

How to Create and Publish Python Library

Introduction

In this guide, we'll walk through the process of creating and publishing your first Python library. Building a Python library allows you to share your code with others, making it easier for them to use and contribute to your projects

When to create a library?

When you have just finished a long project, full of pitfalls, where you felt sometimes helpless to solve certain situations; You had to search deep among the least upvoted posts of StackOverflow, and, as a result, you have produced many chunk of code that solve concrete problems.

At this point, you have the possibility to create a library: You will only have to copy some code from your project. Easy!

Why create a library?

First, because the code you have written may, in the future, save other developers from going through the same long and expensive research process you did.

Also, if your code provides a high level interface to solve some problems, you will help democratize the resolution of these problems to less experienced programmers. Therefore, your high-level library could, for example, become an entry point for others to easily get a quick result.

Get started: Maximize compatibility:

First, when creating your library's project in your favorite IDE, use an older version of Python, so that your library can be compatible with as many current workflows as possible.

For example: Python 3.9 released 2 months ago. If you decide to start with this new Python and use the new features of this version, only developers who have applications already running Python 3.9 will be able to use your library... So, very few people!

So, go back a few versions and take Python 3.11.

Table of Contents

- [Setting Up Your Development Environment](#)
- [Creating Your Python Library](#)
- [Write Your First Library](#)
- [Publishing Your Library](#)
- [Build your library](#)

Setting Up Your Development Environment

Before we begin, make sure you have Python and pip installed on your system. You'll also need a text editor or an integrated development environment (IDE) for writing code.

Installing Necessary Tools

```
# Install setuptools and wheel
pip install setuptools wheel
```

Creating Your Python Library

Choose a name for your library and create a new directory with that name.

To create a Python library with Python 3.11 for splitting PDF and DOCX files, you can follow these steps:

Organize your project structure:

```
MyFirstLibrary/  
  
MyFirstLibrary/  
    __init__.py  
    MyFirstLibrary.py  
  
setup.py  
README.md
```

Write Your First Library

Write your Python code in ***MyFirstLibrary.py***. This code will form the basis of your library.

MyFirstLibrary.py

```
# hello world.py  
  
def helloworld(name):  
  
    print(f"Hello World, {name}!")
```

Keep the ***__init__.py*** empty unless you have multiple functions in the ***MyFirstLibrary.py*** file.

For example, if there are *helloworld* and *helloIndia* functions are written then your ***__init__.py*** file contains below code

__init__.py

```
from MyFirstLibrary import helloworld, helloIndia
```

Publishing Your Library

Create a ***setup.py*** file in your library directory to provide metadata about your library.:

setup.py

```
from setuptools import setup, find_packages  
  
setup(  
    name="MyFirstLibrary",  
    version="1.0",  
    description='A library for printing the hello world',  
    author='Your Name',  
    author_email='your.email@example.com',  
    packages=find_packages(),  
    install_requires=[  
        # Add any dependency  
    ],  
    python_requires='>=3.11',  
)
```

Build your library

Build your library distribution package using the below command

bash

```
python setup.py sdist bdist_wheel
```

Congratulations! You've successfully created your first Python library !!