

Unit Testing - Recap

with JUnit

Unit Testing

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently analysed for proper operation.

Types of Testing

- Unit Testing
- Integration Testing
- Regression Testing
- ...



Integration Testing

Individual modules are combined and tested as a group.
Data transfer between the modules is tested as well.

Regression Testing

Regression means retesting the unchanged parts of the application.

Regression:
"when you fix one bug, you
introduce several newer bugs."



Regression Testing

Test cases are re-executed in order to check whether the previous functionality of the application is working fine and the new changes have not introduced any new bugs.

This test can be performed on a new build when there is a significant change in the original functionality that too even in a single bug fix.

Manual Testing

The oldest type of software testing.

It requires a tester to perform manual test operations on the test software without automation scripts.

The tester choose which tests to run, when to run them, and how many times.



Running tests...



Automated Testing

Automated testing

To automatically verify main functionality, ensure new version does not cause new defects, provide regression testing and help the teams to run a large number of tests in a short period of time.

Companies having great number of projects are looking for specialists in the field of automated testing.

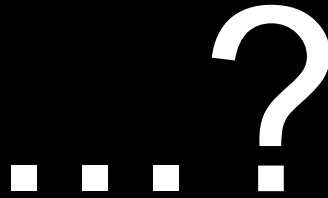
Automated testing tools



Continuous Integration

with Jenkins

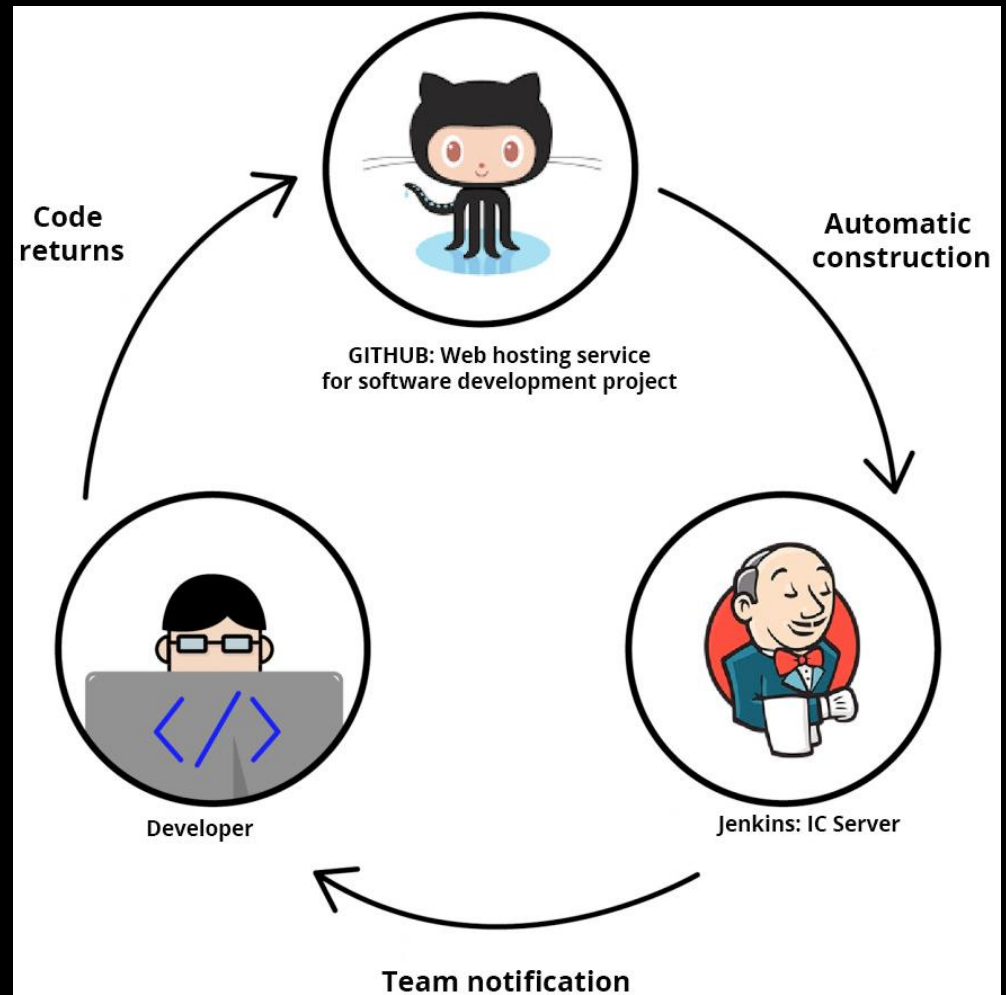
Continuous Integration



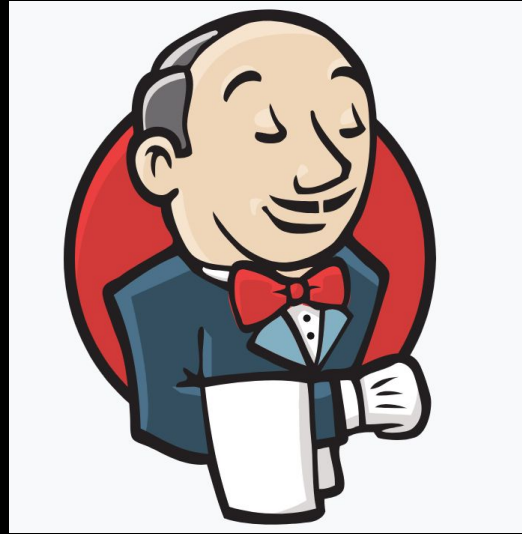
Continuous Integration

In its simplest form, it involves a tool that monitors your version control system for changes. Whenever a change is detected, this tool automatically compiles and tests your application. If something goes wrong, the tool immediately notifies the developers so that they can fix the issue immediately.

- Continuous Integration
- Github
- Jenkins



Jenkins



<https://jenkins.io/>

Jenkins

Plugins



<https://plugins.jenkins.io/>

Change UI Language

1. Click Manage Jenkins > Manage Plugins > ['Available' tab]
2. In the Filter, search for: Locale .
3. Click on Locale Plugin checkbox and Install without restart button.
4. After installation is complete:
 - Under Manage Jenkins > Configure System there should be a "Locale" section.
 - Enter the default language_LOCALE code for English: en_US
 - Click on Ignore browser preference and force this language to all users checkbox.

Create Job

Job Name --> Configure

Source Code Management

None

Git

Repositories

Repository URL

Credentials

Schedule a build

Jenkins uses a cron expression, and the different fields are:

1. MINUTES Minutes in one hour (0-59)
2. HOURS Hours in one day (0-23)
3. DAYMONTH Day in a month (1-31)
4. MONTH Month in a year (1-12)
5. DAYWEEK Day of the week (0-7) where 0 and 7 are sunday

MINUTES	HOURS	DAYMONTH	MONTH	DAYWEEK
<code>*/5</code>	<code>*</code>	<code>*</code>	<code>*</code>	<code>*</code>

To schedule a build every 5 minutes : `*/5 * * * *`

To schedule a build every day at 8h00, this will do the job : `0 8 * * *`

Configure JDK

Manage Jenkins > Global Tool Configuration

JDK

JDK installations

 JDK

Name

JAVA_HOME

Install automatically ?

List of JDK installations on this system

Configure JDK & Maven

Manage Jenkins --> Global Tool Configuration

Maven

Maven installations

Add Maven

Maven

Name

MAVEN_HOME



Install automatically

Delete Maven




Job Name --> Configure


Build

Invoke top-level Maven targets  

Maven Version

Goals 

Advanced...

Add build step 

Email Notifications

Manage Jenkins → Configure System

E-mail Notification

SMTP server

Default user e-mail suffix

Use SMTP Authentication

User Name

Password

Use SSL

Use TLS

SMTP Port

Reply-To Address

Charset

Job Name --> Configure

Post-build Actions

E-mail Notification ✕ ?

Recipients

Whitespace-separated list of recipient addresses. May reference build parameters like \$PARAM. E-mail will be sent when a build fails, becomes unstable or returns to stable.

Send e-mail for every unstable build

Send separate e-mails to individuals who broke the build

Add post-build action ▾

Webhook

- Webhooks allow you to build or set up GitHub Apps which subscribe to certain events on GitHub.com.
- When one of those events is triggered, we'll send a HTTP POST payload to the webhook's configured URL.
- Webhooks can be used to update an external issue tracker, trigger CI builds, update a backup mirror, or even deploy to your production server.

<https://developer.github.com/webhooks/>



Search or jump to...

[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)



[FabrizioFornari](#) / [BasicJUnitTests](#)

[Unwatch](#) 1

[★ Star](#) 1

[Fork](#) 1

[Code](#)

[Issues](#) 0

[Pull requests](#) 0

[Projects](#) 0

[Wiki](#)

[Insights](#)

[Settings](#)

[Options](#)

[Collaborators](#)

[Branches](#)

[Webhooks](#)

[Integrations & services](#)

[Deploy keys](#)

[Moderation](#)

[Interaction limits](#)

Webhooks / Manage webhook

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#).

Payload URL *

Content type

Secret

Which events would you like to trigger this webhook?

- Just the push event.
- Send me **everything**.
- Let me select individual events.

Active

We will deliver event details when this hook is triggered.

[Update webhook](#)

[Delete webhook](#)

Jenkins on Tomcat

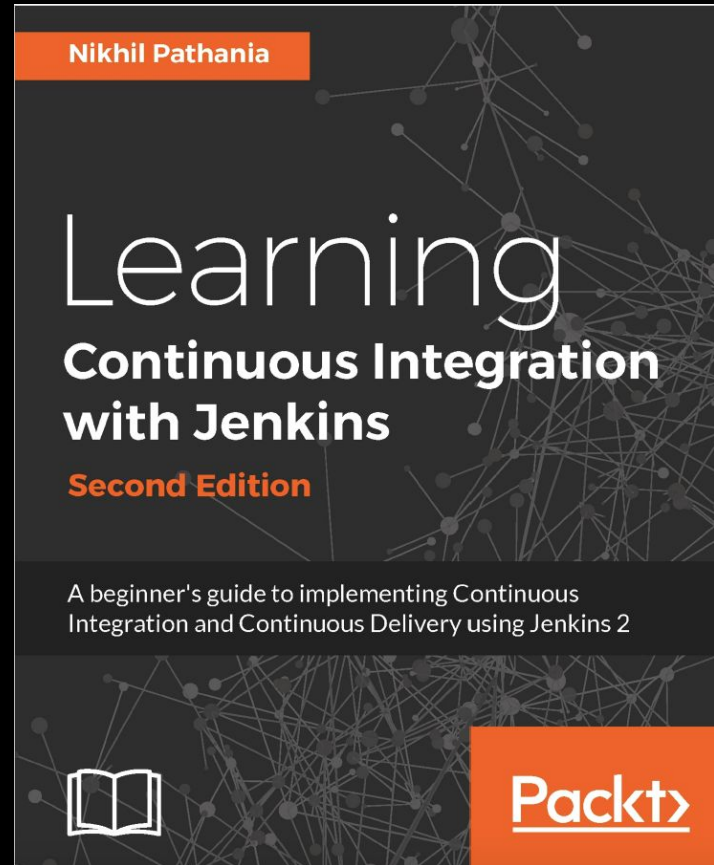
Download the latest .war file

<https://www.jenkins.io/doc/book/installing/#war-file>

Deploy the .war on Tomcat

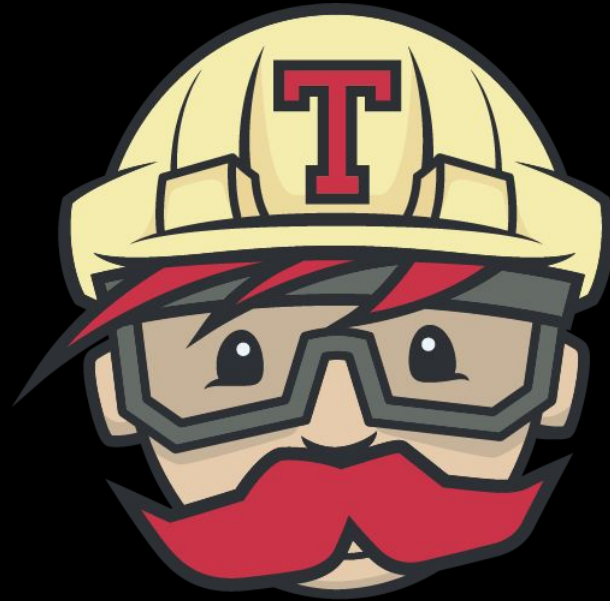


Suggested Book



Last but not
least...

Not only Jenkins



TRAVIS CI

<https://travis-ci.org/>

Travis CI

Travis CI is a hosted, distributed continuous integration service used to build and test software projects hosted at GitHub.

Travis CI is configured by adding a file named `.travis.yml`, which is a YAML format text file, to the root directory of the repository.

```
# this is a java project using maven
language: java
# install
install: mvn install
```

Travis CI Documentation: <https://docs.travis-ci.com/>

What is Web Testing?



What is Selenium?



Selenium

is a chemical element with symbol **Se** and atomic number **34**

group	1*	2											13	14	15	16	17	18				
1*	Ia	IIa											IIIb	IVb	Vb	VIb	VIIb	VIIIb				
1	Ia	IIa											IIIa	IVa	Va	VIa	VIIa	0				
1	H																	He				
2	Li	Be											B	C	N	O	F	Ne				
3	Na	Mg	IIIa**	IVa	Va	VIa	VIIa	VIIIa	IXa	Xa	XIa	XIIa	Al	Si	P	S	Cl	Ar				
			IIIb***	IVb	Vb	VIb	VIIb	VIIIb	IXb	Xb	XIb	XIIb										
4	K	Ca	Sc	Ti	V	Cr	Mn	Fe	Co	Ni	Cu	Zn	Ga	Ge	As	Se	Br	Kr				
5	Rb	Sr	Y	Zr	Nb	Mo	Tc	Ru	Rh	Pd	Ag	Cd	In	Sn	Sb	Te	I	Xe				
6	Cs	Ba	La	Hf	Ta	W	Re	Os	Ir	Pt							Bi	Po	At	Rn		
7	Fr	Ra	Ac	****	****	****	****	****	****	****	****	****										
6				58	59	60	61	62	63	64							69	70	71			
				Ce	Pr	Nd	Pm	Sm	Eu	Gd							Tm	Yb	Lu			
7				90	91	92	93	94	95	96	97	98	99	100	101	102	103					
				Th	Pa	U	Np	Pu	Am	Cm	Bk	Cf	Es	Fm	Md	No	Lr					

34
Se

alkali metals
 alkaline earth metals
 transition metals
 other metals
 other nonmetals
 halogens
 actinides
 noble gases
 lanthanides

* Numbering system recommended by the International Union of Pure and Applied Chemistry (IUPAC)

** Previous IUPAC numbering system

*** Numbering system recommended by the Chemical Abstracts Service

What is Selenium?

Selenium is a suite of tools to automate web browsers across many platforms.

We will use it for automating tests of web applications

Why is it called Selenium then?

Why is it called Selenium then?

During the development of selenium core 2004 there was another competitive product developed by a company called **Mercury Interactive**.

A joke by Jason Huggins saying that *“mercury poisoning can be cured by taking selenium supplements”*.

Selenium is used to remove the toxic content mercury from the human body, so Jason coined the term Selenium for their creative open-source project.

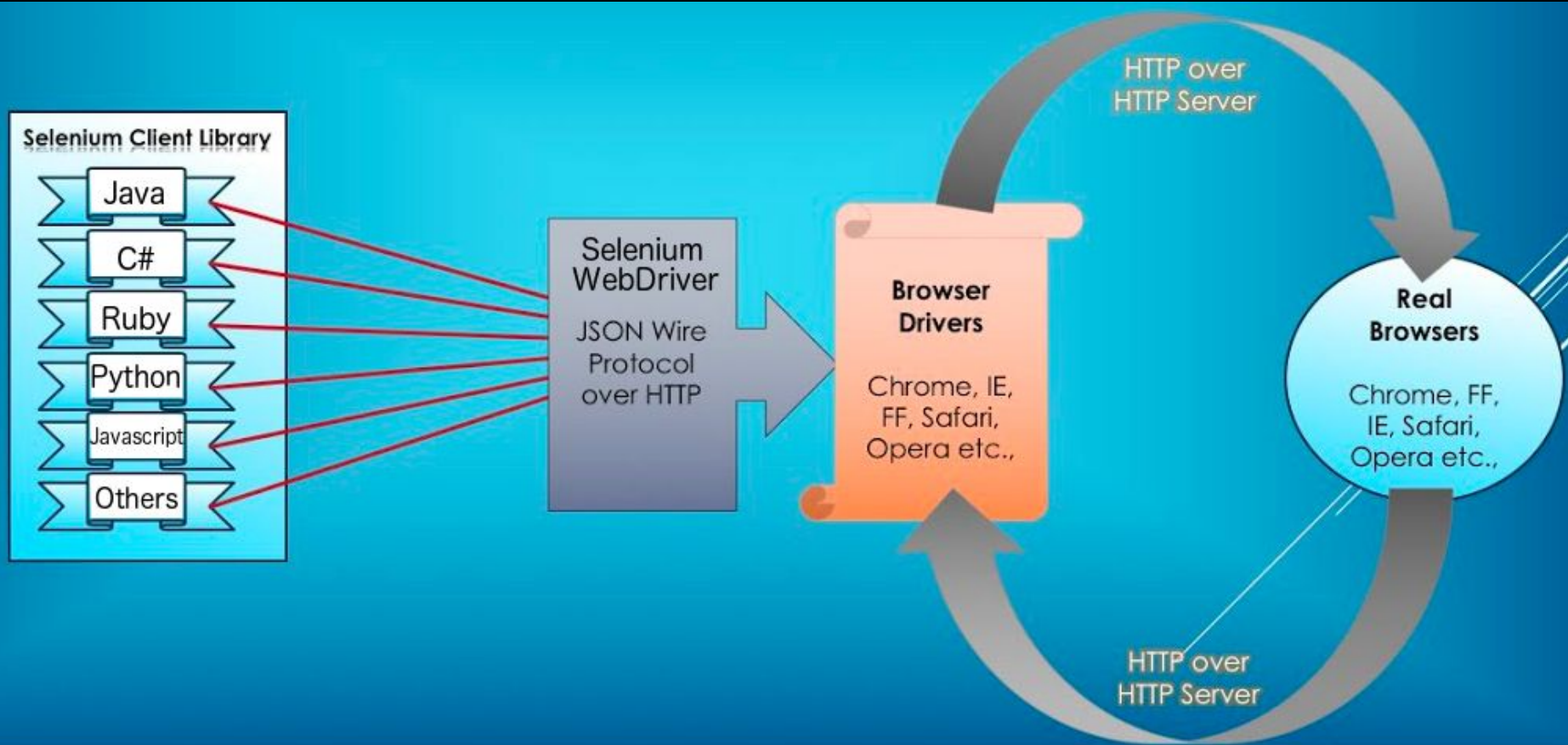
Which is the idea?

to remotely control browsers so that we can do things like write automated tests for the content they run or tests for the browser UI itself.

Should I write a test in a different way per each browser that is out there? No, to this end, a group of people from several organizations is working on the WebDriver Specification.

<https://w3c.github.io/webdriver/>

Selenium Architecture



Selenium WebDriver

Selenium WebDriver provides APIs so that you can write code in your favourite language to simulate user actions like this:

```
client.get("http://pros.unicam.it/")
link = client.find_element_by_id("participate")
link.click()
```

Underneath that API, commands are transmitted via JSON over HTTP.

For example,
to tell the browser to navigate to a url, a client sends a POST request to the endpoint */session/{session id of the browser instance you're talking to}/url* with body *{"url": "http://pros.unicam.it/"}*.

Browser Driver

At the beginning Selenium had to develop drivers for some browser they wanted to interact with.

Then, browser vendors started implementing the Selenium JSON Wire Protocol themselves!

This makes a lot of sense: they're in the best position to maintain the server side and they can build the necessary behaviour directly into the browser.

It started with OperaDriver in 2009-2011, and then others followed such as ChromeDriver and Mozilla's geckodriver with Marionette. This is where the motivation for a WebDriver standard comes from.

Selenium

GO TO: <https://www.seleniumhq.org/>

Hands on!

Clone

<https://github.com/FabrizioFornari/PracticeWithSelenium.git>

Selenium Locators



Preferred selector order : id > name > css > xpath

HTML

- Html is a standard markup language for creating Web pages. Html elements are the building blocks of HTML pages. HTML tags label pieces of content, such as “heading”, “paragraph”, “table”, and so on.
- HTML elements usually consists of start tag and end tag with content inserted between them.
- For example:
 - `<h1>` An example for an HTML title `</h1>`
 - `<p>` An example for an HTML paragraph `</p>`

CSS

- CSS is a language that describes the style of an HTML document.
- CSS describes how HTML elements should be displayed.
- A CSS rule-set consists of a selector and a declaration block. For example: selector - h1, declaration – {color:blue;}.

```
<style>

body {
  background-color: lightblue;
}

</style>
```

CSS

- CSS selectors are used to "find" (or select) HTML elements based on their element name, id, class, attribute, and more.
- Some Selectors:
 - Element selector : The element selector selects elements based on the element name. for example: p, h1 etc.
 - ID selector: The id selector uses the id attribute of an HTML element to select a specific element. For example: #id1, #para2 etc.
- A CSS rule-set consists of a selector and a declaration block. For example: selector - h1, declaration – {color:blue;}.

XPath

- XPath stands for XML Path Language, it can be used to navigate through elements and attributes in an XML document.
- XPath uses path expressions to select nodes or node-sets in an XML document

Syntax:

- // - Selects nodes in the document from the current node that match the selection no matter where they are.
- @ - Selects attributes

XPath

- Example

```
<!DOCTYPE html>
<html>
<head>
<title> Page Title</title>
</head>
<body>
<h1 id="h1_id"> This is part of the presentation title</h1>
<p>This is the paragraph.</p>
</body>
</html>
```

The query: `//h1[@id='h1_id']` will get the h1 element

Selenium WebDriver

- A Selenium Web driver must be created
- For using Chrome:
 - `System.setProperty("webdriver.chrome.driver", projectPath+"/drivers/chromedriver");*`
 - `WebDriver driver = new ChromeDriver();`
- Interaction with the Chrome instance will be made in the code on the driver.

***Note:** you need to specify, before instantiating the `WebDriver`, the path to the actual driver that you downloaded following instructions from the selenium website <https://www.seleniumhq.org/download/>.

Selenium WebDriver

- Navigation using a Selenium WebDriver is very simple, given a defined URL. It can be done in two ways, `driver.get(...)` or `driver.navigate().to(...)`
 - `driver.get("https://www.google.com/");`
 - `driver.navigate().to("https://www.google.com/");`
- *The `driver.get(...)` and `driver.navigate().to(...)` do exactly the same thing. `driver.navigate()` supports also `driver.navigate().forward()` and `driver.navigate().backward()`*

Finding Web Elements

- Web elements can be defined as each opening and closing tags in the web page. For example: `<button>Click Me</button>` - a web element.
- Finding a web element and interacting with it can be done in several ways: - ID. - Class. - Name. - Xpath. - Css Selector, etc.

Finding Web Elements

- An example:
 - Assuming that we have the following web page:

```
<html>  
  <body>  
    <button id= "my_button"> Click Me</button>  
  </body>  
</html>
```
 - The following lines of code will be used for clicking the button:

```
WebElement button = driver.findElement(By.id(" my_button "));  
button.click();
```

Selenium IDE

Download it from:

<https://www.seleniumhq.org/selenium-ide/>

and let us see what we can do with it...

Katalon Recorder

Download the extension for the browser you want to use

Try to record the same test

Try to find any difference