Graphical User Interface

Part II

Part 2

Canvas

The Canvas is a rectangular area intended for drawing pictures or other complex layouts. You can place graphics, text, widgets, or frames on a Canvas.

Syntax:

```
    w = Canvas ( master, option=value, ... )
```

Parameters:

master: This represents the parent window.

options: Here is the list of most commonly used options for this widget. These options can be used as key-value pairs separated by commas.

The Canvas widget can support the following standard items:

arc: Creates an arc item.

```
coord = 10, 50, 240, 210

arc = canvas.create_arc(coord, start=0, extent=150, fill="blue")
```

image: Creates an image item, which can be an instance of either the BitmapImage or the PhotoImage classes.

```
filename = PhotoImage(file = "sunshine.gif")

image = canvas.create_image(50, 50, anchor=NE, image=filename)
```

line: Creates a line item.

```
line = canvas.create_line(x0, y0, x1, y1, ..., xn, yn, options)
```

oval: Creates a circle or an ellipse at the given coordinates.

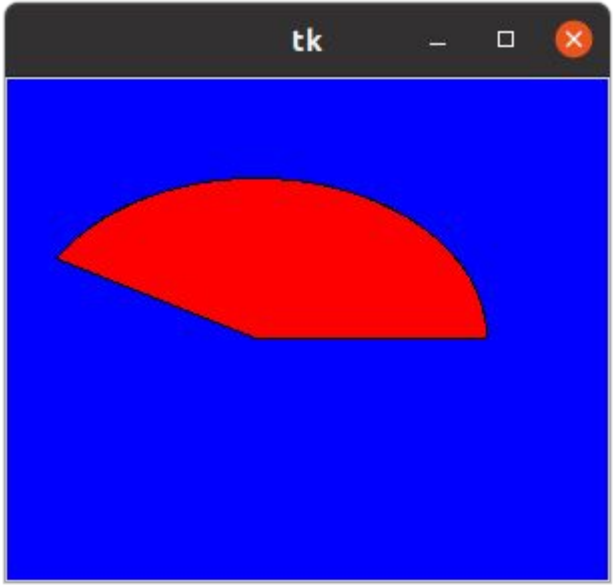
```
oval = canvas.create_oval(x0, y0, x1, y1, options)
```

polygon: Creates a polygon item that must have at least three vertices.

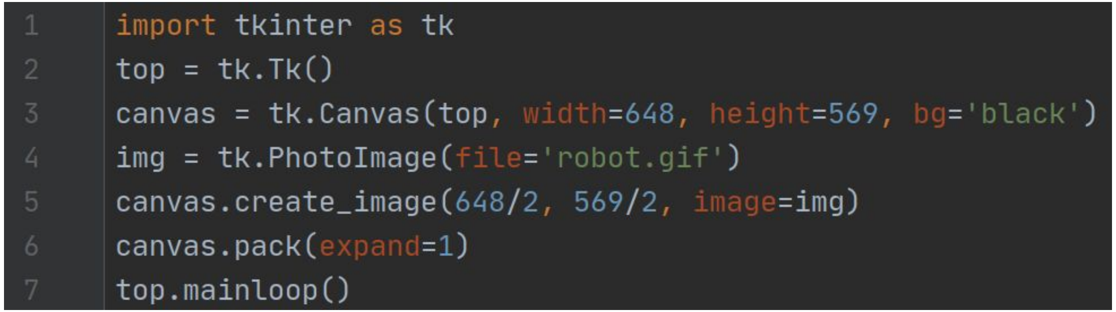
```
oval = canvas.create_polygon(x0, y0, x1, y1,...xn, yn, options)
```

Canvas - Example

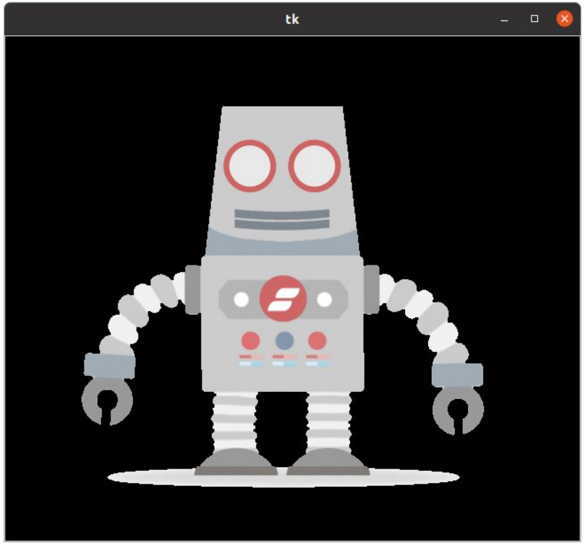
```
import tkinter as tk

top = tk.Tk()

C = tk.Canvas(top, bg="blue", height=250, width=300)

coord = 10, 50, 240, 210

arc = C.create_arc(coord, start=0, extent=150,
fill="red")

C.pack()

top.mainloop()
```



GIF image

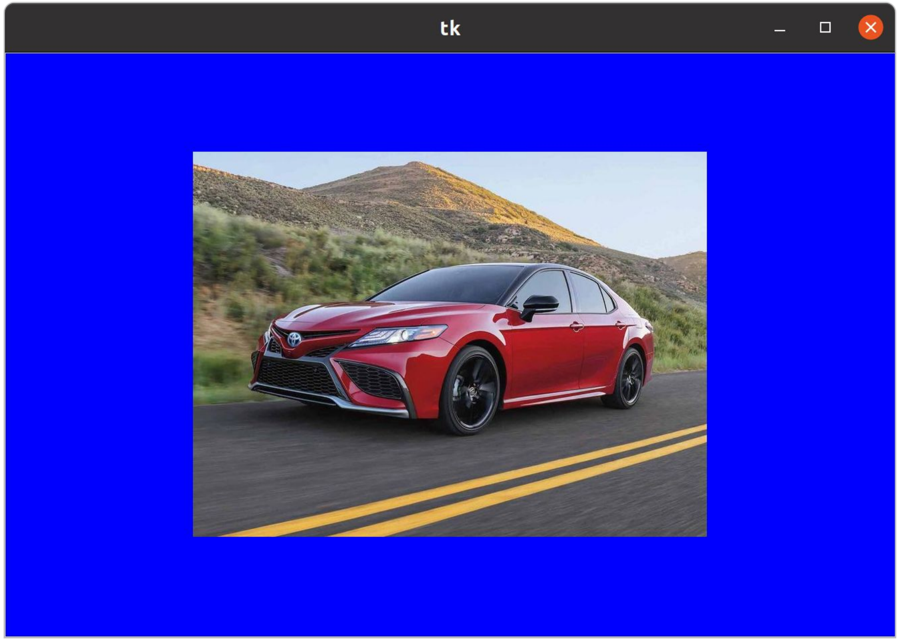


```python
import tkinter as tk
top = tk.Tk()
canvas = tk.Canvas(top, width=648, height=569, bg='black')
img = tk.PhotoImage(file='robot.gif')
canvas.create_image(648/2, 569/2, image=img)
canvas.pack(expand=1)
top.mainloop()
```



JPG image

```
import tkinter as tk
from PIL import Image, ImageTk
top = tk.Tk()
c = tk.Canvas(top, bg='blue', height=872,
width=1329)
load = Image.open('camry.jpg')
img = ImageTk.PhotoImage(load)
c.create_image(1329/2, 872/2, image=img)
c.pack()
top.mainloop()
```



Lab 23-1

Make a python gui program that displays a jpg image on a window using tkinter and canvas.

Submit the captured image of the window you created.

Frame

The Frame widget is very important for the process of grouping and organizing other widgets in a somehow friendly way. It works like a container, which is responsible for arranging the position of other widgets.

It uses rectangular areas in the screen to organize the layout and to provide padding of these widgets. A frame can also be used as a foundation class to implement complex widgets.

Frame

Syntax:

Here is the simple syntax to create this widget:

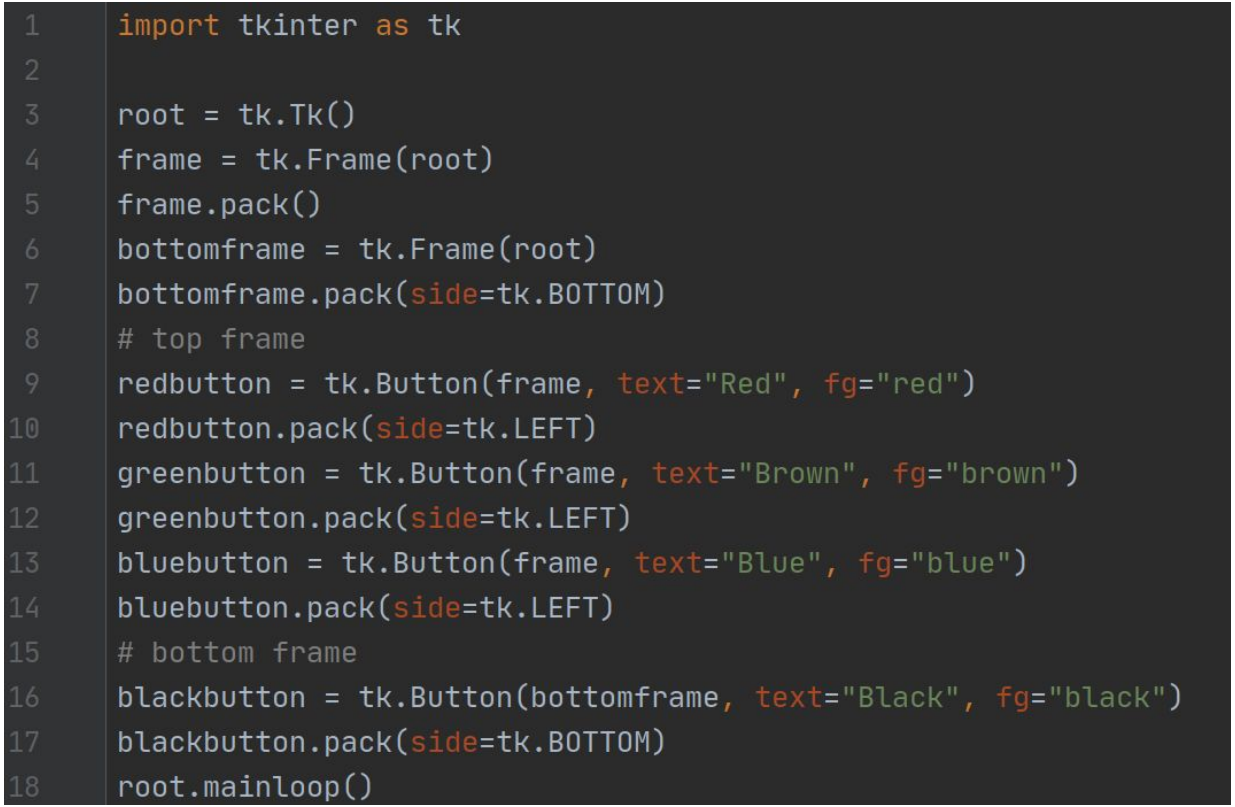
```
w = Frame ( master, option, ... )
```

Parameters:

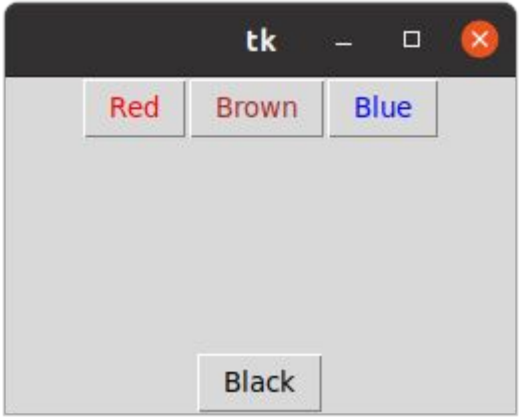
master: This represents the parent window.

options: Here is the list of most commonly used options for this widget. These options can be used as key-value pairs separated by commas.

Example



```python
import tkinter as tk

root = tk.Tk()
frame = tk.Frame(root)
frame.pack()
bottomframe = tk.Frame(root)
bottomframe.pack(side=tk.BOTTOM)
# top frame
redbutton = tk.Button(frame, text="Red", fg="red")
redbutton.pack(side=tk.LEFT)
greenbutton = tk.Button(frame, text="Brown", fg="brown")
greenbutton.pack(side=tk.LEFT)
bluebutton = tk.Button(frame, text="Blue", fg="blue")
bluebutton.pack(side=tk.LEFT)
# bottom frame
blackbutton = tk.Button(bottomframe, text="Black", fg="black")
blackbutton.pack(side=tk.BOTTOM)
root.mainloop()
```



Listbox

The Listbox widget is used to display a list of items from which a user can select a number of items

Syntax:

Here is the simple syntax to create this widget:

```
    w = Listbox ( master, option, ... )
```

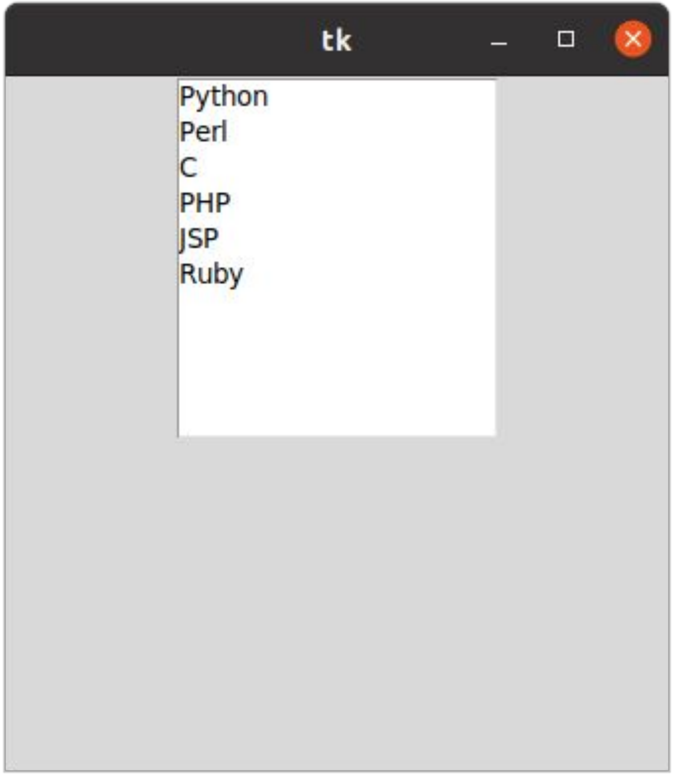
Parameters:

master: This represents the parent window.

options: Here is the list of most commonly used options for this widget. These options can be used as key-value pairs separated by commas.

Example

```
import tkinter as tk
top = tk.Tk()
Lb1 = tk.Listbox(top)
Lb1.insert(1, "Python")
Lb1.insert(2, "Perl")
Lb1.insert(3, "C")
Lb1.insert(4, "PHP")
Lb1.insert(5, "JSP")
Lb1.insert(6, "Ruby")
Lb1.pack()
top.mainloop()
```



Python
Perl
C
PHP
JSP
Ruby

Scrollbar

This widget provides a slide controller that is used to implement vertical scrolled widgets, such as Listbox, Text, and Canvas. Note that you can also create horizontal scrollbars on Entry widgets.

Syntax:

Here is the simple syntax to create this widget:

```
w = Scrollbar ( master, option, ... )
```

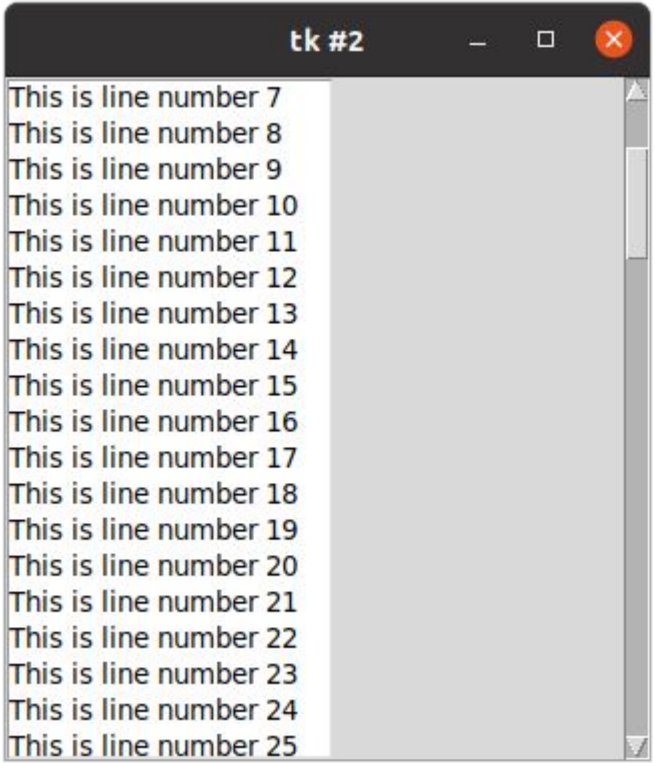
Parameters:

master: This represents the parent window.

options: Here is the list of most commonly used options for this widget. These options can be used as key-value pairs separated by commas.

Example

```
import tkinter as tk

root = tk.Tk()
scrollbar = tk.Scrollbar(root)
scrollbar.pack(side=tk.RIGHT, fill=tk.Y )

mylist = tk.Listbox(root, yscrollcommand = scrollbar.set )
for line in range(100):
    mylist.insert(tk.END, "This is line number " + str(line))

mylist.pack(side=tk.LEFT, fill = tk.BOTH )
scrollbar.config(command=mylist.yview )

root.mainloop()
```



```
This is line number 7
This is line number 8
This is line number 9
This is line number 10
This is line number 11
This is line number 12
This is line number 13
This is line number 14
This is line number 15
This is line number 16
This is line number 17
This is line number 18
This is line number 19
This is line number 20
This is line number 21
This is line number 22
This is line number 23
This is line number 24
This is line number 25
```

Pandastable

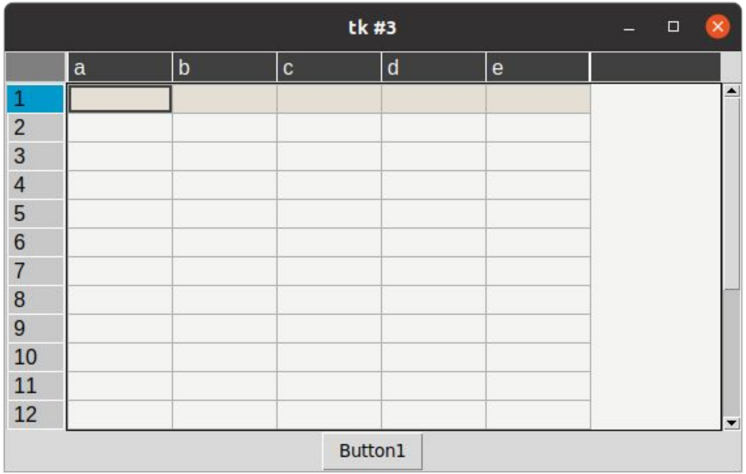
The pandastable library provides a table widget for Tkinter with plotting and data manipulation functionality. It uses the pandas DataFrame class to store table data. Pandas is an open source Python library providing high-performance data structures and data analysis tools. Tkinter is the standard GUI toolkit for python. It is intended for the following uses:

- for python/tkinter GUI developers who want to include a table in their application that can store and process large amounts of data
- for non-programmers who are not familiar with Python or the pandas API and want to use the included DataExplore application to manipulate/view their data
- it may also be useful for data analysts and programmers who want to get an initial interactive look at their tabular data without coding

Pandastable

Create a parent frame and then add the table to it:

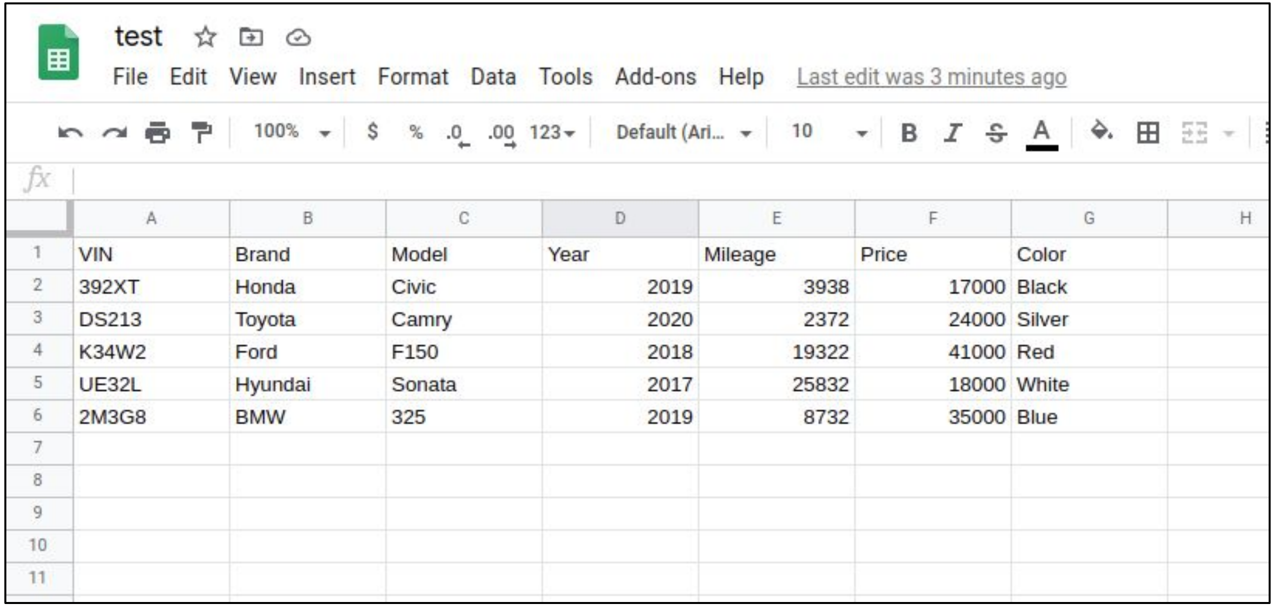
```
from pandastable import Table
import tkinter as tk

root = tk.Tk()
frame = tk.Frame(root)
frame.pack()
pt = Table(frame)
pt.show()
bottomframe = tk.Frame(root)
bottomframe.pack()
b1 = tk.Button(bottomframe, text="Button1", fg="black")
b1.pack()

root.mainloop()
```



	a	b	c	d	e
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					

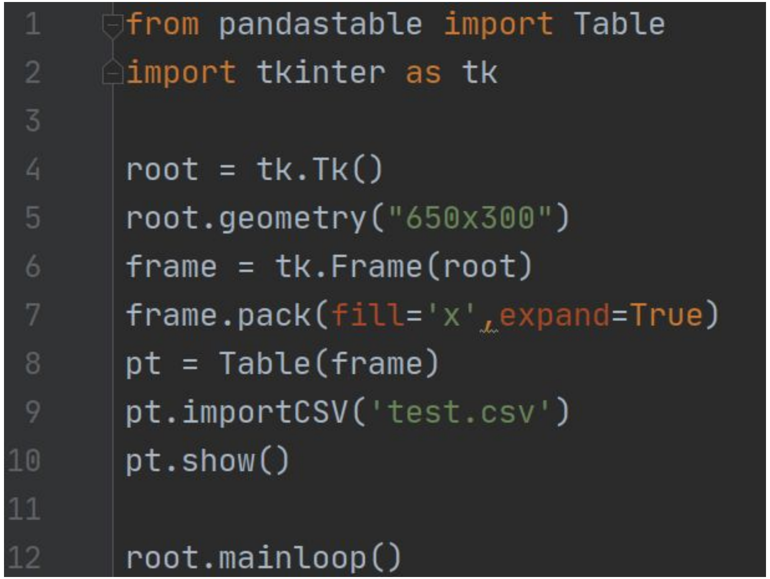
Import from csv file

```
pt.importCSV('test.csv')
```

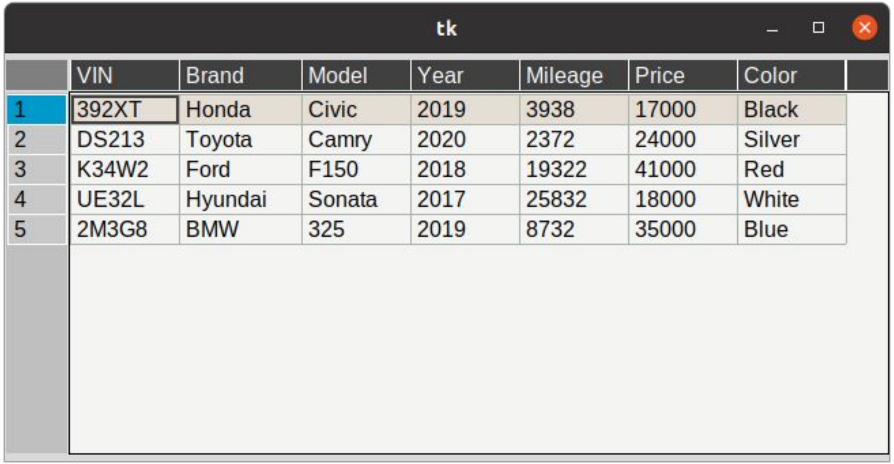


test
File Edit View Insert Format Data Tools Add-ons Help Last edit was 3 minutes ago
100%
$ % .0 .00 123
Default (Ari...
10
B I S A
fx

	A	B	C	D	E	F	G	H
1	VIN	Brand	Model	Year	Mileage	Price	Color	
2	392XT	Honda	Civic	2019	3938	17000	Black	
3	DS213	Toyota	Camry	2020	2372	24000	Silver	
4	K34W2	Ford	F150	2018	19322	41000	Red	
5	UE32L	Hyundai	Sonata	2017	25832	18000	White	
6	2M3G8	BMW	325	2019	8732	35000	Blue	
7								
8								
9								
10								
11								



```python
from pandastable import Table
import tkinter as tk

root = tk.Tk()
root.geometry("650x300")
frame = tk.Frame(root)
frame.pack(fill='x',expand=True)
pt = Table(frame)
pt.importCSV('test.csv')
pt.show()

root.mainloop()
```

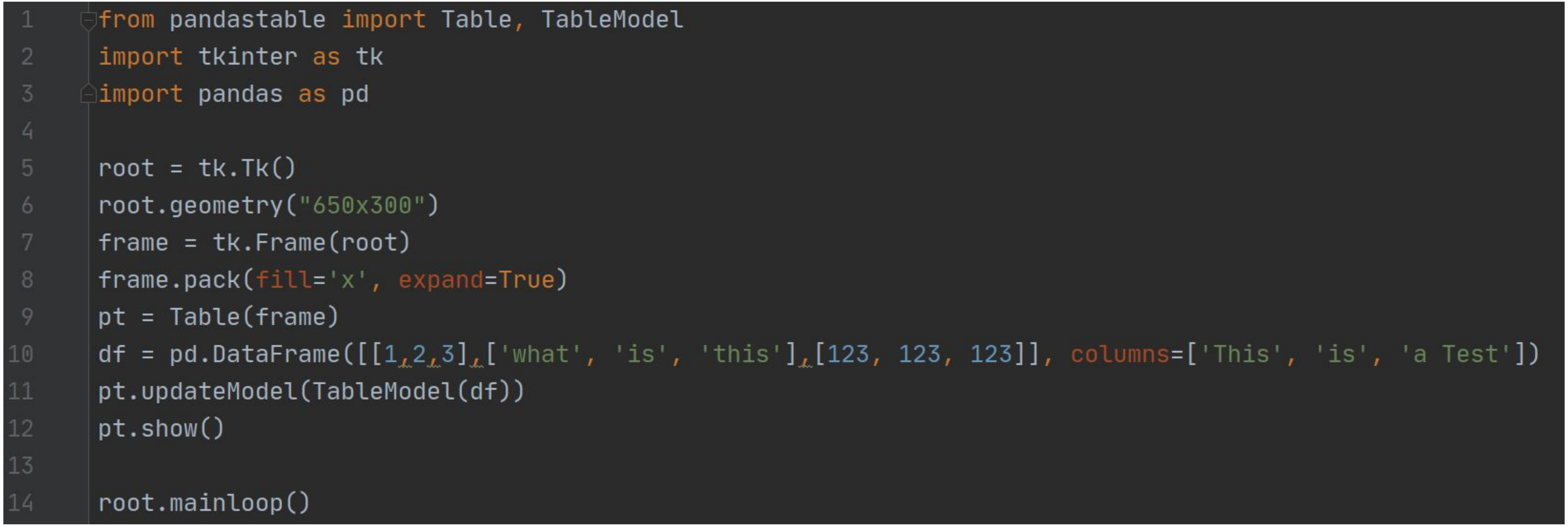


	VIN	Brand	Model	Year	Mileage	Price	Color
1	392XT	Honda	Civic	2019	3938	17000	Black
2	DS213	Toyota	Camry	2020	2372	24000	Silver
3	K34W2	Ford	F150	2018	19322	41000	Red
4	UE32L	Hyundai	Sonata	2017	25832	18000	White
5	2M3G8	BMW	325	2019	8732	35000	Blue

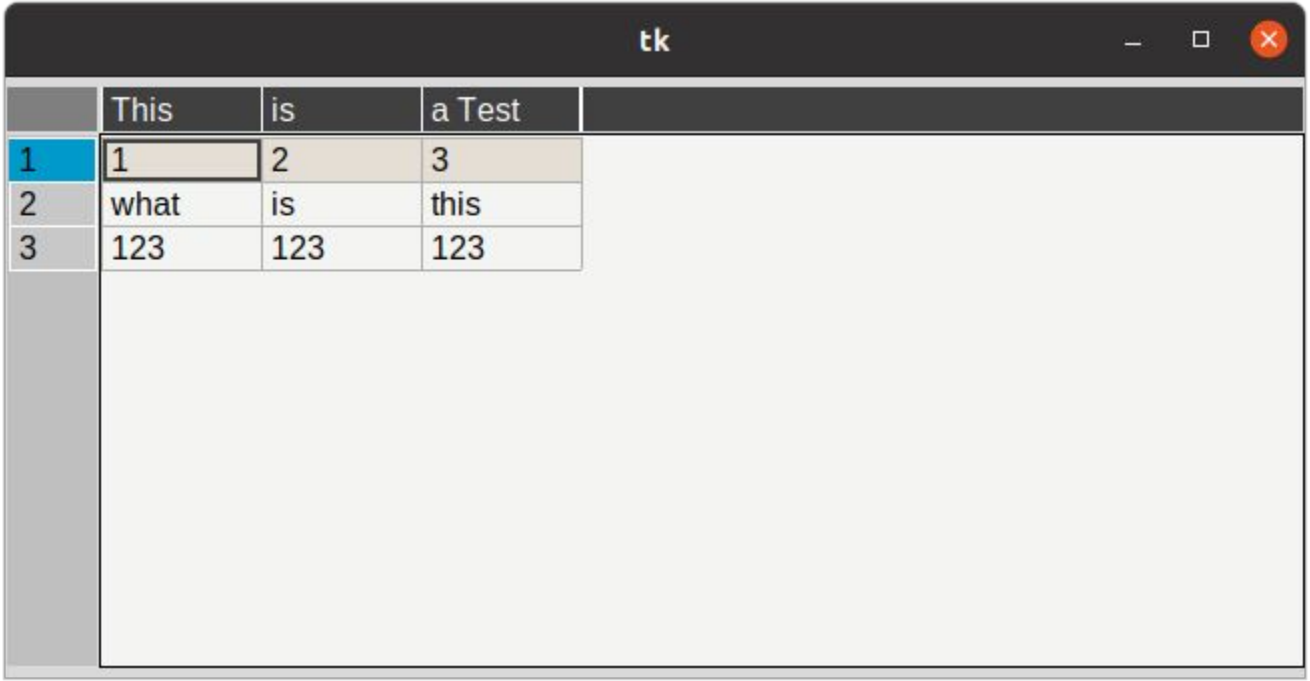
Lab 23-2

Create a Python GUI application that presents a table of cars from your MySQL database like the previous slide above. Please submit both the source code and a screenshot of the created GUI window.

From panda’s dataframe



```
from pandastable import Table, TableModel
import tkinter as tk
import pandas as pd

root = tk.Tk()
root.geometry("650x300")
frame = tk.Frame(root)
frame.pack(fill='x', expand=True)
pt = Table(frame)
df = pd.DataFrame([[1,2,3],['what', 'is', 'this'],[123, 123, 123]], columns=['This', 'is', 'a Test'])
pt.updateModel(TableModel(df))
pt.show()

root.mainloop()
```



	This	is	a Test
1	1	2	3
2	what	is	this
3	123	123	123