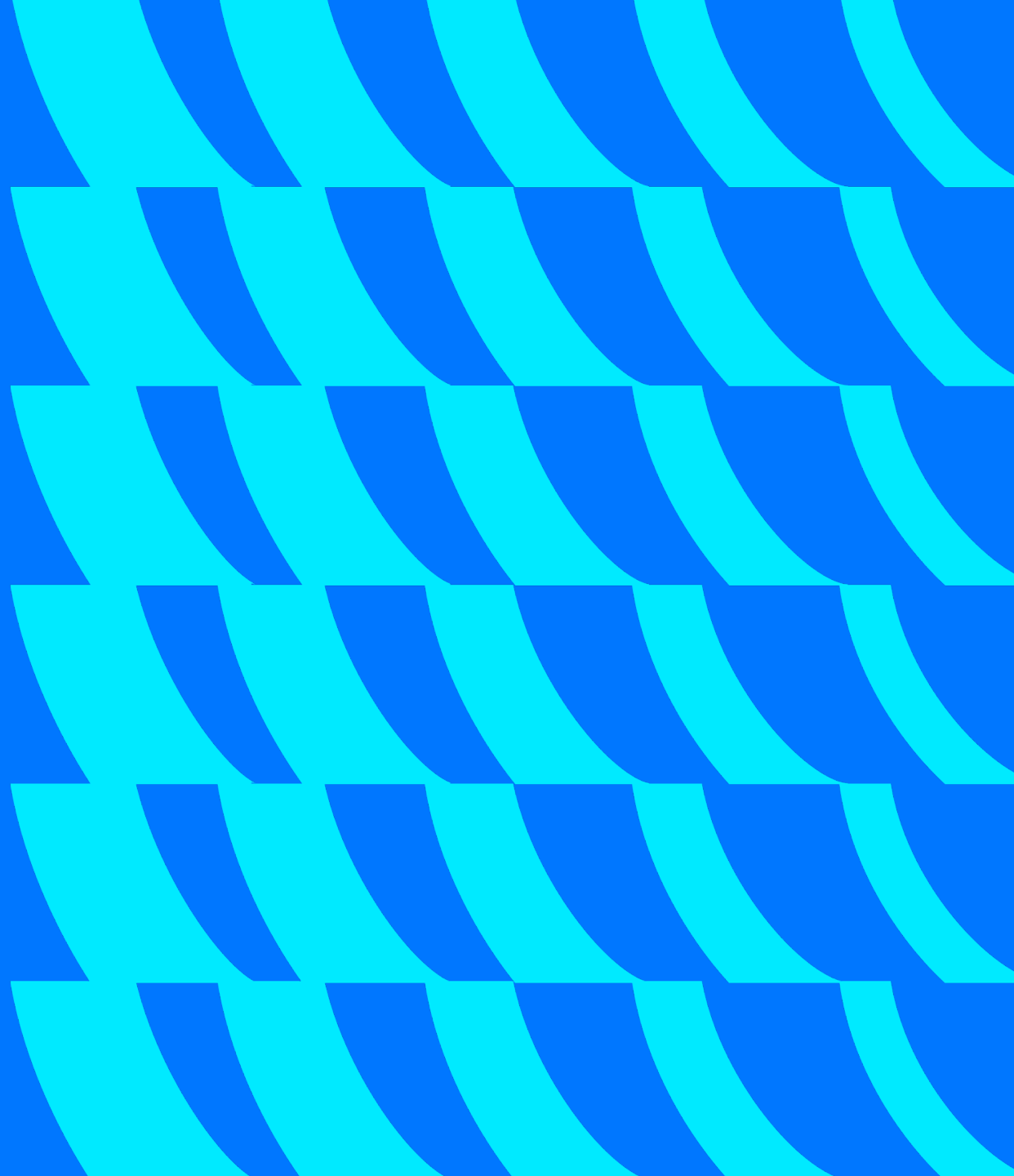


Обеспечение качества



Константин Ермаков

Не забудьте
отметиться
на портале



Тестирование UI: Selenium

- Selenium WebDriver
- Локаторы, атрибуты
- Взаимодействие со страницей
- Chrome Options
- Альтернативы

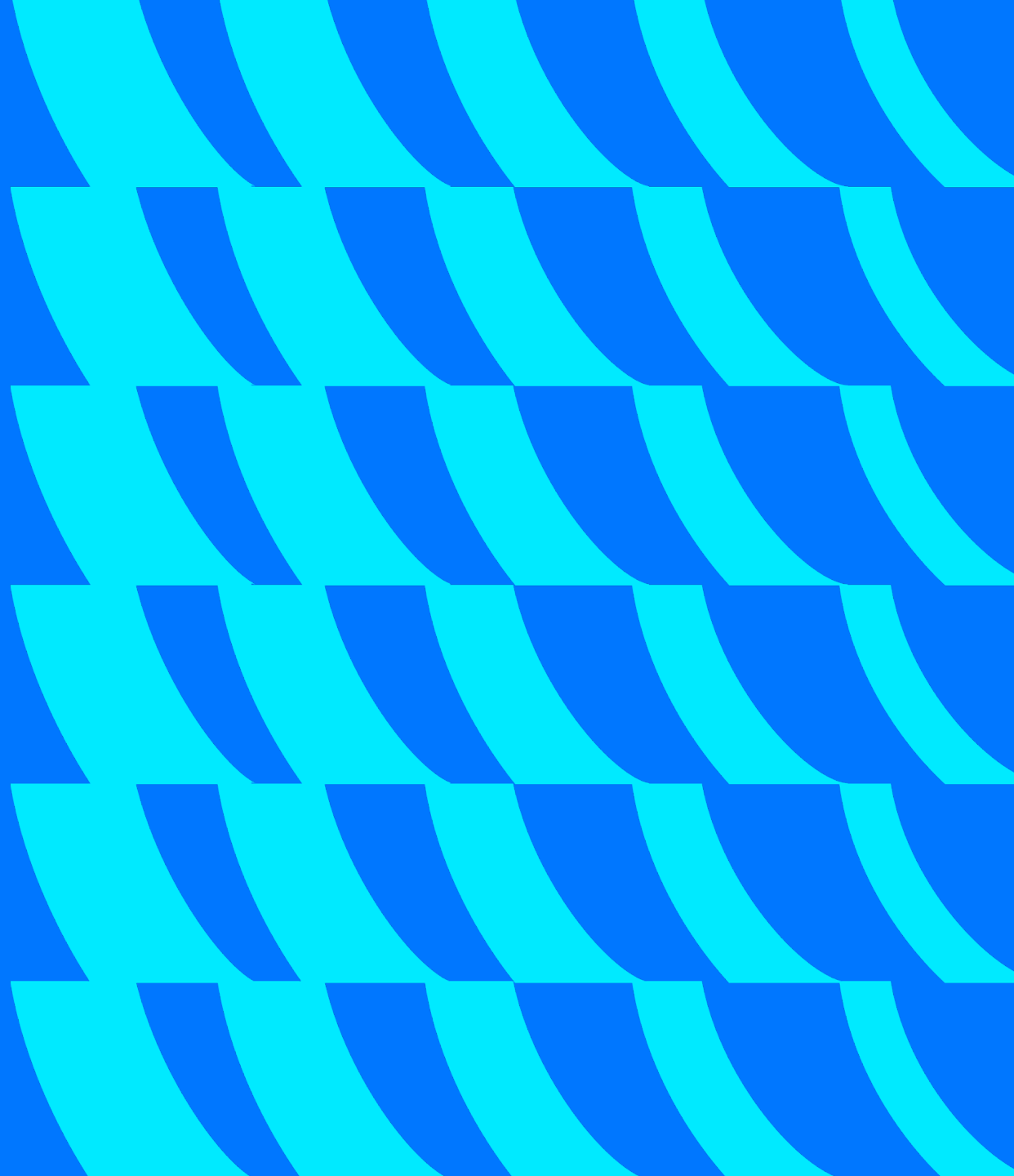
Selenium

Что такое Selenium?

Selenium – проект нескольких продуктов с открытым исходным кодом для работы с браузерами:

- Selenium WebDriver
- Selenium RC
- Selenium Server
- Selenium Grid
- Selenium IDE

Selenium WebDriver



Что такое Selenium WebDriver?

Selenium WebDriver – набор программных библиотек для управления браузерами



Selenium WebDriver



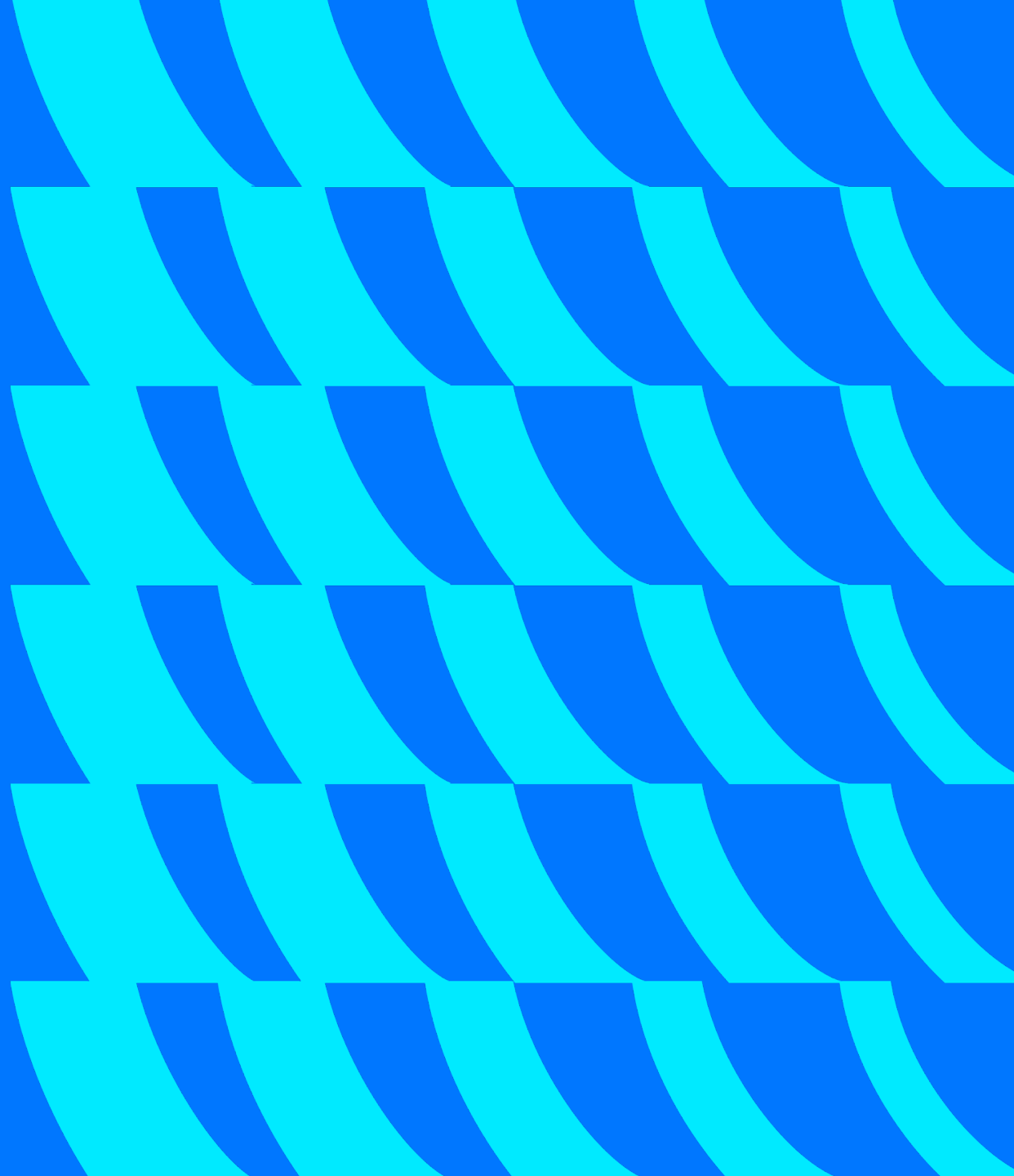
Selenium WebDriver

```
from selenium import webdriver

driver = webdriver.Chrome()
driver.get("http://www.google.com")
assert "Google" in driver.title
assert "Yandex" not in driver.page_source

driver.close()
```

Selenium WebDriver: локаторы



Selenium WebDriver: локаторы By CSS id

```
<html>
  <body>
    <form id="loginForm">
      <input name="username" type="text" />
      <input name="password" type="password" />
      <input name="continue" type="submit" value="Login" />
    </form>
  </body>
</html>
```

```
login_form = driver.find_element_by_id("loginForm")
```

Selenium WebDriver: локаторы By class name

```
<html>
  <body>
    <p class="content">Site p content goes here.</p>
    <a class="content">Site a content goes here.</a>
  </body>
</html>
```



```
content = driver.find_element_by_class_name('content')
content = driver.find_elements_by_class_name('content')
```

Selenium WebDriver: локаторы By name attribute

```
<html>
  <body>
    <form id="loginForm">
      <input name="username" type="text" />
      <input name="password" type="password" />
      <input name="continue" type="submit" value="Login" />
      <input name="continue" type="button" value="Clear" />
    </form>
  </body>
</html>
```

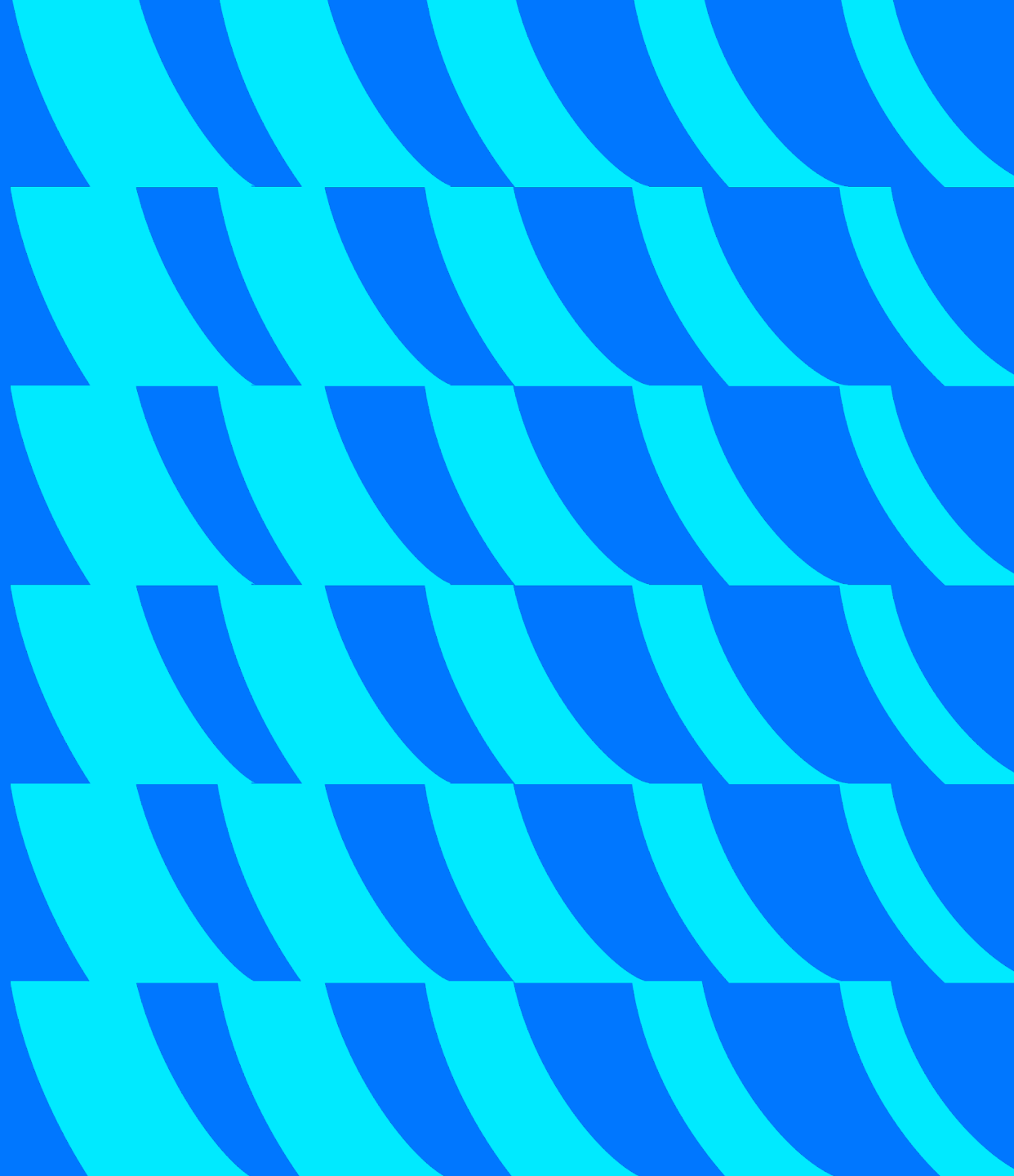
```
username = driver.find_element_by_name('username')
password = driver.find_element_by_name('password')
```

Selenium WebDriver: локаторы By XPath

```
<html>
  <body>
    <form id="loginForm">
      <input name="username" type="text" />
      <input name="password" type="password" />
      <input name="continue" type="submit" value="Login" />
      <input name="continue" type="button" value="Clear" />
    </form>
  </body>
</html>
```

```
login_form = driver.find_element_by_xpath("/html/body/form[1]")
login_form = driver.find_element_by_xpath("//form[1]")
login_form = driver.find_element_by_xpath("//form[@id='loginForm']")
```

Selenium: взаимодействие со страницей



Selenium WebDriver: взаимодействие со страницей

```
<input type="text" name="passwd" id="passwd-id">Input password here</input>
```

```
element = driver.find_element_by_id("passwd-id")
element = driver.find_element_by_name("passwd")
element = driver.find_element_by_xpath("//input[@id='passwd-id']")
element = driver.find_element_by_xpath("//input[contains(text(), 'Input password')]")
```

```
>>> element
<selenium.webdriver.remote.webelement.WebElement>
  id: '4e4387a1-6987-4050-a296-a1f395ca6d81'
  location: {'x': 1042, 'y': 90}
  rect: {'height': 37, 'width': 224, 'x': 1041.859375, 'y': 89.90625}
  size: {'height': 37, 'width': 224}
  tag_name: 'input'
  text: 'text'
```

```
>>> dir(element)
is_displayed
send_keys
clear
click
```


Selenium WebDriver: взаимодействие со страницей

```
<input type="text" name="passwd" id="passwd-id" />
```

```
from selenium.webdriver.common.keys import Keys
```

```
element = driver.find_element_by_id("passwd-id")  
element.clear()  
element.send_keys("MyPassword")  
element.send_keys(Keys.RETURN)
```

Selenium WebDriver: pytest example

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys

def test():
    driver = webdriver.Chrome()
    driver.get("http://www.python.org")
    assert "Python" in driver.title

    elem = driver.find_element_by_name("q")
    elem.send_keys("pycon")
    elem.send_keys(Keys.RETURN)
    assert "No results found." not in driver.page_source

    driver.close()
```

Атрибуты

Атрибуты

```
element = driver.find_element_by_id("some_id")  
element.get_attribute('display')
```

Chrome Options

Chrome Options

```
options = webdriver.ChromeOptions()  
options.add_argument("--window-size=800,600")  
options.set_headless()
```

Альтернативы

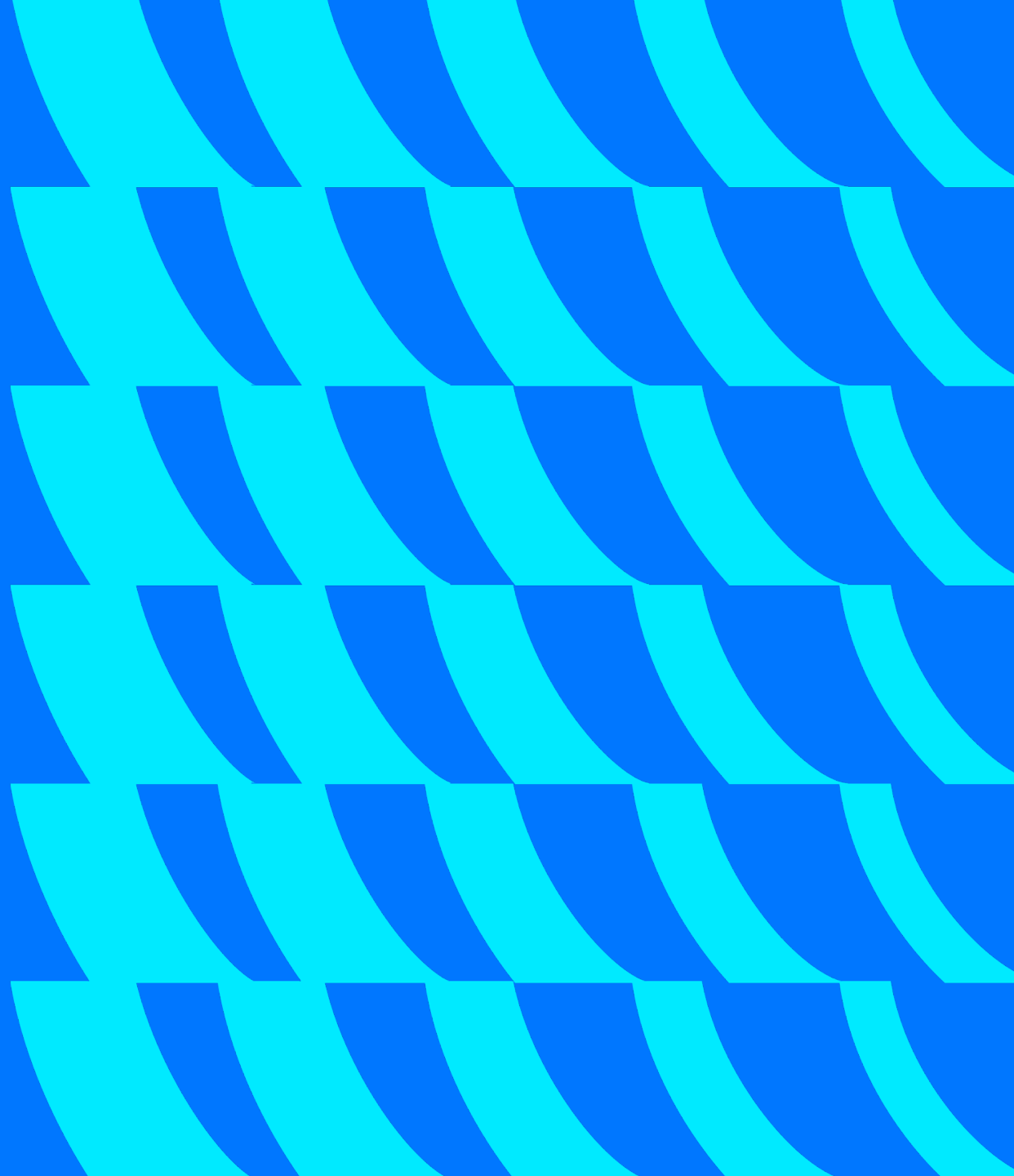
Альтернативы

- TestCafe
- Cypress
- Playwright
- Watir
- Sahi

Тестирование UI: Page Object

- Суть паттерна
- В чем преимущество
- Примеры
- Преобразуем код к новому виду

Page Object:
Суть
паттерна



Что такое Page Object?

Это популярный паттерн, который является по сути стандартом в автоматизации тестирования UI веб-продуктов.

Основная идея состоит в том, чтобы разделить логику тестов от реализации.

Page Object:
В чем
преимущество

Преимущества Page Object

- Разделение кода тестов и их реализации
- Уменьшение дублирования кода
- Увеличение переиспользуемости кода
- Увеличение читаемости кода

Page Object: Примеры

Selenium WebDriver: pytest example

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys

def test():
    driver = webdriver.Chrome()
    driver.get("http://www.python.org")
    assert "Python" in driver.title

    elem = driver.find_element_by_name("q")
    elem.send_keys("pycon")
    elem.send_keys(Keys.RETURN)
    assert "No results found." not in driver.page_source

    driver.close()
```

Selenium WebDriver: pytest example with PO

```
import pytest
from base_tests.base import BaseCase

class TestOne(BaseCase):
    @pytest.mark.parametrize(
        'query', [
            pytest.param(
                'pycon'
            ),
            pytest.param(
                'python'
            ),
        ])
    def test_search(self, query):
        self.base_page.search(query)
        assert "No results found." not in self.driver.page_source
```


Selenium WebDriver: pytest example Base Page

```
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.support.wait import WebDriverWait
from ui.locators.pages_locators import BasePageLocators

CLICK_RETRY = 3

class BasePage(object):
    locators = BasePageLocators()

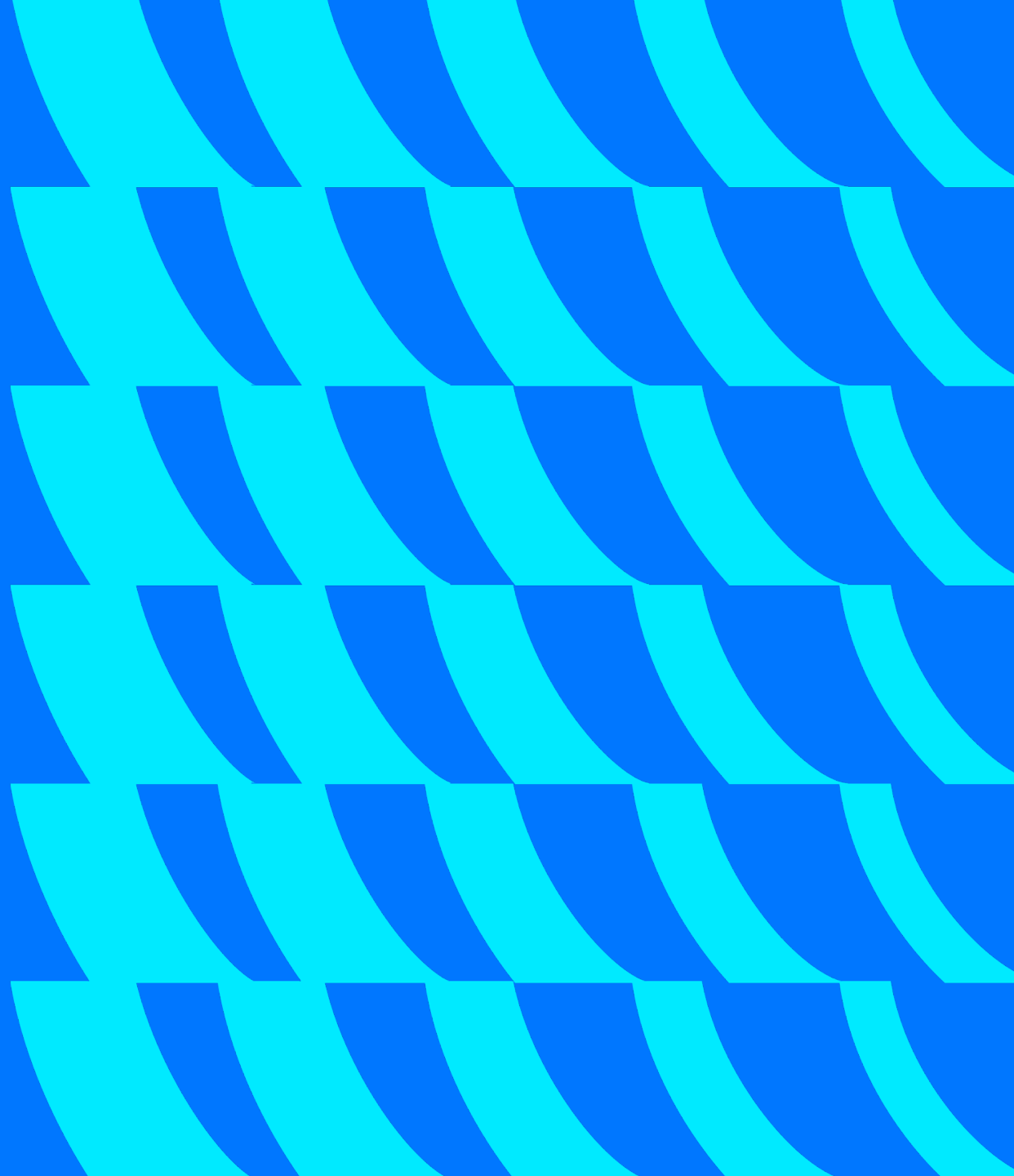
    def __init__(self, driver):
        self.driver = driver

    def find(self, locator, timeout=None):
        return self.wait(timeout).until(EC.presence_of_element_located(locator))

    def wait(self, timeout=None):
        if timeout is None:
            timeout = 5
        return WebDriverWait(self.driver, timeout=timeout)

    def search(self, query):
        search = self.find(self.locators.QUERY_LOCATOR)
        search.clear()
        search.send_keys(query)
        self.click(self.locators.GO_LOCATOR)
```

Не забудьте
оставить
ОТЗЫВ



Спасибо
за внимание