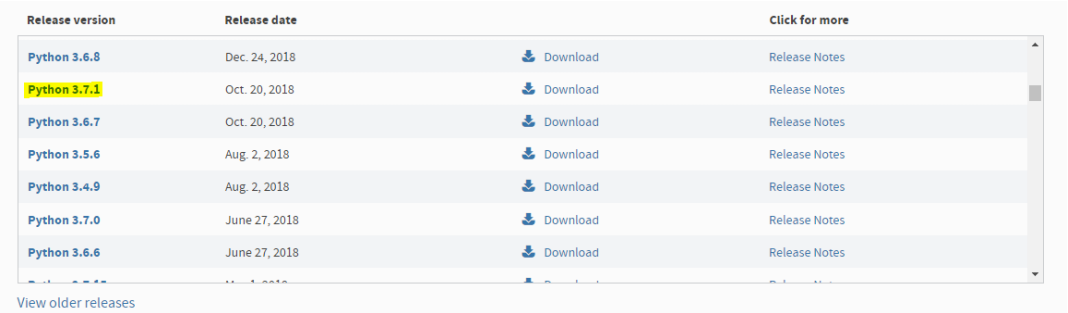
App Development with Kivy

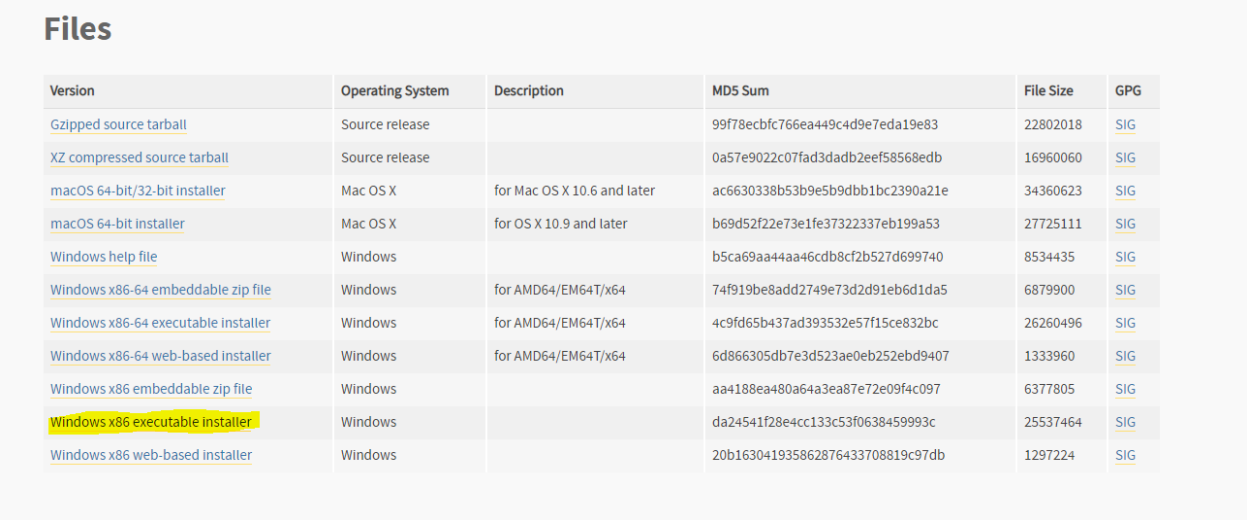
Kivy is an open source Python library used for mobile app development. Kivy is cross-platform compatible kit; it works on Windows, MacOS, Linux. We will cover some key concepts that make Kivy a very useful Python library.

1. To use Kivy, we must first have some base software installed. Go to https://www.python.org/downloads/ and download Python 3.7.1. (NOTE: Kivy might not work on any version later than this.)



Release version	Release date		Click for more
Python 3.6.8	Dec. 24, 2018	Download	Release Notes
Python 3.7.1	Oct. 20, 2018	Download	Release Notes
Python 3.6.7	Oct. 20, 2018	Download	Release Notes
Python 3.5.6	Aug. 2, 2018	Download	Release Notes
Python 3.4.9	Aug. 2, 2018	Download	Release Notes
Python 3.7.0	June 27, 2018	Download	Release Notes
Python 3.6.6	June 27, 2018	Download	Release Notes

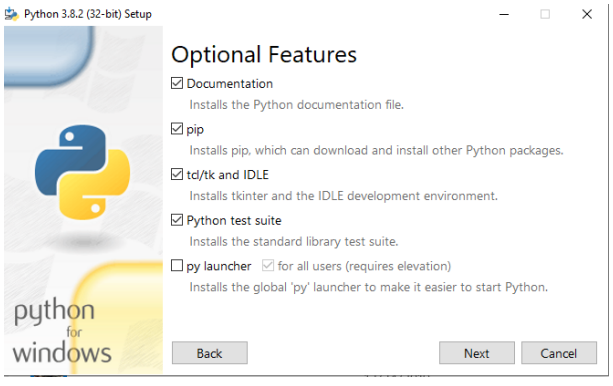
View older releases



Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		99f78ecbfc766ea449c4d9e7eda19e83	22802018	SIG
XZ compressed source tarball	Source release		0a57e9022c07fad3dadb2eef58568edb	16960060	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	ac6630338b53b9e5b9dbb1bc2390a21e	34360623	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	b69d52f22e73e1fe37322337eb199a53	27725111	SIG
Windows help file	Windows		b5ca69aa44aa46cdb8cf2b527d699740	8534435	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	74f919be8add2749e73d2d91eb6d1da5	6879900	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	4c9fd65b437ad393532e57f15ce832bc	26260496	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	6d866305db7e3d523ae0eb252ebd9407	1333960	SIG
Windows x86 embeddable zip file	Windows		aa4188ea480a64a3ea87e72e09f4c097	6377805	SIG
Windows x86 executable installer	Windows		da24541f28e4cc133c53f0638459993c	25537464	SIG
Windows x86 web-based installer	Windows		20b16304193586287643370881 9c97db	1297224	SIG

2. In the installation window, it *might* ask you to select a few options. Make sure to check the box next to pip, since we will need this to install Kivy.



3. Open your command line and run the following command to update pip: py -m pip install --upgrade pip wheel setuptools.

```
Collecting pip
  Downloading pip-20.1.1-py2.py3-none-any.whl (1.5 MB)
     |████████████████████████████████| 1.5 MB 1.1 MB/s
Requirement already up-to-date: wheel in c:\users\marlon\anaconda3\lib\site-packages (0.34.2)
Collecting setuptools
  Downloading setuptools-47.1.1-py3-none-any.whl (583 kB)
     |████████████████████████████████| 583 kB 3.3 MB/s
Installing collected packages: pip, setuptools
  Attempting uninstall: pip
    Found existing installation: pip 20.0.2
    Uninstalling pip-20.0.2:
      Successfully uninstalled pip-20.0.2
  Attempting uninstall: setuptools
    Found existing installation: setuptools 45.2.0.post20200210
    Uninstalling setuptools-45.2.0.post20200210:
      Successfully uninstalled setuptools-45.2.0.post20200210
Successfully installed pip-20.1.1 setuptools-47.1.1
```

4. Next run the following command: py -m pip install docutils pygments pypiwin32 kivy.deps.sdl2 kivy.deps.glew

```
Requirement already satisfied: docutils in c:\users\marlon\anaconda3\lib\site-packages (0.16)
Requirement already satisfied: pygments in c:\users\marlon\anaconda3\lib\site-packages (2.5.2)
Collecting pypiwin32
  Downloading pypiwin32-223-py3-none-any.whl (1.7 kB)
Collecting kivy.deps.sdl2
  Downloading kivy_deps.sdl2-0.2.0-cp37-cp37m-win_amd64.whl (2.5 MB)
     |████████████████████████████████| 2.5 MB 1.3 MB/s
Collecting kivy.deps.glew
  Downloading kivy_deps.glew-0.2.0-cp37-cp37m-win_amd64.whl (123 kB)
     |████████████████████████████████| 123 kB 6.8 MB/s
Requirement already satisfied: pywin32>=223 in c:\users\marlon\anaconda3\lib\site-packages (from pypiwin32) (227)
Installing collected packages: pypiwin32, kivy.deps.sdl2, kivy.deps.glew
Successfully installed kivy.deps.glew kivy.deps.sdl2 pypiwin32-223
```

5. Run the following commands (one at a time) to install some Kivy dependencies:
    - py -m pip install kivy.deps.gstreamer
    - py -m pip install kivy.deps.angle
    - py -m pip install pygame

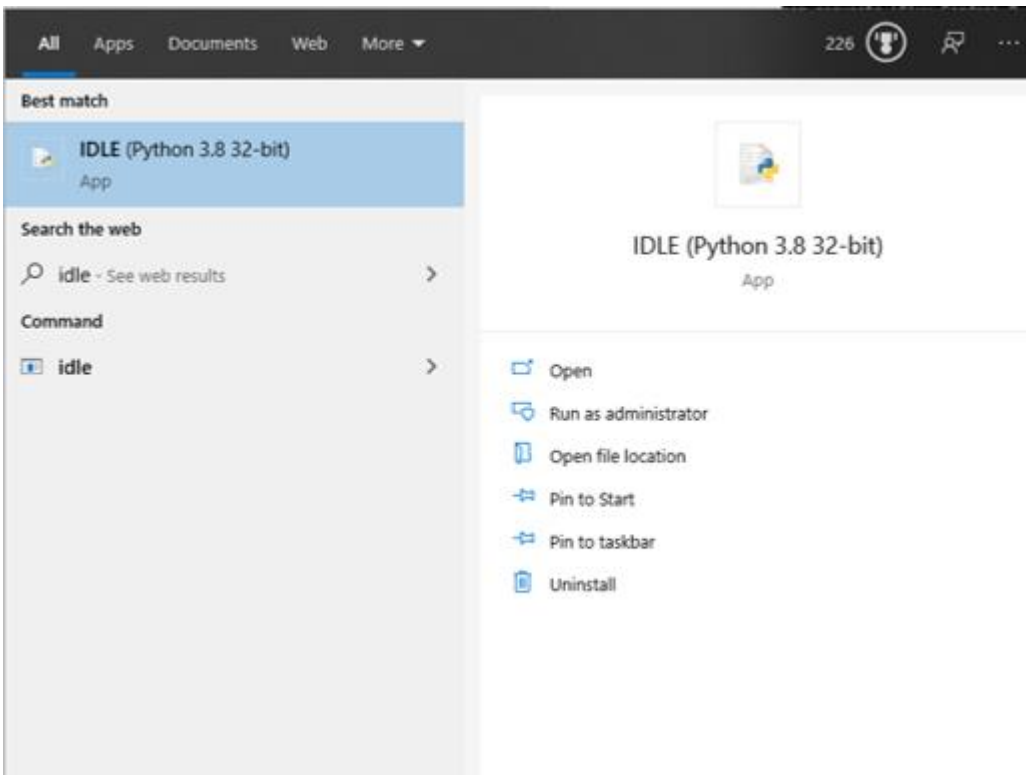
```
Collecting pypiwin32
  Downloading pypiwin32-223-py3-none-any.whl (1.7 kB)
Collecting kivy.deps.sdl2
  Downloading kivy_deps.sdl2-0.2.0-cp37-cp37m-win_amd64.whl (2.5 MB)
     |████████████████████████████████| 2.5 MB 1.3 MB/s
Collecting kivy.deps.glew
  Downloading kivy_deps.glew-0.2.0-cp37-cp37m-win_amd64.whl (123 kB)
     |████████████████████████████████| 123 kB 6.8 MB/s
Requirement already satisfied: pywin32>=223 in c:\users\marlon\anaconda3\lib\site-packages (from pypiwin32) (227)
Installing collected packages: pypiwin32, kivy.deps.sdl2, kivy.deps.glew
Successfully installed kivy.deps.glew kivy.deps.sdl2 pypiwin32-223

C:\Users\Marlon>•python -m pip install kivy.deps.gstreamer
'•python' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Marlon>python -m pip install kivy.deps.gstreamer
Collecting kivy.deps.gstreamer
  Downloading kivy_deps.gstreamer-0.2.0-cp37-cp37m-win_amd64.whl (77.6 MB)
     |████████████████████████████████| 77.6 MB 3.2 MB/s
Installing collected packages: kivy.deps.gstreamer
Successfully installed kivy.deps.gstreamer

C:\Users\Marlon>python -m pip install pygame
Collecting pygame
  Downloading pygame-1.9.6-cp37-cp37m-win_amd64.whl (4.3 MB)
     |████████████████████████████████| 4.3 MB 1.3 MB/s
Installing collected packages: pygame
Successfully installed pygame-1.9.6
```

6. Finally, we can install Kivy: py -m pip install kivy

```
Collecting kivy
  Downloading Kivy-1.11.1-cp37-cp37m-win_amd64.whl (4.1 MB)
     |████████████████████████████████| 4.1 MB 1.3 MB/s
Requirement already satisfied: pygments in c:\users\marlon\anaconda3\lib\site-packages (from kivy) (2.5.2)
Collecting Kivy-Garden>=0.1.4
  Downloading kivy-garden-0.1.4.tar.gz (6.8 kB)
Requirement already satisfied: docutils in c:\users\marlon\anaconda3\lib\site-packages (from kivy) (0.16)
Requirement already satisfied: requests in c:\users\marlon\anaconda3\lib\site-packages (from Kivy-Garden>=0.1.4->kivy) (
2.22.0)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\marlon\anaconda3\lib\site-packages (from requests->Kivy-Ga
rden>=0.1.4->kivy) (2019.11.28)
Requirement already satisfied: idna<2.9,>=2.5 in c:\users\marlon\anaconda3\lib\site-packages (from requests->Kivy-Garden
>=0.1.4->kivy) (2.8)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in c:\users\marlon\anaconda3\lib\site-packages (f
rom requests->Kivy-Garden>=0.1.4->kivy) (1.25.8)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in c:\users\marlon\anaconda3\lib\site-packages (from requests->Kivy
-Garden>=0.1.4->kivy) (3.0.4)
Building wheels for collected packages: Kivy-Garden
  Building wheel for Kivy-Garden (setup.py) ... done
  Created wheel for Kivy-Garden: filename=Kivy_Garden-0.1.4-py3-none-any.whl size=4534 sha256=0c84d35a730f919de10a2dc724
f19a8ef5390808a3e1dc6b308ced7bc4091a3b
  Stored in directory: c:\users\marlon\appdata\local\pip\cache\wheels\3f\43\e3\50289d555356f0421d1c388c82d052d5788f22a34
d0cd8659d
Successfully built Kivy-Garden
Installing collected packages: Kivy-Garden, kivy
Successfully installed Kivy-Garden-0.1.4 kivy-1.11.1
```

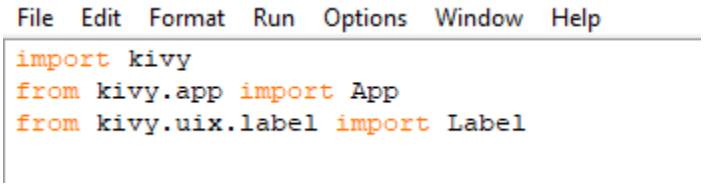
7. Type the following commands (one at a time) to ensure that everything was set up correctly:
   - py
   - Import kivy

8. If the above commands executed with no errors, we are ready to start using Kivy. Open IDLE, which is the Python environment that comes with the Python interpreter you just downloaded. You can open it by searching it in the windows search bar.



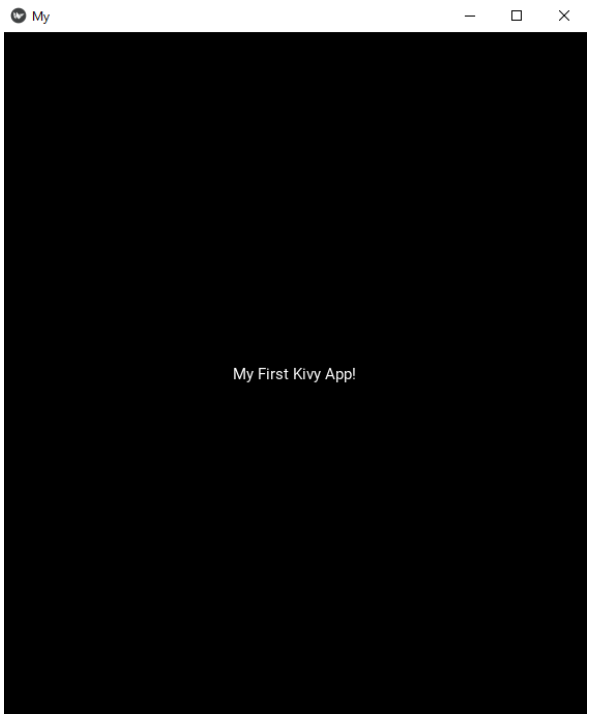
9. In the IDLE environment, click File > New File, and insert the following code. We are simply importing Kivy and some of its modules to use in our app:

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



10. Add the following code. We are creating a class called MyApp which inherits App, a module in the Kivy library. We are also defining a build function that displays a label. Finally, we run the app using MyApp().run(). Notice how Kivy takes care of the formatting of the text, and if you resize the window, the label will always be centered (Which is useful if you're going to use this app on many different platforms.) This is one of the advantages of using Kivy, it takes care of the low-level programming aspect:

```
import kivy

from kivy.app import App

from kivy.uix.label import Label


class MyApp(App):

    def build(self):

        return Label(text="My First Kivy App!")


if __name__ == "__main__":

    MyApp().run()
```



11. Let's create an app that can updates values in our 000webhost database. We will create two switches "LED 1" and "SW 1" that can be turned on and off from the app. Make a new IDLE file and insert the following code.

```
import kivy
from kivy.app import App
from kivy.uix.switch import Switch
from kivy.uix.gridlayout import GridLayout
from kivy.uix.label import Label

class SwitchContainer(GridLayout): #Create a class that uses the GridLayout module
    def __init__(self, **kwargs):
        super(SwitchContainer, self).__init__(**kwargs)
        self.cols = 2

        self.add_widget(Label(text="LED 1: ")) #Create a label that displays "LED 1"
        self.settings = Switch(active=False)

        self.add_widget(self.settings) #Create a switch that can be turned off or on
        self.settings.bind(active=switch_callback1)

        self.add_widget(Label(text="SW 1: "))
        self.settings = Switch(active=False)

        self.add_widget(self.settings)
        self.settings.bind(active=switch_callback2)



def switch_callback1(switchObject, switchValue): #outputs the status of the switch to the console
    print('Value of LED 1:', switchValue)

def switch_callback2(switchObject, switchValue):
    print('Value of SW 1: ', switchValue)

class SwitchExample(App):
```

```
    def build(self):
        return SwitchContainer()


if __name__ == '__main__':
    SwitchExample().run()
```