

CYPRESS CHEAT SHEET – Commands, Syntax & Examples

1. Test Structure

describe() – Suite

```
describe('Suite name', () => { ... })
```

it() – Test case

```
it('should do something', () => { ... })
```

Hooks

```
before(() => {})
beforeEach(() => {})
afterEach(() => {})
after(() => {})
```

2. Navigation

```
cy.visit('/login')           // Open page
cy.visit(url, options)
cy.reload()                 // Refresh page
cy.url().should('include', '/home') // Verify URL
cy.title().should('eq', 'Dashboard') // Verify title
```

3. Selectors & Locating Elements

```
cy.get('.class')           // CSS
cy.get('#id')
cy.get('[data-cy="button"]') // Best Practice
```

```
cy.contains('Login')           // By text
cy.contains('button', 'Save')
cy.get('form').find('input')    // Inside element
cy.get('section').within(() => { ... }) // Scoped search
```

4. Actions

```
cy.get('button').click()        // Click
cy.get('input').type('Hello')   // Type text
cy.get('input').type('text{enter}') // Keyboard keys
cy.get('input').clear()         // Clear field
cy.get('input[type=checkbox]').check()
cy.get('input[type=radio]').check('male')
cy.get('select').select('India') // Dropdown
cy.scrollTo('bottom')
cy.get('#el').scrollIntoView()
```

5. Assertions

```
cy.get(el).should('exist')
cy.get(el).should('be.visible')
cy.get(el).should('have.text', 'Hello')
cy.get(el).should('contain', 'Hello')
cy.get(el).should('have.value', 'Admin')
cy.get(el).should('have.class', 'active')
cy.get(list).should('have.length', 5)
cy.get(el).should('be.checked')
```

Chaining

```
cy.get(el).should('be.visible').and('be.enabled')
```

Explicit (Chai)

```
cy.then((val) => { expect(val).to.eq(10) })
```

6. Retry-Able & Waiting

```
cy.wait(1000)          // Hard wait (avoid)
cy.intercept('/api/users').as('users')
cy.wait('@users')      // Wait for API
cy.clock()             // Freeze time
cy.tick(5000)          // Move clock
```

7. Network Requests

Intercept

```
cy.intercept('GET', '/api/users').as('users')
cy.wait('@users').its('response.statusCode').should('eq', 200)
```

Mock Response

```
cy.intercept('GET', '/api/users', {
  statusCode: 200,
  body: [{ id: 1, name: 'User' }]
})
```

API Testing

```
cy.request('GET', '/api/data')
cy.request('POST', '/login', { user, pass })
```

8. Variables, Aliases & Chaining

Alias

```
cy.get('table').as('userTable')
cy.get('@userTable').should('be.visible')
```

Wrap

```
cy.wrap({ a: 1 }).its('a').should('eq', 1)
```

Then

```
cy.get('input').then(($el) => {
  const val = $el.val()
  expect(val).to.include('@')
})
```

9. Fixtures

```
cy.fixture('user.json').then((u) => {
  cy.get('#email').type(u.email)
})
```

With intercept

```
cy.fixture('users.json').then((data) => {
  cy.intercept('/api/users', data)
})
```

10. Browser State

```
cy.window().then(win => win.localStorage.clear())
cy.document().its('contentType').should('eq', 'text/html')
cy.viewport(1280, 720)           // Desktop
cy.viewport('iphone-6')         // Mobile preset
```

11. File Upload / Download

```
cy.get('input[type="file"]').selectFile('cypress/fixtures/file.txt')
cy.readFile('path/to/file.json')
cy.writeFile('path/to/file.json', { name: 'Murali' })
```

12. System & Backend Tasks

```
cy.exec('npm run reset-db')      // OS command
```

```
cy.task('log', 'Start test') // Node tasks
```

13. Sessions & Authentication

```
cy.session('login', () => {
  cy.request('POST', '/login', { user, pass })
})
cy.visit('/dashboard')
```

14. Shadow DOM

```
cy.get('my-component').shadow().find('.btn').click()
```

15. Iframe Handling

```
cy.get('iframe').its('0.contentDocument.body')
  .should('not.be.empty')
  .then(cy.wrap)
  .find('#btn').click()
```

16. Custom Commands

```
// commands.js
Cypress.Commands.add('login', (email, pass) => {
  cy.visit('/login')
  cy.get('#email').type(email)
  cy.get('#password').type(pass)
  cy.contains('Login').click()
})

// Usage:
cy.login('[email protected]', 'secret')
```

17. Utilities

```
cy.log('message')
cy.debug()
cy.screenshot()           // Full page
cy.screenshot('homepage') // Named
cy.viewport('macbook-15') // Presets
```

18. Best Practices

- ✓ Use `data-cy` attributes
- ✓ Prefer `cy.contains()` over complex selectors
- ✓ Avoid `cy.wait(3000)`
- ✓ Use `intercept()` for network waits
- ✓ Use **custom commands** to reduce repetition
- ✓ Keep tests atomic & independent