

# Ожидание

```
from playwright.sync_api import Page, expect

DOCS_URL = "https://playwright.dev/python/docs/intro"

def test_first(page: Page):
    page.goto("https://playwright.dev/python")
    link = page.get_by_role("link", name="GET STARTED")
    link.click()
    #assert page.url == DOCS_URL
    expect(page).to_have_url(DOCS_URL)
```

expect(page).to_have_url	наличие url
expect(page).to_have_title	наличие title
link = page.get_by_role("link", name="GET STARTEDer") expect(link).to_be_visible()	Видимость элемента в переменной link
expect(link).to_be_enabled()	Доступный элемент
expect(heading).to_contain_text()	Присутствие текста (часть)
expect(heading).to_have_text()	Присутствие текста (полное совпадение)
expect(mylink).to_have_class()	Наличие класса у элемента Несколько классов: "class1 class2" Должно быть полное соответствие. Но можно использовать регулярки. <div><pre>expect(mylink).to_have_class(     re.compile(r"navbar_link") )</pre></div>
expect(mylink).to_have_id()	Наличие id
expect(mylink).to_have_attribute(attr_name, attr_value)	Наличие атрибута. При необходимости можно указать значение атрибута.
expect(mylink).to_be_editable()	
expect(mylink).to_be_empty()	
expect(mycheckbox).to_be_checked()	

<code>expect(mymenu).to_have_value()</code>	Элемент в меню выбора
<code>expect(mymultimenu).to_have_values([])</code>	Несколько выбранных

`not_` - префикс отрицания

---

Revision #1  
Created 18 April 2025 15:25:55 by Admin  
Updated 18 April 2025 16:25:55 by Admin