Simple GUI Development
Event-driven programming

• Event-driven: GUI programs respond to actions regardless of the order in which they occur. Like, a web browser

• Bind *events* with *event handlers*

  event handler: code that runs when the events occur

By defining all of your **objects**, **events**, and **event handlers**, you establish how your program works.
Root window

```
from tkinter import *

root = Tk()
root.title("simple GUI")
root.geometry("200x100")

root.mainloop()
```
Create a frame

• A Frame is a widget that can hold other widgets

• A Frame is used as a base on which to place other things

app = Frame(root)
app.grid()
Buttons

- A Button widget can be activated by the user to perform some action.
from tkinter import *

root = Tk()      # create the root window
root.title("Simple GUI")
root.geometry("500x200")

app = Frame(root)
app.grid()

button1 = Button(app, text = "I do nothing!")
button1.grid()

button2 = Button(app)
button2.grid()
button2.configure(text = "Me too!")

button3 = Button(app)
button3.grid()

button3["text"] = "Same here!"

root.mainloop()
Create a GUI using class

- Defining the Application class:

```python
class Application(Frame):
    """ A GUI application with three buttons."""

    def __init__(self, master = None):
        """ Initialize the frame. """
        Frame.__init__(self, master)
        self.grid()
        self.create_widgets()
```
Create a GUI using class

• Define a method to create widgets

```python
def create_widgets(self):
    self.button1 = Button(self, text = "I do nothing!")
    self.button1.grid()

    self.button2 = Button(self)
    self.button2.grid()
    self.button2.configure(text = "Me too!")

    self.button3 = Button(self)
    self.button3.grid()
    self.button3["text"] = "Same here!"
```
Binding widgets and event handlers

• Binding the event handler

```python
def createWidgets(self):
    self.button1 = Button(self)
    self.button1["text"] = "Total clicks: 0"
    self.button1["command"] = self.update_count
    self.button1.grid()
```
Binding widgets and event handlers

• Create the event handler

```python
def update_count(self):
    self.button_clicks += 1
    self.button1["text"] = "Total clicks: " + str(self.button_clicks)
```
Click counter

```python
from tkinter import *

class Application(Frame):
    def __init__(self, master=None):
        Frame.__init__(self, master)
        self.button_clicks = 0
        self.grid()
        self.createWidgets()

    def createWidgets(self):
        self.button1 = Button(self)
        self.button1['text'] = "Total clicks: 0"
        self.button1['command'] = self.update_count
        self.button1.grid()

    def update_count(self):
        self.button_clicks += 1
        self.button1['text'] = "Total clicks: " + str(self.button_clicks)

root = Tk()
root.title("Click counter")
root.geometry("300x200")
app = Application(master=root)
app.mainloop()
```
Click counter

Click the button
One time