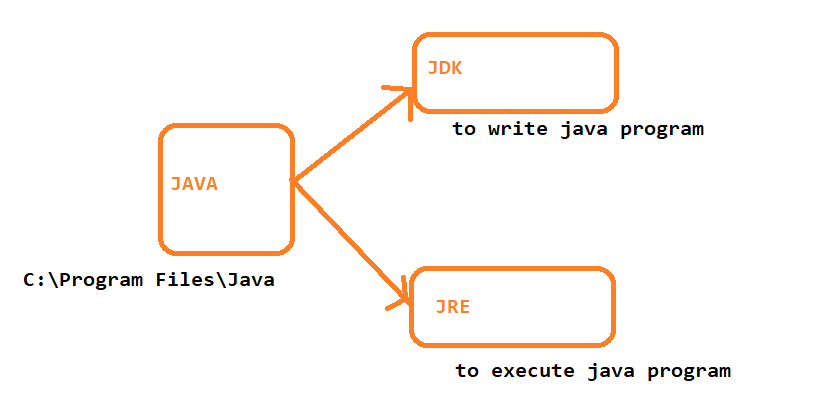
Java

* Brush-up on Java concepts
  + OOPS Concepts
    - Classes
    - Objects
    - Abstraction
    - Encapsulation
    - polymorphism
    - Inhertence
  + SIB and IIB
  + Interfaces
  + Access Specifiers
  + Access Modifiers / Non functional Access Specifiers
  + Abstract Class
  + packages
  + import statements
  + data types – types
  + variables
  + methods
  + loops – types of loops
  + conditional statements – types of conditional statements
  + **String – Class**
  + **File –Reading and writing in JAVA**
  + Exceptions how to handle Exceptions
  + **Collections**
  + **Generics**
* Java-1.8
  + Inner Class – Advanced Java
  + Interfaces with Advanced Features
  + forEach & Consumer Class
  + Lambda Expressions
  + **:: Operator**
  + Stream APIs
  + Improvements made to Collection API
  + Improvements made to IO package
  + Date / Time APIs

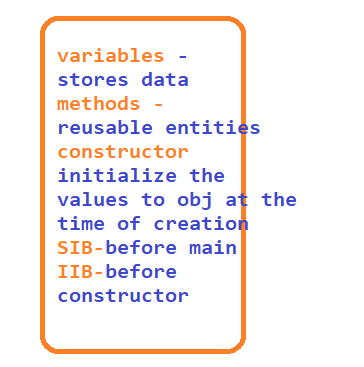
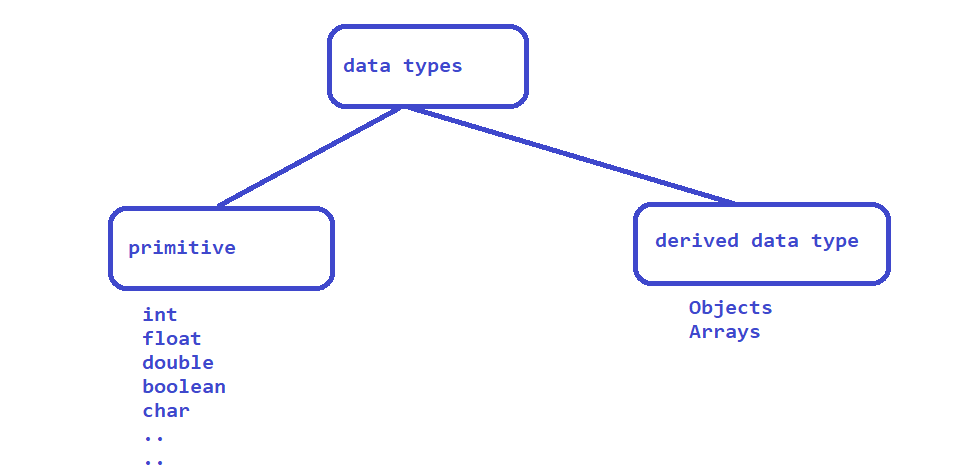
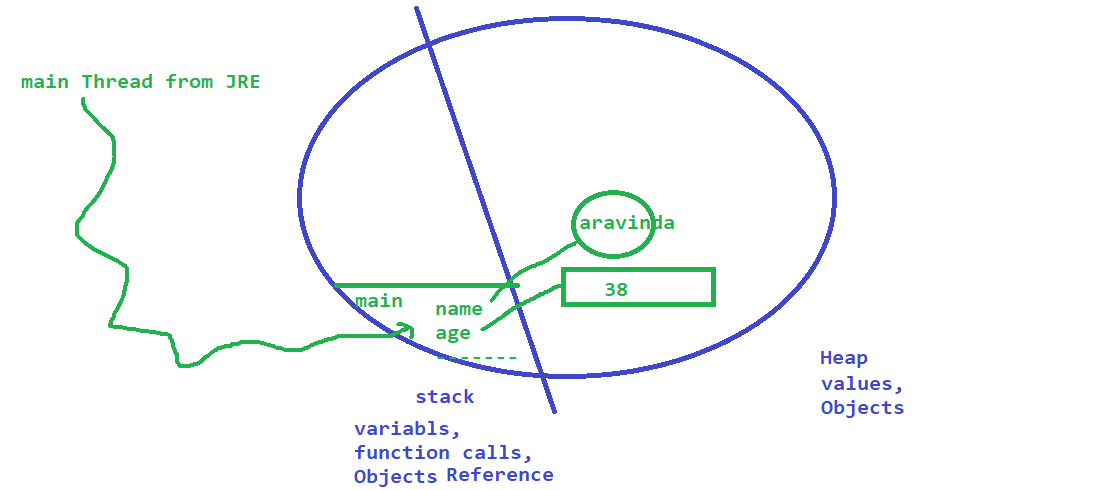
Introduction to Software and libs



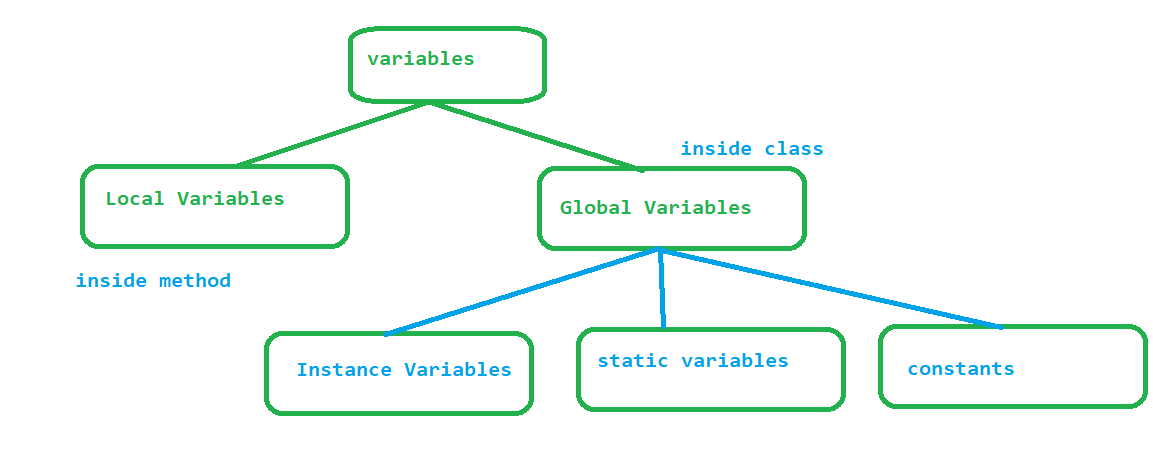
C:\Program Files\Java\jdk1.8.0\_201\bin

C:\Program Files\Java\jdk1.8.0\_201\lib

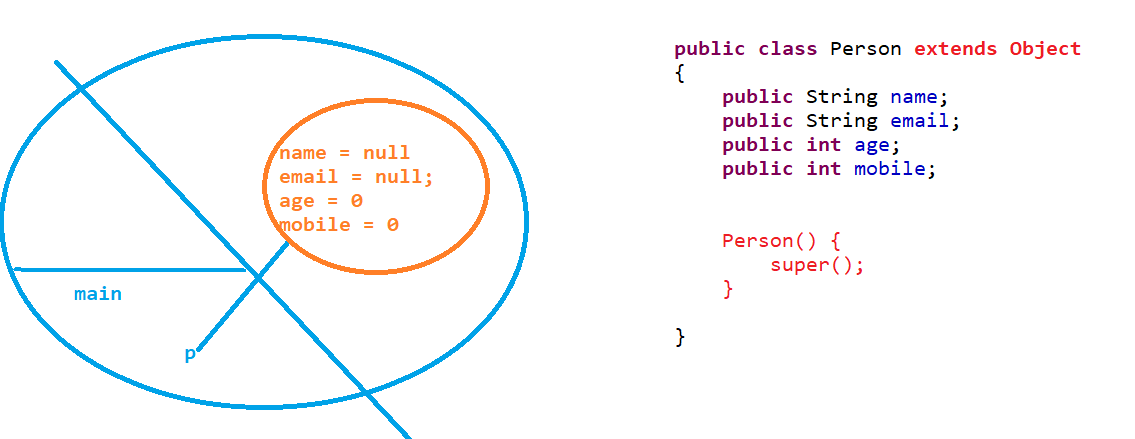
Oops Concepts

* Class
* 
* Object
* package
  + 
* Data types
* 
* 

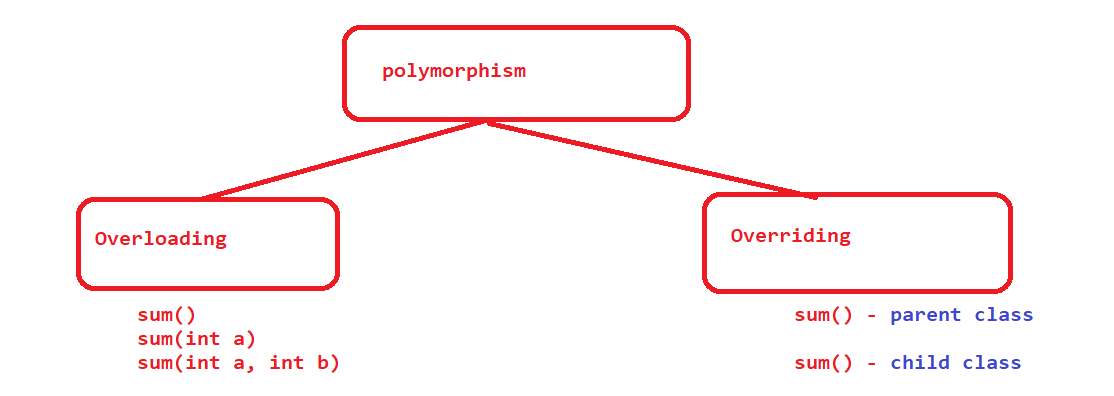
Variables



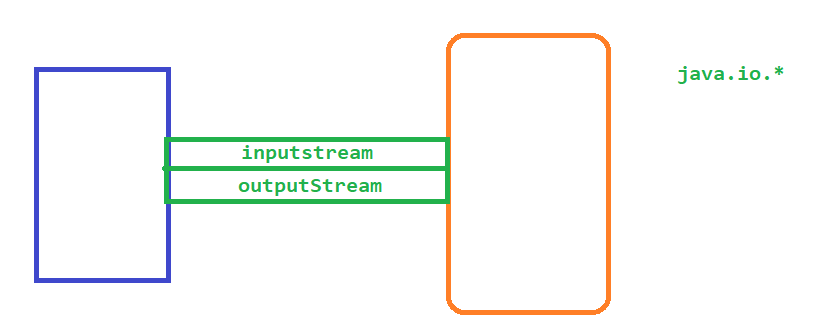
Constructor



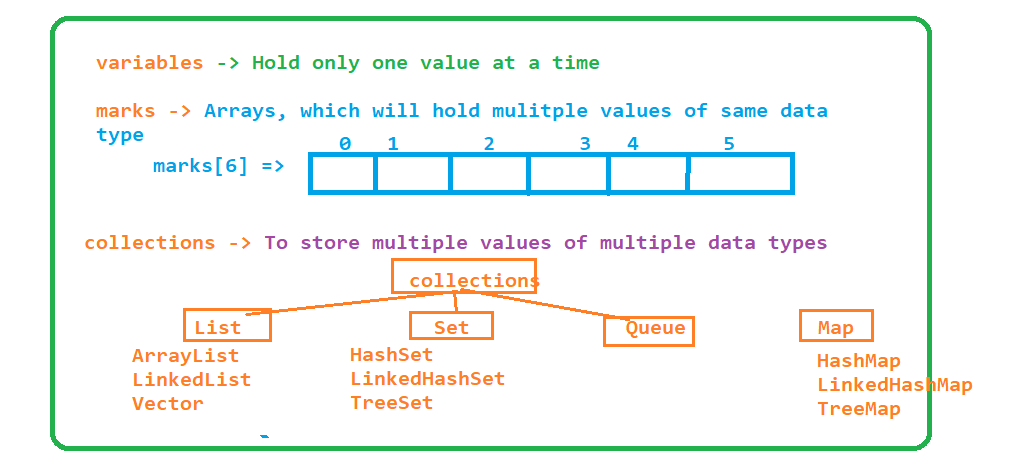
Polymorphism



Files

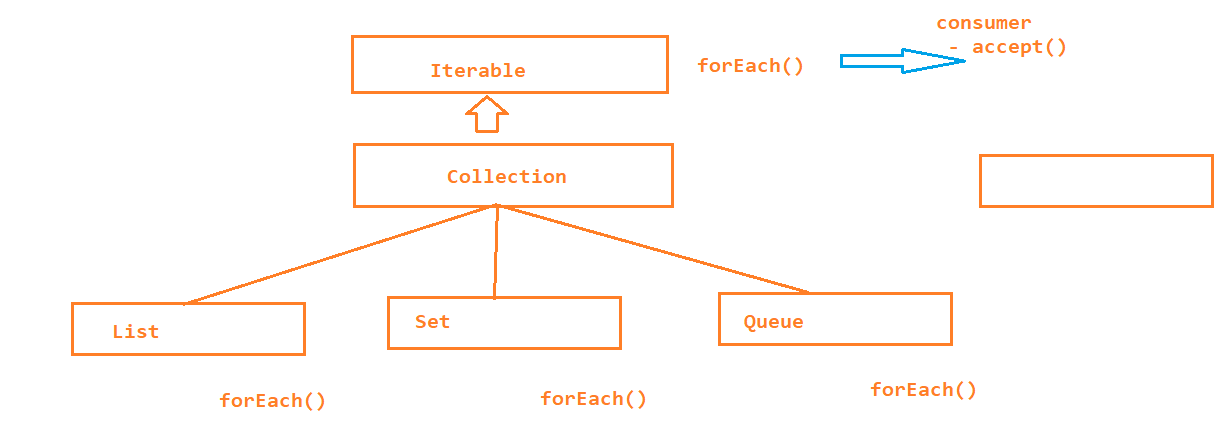


Collections – java.util.\*



Java-1.8

## forEach



\* Consumer is a interface introduced from java 1.8 which has an unimplemented method accept

* forEach method is also introduced from java1.8 which is present inside Iterable interface
* Iterable is a parent for all the collection classes
* Foreach expect object to Consumer class
* to Foreach loop we have to implement the actual logic inside the accept method of consumer class

1. Inner Class

Inner classes are also called as Nested Classes, mainly used to achieve

1. Better Encapsulation

2. To group the logically similar classes

3. for better readable and maintainable code

Types :

1. static inner class

* is a member of parent class
* it is accessible from all the static members of the parent class

2. local inner class

* it is a member of a method
* local inner classes are accessible only inside the method, not from any other member of a class
* LIC are local to the method only

3. Anonymous inner class

* are the class without any name
* we can create object to interface using anonymous inner classes

1. Lambda Expressions

Lambda expressions are the block of code, which can be executed without object

Lambda expressions are similar to methods ( advanced methods )

Faster execution – because it executes without object

While writing lambda expression

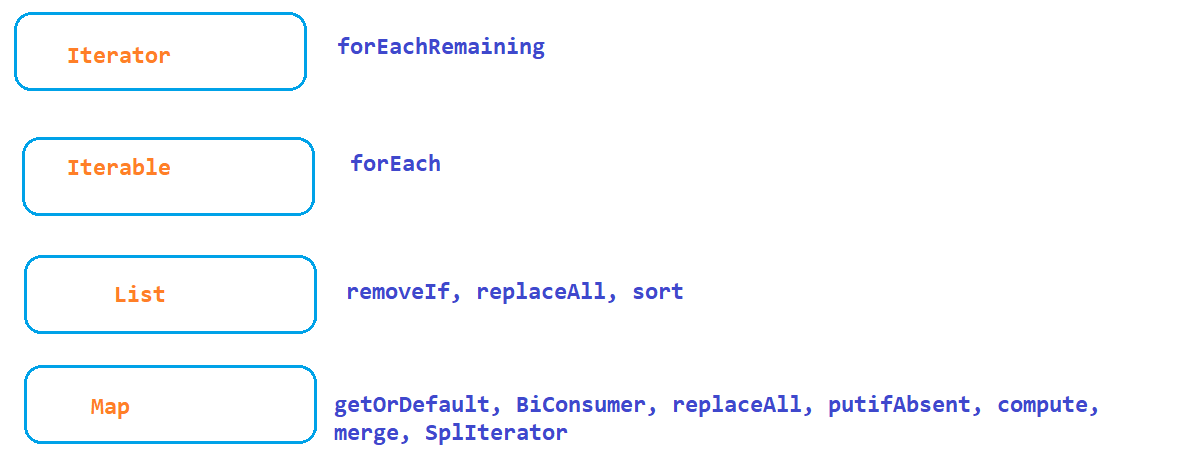
* Lambda expressions with arguments
  + Data type is optional when we are implementing method using lambda expression
* Lambda expressions without arguments
* Lambda expressions with return type
* Lambda expressions without return type
  + if we have one line implementation then return keyword is not needed
  + return keyword is needed if there is multiple line of implementation
* Lambda expressions for forEach loop

:: Operator

:: is used to assign the existing implementation to the unimplemented method of a Interface

* Unimplemented method of an interface can be implemented by
  + Class
  + Annonymous inner class
  + Lambda expression etc

Collections



IO Improvements

* list – list all the files of the current folder / dir, it will not go to subdir
* walk - – list all the files of the current folder as well as in the sub dir till the last file is reached
* lines – will read and print each and every line in a file

Steam

* in java.util package there is a sub package called stream introduced in 1.8 mainly to process the data
* Processing group of elements that are in array or Collections can be easily done here.
* Processing - > finding max, min, avg, count, etc

Date and Time

* Thread safety
* Difficult to implement date and time with it comes to different time zones
* Very Complicated before 1.8
  + LocalDate
  + LocalTiime
  + LocalDateTime

Selenium

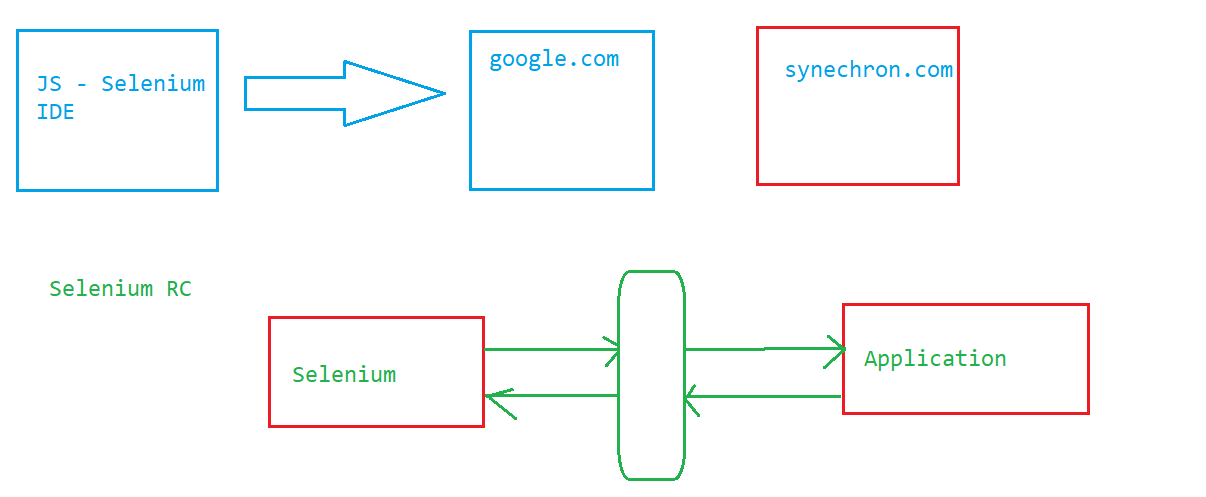
## Browser automation tool, any application that can be opened on a browser will be automated using selenium

* Selenium IDE
* Selenium WebDriver
* ~~Selenium RC~~
* Selenium GRID

# Features of Selenium -https://www.selenium.dev/

* Open Source
* Supports multiple programming languages – Java, c#, Ruby, Python, JavaScript, Kotlin
* it supports almost all popular browsers
* No dedicated machine is required for Test Execution
* You can execute your tests on any platform
* plugin – pre existing frameworks like testing,junit, custom reports to selenium
* Distributed Execution
* Parallel Execution

History



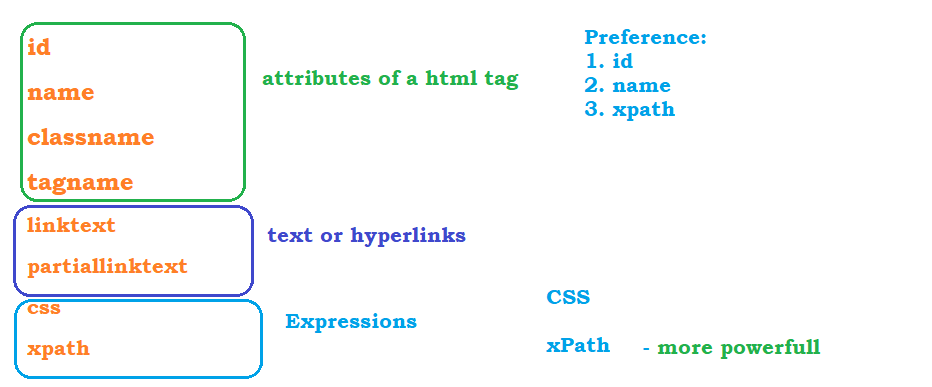
Selenium WebDriver - JAVA

Selenium IDE

* only for Beginners
* supports Record and playback
* How to perform Validation
* Batch Execution
* it was an add-on only with FF, but from selenium3.0 it is available in both FF and chrome
* Chrome : <https://chrome.google.com/webstore/category/extensions>

**Identification**

* id
* name
* classname
* tagname
* linktext
* partiallinktext
* css
* xpath



# CSS :

1. html[attribute=’value’]
   1. input[placeholder='Enter your job title']
2. Class name :
   1. htmltag[attribute=’value’]

OR

* 1. htmltag.classValue

OR

* 1. .classValue

1. ID
   1. htmltag[attribute=’value’]
   2. htmltag#idValue
      1. input#first-name
   3. #idValue

#first-name

1. Patent to child
   1. div[class='col-sm-8 col-sm-offset-2'] > #first-name

xpath

- absolute xpath🡺 /

- relative xpath 🡺 //

# Basic Xpath

* //html[@attribute = ‘value’]
* //input[@placeholder=’Enter last name’]

# Logical Operators in Xpath

* and
  + //htmltag[@attribute1=’value1’ and attribute2=’value2’]
  + //input[@class='form-control' and @placeholder='Enter first name']
* or
  + //htmltag[@attribute1=’value1’ or attribute2=’value2’]
  + //input[@type='radio' or @type='checkbox' or @type='text']
* not
  + //input[not(@type='text')]
  + //td[@class='current day']
* //td[@class='current day' or @class='wd day' or @class='we day']

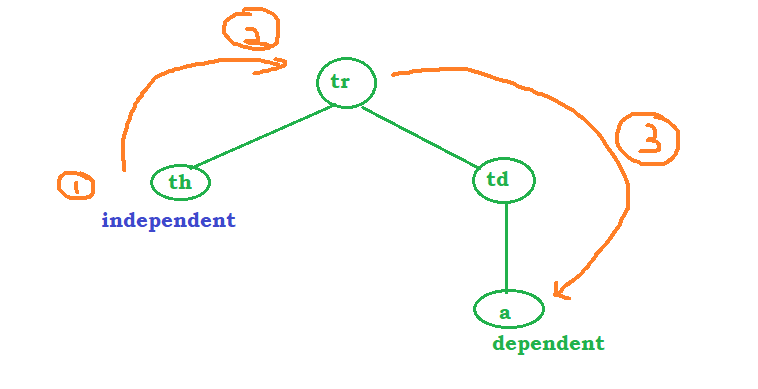
# Functions in Xpath

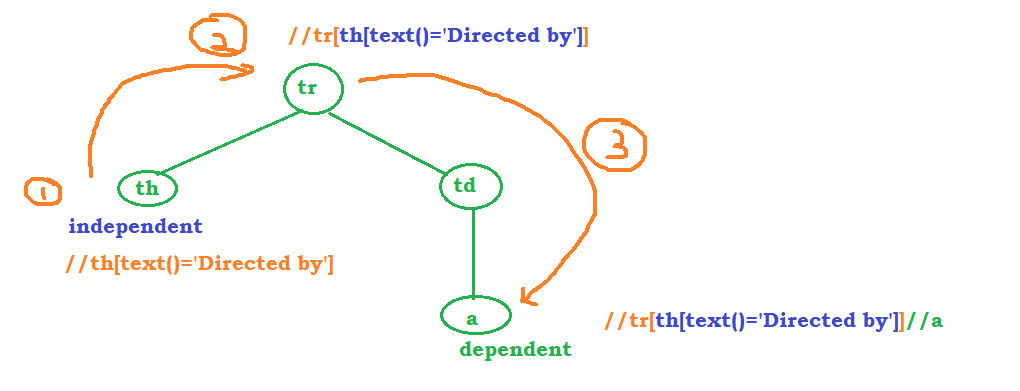
* text()
  + //htmltag[text()=’exact text on screen’]
  + //button[text()='Search Buses']
* contains(arg1,arg2)
  + arg1 - can be any attribute or text() function
  + arg2 – partial text corresponding to arg1
  + //button[contains(text(),'Search')]
  + //a[contains(@class, 'logo')]
  + //td[(@class='current day' or @class='wd day' or @class='we day') and text()='26']
* starts-with(arg1,arg2)
  + arg1 - can be any attribute or text() function
  + arg2 – starting text corresponding to arg1
  + //h3[starts-with(text(),'Synechron')]

# Traversing from parent to Child

* if child is a immediate child then use /
  + //div[@class='col-sm-8 col-sm-offset-2']/input[@id='first-name']
* if the child is not a immediate child
  + //div[@class='form-group']//input[@id='first-name']

# Traversing from child to parent

* Whenever we have dependent and independent elements we have to traverse from child to parent
  1. Always write xpath to independent element
  2. traverse to the parent until both dependent and independent elements are visible
  3. traverse to the required dependent element
* 



* //tr[th[text()='Directed by']]//a
* //tbody[tr[th[text()='Body']]]//td[@class='nfo']

# Xpath using Axes Functions – to traverse between siblings, ancestors etc

* **To Traverse to the next sibling** : //th[text()='Directed by']/following-sibling::td
* **To Traverse to the preceding sibling** : //td[@class='nfo']/preceding-sibling::td
* //span[text()='Marketing']/preceding-sibling::span - <https://en.wikipedia.org/wiki/K.G.F:_Chapter_1>
* **To Traverse to Parent :** 
  + //th[text()='Body']/parent::tr/parent::tbody//td[@class='nfo']
* **To Traverse to ancestor :** 
  + //th[text()='Body']/ancestor::tbody//td[@class='nfo']

<https://holidayz.makemytrip.com/holidays/india/search?dest=Bali>

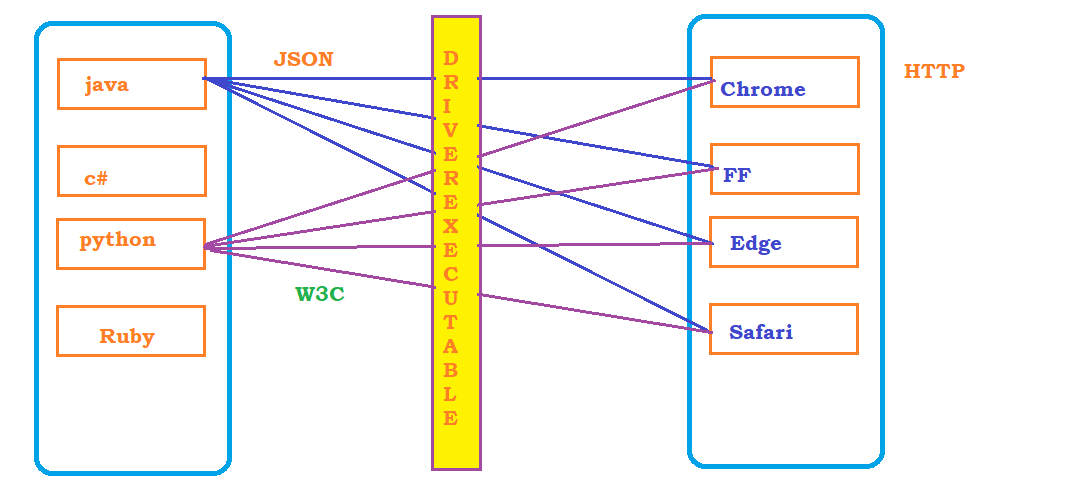
//h4[text()='Bali Super Saver (Free & Easy)']/ancestor::div[@class='itemCard packageCard']//p[contains(@class,'blackText ')]

<https://www.amazon.in/s?k=mobile&i=electronics&crid=3K0L2WY98HGW2&sprefix=mobile%2Celectronics%2C222&ref=nb_sb_noss_1>

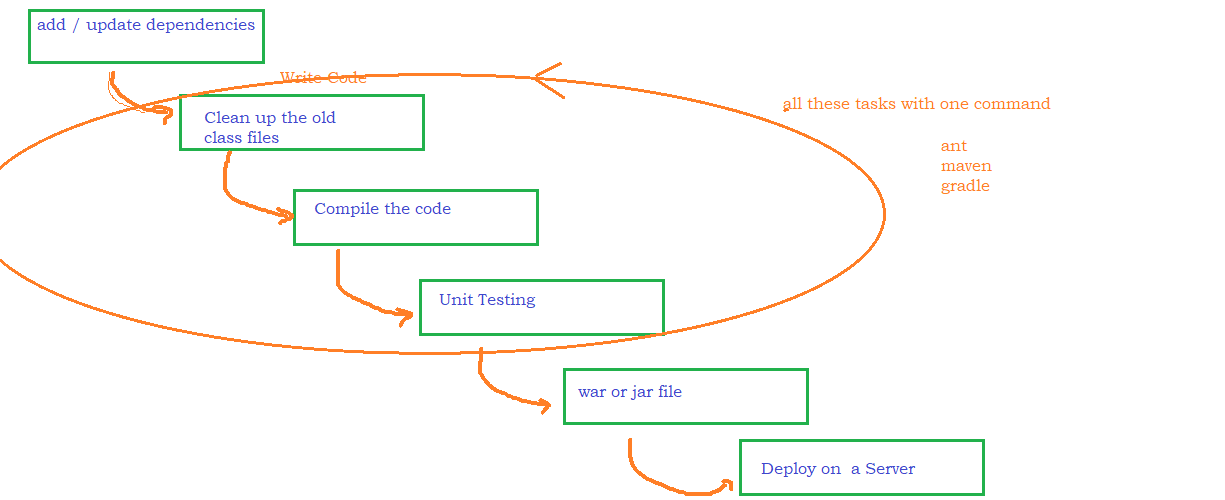
//span[contains(text(),'Samsung Galaxy M53 5G (Mystique Green, 6GB, 128GB Storage) | 108MP | sAmoled+ 120Hz | 12GB RAM with RAM Plus')]/ancestor::div[@class='a-section a-spacing-small a-spacing-top-small']//span[@class="a-price-whole"]

* **Traverse to all Siblings till beginning of the page or end of the page**
* //span[text()='Marketing']/ancestor::li/following::a
* //span[text()='Marketing']/ancestor::li/preceding::a

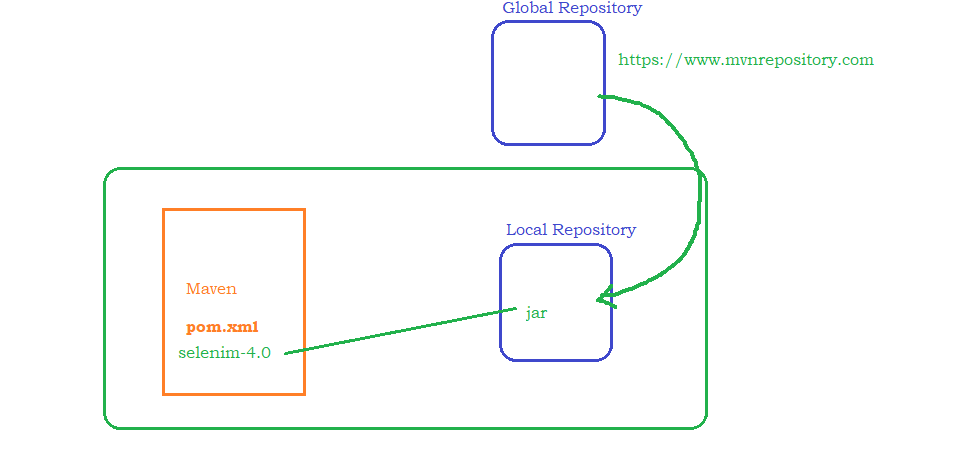
Architecture



Introduction to Maven



# Repository:



Install Maven

* <https://maven.apache.org/install.html>

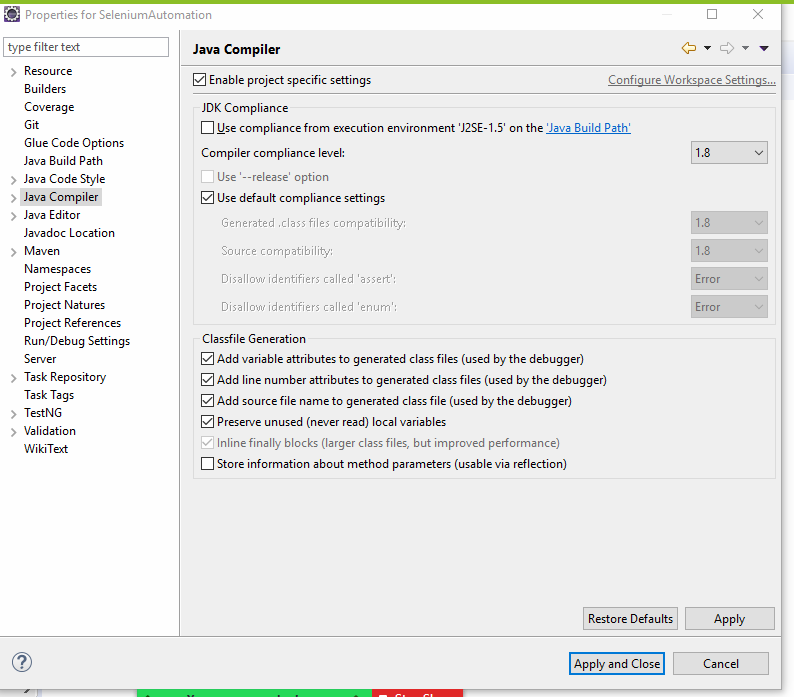
- Download and unzip the file <https://dlcdn.apache.org/maven/maven-3/3.8.6/binaries/apache-maven-3.8.6-bin.zip>

* Set environment Variables
  + MAVEN\_HOME - D:\maven\apache-maven-3.8.6
  + M2 - D:\maven\apache-maven-3.8.6\bin
  + PATH - D:\maven\apache-maven-3.8.6\bin
* Open command prompt and execute mvn -version

Creating Maven project



# Update compiler to latest

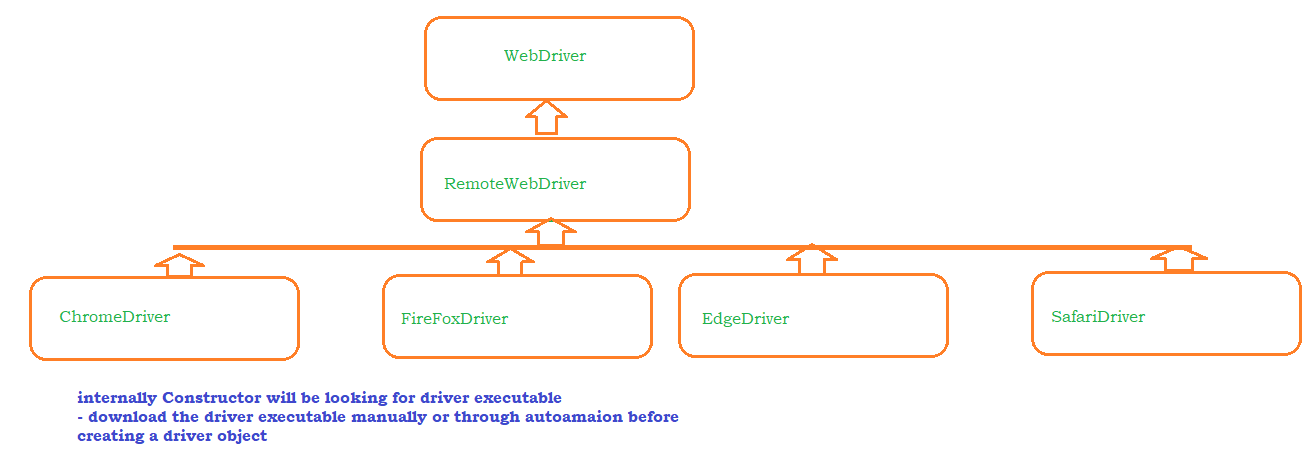


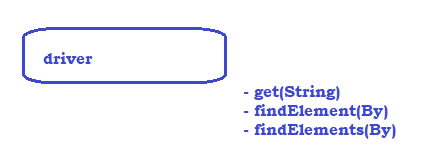
# Update JRE to latest

# Repository location

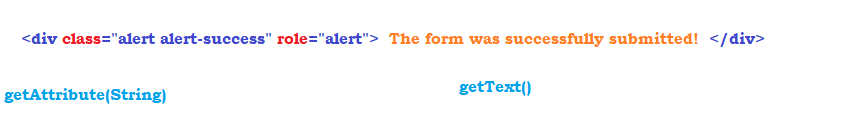
C:\Users\Aravind\.m2\repository\org\seleniumhq\selenium\selenium-java\4.4.0

Selenium Lib Overview





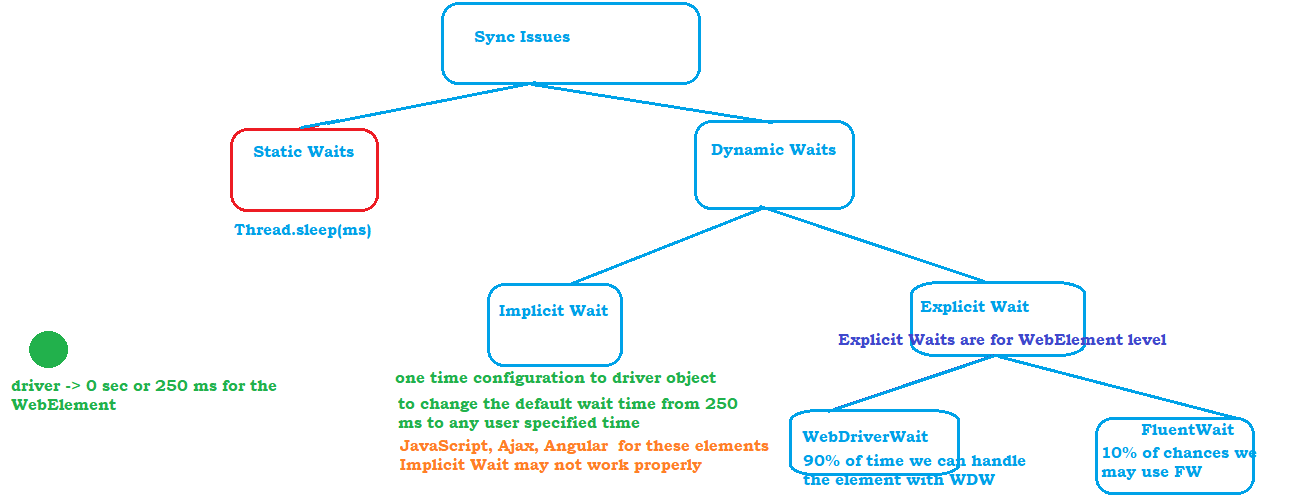
getAttribute(String) and getText()



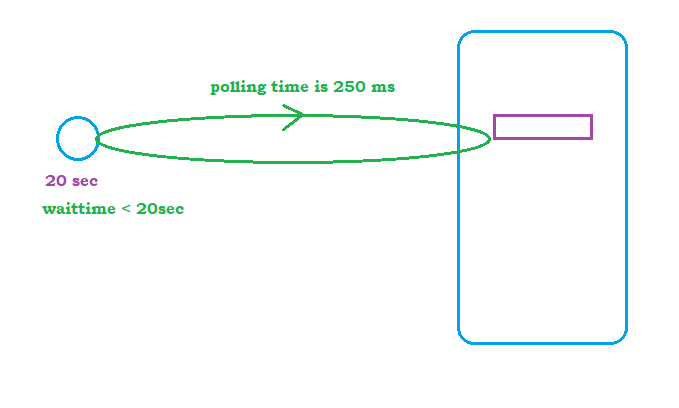
Sync issues

Execution speed of the tool is much faster than the execution speed of the application because of this some tests may fail . to fix the sync issues we have to make sure execution speed of the tool is same as the execution speed of the application.

Basically we have to reduce the execution speed of the tool



Implicit wait



Explicit Wait - WebDriverWait

1. Create an object to WEbDriverWait by passing driver obj and max wait time
2. use until method
3. Call required method from ExpectedConditions Class to the until method

ExplicitWait – FluentWait

1. Only the functions which are present inside the ExpectedConditions class has to be used in WebDriverWait
2. if you want to change the polling time
3. When you want to write you own wait logic
4. Ignore some exceptions during wait

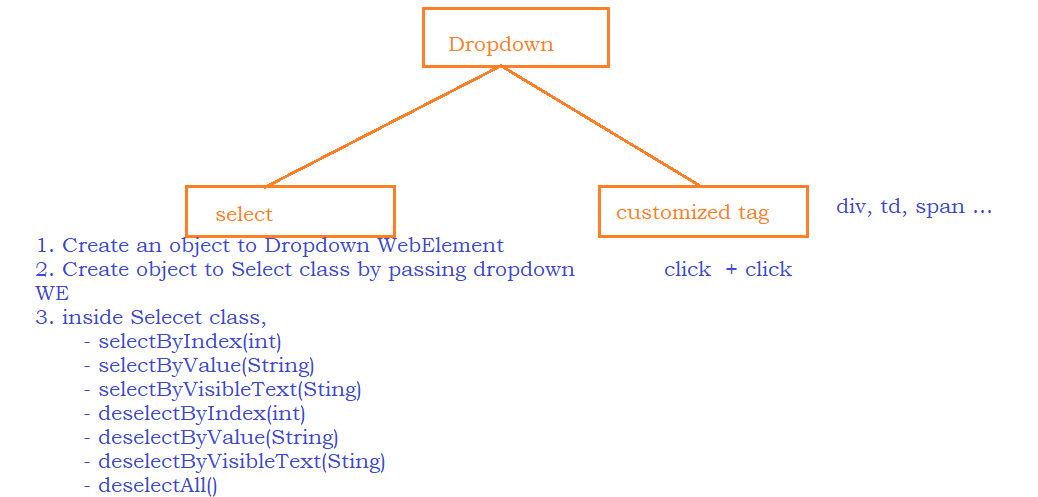
# Steps:

1. Create Wait object, while creating wait object
   1. What is the WebElement
   2. What is the max timeout
   3. What is the polling timeout
   4. What are the exceptions to ignore
2. Implement your own wait logic
3. Use Until function present in wait object

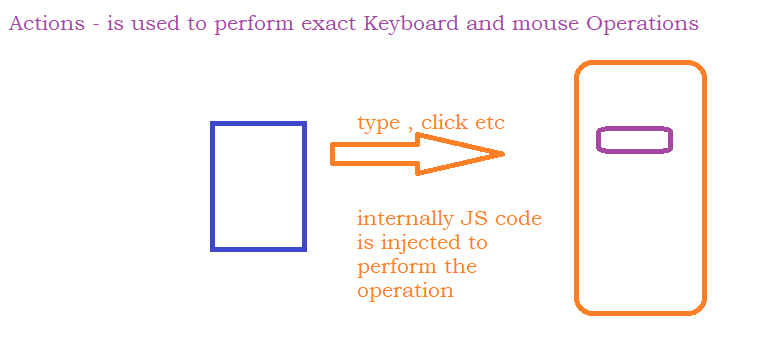
Browser Operations



Handling Dropdowns



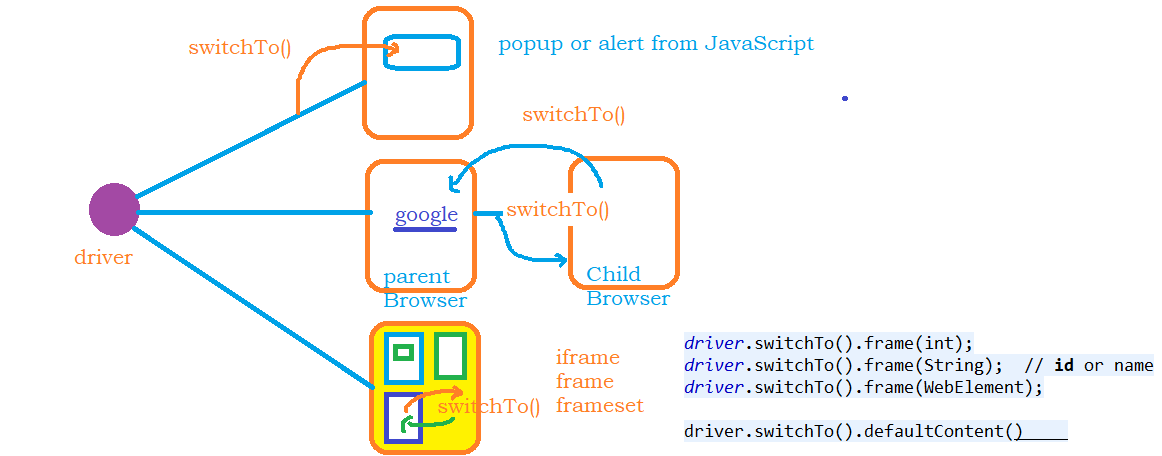
Actions



Automate Downloading Driver Executable

* Update POM.xml with WebdriverManager library
* <https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager>
* Use WebDriverManager.*chromedriver*().setup(); to download the driver executable
* call the above line before creating a driver

SwitchTo



Data driven Testing

