// test success

describe("My first simple test with Cypress", () => {

  it("True should be True", () => {

      expect(true).to.equal(true)

  })

})

------------------------------------------------------------------------------

// test failure

describe("My first simple test with Cypress", () => {

  it("True should be True", () => {

      expect(true).to.equal(false)

  })

})

------------------------------------------------------------------------------

// test with 2 describe block, each with 2 steps inside

describe("My second test with Cypress part 1", () => {

  it("True should be True", () => {

    expect(true).to.equal(true)

  })

  it("5 should be 5", () => {

    expect(5).to.equal(5)

  })

})

describe("My second test with Cypress part 2", () => {

  it("False should be False", () => {

    expect(false).to.equal(false)

  })

  it("10 should be 10", () => {

    expect(10).to.equal(10)

  })

})

------------------------------------------------------------------------------

// loading URL with timeout option (fails when timeout exceeds time given; 1000ms = 1s)

describe("Browser actions", () => {

  it("should load correct URL", () => {

    cy.visit("https://example.com/", { timeout: 10000 })

  })

})

------------------------------------------------------------------------------

// checking whether the URL is correct (checking whether a certain text is a part of an URL)

it("should check correct URL", () => {

  cy.url().should("include", "example.com")

})

// można zapisać obie komendy w jednym it:

it("should load correct URL and check a certain part of it", () => {

cy.visit("https://books.toscrape.com/index.html", {timeout: 10000})

cy.url().should("include", "index.html")

})

------------------------------------------------------------------------------

// implicit wait (optional) - if we want to wait more than default cypress waiting time to move on to the next test (next it)

it("should wait for 3 seconds", () => {

  cy.wait(3000)

})

------------------------------------------------------------------------------

// pauses the execution of the test until the resume button is clicked

it("should pause the execution", () => {

  cy.pause()

})

------------------------------------------------------------------------------

// checking, whether an element (e.g. a header h1) is visible on the page

it("should check for correct element on the page", () => {

  cy.get("h1").should("be.visible")

})

------------------------------------------------------------------------------

// clicks on links and buttons to check if they're working

// verifies and displays the number of elements (e.g. search results)

describe("Basic browser actions 2", () => {

  it("should click on 'Fantasy' category", () => {

    cy.get("a").contains("Fantasy").click()

    cy.get("h1").contains("Fantasy")

  })

  it("should display the number of results", () => {

    cy.get(".product\_pod").its("length").should("eq", 20)

  })

})

// grabs a searchbox, types something inside, then clicks the search button or presses enter

it("Google Search", () => {

  cy.visit("https://www.google.pl/");

  cy.get('.gLFyf').type("Automation Step by Step {Enter}");

  // alternatywa: cy.contains('Szukaj w Google').click()

})