Assertions in Playwright

What is an Assertion?

Assertions are used to perform validations on a webpage. They help verify that elements, attributes, and other expected behaviours are correctly displayed or functioning as intended.

Types of Assertions

Hard Assertion

- If an assertion fails, the test execution stops immediately.
- Playwright's built-in assertions are **hard assertions**.

  Example:

  ```
  await expect(page).toHaveUrl('https://example.com');
  await expect(page).toHaveTitle('SKY'); // If this fails, the test
  will stop here
  await expect(await page.locator('header')).toBeVisible(); // This
  will not execute
  ```

Soft Assertion

- Even if an assertion fails, the test continues execution.
- Soft assertions do not terminate the test execution but mark it as failed.

  Example:

  ```
  await expect.soft(page).toHaveUrl('https://example.com');
  await expect.soft(page).toHaveTitle('SKY'); // Test continues even if
  this fails
  await expect.soft(await page.locator('header')).toBeVisible();
  ```

By using soft assertions, we can verify multiple conditions without terminating the test execution early.

Example DOM

```html
<html>
<head>
  <title>NBCU Page Title</title>
</head>
<body>
  <header class="pageHeader">
    <h4>NBCUniversal</h4>
  </header>
  <img class="header-logo" src="logo.png" alt="Logo">
  <input type="text" id="username" value="Yogesh">
  <input type="checkbox" id="checkBox" checked>
  <input type="radio" id="radioMr" name="gender" checked>
  <input type="text" id="searchbox">
  <select class="dropdown">
    <option>Option 1</option>
    <option>Option 2</option>
    <option>Option 3</option>
    <option>Option 4</option>
    <option>Option 5</option>
    <option>Option 6</option>
    <option>Option 7</option>
    <option>Option 8</option>
  </select>
</body>
</html>
```

Frequently Used Assertions in Playwright

1. **toHaveUrl()** – Checks whether the page has the correct URL.

```
await expect(page).toHaveUrl('https://example.com');
```

2. **toHaveTitle()** – Checks whether the page has the correct title.

```
await expect(page).toHaveTitle('NBCU Page Title');
```

3. **toHaveAttribute()** – Checks whether an element in the DOM has a specific attribute.

```
const usernameField = await page.locator('#username');
await expect(usernameField).toHaveAttribute('type', 'text');
```

4. **toHaveValue()** – Checks whether a text field contains a specific value.

```
const usernameField = await page.locator('input#username');
await usernameField.fill('Yogesh');
await expect(usernameField).toHaveValue('Yogesh');
```

5. **toHaveCount()** – Checks the count of elements, such as the number of options in a dropdown.

```
const dropdownOptions = await page.locator('select.dropdown option');
await expect(dropdownOptions).toHaveCount(8);
```

6. **toHaveText()** – Checks whether an element has the exact matching text.

```
const pageHeader = await page.locator('.pageHeader h4');
await expect(pageHeader).toHaveText('NBCUniversal');
```

7. **toContainText()** – Checks whether an element contains the specified text.

```
const pageHeader = await page.locator('.pageHeader h4');
await expect(pageHeader).toContainText('NBCU');
```

8. **toBeVisible()** – Checks whether an element is visible on the page.

```
const logoElement = await page.locator('.header-logo');
await expect(logoElement).toBeVisible();
```

9. **toBeEnabled()** – Checks whether an input field or button is enabled.

```
const searchBox = await page.locator('#searchbox');
await expect(searchBox).toBeEnabled();
```

10. **toBeChecked()** – Checks whether a checkbox or radio button is checked.

```
const radioMr = await page.locator('#radioMr');
await radioMr.click();
await expect(radioMr).toBeChecked();
```

```
    const checkBox = await page.locator('#checkBox');
    await expect(checkBox).toBeChecked();
```

Negative Assertions

Playwright allows us to perform negative assertions using `.not`.

```
await expect(page).not.toHaveUrl('https://example.com');
```

Summary

Assertions in Playwright help validate various aspects of a webpage, such as URLs, titles, text, visibility, and attributes. Playwright provides both **hard assertions**, which stop test execution on failure, and **soft assertions**, which allow the test to continue even if a failure occurs. Using assertions correctly ensures robust and reliable automated testing.