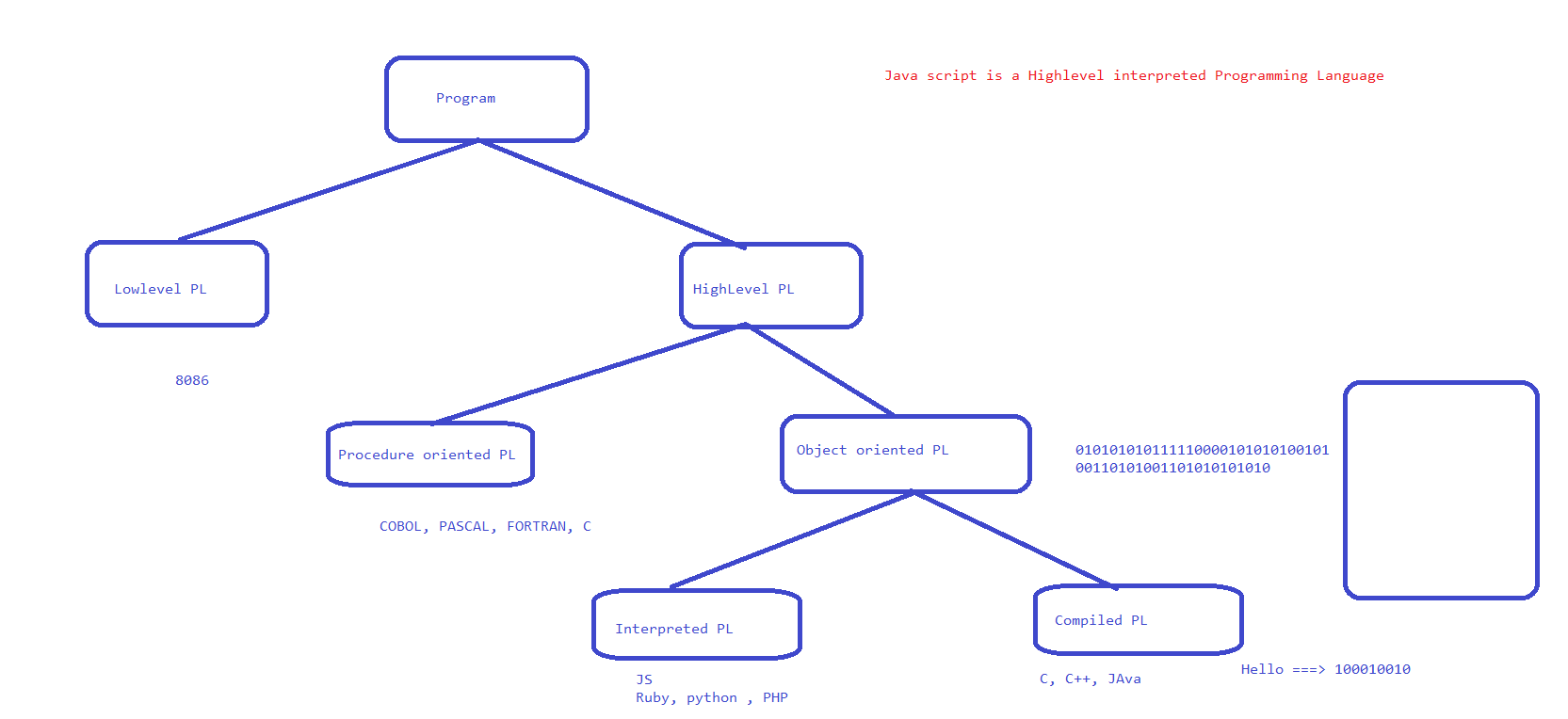
# Java Script

1. History
2. Different from JAVA or OOP
3. Extensions – JS
4. Variables
5. Operators
6. Data types
7. Strings
8. Conditional Statements
9. Looping Statements
10. Arrays
    1. push
    2. pop
    3. shift
    4. unshift
    5. delete
    6. splice
    7. slice
    8. concat
    9. join
11. Functions
    1. function definition
    2. function expression
    3. Arrow Functions
12. Collections
13. OOPS

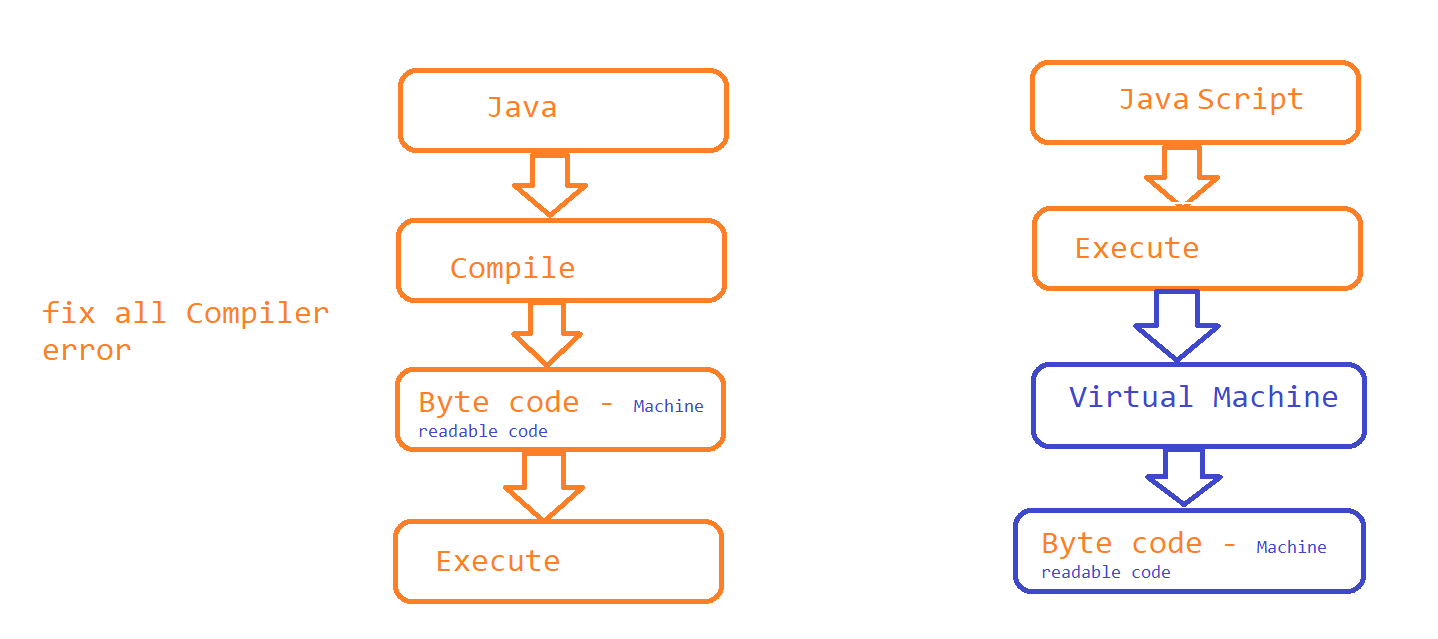
# Protractor

1. Why ????
2. How this tool is built ?
3. What are the features ?
4. How protractor works ?
5. Protractor Setup
   1. Introduction to maven
   2. Maven life cycle
   3. pom.xml – introduction and useage
   4. updating pom file to configure protractor
6. Async Nature of JS
7. Plugins
8. Jasmine FW
9. Finding Elements in protractor
10. Working with more than one elements in protractor
11. Chain Locators
12. Capturing Screenshots of Failures
13. Reporting
    1. Junit Reports
    2. HTML
    3. Allure Reports
14. Execute Tests on different browser
15. Automate non-angular applications using protractor
    1. Handling sync issues
    2. Browser actions
    3. Actions
    4. switchTo
16. Data driven Testing in protractor
    1. global data
    2. local data
17. POM – Framework
18. Executing the Test Suite
19. Jenkins Execution
    1. configure
    2. auto trigger build

Java script

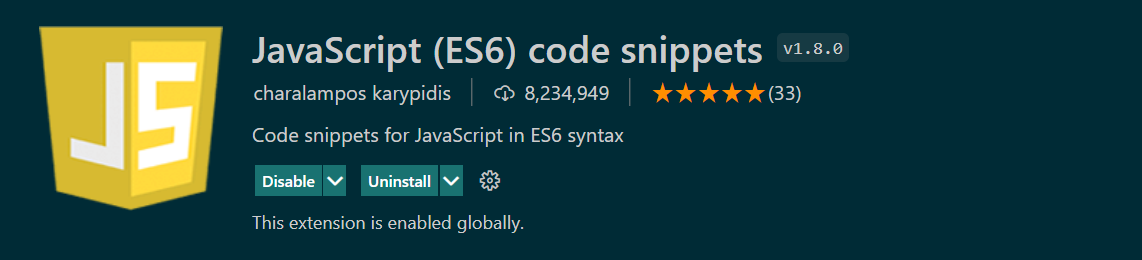
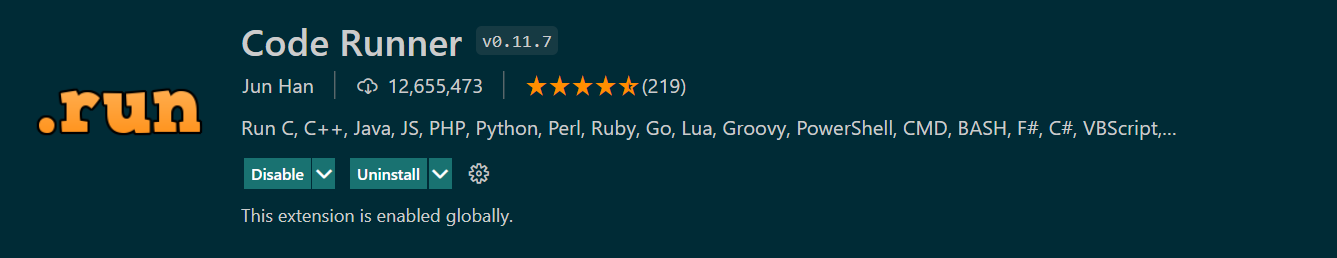
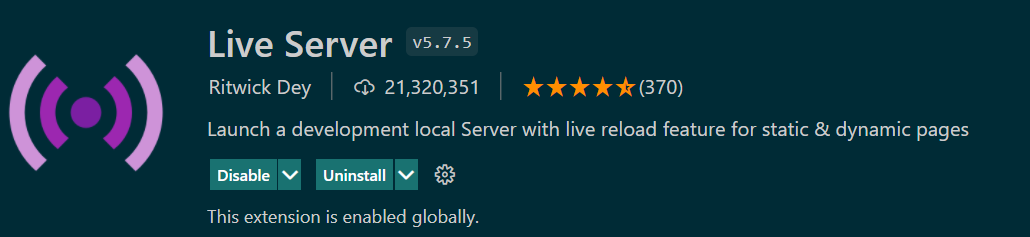


Work flow

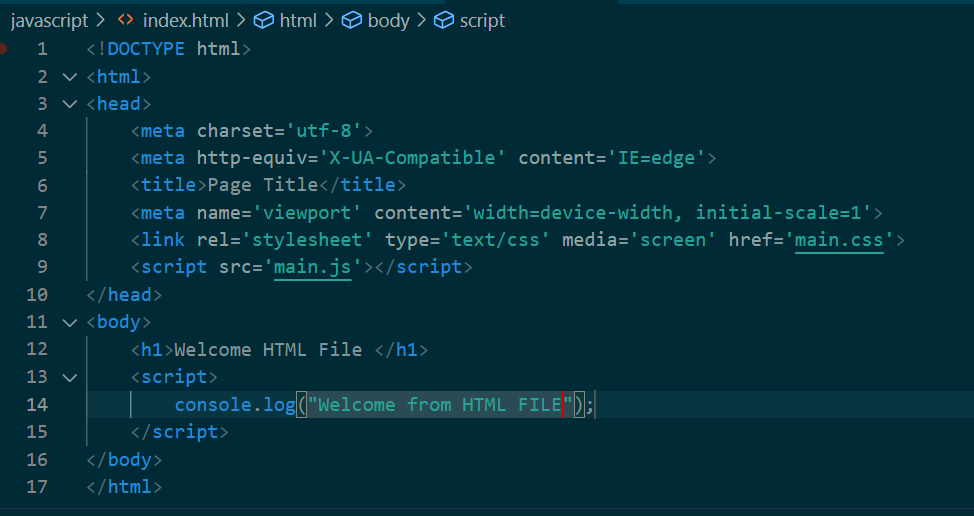
****

* Java Script was introduced to market in the year 1995 by Bredan Eich.
* Java script is a programming language for web
* Latest ERA, Java script can also be used in a non browser environment ie node.js application
* Java script is made as ECMA Standard specification
* 1999 – Ajax is released
* Node.js
* Java Script can be used for Developing
  + UI
  + Web
  + Mobile
  + Desktop
* Java script Runs on every browser
* <https://en.wikipedia.org/wiki/JavaScript_engine#:~:text=A%20JavaScript%20engine%20is%20a,every%20major%20browser%20has%20one>

Visual Studio Code Editor

* Download the software from <https://code.visualstudio.com/download>
* Follow the installation steps
* Install Extensions / plugins
* Java Script code snippet
* 
* Code Runner
* 
* Live Server
* 

## Install Node JS software on your machine if you want to execute java script program in a non-browser environment

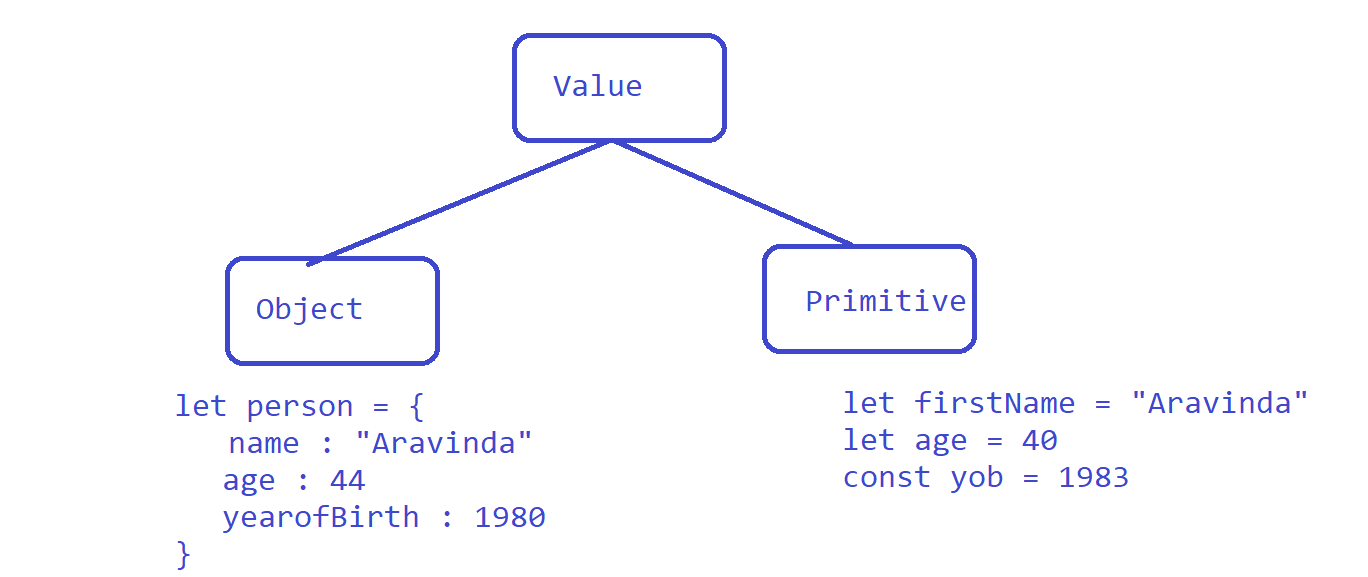
* Download and install <https://nodejs.org/en/download/>
* Once installed open command prompt and type,
  + node –v
  + npm –v
* NOTE: IF YOU HAVE ANY ADMIN RESTRICTIONS FOLLOW THE WORK ARROUND
* create a html file and execute java script inside the html file
  + Writing HTML File
  + create a file with any name and .html as extension
  + Crrl + Space and select HTML Sample
* HTML File with JS Code inside HTML File
* 
* HTML File with JS Code inside separate Java Script File

Variables

Variables are used to store the value. A value can be categorized into primitive or object.

JS is not strictly type caste programming language, we can use var, let or const to store any type of values.

NOTE : Avoid using var as its a old way of declaring a variables



Operators in Java Script

# Arithmetic Operators

* Addition 🡪 +
* subtraction 🡪 -
* Multiplication 🡪 \*
* Division 🡪 /
* Module 🡪 %
* Exponential 🡪 \*\*
* Increment 🡪 ++

# Assignment Operators

* Assignment = x=y
* Add and assign += x+=y x = x + y
* Subtract and assign -= x-=y x = x - y
* Multiply and assign \*= x\*=y x = x \* y
* Divide and assign /= x/=y x = x / y
* Module and assign %= x%=y x = x % y
* Exponential and assign \*\*= x\*\*=y x = x \*\* y

# Comparison Operators

* check equal value == x==y
* check equal value & Type === x===y
* check not equal value != x!=y
* check not equal value, Type!== x!==y
* Greater than > x>y
* Less than < x<y
* Greater than or Equal >= x>=y
* Less than or Equal <= x<=y
* Ternary Operator ?: (x>y)?stmts1:stmts2

# Logical Operators

* Logical AND && => true, when both Operands returns true
* Logical OR || => true, when any one Operands returns true
* Logical Not ! => Inverse of the result

# Type of Operators

* typeof – returns the data type
* instanceof

# Primitive Data types

1. Number
2. String
3. Boolean
4. Undefined
5. Null
6. Symbol
7. BigInt

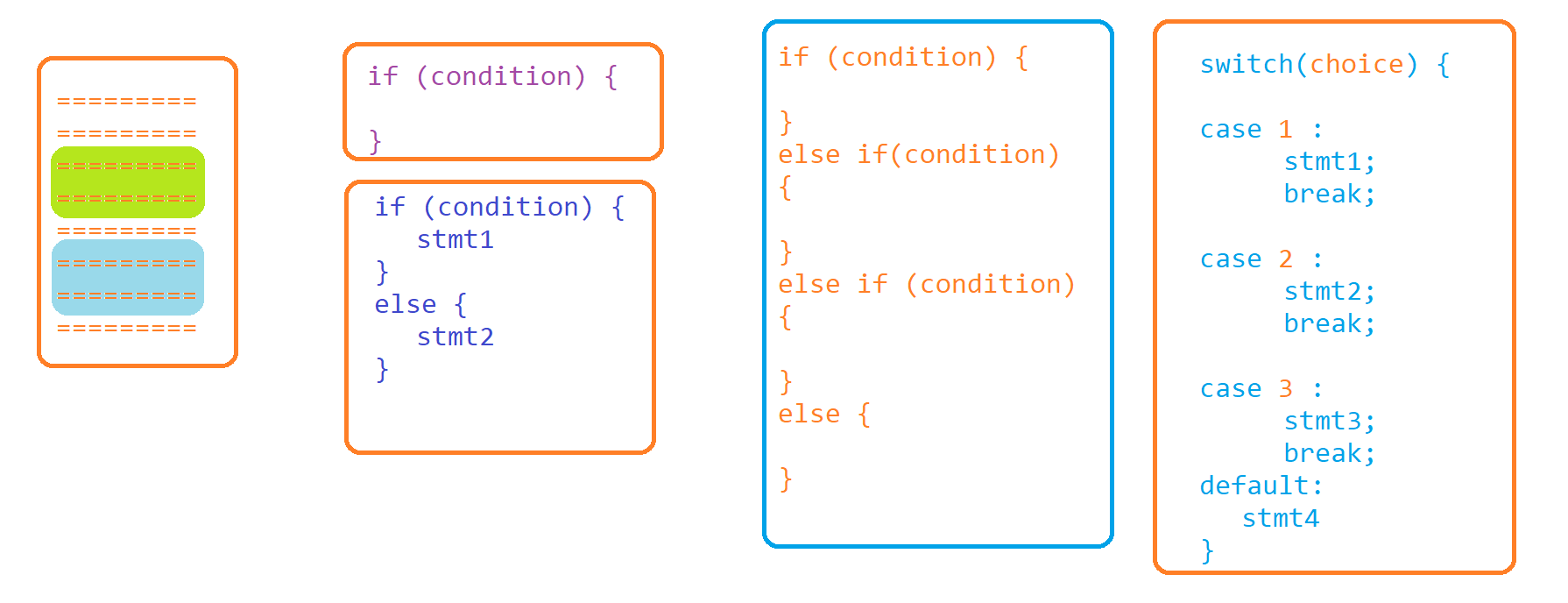
Strings

* String construction using String concatenation
* String construction using back-tick notation or template literal
* EX :



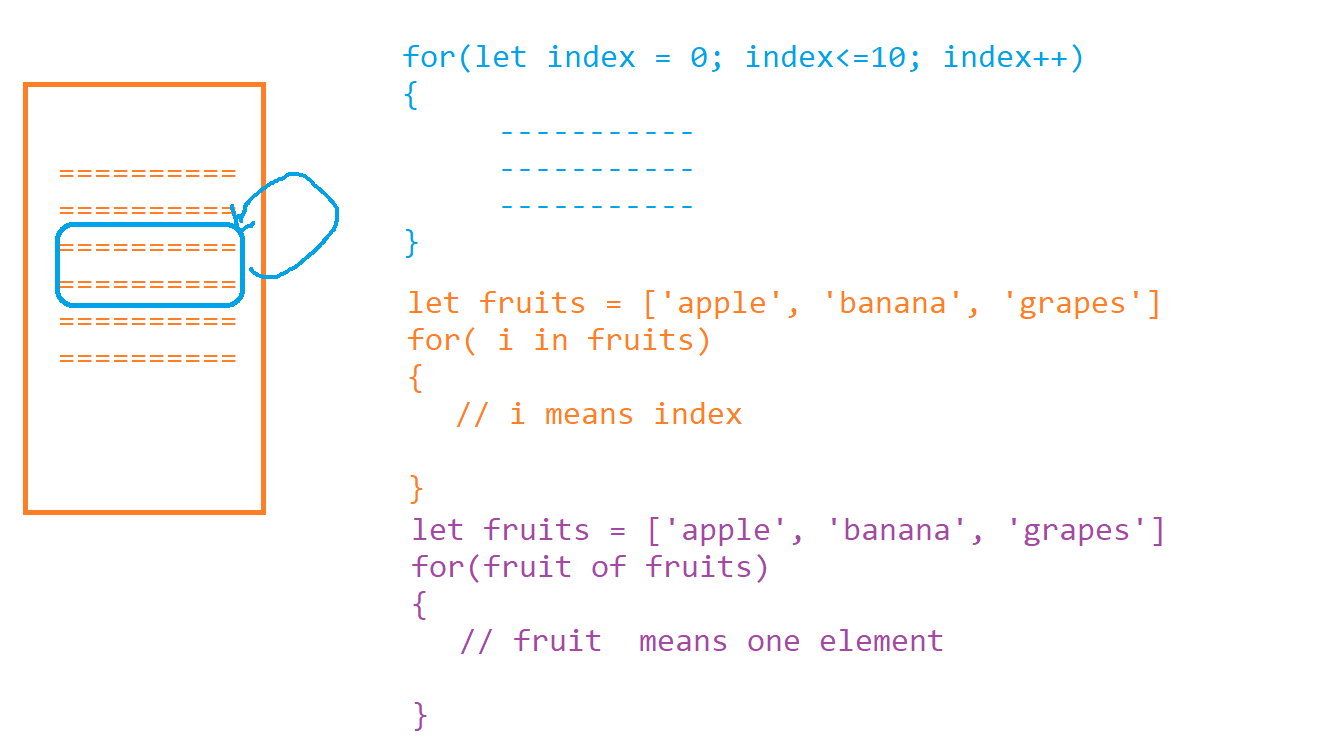
Conditional Statements

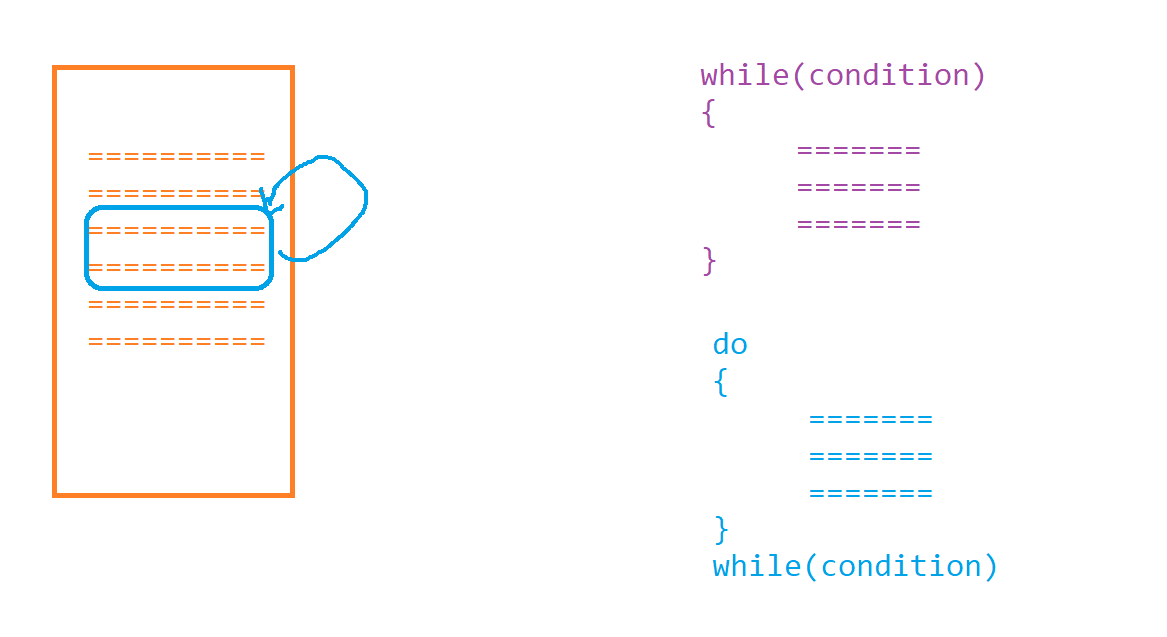
* if
* if.,else
* if.,elseif, else..
* switch



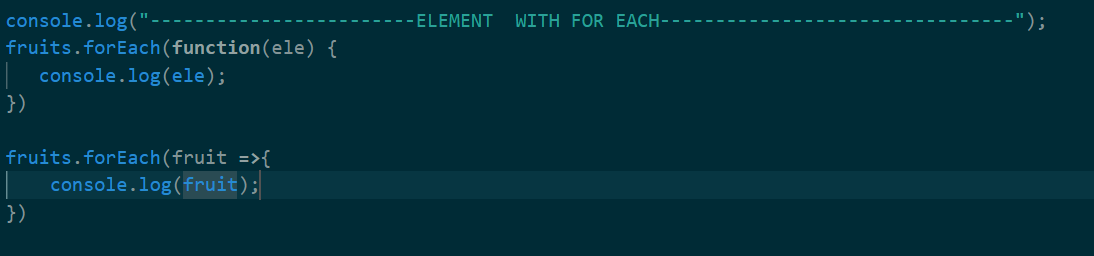
Looping Statements

* for
* for in
* for of
* for each
* while
* do while





**Foreach loop**

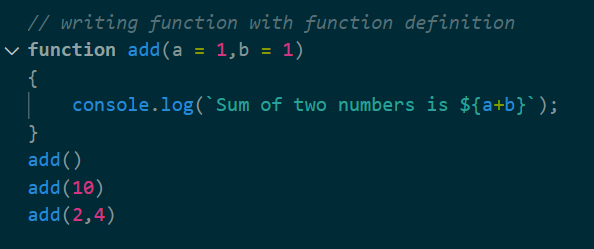


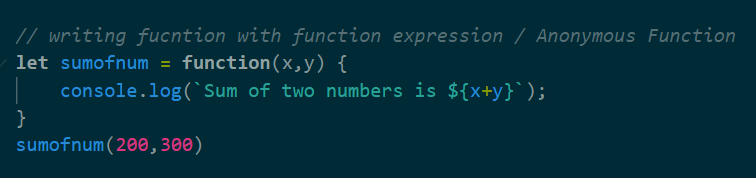
Arrays

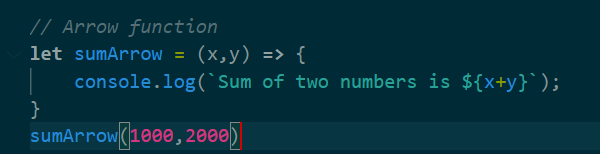
* push – insert the element at the end
* pop – delete the element at the end
* unshift – insert the element in the beginning
* shift – delete the element in the beginning
* delete[index] – delete the element with specified index and retain the index
* splice – delete or insert the elements in the original array
* slice - delete or insert the elements not in the original array, it retains the array as it is
* concat - to combine 2 arrays
* join – to convert array to string with some delimiter
* 2Darrray

Functions

* Functions with definitions



* Function with Expression OR Anonymous Function 
* Arrow functions



Collections

* Map
* Set

Classes and Objects

Protractor Setup

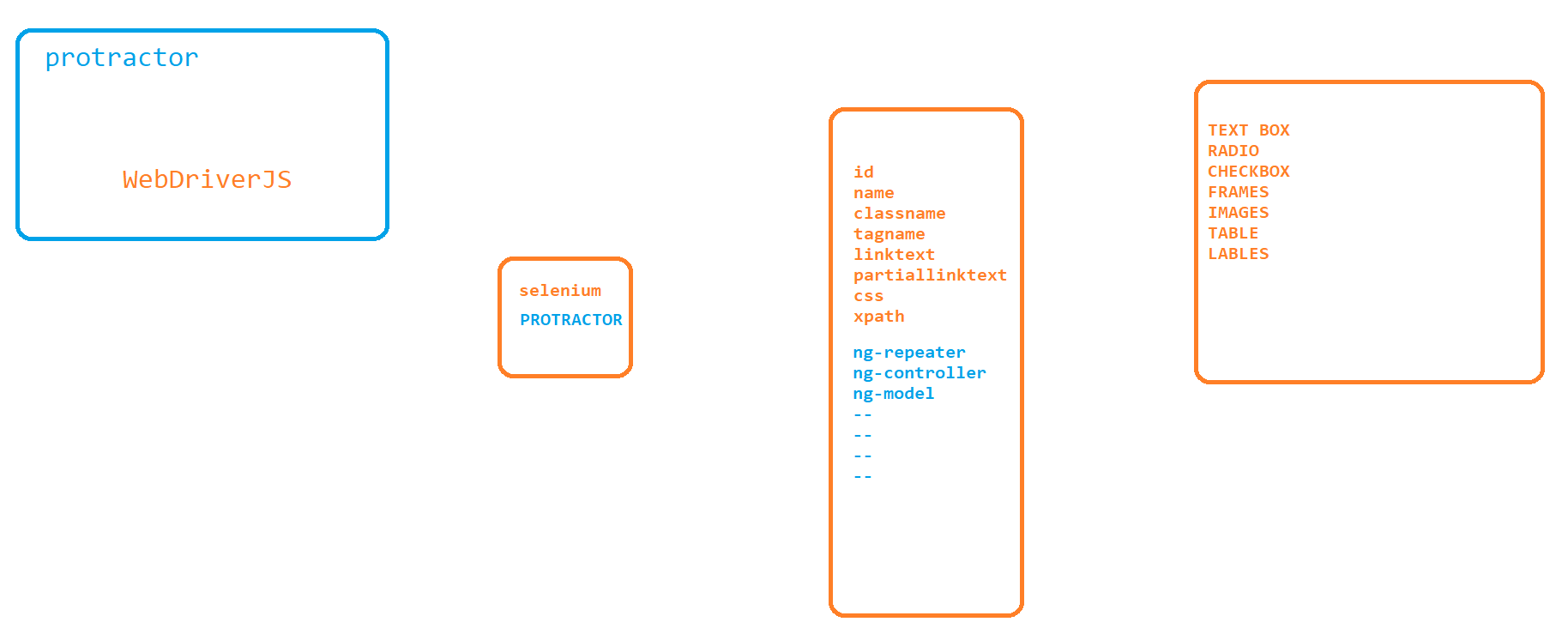
1. Download and install node.js with admin credentials
2. Install Protractor - <http://www.protractortest.org/#/>
   1. npm install -g protractor
3. Install webdriver-manager
   1. webdriver-manager update
4. Write a Test and configuration file from the official document
5. Open terminal and execute
   1. protractor conf.js

Drawbacks of Selenium

* Selenium is not suitable for angular Applications, because complex coding is required
* In Angular applications client side validations are more.
* Angular specific HTML Tags, extra HTML Tags, attributes which selenium cannot handle

Protractor

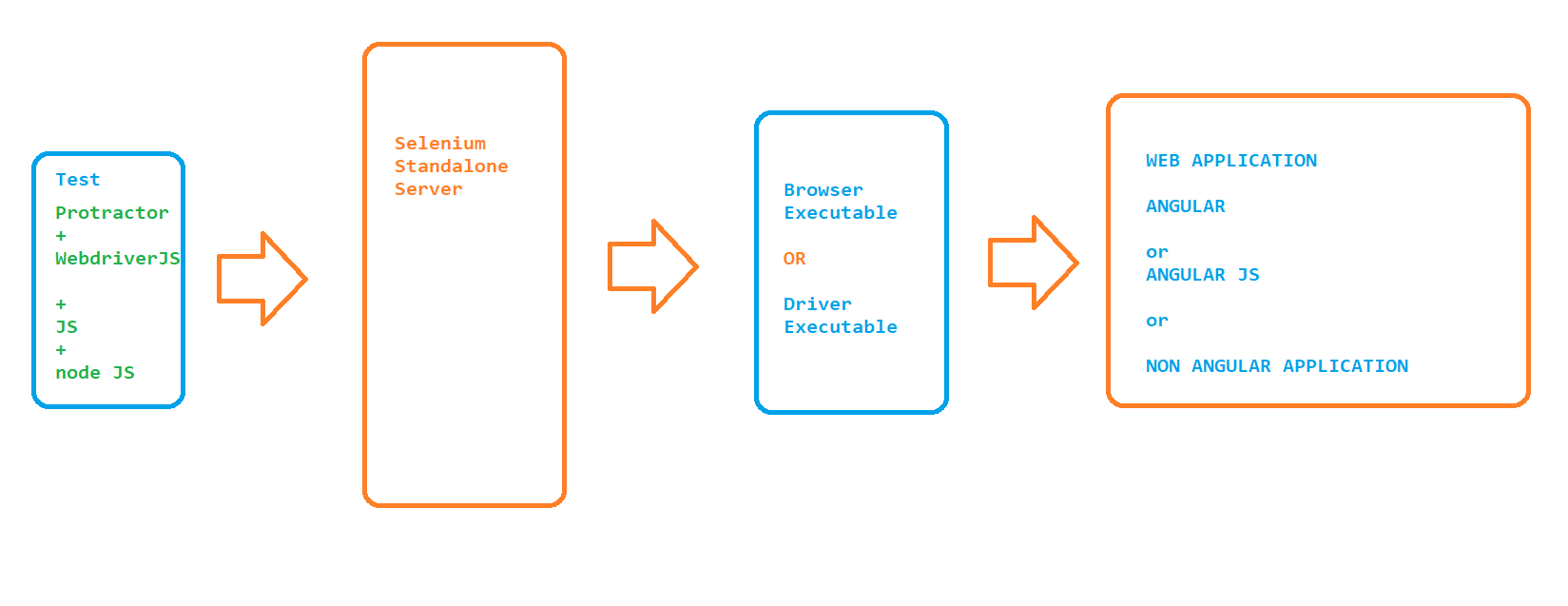
Protractor is a and-to-end automation tool for angular specific applications



Why Protractor

* Advanced then selenium because it is build on selenium with extra identifiers to locate angular specific elements
* Sync issues can be easily handled in Protractor
* Dynamic elements present in the webpage can be handled easily

How Protractor works

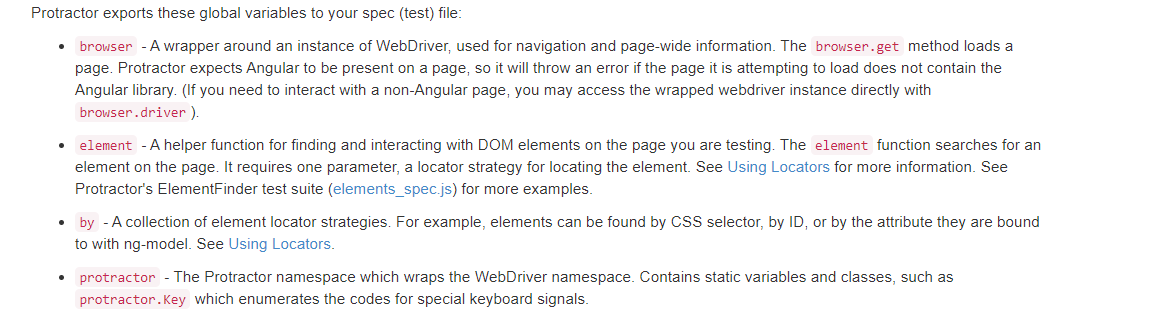


Understand the First Test

describe block - represents the scenario

it block – represents the test

Global Variables



Asynchronous nature of JS

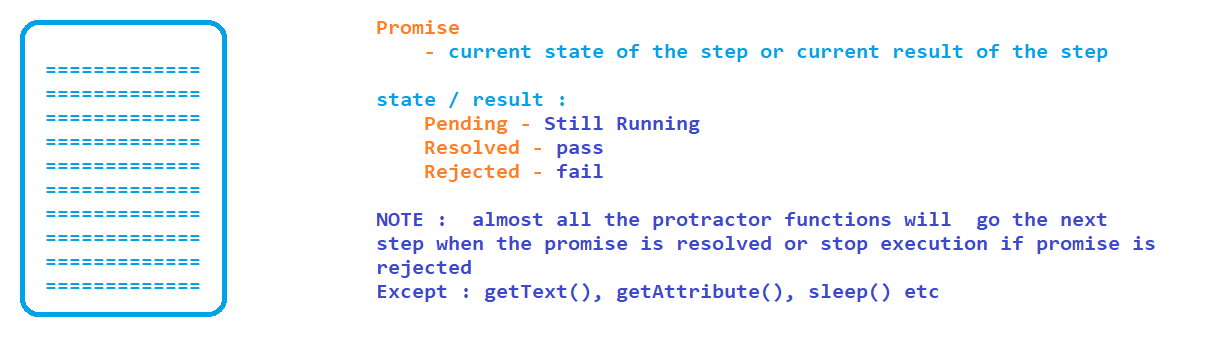
Execution will not wait until the line of code completes the execution (in case of sleep or waiting for other resources), it will go and execute the next line.

NOTE : almost all the protractor functions are synchronous in nature

Except those which you read from the application

Ex : getText, getAttribute etc

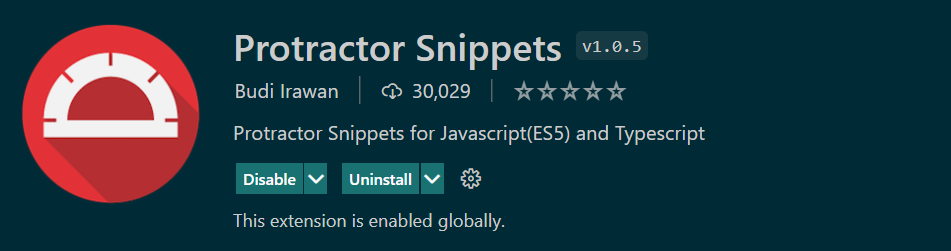
# **Promise :**

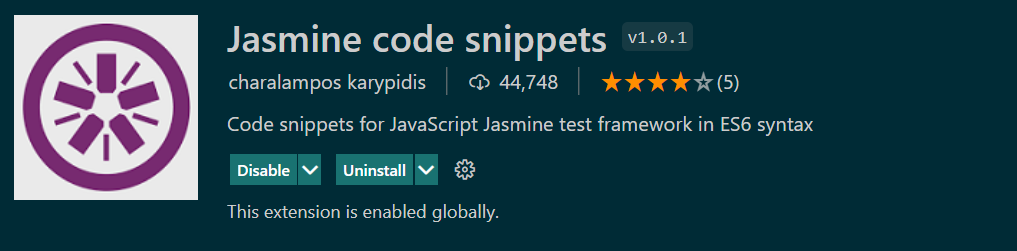


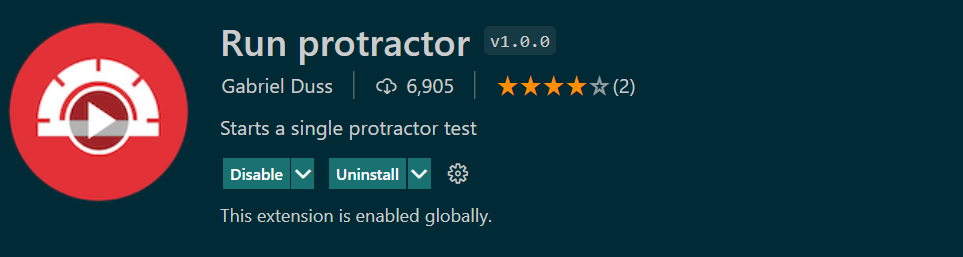
At any given point of time if we want to have synchronous behavior then we have to resolve the promise.

To resolve the promise use then() and a callback function

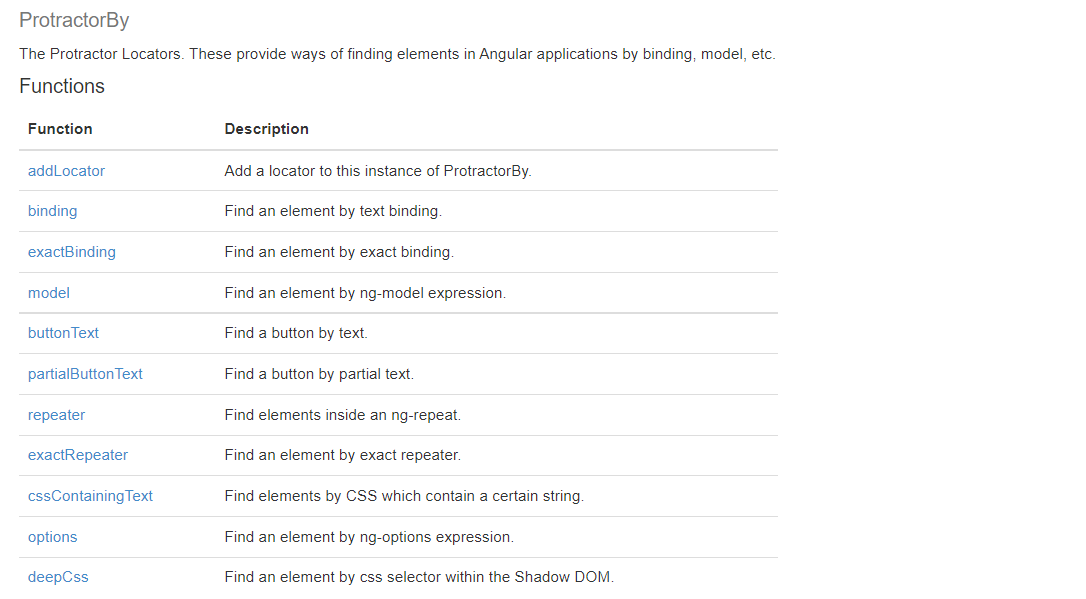
Plugins







Finding Element in Protractor

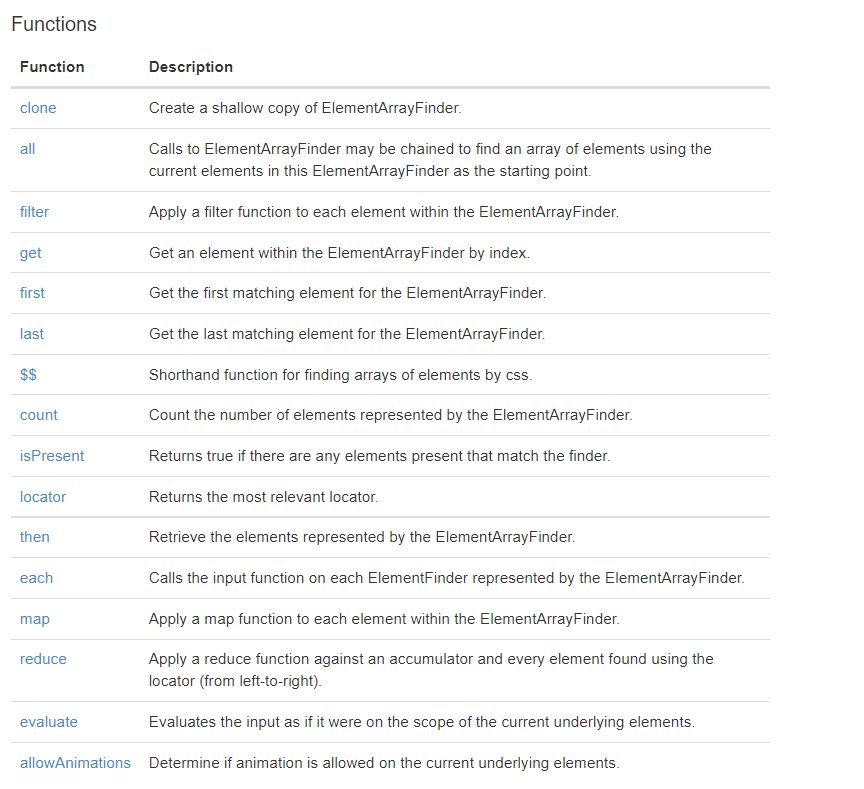


Validations in Protractor

- <https://jasmine.github.io/tutorials/your_first_suite>

Play with more than one element

In protractor we have to use element.all method to find more than one element



## Chain locators :

Finding an elements which is present inside the Element

Capture Screenshots in protractor

1. Go to npm website : <https://www.npmjs.com/>
2. Search for protractor-jasmine2-screenshot-reporter
   1. <https://www.npmjs.com/package/protractor-jasmine2-screenshot-reporter>
3. Follow the instructions to install the plugin
   1. npm i protractor-jasmine2-screenshot-reporter
4. Verify node\_module folder is created and it has the package installed
5. Update the conf.js with the instructions provided in the documentation
6. Execute the test and verify the screen-shot is captured

Allure Reports

1. Go to npm website : <https://www.npmjs.com/>
2. Search for jasmine-allure-report : <https://www.npmjs.com/package/jasmine-allure-reporter>
3. Follow the instructions to install the plugin
   1. npm i jasmine-allure-reporter
4. Verify node\_module folder is created and it has the package installed
5. Update the conf.js with the instructions provided in the documentation
6. Execute the test and verify the xml output generated under allure-reports
7. To convert the xml output to HTML use allure command line
   1. <https://www.npmjs.com/search?q=allure%20commandline>
   2. npm i allure-commandline
8. Execute allure serve <<location of xml file>>

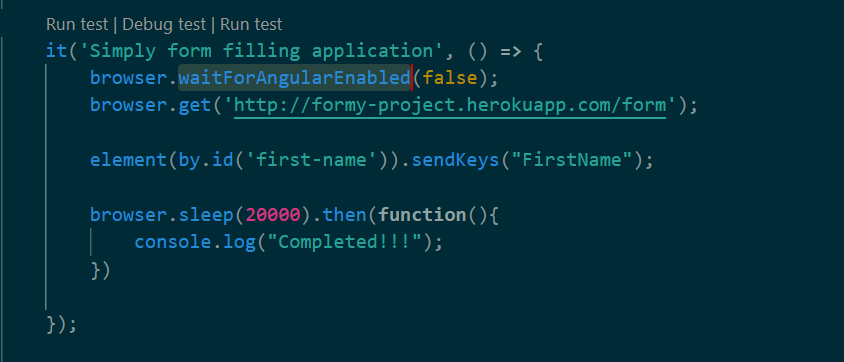
HTML Reports using protractor

1. Go to npm website : <https://www.npmjs.com/>
2. Search for protractor-html-reporter-2
   1. <https://www.npmjs.com/search?q=protractor%20html%20reporter2>
3. Install the plugin
   1. npm i protractor-html-reporter-2
4. Verify node\_module folder is created and it has the package installed
5. Update the conf.js with the instructions provided in the documentation
6. Execute the test and HTML Output

Executing tests on a different Browser

* Download the latest driver executable for the browser because by default only chrome and FF browser drivers are downloaded
  + Webdriver-manager update –ie
* enable browsername in the capabalities
  + 

Automate non-Angular Applications



Browser Operations

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

browser.driver.navigate().back()

browser.driver.navigate().forward()

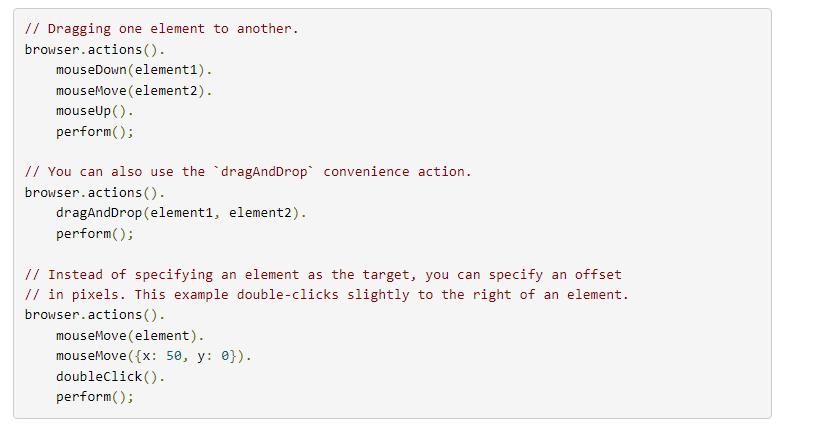
browser.driver.getCurrentUrl()

Handling Sync Issues OR Using Explicit Waits

<http://www.protractortest.org/#/api?view=ProtractorExpectedConditions.prototype.visibilityOf>

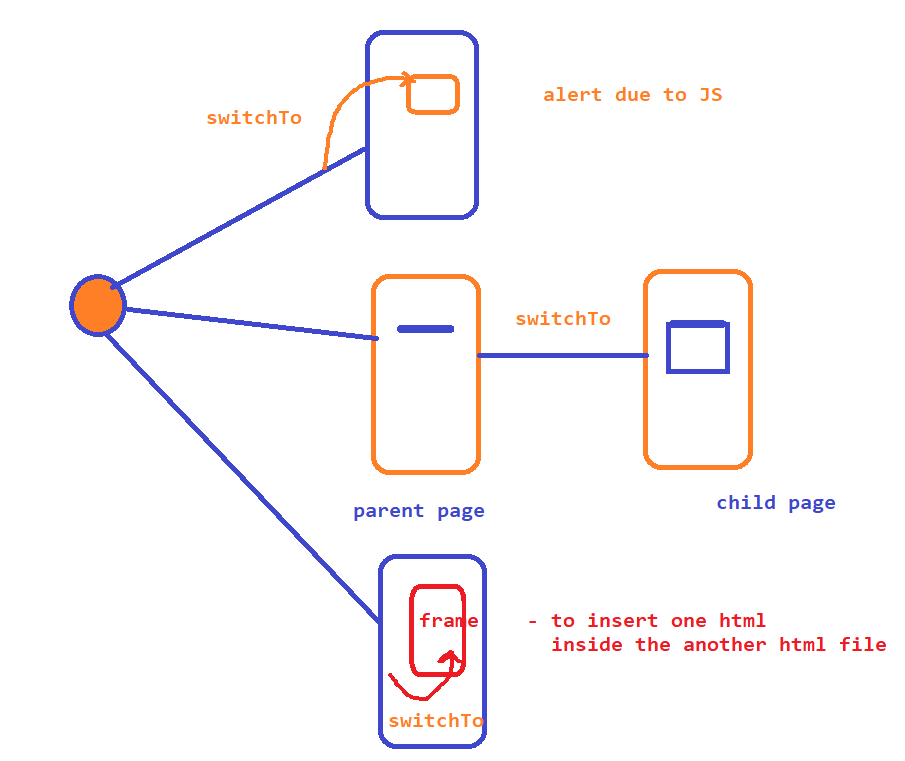


Actions

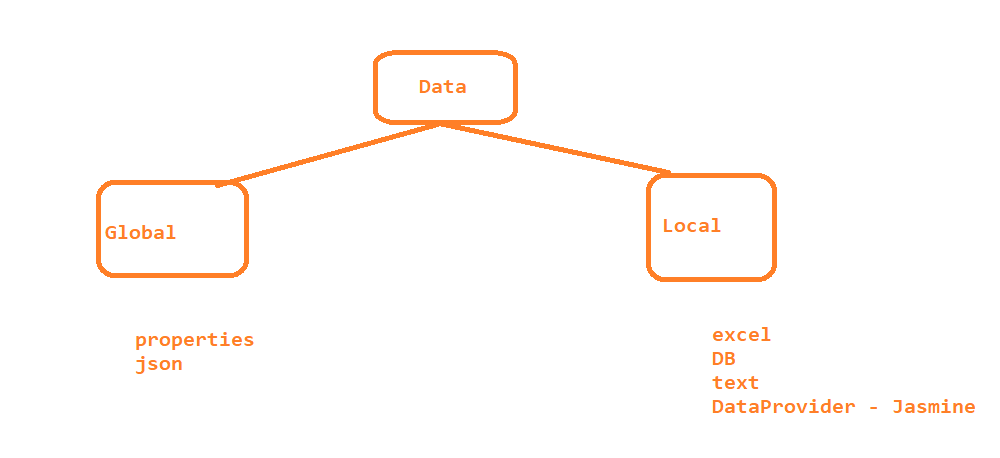
* <http://www.protractortest.org/#/api?view=webdriver.WebDriver.prototype.actions>
* 

SwitchTo

- windows api - <https://www.selenium.dev/selenium/docs/api/javascript/module/selenium-webdriver/lib/webdriver_exports_WebDriver.html>



Data driven Testing



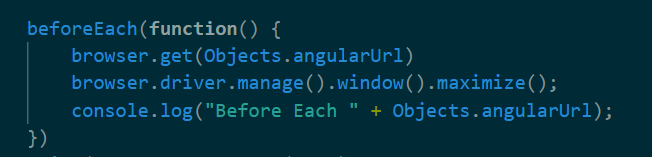
- Executing the test with different set of test data is called data driven testing

## Use Data Provider given by Jasmine

1. Download the plugin
2. Install the plugin
   1. npm i jasmine-data-provider
3. Follow the instructions

BeforeEach

Any pre condition to the test can be kept in one place instead of keeping in all the tests that is inside beforeEach function



EX: Launch, login, load data. etc

POM

1. create a page class add all the
   1. variables - webelements
   2. methods - to play with web elements
   3. export the class
2. Write a Test using Page
   1. import the page
   2. use the methods defined inside the page

Executing Test Suite

