

# **Python Project Report**

**On**

**CGPA Calculator**

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

**By**

**Abhishek singh:12205842**

**Anuratna pandey:12204319**

**Navin kumar:12204830**

**Section : K21LL**



**School of Computer Science and Engineering**

Lovely Professional University

Phagwara, Punjab (India)

# OBJECTIVE

For college students keeping track of their cgpa is pretty important so that they can keep a balance between their academics and co curricular activities along with good marks thus its much difficult to keep on calculating again and again the tgpas and cgpa and hence this CGPA calculator helps to instantly calculate the tgpas and cgpa of a student based on certain inputs.

We have created a very user friendly graphical user interface so that user can easily run through it without any difficulty.

The project also stores all the data in a database for permanent storage. If a user wants to check already entered data we have also provided an option for that and he/she can also see the grade points related to each grade if they wish to.

The major purpose of the project is to help students keep track of their grades and markings. Also they can predict how much they have to score to keep the desired cgpa for the year.

Calculation of tgpas and cgpa have become much more easier through this project the student are easily able to know the grading system.

# INTRODUCTION

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source the fast edit-test-debug cycle makes this simple approach very effective.

While The Python Language Reference describes the exact syntax and semantics of the Python language, this library reference manual describes the standard library that is distributed with Python. It also describes some of the optional components that are commonly included in Python distributions.

Python's standard library is very extensive, offering a wide range of facilities as indicated by the long table of contents listed below. The library contains

built-in modules (written in C) that provide access to system functionality such as file I/O that would otherwise be inaccessible to Python programmers, as well as modules written in Python that provide standardized solutions for many problems that occur in everyday programming. Some of these modules are explicitly designed to encourage and enhance the portability of Python programs by abstracting away platform-specifics into platform-neutral APIs.

The Python installers for the Windows platform usually include the entire standard library and often also include many additional components. For Unix-like operating systems Python is normally provided as a collection of packages, so it may be necessary to use the packaging tools provided with the operating system to obtain some or all of the optional components.

In addition to the standard library, there is a growing collection of several thousand components (from individual programs and modules to packages and entire application development frameworks), available from the Python Package Index.

### Important standard python libraries:

1. Requests: The most famous http library written by kenneth reitz. It's a must have for every python developer.
2. Scrapy: If you are involved in webscraping then this is a must have library for you.
3. tkinter: A gui toolkit for python.
4. Pillow: A friendly fork of PIL (Python Imaging Library). It is more user friendly than PIL and is a must have for anyone who works with images.
5. SQLAlchemy: A database library.
6. BeautifulSoup: It's slow but this xml and html parsing library is very useful for beginners.
7. Twisted: The most important tool for any network application developer. It has a very beautiful api and is used by a lot of famous python developers.
8. NumPy: It provides some advance math functionalities to python.

9. SciPy: It is a library of algorithms and mathematical tools for python and has caused many scientists to switch from ruby to python.
10. matplotlib. A numerical plotting library. It is very useful for any data scientist or any data analyzer.
11. Pygame. Which developer does not like to play games and develop them ? This library will help you achieve your goal of 2d game development.
12. Pyglet. A 3d animation and game creation engine. This is the engine in which the famous python port of minecraft was made
13. pyQT. A GUI toolkit for python. It is my second choice after wxpython for developing GUI's for my python scripts.
14. pyGtk. Another python GUI library. It is the same library in which the famous.Bittorrent client is created.
15. Scapy. A packet sniffer and analyzer for python made in python.
16. pywin32. A python library which provides some useful methods and classes for interacting with windows.
17. nlTK. Natural Language Toolkit – I realize most people won't be using this one, but it's generic enough. It is a very useful library if you want to manipulate strings. But it's capacity is beyond that. Do check it out.
18. nose: A testing framework for python. It is used by millions of python developers. It is a must have if you do test driven development.
19. SymPy: SymPy can do algebraic evaluation, differentiation, expansion, complex numbers, etc. It is contained in a pure Python distribution.
20. IPython: I just can't stress enough how useful this tool is. It is a python prompt on steroids. It has completion, history, shell capabilities, and a lot more. Make sure that you take a look at it.

In this project we have made a GUI interface to calculate CGPA by taking all the details from the student like Grades, Credits and subjects for different semesters. In this GUI we have also shown a separate area with grading points which will tell user about how points are being assigned to each and every grade. This GUI project will help many universities and

colleges as it is simple to use and its User Interface is very user friendly. This GUI also has save button that will automatically save all the details of student along with all the CGPA in the connected database in the backend.

One extra button of reset is also there which will clear all the details of the student if in case he/she fills wrong data by mistake. One special feature is also there in this GUI is that there is button which says 'Click Me'. After clicking on this button it will automatically load one text box and it will tell user about all the grading points .Lastly there is REMARKS field which will show the remarks to students like good and very good according to the CGPA which will be calculated in this GUI.

This whole GUI is made by using Tkinter which is one of the best library of Python for making GUI. And it provides each and every single feature of GUI like use of buttons, checkboxes, entry boxes and labels. Its all commands are very simple and easy to implement.

For database we have used SQLite3 which is a very good database technique generally used in collaboration with python.

# Source code

```
from tkinter import *
from decimal import Decimal
import sqlite3
from tkinter import ttk
import re
```

```
def change(a):
```

```
    a=list(a)
    str=""
    for i in a:
        i=ord(i)
        i=i+1
        i=chr(i)
        str=str+i
    return str
```

```
def unchange(a):
```

```
    a=list(a)
    st=""
    for i in a:
        i=ord(i)
        i=i-1
        i=chr(i)
        st=st+i
    return st
```

```
def tgpa1():
```

```
    a=int(a1.get())
    b=int(b1.get())
    c=int(c1.get())
    d=int(d1.get())
    e=int(e1.get())
    f=int(f1.get())
    g=int(g1.get())
    h=int(h1.get())
    i=int(i1.get())
    j=int(j1.get())
```



```

k=int(k1.get())
l=int(l1.get())
re1=(a*b)+(c*d)+(e*f)+(g*h)+(i*j)+(k*l)
xs1=(b+d+f+h+j+l)*9.5
pres1=re1/xs1
res1=round(pres1,2)
if (res1>10):
    dire5.set('10')
else:
    dire5.set(res1)
return res1

```

```

def tgpa2():
    a3=int(a2.get())
    b3=int(b2.get())
    c3=int(c2.get())
    d3=int(d2.get())
    e3=int(e2.get())
    f3=int(f2.get())
    g3=int(g2.get())
    h3=int(h2.get())
    i3=int(i2.get())
    j3=int(j2.get())
    k3=int(k2.get())
    l3=int(l2.get())
    re3=(a3*b3)+(c3*d3)+(e3*f3)+(g3*h3)+(i3*j3)+(k3*l3)
    xs3=(b3+d3+f3+h3+j3+l3)*9.5
    pres3=re3/xs3
    res2=round(pres3,2)
    if (res2>10):
        dire4.set('10')
    else:
        dire4.set(res2)
    return res2

```

```

def cgpa():
    a=int(a1.get())
    b=int(b1.get())
    c=int(c1.get())
    d=int(d1.get())
    e=int(e1.get())

```

```

f=int(f1.get())
g=int(g1.get())
h=int(h1.get())
i=int(i1.get())
j=int(j1.get())
k=int(k1.get())
l=int(l1.get())
a3=int(a2.get())
b3=int(b2.get())
c3=int(c2.get())
d3=int(d2.get())
e3=int(e2.get())
f3=int(f2.get())
g3=int(g2.get())
h3=int(h2.get())
i3=int(i2.get())
j3=int(j2.get())
k3=int(k2.get())
l3=int(l2.get())
name = str(name1.get())
name=change(name)

reg = str(reg1.get())
reg=change(reg)

```

```

re3=(a3*b3)+(c3*d3)+(e3*f3)+(g3*h3)+(i3*j3)+(k3*l3)+(a*b)
+(c*d)+(e*f)+(g*h)+(i*j)+(k*l)
xs3=(b3+d3+f3+h3+j3+l3+b+d+f+h+j+l)*9.5
pres3=re3/xs3
res3=round(pres3,2)
if (res3>10):
    dire.set('10')
else:
    dire.set(res3)
remark=res3
if(remark<6):
    dire6.set('Need of Improvment [C]')
elif(remark>=6 and remark<7):
    dire6.set('Ok but can be Improved [B]')
elif(remark>=7 and remark<8):
    dire6.set('Good Keep Going [B+]')
elif(remark>=8 and remark<9):

```

```

        dire6.set('V.Good Keep Going [A]')
    elif(remark>=9 and remark<=10):
        dire6.set('Brilliant [A+]')
    else:
        pass

```

```

    tgpaa=tgpa1()
    tgpab=tgpa2()
    cgpa = res3
    conn = sqlite3.connect('mysql.db')
    with conn:
        cursor = conn.cursor()
        cursor.execute('INSERT INTO result(name, id, tgpa1,
tgpa2, cgpa) VALUES(?,?,?,?),(name, reg, tgpaa, tgpab,
cgpa))
        db.close()

```

```

def search():
    search=str(search1.get())
    search=change(search)
    conn = sqlite3.connect('mysql.db')
    cursor = conn.cursor()

    cursor.execute('SELECT name FROM result where id
='+search)
    a=cursor.fetchall()
    a=str(a)
    a=unchange(a)
    a = " ".join(re.findall("[a-z]+", a))
    pos1.set(a.capitalize())

    cursor.execute('SELECT id FROM result where id
='+search)
    b=cursor.fetchall()
    b=str(b)
    b=unchange(b)
    b = " ".join(re.findall("[0-9]+", b))
    pos2.set(b)

```

```

    cursor.execute('SELECT tgpa1 FROM result where id

```

```

='+search)
    c=cursor.fetchall()
    pos3.set(c)

    cursor.execute('SELECT tgpa2 FROM result where id
='+search)
    d=cursor.fetchall()
    pos4.set(d)

    cursor.execute('SELECT cgpa FROM result where id
='+search)
    e=cursor.fetchall()
    pos5.set(e)

def reset():
    a1.set("")
    b1.set("")
    c1.set("")
    d1.set("")
    e1.set("")
    f1.set("")
    g1.set("")
    h1.set("")
    i1.set("")
    j1.set("")
    k1.set("")
    l1.set("")
    a2.set("")
    b2.set("")
    c2.set("")
    d2.set("")
    e2.set("")
    f2.set("")
    g2.set("")
    h2.set("")
    i2.set("")
    j2.set("")
    k2.set("")
    l2.set("")
    name1.set("")
    reg1.set("")
    search1.set("")
    pos1.set("")

```

```
pos2.set("")
pos3.set("")
pos4.set("")
pos5.set("")
dire5.set("")
dire.set("")
dire4.set("")
dire6.set("")
```

```
win=Tk()
win.geometry("680x750")
win.title('Cgpa Calculator')
win.configure(bg='#00ffcc')
```

```
db = sqlite3.connect('mysql.db')
cursor = db.cursor()
cursor.execute("CREATE TABLE IF NOT EXISTS
result(name TEXT, id TEXT PRIMARY KEY, tgpa1 REAL,
tgpa2 REAL, cgpa REAL)")
db.commit()
```

```
a1=StringVar()
b1=StringVar()
c1=StringVar()
d1=StringVar()
e1=StringVar()
f1=StringVar()
g1=StringVar()
h1=StringVar()
i1=StringVar()
j1=StringVar()
k1=StringVar()
l1=StringVar()
```

```
a2=StringVar()
b2=StringVar()
c2=StringVar()
d2=StringVar()
e2=StringVar()
f2=StringVar()
g2=StringVar()
```

```
h2=StringVar()  
i2=StringVar()  
j2=StringVar()  
k2=StringVar()  
l2=StringVar()
```

```
name1=StringVar()  
reg1=StringVar()  
search1=StringVar()
```

```
pos1 = StringVar()  
pos2 = StringVar()  
pos3 = StringVar()  
pos4 = StringVar()  
pos5 = StringVar()
```

```
lab=Label(win,text="",bg="#00ffcc")  
lab.grid(row=1,column=1)  
dire=Label(win,text="SEMESTER-1",font="Helvetica 15  
bold",bg="#00cca3")  
dire.grid(row=1,column=6)
```

```
dire=Label(win,text="Marks ",bg="#00ffcc",font="Times  
11 bold")  
dire.grid(row=2,column=6)  
dire=Label(win,text="Credit ",bg="#00ffcc",font="Times  
11 bold")  
dire.grid(row=2,column=8)
```

```
dire=Label(win,text="Subject 1 :  
",bg="#00ffcc",font="Times 11 bold")  
dire.grid(row=4,column=1)
```

```
entry1=Entry(win,textvariable=a1,justify='center')  
entry1.grid(row=4,column=6)
```

```
entry2=Entry(win,textvariable=b1,justify='center')  
entry2.grid(row=4,column=8)
```

```
dire=Label(win,text="Subject 2 :")
```

```
",bg="#00ffcc",font="Times 11 bold")
dire.grid(row=5,column=1)
```

```
entry3=Entry(win,textvariable=c1,justify='center')
entry3.grid(row=5,column=6)
```

```
entry4=Entry(win,textvariable=d1,justify='center')
entry4.grid(row=5,column=8)
```

```
dire=Label(win,text="Subject 3 :
",bg="#00ffcc",font="Times 11 bold")
dire.grid(row=6,column=1)
```

```
entry5=Entry(win,textvariable=e1,justify='center')
entry5.grid(row=6,column=6)
```

```
entry6=Entry(win,textvariable=f1,justify='center')
entry6.grid(row=6,column=8)
```

```
dire=Label(win,text="Subject 4 :
",bg="#00ffcc",font="Times 11 bold")
dire.grid(row=7,column=1)
```

```
entry7=Entry(win,textvariable=g1,justify='center')
entry7.grid(row=7,column=6)
```

```
entry8=Entry(win,textvariable=h1,justify='center')
entry8.grid(row=7,column=8)
```

```
dire=Label(win,text="Subject 5 :
",bg="#00ffcc",font="Times 11 bold")
dire.grid(row=8,column=1)
```

```
entry9=Entry(win,textvariable=i1,justify='center')
entry9.grid(row=8,column=6)
```

```
entry10=Entry(win,textvariable=j1,justify='center')
entry10.grid(row=8,column=8)
```

```
dire=Label(win,text="Subject 6 :
",bg="#00ffcc",font="Times 11 bold")
dire.grid(row=9,column=1)
```

```
entry11=Entry(win,textvariable=k1,justify='center')
```

```
entry11.grid(row=9,column=6)
```

```
entry12=Entry(win,textvariable=l1,justify='center')  
entry12.grid(row=9,column=8)
```

```
button1=ttk.Button(win,text="Find  
Tgpa",command=tgpa1)  
button1.grid(row=6,column=14)
```

```
dire5=StringVar()
```

```
dire1=Label(win,text="Tgpa1:",bg="#00ffcc",font="Times  
11 bold")  
dire1.grid(row=8,column=12)  
dire1=Label(win,textvariable=dire5,bg="#00ffcc")  
dire1.grid(row=8,column=14)
```

```
lab=Label(win,text="      ",bg="#00ffcc")  
lab.grid(row=10,column=1)  
dire=Label(win,text="SEMESTER-2",font="Helvetica 15  
bold",bg="#00cca3")  
dire.grid(row=10,column=6)
```

```
dire=Label(win,text="Marks  ",bg="#00ffcc",font="Times  
11 bold")  
dire.grid(row=11,column=6)  
dire=Label(win,text="Credit  ",bg="#00ffcc",font="Times  
11 bold")  
dire.grid(row=11,column=8)
```

```
dire=Label(win,text="Subject 1 :  
",bg="#00ffcc",font="Times 11 bold")  
dire.grid(row=12,column=1)
```

```
entry1=Entry(win,textvariable=a2,justify='center')  
entry1.grid(row=12,column=6)
```

```
entry2=Entry(win,textvariable=b2,justify='center')  
entry2.grid(row=12,column=8)
```

```
dire=Label(win,text="Subject 2 :  
",bg="#00ffcc",font="Times 11 bold")  
dire.grid(row=13,column=1)
```



```
entry3=Entry(win,textvariable=c2,justify='center')
entry3.grid(row=13,column=6)
```

```
entry4=Entry(win,textvariable=d2,justify='center')
entry4.grid(row=13,column=8)
```

```
dire=Label(win,text="Subject 3 :
",bg="#00ffcc",font="Times 11 bold")
dire.grid(row=14,column=1)
```

```
entry5=Entry(win,textvariable=e2,justify='center')
entry5.grid(row=14,column=6)
```

```
entry6=Entry(win,textvariable=f2,justify='center')
entry6.grid(row=14,column=8)
```

```
dire=Label(win,text="Subject 4 :
",bg="#00ffcc",font="Times 11 bold")
dire.grid(row=15,column=1)
```

```
entry7=Entry(win,textvariable=g2,justify='center')
entry7.grid(row=15,column=6)
```

```
entry8=Entry(win,textvariable=h2,justify='center')
entry8.grid(row=15,column=8)
```

```
dire=Label(win,text="Subject 5 :
",bg="#00ffcc",font="Times 11 bold")
dire.grid(row=16,column=1)
```

```
entry9=Entry(win,textvariable=i2,justify='center')
entry9.grid(row=16,column=6)
```

```
entry10=Entry(win,textvariable=j2,justify='center')
entry10.grid(row=16,column=8)
```

```
dire=Label(win,text="Subject 6 :
",bg="#00ffcc",font="Times 11 bold")
dire.grid(row=17,column=1)
```

```
entry11=Entry(win,textvariable=k2,justify='center')
entry11.grid(row=17,column=6)
```

```
entry12=Entry(win,textvariable=l2,justify='center')
entry12.grid(row=17,column=8)
```

```
button1=ttk.Button(win,text="Find
Tgpa",command=tgpa2)
button1.grid(row=14,column=14)
```

```
dire4=StringVar()
```

```
dire1=Label(win,text="Tgpa2:",bg="#00ffcc",font="Times
11 bold")
dire1.grid(row=16,column=12)
dire1=Label(win,textvariable=dire4,bg="#00ffcc")
dire1.grid(row=16,column=14)
```

```
lab=Label(win,text="      ",bg="#00ffcc")
lab.grid(row=18,column=6)
```

```
dire=Label(win,text="Name : ",bg="#00ffcc",font="Times
11 bold")
dire.grid(row=19,column=1)
```

```
entry14=Entry(win,textvariable=name1,justify='center')
entry14.grid(row=19,column=6)
```

```
dire=Label(win,text="Reg No :
",bg="#00ffcc",font="Times 11 bold")
dire.grid(row=21,column=1)
```

```
entry15=Entry(win,textvariable=reg1,justify='center')
entry15.grid(row=21,column=6)
```

```
lab=Label(win,text="      ",bg="#00ffcc")
lab.grid(row=22,column=6)
```

```
button1=ttk.Button(win,text="Find
cgpa",width=10,command=cgpa)
button1.grid(row=23,column=4)
```

```
dire=StringVar()
```

```
dire1=Label(win,text="Cgpa : ",bg="#00ffcc",font="Times
11 bold")
dire1.grid(row=23,column=6)
```

```
dire1=Label(win,textvariable=dire,bg="#00ffcc")
dire1.grid(row=23,column=8)
```

```
dire6=StringVar()
```

```
dire1=Label(win,text="Remark :
",bg="#00ffcc",font="Times 11 bold")
dire1.grid(row=25,column=4)
dire1=Label(win,textvariable=dire6,bg="#00ffcc")
dire1.grid(row=25,column=6)
```

```
lab=Label(win,text="      ",bg="#00ffcc")
lab.grid(row=26,column=6)
```

```
button_reset=ttk.Button(win,text="Reset All The
Data",width=20,command=reset)
button_reset.grid(row=27,column=6)
```

```
lab=Label(win,text="      ",bg="#00ffcc")
lab.grid(row=28,column=6)
```

```
entry14=Entry(win,textvariable=search1,justify='center')
entry14.grid(row=29,column=6)
```

```
button1=ttk.Button(win,text="Search",width=10,comman
d = search)
button1.grid(row=29,column=7)
```

```
pos6 = Label(win,text =
"Name:",bg="#00ffcc",font="Times 11 bold")
pos6.grid(row=30,column=1)
pos6=Label(win,textvariable=pos1,bg="#00ffcc")
pos6.grid(row=30,column=6)
```

```
pos7 = Label(win,text = "Id:",bg="#00ffcc",font="Times 11
bold")
pos7.grid(row=31,column=1)
pos7=Label(win,textvariable=pos2,bg="#00ffcc")
pos7.grid(row=31,column=6)
```

```
pos8 = Label(win,text =
```

```
"Tgpa1:",bg="#00ffcc",font="Times 11 bold")  
pos8.grid(row=32,column=1)  
pos8=Label(win,textvariable=pos3,bg="#00ffcc")  
pos8.grid(row=32,column=6)
```

```
pos9 = Label(win,text =  
"Tgpa2:",bg="#00ffcc",font="Times 11 bold")  
pos9.grid(row=33,column=1)  
pos9=Label(win,textvariable=pos4,bg="#00ffcc")  
pos9.grid(row=33,column=6)
```

```
pos10 = Label(win,text =  
"Cgpa:",bg="#00ffcc",font="Times 11 bold")  
pos10.grid(row=34,column=1)  
pos10=Label(win,textvariable=pos5,bg="#00ffcc")  
pos10.grid(row=34,column=6)
```

```
win.mainloop()
```

# Screenshots of gui

Cgpa Calculator

—

×

SEMESTER-1

Marks

Credit

Subject 1 :

Subject 2 :

Subject 3 :

Subject 4 :

Subject 5 :

Subject 6 :

Find Tgpa

Tgpa1:

SEMESTER-2

Marks

Credit

Subject 1 :

Subject 2 :

Subject 3 :

Subject 4 :

Subject 5 :

Subject 6 :

Find Tgpa

Tgpa2:

Name :

Reg No :

Find cgpa

Cgpa :

Remark :

Reset All The Data

Search

Name:

Id:

Tgpa1:

Tgpa2:

Cgpa:

# GUI AS A WHOLE

## RESULT

Through this project the students would be able to easily calculate their tgpas and cgpa in a faster and easier manner and as developers we were also able to learn the features of python programming language and also understand the working of the graphical user interface how to use the tkinter library to make one and the SQLite3 for the database which act as a storage system in the backend for our project.

■ We were also able to come up with certain new ideas which we have implemented in our project. By making this project we have explored the vast field of python programming language and were able to use our theoretical knowledge in a practical project which has helped us in understanding the concepts far better than any bookish knowledge.

Cgpa Calculator

SEMESTER-1

	Marks	Credit	
Subject 1 :	50	100	
Subject 2 :	42	100	
Subject 3 :	12	100	Find Tgpa
Subject 4 :	85	100	
Subject 5 :	70	100	Tgpa1: 4.75
Subject 6 :	12	100	

SEMESTER-2

	Marks	Credit	
Subject 1 :	98	100	
Subject 2 :	85	100	
Subject 3 :	90	100	Find Tgpa
Subject 4 :	82	100	
Subject 5 :	87	100	Tgpa2: 9.44
Subject 6 :	96	100	

Name : abc

Reg No : 12204319

Find cgpa

Cgpa : 7.1

Remark : Good Keep Going [B+]

Reset All The Data

Search

Name:

Id:

Tgpa1:

Tgpa2:

Cgpa:

# REFERENCE

## Books:

- Introduction to Programming Using Python - Y. Liang
- Head-First Python, 2nd edition
- Python Cookbook

## Websites:

- <https://stackoverflow.com>
- <https://www.geeksforgeeks.org/python-programming-language>
- [https://www.tutorialspoint.com/python/python\\_gui\\_programming.htm](https://www.tutorialspoint.com/python/python_gui_programming.htm)



