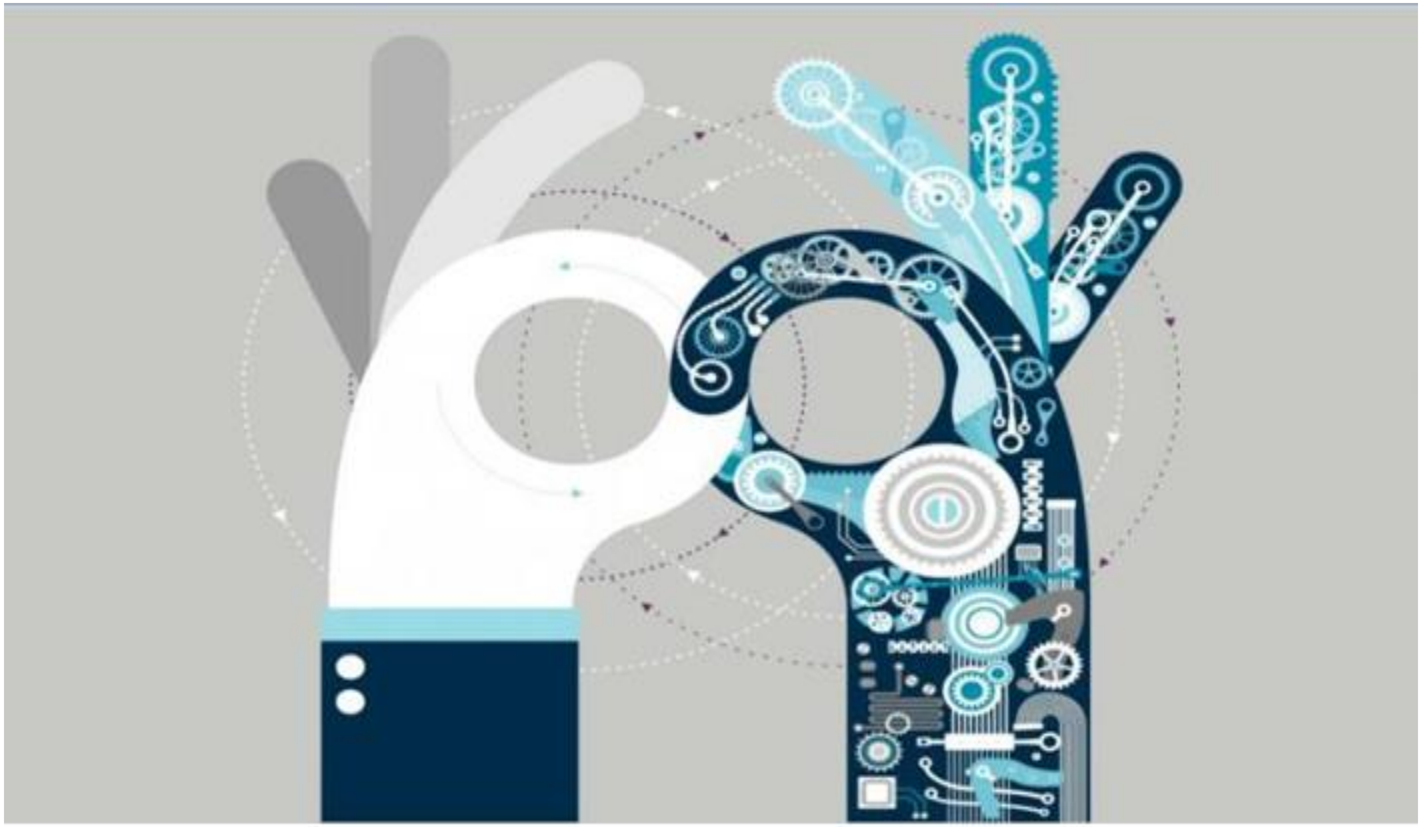
Welcome to the world of software testing



Agenda

- Introduction to Automation Testing
- Manual Testing Vs Automation Testing
- When do we need Automation?
- Limitations of Automation Testing
- How to choose right tool for automation?
- Introduction to Selenium
- Components of Selenium
- Introduction to Selenium WebDriver
- Installation of Selenium
- WebElements
- Locators
- CSS Selector
- XPath - Absolute & Relative
- Waits in Selenium

AUTOMATION TESTING

Process of converting manual test cases into the test scripts with the help of automation tools or any programming language

When we can go for Automation Testing?

✓ When the cost makes sense

✓ When using repetitive tests

✓ When time will be saved

✓ When quality is sure to be improved

✓ When tests are run frequently

✓ When you need to run multiple tests at once

Manual Vs Automation

- Test cases are executed manually.
- Difficult to ensure sufficient test coverage
- May difficult to test on different browsers
- you need to sit in front of your system and execute test cases

- Test cases are executed automatically with the help of tools.
- Easy to ensure greater test coverage.
- We can easily test on different
- You just have to run Automation scripts you can run it overnight!

Limitations of Automation Testing

- Cannot perform testing for Images.
- Captcha, Barcodes.
- Continues maintenance of code.
- Cannot perform testing for audio or video

Tools

TOOLS FOR AUTOMATION TESTING

Se			cy	TC
Web	Web, API, Mobile, Desktop	Mobile (iOS, Android)	Web	Web, Mobile, Desktop
Windows, Mac OS, Linux	Windows, Mac OS, Linux	Windows, Mac OS	Windows, Mac OS, Linux	Windows
Java, C#, Python, JavaScript, Ruby, PHP, Perl	Java & Groovy	Java, C#, Python, JavaScript, Ruby, PHP, Perl	JavaScript	C++,C#, Python, JavaScript, VBScript, Jscript, Delphi

SELENIUM & ITS FEATUES

Selenium is a free automated testing tool

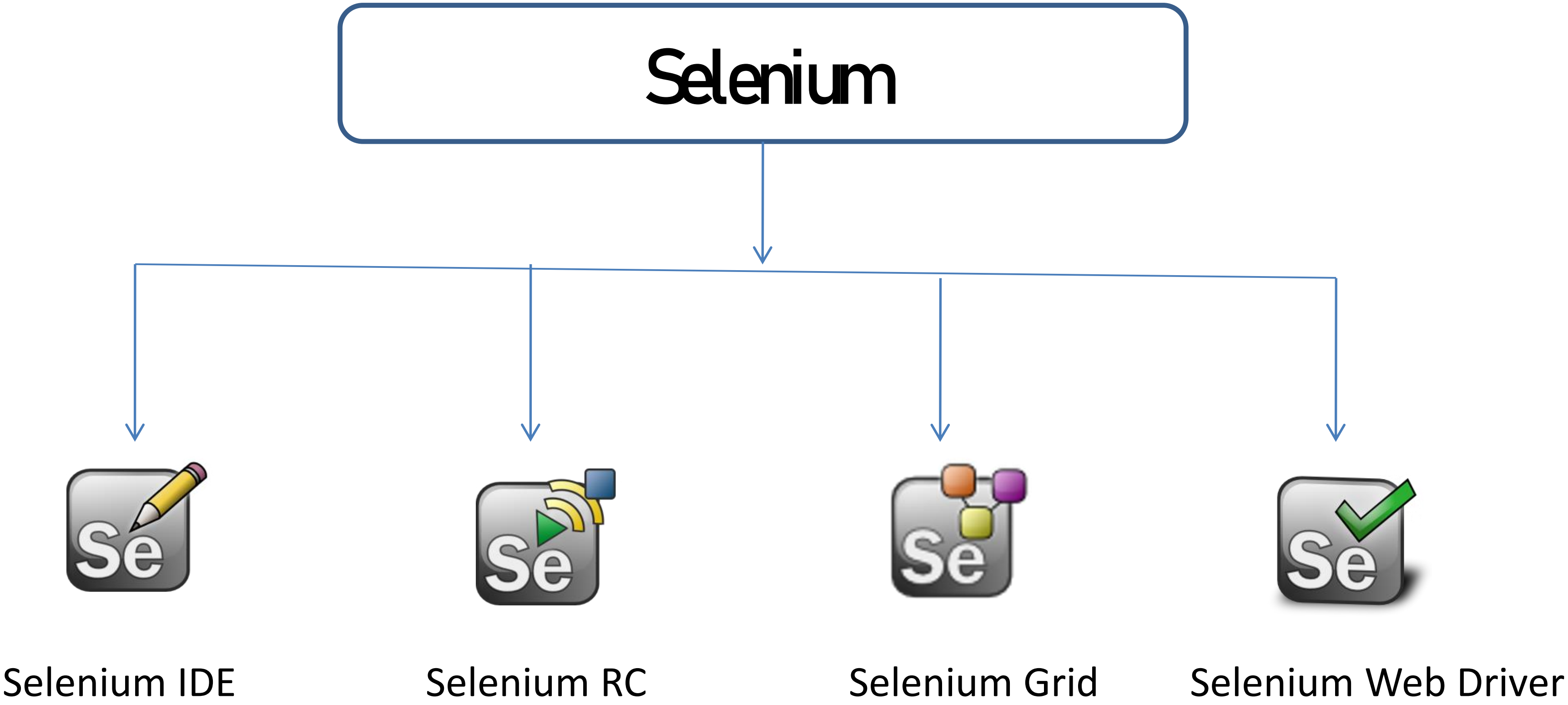
Features

- Open Source
- Supports Multiple operating system
- Supports multiple browsers
- Supports multiple programming languages
- Supports multiple framework
- Supports parallel and cross browser execution

How to select right tool for Automation?

- Project Requirements
- Team skills / Learning Curve
- Budget
- Ease of Test case Creation and Maintenance
- Reusability
- Data-Driven Testing
- Reporting
- Support for Collaboration

SELENIUM COMPONENTS



Selenium

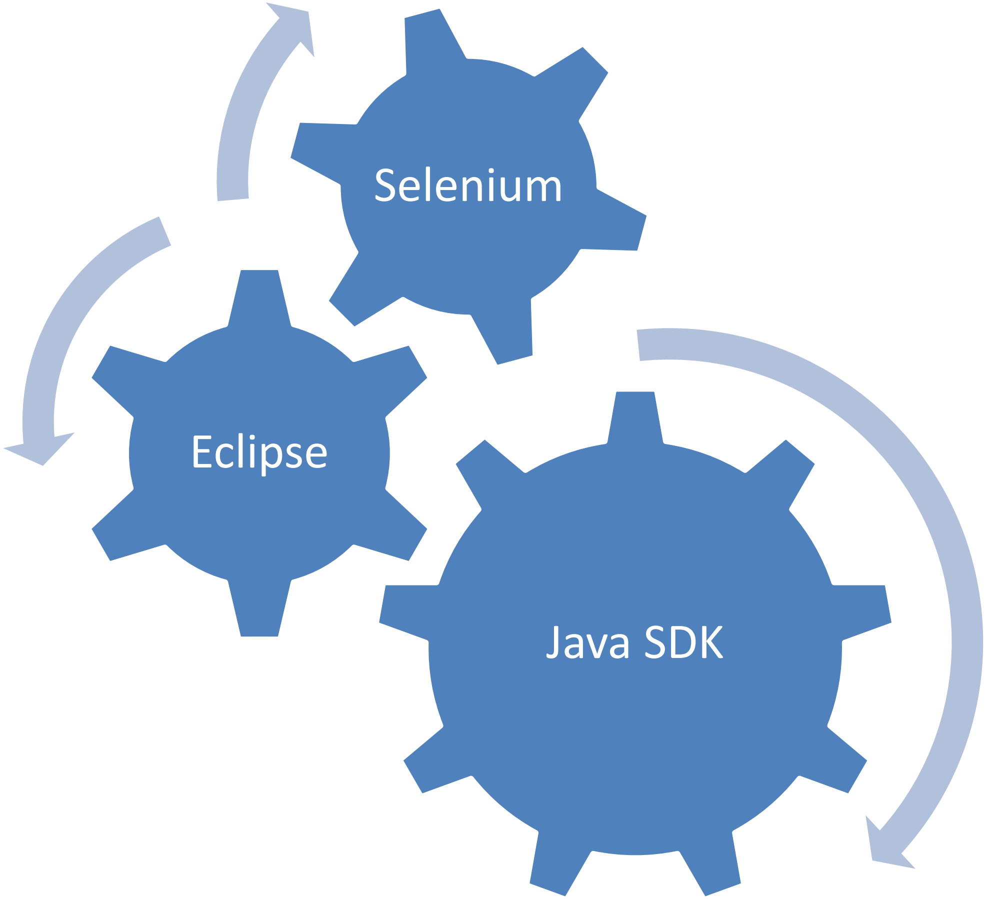
Selenium IDE

Selenium RC

Selenium Grid

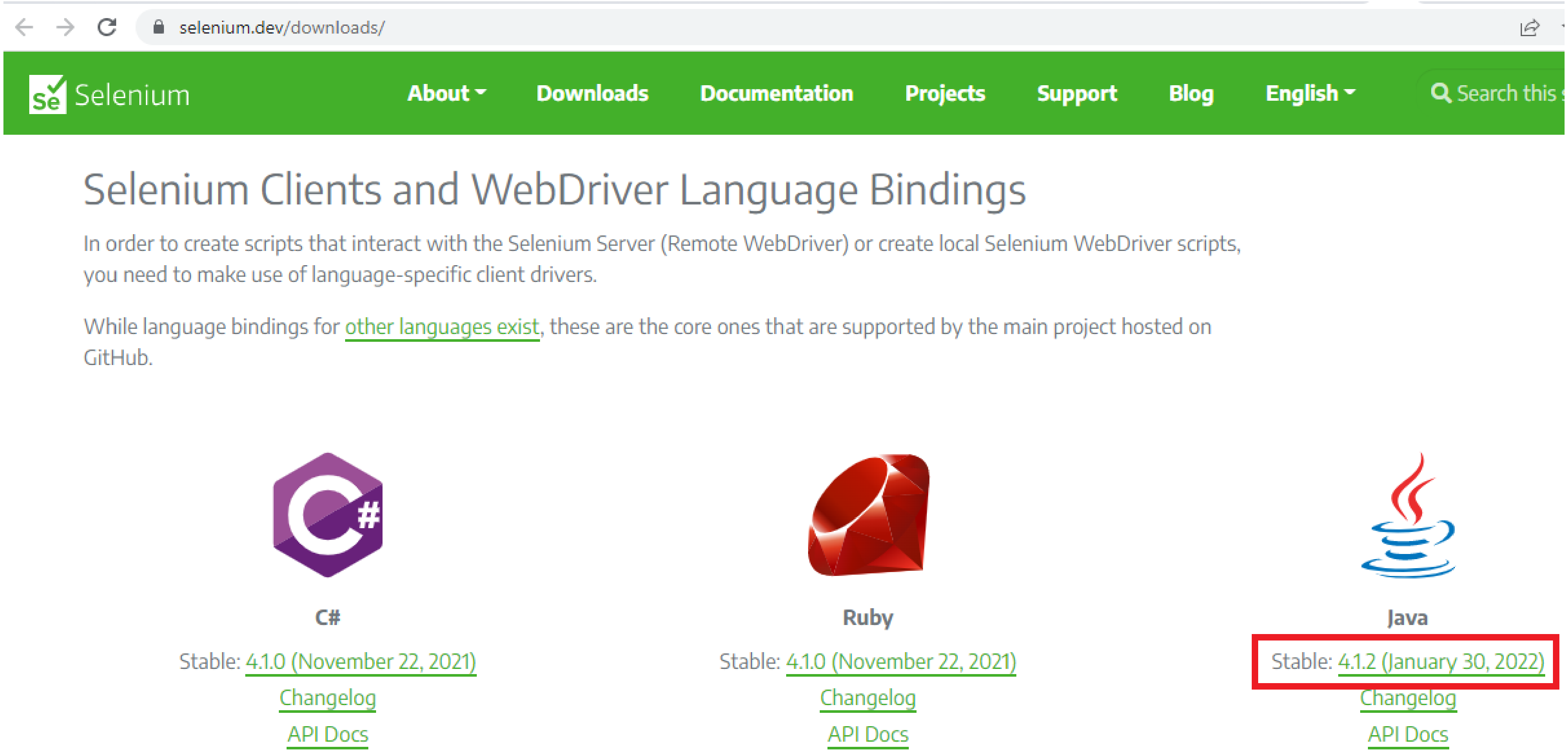
Selenium Web Driver

PREREQUISITES - TO USE SELENIUM



INSTALLATION

Navigate to https://selenium.dev/downloads/ and check for **Selenium Clients and Web Driver Language Bindings**



Selenium Clients and WebDriver Language Bindings

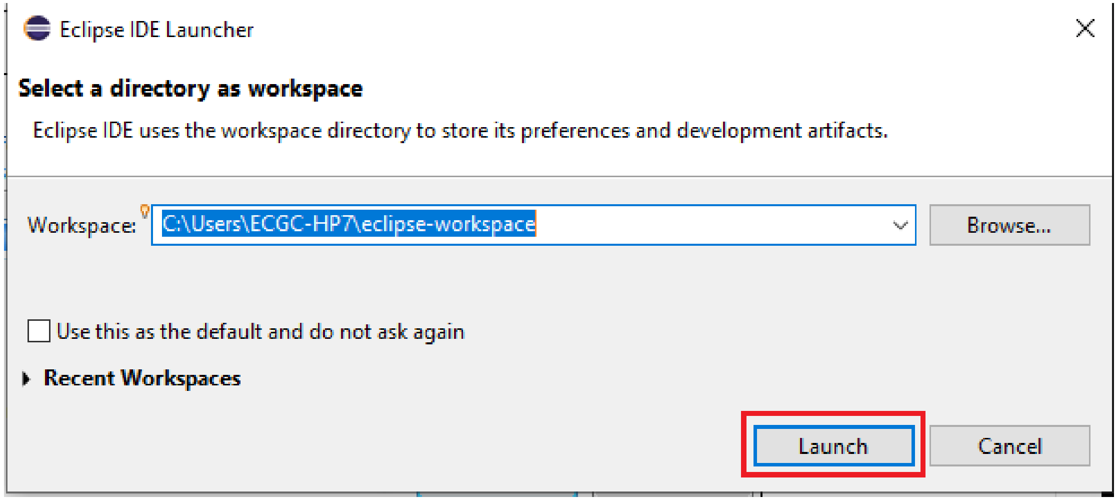
In order to create scripts that interact with the Selenium Server (Remote WebDriver) or create local Selenium WebDriver scripts, you need to make use of language-specific client drivers.

While language bindings for other languages exist, these are the core ones that are supported by the main project hosted on GitHub.

C#
Stable: 4.1.0 (November 22, 2021)
Changelog
API Docs

Ruby
Stable: 4.1.0 (November 22, 2021)
Changelog
API Docs

Java
Stable: 4.1.2 (January 30, 2022)
Changelog
API Docs



Eclipse IDE Launcher

Select a directory as workspace

Eclipse IDE uses the workspace directory to store its preferences and development artifacts.

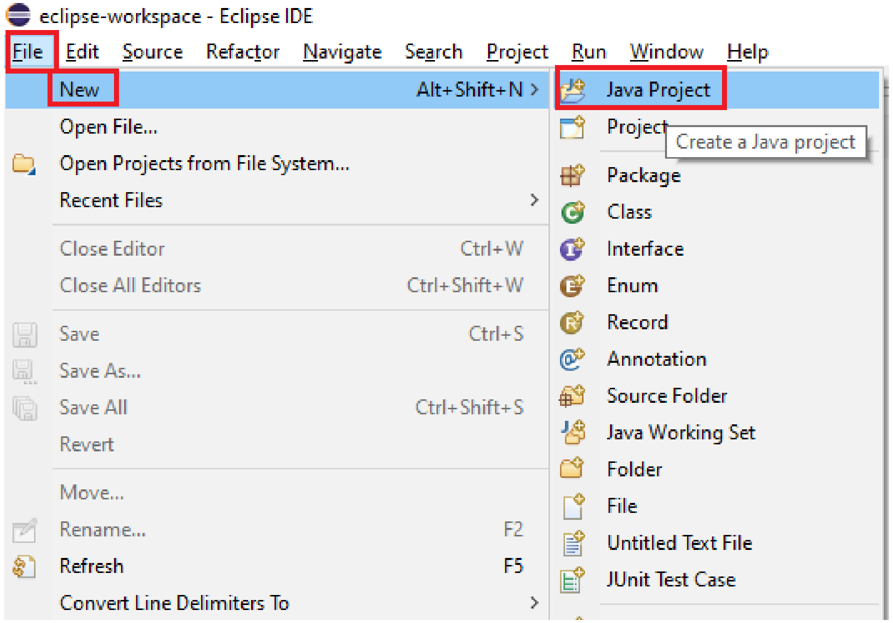
Workspace: C:\Users\ECGC-HP7\eclipse-workspace

□ Use this as the default and do not ask again

Recent Workspaces

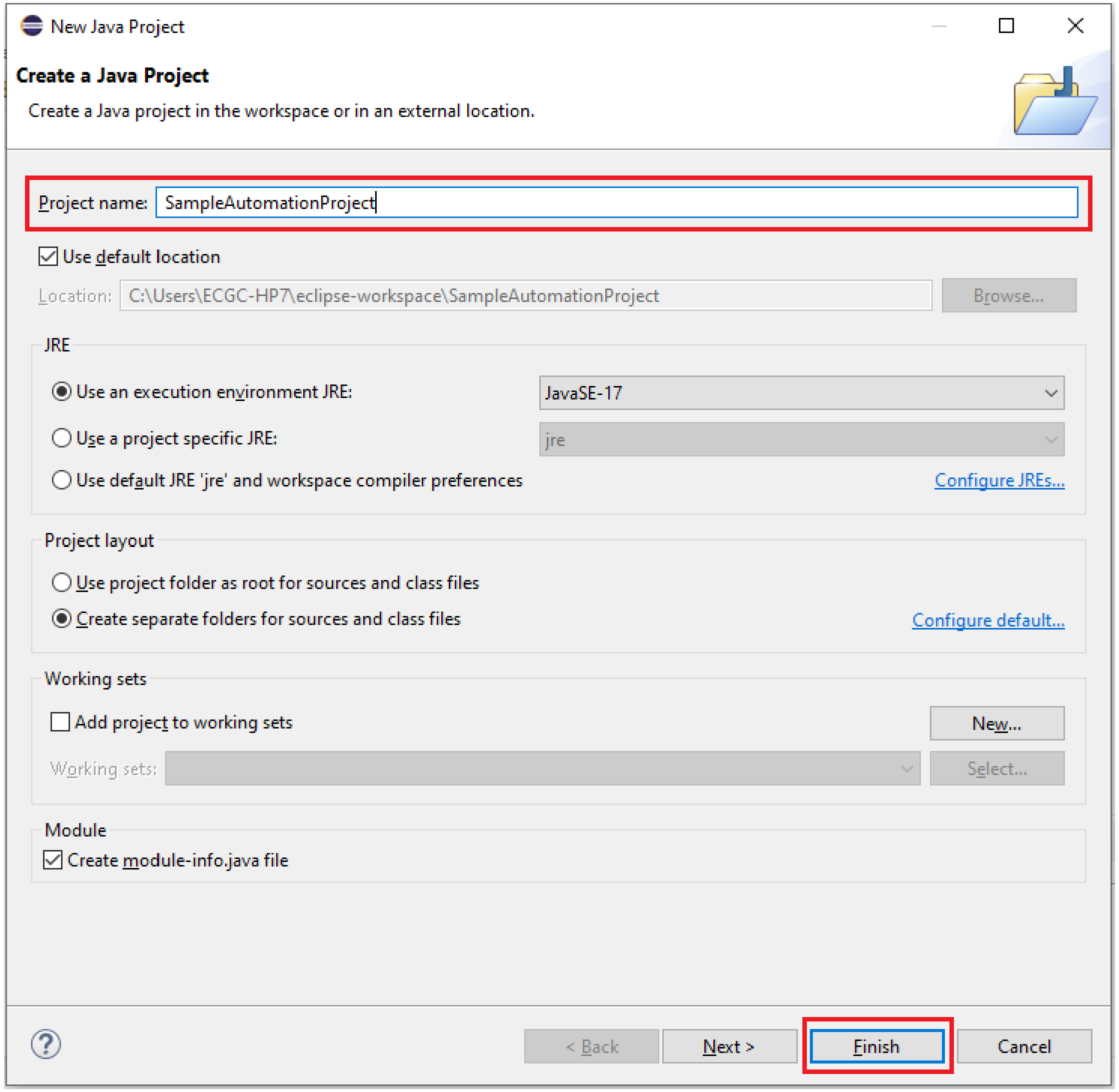
1. Launch **Eclipse** and select the default workspace.
Click on **Launch**

2. Click on **File -> New -> Java Project**



eclipse-workspace - Eclipse IDE

3. Enter a **Project name** and click on **Finish.**



New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: SampleAutomationProject

☑ Use default location

Location: C:\Users\ECGC-HP7\eclipse-workspace\SampleAutomationProject

JRE

● Use an execution environment JRE: JavaSE-17

○ Use a project specific JRE: jre

○ Use default JRE 'jre' and workspace compiler preferences

Project layout

○ Use project folder as root for sources and class files

● Create separate folders for sources and class files

Working sets

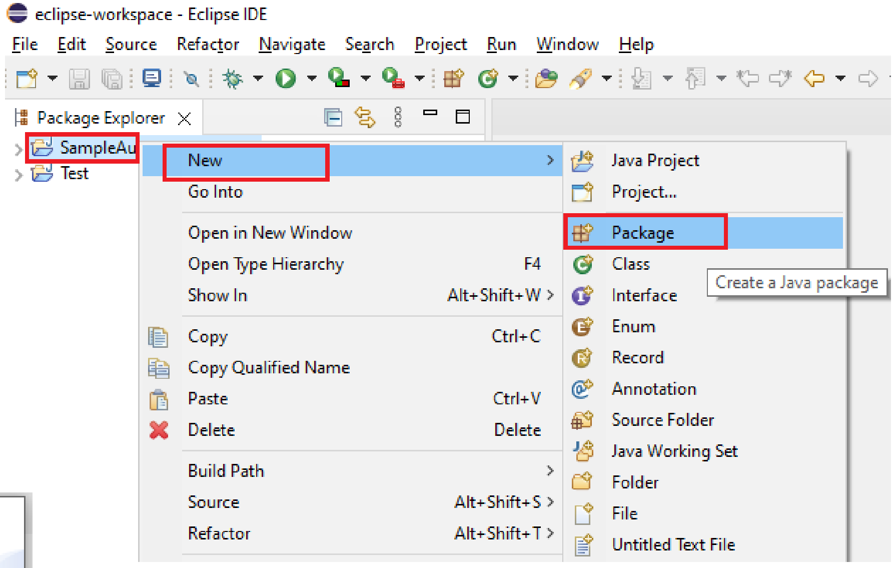
☐ Add project to working sets

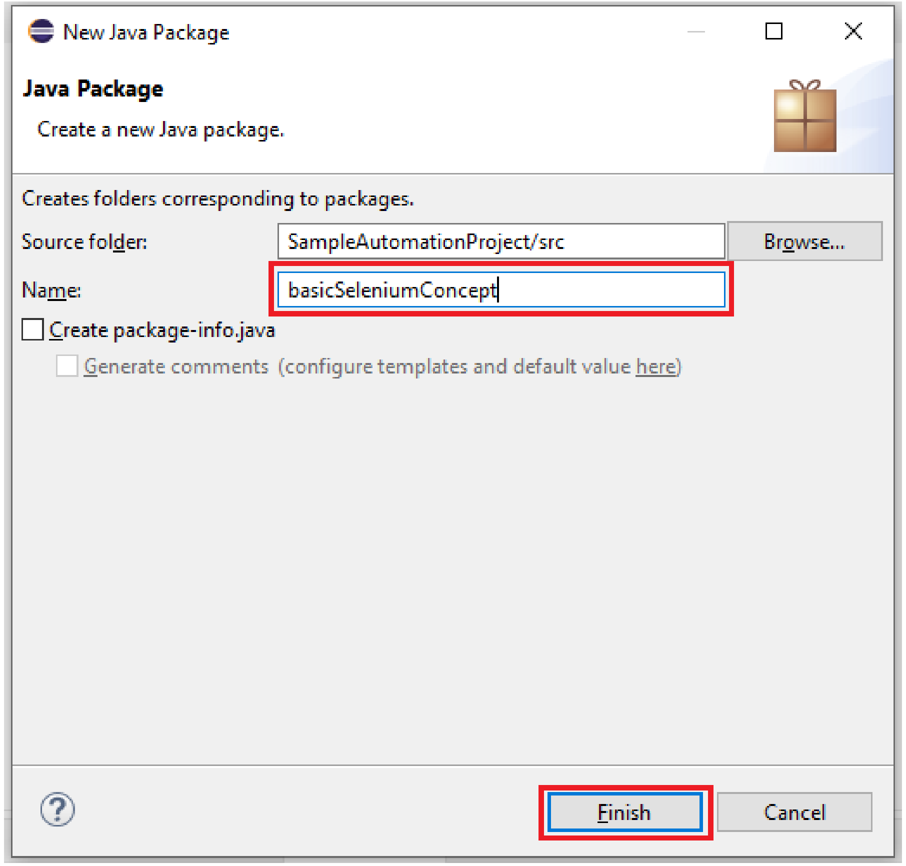
Working sets:

Module

☑ Create module-info.java file

4. **Right Click** on the newly created Project, then **New -> Package**





Java Package

Create a new Java package.

Creates folders corresponding to packages.

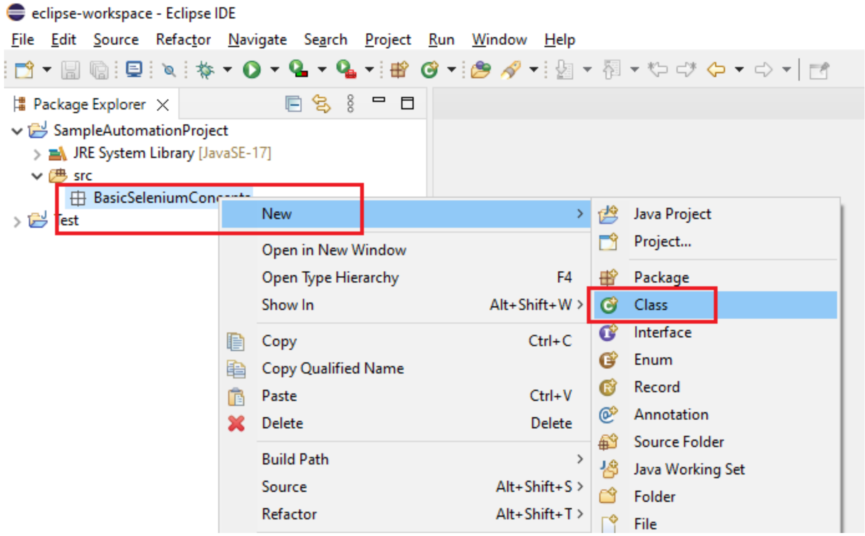
Source folder: SampleAutomationProject/src

Name: basicSeleniumConcept

☐ Create package-info.java

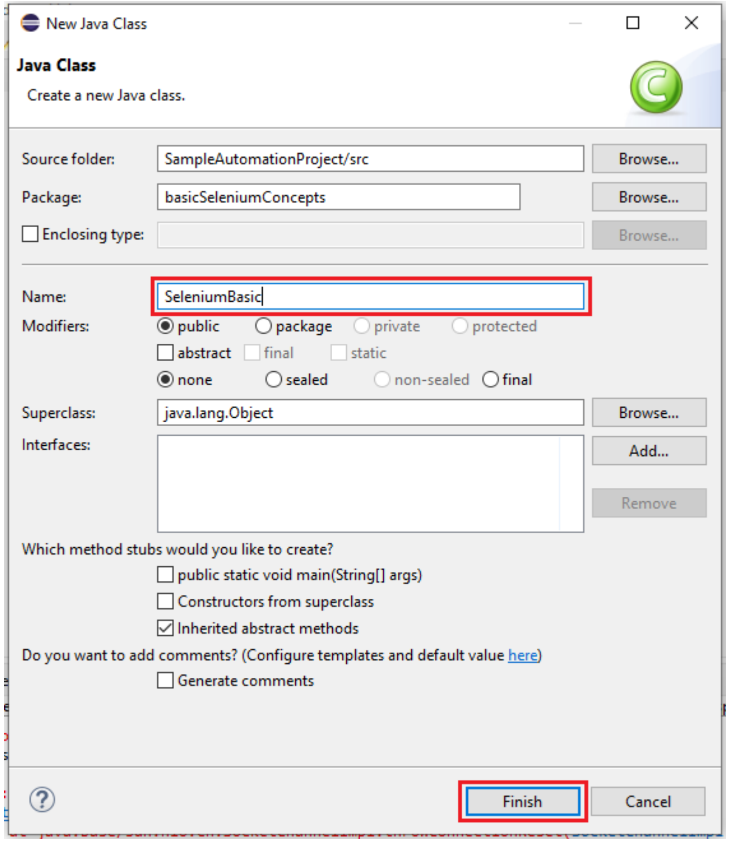
☐ Generate comments (configure templates and default value here)

5. Enter **Package Name** and click on **Finish.**



6. **Right Click** on newly created package and then **New -> Class**

7. Enter **Class Name** and then click on **Finish.**



Source folder: SampleAutomationProject/src

Package: basicSeleniumConcepts

□ Enclosing type:

Name: SeleniumBasic

Modifiers: ● public ○ package ○ private ○ protected

□ abstract □ final □ static

● none ○ sealed ○ non-sealed ○ final

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

□ public static void main(String[] args)

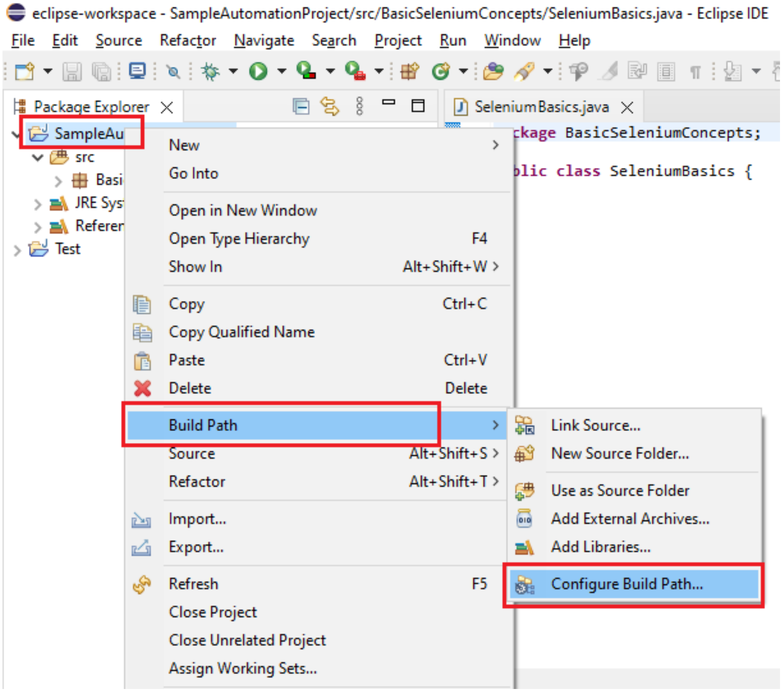
□ Constructors from superclass

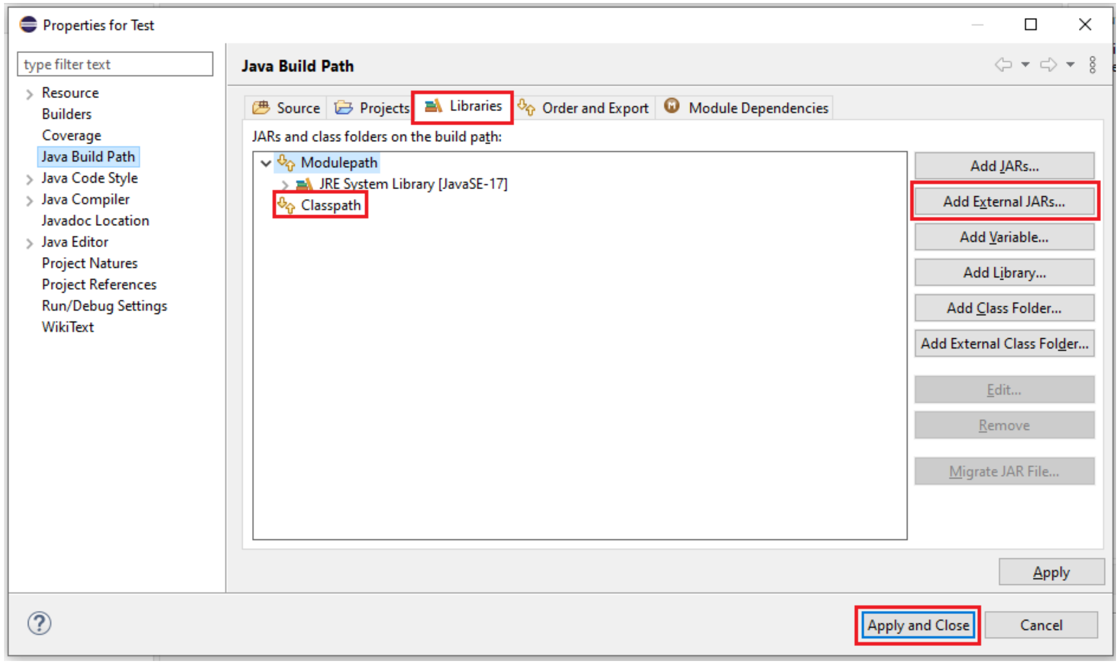
☑ Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)

□ Generate comments

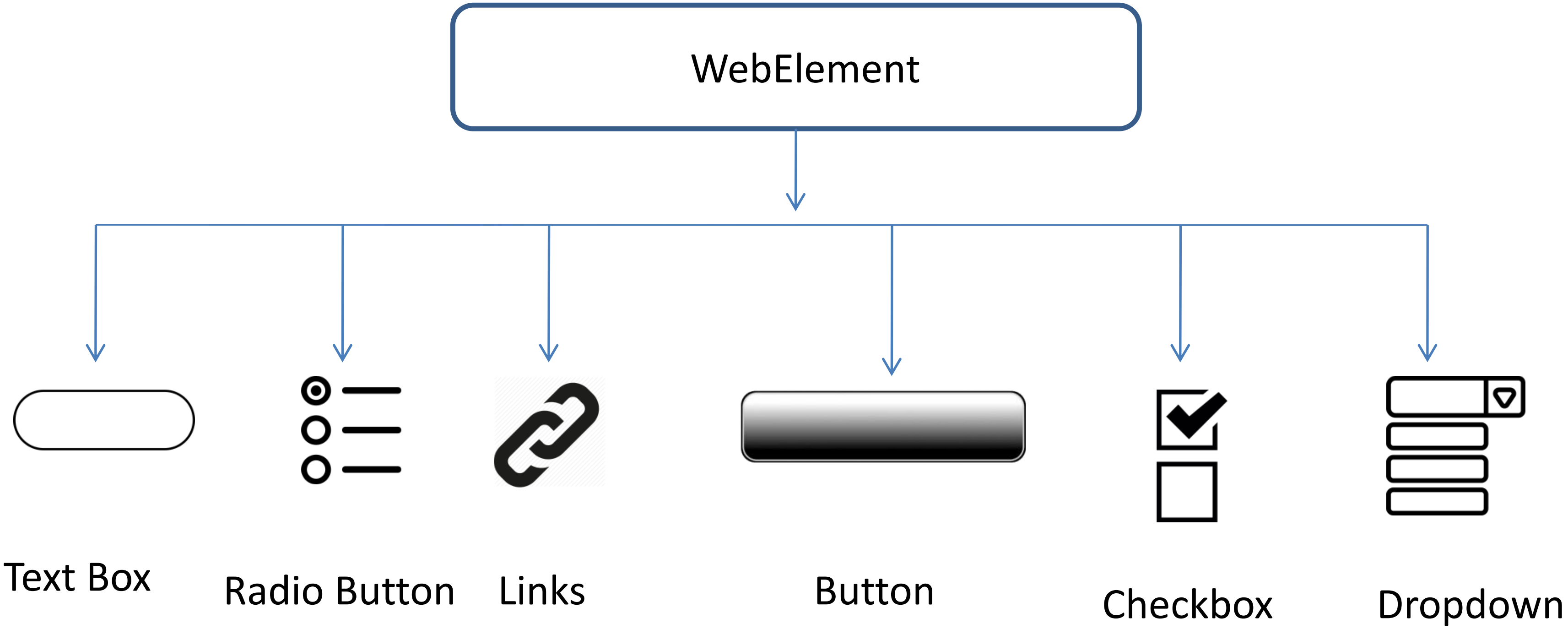
8. **Right Click** on project folder and then **Build Path -> Configure Build Path**





9. Click on **Libraries** and then Select **Classpath,** then click On **Add External JARs.** Add the downloaded Selenium Jar files and then **Apply and Close.**

WEBELEMENT & ITS TYPES



WebElement

Text Box

Radio Button

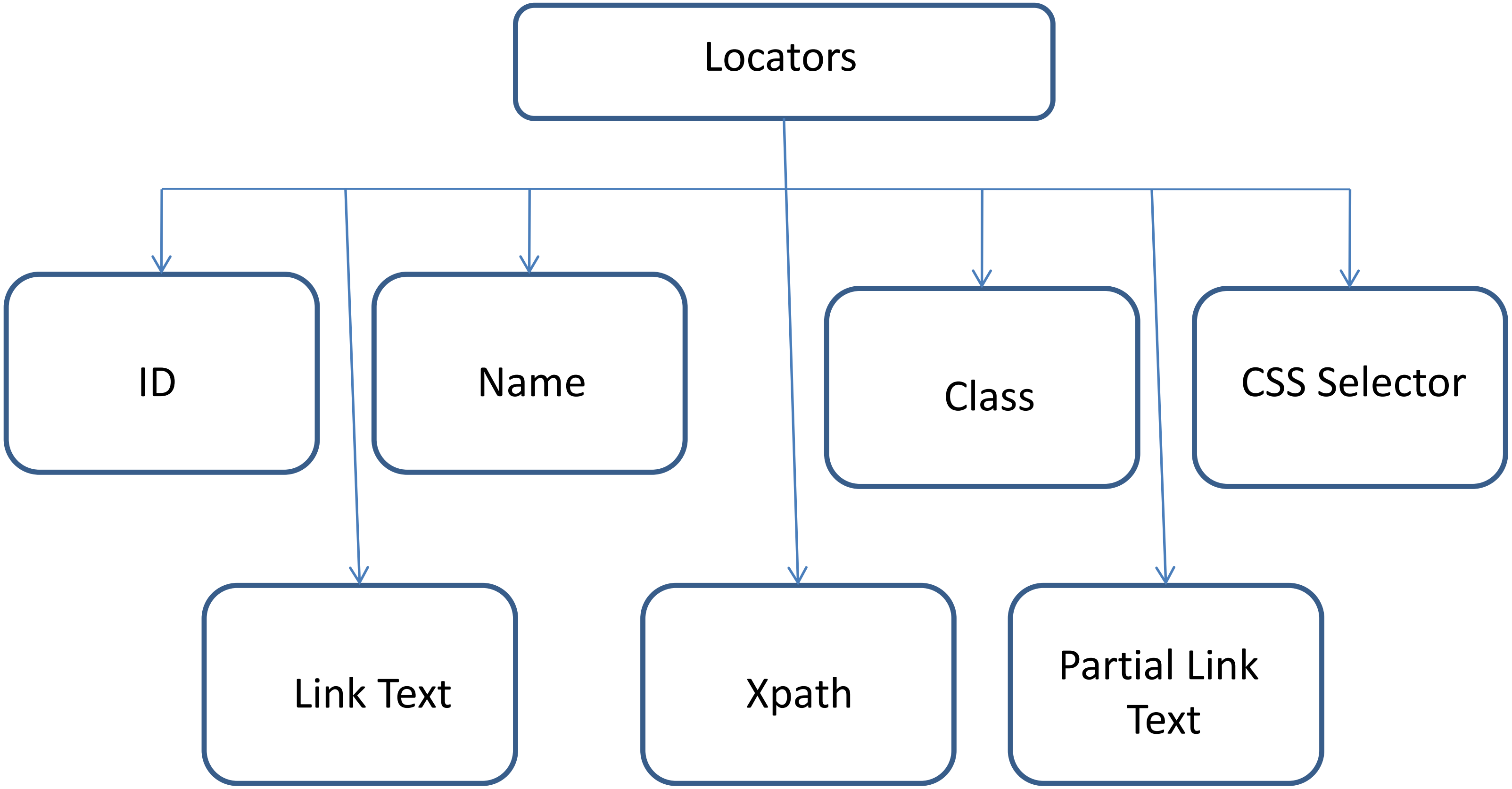
Links

Button

Checkbox

Dropdown

LOCATORS & ITS TYPES



Locators

ID

Name

Class

CSS Selector

Link Text

Xpath

Partial Link Text

XPATH

- Direct way to select an element
- Starts from the root node
- Uses single slash (/)

Absolute XPath

A

- Easy way to select an element
- Starts from middle of HTML DOM structure
- Uses double slash (//)

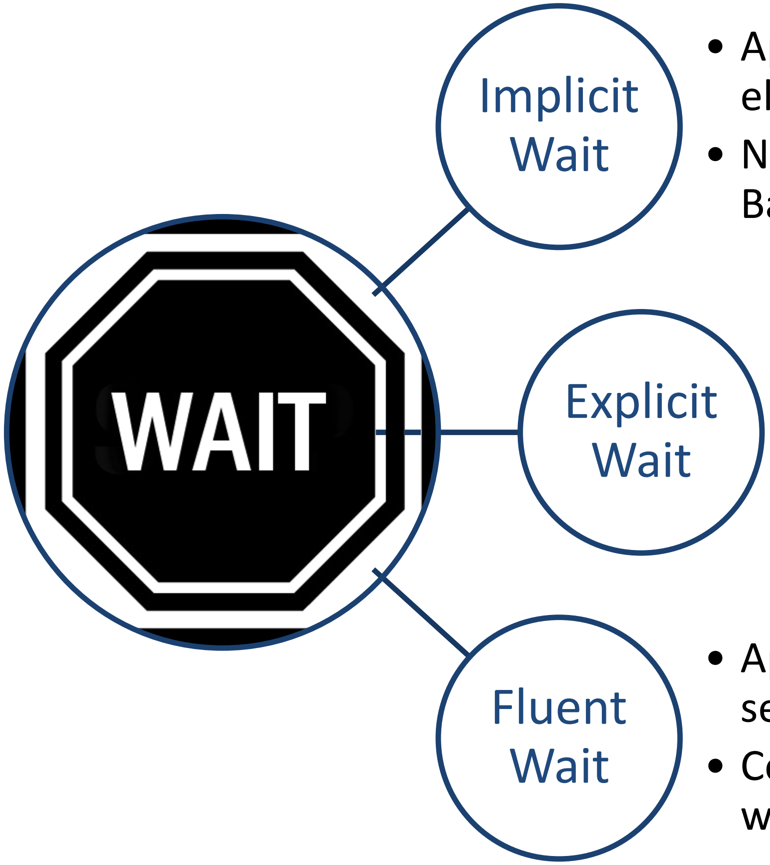
Relative XPath

R

RELATIVE XPATH

- Basic Xpath
- Contains
- OR & AND
- Starts-with
- Text()
- Last()
- Position()
- Following
- Preceding

WAITS



Implicit Wait

- Applies to all elements
- Non Condition Based

Explicit Wait

- Applies to selected element
- Condition Based

Fluent Wait

- Applies to selected element
- Condition Based with Poll Time