


# Cypress to Playwright converters

```
TS cypress.spec.ts
1 describe('example to-do app', () => {
2   beforeEach(() => {
3     cy.visit('https://example.cypress.io/todo')
4   })
5
6   it('displays two todo items by default', () => {
7     cy.get('.todo-list li').should('have.length', 2)
8     cy.get('.todo-list li').first().should('have.text', 'Pay electric bill')
9     cy.get('.todo-list li').last().should('have.text', 'Walk the dog')
10  })
11
12  it('can add new todo items', () => {
13    const newItem = 'Feed the cat'
14    cy.get('[data-test=new-todo]').type(`${newItem}{enter}`)
15    cy.get('.todo-list li')
16      .should('have.length', 3)
17      .last()
18      .should('have.text', newItem)
19  })
20 })
```



```
TS playwright.spec.ts
1 import { test, expect } from '@playwright/test';
2
3 test.describe('example to-do app', () => {
4   test.beforeEach(async ({ page }) => {
5     await page.goto('https://example.cypress.io/todo');
6   });
7
8   test('displays two todo items by default', async ({ page }) => {
9     await expect(page.locator('.todo-list li')).toHaveCount(2);
10    await expect(page.locator('.todo-list li').first()).toHaveText('Pay electric bill');
11    await expect(page.locator('.todo-list li').last()).toHaveText('Walk the dog');
12  });
13
14  test('can add new todo items', async ({ page }) => {
15    const newItem = 'Feed the cat';
16    await page.locator('[data-test=new-todo]').fill(newItem);
17    await page.locator('[data-test=new-todo]').press('Enter');
18    await expect(page.locator('.todo-list li')).toHaveCount(3);
19    await expect(page.locator('.todo-list li').last()).toHaveText(newItem);
20  });
21 });
```



# Cypress to **Playwright** converters

TS cypress.spec.ts

```
1 describe('example to-do app', () => {
2   beforeEach(() => {
3     cy.visit('https://example.cypress.io/todo')
4   })
5
6   it('displays two todo items by default', () => {
7     cy.get('.todo-list li').should('have.length', 2)
8     cy.get('.todo-list li').first().should('have.text', 'Pay electric bill')
9     cy.get('.todo-list li').last().should('have.text', 'Walk the dog')
10  })
11
12  it('can add new todo items', () => {
13    const newItem = 'Feed the cat'
14    cy.get('[data-test=new-todo]').type(`${newItem}{enter}`)
15    cy.get('.todo-list li')
16      .should('have.length', 3)
17      .last()
18      .should('have.text', newItem)
19  })
20 })
```



[timdeschryver.dev/bits](https://timdeschryver.dev/bits)

# Cypress to **Playwright** converters

TS playwright.spec.ts

```
1 import { test, expect } from '@playwright/test';
2 test.describe('example to-do app', () => {
3   test.beforeEach(async ({ page }) => {
4     await page.goto('https://example.cypress.io/todo');
5   });
6
7   test('displays two todo items by default', async ({ page }) => {
8     await expect(page.locator('.todo-list li')).toHaveCount(2);
9     await expect(page.locator('.todo-list li').first()).toHaveText('Pay electric bill');
10    await expect(page.locator('.todo-list li').last()).toHaveText('Walk the dog');
11  });
12
13  test('can add new todo items', async ({ page }) => {
14    const newItem = 'Feed the cat';
15    await page.locator('[data-test=new-todo]').fill(newItem);
16    await page.locator('[data-test=new-todo]').press('Enter');
17    await expect(page.locator('.todo-list li')).toHaveCount(3);
18    await expect(page.locator('.todo-list li').last()).toHaveText(newItem);
19  });
20 });
```



[timdeschryver.dev/bits](https://timdeschryver.dev/bits)