AKTU B.Tech Second Year

PYTHON PROGRAMMING

Unit 5

Python file operations

LEC-5

Today's Target

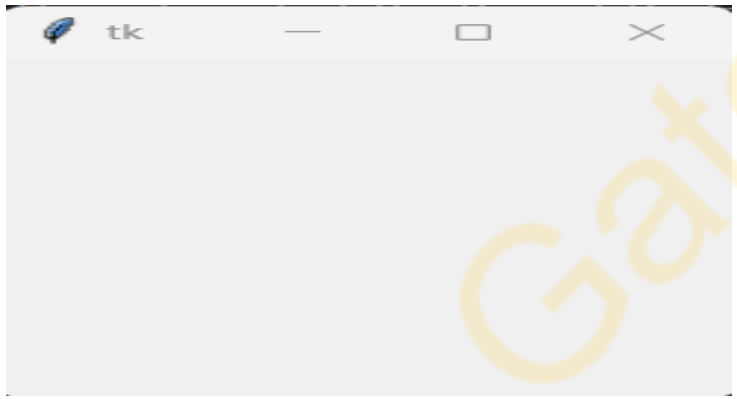
- **Tkinter**
- **module**

By
Pragya Rajvanshi

Gateway classes

Tkinter

- Python Tkinter is a standard GUI (Graphical User Interface) library for Python which provides a fast and easy way to create desktop applications.
- Tkinter provides a variety of widgets like buttons, labels, text boxes, menus and more that can be used to create interactive user interfaces.



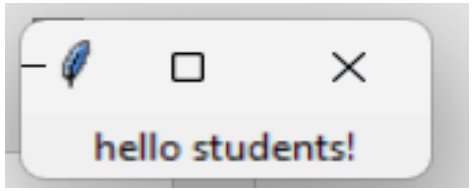

```
import tkinter
m = tkinter.Tk()
'''
widgets are added here
'''
m.mainloop()
```

Gateway classes

Tkinter Widget

- **1. Label**

```
from tkinter import *
root = Tk()
w = Label(root, text='hello students!')
w.pack()
root.mainloop()
```



Gateway classes

Tkinter Widget

```
from tkinter import *
root = Tk()
w = Label(root, text="hello", font=("Arial", 45, "bold"), fg="red",
          bg="yellow",width=20, height=2)
w.pack()
root.mainloop()
```



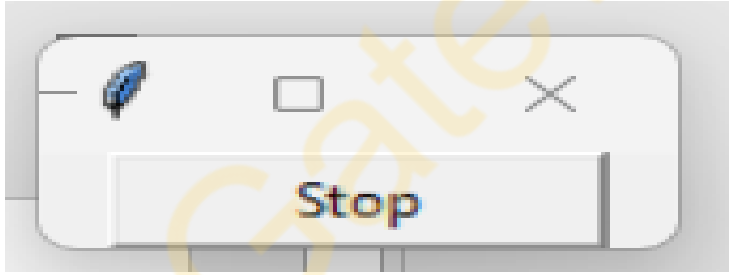
Gateway Classes

Tkinter Widget

➢ 1. Button

a **Button** is a widget that users can click to perform an action, such as closing a window, running a function, or opening another window

```
import tkinter as tk
r = tk.Tk()
r.title('Counting Seconds')
button = tk.Button(r, text='Stop', width=12, command=r.destroy)
button.pack()
r.mainloop()
```



Gateway classes

Tkinter Widget

```
import tkinter as tk
r = tk.Tk()
r.title('Counting Seconds')
button = tk.Button(r, text='Stop', width=12, command=r.destroy,
                   fg="white", bg="blue",font=("Arial", 14, "bold"))
button.pack()
r.mainloop()
```



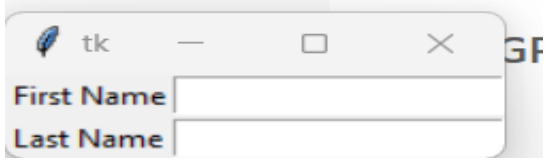
Gateway classes

3.Entry

The **Entry** widget in Tkinter is used to create a single-line text input field where users can type and edit text. It's commonly used in forms or dialogs to capture user input.

Entry(master, options)

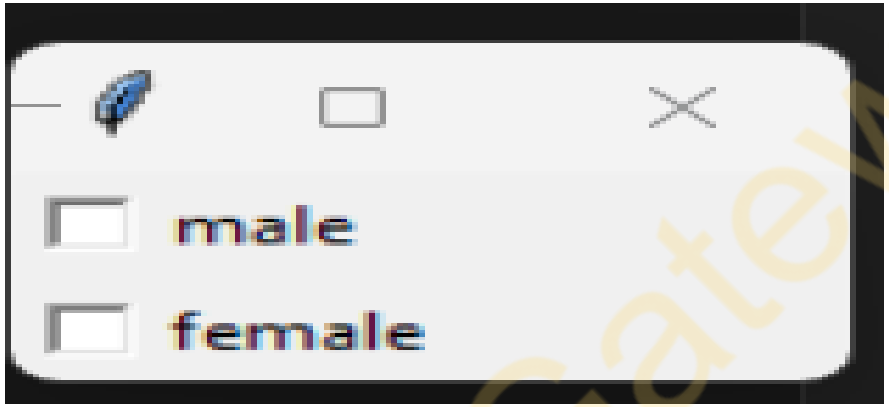
```
from tkinter import *
master = Tk()
Label(master, text='First Name').grid(row=0)
Label(master, text='Last Name').grid(row=1)
e1 = Entry(master)
e2 = Entry(master)
e1.grid(row=0, column=1)
e2.grid(row=1, column=1)
mainloop()
```



Gateway classes

Checkbutton

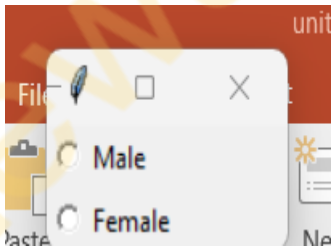
```
from tkinter import *
master = Tk()
var1 = IntVar()
Checkbutton(master, text='male', variable=var1).grid(row=0, sticky=W)
var2 = IntVar()
Checkbutton(master, text='female', variable=var2).grid(row=1, sticky=W)
mainloop()
```



Gateway classes

Radiobutton

```
from tkinter import *
master = Tk()
gender = IntVar()
Radiobutton(master, text='Male', variable=gender, value=1).grid(row=0, sticky=W)
Radiobutton(master, text='Female', variable=gender, value=2).grid(row=1, sticky=W)
mainloop()
```



Gateway classes

What is python module? How to create and import a module in python?

A module in Python is a file containing Python functions, classes, or variables that can be used in other programs. It helps in organizing and reusing code. A module typically has a .py extension.

How to Create a Python Module?

main.py | mymodule.py

```python
def greet(name):
    return f"Hello, {name}!"

pi = 3.14159
```

main.py | mymodule.py

```python
from mymodule import greet, pi
print(greet("rani"))
print(pi)
```

```
Hello, rani!
3.14159
```

Gateway classes

Different ways to import module

1. Import specific functions/variables

```
from mymodule import greet, pi
print(greet("Bob"))
print(pi)
```

2. Import with an alias

```
import mymodule as mm
print(mm.greet("Charlie"))
print(mm.pi)
```

3.Import everything(*)

```
from mymodule import *
print(greet("David"))
print(pi)
```

Gateway classes

Significance of module in python

1. Code Reusability

Instead of rewriting the same code, you can import and use existing modules

2. Better Code Organization

Helps divide large programs into smaller, manageable files.

3. Avoiding Code Duplication

Saves time by defining functions once and reusing them across different projects.

4.Built-in Modules

Python provides many useful built-in modules like math

Gateway classes

1. Import library

- **Imports the entire module** but requires **module name** when accessing functions or variables.

```
import math
print(math.sqrt(25))
```

- Avoids **name conflicts** (functions/variables are accessed with the module name).
- Makes code more **readable and organized**.
- Need to type **module name** every time

from library import *

Imports everything from the module **directly** into the current namespace (no need to use the module name).

```
from math import *
print(sqrt(25))
```

- Shorter and easier to use functions directly.
- **Risk of name conflicts** (e.g., if another module has sqrt(), it may overwrite).
- Reduces **code clarity** (harder to track which function belongs to which module).

Gateway Classes

Built in function matplotlib

Plot(): used to create line plots.

Bar(): used to create bar chart.

Pie(): used to create pie chart.

hist(): used to create histogram.

Scatter(): use to create scatter plot

Title(): add title to the plot.

Legend(): add a legend.

xlabels() and ylabels(): add label to the axis.

Gateway classes



Gateway Classes