

Selenium WebDriverで学ぶ システムテスト自動化の第一歩

伊藤 望
テスト自動化研究会(STAR)

システムテスト実行の自動化

- GUI(画面)自動テストツール
- 画面操作を自動化し、テスト作業を効率化!
- Selenium, QTP, UWSC, など様々なツールがある



- ブラウザ・モバイルのテストツール
- オープンソース

今日学ぶこと

Seleniumの基本的な使い方

いろいろなSelenium ①

- Selenium IDE
 - ブラウザ操作の記録と再生



いろいろなSelenium ②

- Selenium WebDriver
 - プログラミング言語のコードから実行



効率よくテストをメンテナンスするなら

- Selenium IDE
 - 手軽にテストを作れる
- Selenium WebDriver
 - 長期にわたってメンテナンスし続けるならこちら
- 今回は「Selenium WebDriver」について学びます

タイムテーブル

Selenium WebDriverの使い方

1. 入門課題	60分
2. 実践課題	30分

Selenium WebDriverの使い方

1. 入門課題 (60分)

入門課題

- Selenium WebDriverの基礎を学びます
- 5分程度のミニ課題 × 5
- 必要なもの
 - Eclipse
 - Google Chrome
 - 課題プログラムインストールキット

入門課題その1 「動かしてみよう、Selenium」

1. Eclipseを起動します
2. test/introwork/IntroWork1.javaを開いてください

入門課題その1

「IntroWork1.java」を実行し、
成功することを確認してください

□ 手順

1. test/introwork/IntroWork1.javaを右クリックし、「実行」>「JUnitテスト」を選びます
2. テストが実行され、結果が緑になれば成功です

入門課題その1 解説 JUnit

- テストの実行には、テストフレームワーク「JUnit」を使っています
- @Before
 - 初期処理
- @Test
 - メインとなるテスト処理
- @After
 - 終了処理

入門課題その1 解説 @Before

```
@Before
public void setUp() {
    // chromedriverのインストール場所を指定
    System.setProperty(
        "webdriver.chrome.driver",
        chromeDriverPath());

    // WebDriverのインスタンスを生成しブラウザを起動
    driver = new ChromeDriver();
}
```

入門課題その1 解説

@Test

```
@Test
public void test() {
    .....

    // 指定したURLのウェブページに移動
    driver.get(url);

    // 文字列入力・クリックなどの処理
    .....
}
```

入門課題その1 解説

@After

```
@After
public void tearDown() {
    // ブラウザを閉じ、WebDriverを終了する
    driver.quit();
}
```

入門課題その2 「クリックしてみよう」

1. test/introwork/IntroWork2.javaを右クリックし、「実行」>「JUnitテスト」を選びます
2. 「OK」ボタンが置かれたページが表示されます

入門課題その2

「OK」ボタンをクリックする処理を、IntroWork2.javaに実装してください

入門課題その2 「クリックしてみよう」

- 「OK」ボタンのidを調べます
 1. introWork/introWork2.htmlを、Google Chromeから直接開きます
 2. 「OK」ボタンを右クリックし「要素の検証」を選びます
- Sleep処理を消して、クリック操作を記述します

```
WebElement okButton
    = driver.findElement(By.id("要素のid"));
okButton.click();
```

- 書けたら実行してみます

入門課題その2 「クリックしてみよう」

- 動きが速すぎて、クリックできたか分からない時は
 1. driver.quit()にブレークポイントを置きます
 2. IntroWork2.javaを右クリックし、「デバッグ」>「JUnitテスト」からテストを実行します
 3. ブレークポイントでテストが一時停止するので、クリックできたか確認できます
 4. 「F8」キーで実行を再開します

入門課題その2 解答例

```
@Test
public void test() {
    .....
    driver.get(url);

    WebElement okButton
        = driver.findElement(By.id("ok_button"));
    okButton.click();
}
```

入門課題その3 「文字列を入力してみよう」

1. IntroWork3.javaをJUnitテストとして実行します
2. テキスト入力欄が置かれたページが表示されます

入門課題その3 (5分)

テキスト入力欄の「Test」という文字列を消して、代わりに「Selenium」という文字列を入力する処理を、IntroWork3.javaに実装してください

入門課題その3 「文字列を入力してみよう」

□ ヒント

- clearメソッドで、入力欄を一度空にする
- sendKeysメソッドで、文字列「Selenium」を入力する

```
WebElement input
    = driver.findElement(By.id("要素のid"));
input.clear();
input.sendKeys("文字列");
```

入門課題その3 解答例

```
@Test
public void test() {
    .....
    driver.get(url);

    WebElement subject
        = driver.findElement(By.id("subject"));
    subject.clear();
    subject.sendKeys("Selenium");
}
```

入門課題その4 「チェックボックスを選択してみよう」

- IntroWork4.javaをJUnitテストとして実行すると、課題ページが表示されます

入門課題その4 (5分)

チェックボックスのチェックをオンにする処理を、IntroWork4.javaに実装してください

入門課題その4 「チェックボックスを選択してみよう」

□ ヒント

- チェックボックスのチェックの切り替えは「click」で行います
- 既にチェック状態なら、チェックを切り替えないようにします

```
if (!element.isSelected()) {
    element.click();
}
```

入門課題その4 解答例

```
@Test
public void test() {
    .....
    driver.get(url);

    WebElement allowedCheck = driver.findElement(
        By.id("allowed_check"));
    if (!allowedCheck.isSelected()) {
        allowedCheck.click();
    }
}
```

入門課題その5 「表示された値のチェックをしてみよう」

- IntroWork5.javaをJUnitテストとして実行すると、課題ページが表示されます

入門課題その5 (5分)

表示された金額の値が「9000」であることをチェックする処理を、IntroWork5.javaに実装してください

入門課題5 「表示された値のチェックをしてみよう」

□ ヒント

- getTextにより表示されているテキストを取得
- JUnitのAssertThatメソッドを使って、値が9000であることをチェック

```
import static org.junit.Assert.*;
import static org.hamcrest.core.Is.*;

.....

WebElement total
    = driver.findElement(By.id("要素のid"));
assertThat(total.getText(), is("値"));
```

入門課題その5 解答例

```
import static org.junit.Assert.*;
import static org.hamcrest.core.Is.*;

.....
@Test
public void test() {
    .....
    driver.get(url);

    WebElement total
        = driver.findElement(By.id("total"));
    assertThat(total.getText(), is("9000"));
}
```

入門課題で学んだこと

- クリック
- 文字列入力
- チェックボックス
- 値チェック

Selenium WebDriverの使い方

2. 実践課題 (30分)

実践課題

- test/practicework/PracticeWork1.javaをJUnitテストとして実行すると、「STARホテル宿泊予約画面」が表示されます

実践課題 (30分)

docs/TestCase.pdfの「実践課題」テストケースを、PracticeWork1.javaに実装してください。

- 予約処理の自動化
- 確認画面の値チェックの自動化

□ ヒント

- reserveApp/index.htmlがHTMLファイルです。

実践課題 解答例

- `answer/practicework/work1/PracticeWork1.java`

まとめ

- 今日学んだこと
 - Selenium WebDriverの基本的な使い方
- 今回取り上げなかった話題
 - ページオブジェクトデザインパターン
 - Selenium IDEを使ったスクリプト生成

お疲れさまでした
