Playwright Locators

In Playwright, **locators** are one of the most important concepts.
They provide a **stable and reliable way to find elements on a webpage**.

Why important?

- Locators in Playwright **auto-wait** for elements to be ready.
- They also **retry automatically** if the element is not immediately available.
- This makes your tests less flaky compared to plain CSS or XPath selectors.

Common Built-in Locators

Locator Method	Use Case
page.get_by_role()	Locate elements by accessibility roles like button, checkbox, heading, etc.
page.get_by_text()	Locate by visible text content.
page.get_by_label()	Locate form controls using associated label text.
page.get_by_placeholder()	Locate inputs via placeholder text.
page.get_by_alt_text()	Locate images by their alt attribute.
page.get_by_title()	Locate elements by their title attribute.
page.get_by_test_id()	Locate by custom attribute like data-testid.

Locate by Role

get_by_role() finds elements by how users/assistive technologies see them (e.g., buttons, checkboxes, headings).

Example DOM:

```
<h3>Sign up</h3>

<label><input type="checkbox" /> Subscribe</label><br/>

<button>Submit</button>
```

Test:

```
from playwright.sync_api import Page, expect


def test_locate_by_role(page: Page):
    page.goto("http://example.com")


    expect(page.get_by_role("heading", name="Sign up")).to_be_visible()
    page.get_by_role("checkbox", name="Subscribe").check()
    page.get_by_role("button", name="Submit").click()
```

✅ Use when: Working with **interactive elements** like buttons, checkboxes, links, tables, etc.

Locate by Label

get_by_label() is best for form fields linked with a <label>.

Example DOM:

```
<label>Password <input type="password" /></label>
```

Test:

```
def test_locate_by_label(page: Page):
    page.goto("http://example.com")
    page.get_by_label("Password").fill("secret123")
```

Locate by Placeholder

get_by_placeholder() finds input fields using placeholder text.

Example DOM:

```
<input type="email" placeholder="name@example.com" />
```

Test:

```
def test_locate_by_placeholder(page: Page):
    page.goto("http://example.com")
    page.get_by_placeholder("name@example.com").fill("playwright@test.com")
```

Locate by Text

get_by_text() searches by visible text content.

Example DOM:

```
<span>Welcome, John</span>
```

Test:

```
def test_locate_by_text(page: Page):
   page.goto("http://example.com")

   # Partial match
   expect(page.get_by_text("Welcome")).to_be_visible()

   # Exact match
   expect(page.get_by_text("Welcome, John", exact=True)).to_be_visible()
```

✅ Use for **non-interactive elements** like <div>, <span>, <p>.
(For clickable elements, prefer role-based locators).

Locate by Alt Text

get_by_alt_text() finds images by their alt attribute.

Example DOM:

```
<img alt="playwright logo" src="/img/playwright-logo.svg" width="100" />
```

Test:

```
def test_locate_by_alt_text(page: Page):
   page.goto("http://example.com")
   page.get_by_alt_text("playwright logo").click()
```

Locate by Title

get_by_title() finds elements using the title attribute.

Example DOM:

```
<span title="Issues count">25 issues</span>
```

Test:

```
def test_locate_by_title(page: Page):
   page.goto("http://example.com")
   expect(page.get_by_title("Issues count")).to_have_text("25 issues")
```

Locate by Test ID

get_by_test_id() works with attributes like data-testid.

Example DOM:

```
<button data-testid="directions">Directions</button>
```

Test:

```
def test_locate_by_testid(page: Page):
    page.goto("http://example.com")
    page.get_by_test_id("directions").click()
```

✅ Use when:

- Text or role locators are not reliable.