**Create Wheel file**

**What is wheel package?**

A built-package format for Python. A wheel is a ZIP-format archive with a specially formatted filename and the. whl extension. We can create a package containing more than one modules and share the package to fellow programmers. Instead of installing the components one by one, it can be installed together as a single package, and components can be imported individually.

Wheels make the end-to-end installation of Python packages faster for following reasons:

1.wheels are typically smaller in size than source distributions, meaning they can move faster across a network.

2.Installing from wheels directly avoids the intermediate step of building packages off of the source distribution.

3. There's no need for a compiler to install wheels that contain compiled extension modules.

A wheel filename is broken down into parts separated by hyphens:

**{dist}-{version}(-{build})?-{python}-{abi}-{platform}.whl**

Abi stands for application binary interface

Given is a wheel package name created by me. Compare these .

**MyPackage-0.0.1-py3-none-any.whl**

**STEPS OF CREATING A WHEEL PACKAGE**

**1.Create Root folder**

First, we have to create a root folder, in which we will save a Folder and a python file called setup.py

The Folder is named as the package name we desire. I used My Package as the folder name.

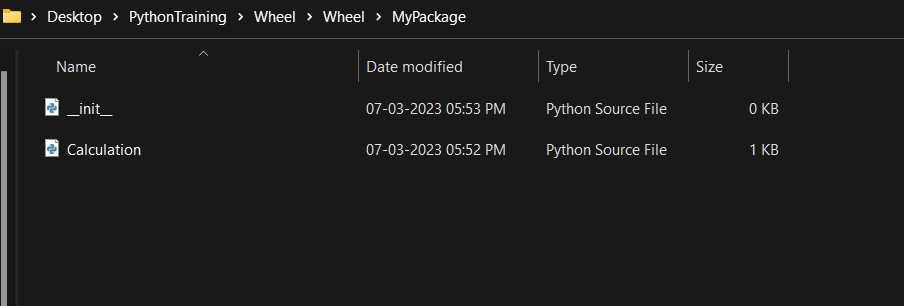
Inside this folder, we place the module(s) (Module = file with .py extension) that we need to bundle as a package.

**2.Create init file**

An init file is also created inside this folder with name \_\_init\_\_.py .This file can be empty.

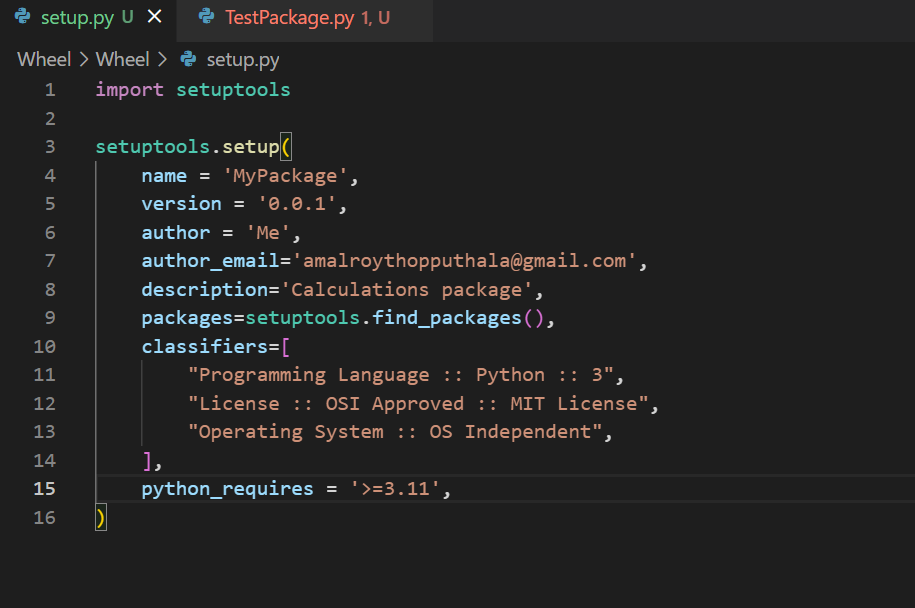
The \_\_init\_\_.py files are required **to make Python treat directories containing the file as packages**. This prevents directories with a common name, such as string, unintentionally hiding valid modules that occur later on the module search path.

If we have multiple folders containing modules, we have ensure that there are init files in all folders

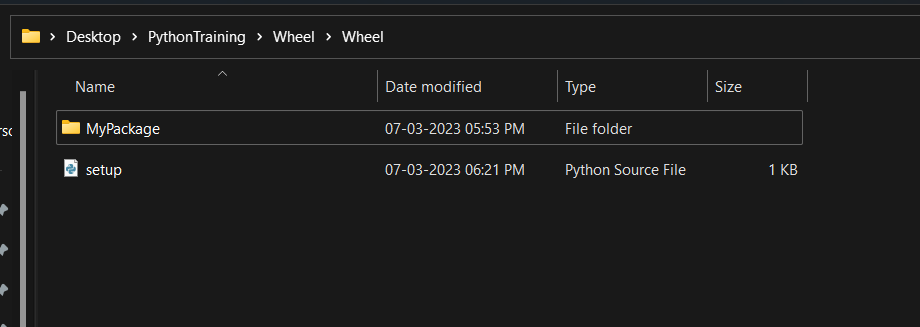


**3.Create setup file**

The setup.py file is the file that we call to create a wheel file.In this file,we save the metadata of the package that we are going to create. Metadata is information like author,author\_mail,python version required. Etc.



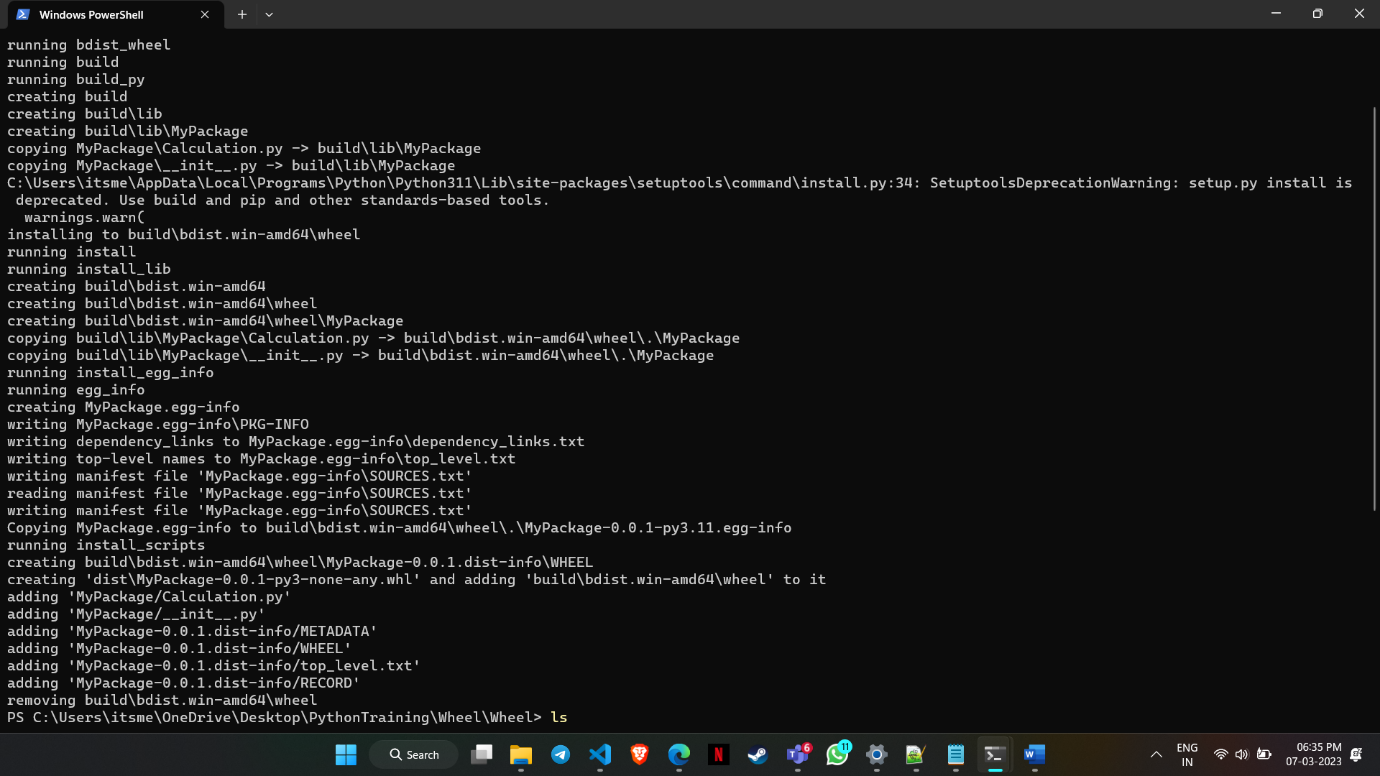
The contents of the root folder now looks like this:



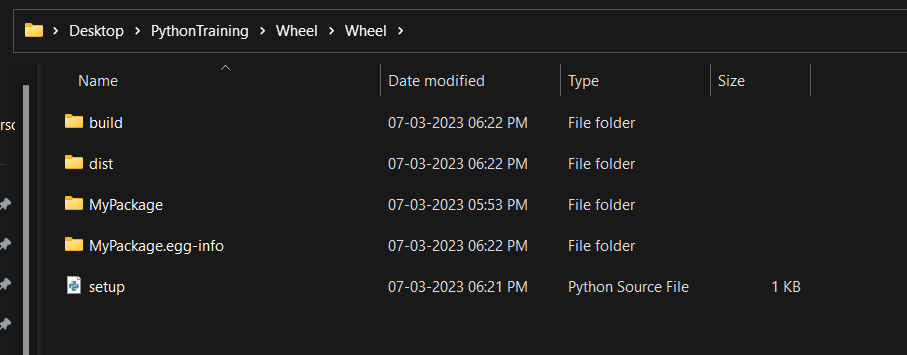
**4.Create wheel package**

Now, to create wheel package, we have to run the command python .\setup.py bdist\_wheel from terminal

The output will look like this:

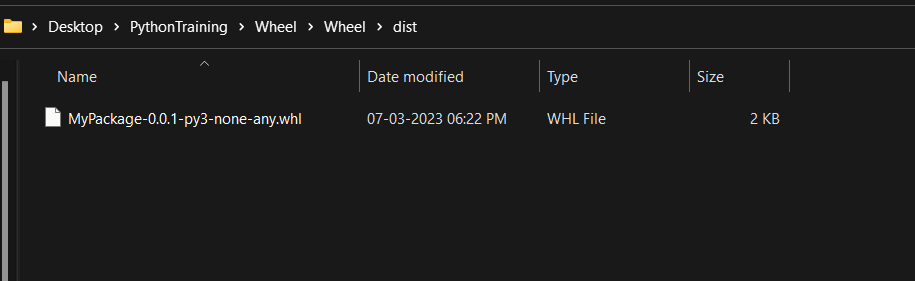


After running this command, we can see that new folders are created in our root folder.



**5.Install wheel package**

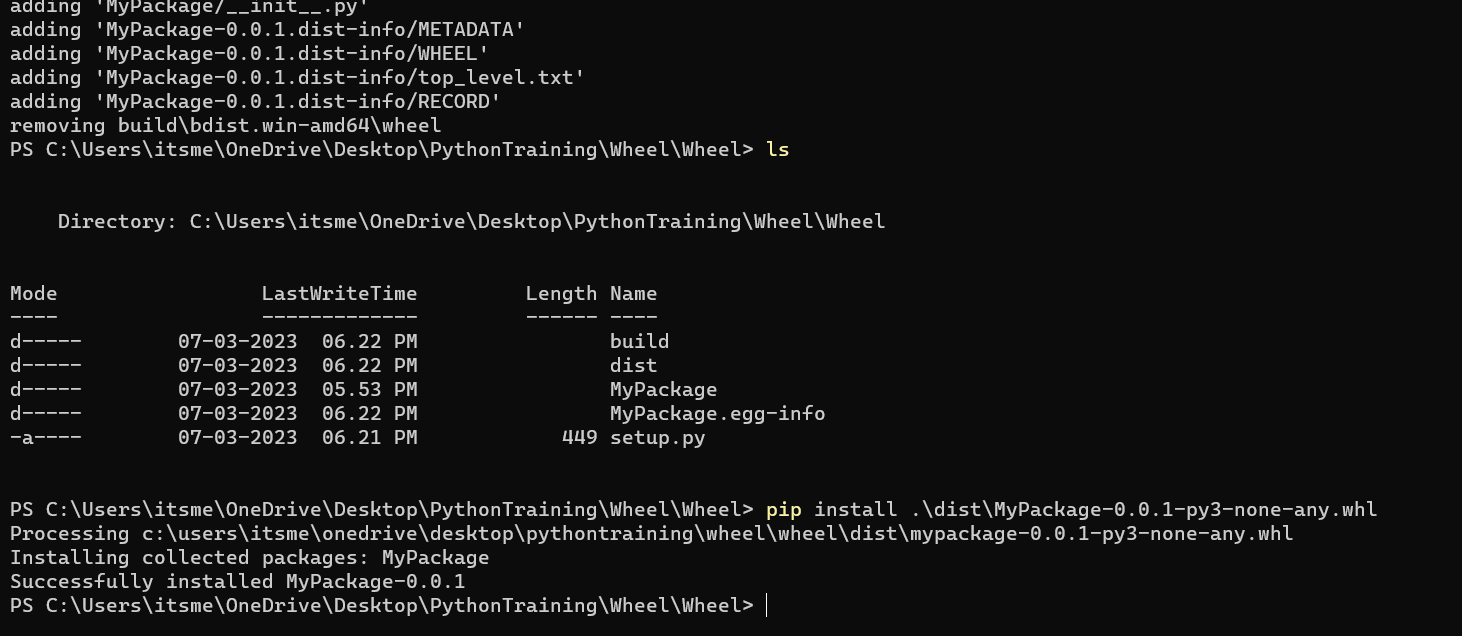
The .whl file is created inside the dist folder. Now this .whl file can be installed and the custom package will be added to the packages available in our machine



To install our wheel file, open the root folder in terminal and install the package using pip.

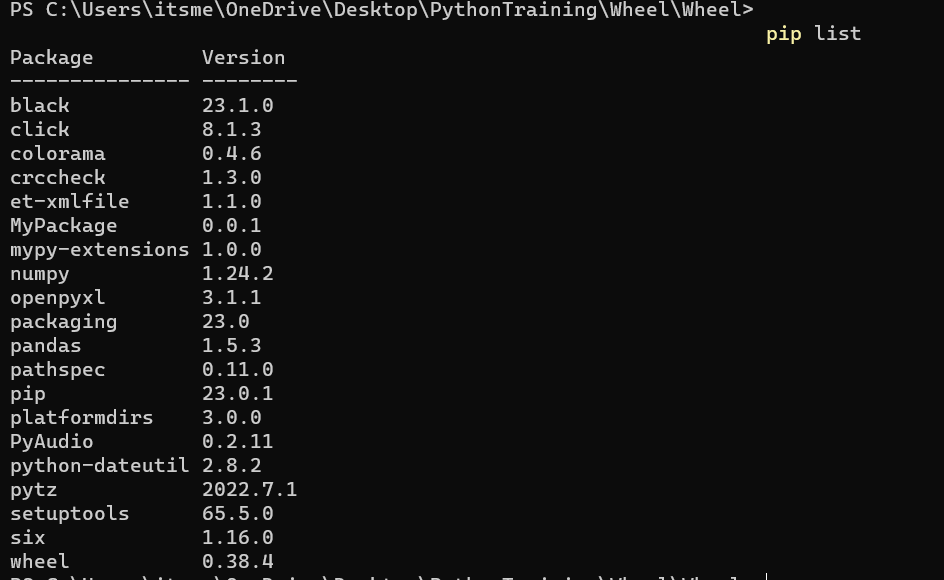
The command is pip install .\dist\<PackageNameAsInDistFolder>.whl

The name of the folder containing the. whl, i.e., dist, is specified in the command.



**6.Test if installation is successfull**

Now, to test if the package is installed, we can use the pip list command and check if the package name is in the list displayed



We can see that the package is now installed (Here,6th item in list)

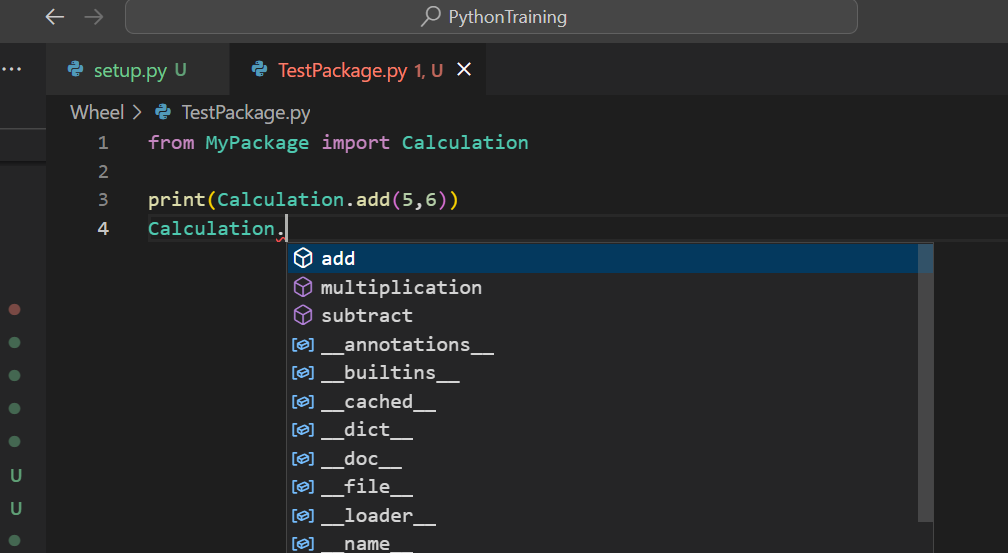
**7.Check operation of wheel package**

Now, to check the operation of our package, create a python source file and import the module from package.

Type: from <Package Name> import <Module Name>

If the package and module name are auto detected, it means the installation was success.

We can call the functions that we defined in our module as below



We can see that the functions we created are detected. This means that the package was installed successfully.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Reference**: [55. Package your Python code as Wheel File - YouTube](https://www.youtube.com/watch?v=TMTHNCJo-Tc&ab_channel=WafaStudies)