Cypress Documentation

- ## Cypress installation

1. Install Node in the machine

   URL - https://nodejs.org/en/

i. Check Node installation - "node --version" or "node -v"

ii. Check npm (node package manager) - "npm -v"

2. 2. Install VS code

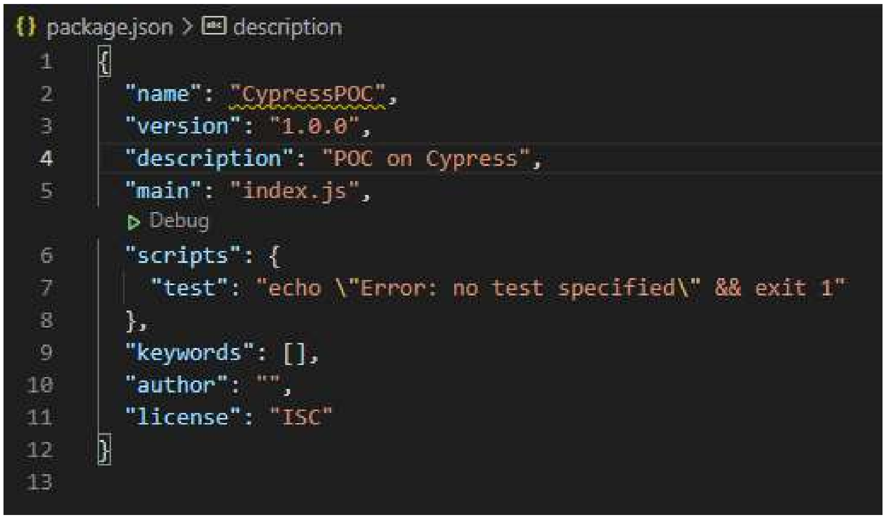
   URL - https://code.visualstudio.com/download

- ## Workspace and project creation

1. Create a workspace folder in any preferred drive-in machine and

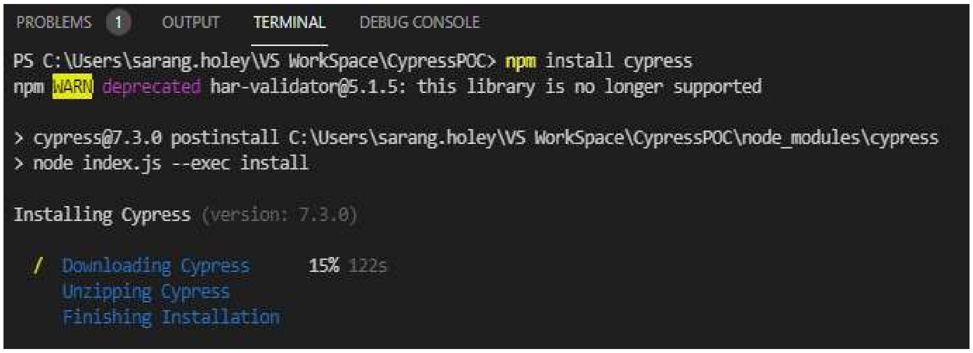
   create a folder inside the workspace with your project name.

2. Open the sub folder created in the workspace in VS Code.
3. Open terminal from VS Code pointing to the project folder.
4. Type Command “npm init -y” **“npm -i init**” hit enter.
5. This will create a package.json file in the project, this helps to manage the dependencies of the project and execute the project from the command line.

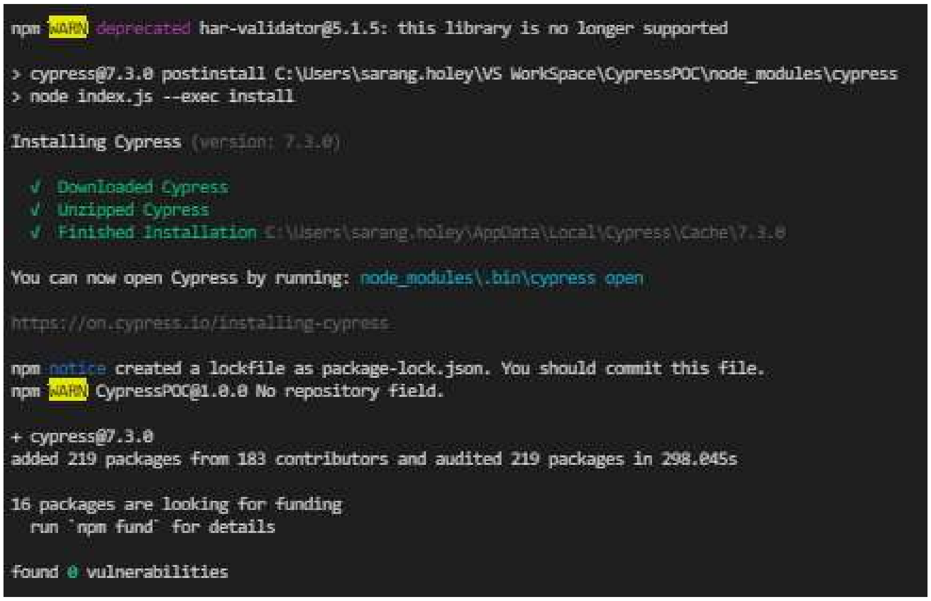


```
{
  "name": "CypressPOC",
  "version": "1.0.0",
  "description": "POC on Cypress",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

```

6. Go to the terminal with the same project location and type – “npm install cypress” or “npm install cyress@6.0” [for specific cypress version installation] and hit enter. This will install cypress in the machine.

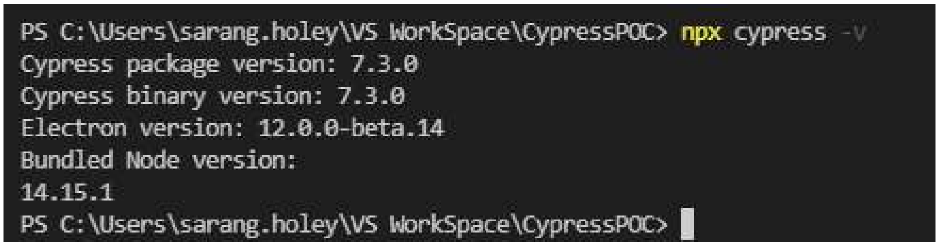


```
PS C:\Users\sarang.holey\VS WorkSpace\CypressPOC> npm install cypress
npm WARN deprecated har-validator@5.1.5: this library is no longer supported

> cypress@7.3.0 postinstall C:\Users\sarang.holey\VS WorkSpace\CypressPOC\node_modules\cypress
> node index.js --exec install

Installing Cypress (version: 7.3.0)

  / Downloading Cypress      15% 122s
    Unzipping Cypress
    Finishing Installation
```



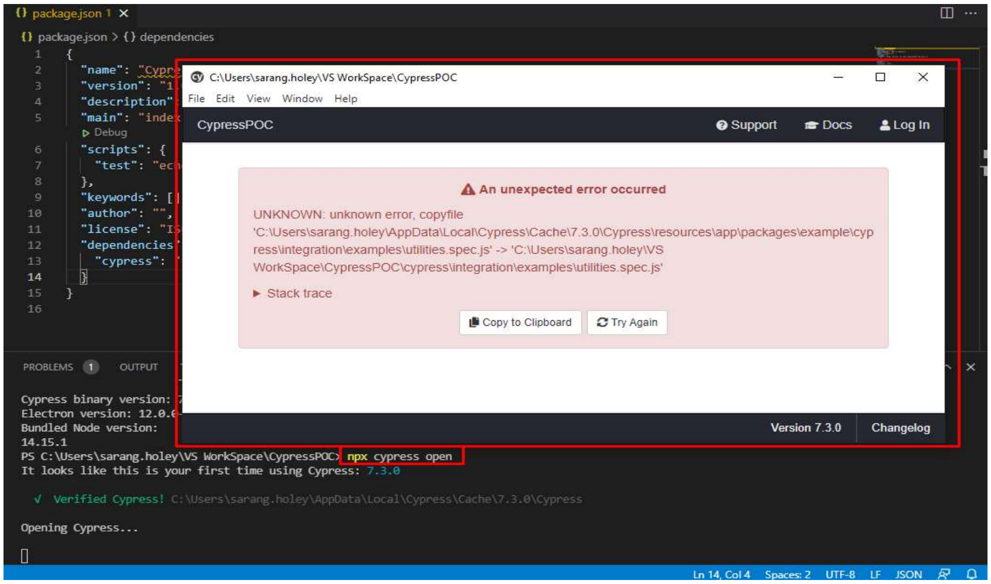
```
npm WARN deprecated har-validator@5.1.5: this library is no longer supported

> cypress@7.3.0 postinstall C:\Users\sarang.holey\VS WorkSpace\CypressPOC\node_modules\cypress
> node index.js --exec install

Installing Cypress (version: 7.3.0)

  √  Downloaded Cypress
  √  Unzipped Cypress
  √  Finished Installation C:\Users\sarang.holey\AppData\Local\Cypress\Cache\7.3.0

You can now open Cypress by running: node_modules\.bin\cypress open

https://on.cypress.io/installing-cypress

npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN CypressPOC@1.0.0 No repository field.

+ cypress@7.3.0
added 219 packages from 183 contributors and audited 219 packages in 298.045s

16 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

7. To check the cypress installed version use command.
    a. “npx cypress -v” or “npx cypress --version”
    b. npx is a command to check the version in the local folder.



```
PS C:\Users\sarang.holey\VS WorkSpace\CypressPOC> npx cypress -v
Cypress package version: 7.3.0
Cypress binary version: 7.3.0
Electron version: 12.0.0-beta.14
Bundled Node version:
14.15.1
PS C:\Users\sarang.holey\VS WorkSpace\CypressPOC> 
```

8. Now to open cypress in your current project type command in terminal (opened in same project directory)
    a. “npx cypress open”
    b. This will open a cypress window and add some folders in the project also.



```
{
  "name": "Cypre
  "version": "1
  "description"
  "main": "inde
  "scripts": {
    "test": "ec
  },
  "keywords": [
  "author": "",
  "license": "I
  "dependencies
    "cypress":
  }
}
```

An unexpected error occurred

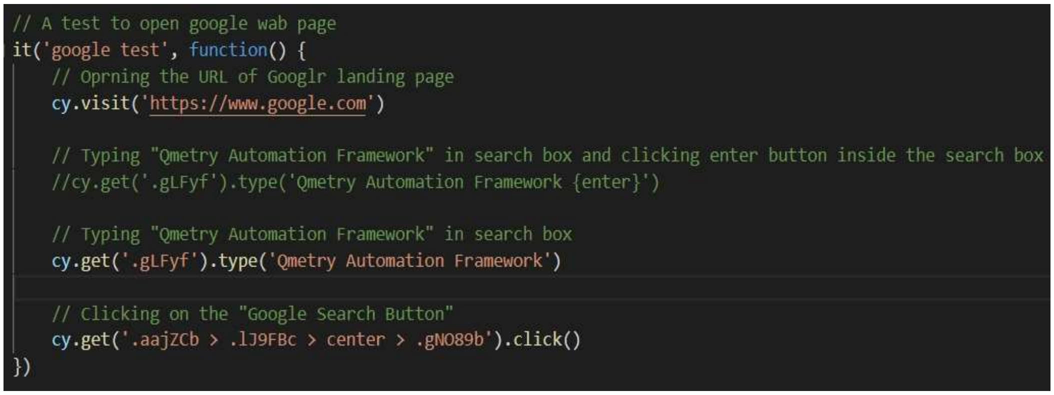
UNKNOWN: unknown error, copyfile 'C:\Users\sarang.holey\AppData\Local\Cypress\Cache\7.3.0\Cypress\resources\app\packages\example\cypress\integration\examples\utilities.spec.js' -> 'C:\Users\sarang.holey\VS WorkSpace\CypressPOC\cypress\integration\examples\utilities.spec.js'

```
Cypress binary version: 7
Electron version: 12.0.
Bundled Node version:
14.15.1
PS C:\Users\sarang.holey\VS WorkSpace\CypressPOC> npx cypress open
It looks like this is your first time using Cypress: 7.3.0

  √  Verified Cypress! C:\Users\sarang.holey\AppData\Local\Cypress\Cache\7.3.0\Cypress

Opening Cypress...
```

    c. Now you can close the cypress tool window.

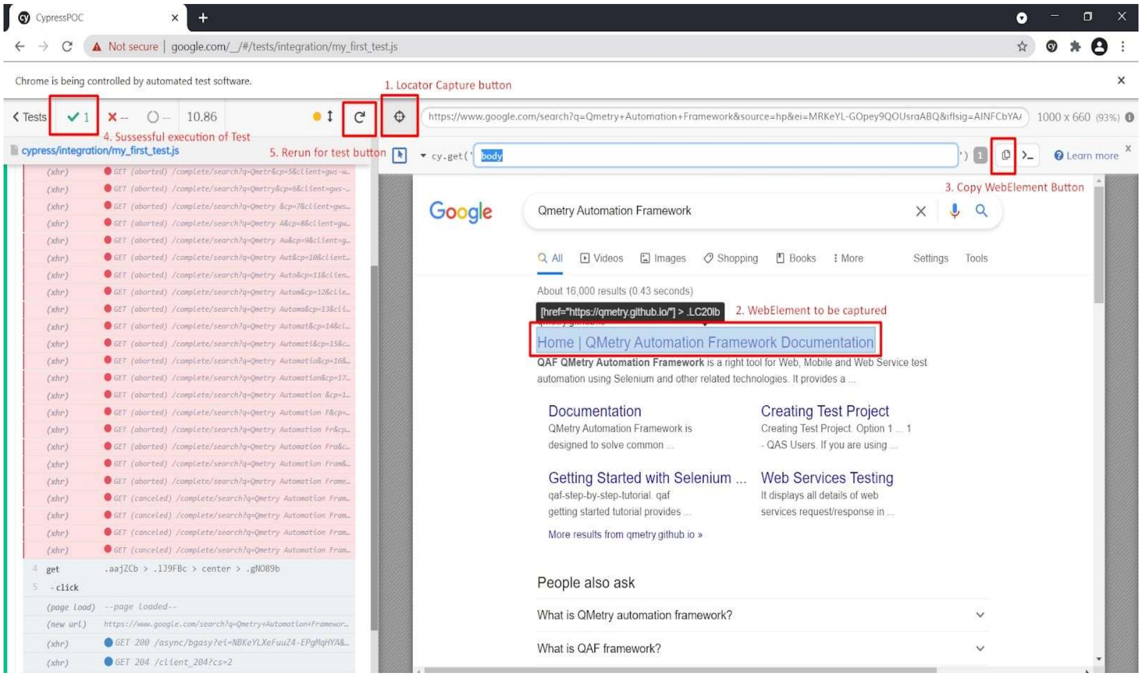
- ## Creation of first test using Cypress

   1. Create a file in integration folder with name “my_first_test.js”
   2. In this file mention at top - /// <reference types=”cypress” />
   3. This will allow the VS code to auto complete the commands from the cypress library.



```
// A test to open google wab page
it('google test', function() {
    // Oprning the URL of Googlr landing page
    cy.visit('https://www.google.com')

    // Typing "Qmetry Automation Framework" in search box and clicking enter button inside the search box
    //cy.get('.gLFyf').type('Qmetry Automation Framework {enter}')

    // Typing "Qmetry Automation Framework" in search box
    cy.get('.gLFyf').type('Qmetry Automation Framework')

    // Clicking on the "Google Search Button"
    cy.get('.aajZCb > .lJ9FBc > center > .gNO89b').click()
})
```

   4. Type as mentioned for the first test
   5. Now type “npx cypress open” in the terminal to execute the test.
   6. This will open the cypress window as well as the browser with the given URL in step no 4.
   7. Following image covered the basic and important component of cypress window



CypressPOC
Not secure | google.com/__/#/tests/integration/my_first_test.js
Chrome is being controlled by automated test software.
1. Locator Capture button
Tests
✓ 1
✕ --
○ --
10.86
4. Sussessful execution of Test
5. Rerun for test button
cypress/integration/my_first_test.js
https://www.google.com/search?q=Qmetry+Automation+Framework&source=hp&ei=MRKeYL-GOpey9QOUsraABQ&iflsig=AINFCbYA
1000 x 660 (93%)
cy.get('body')
Learn more
3. Copy WebElement Button
Google
Qmetry Automation Framework
All
Videos
Images
Shopping
Books
More
Settings
Tools
[href="https://qmetry.github.io/"] > .LC20lb
2. WebElement to be captured

```
(xhr)        GET (aborted) /complete/search?q=Qmetr&cp=5&client=gws-w…
(xhr)        GET (aborted) /complete/search?q=Qmetry&cp=6&client=gws-…
(xhr)        GET (aborted) /complete/search?q=Qmetry &cp=7&client=gws…
(xhr)        GET (aborted) /complete/search?q=Qmetry A&cp=8&client=gw…
(xhr)        GET (aborted) /complete/search?q=Qmetry Au&cp=9&client=g…
(xhr)        GET (aborted) /complete/search?q=Qmetry Aut&cp=10&client…
(xhr)        GET (aborted) /complete/search?q=Qmetry Auto&cp=11&clien…
(xhr)        GET (aborted) /complete/search?q=Qmetry Autom&cp=12&clie…
(xhr)        GET (aborted) /complete/search?q=Qmetry Automa&cp=13&cli…
(xhr)        GET (aborted) /complete/search?q=Qmetry Automat&cp=14&cl…
(xhr)        GET (aborted) /complete/search?q=Qmetry Automati&cp=15&c…
(xhr)        GET (aborted) /complete/search?q=Qmetry Automatio&cp=16&…
(xhr)        GET (aborted) /complete/search?q=Qmetry Automation&cp=17…
(xhr)        GET (aborted) /complete/search?q=Qmetry Automation &cp=1…
(xhr)        GET (aborted) /complete/search?q=Qmetry Automation F&cp=…
(xhr)        GET (aborted) /complete/search?q=Qmetry Automation Fr&cp…
(xhr)        GET (aborted) /complete/search?q=Qmetry Automation Fra&c…
(xhr)        GET (aborted) /complete/search?q=Qmetry Automation Fram&…
(xhr)        GET (aborted) /complete/search?q=Qmetry Automation Frame…
(xhr)        GET (canceled) /complete/search?q=Qmetry Automation Fram…
(xhr)        GET (canceled) /complete/search?q=Qmetry Automation Fram…
(xhr)        GET (canceled) /complete/search?q=Qmetry Automation Fram…
(xhr)        GET (canceled) /complete/search?q=Qmetry Automation Fram…
4 get        .aajZCb > .lJ9FBc > center > .gNO89b
5  -click
(page load)  --page loaded--
(new url)    https://www.google.com/search?q=Qmetry+Automation+Framewor…
(xhr)        GET 200 /async/bgasy?ei=NBKeYLXeFuuZ4-EPgMqHYA&…
(xhr)        GET 204 /client_204?cs=2
```

About 16,000 results (0.43 seconds)

Home | QMetry Automation Framework Documentation

QAF QMetry Automation Framework is a right tool for Web, Mobile and Web Service test automation using Selenium and other related technologies. It provides a ...

Documentation
QMetry Automation Framework is designed to solve common ...

Creating Test Project
Creating Test Project. Option 1 ... 1 - QAS Users. If you are using ...

Getting Started with Selenium ...
qaf-step-by-step-tutorial. qaf getting started tutorial provides ...

Web Services Testing
It displays all details of web services request/response in ...

More results from qmetry.github.io »

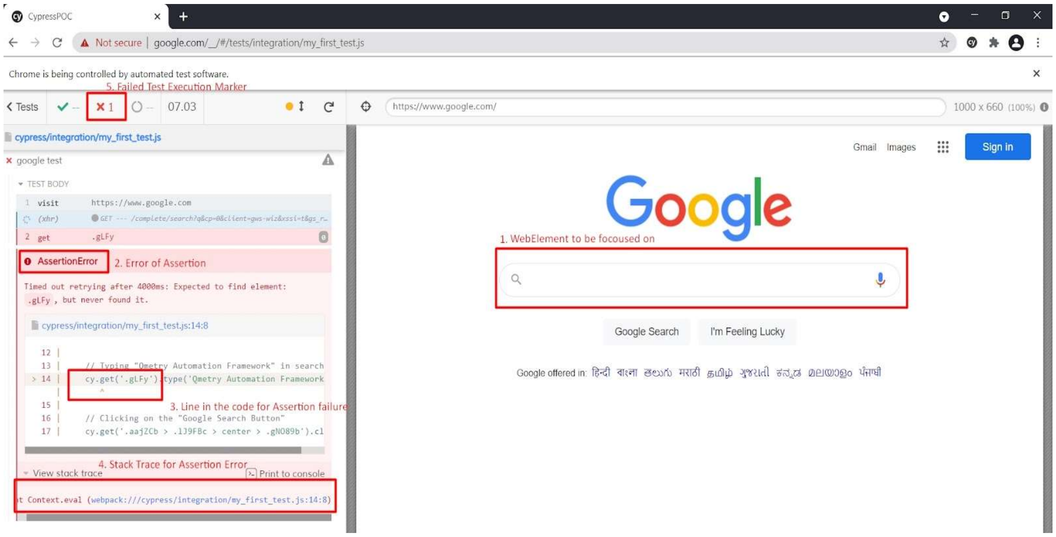
People also ask

What is QMetry automation framework?

What is QAF framework?

- 1. Locator Capture button
- 2. WebElement to be captured
- 3. Copy WebElement button
- 4. Successful Execution of test
- 5. Rerun for the test button

8. Now if the webelement locator is not matched with the actual locator then following image will be seen


CypressPOC
Not secure | google.com/__/#/tests/integration/my_first_test.js
Chrome is being controlled by automated test software.
Tests
✕ 1
07.03
https://www.google.com/
1000 x 660 (100%)
Gmail
Images
Sign in
Google Search
I'm Feeling Lucky
Google offered in: हिन्दी বাংলা తెలుగు मराठी தமிழ் ગુજરાતી ಕನ್ನಡ മലയാളം ਪੰਜਾਬੀ
View stack trace
Print to console

5. Failed Test Execution Marker

1. WebElement to be focused on

cypress/integration/my_first_test.js

✕ google test

TEST BODY

```
1 visit    https://www.google.com
  (xhr)    GET --- /complete/search?q&cp=0&client=gws-wiz&xssi=t&gs_r…
2 get      .gLFy
```

AssertionError — 2. Error of Assertion

```
Timed out retrying after 4000ms: Expected to find element:
.gLFy , but never found it.
```

cypress/integration/my_first_test.js:14:8

```

    // Typing "Qmetry Automation Framework" in search
    cy.get('.gLFy').type('Qmetry Automation Framework
       ^

    // Clicking on the "Google Search Button"
    cy.get('.aajZCb > .lJ9FBc > center > .gNO89b').cl
```

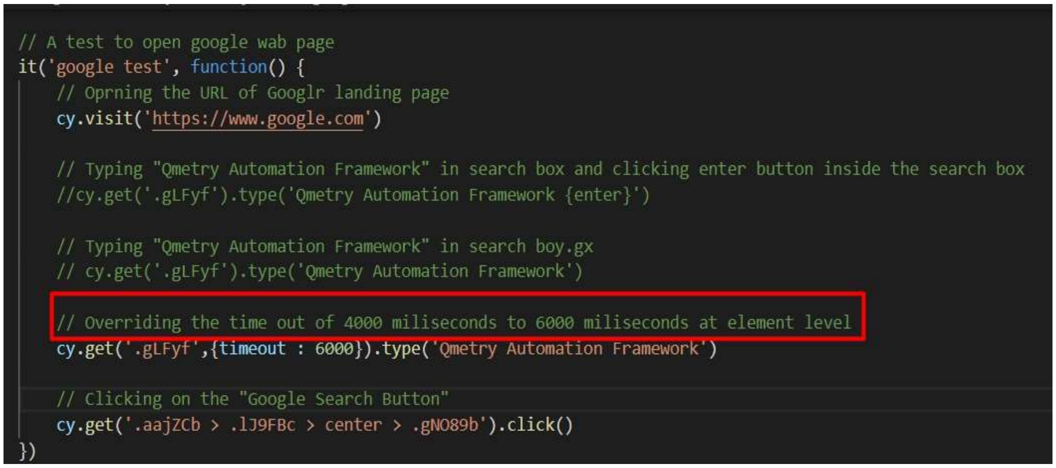
3. Line in the code for Assertion failure

4. Stack Trace for Assertion Error

```
at Context.eval (webpack:///cypress/integration/my_first_test.js:14:8)
```

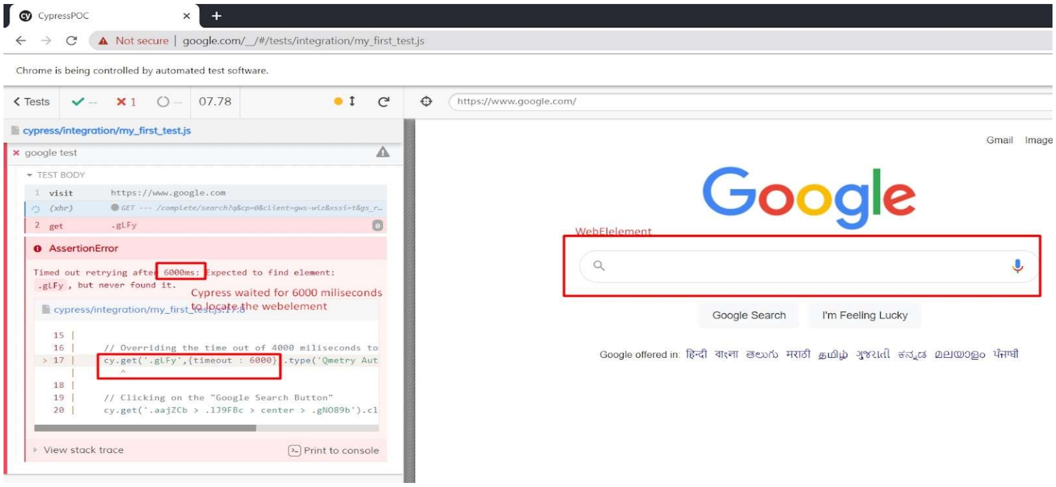
Note - Cypress by default waits for 4000 milliseconds to locate the webelement on the web page before showing any error.

9. An custom time out can also be added at element level as well as global level to the script for finding of the delayed webelement



```
// A test to open google wab page
it('google test', function() {
    // Oprning the URL of Googlr landing page
    cy.visit('https://www.google.com')

    // Typing "Qmetry Automation Framework" in search box and clicking enter button inside the search box
    //cy.get('.gLFyf').type('Qmetry Automation Framework {enter}')

    // Typing "Qmetry Automation Framework" in search boy.gx
    // cy.get('.gLFyf').type('Qmetry Automation Framework')

    // Overriding the time out of 4000 miliseconds to 6000 miliseconds at element level
    cy.get('.gLFyf',{timeout : 6000}).type('Qmetry Automation Framework')

    // Clicking on the "Google Search Button"
    cy.get('.aajZCb > .lJ9FBc > center > .gNO89b').click()
})
```

Following image shows how cypress wait for 6000 milliseconds before throwing error


CypressPOC
Not secure | google.com/__/#/tests/integration/my_first_test.js
Chrome is being controlled by automated test software.
Tests
✕ 1
07.78
https://www.google.com/
Gmail
Image
WebElelement
Google Search
I'm Feeling Lucky
Google offered in: हिन्दी বাংলা తెలుగు मराठी தமிழ் ગુજરાતી ಕನ್ನಡ മലയാളം ਪੰਜਾਬੀ
View stack trace
Print to console

cypress/integration/my_first_test.js

✕ google test

TEST BODY

```
1 visit    https://www.google.com
  (xhr)    GET --- /complete/search?q&cp=0&client=gws-wiz&xssi=t&gs_r…
2 get      .gLFy
```

AssertionError

```
Timed out retrying after 6000ms: Expected to find element:
.gLFy , but never found it.
```

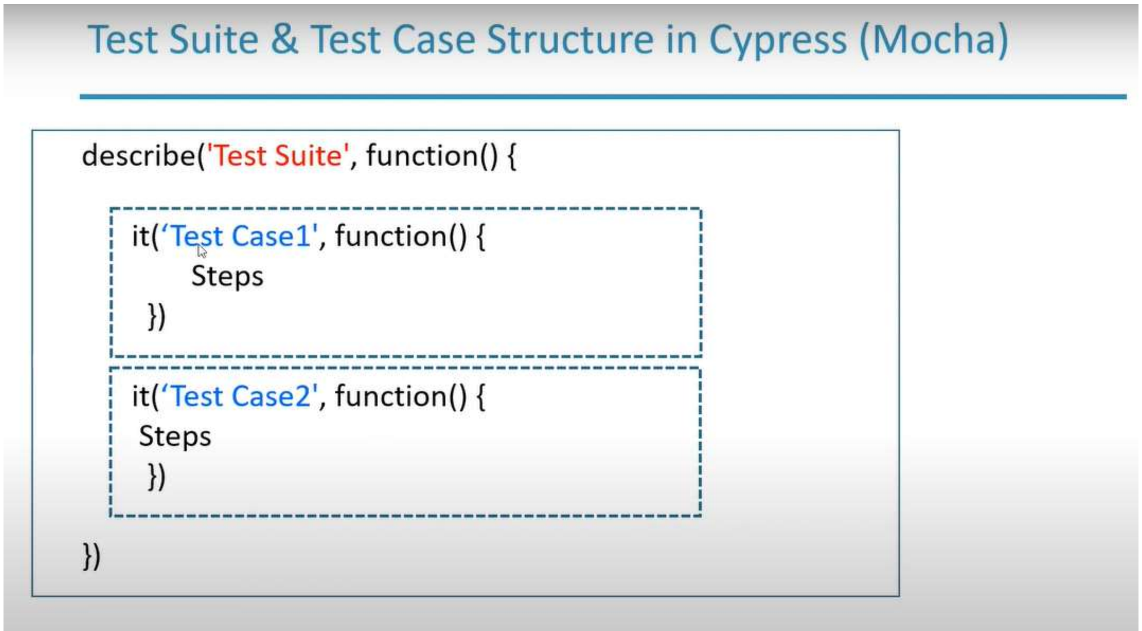
Cypress waited for 6000 miliseconds to locate the webelement

cypress/integration/my_first_test.js:17:8

```

    // Overriding the time out of 4000 miliseconds to
    cy.get('.gLFy',{timeout : 6000}).type('Qmetry Aut
       ^

    // Clicking on the "Google Search Button"
    cy.get('.aajZCb > .lJ9FBc > center > .gNO89b').cl
```

- **Test Framework in Cypress**

1. Cypress supports Mocha as a test framework.
2. By default Mocha is internally available to run the test in Cypress



Test Suite & Test Case Structure in Cypress (Mocha)

```
describe('Test Suite', function() {

    it('Test Case1', function() {
          Steps
     })

    it('Test Case2', function() {
     Steps
     })

})
```

3. The above diagram is a Test Suite and inside it there are multiple blocks we can write as different test cases.

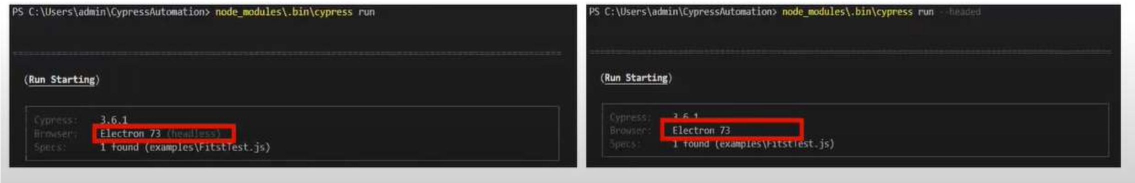
- **Run Cypress Tests in Test Runner & Terminal**

- To open **Test Runner**

```
node_modules\.bin\cypress open
```

- To Run **All the tests** under **examples** directory

```
node_modules\.bin\cypress run

node_modules\.bin\cypress run --headed
```



```
PS C:\Users\admin\CypressAutomation> node_modules\.bin\cypress run

================================================================================

  (Run Starting)

  Cypress:    3.6.1
  Browser:    Electron 73 (headless)
  Specs:      1 found (examples\FitstTest.js)
```

```
PS C:\Users\admin\CypressAutomation> node_modules\.bin\cypress run --headed

================================================================================

  (Run Starting)

  Cypress:    3.6.1
  Browser:    Electron 73
  Specs:      1 found (examples\FitstTest.js)
```

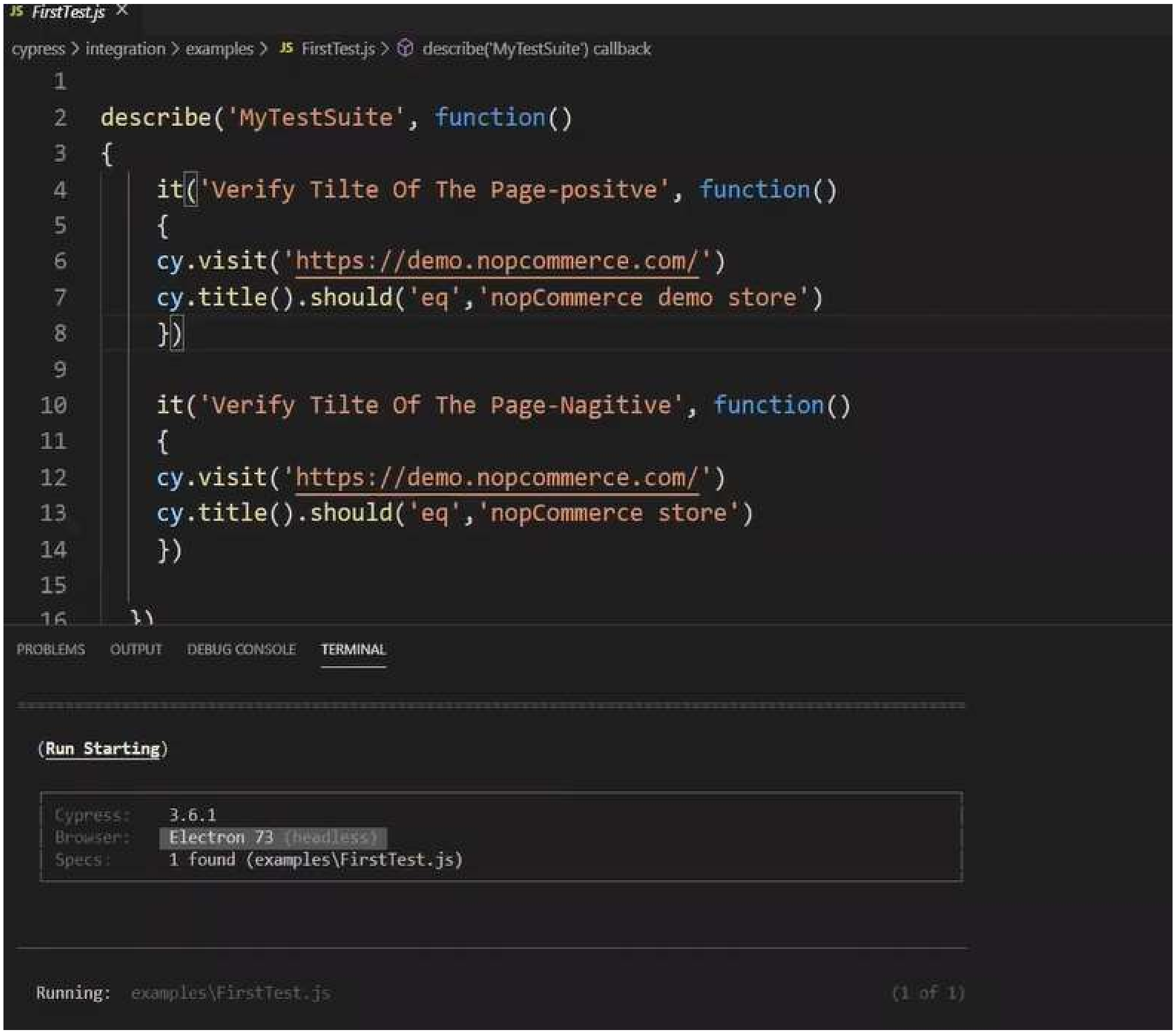
1. npx cypress run

   - Will run all test cases under examples folder under headless mode

2. npx cypress run --headed

   - Will run all test cases under examples folder in opened browser mode

3. Following image shows how the execution report shows in terminal window
4. By default Cypress will execute the test in Electron browser


FirstTest.js ×
cypress > integration > examples > FirstTest.js > describe('MyTestSuite') callback
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL

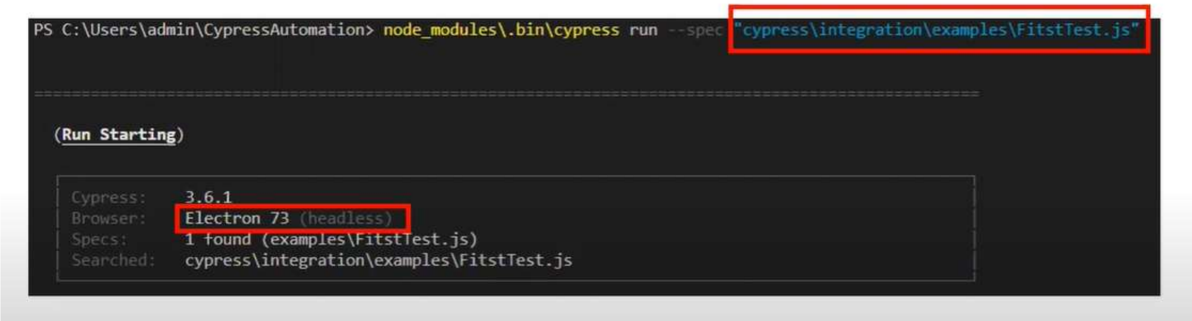
```

describe('MyTestSuite', function()
{
    it('Verify Tilte Of The Page-positve', function()
    {
    cy.visit('https://demo.nopcommerce.com/')
    cy.title().should('eq','nopCommerce demo store')
    })

    it('Verify Tilte Of The Page-Nagitive', function()
    {
    cy.visit('https://demo.nopcommerce.com/')
    cy.title().should('eq','nopCommerce store')
    })

  })
```

```
====================================================================================================

  (Run Starting)

  ┌────────────────────────────────────────────────────────────────────────────────────────────────┐
  │ Cypress:    3.6.1                                                                              │
  │ Browser:    Electron 73 (headless)                                                             │
  │ Specs:      1 found (examples\FirstTest.js)                                                    │
  └────────────────────────────────────────────────────────────────────────────────────────────────┘

────────────────────────────────────────────────────────────────────────────────────────────────────

  Running:  examples\FirstTest.js                                                          (1 of 1)
```

- **Run Specific Cypress Tests in Terminal**

  - To **Run Single test** under **examples** directory

```
node_modules\.bin\cypress run -spec "cypress\integration\examples\FirstTest.js"
```

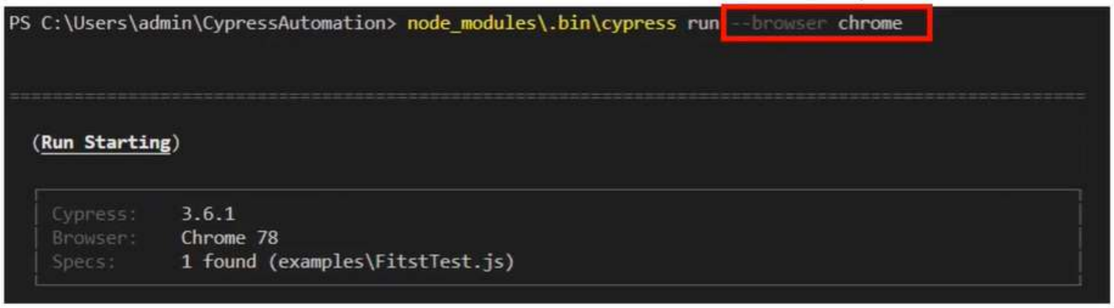


```
PS C:\Users\admin\CypressAutomation> node_modules\.bin\cypress run --spec "cypress\integration\examples\FitstTest.js"

====================================================================================================

  (Run Starting)

  ┌────────────────────────────────────────────────────────────────────────────────────────────────┐
  │ Cypress:    3.6.1                                                                              │
  │ Browser:    Electron 73 (headless)                                                             │
  │ Specs:      1 found (examples\FitstTest.js)                                                    │
  │ Searched:   cypress\integration\examples\FitstTest.js                                          │
  └────────────────────────────────────────────────────────────────────────────────────────────────┘
```

- **Execute the test in another browser Ex. Chrome**

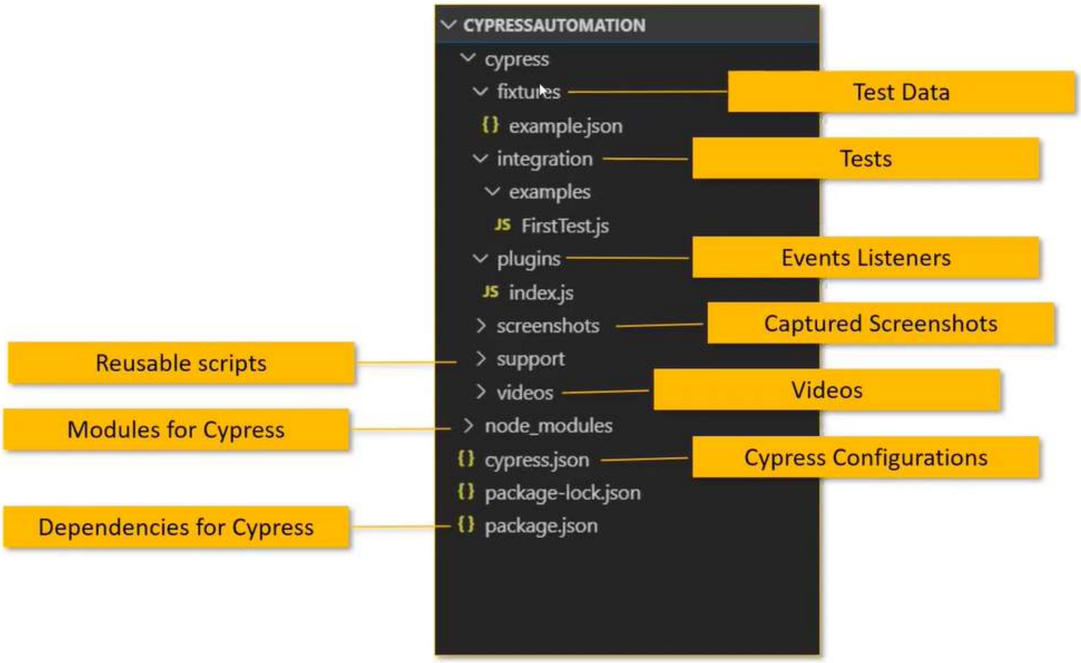
- To Run **All the tests** under **examples** directory using **Chrome**

```
node_modules\.bin\cypress run
```



```
PS C:\Users\admin\CypressAutomation> node_modules\.bin\cypress run --browser chrome

====================================================================================================

  (Run Starting)

  ┌────────────────────────────────────────────────────────────────────────────────────────────────┐
  │ Cypress:    3.6.1                                                                              │
  │ Browser:    Chrome 78                                                                          │
  │ Specs:      1 found (examples\FitstTest.js)                                                    │
  └────────────────────────────────────────────────────────────────────────────────────────────────┘
```

- **Folder Structure of Cypress based FW**

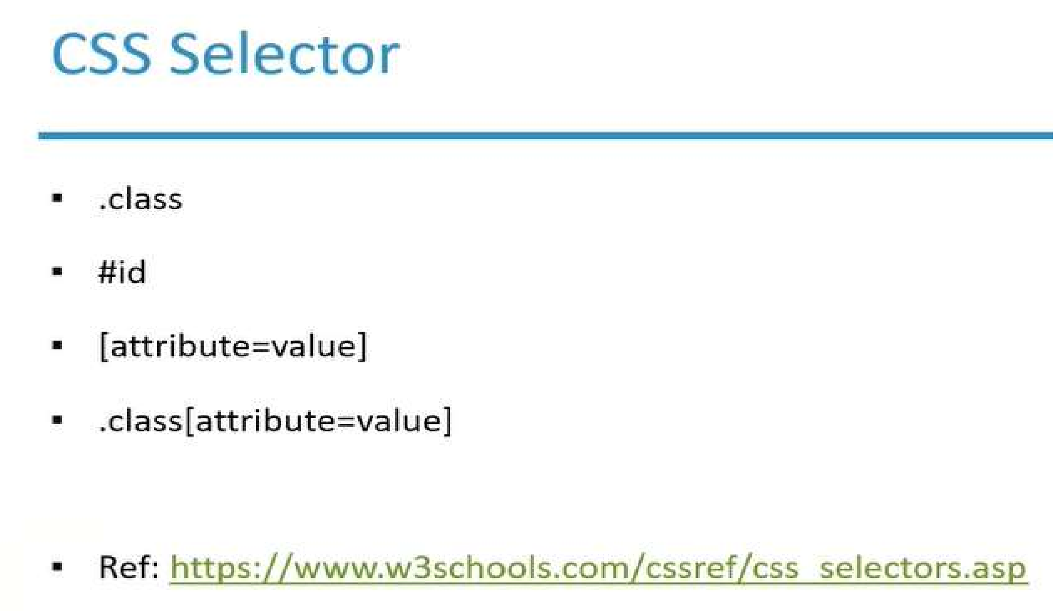


- **Cypress Locators**

Cypress only supports CSS selector to locate WebElement

get()

- **get**
- Get one or more DOM elements by selector.



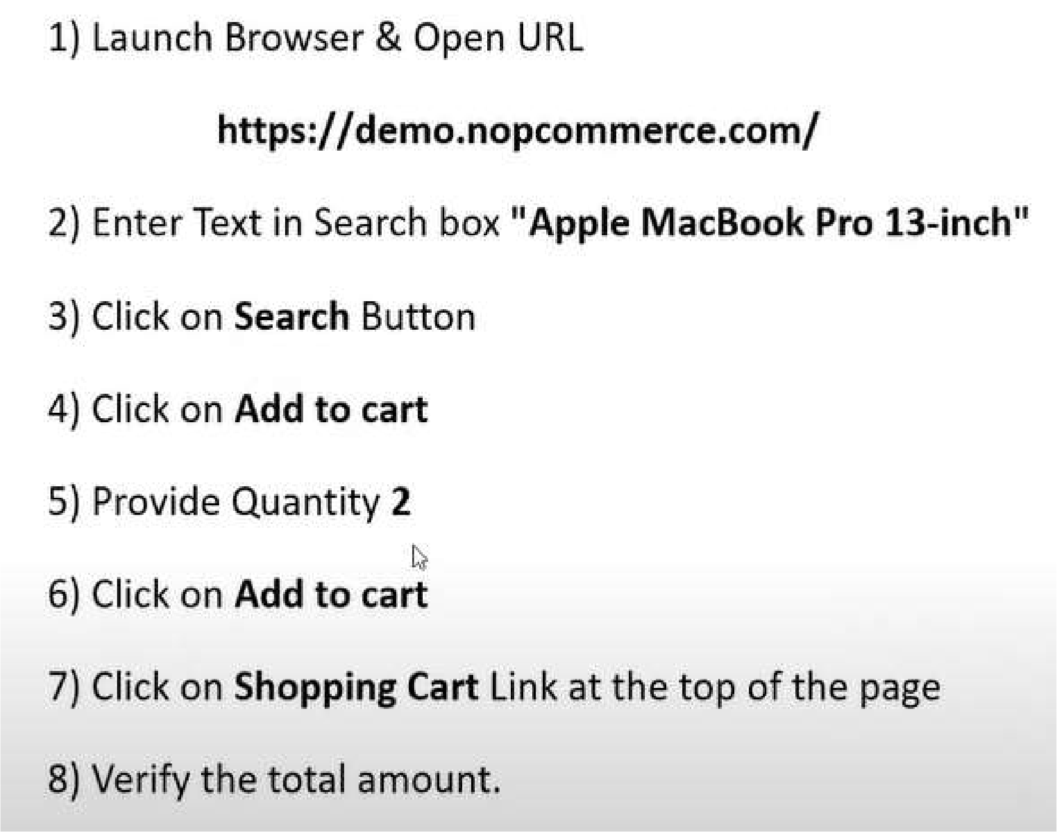
CSS Selector

- .class
- #id
- [attribute=value]
- .class[attribute=value]

- Ref: https://www.w3schools.com/cssref/css_selectors.asp

Url - https://www.w3schools.com/cssref/css_selectors.asp

- **End to end test case automation by Cypress**



1) Launch Browser & Open URL

https://demo.nopcommerce.com/

2) Enter Text in Search box **"Apple MacBook Pro 13-inch"**

3) Click on **Search** Button

4) Click on **Add to cart**

5) Provide Quantity **2**

6) Click on **Add to cart**

7) Click on **Shopping Cart** Link at the top of the page

8) Verify the total amount.

- **Cypress learning links - Source Youtube**
  - QA Camp - https://www.youtube.com/watch?v=IMx7Ng-T2mw&list=PLzDWIPKHyNmK9NX9_ng2IdrkEr8L4WwB0
  - SDET -

    https://www.youtube.com/watch?v=yM38NmF-LC4&list=PLUDwpEzHYYLu4jKg-rNSKH3aJeBinlPXp
  - Naveen Automation Labs

    https://www.youtube.com/watch?v=1bELNGm-_uw&list=PLFGoYjJG_fqoxTBJAUYw_poyBh0DI78HV