Week 1: Introduction to Tkinter and Basic Widgets

Day 1: Introduction to Tkinter

- **Objective:** Understand what Tkinter is and how to set up a basic window.
- **Notes:**
    - Tkinter is the standard GUI library for Python.
    - It is included with standard Python distributions.
    - A Tkinter application consists of a main window, which can contain multiple widgets.
- **Snippet:**

  python
  Copy code

  ```
  import tkinter as tk

  root = tk.Tk()  # Create the main window
  root.title("My First Tkinter Window")
  root.geometry("400x300")  # Set the window size

  root.mainloop()  # Run the application
  ```

Day 2: Settling Widgets in the Window's Interior

- **Objective:** Learn how to place widgets within the window.
- **Notes:**
    - Widgets are the building blocks of a Tkinter application.
    - Common widgets include labels, buttons, and text fields.
- **Snippet:**

  python
  Copy code

  ```
  import tkinter as tk

  root = tk.Tk()
  root.title("Widget Placement Example")
  root.geometry("400x300")

  label = tk.Label(root, text="Hello, Tkinter!")
  label.pack()  # Pack the label into the window

  root.mainloop()
  ```

Week 2: Numeric, Boolean, and Selection Widgets

Day 3: Numeric Widgets

- **Objective:** Understand and use numeric widgets like Spinbox.
- **Notes:**
    - Numeric widgets allow users to input numeric values.
    - The Spinbox widget lets users select from a range of values.
- **Snippet:**

  python
  Copy code

```
import tkinter as tk

root = tk.Tk()
root.title("Spinbox Example")
root.geometry("200x100")

spinbox = tk.Spinbox(root, from_=0, to=10)
spinbox.pack()

root.mainloop()
```

Day 4: Boolean Widgets

- **Objective**: Learn about boolean widgets such as Checkbutton.
- **Notes**:
  - Boolean widgets represent true/false values.
  - The Checkbutton widget can be toggled on or off.
- **Snippet**:

  python
  Copy code

```
import tkinter as tk

root = tk.Tk()
root.title("Checkbutton Example")
root.geometry("200x100")

check_var = tk.BooleanVar()
checkbutton = tk.Checkbutton(root, text="Check me", variable=check_var)
checkbutton.pack()

root.mainloop()
```

Day 5: Selection Widgets

- **Objective**: Use selection widgets like Radiobutton and Listbox.
- **Notes**:
  - Selection widgets allow users to choose from a set of options.
  - The Radiobutton widget lets users select one option from a set.
  - The Listbox widget displays a list of items.
- **Snippet**:

  python
  Copy code

```
import tkinter as tk

root = tk.Tk()
root.title("Selection Widgets Example")
root.geometry("300x200")

radio_var = tk.StringVar()
radiobutton1 = tk.Radiobutton(root, text="Option 1", variable=radio_var, value="1")
radiobutton2 = tk.Radiobutton(root, text="Option 2", variable=radio_var, value="2")
radiobutton1.pack()
radiobutton2.pack()

listbox = tk.Listbox(root)
listbox.insert(1, "Item 1")
listbox.insert(2, "Item 2")
listbox.pack()

root.mainloop()
```

Week 3: String Widgets, Date Picker, Color Picker, and Container Widgets

Day 6: String Widgets

- **Objective:** Learn about string widgets like Entry and Text.
- **Notes:**
  - String widgets allow users to input and display text.
  - The Entry widget is a single-line text field.
  - The Text widget is a multi-line text field.
- **Snippet:**

python
Copy code

```
import tkinter as tk

root = tk.Tk()
root.title("String Widgets Example")
root.geometry("300x200")

entry = tk.Entry(root)
entry.pack()

text = tk.Text(root, height=5, width=30)
text.pack()

root.mainloop()
```

Day 7: Date Picker

- **Objective:** Implement a date picker using tkcalendar.
- **Notes:**
  - tkcalendar is a module for Tkinter to pick dates.
  - It can be installed via pip install tkcalendar.
- **Snippet:**

python
Copy code

```
import tkinter as tk
from tkcalendar import Calendar

root = tk.Tk()
root.title("Date Picker Example")
root.geometry("300x300")

cal = Calendar(root, selectmode='day', year=2023, month=6, day=1)
cal.pack()

def print_date():
    print(cal.get_date())

button = tk.Button(root, text="Get Date", command=print_date)
```

```
button.pack()

root.mainloop()
```

Day 8: Color Picker

- **Objective:** Implement a color picker using colorchooser.
- **Notes:**
    - colorchooser module provides a color picker dialog.
- **Snippet:**

python
Copy code

```
import tkinter as tk
from tkinter import colorchooser

root = tk.Tk()
root.title("Color Picker Example")
root.geometry("300x200")

def choose_color():
    color_code = colorchooser.askcolor(title="Choose a color")
    print(color_code)

button = tk.Button(root, text="Pick a Color", command=choose_color)
button.pack()

root.mainloop()
```

Day 9: Container Widgets

- **Objective:** Understand and use container widgets like Frame and LabelFrame.
- **Notes:**
    - Container widgets are used to group other widgets.
    - Frame is a basic container widget.
    - LabelFrame is a container with a label.
- **Snippet:**

python
Copy code

```
import tkinter as tk

root = tk.Tk()
root.title("Container Widgets Example")
root.geometry("400x300")

frame = tk.Frame(root, bg="lightblue", bd=5)
frame.pack(padx=10, pady=10, fill="both", expand=True)

label_frame = tk.LabelFrame(root, text="This is a LabelFrame")
label_frame.pack(padx=10, pady=10, fill="both", expand=True)

label = tk.Label(frame, text="Inside Frame")
label.pack()

label_in_label_frame = tk.Label(label_frame, text="Inside LabelFrame")
label_in_label_frame.pack()
```

```
root.mainloop()
```

Week 4: Buttons, Canvas, and Geometry Management

Day 10: Buttons

- **Objective**: Understand and use different types of buttons.
- **Notes**:
  - Buttons are used to perform actions.
  - Types of buttons include Button, Checkbutton, Radiobutton, and Menubutton.
- **Snippet**:

  python
  Copy code

```python
import tkinter as tk

root = tk.Tk()
root.title("Buttons Example")
root.geometry("300x200")

def on_click():
    print("Button clicked!")

button = tk.Button(root, text="Click Me", command=on_click)
button.pack()

root.mainloop()
```

Day 11: Canvas

- **Objective**: Learn to draw shapes and images using the Canvas widget.
- **Notes**:
  - The Canvas widget is used for drawing shapes, lines, and images.
- **Snippet**:

  python
  Copy code

```python
import tkinter as tk

root = tk.Tk()
root.title("Canvas Example")
root.geometry("400x400")

canvas = tk.Canvas(root, width=400, height=400, bg="white")
canvas.pack()

# Draw shapes
canvas.create_line(10, 10, 200, 200, fill="blue")
canvas.create_rectangle(50, 50, 150, 150, fill="red")
canvas.create_oval(200, 200, 300, 300, fill="green")

root.mainloop()
```

Day 12: Geometry Management

- **Objective**: Learn different geometry managers (pack, grid, place).
- **Notes**:
    - pack: Organizes widgets in blocks before placing them in the parent widget.
    - grid: Organizes widgets in a table-like structure.
    - place: Places widgets at specific positions.
- **Snippet**:

python
Copy code

```
import tkinter as tk

root = tk.Tk()
root.title("Geometry Management Example")
root.geometry("400x300")

# Using pack
label1 = tk.Label(root, text="Pack Example", bg="red")
label1.pack(fill='x')

# Using grid
label2 = tk.Label(root, text="Grid Example", bg="green")
label2.grid(row=1, column=0, padx=20, pady=20)

# Using place
label3 = tk.Label(root, text="Place Example", bg="blue")
label3.place(x=200, y=150)

root.mainloop()
```

Week 5: Binding Functions, Working with Images

Day 13: Binding Function

- **Objective**: Learn how to bind functions to events.
- **Notes**:
    - Events can be mouse clicks, key presses, etc.
    - Use bind method to bind functions to events.
- **Snippet**:

python
Copy code

```
import tkinter as tk

def on_key(event):
    print(f"Key pressed: {event.keysym}")

root = tk.Tk()
root.title("Binding Function Example")
root.geometry("300x200")

root.bind("<Key>", on_key)

root.mainloop()
```

Day 14: Working with Images

- **Objective**: Learn how to display images using the Label widget.
- **Notes**:
    - Use the PhotoImage class for images.
    - Supported formats include GIF and PNG.
- **Snippet**:

python
Copy code

```
import tkinter as tk

root = tk.Tk()
root.title("Image Example")
root.geometry("400x400")

img = tk.PhotoImage(file="path_to_image.png")
label = tk.Label(root, image=img)
label.pack()

root.mainloop()
```