kivy

DB Generator Frontend Example

Setup/Creating A Simple Application

What is Kivy?

Kivy is a python module that allows for the creation of cross compatible applications using python.

It makes it very easy to reuse the same code on IOS, Andorid, Mac, Windows, Linux and virtually all other well known operating systems.

Creating apps with kivy is great as your code works on every kind of device.

Kivy is similar to TKinter in the way you develop apps. It allows you to create a GUI using widgets and layouts.

Installing Kivy

For detailed instructions on all operating systems please visit the kivy website.

Before we can start using kivy we must download and install it. The easiest way to do this is to use **pip**.

To test if you have pip in your system path open up **cmd and type pip**.

```
$ pip
```

If you do not receive any errors you can continue with the instructions.

If this did NOT work then you need to add pip to your system path.

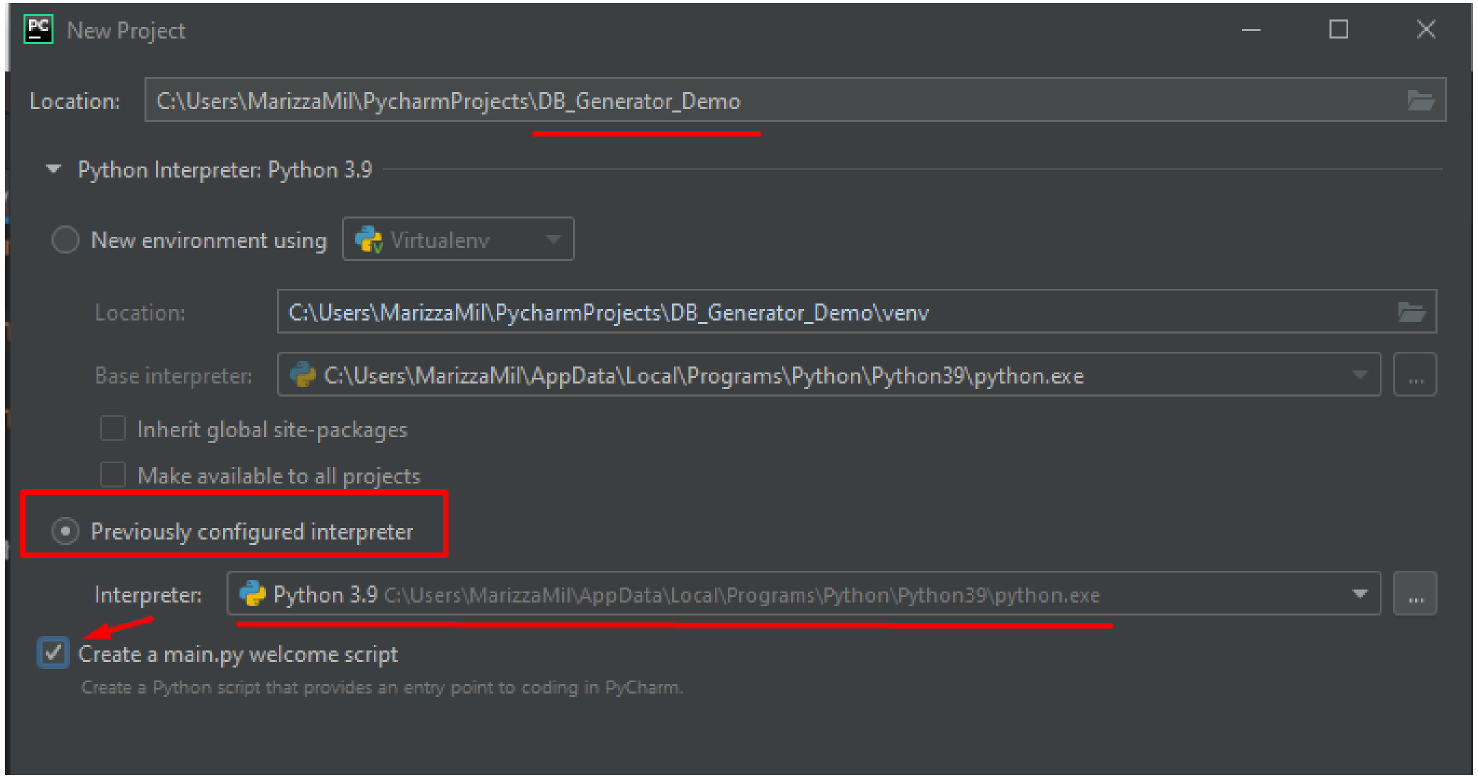
If pip is working then you need to type the following commands into your command prompt.

```
python -m pip install --upgrade pip wheel setuptools
python -m pip install docutils pygments pypiwin32 kivy.deps.sdl2 kivy.deps.glew
python -m pip install kivy.deps.gstreamer
python -m pip install kivy.deps.angle
python -m pip install pygame
python -m pip install kivy
```

If the execution of all of those commands is successful you are ready to move on.

Creating our First App

Once we have Kivy installed and setup we can get to creating our first app.



New Project

Location: C:\Users\MarizzaMil\PycharmProjects\DB_Generator_Demo

▾ Python Interpreter: Python 3.9

○ New environment using Virtualenv

Location: C:\Users\MarizzaMil\PycharmProjects\DB_Generator_Demo\venv

Base interpreter: C:\Users\MarizzaMil\AppData\Local\Programs\Python\Python39\python.exe

□ Inherit global site-packages

□ Make available to all projects

● Previously configured interpreter

Interpreter: Python 3.9 C:\Users\MarizzaMil\AppData\Local\Programs\Python\Python39\python.exe

☑ Create a main.py welcome script

Create a Python script that provides an entry point to coding in PyCharm.

Creating our First App

The first thing we need to do is import the necessary modules.

```
import kivy
from kivy.app import App
from kivy.uix.label import Label
```

The best way to create apps using Kivy is to do so using OOP. Therefore, we need to create a class to represent our window.

This class will inherit from the App class that we imported above. This means it will take all functionally from App.

In doing so all we have to do to make our class functional is implement a method **build()** which will tell Kivy what to place on the screen.

```
class MyApp(App):
def build(self):
return Label(text="Hello world")
```

We simply return a label that says "Hello world".

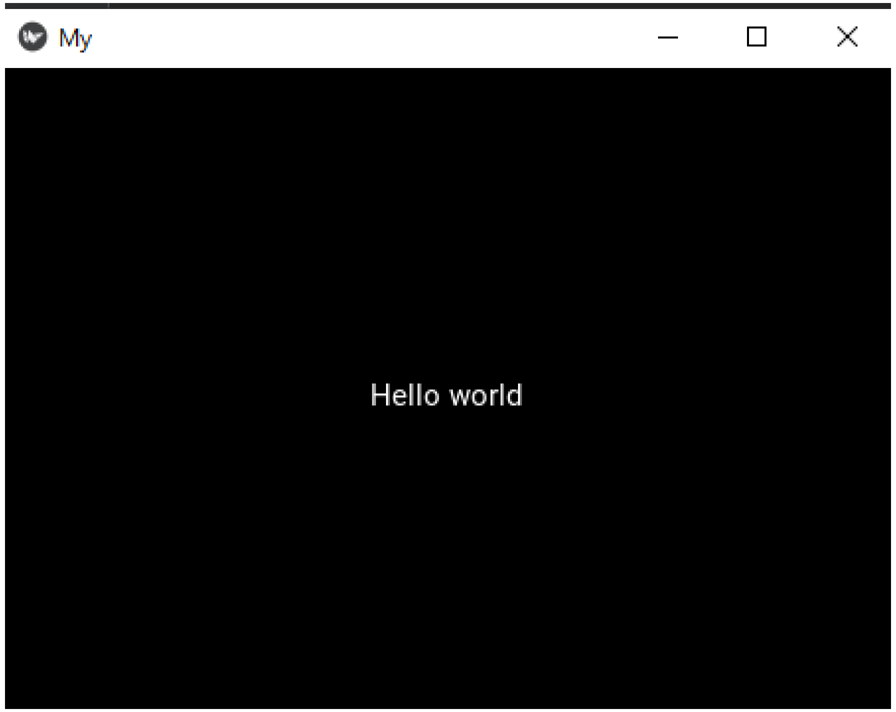
Running our App

Believe it or not we have just created our first app.

To run it we need to add the following code to the end of our program.

```
if __name__ == "__main__":
MyApp().run()
```

Now if we run the program we should see the following.



```
Hello world
```

Full Code

main.py

```
import kivy
from kivy.app import App
from kivy.uix.label import Label


class MyApp(App):
def build(self):
return Label(text="Hello world")


if __name__ == "__main__":
MyApp().run()
```