**What is Cypress?**

· Cypress is a next generation front end Automation testing tool built for the modern web applications

**How Cypress is Unique from other Automation tools?**

Cypress [automatically waits](https://docs.cypress.io/guides/core-concepts/introduction-to-cypress.html#Cypress-is-Not-Like-jQuery) for commands and assertions before moving on. No more async hell.

Ability to test Edge Testcases by Mocking the server response

Cypress takes snapshots as your tests run. We can hover over each commands in the [Command Log](https://docs.cypress.io/guides/core-concepts/test-runner.html#Command-Log) to see exactly what happened at each step.

Because of its Architectural design, Cypress delivers fast, consistent and reliable test execution compared to other Automation tools

View videos of your entire tests execution when run from the Cypress Dashboard.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Cypress built on Node.js and comes packaged as an npm module,

As it is built on Node.js, It uses JavaScript for writing tests. But 90% of coding can be done using Cypress inbuilt commands which are easy to understand.

Cypress also [bundles with jQuery](https://docs.cypress.io/guides/references/bundled-tools.html#Other-Library-Utilities) and inherits many of jQuery methods for UI components Identification

**Cypress Architecture**

Most testing tools (like Selenium) operate by running outside of the browser and executing remote commands across the network. But Cypress engine directly operates inside the browser. In other words, It is the browser that is executing your test code

This enables Cypress to listen and modify the browser behavior at run time by manipulating DOM and altering Network requests and responses on the fly

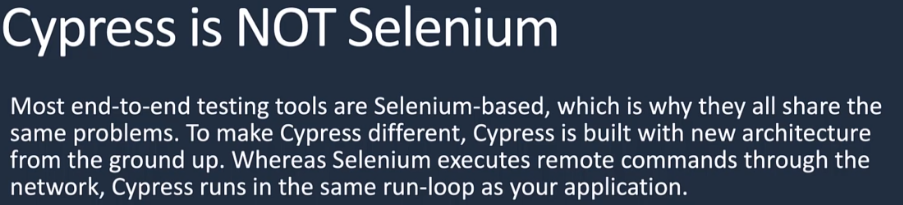
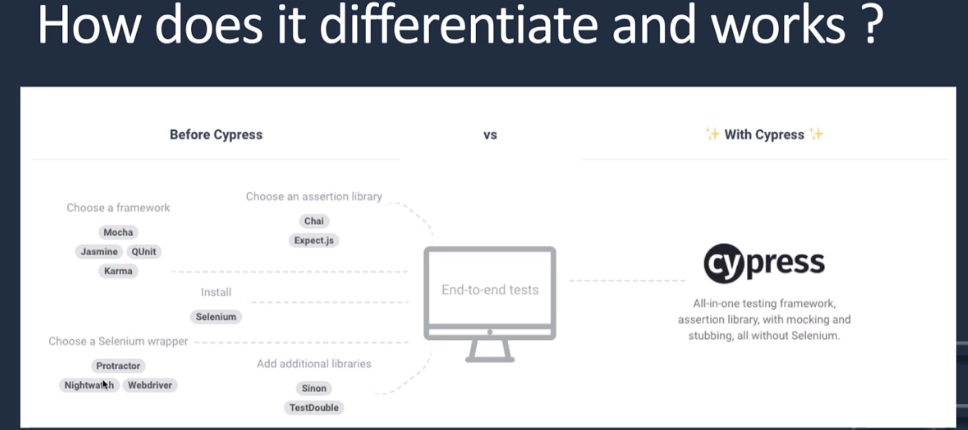
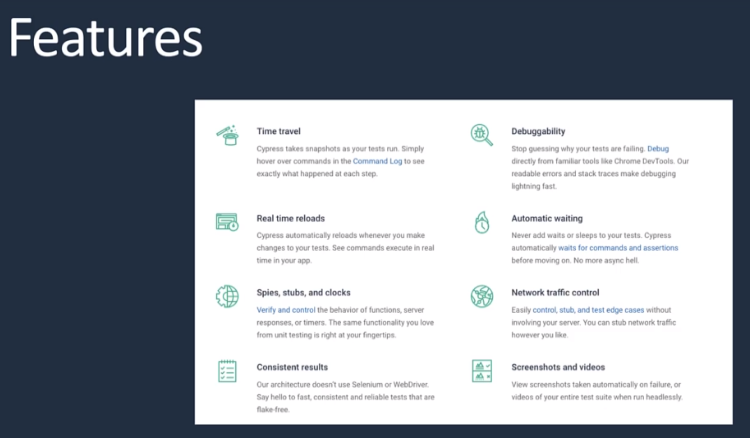
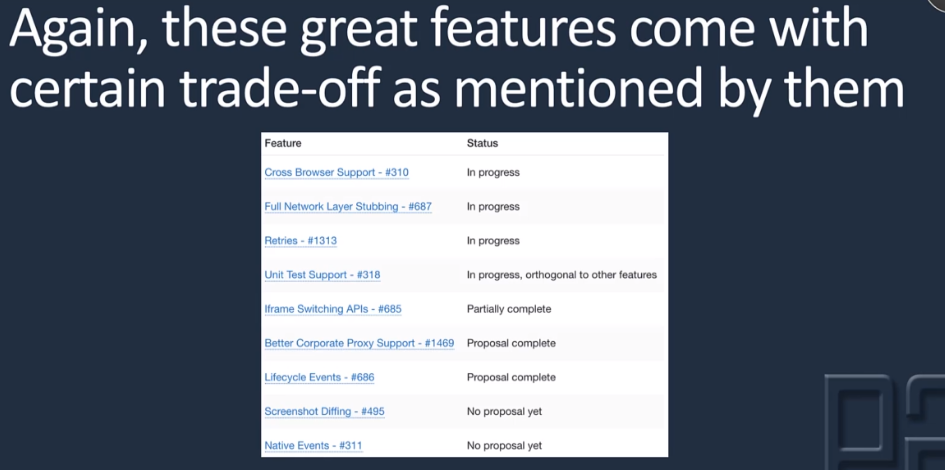
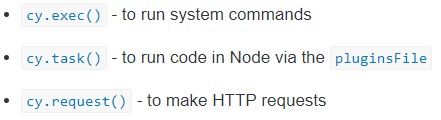
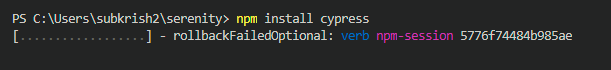
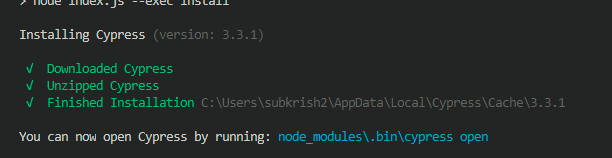
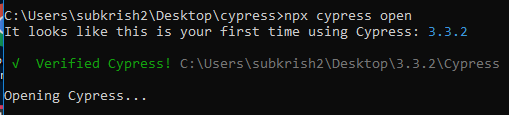
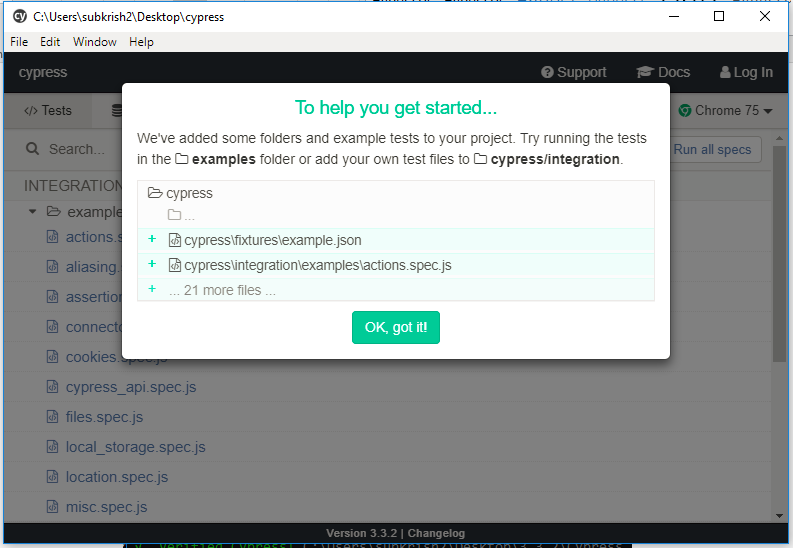
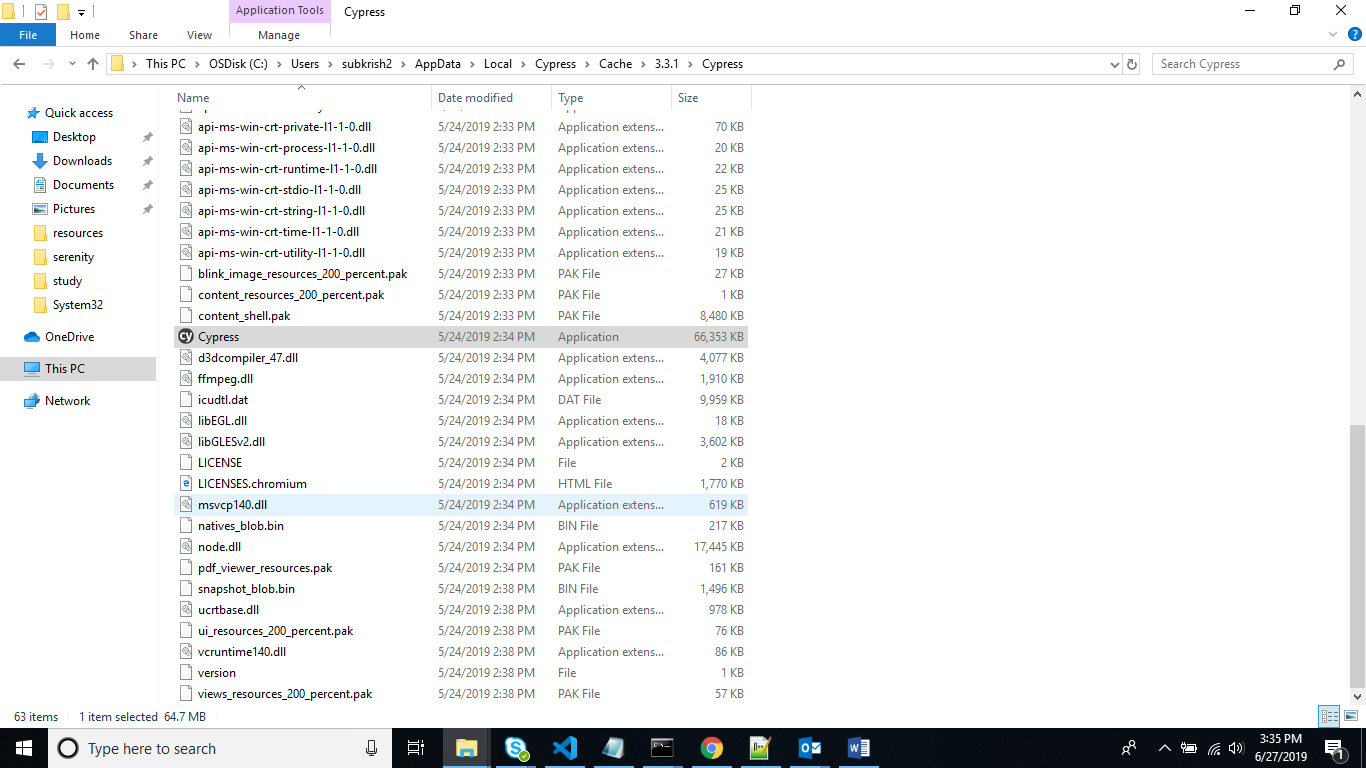
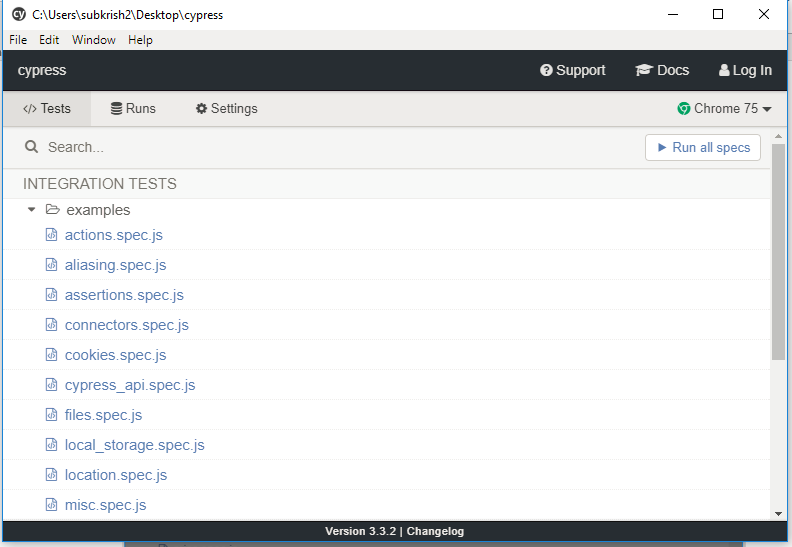
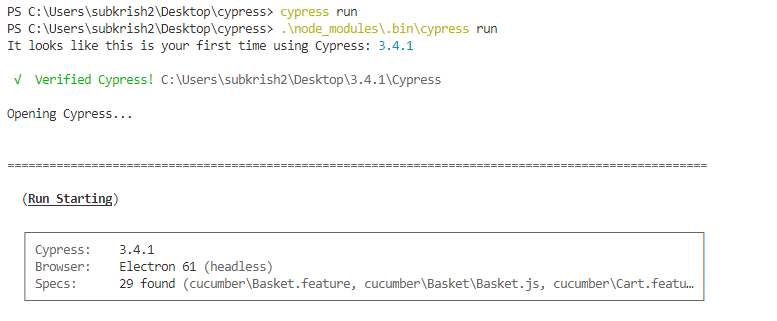
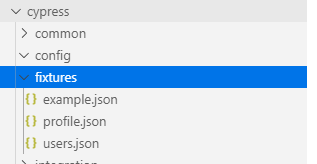
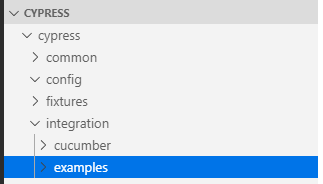
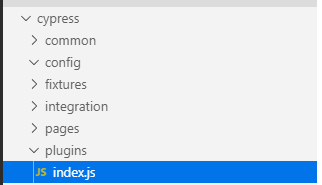
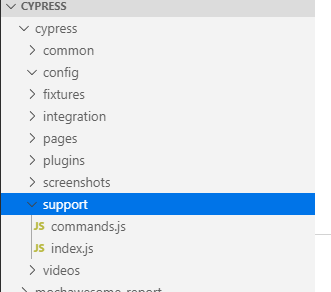
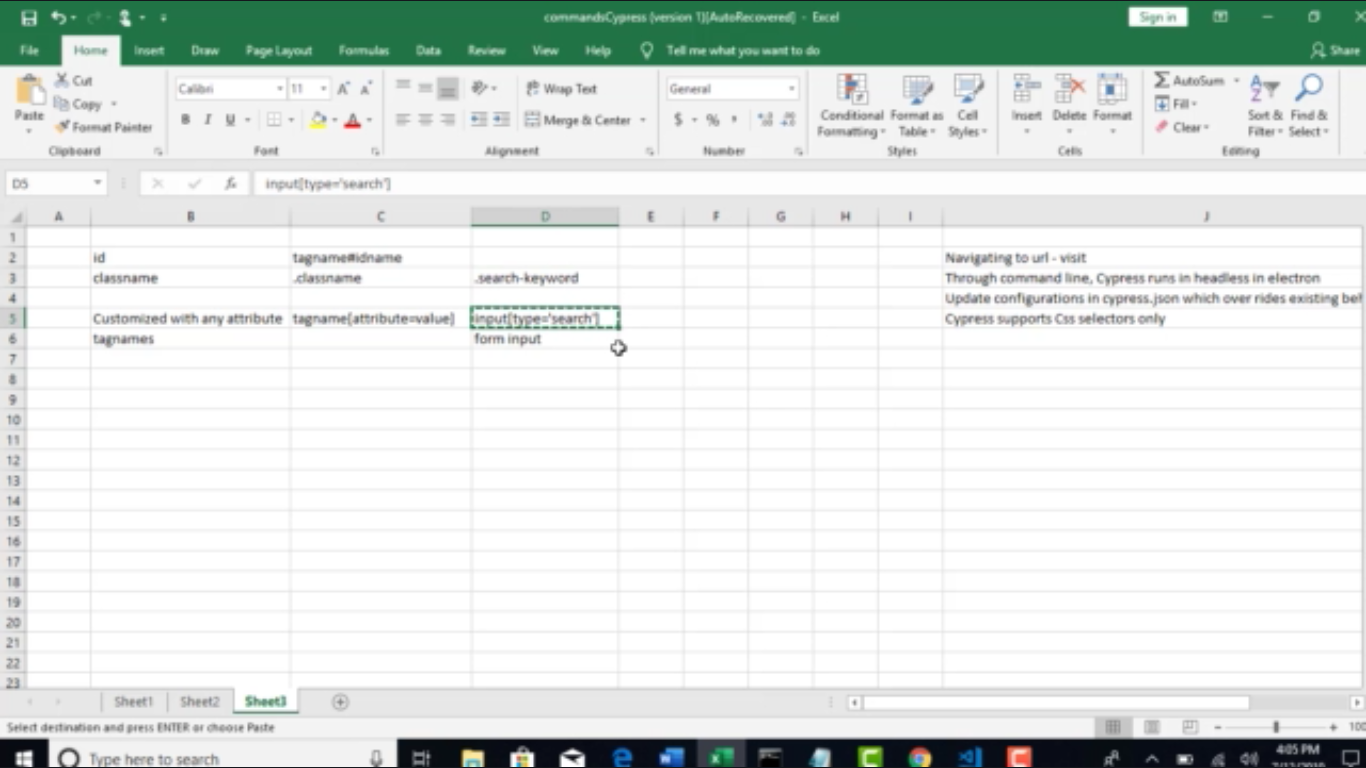
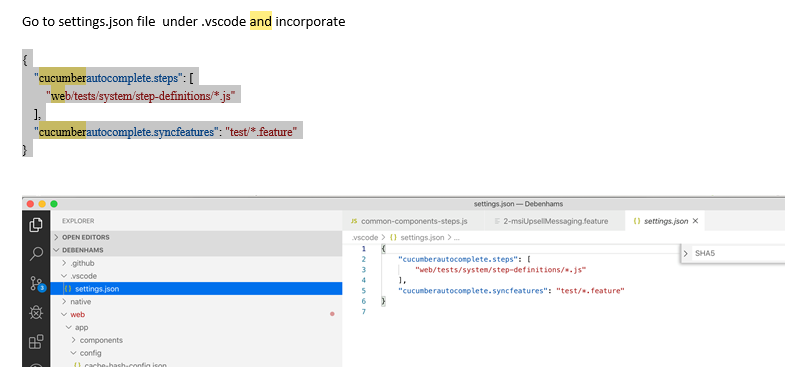
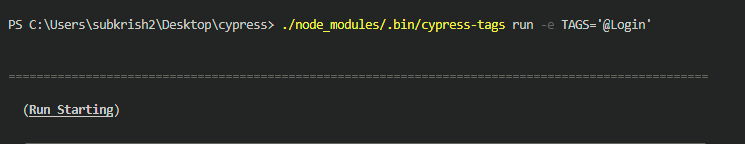
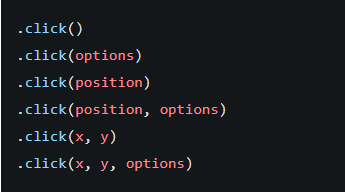
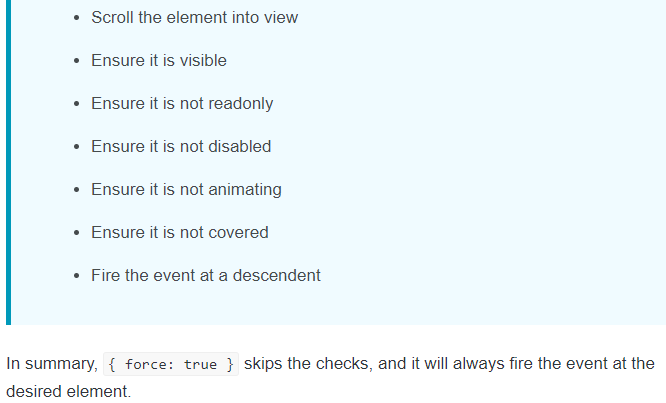
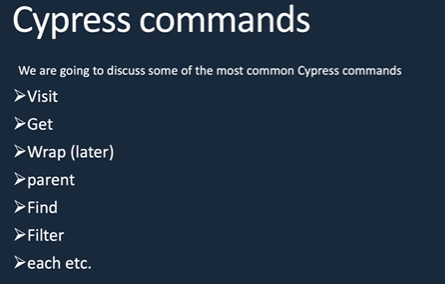
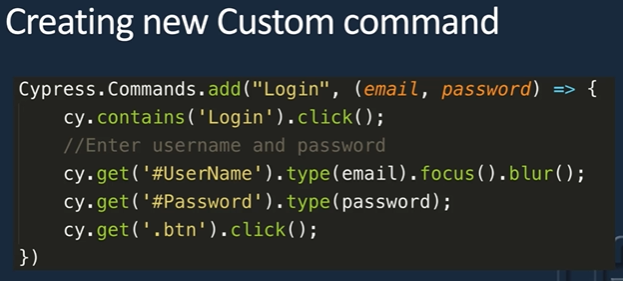
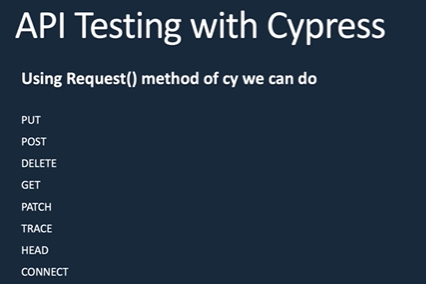
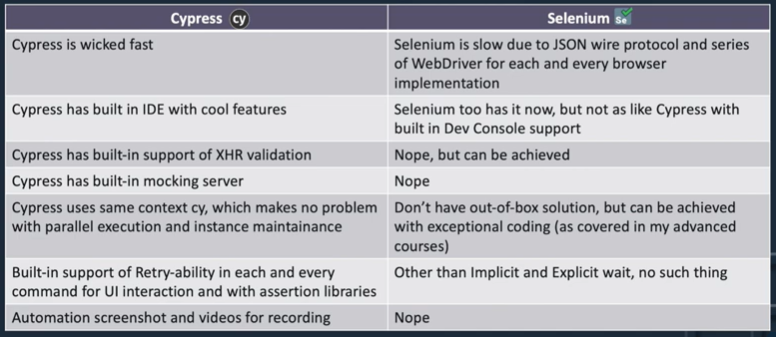
Cypress open doors to New Kind of testing with Having ultimate control over your application (front and back)

Cypress Browser Support:

Chrome  
Electron  
Firefox & IE (Under Construction)

Cypress Components:

Test Runner  
Dash Board Service

1. 
2. 
3. 
4. Currently cypress supports chromium, chrome and electron
5. 
6. Cypress can be used for e2e test, unit test and integration test
7. 
8. Cypress doesn’t kick off the browser automation magic until the test function exits. -> to prevent test falkiness
9. 
10. 
11. Over default installation path as above by using env variable  
    
12. 
13. 
14. Alternately, if the cypress UI doesnot open, navigate to the path below and launch the cypress.exe file
15.   
    Launch cypress.exe file  
    Creates playground for cypress
16.   
    click Run All Specs , This is called as cypress runner
17. Default command timeout is 4000ms
18. Common cypress commands -> click, clear, visit, get, wrap, parent, find, filter,each
19. When cypress is executed from command line, it runs in electron browser and as headless mode  
    
20. All test data must be placed under fixtures folder  
    
21. All test cases must be written under integration folder  
    
22.   
    Plugins are used as event listeners
23.   
    All reusable methods must be placed under support folder and override existing commands
24. Cypress supports only css selectors  
    
25. Auto completion of code  
       
    above line must be added at the start of the file
26. 
27. Execution in cypress:   
    
28.   
    click({force:true}) -> over rides a default behavior and forces the event to occur  
    
29. 
30. Filter() and find() -> filter() finds the dom of the specific selector, find()- > fetches the descendant of the specific selector
31. Invoke() is an alternate for resolving promises using then()
32. Wrap()
33. Implicit assertions -> expect()
34. Explicit assertions -> should()
35. Cypress Hooks -> Before, After, BeforeAll, AfterAll
36. Fixtures() -> reads files from fixtures folder
37. readFile() -> reads files from cypress folder
38. Creating custom command  
    
39.   
    generates step definitions automatically in console
40. Cypress XHR -> XMLHttpRequest is an API in the form of an object whose methods transfer data between web browser and web server
41. 
42. Cypress V/s Selenium  
      
    XHR validation can be done in selenium using browsermop proxy