



Assertions

List of assertions

Assertion	Description
<code>expect(locator).to_be_attached()</code>	Element is attached
<code>expect(locator).to_be_checked()</code>	Checkbox is checked
<code>expect(locator).to_be_disabled()</code>	Element is disabled
<code>expect(locator).to_be_editable()</code>	Element is editable
<code>expect(locator).to_be_empty()</code>	Container is empty
<code>expect(locator).to_be_enabled()</code>	Element is enabled
<code>expect(locator).to_be_focused()</code>	Element is focused
<code>expect(locator).to_be_hidden()</code>	Element is not visible
<code>expect(locator).to_be_in_viewport()</code>	Element intersects viewport
<code>expect(locator).to_be_visible()</code>	Element is visible
<code>expect(locator).to_contain_text()</code>	Element contains text
<code>expect(locator).to_have_attribute()</code>	Element has a DOM attribute
<code>expect(locator).to_have_class()</code>	Element has a class property
<code>expect(locator).to_have_count()</code>	List has exact number of children

Assertion	Description
<code>expect(locator).to_have_css()</code>	Element has CSS property
<code>expect(locator).to_have_id()</code>	Element has an ID
<code>expect(locator).to_have_js_property()</code>	Element has a JavaScript property
<code>expect(locator).to_have_text()</code>	Element matches text
<code>expect(locator).to_have_value()</code>	Input has a value
<code>expect(locator).to_have_values()</code>	Select has options selected
<code>expect(page).to_have_title()</code>	Page has a title
<code>expect(page).to_have_url()</code>	Page has a URL
<code>expect(response).to_be_ok()</code>	Response has an OK status

Custom Expect Message

You can specify a custom expect message as a second argument to the `expect` function, for example:

```
expect(page.get_by_text("Name"), "should be logged in").to_be_visible()
```

When expect fails, the error would look like this:

```
def test_foobar(page: Page) -> None:
>     expect(page.get_by_text("Name"), "should be logged in").to_be_visible()
E     AssertionError: should be logged in
E     Actual value: None
E     Call log:
E     LocatorAssertions.to_be_visible with timeout 5000ms
E     waiting for get_by_text("Name")
```

```
E      waiting for get_by_text("Name")
```

```
tests/test_foobar.py:22: AssertionError
```

Setting a custom timeout

You can specify a custom timeout for assertions either globally or per assertion. The default timeout is 5 seconds.

Global timeout

```
conftest.py
```

```
from playwright.sync_api import expect
```

```
expect.set_options(timeout=10_000)
```

Per assertion timeout

```
test_foobar.py
```

```
from playwright.sync_api import expect
```

```
def test_foobar(page: Page) -> None:  
    expect(page.get_by_text("Name")).to_be_visible(timeout=10_000)
```