Cypress cheat sheet

Below you’ll find some commands and selectors we think would be valuable for you to use while doing the exercises:

[.get()](https://docs.cypress.io/api/commands/get.html#Syntax) – gets one or more DOM elements by selector or alias

* Getting an element using Id:

cy.get('#thisIsAnId')

* Getting an element using Class:

cy.get('.thisIsAClass')

* Selects all HTML child elements specified by “child” of elements specified by “parent”. You can also use [.children()](https://docs.cypress.io/api/commands/children.html#Syntax):

cy.get('parent > child')

//or

cy.get('parent').children()

* Getting the [first](https://docs.cypress.io/api/commands/first.html#Syntax) HTML child from parent:

cy.get('ul > li').first()

* Getting HTML child on index X from parent using [eq()](https://docs.cypress.io/api/commands/eq.html#Syntax):

cy.get('ul > li').eq(X-1)

* Getting the element with a given data attribute:

        cy.get('[attribute="value"]')

[.contains()](https://docs.cypress.io/api/commands/contains.html#Syntax) – gets the DOM element containing the text/integer

* Gets the element containing the text ‘Click me!’

        cy.contains('Click me!')

[.should()](https://docs.cypress.io/api/commands/should.html#Syntax) – creates an [assertion](https://docs.cypress.io/guides/references/assertions.html#Chai), usually chained off [.get()](https://docs.cypress.io/api/commands/get.html#Syntax) and [.contains()](https://docs.cypress.io/api/commands/contains.html#Syntax)

* ‘Getters’ don’t do anything, but they enable you to write clear English sentences
  + to, be, been, is, that, which, and, has, have, with, at, of, same

Assertions include, but not limited to:

* *visible / hidden*

cy.get('#errorMessageBox').should('be.visible')

* *enabled / disabled*

cy.get('Button').should('be.enabled');

* *attr(name, value)*

cy.get('.LinkToGoogle').should('have.attr', 'href', 'https://www.google.com');

* *css(name, value)*

cy.get('#redContainer').should('have.css', 'font-size', '20px');

[.click()](https://docs.cypress.io/api/commands/click.html#Syntax) – clicks on a DOM element, chained off a yielded element

cy.get('.radioButton1').click();

[.dblclick()](https://docs.cypress.io/api/commands/dblclick.html#Syntax) – double clicks on a DOM element

cy.get('Button').dblclick();

[.rightClick()](https://docs.cypress.io/api/commands/rightclick.html#Syntax) – right clicks on a DOM element

cy.contains('Input').rightclick();

[.select()](https://docs.cypress.io/api/commands/select.html#Syntax) – selects an option in a <select> dropdown menu

cy.get('.dropDownFruitMenu').select('Bananas');

[.clear()](https://docs.cypress.io/api/commands/clear.html#Syntax) – clear the value of an input or text field

cy.get('#textBox').clear()

[.request()](https://docs.cypress.io/api/commands/request.html#Syntax) – makes a http request

cy.request('https://www.uio.no/studier/emner/matnat/ifi/IN3240/v21/')

* The request method yields the response as an object literal that contains properties such as:
  + Status
  + Body
  + Headers
  + Duration
* We can then assert the response payload against certain values:

cy.request('https://www.uio.no/studier/emner/matnat/ifi/IN3240/v21/').then(( response) => {

expect(response.body).to.include('<html lang=\"no\">');

        })

To skip a specified suite or test, append .skip() to the function – all nested suites will also be skipped:

    it.skip('Visits the Kitchen Sink', () =>{

        cy.visit('https://example.cypress.io');

    })