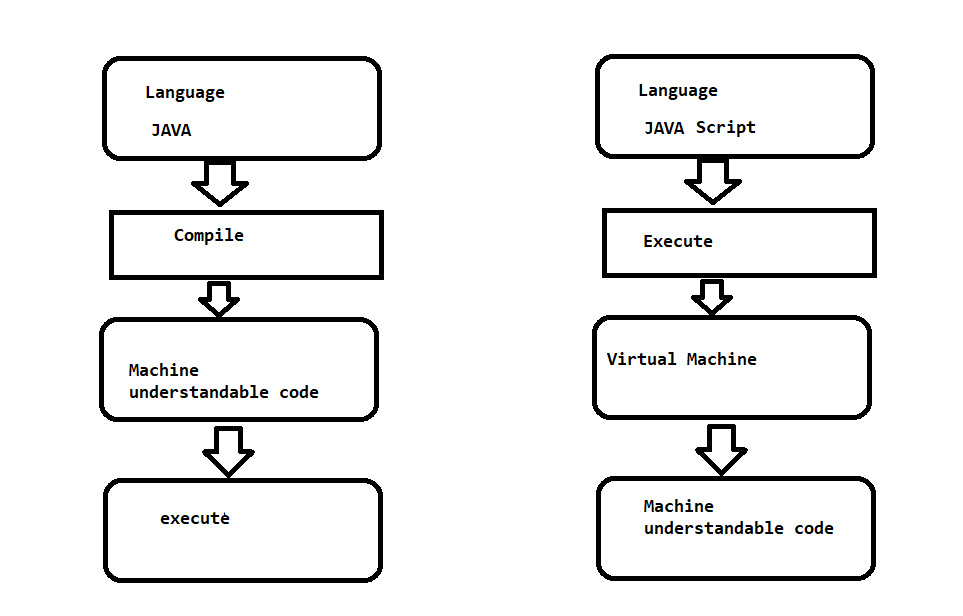


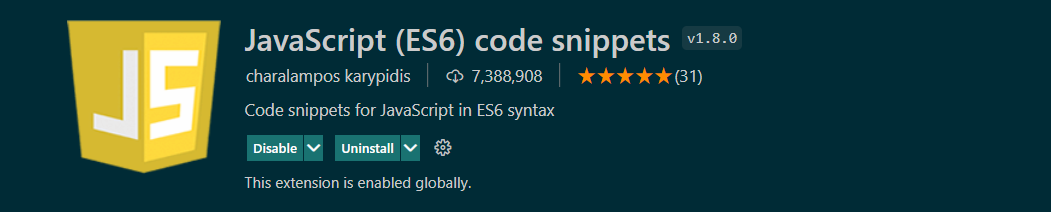
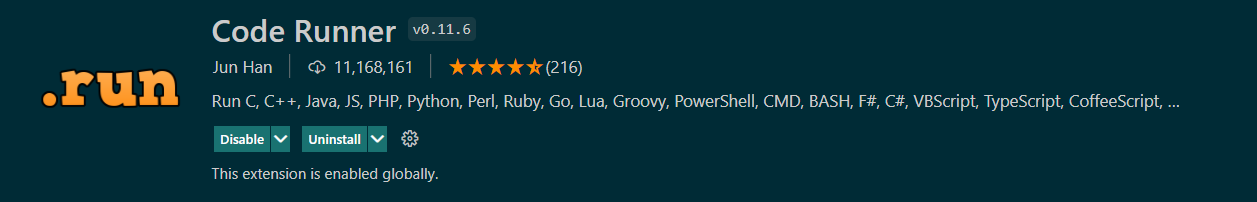
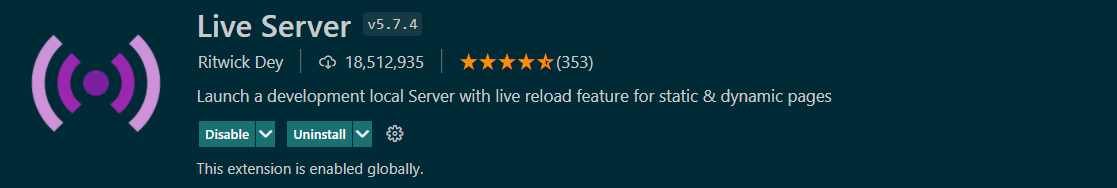
High level Difference



Java Script

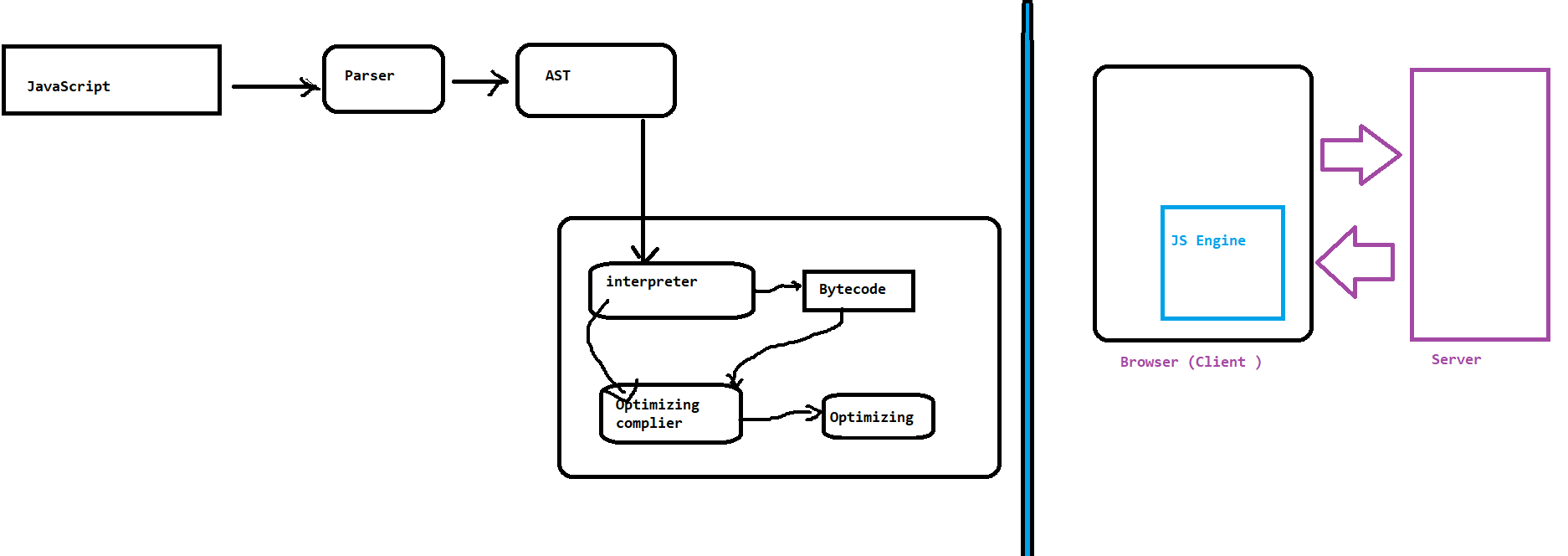
* High level and interpreted programming language
* Brendan Eich in 1995
* it is a programming language for web, now it can be used in non-browser environments also like Node JS

Visual Studio Code Editor

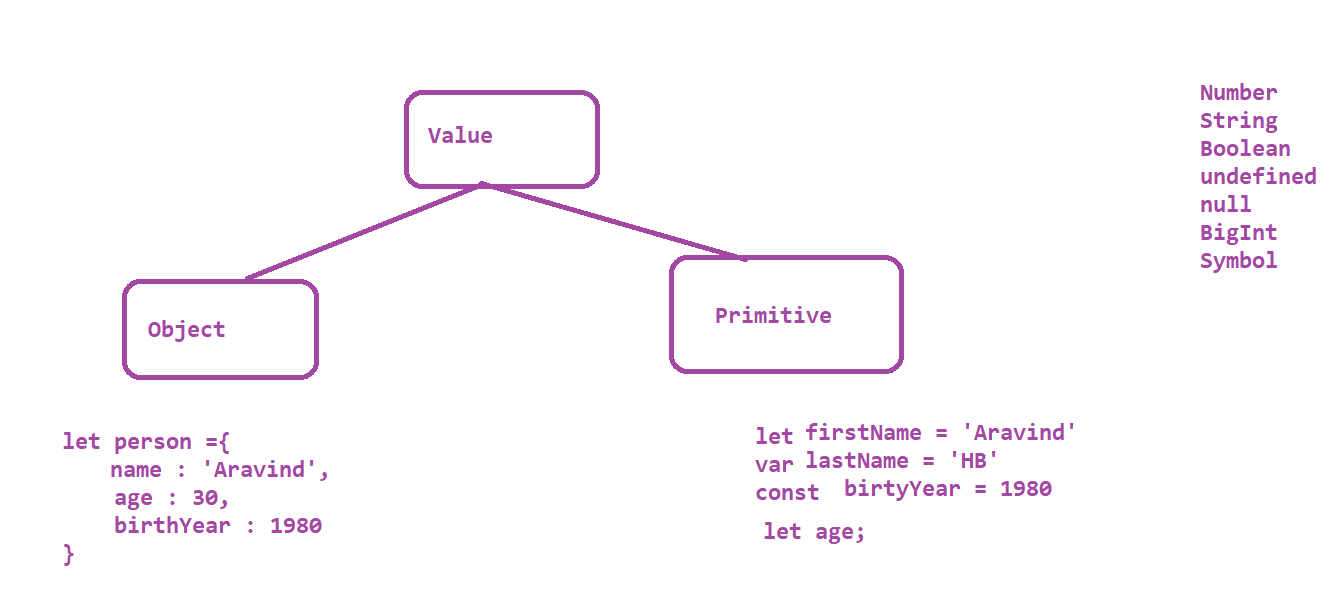
* Download the software from <https://code.visualstudio.com/download>
* Follow the installation Steps
* Select File -> Open Folder
* Use Extensions options (4th option from left panel ) to install the plugins
* To get Java script code snippet
  + 
* To Run any code in VSC
  + 
* To Execute JS on a Browser from VSC
  + 

Install Node JS software on the machine if you want to execute the Java script Program in non-browser environment

1. Download and install <https://nodejs.org/en/download/>
2. Once the installation is complete, on a new terminal
   1. node –v
   2. npm -v



Variables



let and var are the two ways of declaring a variable whose values can be altered later point of time

whereas, const is a variable whose value will never be altered in the program

Avoid using var as it’s a old way of declaring a variable.

Only difference between var and let is ,

let is a block scope and var is a function scope

Operators in JS

1. Arithmetic Operators
   1. Addition 🡪 +
   2. subtraction 🡪 -
   3. Multiplication 🡪 \*
   4. Division 🡪 /
   5. Module 🡪 %
   6. Exponential 🡪 \*\*
   7. Increment 🡪 ++
2. Assignment Operator
   1. Assign = x = y
   2. Add and assign += x+=y x = x + y
   3. Subtract and assign -= x-=y x = x- y
   4. Multiply and assign \*= x\*=y x = x\* y
   5. Divide and assign /= x/=y x = x/ y
   6. Module and assign %= x%=y x = x% y
   7. Exponential and assign \*\*= x\*\*=y x = x- y
3. Comparison Operator
   1. Check equal Value ==
   2. Check equal Value and type ===
   3. Check Not equal Value !=
   4. Check equal Value and type !==
   5. Greater than >
   6. Less than <
   7. Greater than or equal >=
   8. Less than or equal <=
   9. Ternary Operator ?:

4. Logical Operators

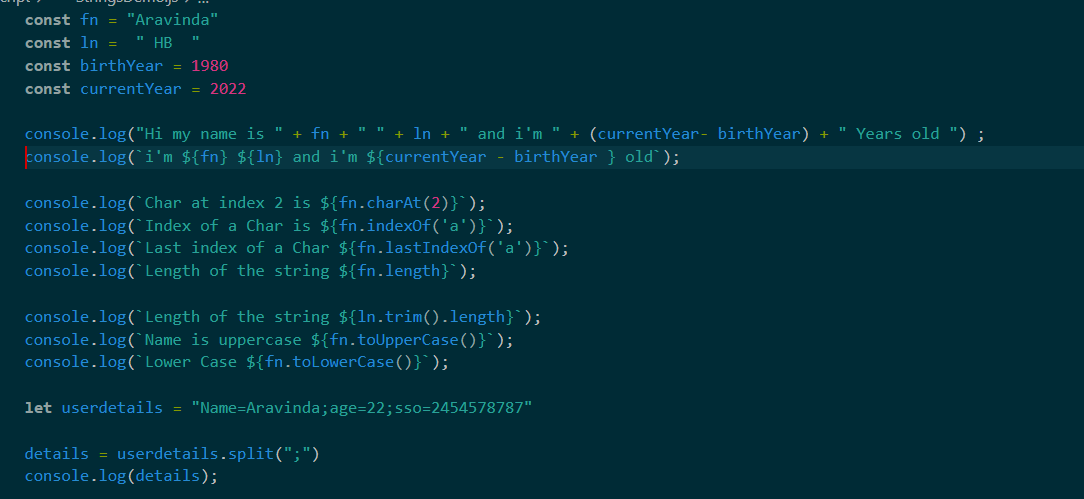
* Logical AND && => True only when both conditions are true
* Logical OR || => True when any one condition is true
* Logical NOT ! => true if the result is false

5. Type of operator

1. typeof

2.instance of

Strings



Conditional Statements

1. if
2. if.. else
3. if..elseif.. else
4. switch

Looping Statements

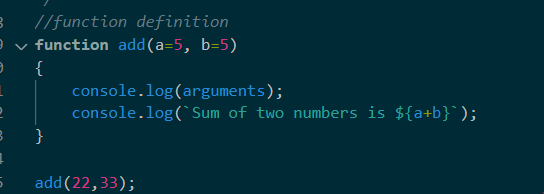
1. for
2. for each
3. while
4. do while
5. for in
6. for of

Arrays

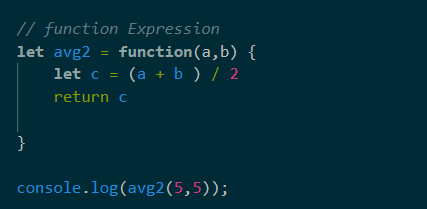
* push
* pop
* shift
* unshift
* delete
* splice
* slice
* concat
* join

Functions

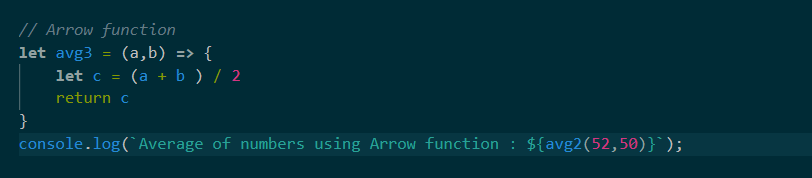
* Function with definition



* Function with Expressions or Anonymous Functions



* *Arrow Function*



Collections

MAP

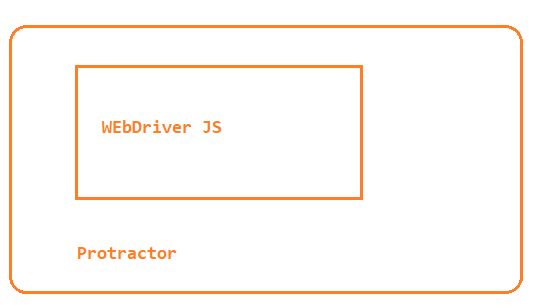
SET

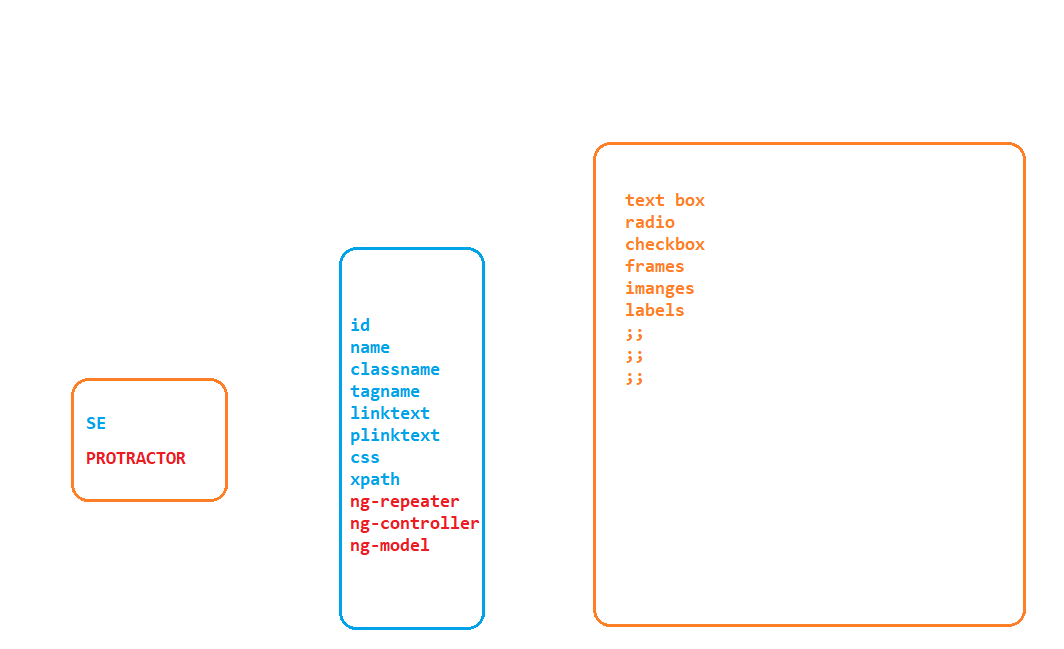
Objects

Protractor

protractor is a end-to end automation tool / framework used mainly for angular and angularJS applications

Protractor is a node JS application which is built on top of Webdriver JS

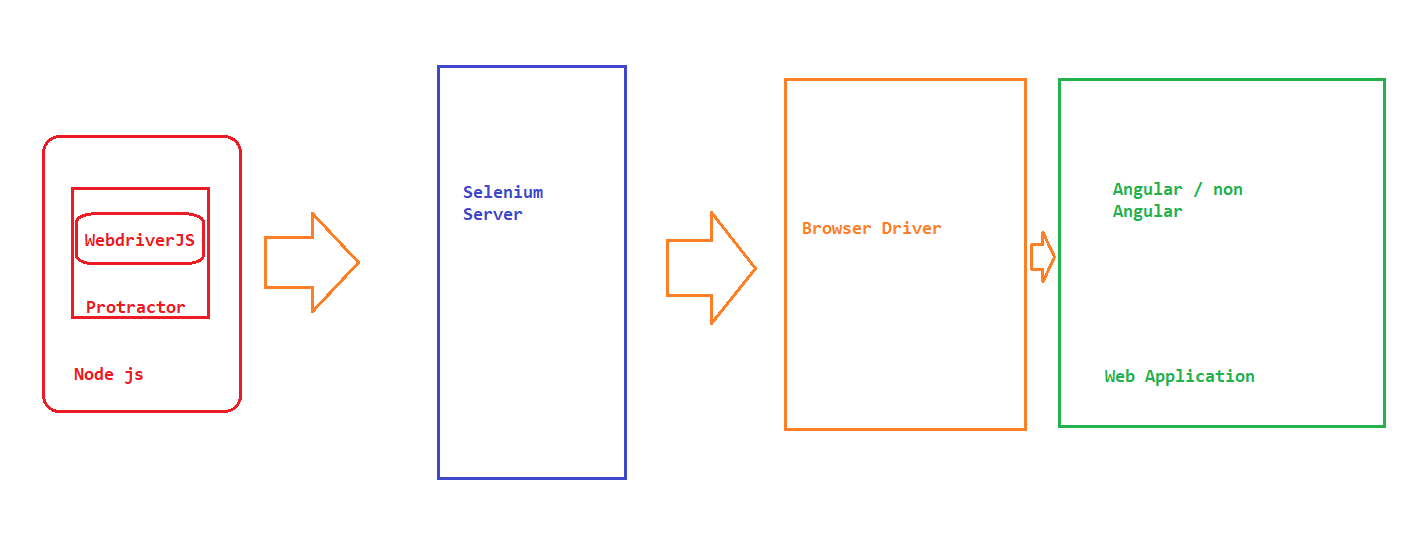




Why protractor

* Selenium Webdriver doesnot have any methods or features to locate elements based on angular specific properties to handle this we can go with protractor
* Sync issues can be easily handled in Protractor for all the Angular specific elements

How protractor works

**

Protractor Setup

1. Download and install node.js
2. Install protractor : [www.protractortest.org](http://www.protractortest.org)
   1. npm install -g protractor
3. *Update WebDriver manager* 
   1. webdriver-manager update
4. *Write Test: Sample Conf.js and Example.js file located at installation dir*
   1. C:\Users\<<USERDIRECTORY>>\AppData\Roaming\npm\node\_modules\protractor\example
5. *Copy paste the Conf .js and Example.js into your project* 
   1. *Example contain* 
      1. *describe block 🡺 One scenario*
      2. *it block 🡪 one Test in the Scenario*
6. *Open terminal and execute* 
   1. *protractor conf.js*

Note : Execution policy Error :

<https://stackoverflow.com/questions/4647429/powershell-on-windows-7-set-executionpolicy-for-regular-users>

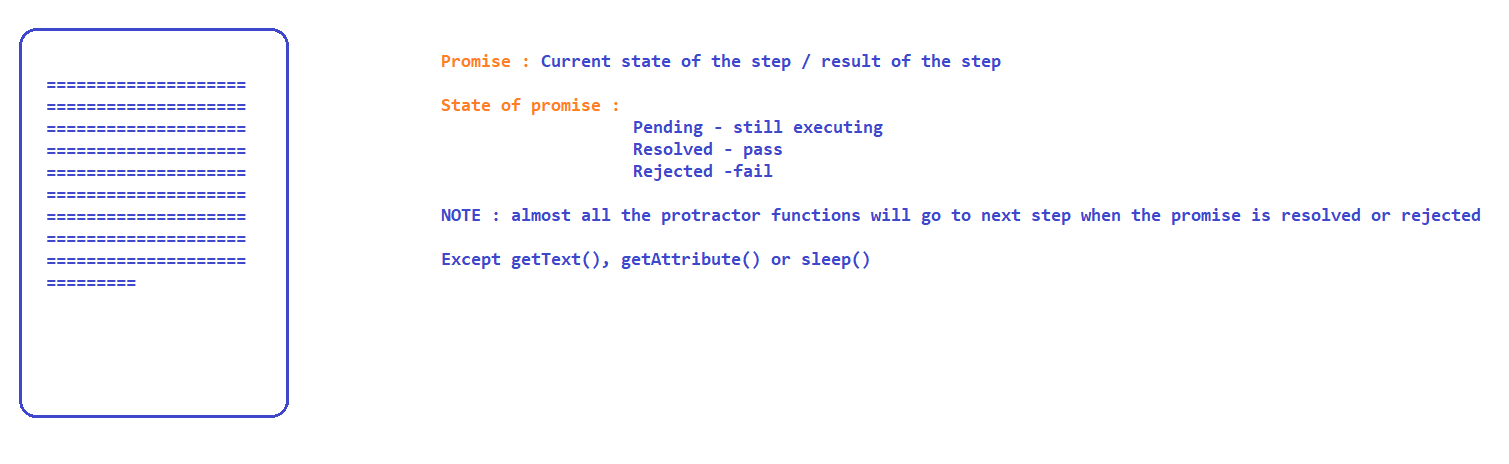
Async Nature of JS

execution will not wait until the line of code completes the execution. it will move to next line for execution

to handle the async nature we have to use call back function.

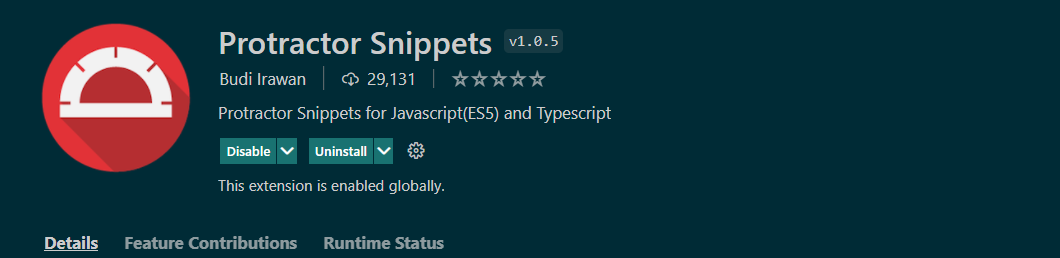
NOTE : almost all the protractor functions are synchronous (control will wait until the line of code completes its execution )in nature. except those which we read text or attribute from the application

Promise :



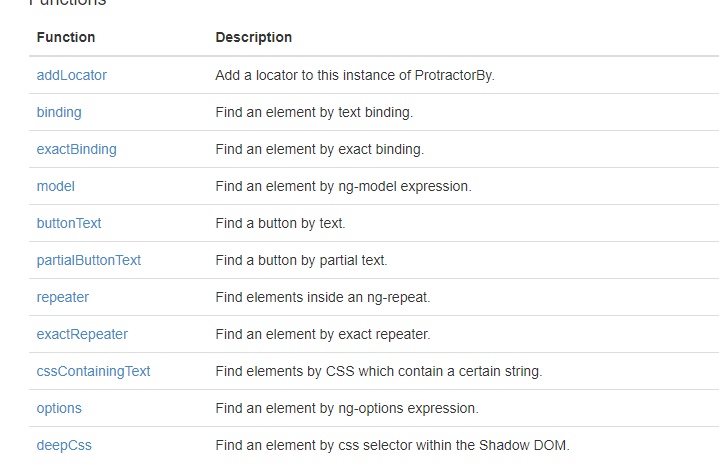
Plugins







Finding Elements in Protractor



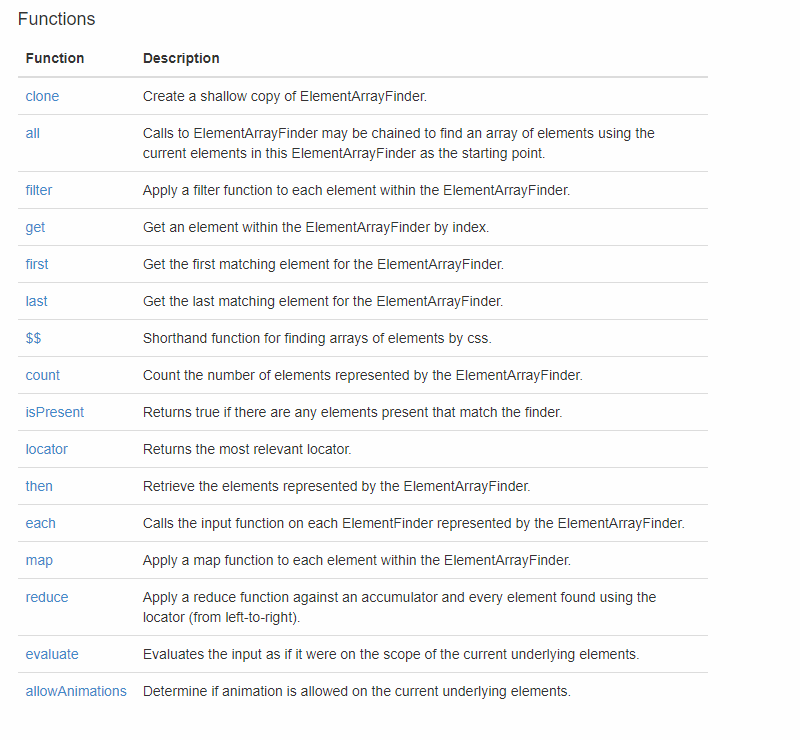
Validation in Protractor

<https://jasmine.github.io/>

Validations can be performed using expect commands in Jasmine : <https://jasmine.github.io/tutorials/your_first_suite>

Playing with more than one element

we have to use elemenet.all method to play with more than one element. and inside element.all we have below functions which helps to play the validations or further calculations



Chain Locators

1. Finding the element inside the element

Capturing Screenshot on Failures:

1. *Go to npm manager website* : <https://www.npmjs.com/>
2. *Search for protractor jasmine2 screenshot package :* <https://www.npmjs.com/package/protractor-jasmine2-screenshot-reporter>
3. *Follow the instructions to install the plugin* 
   * npm i protractor-jasmine2-screenshot-reporter
4. *Verify node\_module folder is created and it has the above package installed*
5. *In your Protractor configuration file(conf.js), register protractor-jasmine2-screenshot-reporter in*
6. *Execute the Test and verify the screenshot captured under target folder*

Allure Reports

1. *Go to npm manager website* : <https://www.npmjs.com/>
2. *search for jasmine-allure-reporter :* [*https://www.npmjs.com/package/jasmine-allure-reporter*](https://www.npmjs.com/package/jasmine-allure-reporter)
3. *Install the plugin using :* npm i jasmine-allure-reporter
4. *Update conf.js as per the instruction*
5. *Execute the test and verify xml output generated under allure-reports folder*
6. *To get the html report, install allure command line* 
   * <https://www.npmjs.com/package/allure-commandline>
7. *Execute allure serve “location of xml file”*

HTML Reports using protractor html reporter:

1. *Go to npm manager website* : <https://www.npmjs.com/>
2. *search for protractor-html-reporter-2:*
   * <https://www.npmjs.com/package/protractor-html-reporter-2>
3. *Install the plugin using :* npm i protractor-html-reporter-2
4. *Update conf.js as per instruction*
5. *Update opPrepare with BELOW CODE TO GET XMLRESULTS*

var jasmineReporters **=** require('jasmine-reporters');

jasmine.getEnv().addReporter(**new** jasmineReporters.JUnitXmlReporter({

    consolidateAll**:** true,

    savePath**:** './',

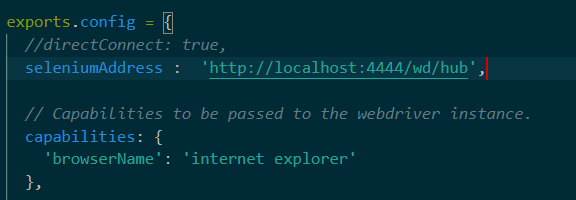
    filePrefix**:** 'xmlresults'

}));

1. *Execute the test*

Execute test on a different browser

* Download the driver executable Webdriver-manager update –ie
* Webdriver-manager start
* Update conf.js



Automate non Angular applications

browser.waitForAngularEnabled(false);

Handling Sync issues

- Create an object to expected conditions class

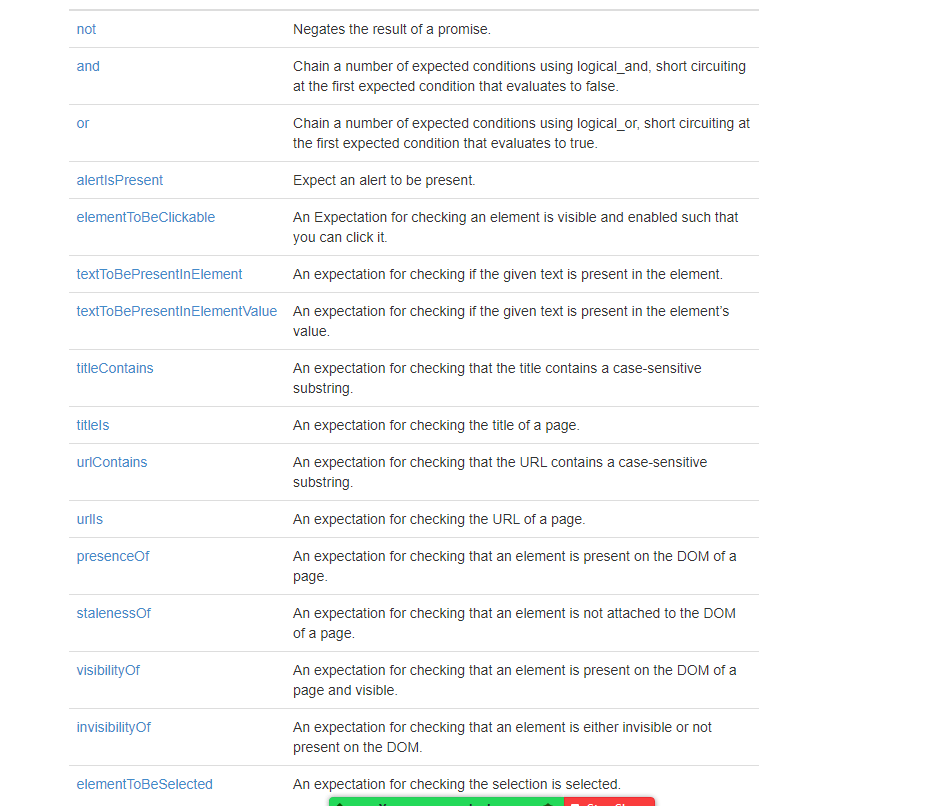
**var** expectedConditions = protractor.ExpectedConditions;

      browser.wait(expectedConditions.visibilityOf(element(by.xpath("//button[@class='\_2KpZ6l \_2doB4z']"))));

      element(by.xpath("//button[@class='\_2KpZ6l \_2doB4z']")).click();

    });

- Inside expected conditions we have n-number of methods to play with different sync issues



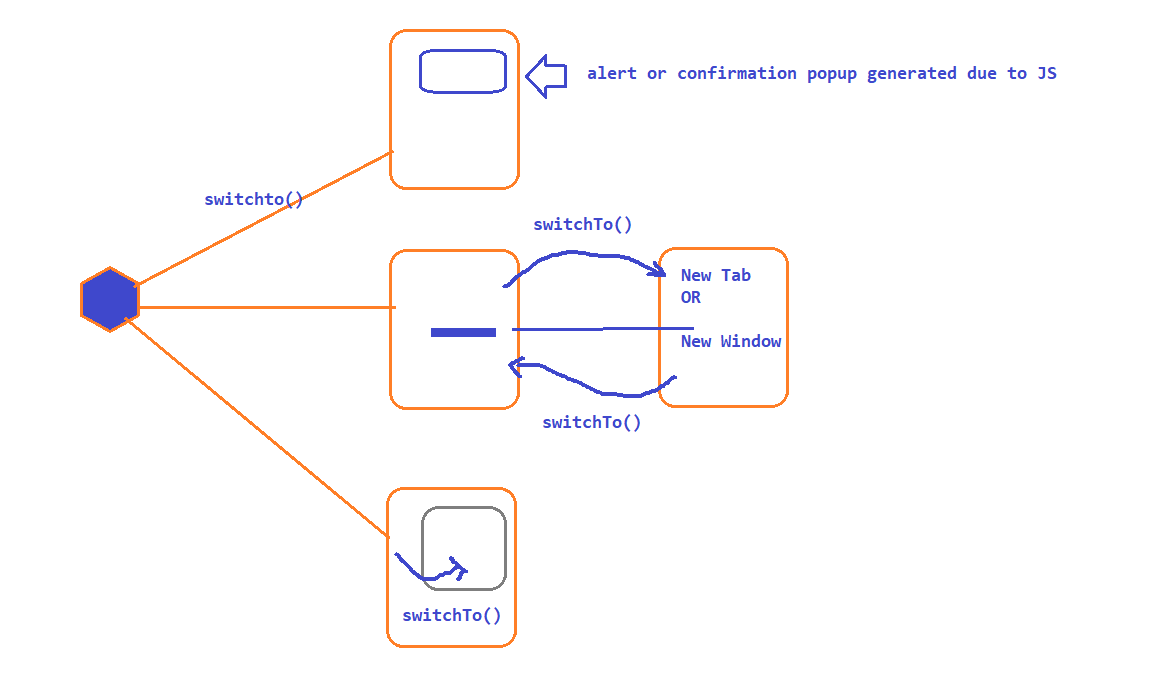
Browser Actions

browser.driver.manage().window().maximize();

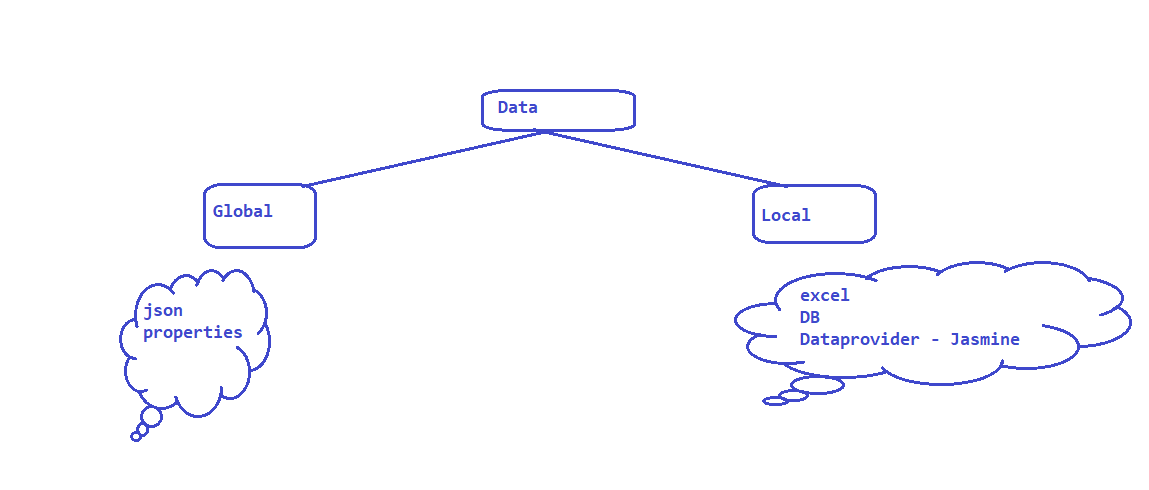
Actions

browser.actions().mouseMove(element(by.xpath("//div[text()='Home']"))).perform();

Switchto



Data driven Testing in protractor



Using Data provider given by Jasmine

1. download jasmine data provider from npm

2. Install data provider plugin : npm i jasmine-data-provider

3. follow the instructions

Executing Test Suite

- update conf.js and add

suites : {

    smoke : ['./smoke/\*.js'],

    regression : ['./ActionsDemo.js','./ChainLocatorsDemo.js']

  },

- run the test using :

protractor .\conf.js --suite=regression