**Table of Contents**

[**Introduction 1**](#_gjdgxs)

[Lab Overview 1](#_1fob9te)

[Lab Objectives 1](#_2et92p0)

[Lab Prerequisites 2](#_tyjcwt)

[**Instructions 2**](#_3dy6vkm)

[Lab Features / Functions 2](#_1t3h5sf)

[Lab Tutorial 2](#_4d34og8)

[Lab Challenge 5](#_2s8eyo1)

[**Expected Results 5**](#_17dp8vu)

[Lab Examples 5](#_3rdcrjn)

[Lab Takeaway 6](#_26in1rg)

[**Additional Resources 6**](#_lnxbz9)

# Introduction

Behave and BDD in Python

## Lab Overview

Behavior-driven development (BDD) is an essential aspect of meeting business requirements. It ensures that the behaviors of a system are defined clearly before being built. In this lab, you will explore Behave, a popular tool for automating browser interactions. You will work with the Python framework to learn how BDD facilitates collaboration among developers, quality assurance teams, and business stakeholders. Behave is a way to test code using the Gherkin syntax, a plain-text, non-technical, open-source programming language designed to be easily learnable. Non-programmers can fluently understand Gherkin, making it more accessible. The framework promotes clear communication of requirements, making it a powerful tool for creating maintainable, easy-to-understand tests in development environments.

## Lab Objectives

* Gain proficiency in using Behave for web automation and testing.
* Learn to write in the Gherkin language to communicate.
* Understand the basic features and steps involved in working with the Behave framework.
* Develop different behavioral features and scenarios to run tests on.
* Solidify your understanding of the BDD framework using Python.

## Lab Prerequisites

* Basic knowledge of Python programming.
* Understanding of how to install Python libraries using pip.
* Installation of the Behave library (pip install behave).
* Familiarity with the basics of file management.

# Instructions

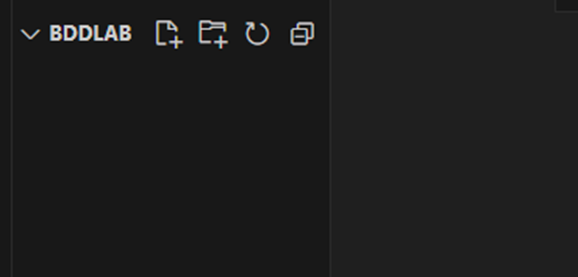
## Lab Features / Functions

This lab is here to provide you with hands-on experience using both Behave and the Gherkin Syntax by BDD's best practices. You will be able to understand the basics of BDD and Behave while expanding your knowledge of the programming language Python. By the end of this lab, you will have set up a beginner-friendly Behave project using Gherkin syntax. At the end, you will execute a simple BDD test.

## Lab Tutorial

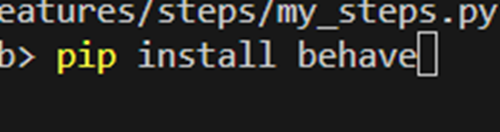
**Assignment Setup:**

1. Set up a virtual environment (recommended) for your testing environment.
2. Open your code editor (examples are Visual Code Studio) and make a new project directory.
3. Name the directory “BDDLAB” for lab purposes.
4. Install Gherkin (pip install gherkin-official).



**Step 1: Install the Behave Framework**

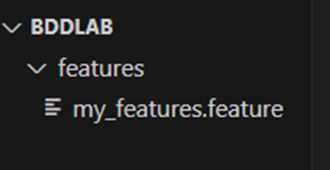
Install Behave using pip:



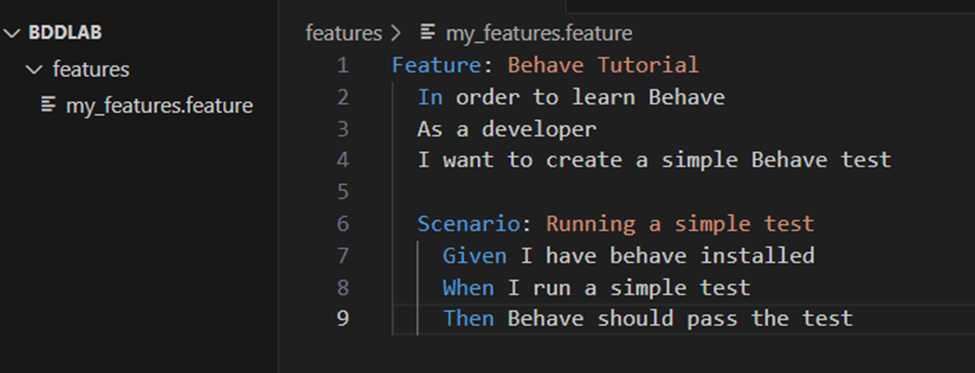
**Step 2: Create Feature File**

Once inside the project directory, create a folder/directory called 'features.'

Now, from the features folder/directory, create a file named my\_features.feature



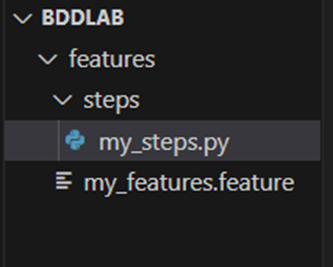
Fill the inside of my\_features.feature with the following code:



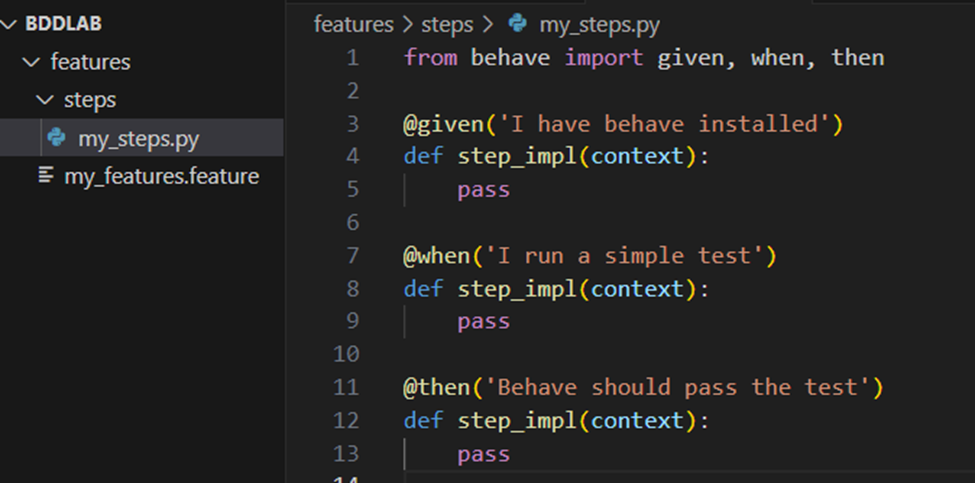
**Step 3: Create Step Definitions**

Once inside the features directory, create a folder/directory called 'steps.'

Now, from inside the steps directory, create a Python file named my\_steps.py



Fill the inside of my\_steps.py with the following code:



**Step 4: Run Behave**

Ensure your virtual environment is still activated.

Open the terminal in Visual Studio Code (or the code editor of your choosing).

Run Behave using the following command in the terminal:



**Step 5: Verify Results**

Observe the output in the terminal. You should see that Behave discovers and executes your scenario successfully. You can also try this with the calculator\_steps.py file by placing it into the steps folder.

## 

## Lab Challenge

To solidify your understanding of Behave and Gherkin. Imagine you are developing a basic web page and need to clearly define the behaviors/features required for entering login credentials and button presses. You can use calculator\_steps.py and calculator\_features.feature to reimagine various scenarios, such as if the user enters valid credentials, forgets their password, or enters invalid credentials. Follow the outlined approach (Given, When, Then, And). Try altering the calculator example included in this lab to describe the basic behaviors of any object, application, or program.

# Expected Results

## Lab Examples

BDD can be used to thoroughly describe product behavior from the user's perspective. The process is about transforming acceptance criteria into automatable behaviors. BDD can be used alongside continuous integration and continuous delivery environments.

## Lab Takeaway

Congrats! Once you've finished the lab, you should understand the basics of Behave and the Gherkin language. You should now be able to apply your understanding of BDD to other programming languages.

# Additional Resources

Gherkin:

<https://cucumber.io/docs/installation/python/>

<https://cucumber.io/docs/guides/overview/>

<https://behave.readthedocs.io/en/stable/gherkin.html>

Behave:

<https://behave.readthedocs.io/en/stable/>

Python:

<https://docs.python.org/3/>