ChatGPT

Cypress Cheat Sheet (All-in-One)

Getting Started

```
npm install cypress --save-dev
npx cypress open     # Opens the Test Runner
npx cypress run      # Runs tests headlessly
```

Cypress File Structure

```
cypress/
  └── integration/   # Test files
  └── fixtures/      # Test data
  └── support/       # Custom commands, setup
cypress.config.js    # Cypress config (v10+)
```

Basic Test Syntax

```
describe('Test Suite', () => {
  it('Test Case', () => {
    // test code
  });
});
```

Selectors

```
cy.get('button')                 // By element
y.get('#login')                  // By ID
cy.get('.btn-primary')           // By class
cy.get('[data-cy=submit]')       // By data attribute (recommended)
```

🖱️ Actions

```
cy.click()              // Click element
cy.type('text')         // Type into input
cy.clear()              // Clear input
cy.select('Option')     // Select dropdown option
cy.check()              // Check checkbox
cy.uncheck()            // Uncheck checkbox
cy.scrollIntoView()     // Scroll into view
```

🔍 Assertions

```
.should('exist')
.should('be.visible')
.should('have.text', 'Hello')
.should('contain', 'Hello')
.should('have.value', '123')
.should('have.class', 'active')
```

⌛ Waits and Timing

```
cy.wait(1000)                    // Wait 1 second
cy.get('button', { timeout: 10000 }) // Custom timeout
cy.wait('@alias')               // Wait for intercept alias
```

API / Network Requests

```
cy.intercept('GET', '/api/users').as('getUsers');
cy.wait('@getUsers');
```

📦 Fixtures (Test Data)

```
cy.fixture('user.json').then((data) => {
  cy.get('#name').type(data.name);
});
```

Custom Commands

In `cypress/support/commands.js`:

```
Cypress.Commands.add('login', (email, password) => {
  cy.get('#email').type(email);
  cy.get('#password').type(password);
  cy.get('form').submit();
});
```

Usage:

```
cy.login('user@example.com', '123456');
```

Debugging

```
cy.log('Message')
cy.debug()
cy.pause()
```

Test Hooks

```
before(() => { /* once before all tests */ });
beforeEach(() => { /* before each test */ });
after(() => { /* after all tests */ });
afterEach(() => { /* after each test */ });
```

Page Navigation

```
cy.visit('/login');
cy.url().should('include', '/login');
```

Utilities

```
cy.wrap(value)             // Wrap a value into Cypress chain
Cypress._                  // Lodash available
Cypress.env('TOKEN')       // Read env variables
```

Full Example Test

```
describe('Login Flow', () => {
  it('logs in successfully', () => {
    cy.visit('/login');
    cy.get('#email').type('test@example.com');
    cy.get('#password').type('password123');
    cy.get('button[type=submit]').click();
    cy.url().should('include', '/dashboard');
  });
});
```

API Testing Example

```
cy.request('POST', '/api/login', {
  email: 'user@example.com',
  password: '123456'
}).then((response) => {
  expect(response.status).to.eq(200);
});
```

BDD (Cucumber) Integration

Install:

```
npm install --save-dev cypress-cucumber-preprocessor
```

In `cypress.config.js`:

```
const createBundler = require('@bahmutov/cypress-esbuild-preprocessor');
const addCucumberPreprocessorPlugin = require('@badeball/cypress-cucumber-
preprocessor').addCucumberPreprocessorPlugin;
```

```
module.exports = (on, config) => {
  addCucumberPreprocessorPlugin(on, config);
  on('file:preprocessor', createBundler());
  return config;
};
```

Use `.feature` files with step definitions.

Component Testing (v10+)

```
npx cypress open --component
```

Supported frameworks: React, Vue, Angular, etc.

🛠 Useful Plugins

- `cypress-axe` – accessibility testing
- `cypress-real-events` – real browser events
- `cypress-file-upload` – file uploads
- `mochawesome` – custom HTML reports

📊 Reporting

```
npm install --save-dev mochawesome
npx cypress run --reporter mochawesome
```

Need additional templates or real test examples? Just ask!