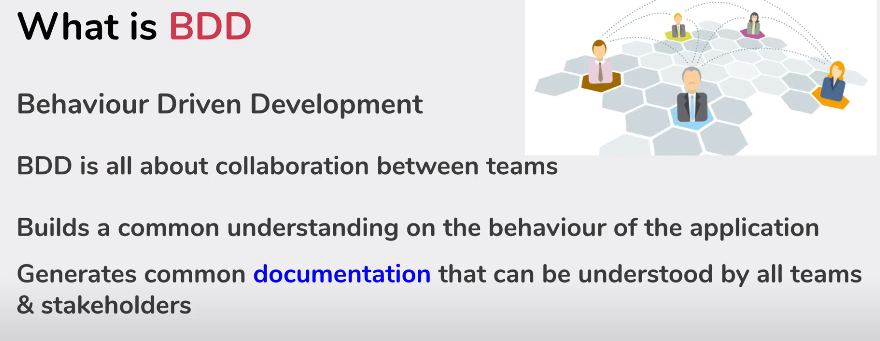
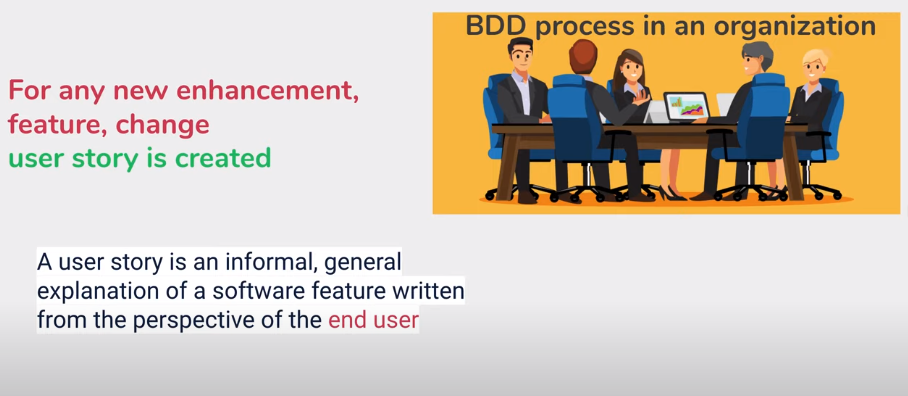
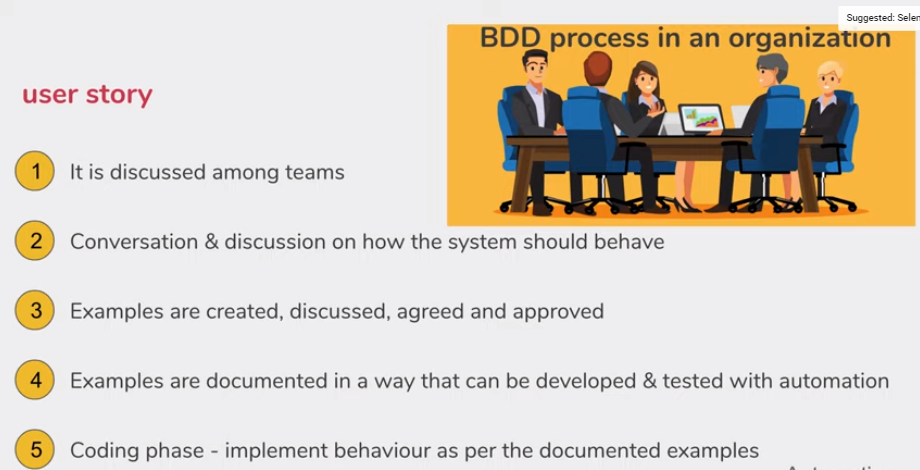
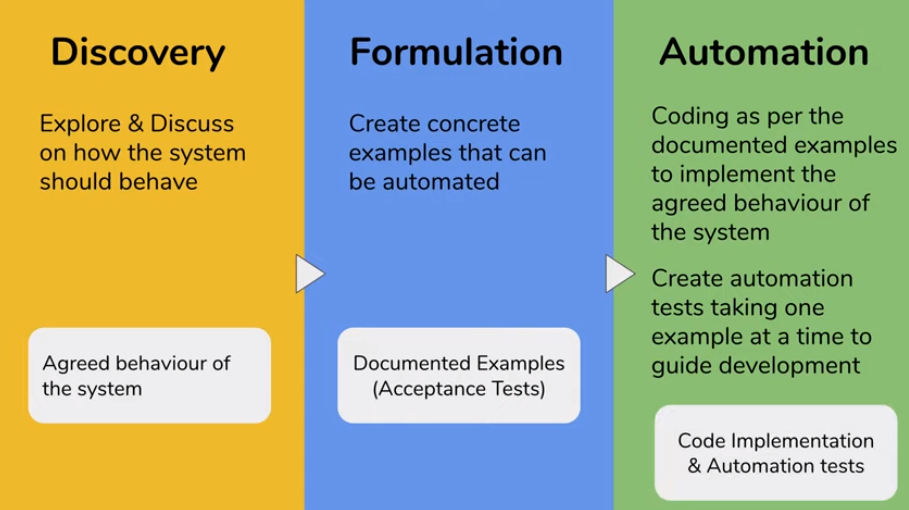
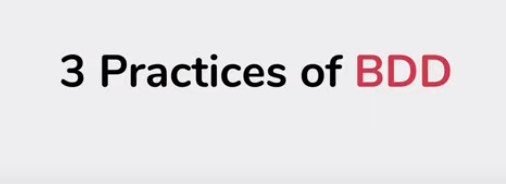
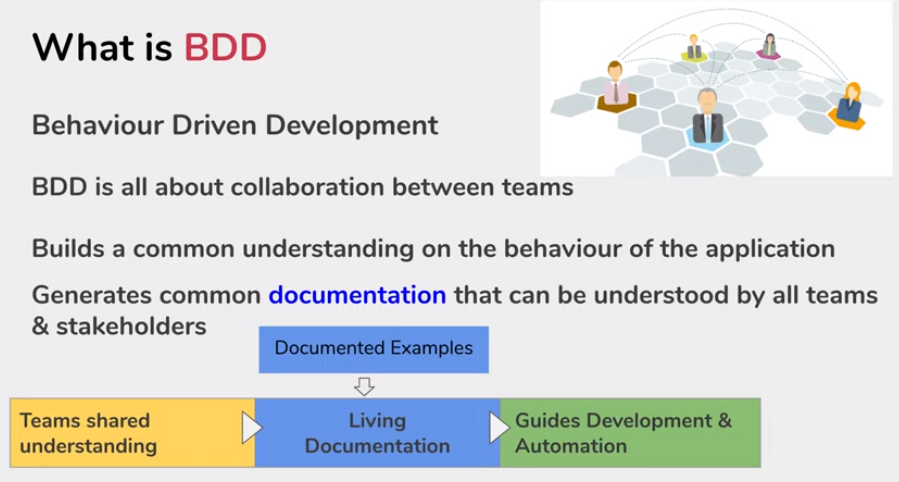
BDD & Cucumber ( Cucumber is for JAVA Selenium)

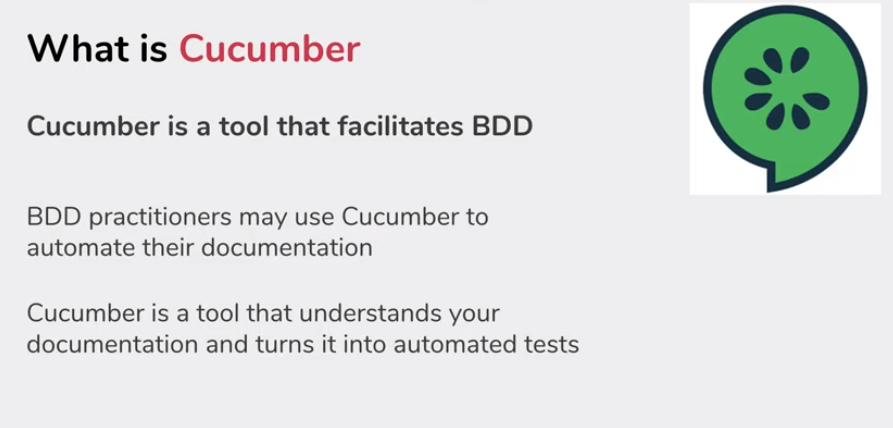












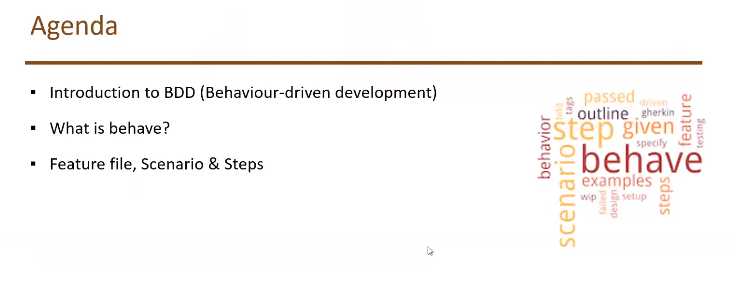
**SDET Video Series on Selenium with Python Behave (BDD Framework):** <https://www.youtube.com/watch?v=JIyvAFBx2Fw&list=PLUDwpEzHYYLsARXz1o3Ldt1FnvRbvlxsS>

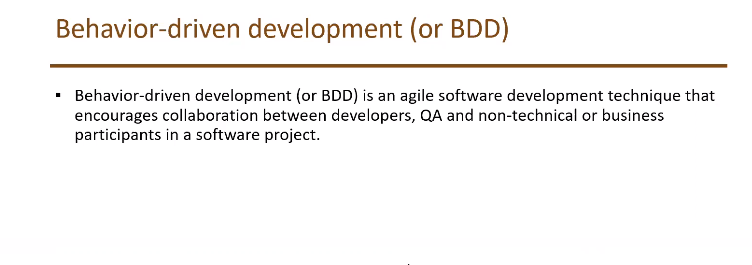
**(7 Video Series)**

Official website for behave: <https://pypi.org/project/behave/>

# **Part 1: Selenium with Python Behave (BDD) Introduction**

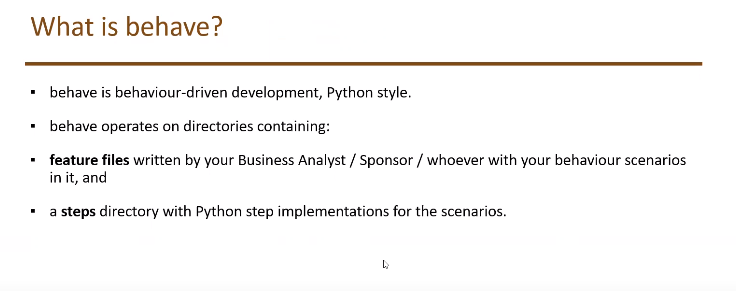






Feature files are nothing but scenarios written in English text, so that everyone can understand. We follow Gherkins language keywords to write feature files.

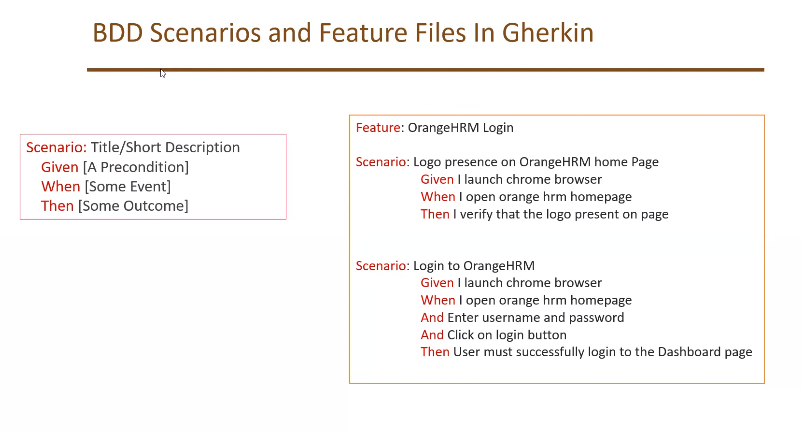
Step directory will contain steps cover a particular scenario written in feature files. We will be importing feature files in steps directory/file.



In BDD framework we need to understand 3 components.

1. Feature files
2. Scenarios
3. Steps to cover scenarios

Note: Feature file contains multiple scenarios and it has .feature extension.



feature file contains feature to be tested written in Gherkins keywords.

1. Given – Represents given condition.
2. When – Represents action to be taken.
3. And – used when multiple actions are mentioned in When keyword
4. Then – Represent outcome.

## Steps file contains python automation code.

## Generally we write methods in steps.py file to cover a particular feature file.

**Example of behave features file.**

## Project description

behave is behavior-driven development, Python style.

Behavior-driven development (or BDD) is an agile software development technique that encourages collaboration between developers, QA and non-technical or business participants in a software project.

behave uses tests written in a natural language style, backed up by Python code.

First, [install \*behave\*.](http://pythonhosted.org/behave/install.html)

Now make a directory called “features/”. In that directory create a file called “example.feature” containing:

# -- FILE: features/example.feature

Feature: Showing off behave

Scenario: Run a simple test

Given we have behave installed

When we implement 5 tests

Then behave will test them for us!

Make a new directory called “features/steps/”. In that directory create a file called “example\_steps.py” containing:

# -- FILE: features/steps/example\_steps.py

from behave import given, when, then, step

@given('we have behave installed')

def step\_impl(context):

pass

@when('we implement {number:d} tests')

def step\_impl(context, number): # -- NOTE: number is converted into integer

assert number > 1 or number == 0

context.tests\_count = number

@then('behave will test them for us!')

def step\_impl(context):

assert context.failed is False

assert context.tests\_count >= 0

Run behave:

$ behave

Feature: Showing off behave # features/example.feature:2

Scenario: Run a simple test # features/example.feature:4

Given we have behave installed # features/steps/example\_steps.py:4

When we implement 5 tests # features/steps/example\_steps.py:8

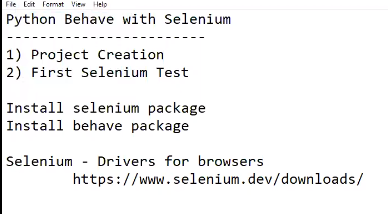
Then behave will test them for us! # features/steps/example\_steps.py:13

1 feature passed, 0 failed, 0 skipped

1 scenario passed, 0 failed, 0 skipped

3 steps passed, 0 failed, 0 skipped, 0 undefined

# **Part 2: Selenium with Python Behave (BDD) | Environment Setup**



Install following things

1. Python
2. PyCharm
3. Selenium library
4. Behave
5. Install python from official website if not installed.

How to set a Python Environment path.

Go to ‘This PC’ icon and right click on it 🡪 go to properties 🡪 go to advanced system setting from right side panel 🡪 click on environment variables 🡪 in system variables click on path variable 🡪 then click on edit button 🡪 then click on new 🡪 add python folder path there 🡪 click on OK 🡪 again 🡪 in system variables click on path variable 🡪 then click on edit button 🡪 then click on new 🡪 add python scripts folder path there 🡪 Complete.

1. Install Pycharm if not installed.
2. Install selenium

In cmd type

pip install selenium or pip3 install selenium

Incase to update already downloaded library

pip install -U selenium or pip3 install -U selenium

1. Install behave

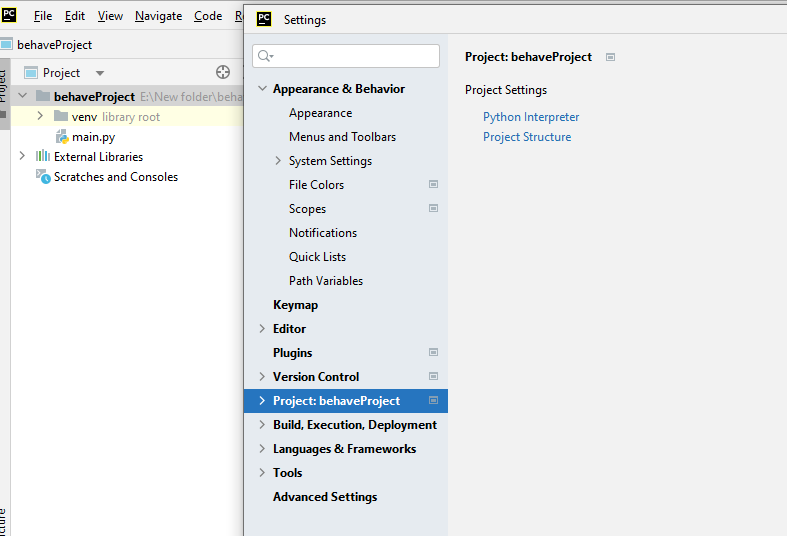
In cmd type

Pip install behave or pip3 install behave

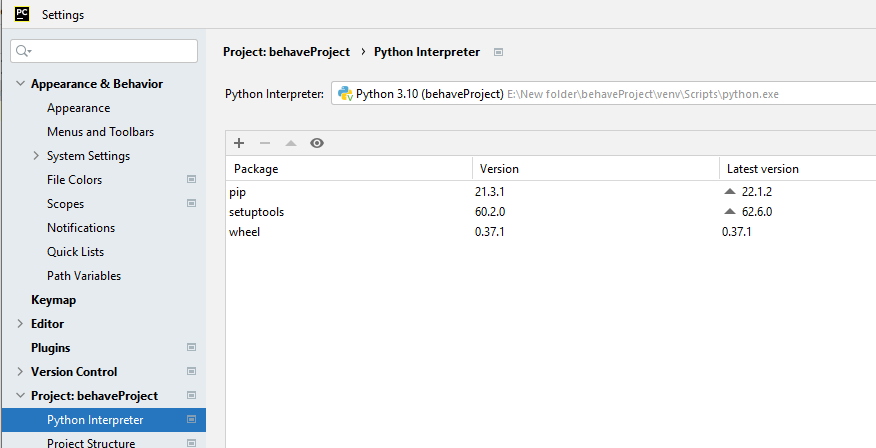
* After completion of installation, launch PyCharm and create a new project.
* Give name as – behaveProject

Once the Project is created add interpreter.

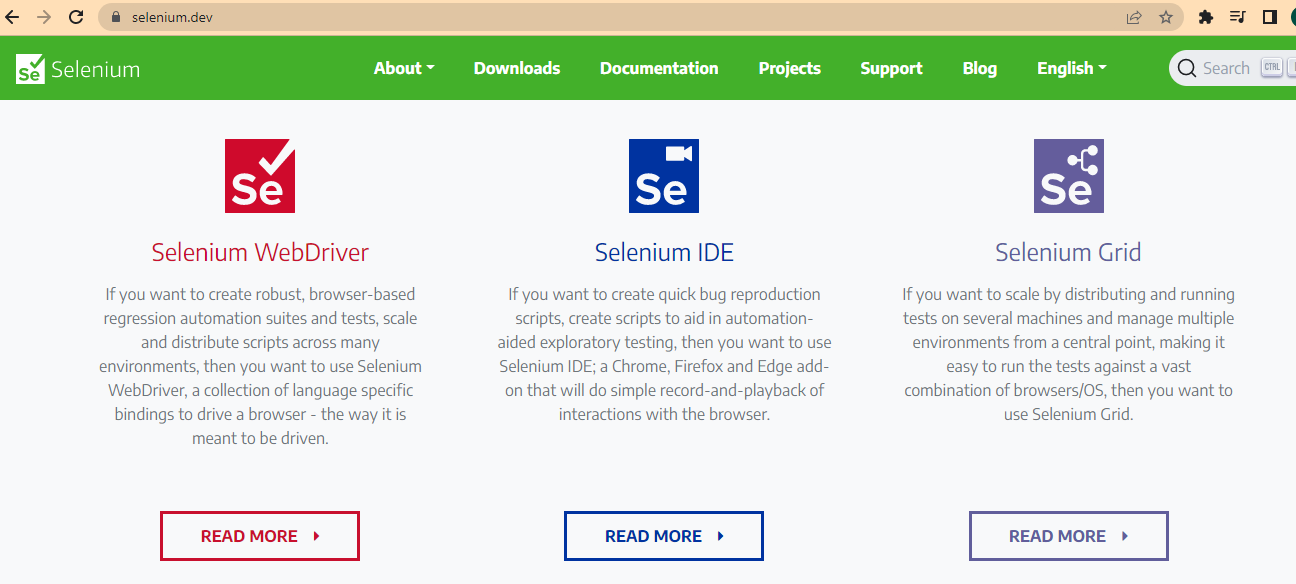
In PyCharm go to file 🡪 settings 🡪 Project 🡪 python interpreter 🡪



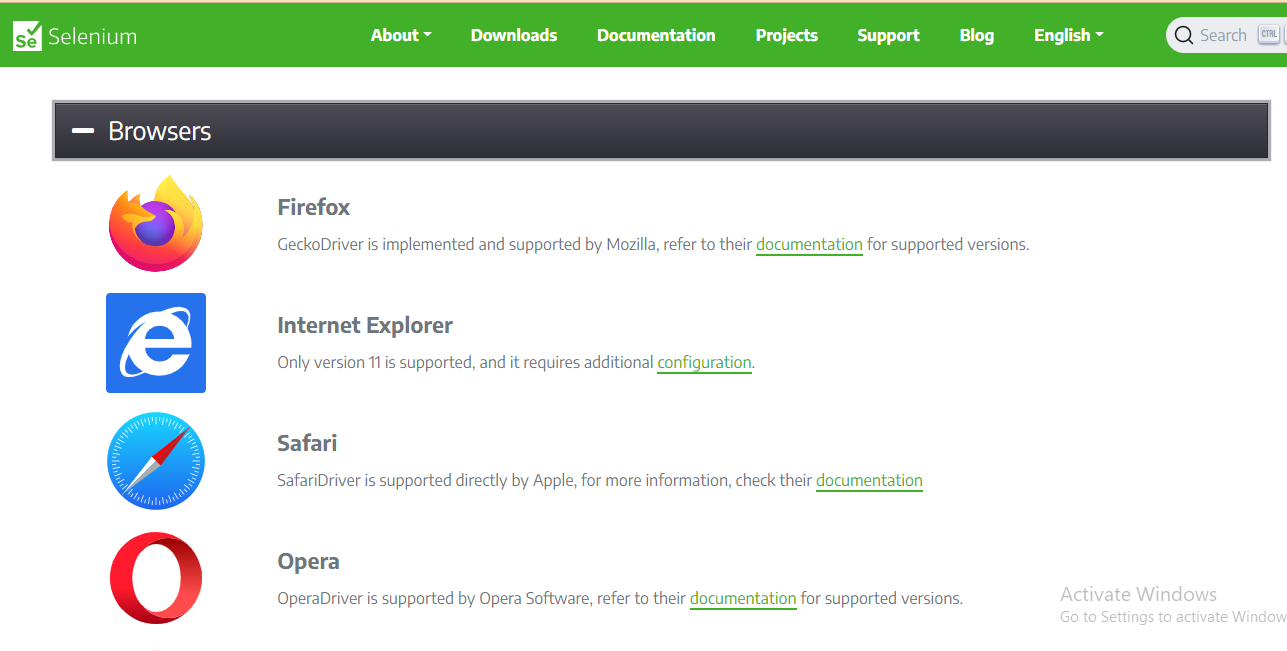
* Click on + icon 🡪 type selenium and install it 🡪then type behave and install it 🡪 click OK and close the window 🡪 now the project is created with behave and selenium



Then we need to download web driver for selenium to control browser. Go to selenium official website i.e. selenium.dev

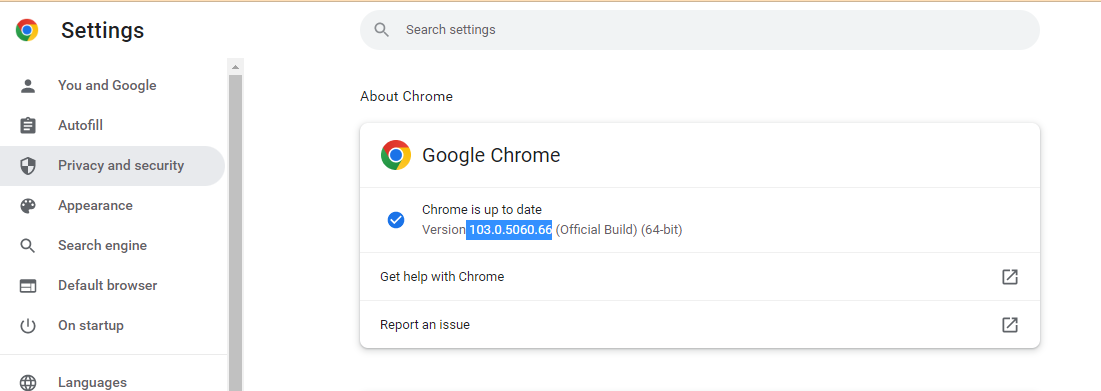


click on downloads 🡪 a new page will open 🡪 scroll down and click on browsers 🡪 then click on respective browser documentation 🡪 download respective chrome webdriver from there.



We can check which chrome browser we are using.

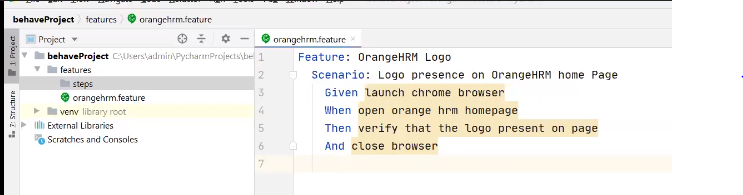
In google click on 3 dots available on top right side 🡪 go to helps 🡪 About Google Chrome.



Now we can begin creating our project.

In behaveProject create a new directory and give name as features. 🡪 inside features create 1 more directory and name it as steps as shown below.

Also create first feature file as shown, make sure to use Gherkins keywords to write scenarios.

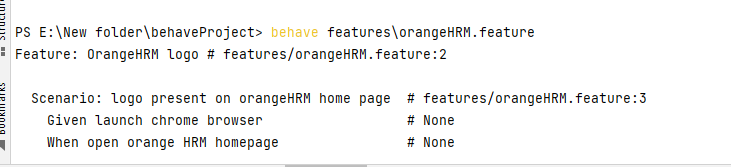


to execute feature file type following command in terminal window

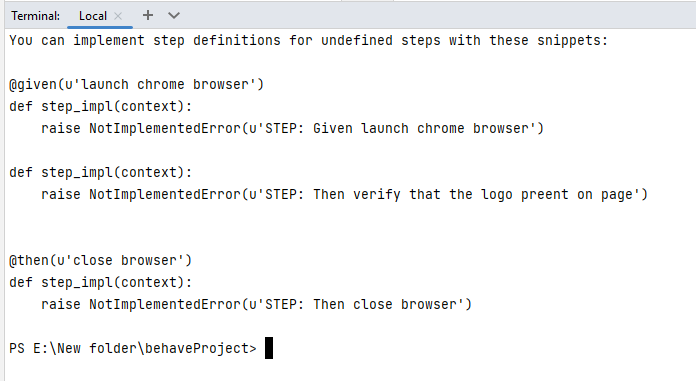
behave features\orangeHRM.feature

it will through error coz steps file is not created.

Note: if we have only 1 feature, we need to specify the location. We can simply give command as “behave” in terminal window to execute the file.



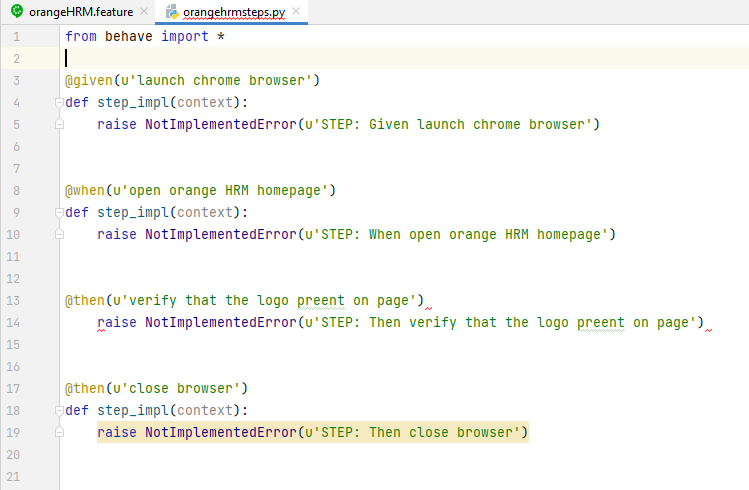
We get suggestion how to create steps file in terminal window itself when it is not created. As shown



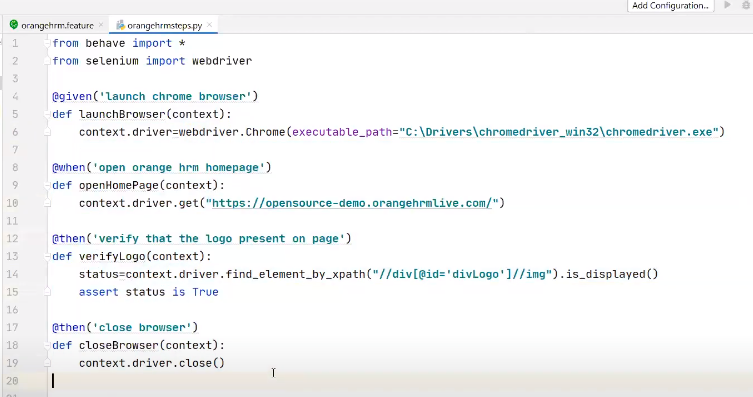
We can copy sample code and edit it as per our need.

Now in steps file we need to import Gherkins keywords.

For that we can specify specific keyword or we can just simply type \*



Now modifying the steps.py file as per our testcase



After executing feature file the complete steps will get executed and following result we will get.

