

Command-Cypress-Interview

1) What is cypress?

- cypress is a next generation testing tool constructed for the modern web.
- It uses javascript for writing tests but goy. coding can be done using cypress inbuilt commands, which are easy to understand.

2) How cypress architecture is different from selenium?

- selenium executes remote command across the network
 - but in cypress it directly operates inside the browser
- Consider in selenium
if I have element.click() command if I execute this script selenium sends this code to browser specific drivers like chromeDriver etc. then this browser will interact with specific browser like chrome browser, firefox
- Consider in cypress
Same command element.click()
Once you execute code in cypress, cypress directly sends the code to browser & click action get execute.

3) Can I use cypress framework with c# / java just like selenium.

- No, cypress is Node.js based framework. cypress supports only javascript and typescript as programming language.

Q) Could you tell me about some differences between cypress & selenium?

→ selenium	cypress.
1) Selenium supports all major languages like C#, Java, Python, Javascript.	cypress support only javascript / typescript programming language.
2) Selenium Commands executed directly through web drivers	cypress command Executed directly on the browser.
3) selenium support all major browsers,	cypress support only chrome browser.
4) selenium appium can be used to test native mobile application	cypress doesn't support any native mobile application testing

5) What is disadvantage of using cypress.

- cypress doesn't support all major languages.
It support only javascript & typescript.
- It doesn't support all browsers like safari
It support only chrome browser
- cypress doesn't support multiple tabs.
- We can not run cypress tests on multiple browsers at the same time.

6) What are the advantages of cypress

- cypress is nodejs Modern framework so it works lessly with single page application.
- cypress provides time travel options, so it takes snapshots of each tests we can see what happened exactly in each step.
- cypress Debuggability feature provides access to developer tools we can debug directly in the browser.
- cypress directly executes commands on browser.
- cypress directly executes command runs test and executes commands directly on browser

7) Can I use JUnit / TestNG in cypress.

- we can not use the JUnit / TestNG in cypress, cypress comes with mocha framework & chai assertion libraries.

8) How can I install cypress

- If you want to install cypress, we need to install Node first, once you install node we can install cypress with command `npm install cypress`

g) How can I open cypress window & execute test

→ We can open Cypress using npx cypress open command.

10) Could you talk something about cypress file & folder structure.

→ The folder named cypress is located in the project root folder which is main folder for cypress automation framework.

→ By default cypress folder contains 4 subfolders.
→ features
→ integration
→ plugins
→ support

→ features - folder can be used to store our external JSON data files (e.g. feature() → command)

→ Integration - folder mainly consists our actual spec/test files.

→ plugins - this is very helpful when we want to have pre processors & post processor.

→ Support - folder contains the special file index.js which will be run before each & every test.

It also used to create utility methods.

11) What is cypress CLI?

- Cypress CLI is unique feature in cypress.
- It provides ability to run our cypress tests in command line.
- It provides many options and flags control the cypress test behavior.
e.g. `npm cypress run` to run your test

12) How can I run single specfile in commandline

- We can run single specfile in commandline using spec option
e.g. `npm cypress run spec="myspec.js"`

13) I am new to cypress, I wanted to execute hello world test script in cypress how can I do that could you explain.

- Once you have installed cypress it will create folder called integration, inside integration folder you can create your first spec file using javascript.

your spec file should contain describe() block & it() block
use the cypress command inside this block and execute the test

14) Why should I create my tests inside integration folder, why can't I create test in other folder.

→ The default tests folder is integration in cypress.

but you can create the tests in other folders in that case you need to change the configuration & also you need to specify

15) How can I see default configuration in cypress

→ Under cypress window:

+
test Runner

✓
settings & configuration

16) How can I create suites in cypress

→ We can create a describe() block the describe block act as suite, inside that each test can be created as single it() block.

17) tell me at least 5 cypress commands.

→ cy.visit: cy.visit() is used to navigate to the specific website.
e.g cy.visit("URL")

cy.log: cy.log() is used for getting display cypress console logs during execution.
e.g cy.log('test 123');

cy.get() - used for getting dom element in cypress
cy.get("#headerme")

5 cy.url() - cy.url is used to get the current URL of the page
cy.url() to get the url back

18) How can I click on the button in cypress.

10 → .click() function is used to click the element on cypress.

eg google Search button

15 cy.get('#Search').click()

19) list Out Some Commands which I can use to interact with DOM element?

→ 20 .click() is used to click on the element
cy.get('#search').click();

.dblclick() is used to double click on the element
cy.get(')').dblclick();

25 .rightclick() is used to right click on the element
cy.get(')').rightclick();

*type() is used to type on element/ text boxes.
30 cy.get(')').type('');

(clear()) is used to clear fields/ texboxes
cy.get(')').clear();

20) What are the selectors supported by cypress?

→ By default cypress supports only CSS selector.

21) Can I use xpath in cypress?

→ Cypress doesn't support xpaths by default, but if we install third party plugin like cypress-xpath.

NPM Install cypress-xpath

22) Consider a scenario, I have link on webpage

if clicking this link opens new window I need to verify some text in that window how can I verify

→ Cypress by default doesn't support working with new windows So we need to follow diff approach

e.g.
take me to google

In the above code clicking on take me to google link opens the new window the reason is we have attribute target="blank"

23) What is environment variable in cypress?

→ The environment variables is a variable whose value sets at the level of the operating system.

In order to use environment variable we need to define in cypress.json

{

"env": {

key: value

}

}

24) How can I change the baseurl in cypress dynamically.

→ When we say executing multiple environment that means we are executing script on Staging.

→ We can change baseurl dynamically by passing baseurl in command line.

First we need to configure cypress.json with baseurl option.

```
code <cypress.json>
  "baseurl": "https://qa.kleb.com"
```

then we can override this baseurl in command line while running our test.

```
npx cypress run --config
```

```
  "baseurl": "https://www.staging.labsite.com"
```

The above code runs the tests on staging website.

→ earlier configured qa.web.com will be open with command line argument baseurl value.

25) What is preserve cookies in cypress? Why do we need that.

→ cypress by default clear the cookies after every test. In order to stop clearing cookies we need to use preserve cookies option in cypress.

describe ("my test", ()=> {

it ('test1', ()=> {

// Some code to test
 });

it ('test2', ()=> {

// Some code to test
 });

it ('test3', ()=> {

// Some code to test
 });

})

In above code we have three test namely test1, test2, test3.

Tests consider you have logged into your application in test1, after execution of test1 or moves to test2, it again asks for login because login session & cookies are cleared after test1.

→ This is default behavior in cypress, we can overcome this problem by adding code in our test.

```
beforeEach(()=> {
  cypress.Cookies.preserveOnce('session-id',
    'ember-token')
})
```

26) What is custom commands in cypress.

→ Cypress comes with default set of commands like cy.visit cy.reload etc. like this we can create our custom commands in cypress.

e.g I can create some commands.

like cy.login()

which will enter the username & password and click on submit.

All custom commands are first configured in

cypress/support/command.js.

```
cypress.commands.addl("login",
  (username, password)=> {
```

```
  cy.get("#username").type(username);
  cy.get("#password").type(password);
  cy.get("#login").click();
});
```

Once we add the above command.

```
cy.visit("http://mysite.com/login.html");
cy.login('myusername', 'mypassword');
```

27) How can I type keypress in cypress.

→ Keypress can be done in cypress using `.type()` command,

If we need to use the keys like CTRL,
SHIFT, ALT

We need to specify in braces.

```
cy.get('login').type('{shift}{alt} hello');
```

28) How can I read the value from cypress configuration file

We can read the cypress config values from cypress using `cypress.config()`

e.g `pageLoadTimeout`

```
{
  "pageLoadTimeout": 60000
}
```

29) How to get the title of the page.

→ `cy.title()`: can be used to get the title of current active window.

for e.g -

cy.visit('https://www.facebook.com')

cy.title().should('eq', 'facebooklogin')

30) How can I use the sleep in cypress.

→ cy.wait() cy.wait is used to wait for specific amount of time

e.g cy.wait(1000)

31) How can I get the first and last child of the selected element?

first()

last() is used to getting first & last child of element

e.g -

cy.get('input[name="row"]')

cy.get('input[name="rows"]').first()

cy.get('input[name="rows"]').last()

32) What is shadow DOM? How can I access shadow DOM in cypress?

→ shadow DOM allows hidden DOM trees to be attached to element in the regular DOM tree.

→ shadow DOM tree starts with a shadow root.

e.g -

<div class="shadow-host">

shadow-root

<button class="my-button">click me</button>

</div>

We can access the above shadow dom using below code in cypress

```
cy.get('.shadow-host').shadow().find('.my-button').click()
```

* demo orangeHRM *

login functionality :-

```

describe ('login functionality', function () {
    it ('verify login functionality', function () {
        cy.visit ('URL')
        cy.get('#id').click().type(' ')
        cy.get('#id').type(' ')
        cy.get('#id').click()
    })
})
    
```

→ npae cypress open

If we want describe

→ for login functionality we need Username & Password

→ first we have to create describe block
the describe block act as suit.

In-side that we create it block
in it block we write title of our testcase
like verify login functionality
then we use commands.

first cy.visit for Navigate specific website
then use cy.get command to get the element
for Username & Password.

then use type to type of element
again use cy.get command for get login button
element & use click() command to click
on button.

describe - group of testcase

it - for single testcase