

# LocatorAssertions

The [LocatorAssertions](#) class provides assertion methods that can be used to make assertions about the [Locator](#) state in the tests.

[Sync](#) [Async](#)

```
from playwright.sync_api import Page, expect

def test_status_becomes_submitted(page: Page) -> None:
    # ..
    page.get_by_role("button").click()
    expect(page.locator(".status")).to_have_text("Submitted")
```

## Methods

### not\_to\_be\_attached

Added in: v1.33

The opposite of [expect\(locator\).to\\_be\\_attached\(\)](#).

#### Usage

```
expect(locator).not_to_be_attached()
expect(locator).not_to_be_attached(**kwargs)
```

#### Arguments

- `attached` [bool](#) (*optional*)
- `timeout` [float](#) (*optional*)

Time to retry the assertion for in milliseconds. Defaults to `5000`.

# not\_to\_be\_checked

Added in: v1.20

The opposite of `expect(locator).to_be_checked()`.

## Usage

```
expect(locator).not_to_be_checked()  
expect(locator).not_to_be_checked(**kwargs)
```

## Arguments

- `timeout` `float` (*optional*) Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

# not\_to\_be\_disabled

Added in: v1.20

The opposite of `expect(locator).to_be_disabled()`.

## Usage

```
expect(locator).not_to_be_disabled()  
expect(locator).not_to_be_disabled(**kwargs)
```

## Arguments

- `timeout` `float` (*optional*) Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

# not\_to\_be\_editable

Added in: v1.20

The opposite of `expect(locator).to_be_editable()`.

## Usage

```
expect(locator).not_to_be_editable()  
expect(locator).not_to_be_editable(**kwargs)
```

## Arguments

- `editable` **bool** (*optional*) Added in: v1.26
- `timeout` **float** (*optional*) Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

## not\_to\_be\_empty

Added in: v1.20

The opposite of `expect(locator).to_be_empty()`.

## Usage

```
expect(locator).not_to_be_empty()  
expect(locator).not_to_be_empty(**kwargs)
```

## Arguments

- `timeout` **float** (*optional*) Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

## not\_to\_be\_enabled

Added in: v1.20

The opposite of `expect(locator).to_be_enabled()`.

## Usage

```
expect(locator).not_to_be_enabled()  
expect(locator).not_to_be_enabled(**kwargs)
```

## Arguments

- `enabled` `bool` (*optional*) Added in: v1.26
- `timeout` `float` (*optional*) Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

## not\_to\_be\_focused

Added in: v1.20

The opposite of `expect(locator).to_be_focused()`.

### Usage

```
expect(locator).not_to_be_focused()  
expect(locator).not_to_be_focused(**kwargs)
```

### Arguments

- `timeout` `float` (*optional*) Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

## not\_to\_be\_hidden

Added in: v1.20

The opposite of `expect(locator).to_be_hidden()`.

### Usage

```
expect(locator).not_to_be_hidden()  
expect(locator).not_to_be_hidden(**kwargs)
```

### Arguments

- `timeout` `float` (*optional*) Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

# not\_to\_be\_in\_viewport

Added in: v1.31

The opposite of `expect(locator).to_be_in_viewport()`.

## Usage

```
expect(locator).not_to_be_in_viewport()  
expect(locator).not_to_be_in_viewport(**kwargs)
```

## Arguments

- `ratio` `float` (*optional*)
- `timeout` `float` (*optional*)

Time to retry the assertion for in milliseconds. Defaults to `5000`.

# not\_to\_be\_visible

Added in: v1.20

The opposite of `expect(locator).to_be_visible()`.

## Usage

```
expect(locator).not_to_be_visible()  
expect(locator).not_to_be_visible(**kwargs)
```

## Arguments

- `timeout` `float` (*optional*) Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

- `visible` `bool` (*optional*) Added in: v1.26

# not\_to\_contain\_text

Added in: v1.20

The opposite of `expect(locator).to_contain_text()`.

## Usage

```
expect(locator).not_to_contain_text(expected)
expect(locator).not_to_contain_text(expected, **kwargs)
```

## Arguments

- `expected` `str|Pattern|List[str]|List[Pattern]|List[str|Pattern]` Added in: v1.18

Expected substring or RegExp or a list of those.

- `ignore_case` `bool (optional)` Added in: v1.23

Whether to perform case-insensitive match. `ignore_case` option takes precedence over the corresponding regular expression flag if specified.

- `timeout` `float (optional)` Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

- `use_inner_text` `bool (optional)` Added in: v1.18

Whether to use `element.innerText` instead of `element.textContent` when retrieving DOM node text.

## not\_to\_have\_attribute

Added in: v1.20

The opposite of `expect(locator).to_have_attribute()`.

## Usage

```
expect(locator).not_to_have_attribute(name, value)
expect(locator).not_to_have_attribute(name, value, **kwargs)
```

## Arguments

- `name` `str` Added in: v1.18

Attribute name.

- `value` `str|Pattern` Added in: v1.18

Expected attribute value.

- `timeout` `float` (*optional*) Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

## not\_to\_have\_class

Added in: v1.20

The opposite of `expect(locator).to_have_class()`.

### Usage

```
expect(locator).not_to_have_class(expected)
expect(locator).not_to_have_class(expected, **kwargs)
```

### Arguments

- `expected` `str|Pattern|List[str]|List[Pattern]|List[str|Pattern]` Added in: v1.18

Expected class or RegExp or a list of those.

- `timeout` `float` (*optional*) Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

## not\_to\_have\_count

Added in: v1.20

The opposite of `expect(locator).to_have_count()`.

### Usage

```
expect(locator).not_to_have_count(count)
expect(locator).not_to_have_count(count, **kwargs)
```

## Arguments

- `count` `int` Added in: v1.18

Expected count.

- `timeout` `float` (*optional*) Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

## not\_to\_have\_css

Added in: v1.20

The opposite of `expect(locator).to_have_css()`.

### Usage

```
expect(locator).not_to_have_css(name, value)
expect(locator).not_to_have_css(name, value, **kwargs)
```

## Arguments

- `name` `str` Added in: v1.18

CSS property name.

- `value` `str`|`Pattern` Added in: v1.18

CSS property value.

- `timeout` `float` (*optional*) Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

## not\_to\_have\_id

Added in: v1.20

The opposite of `expect(locator).to_have_id()`.

### Usage



```
expect(locator).not_to_have_id(id)
expect(locator).not_to_have_id(id, **kwargs)
```

## Arguments

- `id` `str`|`Pattern` Added in: v1.18

Element id.

- `timeout` `float` (*optional*) Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

## not\_to\_have\_js\_property

Added in: v1.20

The opposite of `expect(locator).to_have_js_property()`.

## Usage

```
expect(locator).not_to_have_js_property(name, value)
expect(locator).not_to_have_js_property(name, value, **kwargs)
```

## Arguments

- `name` `str` Added in: v1.18

Property name.

- `value` `Any` Added in: v1.18

Property value.

- `timeout` `float` (*optional*) Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

## not\_to\_have\_text

Added in: v1.20

The opposite of `expect(locator).to_have_text()`.

## Usage

```
expect(locator).not_to_have_text(expected)
expect(locator).not_to_have_text(expected, **kwargs)
```

## Arguments

- `expected` `str|Pattern|List[str]|List[Pattern]|List[str|Pattern]` Added in: v1.18

Expected string or RegExp or a list of those.

- `ignore_case` `bool (optional)` Added in: v1.23

Whether to perform case-insensitive match. `ignore_case` option takes precedence over the corresponding regular expression flag if specified.

- `timeout` `float (optional)` Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

- `use_inner_text` `bool (optional)` Added in: v1.18

Whether to use `element.innerText` instead of `element.textContent` when retrieving DOM node text.

## not\_to\_have\_value

Added in: v1.20

The opposite of `expect(locator).to_have_value()`.

## Usage

```
expect(locator).not_to_have_value(value)
expect(locator).not_to_have_value(value, **kwargs)
```

## Arguments

- `value` `str|Pattern` Added in: v1.18

Expected value.

- `timeout` `float` (*optional*) Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

## not\_to\_have\_values

Added in: v1.23

The opposite of `expect(locator).to_have_values()`.

### Usage

```
expect(locator).not_to_have_values(values)
expect(locator).not_to_have_values(values, **kwargs)
```

### Arguments

- `values` `List[str]|List[Pattern]|List[str|Pattern]`

Expected options currently selected.

- `timeout` `float` (*optional*)

Time to retry the assertion for in milliseconds. Defaults to `5000`.

## to\_be\_attached

Added in: v1.33

Ensures that `Locator` points to an `attached` DOM node.

### Usage

**Sync**    **Async**

---

```
expect(page.get_by_text("Hidden text")).to_be_attached()
```

## Arguments

- `attached` **bool** (*optional*)
- `timeout` **float** (*optional*)

Time to retry the assertion for in milliseconds. Defaults to `5000`.

## to\_be\_checked

Added in: v1.20

Ensures the **Locator** points to a checked input.

## Usage

**Sync**    **Async**

---

```
from playwright.sync_api import expect

locator = page.get_by_label("Subscribe to newsletter")
expect(locator).to_be_checked()
```

## Arguments

- `checked` **bool** (*optional*) Added in: v1.18
- `timeout` **float** (*optional*) Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

## to\_be\_disabled

Added in: v1.20

Ensures the **Locator** points to a disabled element. Element is disabled if it has "disabled" attribute or is disabled via '**aria-disabled**'. Note that only native control elements such as HTML

`button`, `input`, `select`, `textarea`, `option`, `optgroup` can be disabled by setting "disabled" attribute. "disabled" attribute on other elements is ignored by the browser.

## Usage

**Sync**   **Async**

---

```
from playwright.sync_api import expect

locator = page.locator("button.submit")
expect(locator).to_be_disabled()
```

## Arguments

- `timeout` **float** (*optional*) Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

# to\_be\_editable

Added in: v1.20

Ensures the **Locator** points to an editable element.

## Usage

**Sync**   **Async**

---

```
from playwright.sync_api import expect

locator = page.get_by_role("textbox")
expect(locator).to_be_editable()
```

## Arguments

- `editable` **bool** (*optional*) Added in: v1.26
- `timeout` **float** (*optional*) Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

## to\_be\_empty

Added in: v1.20

Ensures the `Locator` points to an empty editable element or to a DOM node that has no text.

### Usage

**Sync**    **Async**

---

```
from playwright.sync_api import expect

locator = page.locator("div.warning")
expect(locator).to_be_empty()
```

### Arguments

- `timeout` `float` (*optional*) Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

## to\_be\_enabled

Added in: v1.20

Ensures the `Locator` points to an enabled element.

### Usage

**Sync**    **Async**

---

```
from playwright.sync_api import expect

locator = page.locator("button.submit")
expect(locator).to_be_enabled()
```

### Arguments

- `enabled` `bool` (*optional*) Added in: v1.26
- `timeout` `float` (*optional*) Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

## to\_be\_focused

Added in: v1.20

Ensures the `Locator` points to a focused DOM node.

### Usage

**Sync**   **Async**

---

```
from playwright.sync_api import expect

locator = page.get_by_role("textbox")
expect(locator).to_be_focused()
```

### Arguments

- `timeout` `float` (*optional*) Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

## to\_be\_hidden

Added in: v1.20

Ensures that `Locator` either does not resolve to any DOM node, or resolves to a `non-visible` one.

### Usage

**Sync**   **Async**

---

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

```
locator = page.locator('.my-element')  
expect(locator).to_be_hidden()
```

## Arguments

- `timeout` `float` (*optional*) Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

## to\_be\_in\_viewport

Added in: v1.31

Ensures the `Locator` points to an element that intersects viewport, according to the `intersection observer API`.

## Usage

**Sync**   **Async**

---

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

```
locator = page.get_by_role("button")  
# Make sure at least some part of element intersects viewport.  
expect(locator).to_be_in_viewport()  
# Make sure element is fully outside of viewport.  
expect(locator).not_to_be_in_viewport()  
# Make sure that at least half of the element intersects viewport.  
expect(locator).to_be_in_viewport(ratio=0.5)
```

## Arguments

- `ratio` `float` (*optional*)

The minimal ratio of the element to intersect viewport. If equals to `0`, then element should intersect viewport at any positive ratio. Defaults to `0`.

- `timeout` `float` (*optional*)



Time to retry the assertion for in milliseconds. Defaults to `5000`.

## to\_be\_visible

Added in: v1.20

Ensures that `Locator` points to an `attached` and `visible` DOM node.

To check that at least one element from the list is visible, use `locator.first`.

### Usage

**Sync**    **Async**

---

```
# A specific element is visible.
expect(page.get_by_text("Welcome")).to_be_visible()

# At least one item in the list is visible.
expect(page.get_by_test_id("todo-item").first).to_be_visible()

# At least one of the two elements is visible, possibly both.
expect(
  page.get_by_role("button", name="Sign in")
  .or_(page.get_by_role("button", name="Sign up"))
  .first
).to_be_visible()
```

### Arguments

- `timeout` `float` (*optional*) Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

- `visible` `bool` (*optional*) Added in: v1.26

## to\_contain\_text

Added in: v1.20

Ensures the `Locator` points to an element that contains the given text. You can use regular expressions for the value as well.

## Usage

**Sync**   **Async**

---

```
import re
from playwright.sync_api import expect

locator = page.locator('.title')
expect(locator).to_contain_text("substring")
expect(locator).to_contain_text(re.compile(r"\d messages"))
```

If you pass an array as an expected value, the expectations are:

1. Locator resolves to a list of elements.
2. Elements from a **subset** of this list contain text from the expected array, respectively.
3. The matching subset of elements has the same order as the expected array.
4. Each text value from the expected array is matched by some element from the list.

For example, consider the following list:

```
<ul>
  <li>Item Text 1</li>
  <li>Item Text 2</li>
  <li>Item Text 3</li>
</ul>
```

Let's see how we can use the assertion:

**Sync**   **Async**

---

```
from playwright.sync_api import expect

# ✓ Contains the right items in the right order
expect(page.locator("ul > li")).to_contain_text(["Text 1", "Text 3", "Text 4"])

# ✗ Wrong order
expect(page.locator("ul > li")).to_contain_text(["Text 3", "Text 2"])

# ✗ No item contains this text
```

```
expect(page.locator("ul > li")).to_contain_text(["Some 33"])
```

```
# * Locator points to the outer list element, not to the list items
expect(page.locator("ul")).to_contain_text(["Text 3"])
```

## Arguments

- `expected` `str|Pattern|List[str]|List[Pattern]|List[str|Pattern]` Added in: v1.18

Expected substring or RegExp or a list of those.

- `ignore_case` `bool (optional)` Added in: v1.23

Whether to perform case-insensitive match. `ignore_case` option takes precedence over the corresponding regular expression flag if specified.

- `timeout` `float (optional)` Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

- `use_inner_text` `bool (optional)` Added in: v1.18

Whether to use `element.innerText` instead of `element.textContent` when retrieving DOM node text.

## to\_have\_attribute

Added in: v1.20

Ensures the `Locator` points to an element with given attribute.

### Usage

**Sync**   **Async**

```
from playwright.sync_api import expect

locator = page.locator("input")
expect(locator).to_have_attribute("type", "text")
```

## Arguments

- `name` `str` Added in: v1.18

Attribute name.

- `value` `str|Pattern` Added in: v1.18

Expected attribute value.

- `timeout` `float` (*optional*) Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

## to\_have\_class

Added in: v1.20

Ensures the `Locator` points to an element with given CSS classes. This needs to be a full match or using a relaxed regular expression.

### Usage

```
<div class='selected row' id='component'></div>
```

**Sync**   **Async**

---

```
from playwright.sync_api import expect

locator = page.locator("#component")
expect(locator).to_have_class(re.compile(r"selected"))
expect(locator).to_have_class("selected row")
```

Note that if array is passed as an expected value, entire lists of elements can be asserted:

**Sync**   **Async**

---

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

```
locator = page.locator("list > .component")
expect(locator).to_have_class(["component", "component selected", "component"])
```

## Arguments

- `expected` `str|Pattern|List[str]|List[Pattern]|List[str|Pattern]` Added in: v1.18

Expected class or RegExp or a list of those.

- `timeout` `float` (*optional*) Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

## to\_have\_count

Added in: v1.20

Ensures the `Locator` resolves to an exact number of DOM nodes.

## Usage

**Sync**   **Async**

---

```
from playwright.sync_api import expect

locator = page.locator("list > .component")
expect(locator).to_have_count(3)
```

## Arguments

- `count` `int` Added in: v1.18

Expected count.

- `timeout` `float` (*optional*) Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

## to\_have\_css

Added in: v1.20

Ensures the **Locator** resolves to an element with the given computed CSS style.

## Usage

**Sync**   **Async**

---

```
from playwright.sync_api import expect

locator = page.get_by_role("button")
expect(locator).to_have_css("display", "flex")
```

## Arguments

- `name` **str** Added in: v1.18

CSS property name.

- `value` **str|Pattern** Added in: v1.18

CSS property value.

- `timeout` **float** (*optional*) Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

## to\_have\_id

Added in: v1.20

Ensures the **Locator** points to an element with the given DOM Node ID.

## Usage

**Sync**   **Async**

---

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

```
locator = page.get_by_role("textbox")
expect(locator).to_have_id("lastname")
```

## Arguments

- `id` `str`|`Pattern` Added in: v1.18

Element id.

- `timeout` `float` (*optional*) Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

## to\_have\_js\_property

Added in: v1.20

Ensures the `Locator` points to an element with given JavaScript property. Note that this property can be of a primitive type as well as a plain serializable JavaScript object.

## Usage

**Sync**   **Async**

---

```
from playwright.sync_api import expect

locator = page.locator(".component")
expect(locator).to_have_js_property("loaded", True)
```

## Arguments

- `name` `str` Added in: v1.18

Property name.

- `value` `Any` Added in: v1.18

Property value.

- `timeout` `float` (*optional*) Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

## to\_have\_text

Added in: v1.20

Ensures the **Locator** points to an element with the given text. You can use regular expressions for the value as well.

### Usage

**Sync**   **Async**

---

```
import re
from playwright.sync_api import expect

locator = page.locator(".title")
expect(locator).to_have_text(re.compile(r"Welcome, Test User"))
expect(locator).to_have_text(re.compile(r"Welcome, .*"))
```

If you pass an array as an expected value, the expectations are:

1. Locator resolves to a list of elements.
2. The number of elements equals the number of expected values in the array.
3. Elements from the list have text matching expected array values, one by one, in order.

For example, consider the following list:

```
<ul>
  <li>Text 1</li>
  <li>Text 2</li>
  <li>Text 3</li>
</ul>
```

Let's see how we can use the assertion:

**Sync**   **Async**

---



```

from playwright.sync_api import expect

# ✓ Has the right items in the right order
expect(page.locator("ul > li")).to_have_text(["Text 1", "Text 2", "Text 3"])

# ✗ Wrong order
expect(page.locator("ul > li")).to_have_text(["Text 3", "Text 2", "Text 1"])

# ✗ Last item does not match
expect(page.locator("ul > li")).to_have_text(["Text 1", "Text 2", "Text"])

# ✗ Locator points to the outer list element, not to the list items
expect(page.locator("ul")).to_have_text(["Text 1", "Text 2", "Text 3"])

```

## Arguments

- `expected` `str|Pattern|List[str]|List[Pattern]|List[str|Pattern]` Added in: v1.18

Expected string or RegExp or a list of those.

- `ignore_case` `bool (optional)` Added in: v1.23

Whether to perform case-insensitive match. `ignore_case` option takes precedence over the corresponding regular expression flag if specified.

- `timeout` `float (optional)` Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

- `use_inner_text` `bool (optional)` Added in: v1.18

Whether to use `element.innerText` instead of `element.textContent` when retrieving DOM node text.

## to\_have\_value

Added in: v1.20

Ensures the `Locator` points to an element with the given input value. You can use regular expressions for the value as well.

## Usage

```
import re
from playwright.sync_api import expect

locator = page.locator("input[type=number]")
expect(locator).to_have_value(re.compile(r"[0-9]"))
```

## Arguments

- `value` `str|Pattern` Added in: v1.18

Expected value.

- `timeout` `float` (*optional*) Added in: v1.18

Time to retry the assertion for in milliseconds. Defaults to `5000`.

## to\_have\_values

Added in: v1.23

Ensures the `Locator` points to multi-select/combobox (i.e. a `select` with the `multiple` attribute) and the specified values are selected.

## Usage

For example, given the following element:

```
<select id="favorite-colors" multiple>
  <option value="R">Red</option>
  <option value="G">Green</option>
  <option value="B">Blue</option>
</select>
```

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

```
locator = page.locator("id=favorite-colors")
locator.select_option(["R", "G"])
expect(locator).to_have_values([re.compile(r"R"), re.compile(r"G")])
```

## Arguments

- `values` `List[str]` | `List[Pattern]` | `List[str|Pattern]`

Expected options currently selected.

- `timeout` `float` (*optional*)

Time to retry the assertion for in milliseconds. Defaults to `5000`.