути кнопку Sign in

Перевірити заголовок сторінки



Закрити браузер



Sign in to GitHub

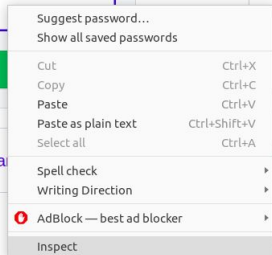
Password

[Forgot password?](#)

Sign in

New to GitHub? [Create a](#)

[Terms](#) [Privacy](#) [Security](#) [Contact GitHub](#)



```

Elements Sources Recorder Performance insights Console Network
<!-- -->
<!-- </textarea></xmp> -->
▼ <form data-turbo="false" action="/session" accept-charset="UTF-8" method="post">
  <input type="hidden" name="authenticity_token" value="yPTE5Nr7gWRT3831qHw4p045r01RhasY4G6AT6h0nKNAAT_D65MAFGMeJXgWSDH125oa0C294t6Davs9Mj0">
  <label for="login_field">Username or email address </label>
  <input type="text" name="login" id="login_field" class="form-control input-block js-login-field" autocapitalize="off" autocorrect="off" autocomplete="username" autofocus">
  <input type="password" name="password" id="password" class="form-control input-block js-password-field" autocomplete="current-password">
  <input type="hidden" class="js-webauthn-support" name="webauthn-support" value="supported">
  <input type="hidden" class="js-webauthn-suvpaas-support" name="webauthn-suvpaas-support" value="unsupported">
  <input type="hidden" name="return_to" id="return_to" value="https://github.com/login" autocomplete="off" class="form-control">
  <input type="hidden" name="allow_signup" id="allow_signup" autocomplete="off" class="form-control">
  <input type="hidden" name="client_id" id="client_id" autocomplete="off" class="form-control">
  <input type="hidden" name="integration" id="integration" autocomplete="off" class="form-control">

```



The screenshot shows the VS Code editor with the Sass code for the 'form-control' class. The code is organized into a tree view on the left, with 'form-control' selected. The main editor shows the Sass code for 'form-control', including the 'padding' and 'border' properties. The 'border' property is highlighted with a red box.

```

form-control {
  padding: 5px 12px;
  font-size: 14px;
  line-height: 20px;
  color: var(--color-fg-default);
  vertical-align: middle;
  background-color:
    var(--color-canvas-default);
  background-repeat: no-repeat;
  background-position: right 8px
    center;
  border: 1px solid
    var(--color-border-default);
}

```

Highlights from the Chrome 109 update

Recorder panel updates

New step context menu, option to copy a single step from a script, remove the first navigation step, and more.

Improved JavaScript debugging

Inline preview for WeakRef, correct preview of shadowed inline variable, and more.

[Go to symbols for TypeScript](#)

Support view and navigate symbols (Ctrl+Shift+O / Cmd+Shift+O) for TypeScript files.

[Learn more](#)

Close



Базові Операції - Пошук елементів



```
test_ui.py x
tests > ui > test_ui.py > ...
You, seconds ago | 1 author (You)
1 import pytest
2
3 from selenium import webdriver
4 from selenium.webdriver.chrome.service import Service
5 from selenium.webdriver.common.by import By
6 import time
7
8 @pytest.mark.ui
9 def test_check_incorrect_username():
10     # Створення об'єкту для керування браузером
11     driver = webdriver.Chrome(
12         service=Service(r"/home/sbutenko/repos/LnD/Become QA Auto/" + "chromedriver")
13     )
14
15     # відкриваємо сторінку https://github.com/login
16     driver.get("https://github.com/login")
17
18     # Знаходимо поле, в яке будемо вводити неправильне ім'я користувача або поштову адресу
19     login_elem = driver.find_element(By.ID, "login_field")
20
21     # Вводимо неправильне ім'я користувача або поштову адресу
22     login_elem.send_keys("sergiibutenko@mistakeinemail.com")
23
24     # Знаходимо поле, в яке будемо вводити неправильний пароль
25     pass_elem = driver.find_element(By.ID, "password")
26
27     # Вводимо неправильний пароль
28     pass_elem.send_keys("wrong password")
29     time.sleep(3)
30
31     # Закриваємо браузер
32     driver.close()
--
```

◎ Базові Операції - Пошук елементів



```
sbutenko@kbp1-lhp-a00394:~/repos/LnD/Become QA Auto
→ Become QA Auto git:(selenium) X pytest -m ui
===== test session starts =====
platform linux -- Python 3.8.10, pytest-7.2.0, pluggy-0.13.1
rootdir: /home/sbutenko/repos/LnD/Become QA Auto, configfile: pytest.ini
plugins: allure-pytest-2.10.0, rerunfailures-10.2, Faker-14.2.0, forked-1.3.0, a
nyio-3.4.0, xdist-2.5.0, metadata-1.10.0, ordering-0.6, html-3.1.1, instafail-0.
4.2
collected 21 items / 20 deselected / 1 selected

tests/ui/test_ui.py . [100%]

===== 1 passed, 20 deselected in 2.48s =====
→ Become QA Auto git:(selenium) X
```

⦿ Selenium WebDriver - алгоритм тесту



Неуспішний сценарій - користувач помилився при вводі даних



Відкрити браузер



Відкрити сторінку для входу в гітхаб



Ввести неправильне ім'я користувача або поштову адресу



Ввести неправильний пароль

Натиснути кнопку Sign in

Перевірити заголовок сторінки



Закрити браузер

Базові Операції - Пошук елементів



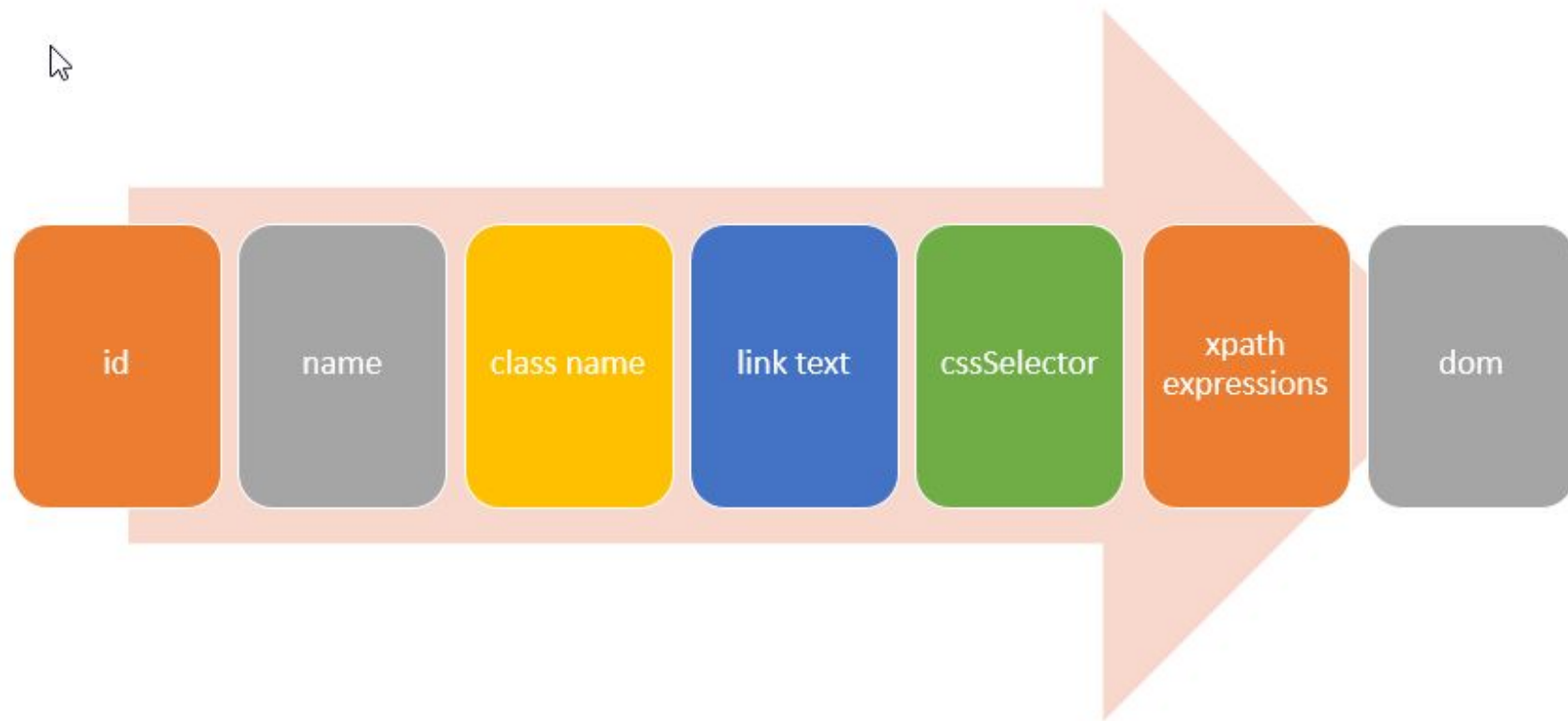
The screenshot shows a web browser displaying the GitHub login page. The page has a white background with the GitHub logo at the top. Below the logo, there's a section titled "Sign in to GitHub" with a form containing "Username or email address" and "Password" fields, a "Forgot password?" link, and a green "Sign in" button. At the bottom, there's a link to "New to GitHub? Create an account".

The browser's developer tools are open, showing the "Elements" panel on the right. The "Sign in" button is selected, and its HTML structure is visible: a nested structure of `<div>`, `<input>`, and `<button>` tags. The "Styles" panel on the right shows the CSS rules applied to the button, including `color-modes.scss:34` and `background-color`.

A context menu is open over the "Sign in" button, showing options like "Back", "Forward", "Reload", "Save as...", "Print...", "Cast...", "Search images with Google", "Send to your devices", "Create QR Code for this page", "Translate to українська", "AdBlock — best ad blocker", "View page source", and "Inspect". The "Inspect" option is highlighted.

At the bottom right, there's a blue banner with a white "new 109" text and a YouTube logo, indicating a new version of the browser.

◎ Пріоритет унікальних ідентифікаторів





Базові Операції - Пошук елементів

The screenshot shows a web browser window with the GitHub login page. The page has a white background with the GitHub logo at the top left. Below the logo is the text "Sign in to GitHub". There are two input fields: "Username or email address" and "Password". Below the password field is a link "Forgot password?". A green "Sign in" button is at the bottom of the login form. A context menu is open over the "Sign in" button, showing options like "Back", "Forward", "Reload", "Save as...", "Print...", "Cast...", "Search images with Google", "Send to your devices", "Create QR Code for this page", "Translate to українська", "AdBlock — best ad blocker", "View page source", and "Inspect". The "Inspect" option is highlighted. The browser's developer tools are open on the right, showing the "Elements" panel with the HTML structure of the page. The "Sign in" button is highlighted in the HTML tree. The browser's address bar shows the URL "https://github.com/session".

Базові Операції - Пошук елементів



```
test_ui.py x
tests > ui > test_ui.py > ...
You, seconds ago | 1 author (You)
1 import pytest
2
3 from selenium import webdriver
4 from selenium.webdriver.chrome.service import Service
5 from selenium.webdriver.common.by import By
6 import time
7
8 @pytest.mark.ui
9 def test_check_incorrect_username():
10     # Створення об'єкту для керування браузером
11     driver = webdriver.Chrome(
12         service=Service(r"/home/sbutenko/repos/LnD/Become QA Auto/" + "chromedriver")
13     )
14
15     # відкриваємо сторінку https://github.com/login
16     driver.get("https://github.com/login")
17
18     # Знаходимо поле, в яке будемо вводити неправильне ім'я користувача або поштову адресу
19     login_elem = driver.find_element(By.ID, "login_field")
20
21     # Вводимо неправильне ім'я користувача або поштову адресу
22     login_elem.send_keys("sergiibutenko@mistakeinemail.com")
23
24     # Знаходимо поле, в яке будемо вводити неправильний пароль
25     pass_elem = driver.find_element(By.ID, "password")
26
27     # Вводимо неправильний пароль
28     pass_elem.send_keys("wrong password")
29
30     # Знаходимо кнопку sign in
31     btn_elem = driver.find_element(By.NAME, "commit")
32
33     # Емулюємо клік лівою кнопкою мишки
34     btn_elem.click()
35     time.sleep(3)
36
37     # Закриваємо браузер
38     driver.close()
```

Базові Операції - Пошук елементів



```
test_ui.py x
tests > ui > test_ui.py > ...
You, seconds ago | 1 author (You)
1 import pytest
2
3 from selenium import webdriver
4 from selenium.webdriver.chrome.service import Service
5 from selenium.webdriver.common.by import By
6 import time
7
8 @pytest.mark.ui
9 def test_check_incorrect_username():
10     # Створення об'єкту для керування браузером
11     driver = webdriver.Chrome(
12         service=Service(r"/home/sbutenko/repos/LnD/Become QA Auto/" + "chromedriver")
13     )
14
15     # відкриваємо сторінку https://github.com/login
16     driver.get("https://github.com/login")
17
18     # Знаходимо поле, в яке будемо вводити неправильне ім'я користувача або поштову адресу
19     login_elem = driver.find_element(By.ID, "login_field")
20
21     # Вводимо неправильне ім'я користувача або поштову адресу
22     login_elem.send_keys("sergiibutenko@mistakeinemail.com")
23
24     # Знаходимо поле, в яке будемо вводити неправильний пароль
25     pass_elem = driver.find_element(By.ID, "password")
26
27     # Вводимо неправильний пароль
28     pass_elem.send_keys("wrong password")
29
30     # Знаходимо кнопку sign in
31     btn_elem = driver.find_element(By.NAME, "commit")
32
33     # Емулюємо клік лівою кнопкою мишки
34     btn_elem.click()
35     time.sleep(3)
36
37     # Закриваємо браузер
38     driver.close()
```

Метод **click** - це операція емуляції натискання лівої кнопки миші на елементі

Синтаксис виклику:
element.click()

◎ Базові Операції - Пошук елементів



```
sbutenko@kbp1-lhp-a00394:~/repos/LnD/Become QA Auto
→ Become QA Auto git:(selenium) X pytest -m ui
===== test session starts =====
platform linux -- Python 3.8.10, pytest-7.2.0, pluggy-0.13.1
rootdir: /home/sbutenko/repos/LnD/Become QA Auto, configfile: pytest.ini
plugins: allure-pytest-2.10.0, rerunfailures-10.2, Faker-14.2.0, forked-1.3.0, a
nyio-3.4.0, xdist-2.5.0, metadata-1.10.0, ordering-0.6, html-3.1.1, instafail-0.
4.2
collected 21 items / 20 deselected / 1 selected

tests/ui/test_ui.py . [100%]

===== 1 passed, 20 deselected in 2.48s =====
→ Become QA Auto git:(selenium) X
```

⦿ Selenium WebDriver - алгоритм тесту



Неуспішний сценарій - користувач помилився при вводі даних



Відкрити браузер



Відкрити сторінку для входу в гітхаб



Ввести неправильне ім'я користувача або поштову адресу



Ввести неправильний пароль



Натиснути кнопку Sign in

Перевірити заголовок сторінки



Закрити браузер

Базові Операції - Пошук елементів



The screenshot shows the GitHub login page in a web browser. A red rectangle highlights the browser's address bar, which contains the URL `github.com/session`. A purple rectangle highlights the error message "Incorrect username or password." and its close button. The login form includes a text input for the username or email address containing `sergiibutenko@mistakeinemail.com`, a password input field, and a green "Sign in" button. Below the form is a link to "Create an account." for new users. At the bottom of the page, there are links for "Terms", "Privacy", "Security", and "Contact GitHub".

Sign in to GitHub - GitHub x

github.com/session

Sign in to GitHub

Incorrect username or password.

Username or email address

sergiibutenko@mistakeinemail.com

Password [Forgot password?](#)

Sign in

New to GitHub? [Create an account.](#)

[Terms](#) [Privacy](#) [Security](#) [Contact GitHub](#)

⦿ Базові Операції - Пошук елементів



```
test_ui.py x
tests > ui > test_ui.py > test_check_incorrect_username
1 import pytest
2
3 from selenium import webdriver
4 from selenium.webdriver.chrome.service import Service
5 from selenium.webdriver.common.by import By
6 import time
7
8 @pytest.mark.ui
9 def test_check_incorrect_username():
10     # Створення об'єкту для керування браузером
11     driver = webdriver.Chrome(
12         service=Service(r"/home/sbutenko/repos/LnD/Become QA Auto/" + "chromedriver")
13     )
14
15     # відкриваємо сторінку https://github.com/login
16     driver.get("https://github.com/login")
17
18     # Знаходимо поле, в яке будемо вводити неправильне ім'я користувача або поштову адресу
19     login_elem = driver.find_element(By.ID, "login_field")
20
21     # Вводимо неправильне ім'я користувача або поштову адресу
22     login_elem.send_keys("sergiibutenko@mistakeinemail.com")
23
24     # Знаходимо поле, в яке будемо вводити неправильний пароль
25     pass_elem = driver.find_element(By.ID, "password")
26
27     # Вводимо неправильний пароль
28     pass_elem.send_keys("wrong password")
29
30     # Знаходимо кнопку sign in
31     btn_elem = driver.find_element(By.NAME, "commit")
32
33     # Емуємо клік лівою кнопкою мишки
34     btn_elem.click()
35
36     # Перевіряємо, що назва сторінки така, яку ми очікуємо
37     assert driver.title == "Sign in to GitHub · GitHub"
38     time.sleep(3)
39
40     # Закриваємо браузер
41     driver.close()
```

Базові Операції - Пошук елементів



```
test_ui.py x
tests > ui > test_ui.py > test_check_incorrect_username
1 import pytest
2
3 from selenium import webdriver
4 from selenium.webdriver.chrome.service import Service
5 from selenium.webdriver.common.by import By
6 import time
7
8 @pytest.mark.ui
9 def test_check_incorrect_username():
10     # Створення об'єкту для керування браузером
11     driver = webdriver.Chrome(
12         service=Service(r"/home/sbutenko/repos/LnD/Become QA Auto/" + "chromedriver")
13     )
14
15     # відкриваємо сторінку https://github.com/login
16     driver.get("https://github.com/login")
17
18     # Знаходимо поле, в яке будемо вводити неправильне ім'я користувача або поштову адресу
19     login_elem = driver.find_element(By.ID, "login_field")
20
21     # Вводимо неправильне ім'я користувача або поштову адресу
22     login_elem.send_keys("sergiibutenko@mistakeinemail.com")
23
24     # Знаходимо поле, в яке будемо вводити неправильний пароль
25     pass_elem = driver.find_element(By.ID, "password")
26
27     # Вводимо неправильний пароль
28     pass_elem.send_keys("wrong password")
29
30     # Знаходимо кнопку sign in
31     btn_elem = driver.find_element(By.NAME, "commit")
32
33     # Емулюємо клік лівою кнопкою мишки
34     btn_elem.click()
35
36     # Перевіряємо, що назва сторінки така, яку ми очікуємо
37     assert driver.title == "Sign in to GitHub · GitHub"
38     time.sleep(3)
39
40     # Закриваємо браузер
41     driver.close()
```

Атрибут **title** - це атрибут, який зберігає в собі заголовок сторінки

Синтаксис виклику:
driver.title

◎ Базові Операції - Пошук елементів



```
sbutenko@kbp1-lhp-a00394:~/repos/LnD/Become QA Auto
→ Become QA Auto git:(selenium) X pytest -m ui
===== test session starts =====
platform linux -- Python 3.8.10, pytest-7.2.0, pluggy-0.13.1
rootdir: /home/sbutenko/repos/LnD/Become QA Auto, configfile: pytest.ini
plugins: allure-pytest-2.10.0, rerunfailures-10.2, Faker-14.2.0, forked-1.3.0, a
nyio-3.4.0, xdist-2.5.0, metadata-1.10.0, ordering-0.6, html-3.1.1, instafail-0.
4.2
collected 21 items / 20 deselected / 1 selected

tests/ui/test_ui.py . [100%]

===== 1 passed, 20 deselected in 2.48s =====
→ Become QA Auto git:(selenium) X
```


⦿ Selenium WebDriver - алгоритм тесту



Неуспішний сценарій - користувач помилився при вводі даних



Відкрити браузер



Відкрити сторінку для входу в гітхаб



Ввести неправильне ім'я користувача або поштову адресу



Ввести неправильний пароль



Натиснути кнопку Sign in



Перевірити заголовок сторінки



Закрити браузер



Підсумки



- ✓ Метод **get()** - це операція відкриття сторінки в браузері.
- ✓ Синтаксис виклику методу **get()**:
 - **driver.get("URL-адреса, яку ви бажаєте відкрити")**
- ✓ Метод **close()** - це операція закриття браузера
- ✓ Синтаксис виклику методу **close()**:
 - **driver.close()**



Підсумки



- ✓ Метод **find_element()** - це операція пошуку потрібного елемента за іменем атрибуту та значенням цього атрибуту
- ✓ Синтаксис виклику методу **find_element()**:
driver.find_element(ІМ'Я АТРИБУТА, ЗНАЧЕННЯ АТРИБУТА)
- ✓ Метод **send_keys()** - це операція введення тексту до елемента сторінки
- ✓ Синтаксис виклику методу **send_keys()**:
element.send_keys(ТЕКСТ, ЯКИЙ ВИ БАЖАЄТЕ ВВЕСТИ)



Підсумки



- ✓ Метод **click()** - це операція емуляції натискання лівої кнопки миші для елемента
- ✓ Синтаксис виклику методу **click()**:
element.click()
- ✓ Атрибут **title** - це атрибут, який зберігає в собі заголовок сторінки
- ✓ Синтаксис виклику атрибуту **title**:
driver.title