DELETE ITEMS

📚 YOU WILL LEARN

- how to force-click an item
- how to execute element hover
- how to test deleting items

- clean up the existing code
  - `git reset ——hard`
  - `git clean -d -f`
- `git checkout b7`
- `npm install`

DELETING ITEMS

Question: how does the user delete an item?

TEST DELETING ITEMS IN PLAYWRIGHT

```javascript
// pw/delete-items.spec.js

test('deletes items', async ({ page }) => {
  // common locators
  const todos = page.locator('.todo-list li')
  await page.goto('/')
  await expect(todos).toHaveCount(3)

  // delete one completed item (the middle one)
  // confirm the remaining two items are still there
  // delete one incomplete item (the first one)
  // confirm the one remaining item
})
```

```
test('deletes items', async ({ page }) => {
  // common locators
  const todos = page.locator('.todo-list li')
  await page.goto('/')
  await expect(todos).toHaveCount(3)
  // delete one completed item (the middle one)
  await todos.nth(1).hover()
  await todos.nth(1).locator('.destroy').click()
  // confirm the remaining two items are still there
  await expect(todos).toHaveCount(2)
  await expect(todos).toHaveText(['Write code', 'Make tests pass'])
  // delete one incomplete item (the first one)
  await todos.nth(0).hover()
  await todos.nth(0).locator('.destroy').click()
  // confirm the one remaining item
  await expect(todos).toHaveCount(1)
```

Playwright solution includes hover() command

TEST DELETING ITEMS IN CYPRESS

```js
// cypress/e2e/delete-items.cy.js

it('deletes items', () => {
  // common locators
  const todos = '.todo-list li'

  cy.visit('/')
  cy.get(todos).should('have.length', 3)

  // delete one completed item (the middle one)
  // confirm the remaining two items are still there
  // delete one incomplete item (the first one)
  // confirm the one remaining item
})
```

Tip: you can force-click a button without it being visible, see cy.click

```javascript
it('deletes items', () => {
  // common locators
  const todos = '.todo-list li'

  cy.visit('/')
  cy.get(todos).should('have.length', 3)
  // delete one completed item (the middle one)
  cy.get(todos)
    .eq(1)
    .find('.destroy')
    // skip visibility check
    .click({ force: true })
  // confirm the remaining two items are still there
  cy.get(todos)
    .should('have.length', 2)
    .then(($li) => Cypress._.map($li, 'innerText'))
```

HOVER COMMAND IN CYPRESS

"Hover" is a *native* browser command / event and cannot be simulated using JavaScript (which is what Cypress commands are). Luckily, there is a Cypress plugin cypress-real-events that implements it.

- `git checkout b8`

```
// cypress/e2e/delete-items.cy.js

// can you hover of the todo item
// and avoid using .click({ force: true })
cy.get(todos).eq(1).find('.destroy').click({ force: true })
```

Can you update the test to use the `cypress-real-events` plugin and call `.hover()`?

CYPRESS hover COMMAND

- Install the plugin using `$ npm i -D cypress-real-events`
- Import it from the spec `import 'cypress-real-events'`
- use new child command `cy.realHover`

```
cy.get(todos).eq(1).realHover().find('.destroy').click()
```

DELETING ITEMS

- Playwright has built-in native events like hover
- Cypress can send native events via a plugin

➡️ Pick the next section or jump to the 06-network chapter