
explorate
GLOBAL
print("please select exactly two objects, the last one gets the modifier unless its not a mesh")
class MirrorX(bpy.types.Operator):
OPERATOR CLASSES
bl_idname = "object.mirror_mirror_x"
bl_label = "Mirror X"
@classmethod
is adds an X mirror to the selected
name = "object.mirror_mirror_x"
bel = "Mirror X"
smethod
oll(cls, context):
eturn context.active_object is not None
mirror_mod.mirror_object
if operation == "MIRROR_X":
mirror_mod.use_x = True
python™
Online Compiler

Create a digital clock by using Tkinter

Code Explanation:

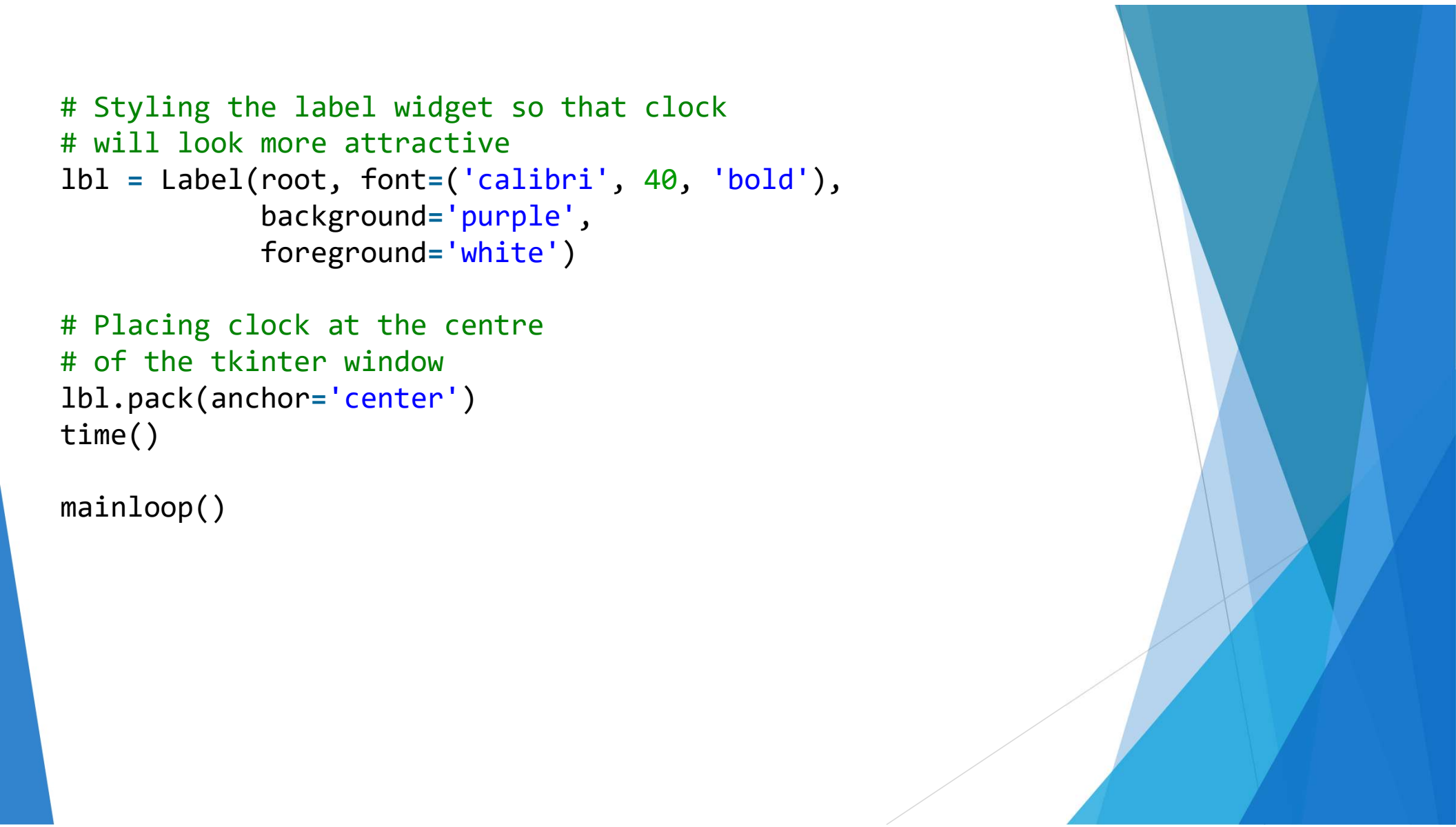
- The code starts by importing the necessary modules.
- The first module is the tkinter library, which provides basic functionality for creating graphical user interfaces (GUIs).
- Next, the strftime function is imported to retrieve system time.
- Next, a window is created and given a title of “Clock.”
- A function called time() is then created to display the current time on the label widget.
- This function uses the strftime() function to format the time string according to system conventions.
- The last part of this code sets up styling for the label widget so that it will look nicer.
- Finally, an instance of Label is created and placed at the center of the window.

- The time() function is executed, and your output should look like this: Clock: Tue Dec 12 08:00:00 2016
- The code creates a window and assigns it the title “Clock”.
- The time() function is then called to display the current time on the label widget.
- The lbl.config() function is used to set the text of the label widget.
- The after() function is used to delay displaying the time for 1000 milliseconds.
- Finally, the style of the label widget is modified with lbl.pack().

Code:

```
# importing whole module
from tkinter import *
from tkinter.ttk import *

# importing strftime function to
# retrieve system's time
from time import strftime

# creating tkinter window
root = Tk()
root.title('Clock')

# This function is used to
# display time on the label

def time():
    string = strftime('%H:%M:%S %p')
    lbl.config(text=string)
    lbl.after(1000, time)
```

```
# Styling the label widget so that clock
# will look more attractive
lbl = Label(root, font=('calibri', 40, 'bold'),
            background='purple',
            foreground='white')

# Placing clock at the centre
# of the tkinter window
lbl.pack(anchor='center')
time()

mainloop()
```



Output: