CSCI 3342/6312 – Tutorial/Assignment #6 **Deadline: 3:00 pm on June 20, 2020**

– Quiz #6 (Separate *Container*) **Deadline: 3:00 pm on June 22, 2020**

Completion: Required **Submission: Required**

Last Name:_________________________ First Name:_________________________

This *tutorial* is to get you started on Android development using Python-based Kivy (library).

1) How and what to submit?

Submit the following (upload in Blackboard to the available container) in "one" PDF document (not in docx or any other format):

i) The certification page (see next page) should be the first page, followed by
ii) your solution to the problems given on this assignment.

One way is to copy the certification page and your solution to the problems into a Word document and then save the Word document as PDF, and upload the PDF version (not docx version). Only the PDF version will be graded.

2) Only ONE upload attempt is allowed:* *Before submitting a document through Blackboard, you should review the document being uploaded to make sure that you are uploading the correct document (e.g. do not upload the assignment belonging to another course). To help you prevent uploading wrong documents, notes (titled "HelpOnSubmissionThroughBlackboard" on how to save & review drafts before final submission have been uploaded under Reference Material folder.

Certification Page

This page must be the first page of your uploaded document.
Your assignment will not be graded without this page (completed with your full name in the area provided) as the first page of your uploaded document.

I, **___________________________________**, certify that the work I am uploading represents my own efforts, and is not copied from anyone else or any other resource (such as Internet). *Furthermore, I certify that I have not let anyone copy from my work.*

Tutorial Portion

Learning Objectives!

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.

Below is a youtube video on Kivy basics:
Video: https://www.youtube.com/watch?v=bMHK6NDVlCM

In this tutorial we will be learning how we can use Kivy library modules to develop a basic mobile interface consisting of two ***SwitchCompat*** entities to display status of LED1 and SW1.

Below is a summary of the sections in this tutorial:

Sections 1 through 7: Installation of Python and Kivy.

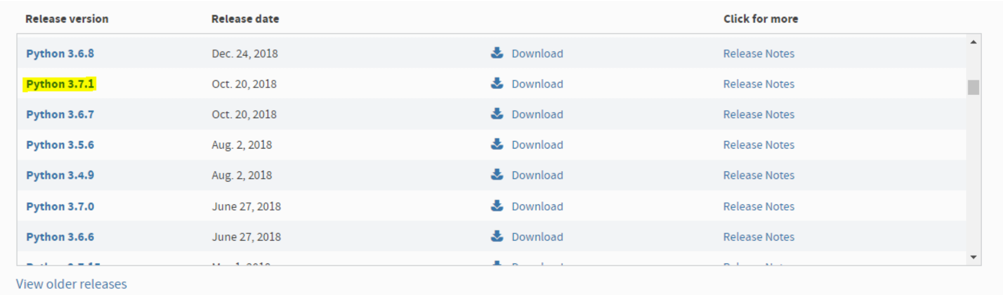
Sections 8, 9, and 10: IDLE environment, and writing/running first App using Kivy.

Section 11: Development of a mobile App using two ***SwitchCompat*** entities.

Section 12: Development of a simple ***Rock-Paper-Scissors*** game.

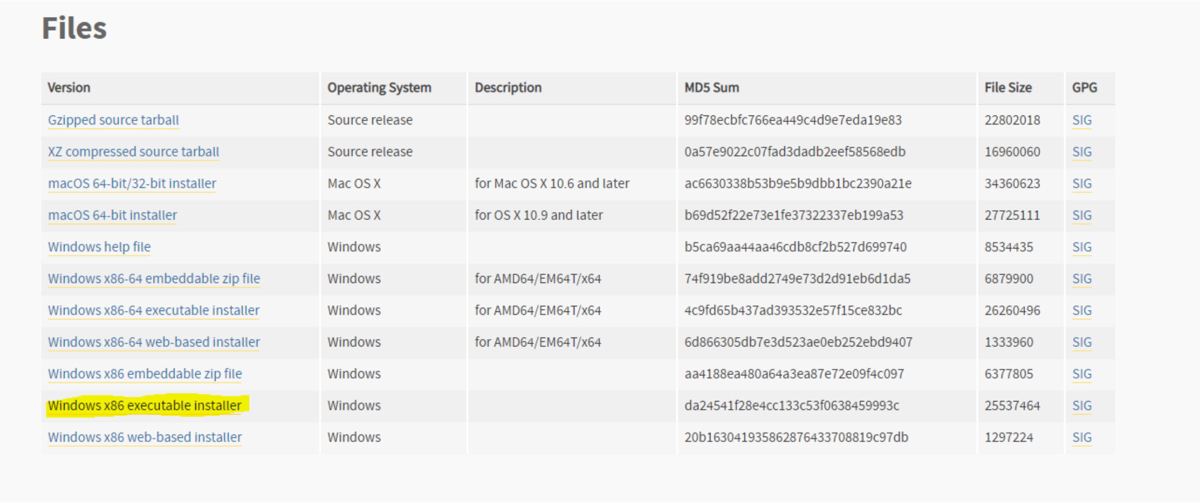
We will be using Windows based machine for this tutorial to get familiar with Kivy.

1. To use Kivy, we must first have some base software installed. Access 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

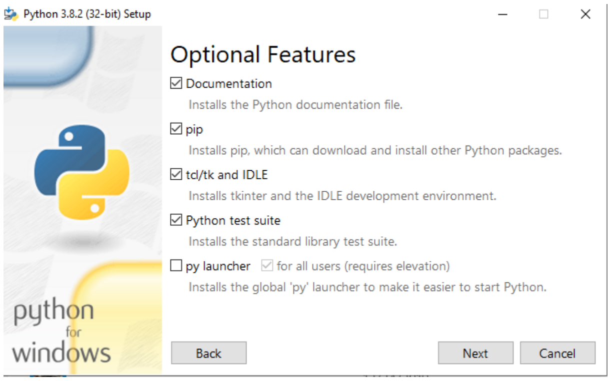
View older releases



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		20b163041935862876433708819c97db	1297224	SIG

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



Python 3.8.2 (32-bit) Setup

Optional Features

☑ Documentation
Installs the Python documentation file.

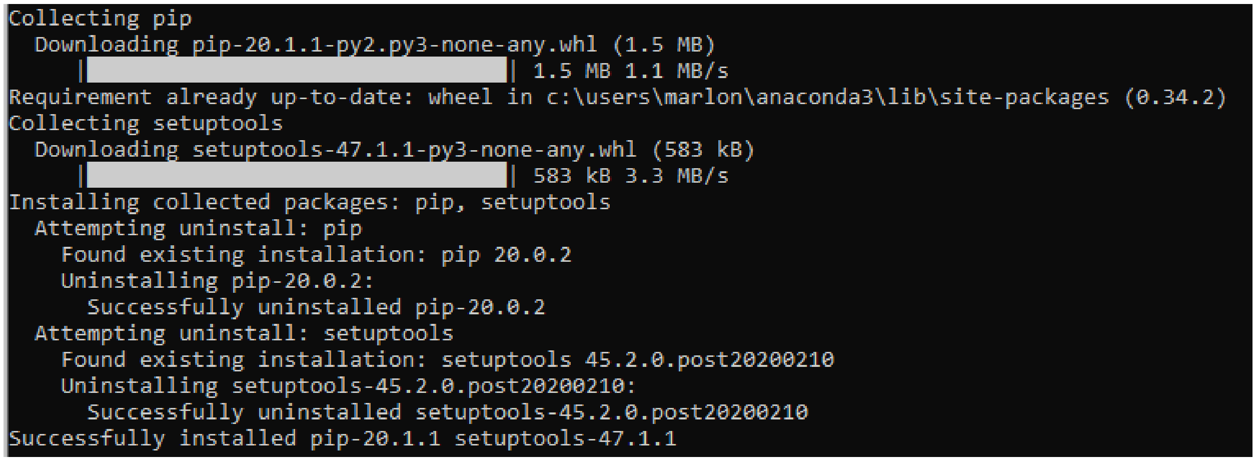
☑ pip
Installs pip, which can download and install other Python packages.

☑ tcl/tk and IDLE
Installs tkinter and the IDLE development environment.

☑ Python test suite
Installs the standard library test suite.

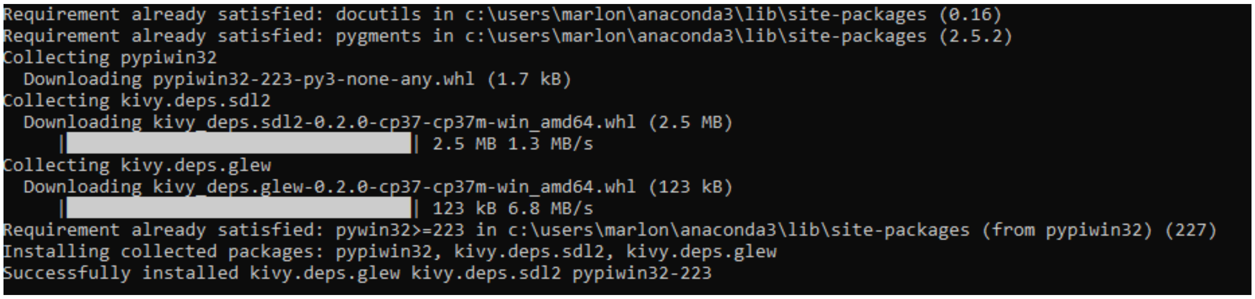
☐ py launcher ☑ for all users (requires elevation)
Installs the global 'py' launcher to make it easier to start Python.

3. Open your command line interface (command prompt / cmd shell) 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
```

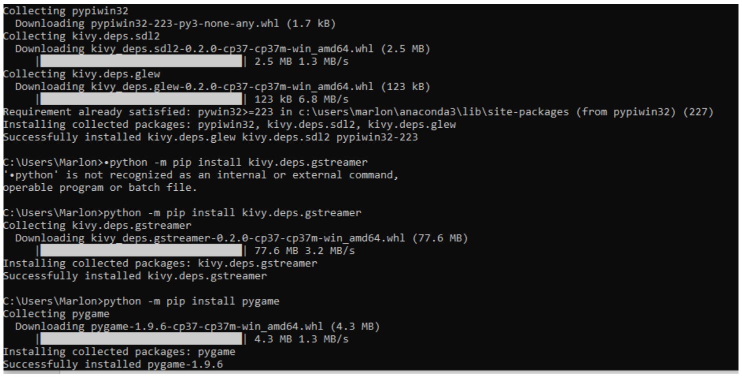
4. Next run the following command: py -m pip install docutils pygments pypiwin32 kivy.deps.sdl2 kivy.deps.glew



```
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
```

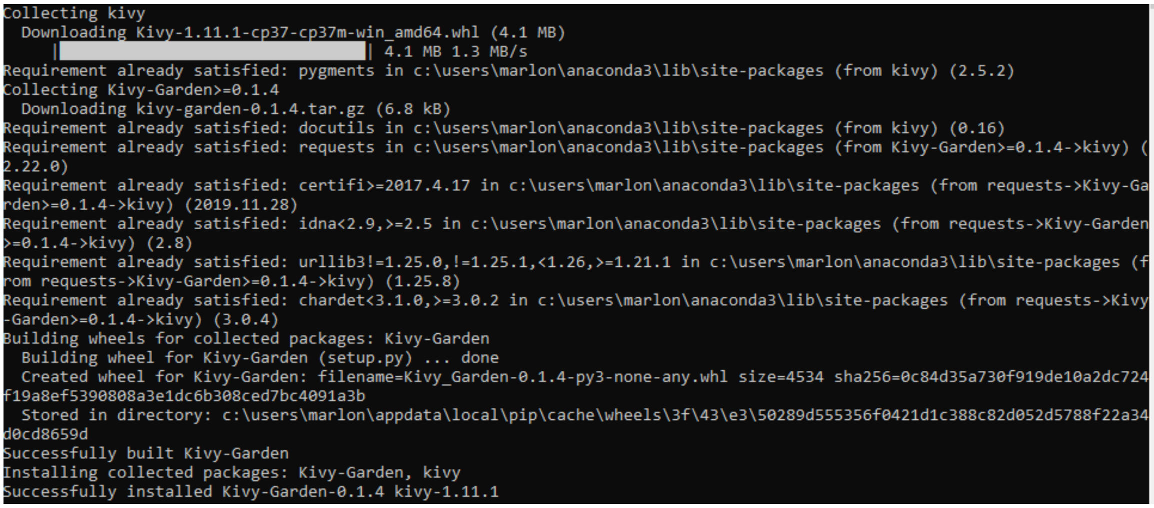
5. Run the following commands (one at a time) to install some Kivy dependencies:

- 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
```

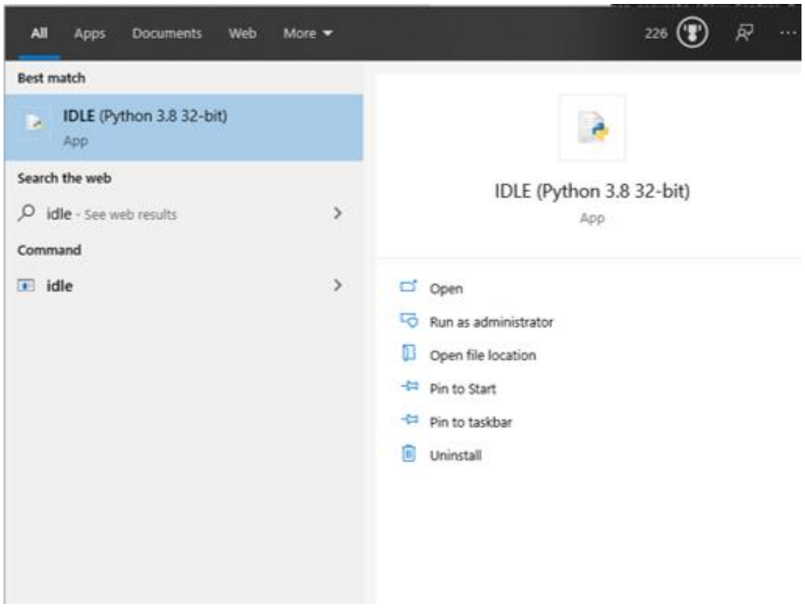
6. Finally, we can install Kivy: py -m pip install kivy



```
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
```

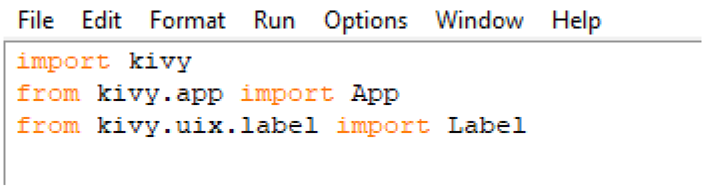
7. Type the following commands (one at a time) to ensure that everything was set up correctly:
   - py
   - Import kivy

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 to import Kivy and some of its modules to use in our app:

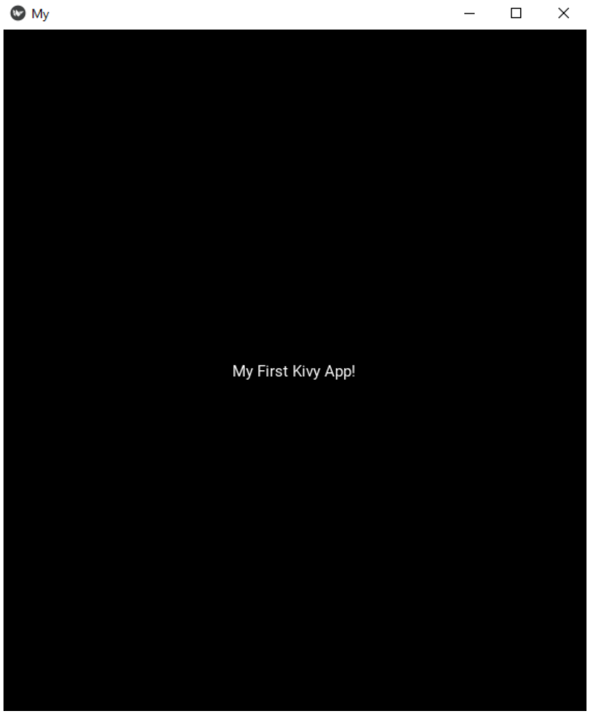
```
import kivy
from kivy.app import App
from kivy.uix.label import Label
```



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

10. Add the code below to create a class called **MyApp** which inherits **App**, a module in the Kivy library. This code also defines a *build* function that displays a label. Finally, we run the app using MyApp().run() by clicking ***Run*** → ***Run Module*** or by ***pressing F5***; *note that the long underscore is a set of two underscores*. 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 aspects:

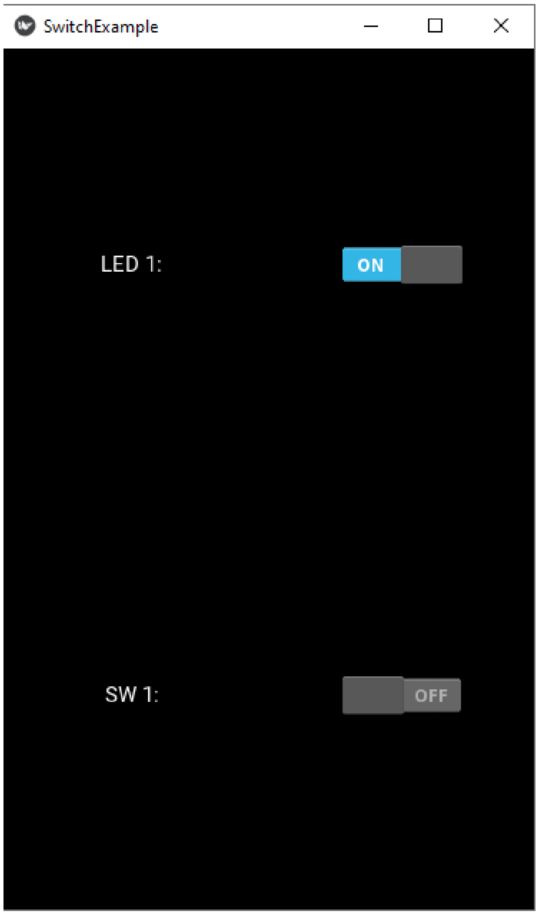
```
import kivy        #import modules
from kivy.app import App
from kivy.uix.label import Label


class MyApp(App):        #build function
   def build(self):
      return Label (text="My First Kivy App!")

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



My First Kivy App!

11. Let’s create an app that can *"eventually" (in a future tutorial)* update values in our ***000webhost*** database (*in this tutorial, we are simply making an Android interface which will be used to update the database in a future tutorial*). We will create ***two "SwitchCompat" switches “LED 1” and “SW 1”*** (software ***visual*** switches to represent status of actuator ***LED 1*** and sensor ***SW 1***) that can be switched on and off from the app. Make a new IDLE file and insert the following code, and run the app by clicking ***Run➔ Run Module*** or by ***pressing F5***:

```
#import needed modules
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 (visual) that can be turned off or on
        self.settings.bind(active=switch_callback1)

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

        self.add_widget(self.settings)              #Create a switch (visual) that can be turned off or on
        self.settings.bind(active=switch_callback2)


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


def switch_callback2(switchObject, switchValue):   #output status of the switch (visual) to the console
    print('Value of SW 1: ', switchValue)


class SwitchExample(App):                           #build function
    def build(self):
        return SwitchContainer()


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



LED 1: ON

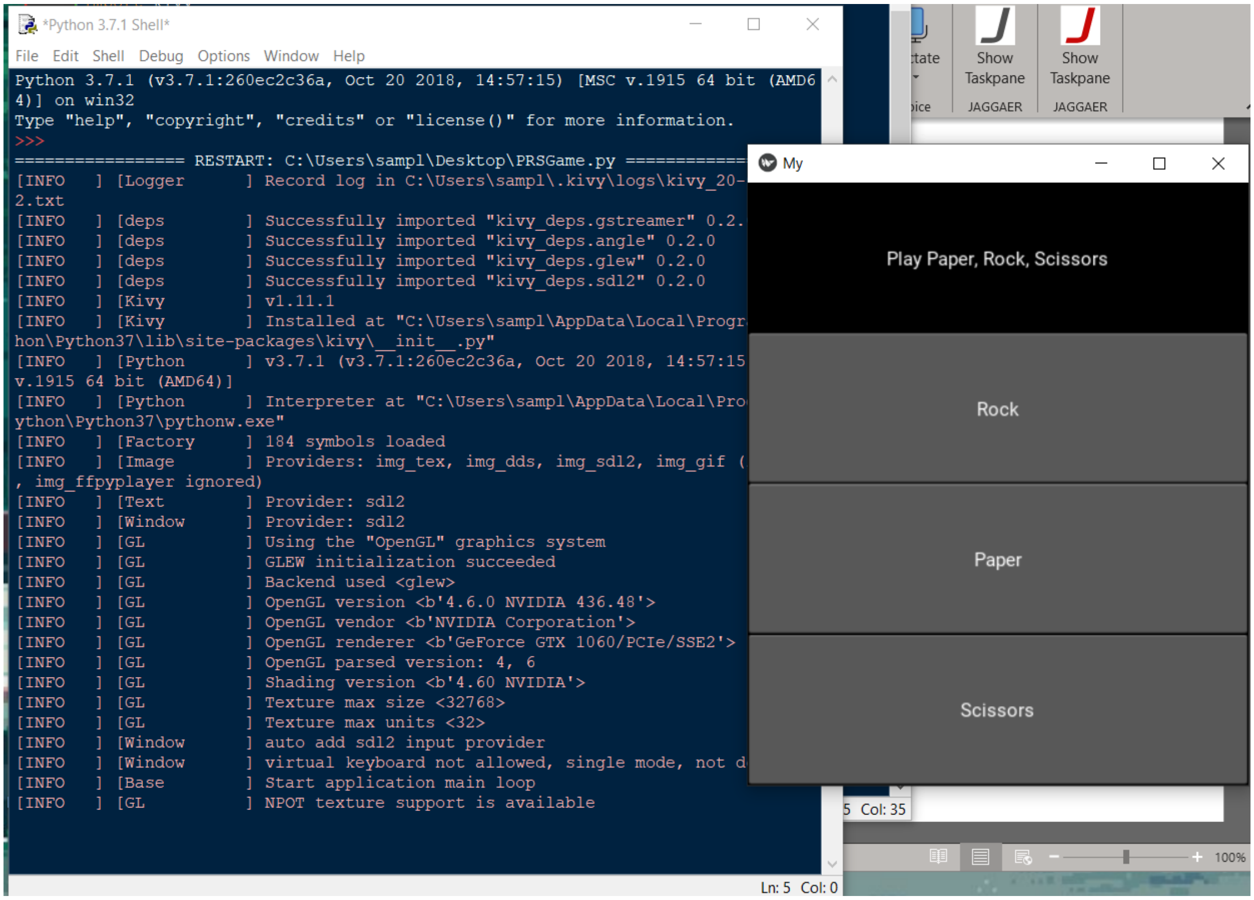
SW 1: OFF

Note that we have been referring to the on/off representation of LED 1 and SW 1 (or any binary entity) as ***software*** or ***visual*** switches. Don't get this confused with the sensor simulated by switch SW1. A software/visual switch simply displays the status of a binary entity, e.g. a binary sensor (such as a switch) or a binary actuator (such as an LED). These are referred to as ***SwitchCompat*** in Android Studio lingo.

12. We can also make a simple game using buttons. We’ll create a simple game of ***Rock-Paper-Scissors***.

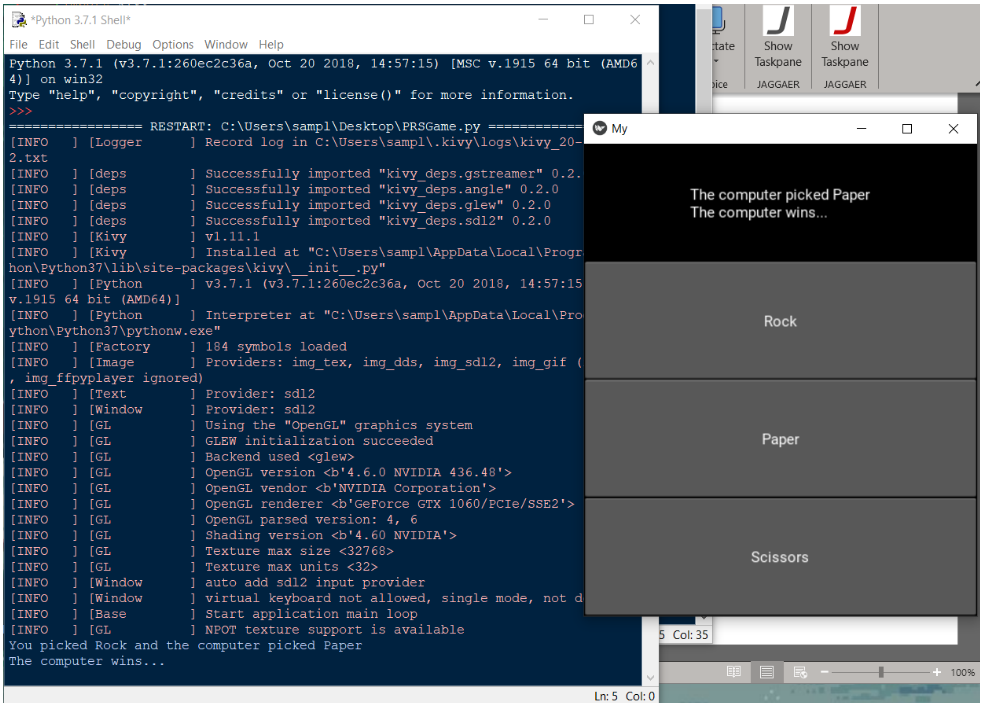
```
#import modules
import kivy
from random import randint
from kivy.app import App
from kivy.uix.label import Label
from kivy.uix.gridlayout import GridLayout
from kivy.uix.button import Button

class LoginScreen(GridLayout):
    def __init__(self, **kwargs):
        super(LoginScreen,self).__init__(**kwargs)
        self.cols = 1                          #Making it 1 column to make it look nicer for mobile

        #Define the buttons so the user can select one and bind them
        self.txtLabel = Label(text='Play Paper, Rock, Scissors')

        self.btnRock = Button(text='Rock')
        self.btnRock.bind(on_press=self.pressed)

        self.btnPaper = Button(text='Paper')
        self.btnPaper.bind(on_press=self.pressed)

        self.btnScissors = Button(text='Scissors')
        self.btnScissors.bind(on_press=self.pressed)

        #Add the buttons to the grid to the displayed
        self.add_widget(self.txtLabel)
        self.add_widget(self.btnRock)
        self.add_widget(self.btnPaper)
        self.add_widget(self.btnScissors)

    #Defining the function for when the buttons are pressed
    def pressed(self, instance):
        #We list the possible choices and pick a random one
        choices = ['Rock', 'Paper', 'Scissors']
        #We need to generate a random number to use as the computer's move
        computer = choices[randint(0,2)]

        #Read the player's choice
        player = instance.text

        #Display your choice and the computer's to the console and window
        print('You picked ' + player + ' and the computer picked ' + computer)
        self.txtLabel.text = 'The computer picked ' + computer
        #Now we find the winner
        if player == computer:
            winner = 'Draw'
        elif player == 'Rock' and computer == 'Scissors':
            winner = 'You win!'
        elif player == 'Rock' and computer == 'Paper':
            winner = 'The computer wins...'
        elif player == 'Paper' and computer == 'Rock':
            winner = 'You win!'
        elif player == 'Paper' and computer == 'Scissors':
            winner = 'The computer wins...'
        elif player == 'Scissors' and computer == 'Paper':
            winner = 'You win!'
        else:
            winner = 'The computer wins...'

        #Output the winner to the console and window
        if winner == 'Draw':
            print('It was a draw. Try again!')
            self.txtLabel.text += '\nIt was a draw. Try again.'
        else:
            print(winner)
            self.txtLabel.text += '\n' + winner

class MyApp(App):                #build function
    def build(self):
        return LoginScreen()

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

Below is the App when launched:



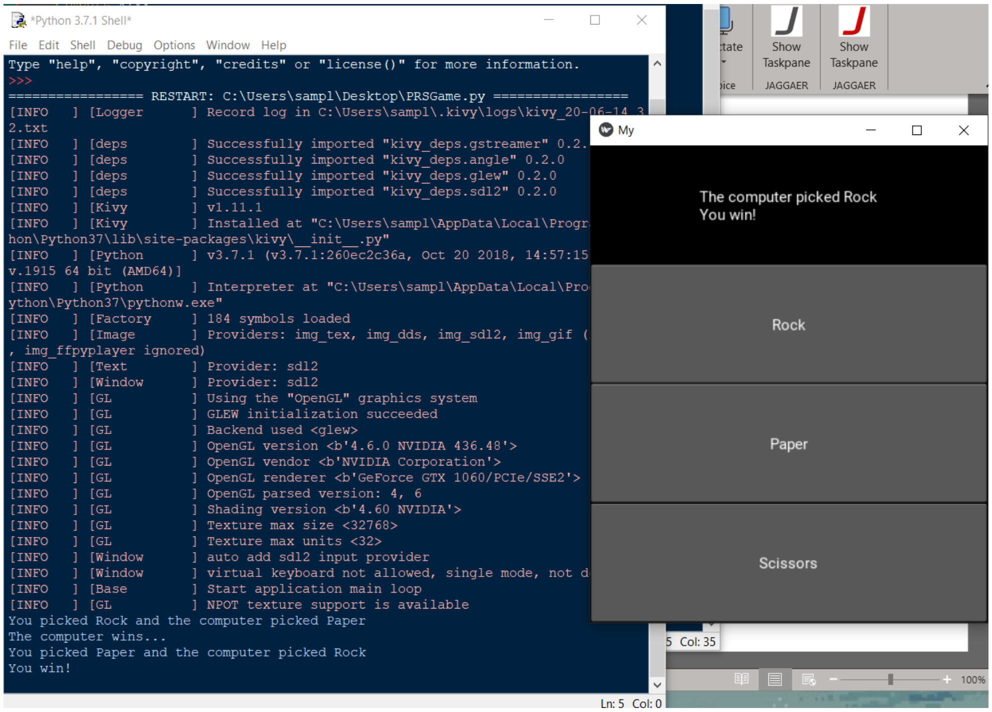
```
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:57:15) [MSC v.1915 64 bit (AMD6
4)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
================= RESTART: C:\Users\sampl\Desktop\PRSGame.py ===========
[INFO   ] [Logger      ] Record log in C:\Users\sampl\.kivy\logs\kivy_20-
2.txt
[INFO   ] [deps        ] Successfully imported "kivy_deps.gstreamer" 0.2.
[INFO   ] [deps        ] Successfully imported "kivy_deps.angle" 0.2.0
[INFO   ] [deps        ] Successfully imported "kivy_deps.glew" 0.2.0
[INFO   ] [deps        ] Successfully imported "kivy_deps.sdl2" 0.2.0
[INFO   ] [Kivy        ] v1.11.1
[INFO   ] [Kivy        ] Installed at "C:\Users\sampl\AppData\Local\Progr
hon\Python37\lib\site-packages\kivy\__init__.py"
[INFO   ] [Python      ] v3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:57:15
v.1915 64 bit (AMD64)]
[INFO   ] [Python      ] Interpreter at "C:\Users\sampl\AppData\Local\Pro
ython\Python37\pythonw.exe"
[INFO   ] [Factory     ] 184 symbols loaded
[INFO   ] [Image       ] Providers: img_tex, img_dds, img_sdl2, img_gif (
, img_ffpyplayer ignored)
[INFO   ] [Text        ] Provider: sdl2
[INFO   ] [Window      ] Provider: sdl2
[INFO   ] [GL          ] Using the "OpenGL" graphics system
[INFO   ] [GL          ] GLEW initialization succeeded
[INFO   ] [GL          ] Backend used <glew>
[INFO   ] [GL          ] OpenGL version <b'4.6.0 NVIDIA 436.48'>
[INFO   ] [GL          ] OpenGL vendor <b'NVIDIA Corporation'>
[INFO   ] [GL          ] OpenGL renderer <b'GeForce GTX 1060/PCIe/SSE2'>
[INFO   ] [GL          ] OpenGL parsed version: 4, 6
[INFO   ] [GL          ] Shading version <b'4.60 NVIDIA'>
[INFO   ] [GL          ] Texture max size <32768>
[INFO   ] [GL          ] Texture max units <32>
[INFO   ] [Window      ] auto add sdl2 input provider
[INFO   ] [Window      ] virtual keyboard not allowed, single mode, not d
[INFO   ] [Base        ] Start application main loop
[INFO   ] [GL          ] NPOT texture support is available
```

And two games of ***Rock-Paper-Scissors***:



```
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:57:15) [MSC v.1915 64 bit (AMD6
4)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================= RESTART: C:\Users\sampl\Desktop\PRSGame.py ============
[INFO   ] [Logger      ] Record log in C:\Users\sampl\.kivy\logs\kivy_20-
2.txt
[INFO   ] [deps        ] Successfully imported "kivy_deps.gstreamer" 0.2.
[INFO   ] [deps        ] Successfully imported "kivy_deps.angle" 0.2.0
[INFO   ] [deps        ] Successfully imported "kivy_deps.glew" 0.2.0
[INFO   ] [deps        ] Successfully imported "kivy_deps.sdl2" 0.2.0
[INFO   ] [Kivy        ] v1.11.1
[INFO   ] [Kivy        ] Installed at "C:\Users\sampl\AppData\Local\Progr
hon\Python37\lib\site-packages\kivy\__init__.py"
[INFO   ] [Python      ] v3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:57:15
v.1915 64 bit (AMD64)]
[INFO   ] [Python      ] Interpreter at "C:\Users\sampl\AppData\Local\Pro
ython\Python37\pythonw.exe"
[INFO   ] [Factory     ] 184 symbols loaded
[INFO   ] [Image       ] Providers: img_tex, img_dds, img_sdl2, img_gif (
, img_ffpyplayer ignored)
[INFO   ] [Text        ] Provider: sdl2
[INFO   ] [Window      ] Provider: sdl2
[INFO   ] [GL          ] Using the "OpenGL" graphics system
[INFO   ] [GL          ] GLEW initialization succeeded
[INFO   ] [GL          ] Backend used <glew>
[INFO   ] [GL          ] OpenGL version <b'4.6.0 NVIDIA 436.48'>
[INFO   ] [GL          ] OpenGL vendor <b'NVIDIA Corporation'>
[INFO   ] [GL          ] OpenGL renderer <b'GeForce GTX 1060/PCIe/SSE2'>
[INFO   ] [GL          ] OpenGL parsed version: 4, 6
[INFO   ] [GL          ] Shading version <b'4.60 NVIDIA'>
[INFO   ] [GL          ] Texture max size <32768>
[INFO   ] [GL          ] Texture max units <32>
[INFO   ] [Window      ] auto add sdl2 input provider
[INFO   ] [Window      ] virtual keyboard not allowed, single mode, not d
[INFO   ] [Base        ] Start application main loop
[INFO   ] [GL          ] NPOT texture support is available
You picked Rock and the computer picked Paper
The computer wins...
```

```
The computer picked Paper
The computer wins...
```



```
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================= RESTART: C:\Users\sampl\Desktop\PRSGame.py =================
[INFO   ] [Logger      ] Record log in C:\Users\sampl\.kivy\logs\kivy_20-06-14_3
2.txt
[INFO   ] [deps        ] Successfully imported "kivy_deps.gstreamer" 0.2.
[INFO   ] [deps        ] Successfully imported "kivy_deps.angle" 0.2.0
[INFO   ] [deps        ] Successfully imported "kivy_deps.glew" 0.2.0
[INFO   ] [deps        ] Successfully imported "kivy_deps.sdl2" 0.2.0
[INFO   ] [Kivy        ] v1.11.1
[INFO   ] [Kivy        ] Installed at "C:\Users\sampl\AppData\Local\Progr
hon\Python37\lib\site-packages\kivy\__init__.py"
[INFO   ] [Python      ] v3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:57:15
v.1915 64 bit (AMD64)]
[INFO   ] [Python      ] Interpreter at "C:\Users\sampl\AppData\Local\Pro
ython\Python37\pythonw.exe"
[INFO   ] [Factory     ] 184 symbols loaded
[INFO   ] [Image       ] Providers: img_tex, img_dds, img_sdl2, img_gif (
, img_ffpyplayer ignored)
[INFO   ] [Text        ] Provider: sdl2
[INFO   ] [Window      ] Provider: sdl2
[INFO   ] [GL          ] Using the "OpenGL" graphics system
[INFO   ] [GL          ] GLEW initialization succeeded
[INFO   ] [GL          ] Backend used <glew>
[INFO   ] [GL          ] OpenGL version <b'4.6.0 NVIDIA 436.48'>
[INFO   ] [GL          ] OpenGL vendor <b'NVIDIA Corporation'>
[INFO   ] [GL          ] OpenGL renderer <b'GeForce GTX 1060/PCIe/SSE2'>
[INFO   ] [GL          ] OpenGL parsed version: 4, 6
[INFO   ] [GL          ] Shading version <b'4.60 NVIDIA'>
[INFO   ] [GL          ] Texture max size <32768>
[INFO   ] [GL          ] Texture max units <32>
[INFO   ] [Window      ] auto add sdl2 input provider
[INFO   ] [Window      ] virtual keyboard not allowed, single mode, not d
[INFO   ] [Base        ] Start application main loop
[INFO   ] [GL          ] NPOT texture support is available
You picked Rock and the computer picked Paper
The computer wins...
You picked Paper and the computer picked Rock
You win!
```

```
The computer picked Rock
You win!
```