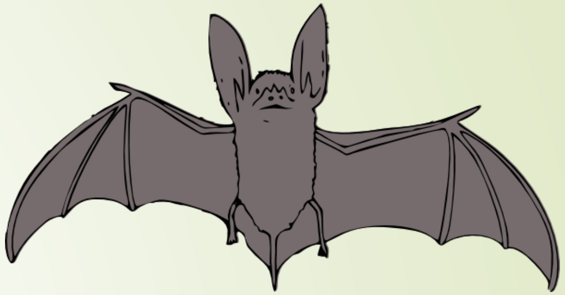
ScriptEditorV3: A Tkinter GUI Python Application

Geography 565 – Python Programming - Summer 2017

Final Class Project

John Marshall

A ‘Cat’, ‘Bat’, ‘Mat’ Philosophy to Code Development



Project Goal

ScriptEditor is designed to help developers edit their Python scripts during application development.

Tkinter Module

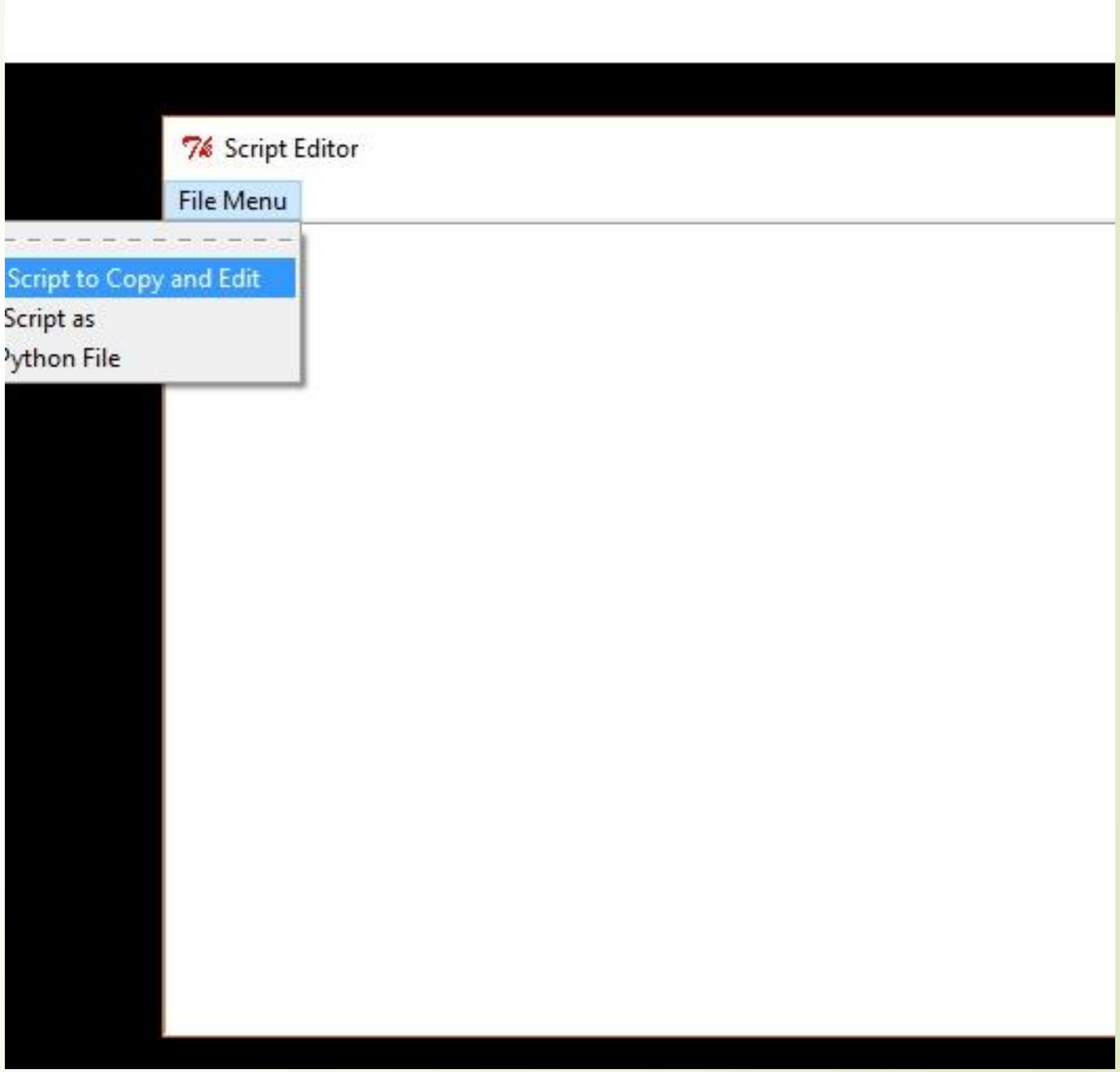
Tkinter is a version of Tk code (used to develop GUIs) specifically designed for use in Python 2. It allows developers to import a comprehensive list of modules useful for creating GUI controls such as labels, text boxes, buttons, etc.

Tkinter Syntax

```
from Tkinter import *
root = Tk()
myApp = myClassConstructor
root.mainloop()
```

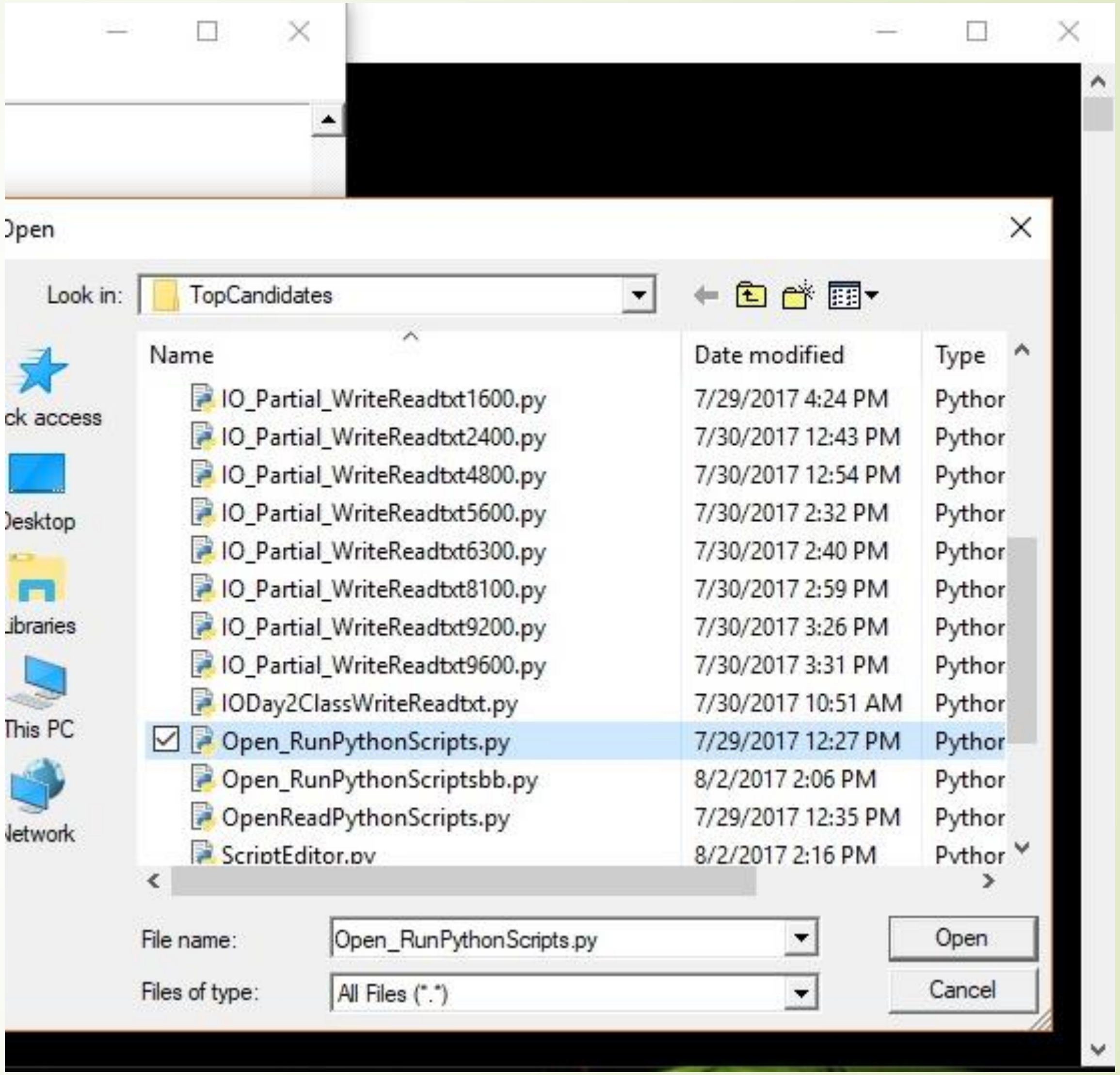
Python Method to GUI

```
#Method that when called allows script/text from a selected file to be loaded into a text window
   def load(self):
    self.file = tkFileDialog.askopenfile()
    self.text.delete(1.0, END)
    if self.file:
     self.text.insert(1.0, self.file.read())
filemenu.add_command(label="Load Script to Copy and Edit",command=self.load)
```



Python Method to GUI

```
#Method that when called allows script/text from a selected file to be loaded into a text window

   def load(self):

     self.file = tkFileDialog.askopenfile()

     self.text.delete(1.0, END)

     if self.file:

       self.text.insert(1.0, self.file.read())

filemenu.add_command(label="Load Script to Copy and Edit",command=self.load)
```



Look in: TopCandidates

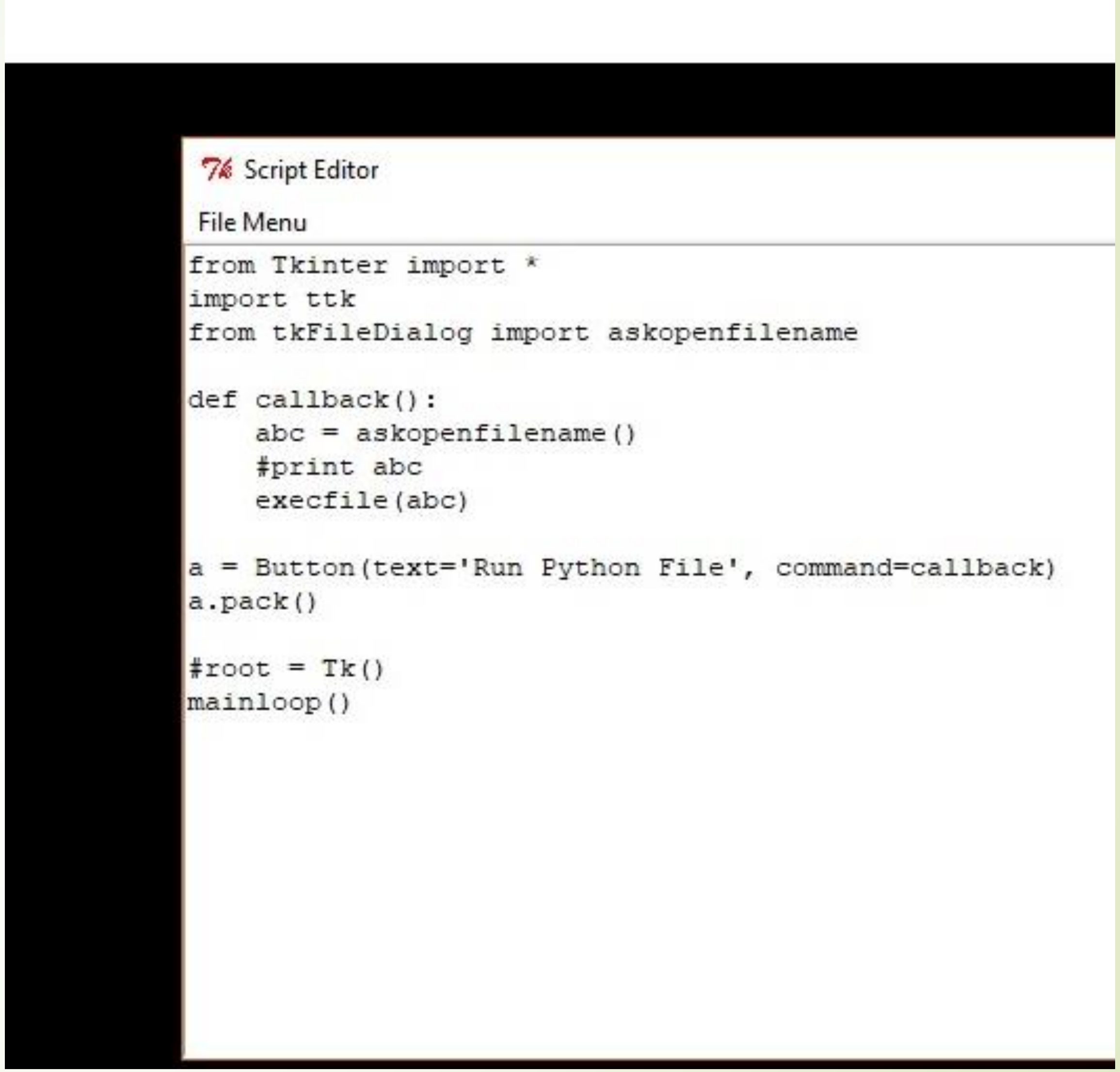
	Name	Date modified	Type
	IO_Partial_WriteReadtxt1600.py	7/29/2017 4:24 PM	Pythor
	IO_Partial_WriteReadtxt2400.py	7/30/2017 12:43 PM	Pythor
	IO_Partial_WriteReadtxt4800.py	7/30/2017 12:54 PM	Pythor
	IO_Partial_WriteReadtxt5600.py	7/30/2017 2:32 PM	Pythor
	IO_Partial_WriteReadtxt6300.py	7/30/2017 2:40 PM	Pythor
	IO_Partial_WriteReadtxt8100.py	7/30/2017 2:59 PM	Pythor
	IO_Partial_WriteReadtxt9200.py	7/30/2017 3:26 PM	Pythor
	IO_Partial_WriteReadtxt9600.py	7/30/2017 3:31 PM	Pythor
	IODay2ClassWriteReadtxt.py	7/30/2017 10:51 AM	Pythor
☑	Open_RunPythonScripts.py	7/29/2017 12:27 PM	Pythor
	Open_RunPythonScriptsbb.py	8/2/2017 2:06 PM	Pythor
	OpenReadPythonScripts.py	7/29/2017 12:35 PM	Pythor
	ScriptEditor.py	8/2/2017 2:16 PM	Pythor

File name: Open_RunPythonScripts.py

Files of type: All Files (*.*)

Python Method to GUI

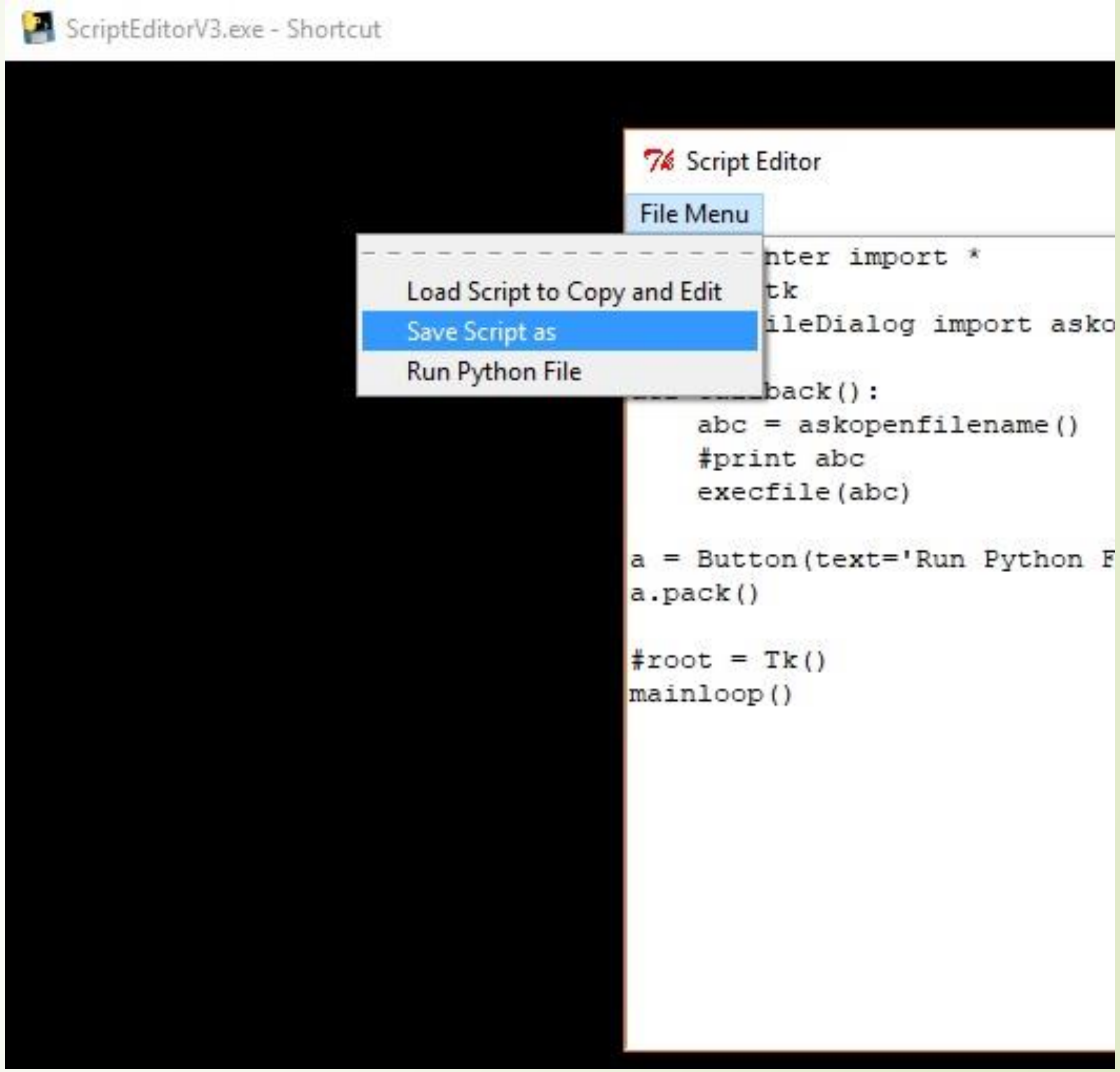
```
#Method that when called allows script/text from a selected file to be loaded into a text window

   def load(self):

     self.file = tkFileDialog.askopenfile()

     self.text.delete(1.0, END)

     if self.file:

       self.text.insert(1.0, self.file.read())

filemenu.add_command(label="Load Script to Copy and Edit",command=self.load)
```



```
from Tkinter import *
import ttk
from tkFileDialog import askopenfilename

def callback():
    abc = askopenfilename()
    #print abc
    execfile(abc)

a = Button(text='Run Python File', command=callback)
a.pack()

#root = Tk()
mainloop()
```

Python Method to GUI

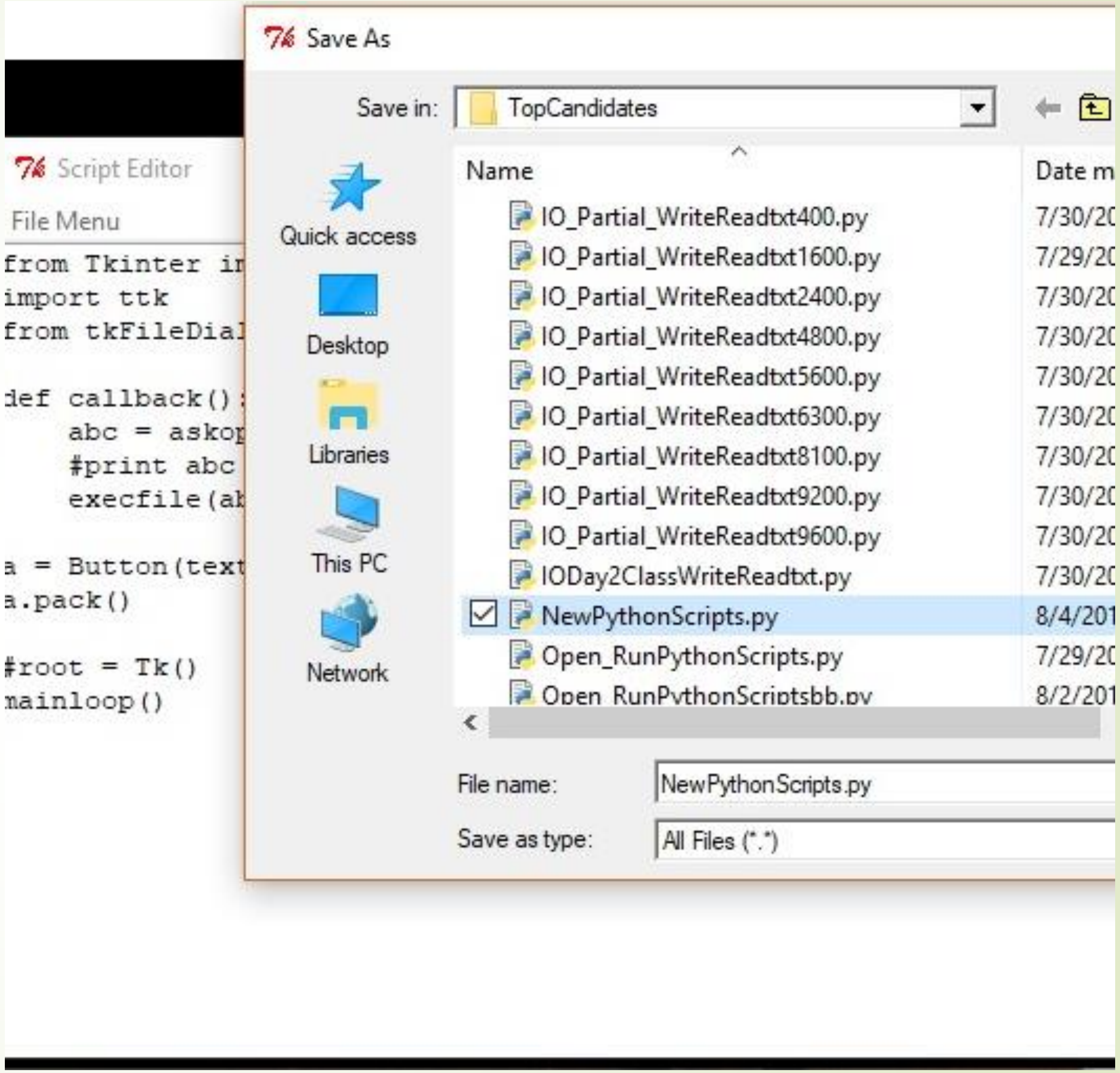
```
 #Method that when called allows a selected file to be saved in a file directory

    def saveas(self):

      self.file = tkFileDialog.asksaveasfile()

      if self.file:

        self.file.write(self.text.get(1.0, END))

filemenu.add_command(label="Save Script as",command=self.saveas)
```



```
nter import *
tk
ileDialog import asko

back():
    abc = askopenfilename()
    #print abc
    execfile(abc)

a = Button(text='Run Python F
a.pack()

#root = Tk()
mainloop()
```

Python Method to GUI

```
 #Method that when called allows a selected file to be saved in a file directory

     def saveas(self):

       self.file = tkFileDialog.asksaveasfile()

       if self.file:

         self.file.write(self.text.get(1.0,
END))

filemenu.add_command(label="Save Script as",command=self.saveas)
```



```
from Tkinter ir
import ttk
from tkFileDial

def callback():
    abc = askop
    #print abc
    execfile(ak

a = Button(text
a.pack()

#root = Tk()
mainloop()
```

Save in: TopCandidates

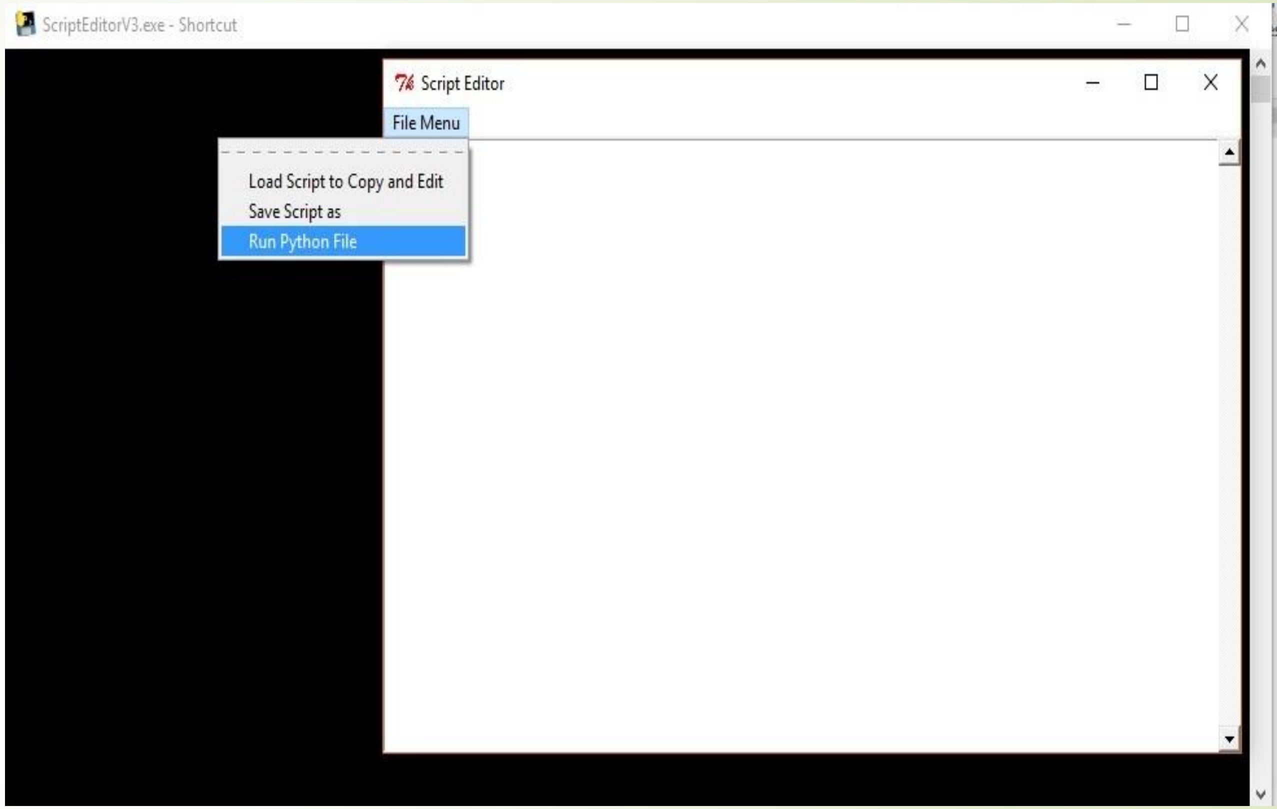
	Name	Date m
	IO_Partial_WriteReadtxt400.py	7/30/20
	IO_Partial_WriteReadtxt1600.py	7/29/20
	IO_Partial_WriteReadtxt2400.py	7/30/20
	IO_Partial_WriteReadtxt4800.py	7/30/20
	IO_Partial_WriteReadtxt5600.py	7/30/20
	IO_Partial_WriteReadtxt6300.py	7/30/20
	IO_Partial_WriteReadtxt8100.py	7/30/20
	IO_Partial_WriteReadtxt9200.py	7/30/20
	IO_Partial_WriteReadtxt9600.py	7/30/20
	IODay2ClassWriteReadtxt.py	7/30/20
☑	NewPythonScripts.py	8/4/201
	Open_RunPythonScripts.py	7/29/20
	Open_RunPythonScriptsbb.py	8/2/201

File name: NewPythonScripts.py

Save as type: All Files (*.*)

Python Method to GUI

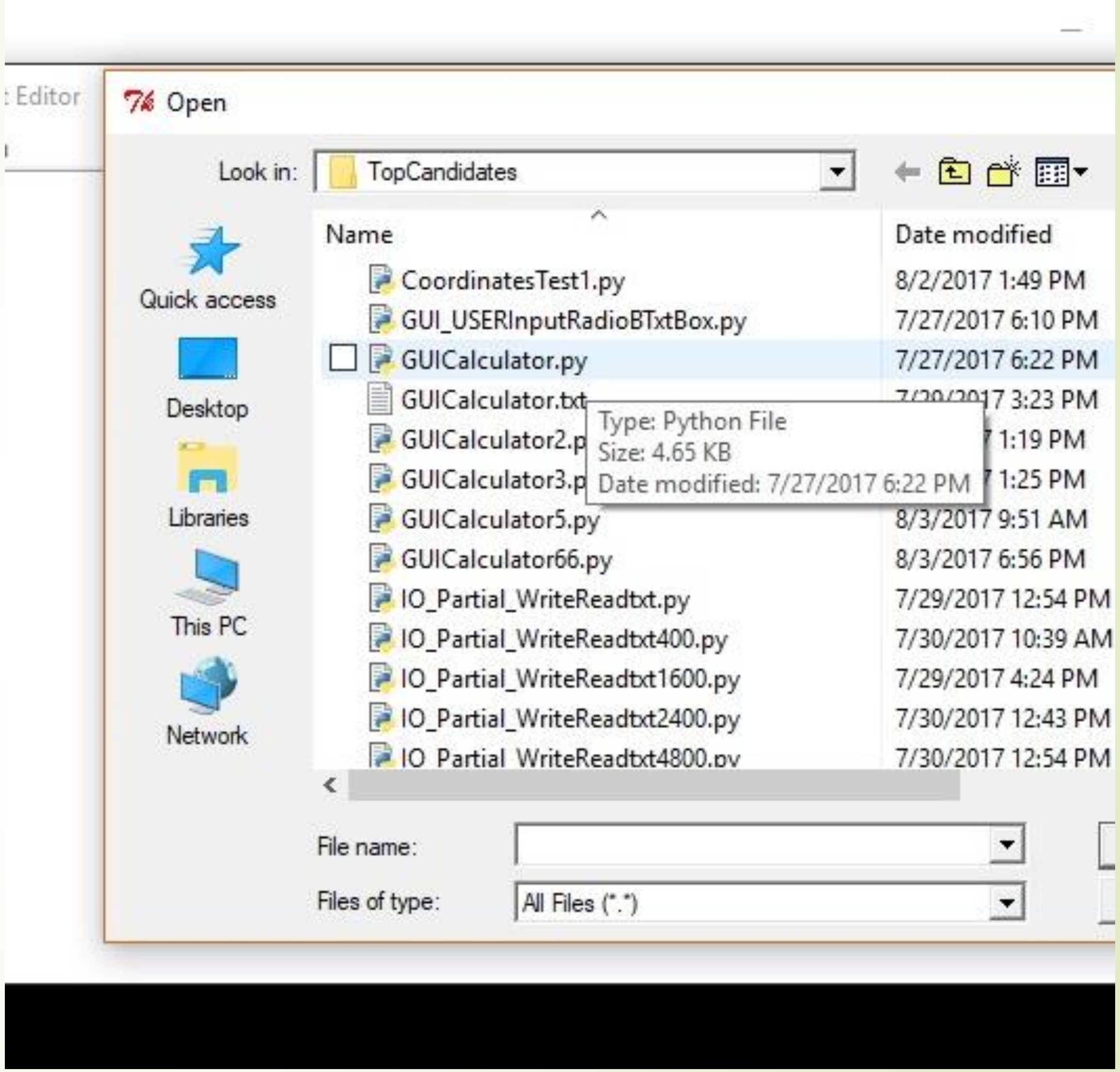
```
#Method that when called allows execution (run) of a file
    def callback(self):
        self.file = tkFileDialog.askopenfilename()
        if self.file:
            execfile(self.file)

filemenu.add_command(label="Run Python File",command=self.callback)
```



Python Method to GUI

#Method that when called allows execution (run) of a file

```
    def callback(self):
        self.file =
tkFileDialog.askopenfilename()
        if self.file:
            execfile(self.file)


 filemenu.add_command(label="Run
Python File",command=self.callback)
```



Look in: TopCandidates

Name	Date modified
CoordinatesTest1.py	8/2/2017 1:49 PM
GUI_USERInputRadioBTxtBox.py	7/27/2017 6:10 PM
GUICalculator.py	7/27/2017 6:22 PM
GUICalculator.txt	7/29/2017 3:23 PM
GUICalculator2.p	7 1:19 PM
GUICalculator3.p	7 1:25 PM
GUICalculator5.py	8/3/2017 9:51 AM
GUICalculator66.py	8/3/2017 6:56 PM
IO_Partial_WriteReadtxt.py	7/29/2017 12:54 PM
IO_Partial_WriteReadtxt400.py	7/30/2017 10:39 AM
IO_Partial_WriteReadtxt1600.py	7/29/2017 4:24 PM
IO_Partial_WriteReadtxt2400.py	7/30/2017 12:43 PM
IO_Partial_WriteReadtxt4800.py	7/30/2017 12:54 PM

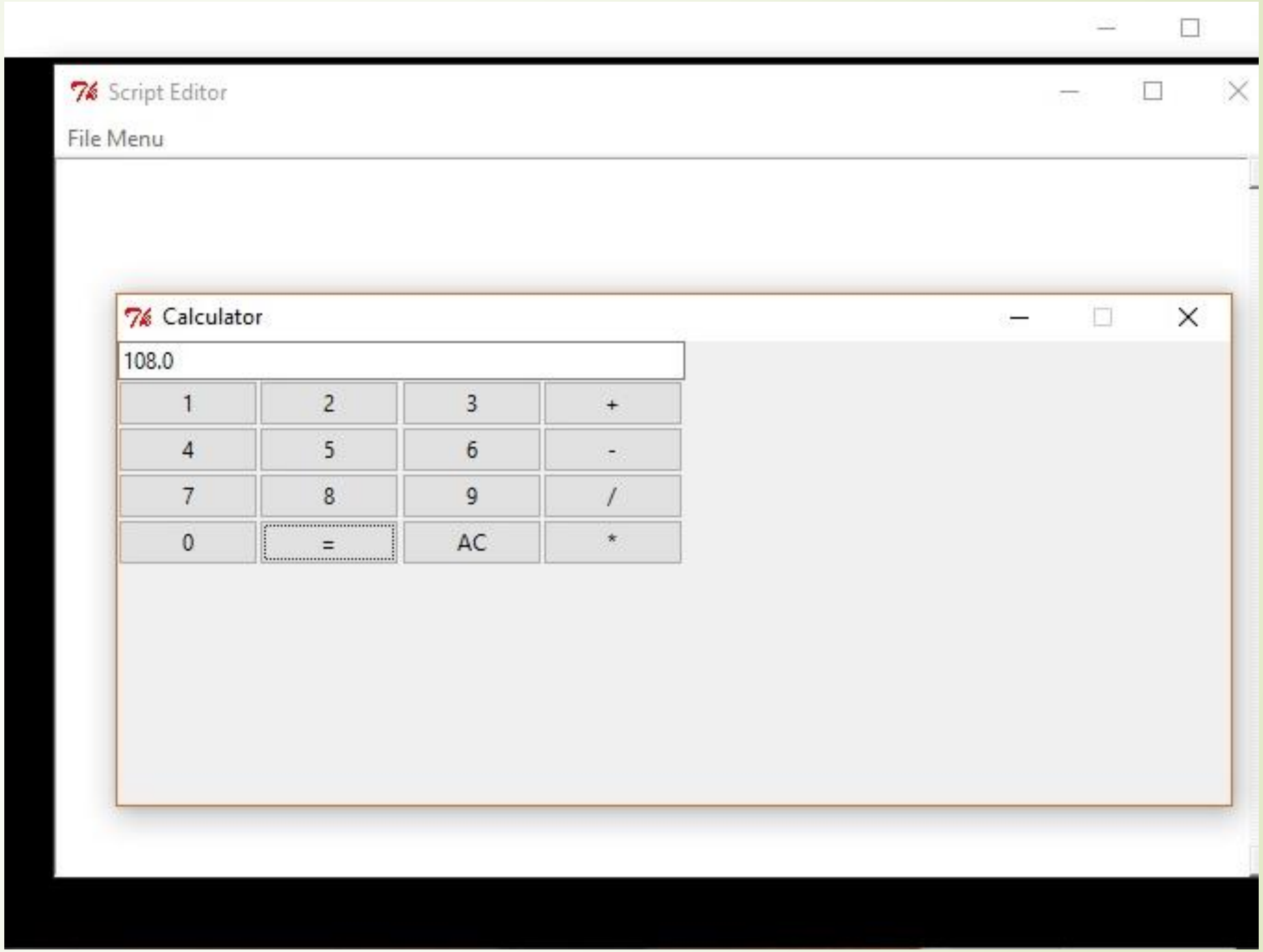
Type: Python File
Size: 4.65 KB
Date modified: 7/27/2017 6:22 PM

File name:

Files of type: All Files (*.*)

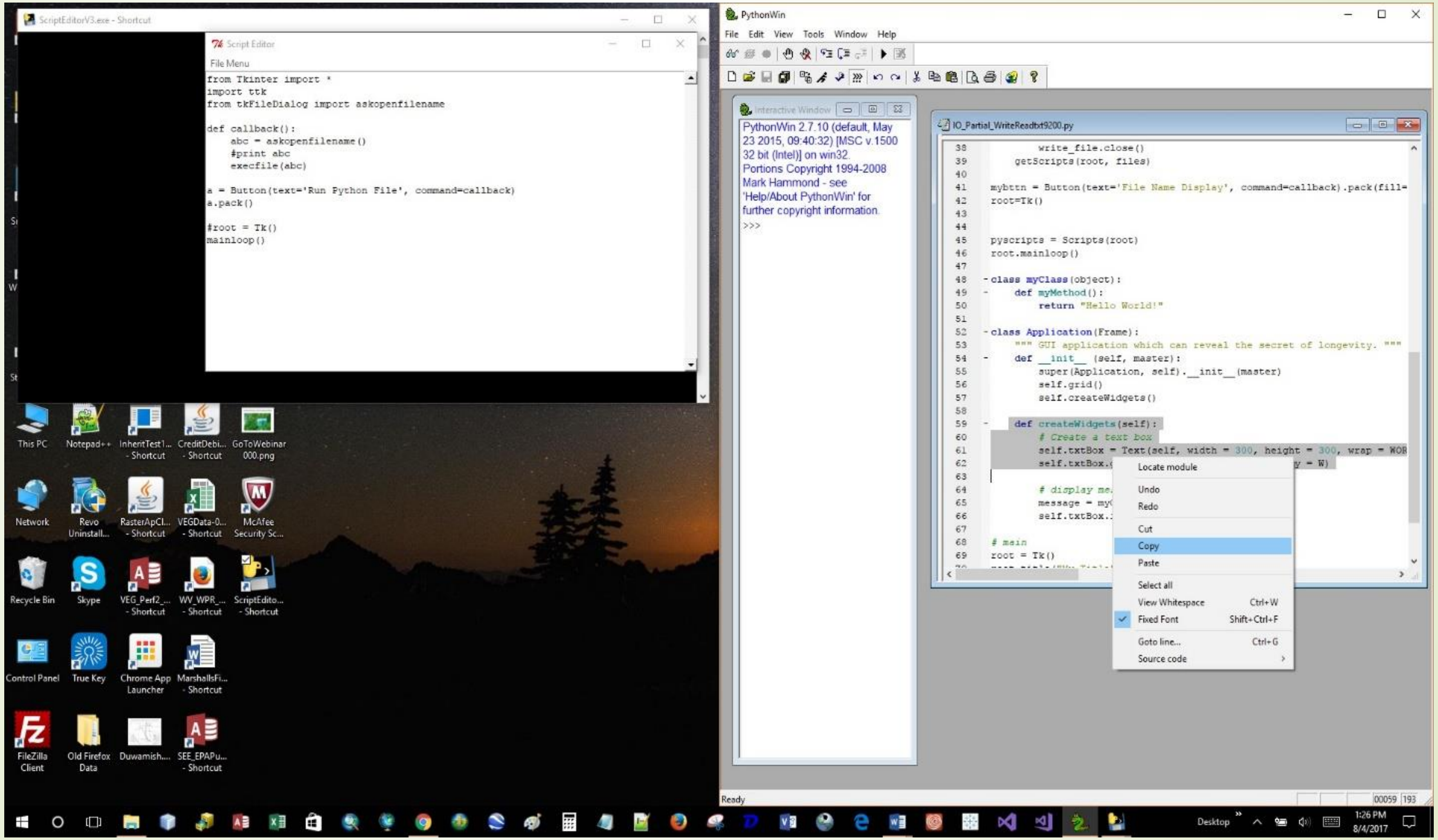
Python Method to GUI

```
#Method that when called allows execution (run) of a file
    def callback(self):
        self.file = tkFileDialog.askopenfilename()
        if self.file:
            execfile(self.file)



filemenu.add_command(label=" Run Python File",command=self.callback)
```



108.0

ScriptEditorV3 with Python IDE



```
from Tkinter import *
import ttk
from tkFileDialog import askopenfilename

def callback():
    abc = askopenfilename()
    #print abc
    execfile(abc)

a = Button(text='Run Python File', command=callback)
a.pack()

#root = Tk()
mainloop()
```

```
PythonWin 2.7.10 (default, May
23 2015, 09:40:32) [MSC v.1500
32 bit (Intel)] on win32.
Portions Copyright 1994-2008
Mark Hammond - see
'Help/About PythonWin' for
further copyright information.
>>>
```

```
            write_file.close()
        getScripts(root, files)

mybttn = Button(text='File Name Display', command=callback).pack(fill=
root=Tk()


pyscripts = Scripts(root)
root.mainloop()

class myClass(object):
    def myMethod():
        return "Hello World!"

class Application(Frame):
    """ GUI application which can reveal the secret of longevity. """
    def __init__ (self, master):
        super(Application, self).__init__(master)
        self.grid()
        self.createWidgets()

    def createWidgets(self):
        # Create a text box
        self.txtBox = Text(self, width = 300, height = 300, wrap = WOR
        self.txtBox.(                    y = W)

        # display me
        message = my(
        self.txtBox.i

# main
root = Tk()
```

Python Script to Executable Standalone Application

- Reviewed documentation for ‘PyInstaller’
- Found Python installation folder on my ‘C’ drive and navigated to the ‘Scripts’ folder (e.g., C:\Python27\ArcGIS10.4\Scripts).
- Opened my command prompt in computers with Windows 10 OS and right clicked on windows icon, selected Run, and entered ‘cmd’ into open textbox.

Python Script to Executable Standalone Application

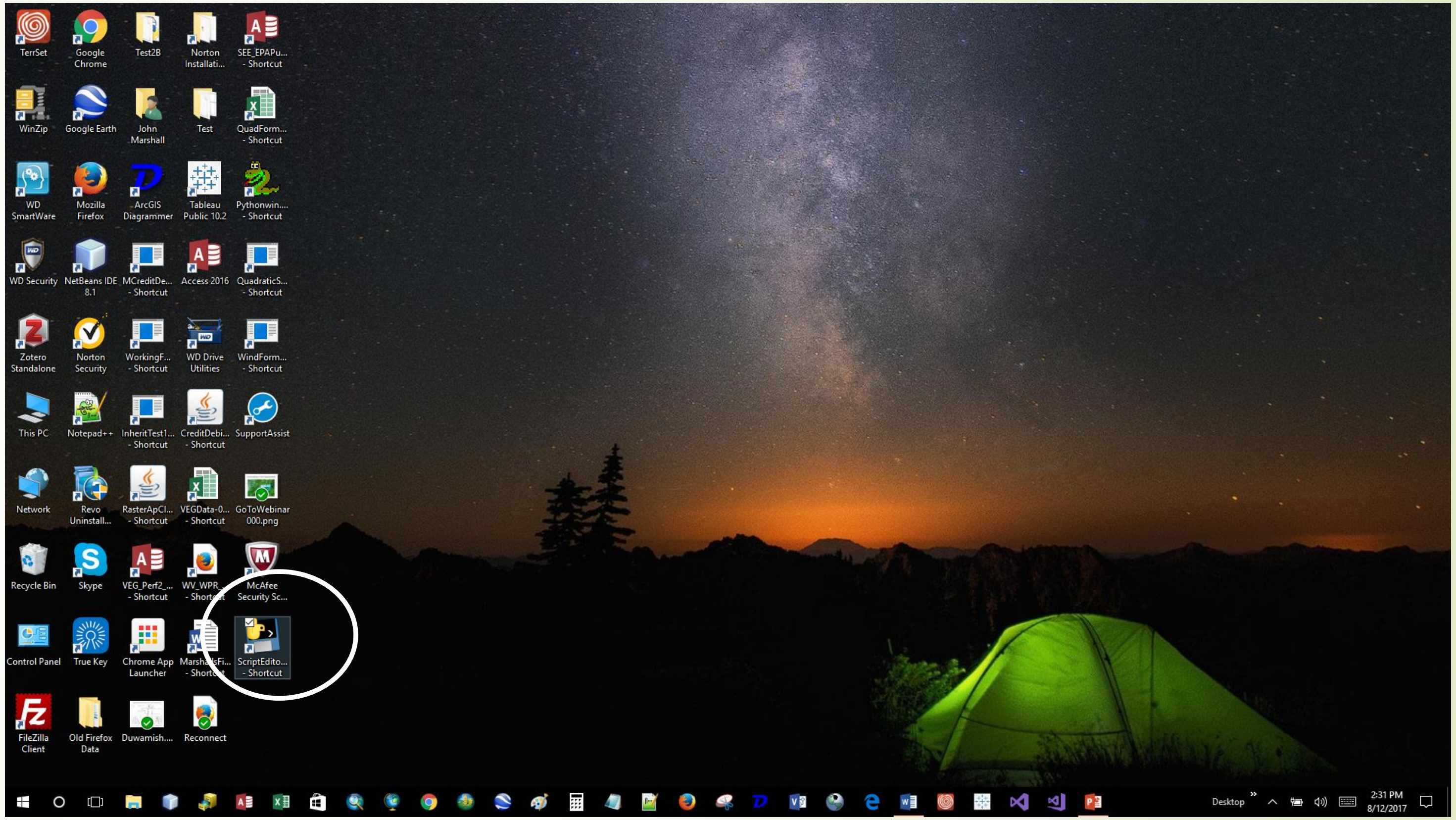
- In command line, entered ‘cd’ to change directory then copied and pasted (e.g., <control> C, <control> V) the directory path to the ‘Scripts’ folder after ‘cd’ and selected ‘Enter’ (e.g., cd C:\Python27\ArcGIS10.4\ Scripts <Enter>).
- In new directory, I entered ‘pip install pyinstaller’ (e.g., C:\Python27\ ArcGIS10.4\Scripts> pip install pyinstaller<Enter>). I made sure I was connected to the internet to enable pip to find and install pyinstaller. It took a few seconds for the installation to complete. Once completed I observed the installed new files in my ‘Scripts’ folder.

Python Script to Executable Standalone Application

- I navigated back to my command window and made sure it was still using the ‘Scripts’ directory (e.g., ‘C:\Python27\ArcGIS10.4\ Scripts>’). Then I entered: pyinstaller.exe - -onefile and the directory path to my Python script (ScriptEditorV3.py) to create a standalone executable application (e.g., C:\Python27\ ArcGIS10.4\ Scripts> pyinstaller.exe –onefile C:\EsriPress\Python\ Data\ Exercise13\ ScriptEditorV3.py<Enter>).
- Once the procedure completed running, the last line of the run output indicated where my execution file had been placed (e.g., C:\Python27\ArcGIS10.4 \ Scripts\dist\ScriptEditorV3.exe).

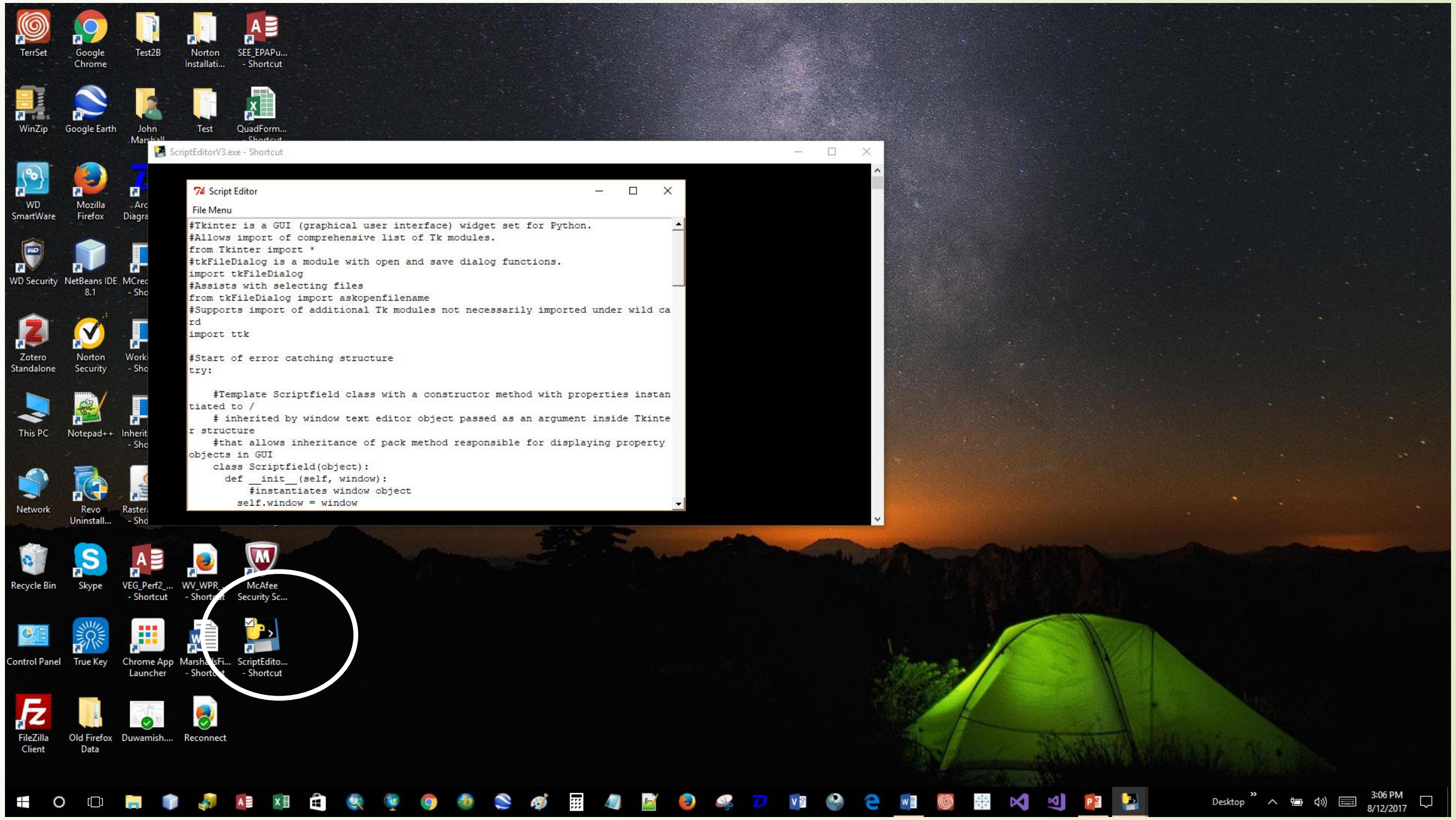
ScriptEditorV3.exec

DESKTOP SHORTCUT



ScriptEditorV3.exec

DESKTOP SHORTCUT



```
#Tkinter is a GUI (graphical user interface) widget set for Python.
#Allows import of comprehensive list of Tk modules.
from Tkinter import *
#tkFileDialog is a module with open and save dialog functions.
import tkFileDialog
#Assists with selecting files
from tkFileDialog import askopenfilename
#Supports import of additional Tk modules not necessarily imported under wild ca
rd
import ttk

#Start of error catching structure
try:

    #Template Scriptfield class with a constructor method with properties instan
tiated to /
    # inherited by window text editor object passed as an argument inside Tkinte
r structure
    #that allows inheritance of pack method responsible for displaying property
objects in GUI
    class Scriptfield(object):
      def __init__(self, window):
          #instantiates window object
        self.window = window
```

ScriptEditorV3.py Script, Standalone Execution File, and Documentation

https://drive.google.com/open?id=0B4B9I8GgTZdeODA0eUdrdDJqRkU

https://docs.python.org/2/library/tkinter.html