University College Cork
Computer Science
Department

Programming Training

Main Points:

- Python Turtle
- Fractals

Topics

This Week:

- Tkinter for GUI Interfaces
- Some examples

Tkinter Overview

- Set of widgets designed by John K. Ousterhout, 1987
- Tkinter == Tool Kit Interface
- Mean to be driven by Tcl (Toolkit Control Language)
- Tkinter is the Python Tcl/Tk Interface
  - Very easy to use with few UI classes
  - 3 ways to manage the layout geometry
  - Simple way to deal with events

Tkinter Overview

- Widgets to work with:

Button	The Button widget is used to display buttons in your application.
Canvas	The Canvas widget is used to draw shapes, such as lines, ovals, polygons and rectangles, in your application.
Checkbutton	The Checkbutton widget is used to display a number of options as checkboxes. The user can select multiple options at a time.
Entry	The Entry widget is used to display a single-line text field for accepting values from a user.
Frame	The Frame widget is used as a container widget to organize other widgets.
Label	The Label widget is used to provide a single-line caption for other widgets. It can also contain images.
Listbox	The Listbox widget is used to provide a list of options to a user.
Menubutton	The Menubutton widget is used to display menus in your application.
Menu	The Menu widget is used to provide various commands to a user. These commands are contained inside Menubutton.

Tkinter Overview

- Widgets to work with:

Message	The Message widget is used to display multiline text fields for accepting values from a user.
Radiobutton	The Radiobutton widget is used to display a number of options as radio buttons. The user can select only one option at a time.
Scale	The Scale widget is used to provide a slider widget.
Scrollbar	The Scrollbar widget is used to add scrolling capability to various widgets, such as list boxes.
Text	The Text widget is used to display text in multiple lines.
Toplevel	The Toplevel widget is used to provide a separate window container.
Spinbox	The Spinbox widget is a variant of the standard Tkinter Entry widget, which can be used to select from a fixed number of values.
PanedWindow	A PanedWindow is a container widget that may contain any number of panes, arranged horizontally or vertically.
LabelFrame	A labelframe is a simple container widget. Its primary purpose is to act as a spacer or container for complex window layouts.

Tkinter Overview

- All widgets have some standard attributes to give the size, color, font, anchor, bitmaps etc:
- Dimension = various width, height, sizes of the widget
  - expressed in pixels
  - width, height = required sizes for the widget
  - padx, pady = extra space the widgets asks for
  - borderwidth = the width of the border rectangle of the widget

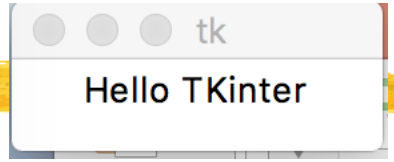
Tkinter Overview

- All widgets have some standard attributes to give the size, color, font, anchor, bitmaps etc:
- Color = attributes for the background, foreground etc
  - color = given as string e.g. “white” of hexa-number “#ffffff”
  - background or bg for the widget background
  - foreground or fg for the widget foreground
  - activebackground and activeforeground
  - selectforeground and disabledforeground

Tkinter Overview

- One geometry manager for the layout to organise widgets with:
  - grid method ➔ widget takes a cell within a grid
  - place method ➔ widget is position to some given coords.
  - pack method ➔ widget is pack in the frame as it comes
- Simple way to work with
  - widget.grid(row = 2, column = 1)
  - widget.pack()
  - widget.place(x=100, y=200)

Tkinter Example



example1.py - /Users/stabirca/Desktop/python/programs - mscim/week5/example1.py (3.5.2)

```
from tkinter import *

root = Tk()
label = Label( root, text = "Hello TKinter")
label.pack()
root.mainloop()
```

Ln: 7 Col: 0

Tkinter Example

Some explanation:

1. tkinter module is imported first.
2. root = instance of Tk to create the UI frame
3. label = instance of Label for the parent root to hold the text Hello Tkinter
4. the label is then packed into root
5. root.mainloop() deals with the frame events

Tkinter - How to?

Simplest way (procedural way):

1. Import the modules needed by the interface.
2. Create the frame as instance of Tk()
3. Create all the components needed in the interface
    1. Construct the component
    2. Setup some of their attributes
    3. Give their position in the layout
4. root.mainloop()

Widgets - Label

Label = box to display/change a text or an image.

Constructor

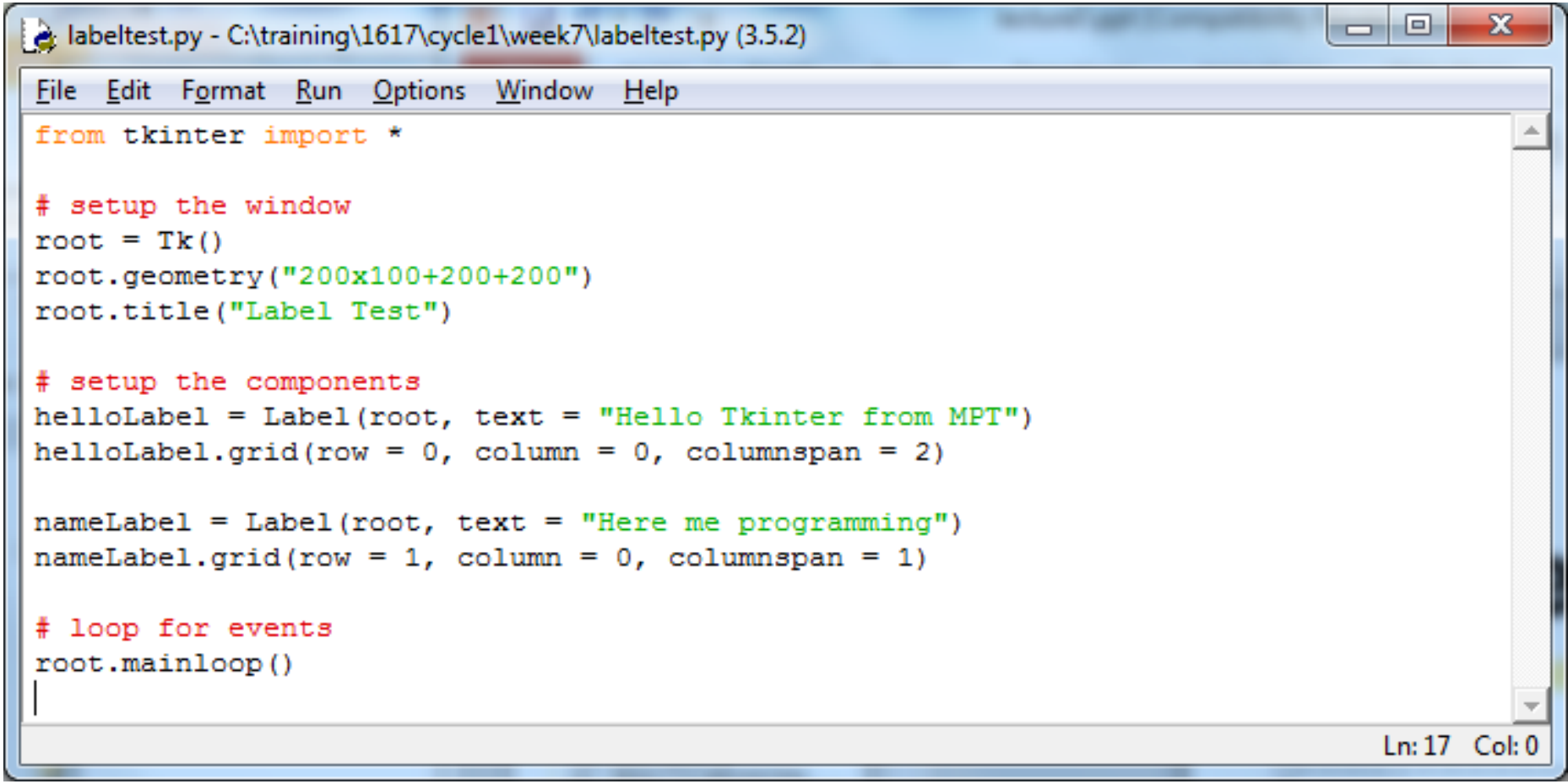
label = Label (master, option, ...)

Options refer to the following:

- dimension, color, etc
- manipulating the text or the image
  - text, image
  - underline
  - textvariable

textvariable is needed for manipulating the label text.

Label


labeltest.py - C:\training\1617\cycle1\week7\labeltest.py (3.5.2)
File Edit Format Run Options Window Help
from tkinter import *
# setup the window
root = Tk()
root.geometry("200x100+200+200")
root.title("Label Test")
# setup the components
helloLabel = Label(root, text = "Hello Tkinter from MPT")
helloLabel.grid(row = 0, column = 0, columnspan = 2)
nameLabel = Label(root, text = "Here me programming")
nameLabel.grid(row = 1, column = 0, columnspan = 1)
# loop for events
root.mainloop()
Ln: 17 Col: 0

Widgets - Entry

Entry = Widget to manipulate a single line of text.

Constructor

entry = Entry (master, option, ...)

Options refer to the following:

- dimension, color, etc
- manipulating the text by textvariable
- show to define an echo character

Various ivar methods for entry text manipulation:

- get() ➔ gets all the characters
- delete(first, last) ➔ deletes from first to last
- insert(index, word) ➔ inserts word at the index position

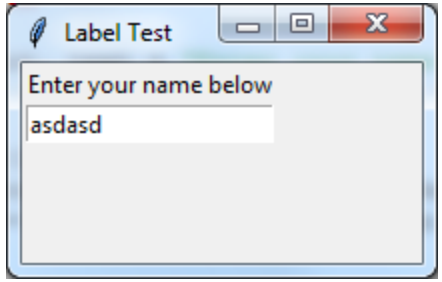
Entry

entrytest.py - C:/training/1617/cycle1/week7/entrytest.py (3.5.2)

File Edit Format Run Options Window Help

```
from tkinter import *

# setup the window
root = Tk()
root.geometry("200x100+200+200")
root.title("Label Test")

# setup the components

nameLabel = Label(root, text = "Enter your name below")
nameLabel.grid(row = 0, column = 0, columnspan = 1)

entryVar = StringVar()
helloEntry = Entry(root, textvariable = entryVar)
helloEntry.grid(row = 1, column = 0, columnspan = 2)

# loop for events
```

Ln: 15 Col: 2



Widgets - Button

Button = display a text or image and to react to a click.

Constructor

button = Button (master, option, ...)

Options refer to the following:

- dimension, color, etc
- command to set the callback function

Two ivar methods for entry text manipulation:

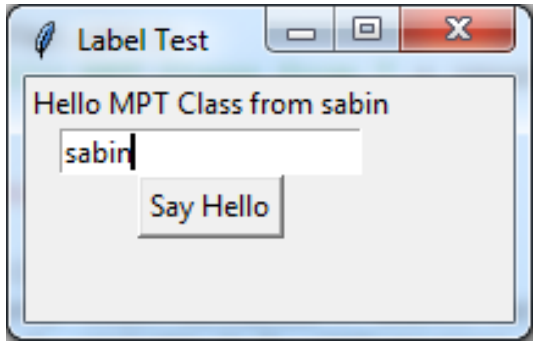
- flash() ➔ button will flash between states
- invoke() ➔ explicit call of the button collback

Button

entrytest.py - C:/training/1617/cycle1/week7/entrytest.py (3.5.2)

File Edit Format Run Options Window Help

```
from tkinter import *

# setup the window
root = Tk()
root.geometry("200x100+200+200")
root.title("Label Test")

def helloCallback():
    yourname = entryVar.get()
    labelVar.set("Hello MPT Class from " + yourname)
#end helloCallback

# setup the components

labelVar = StringVar()
nameLabel = Label(root, textvariable = labelVar)
nameLabel.grid(row = 0, column = 0, columnspan = 1)
labelVar.set("Type in your name")

entryVar = StringVar()
helloEntry = Entry(root, textvariable = entryVar)
helloEntry.grid(row = 1, column = 0, columnspan = 2)

helloButton = Button(root, text = "Say Hello", command = helloCallback)
helloButton.grid(row = 2, column = 0, columnspan = 2)


# loop for events
root.mainloop()
```



Ln: 11 Col: 0

Widgets - Listbox

Listbox = widget to display a (long) list of items from which we can select.

Constructor

list= Listbox(master, option, ...)

Options refer to the following:

- dimension, color, etc
- selectmode gives how we can select the items from list
  - SINGLE ➔ one item to select when clicked
  - BROWSE ➔ one item to select that drags with the mouse
  - MULTIPLE ➔multiple selection by toggling items

Widgets - Listbox

Lots of functions to manage the listbox and its selection.

Manage the list:

- size() ➔ number of items in listbox
- delete(first, last) ➔ delete all items between the specified first and last
- insert(index, *items) ➔ insert the items from index
- etc

Selection

- curselection() ➔ returns the tuple with the selected indices.

Widgets - Listbox

Listbox-es are created usually from array of items using a for loop.

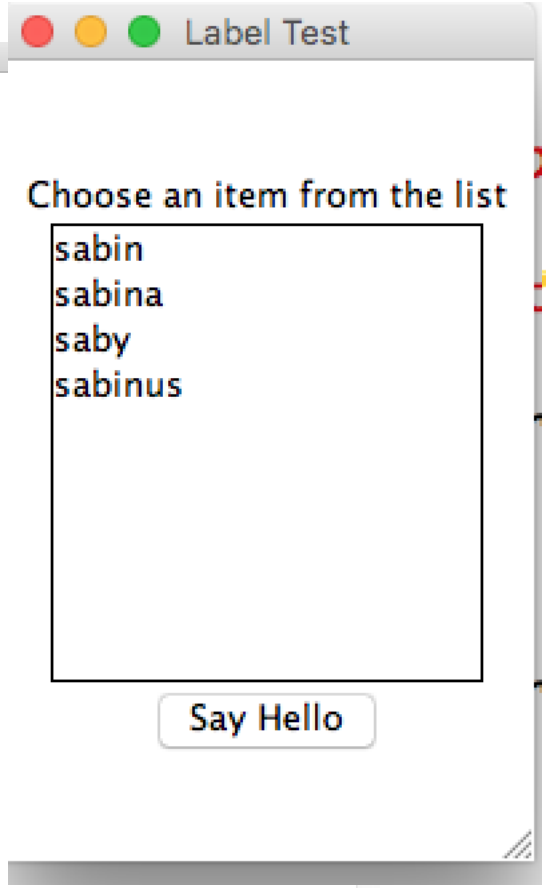
```
listbox = Listbox(root)
items = [“item0”, “item1”, “item2”, “item3”, “item4”]
for item in items:
        listbox.insert(END, item)
```

Listbox-es do not have command however we can bind with it.

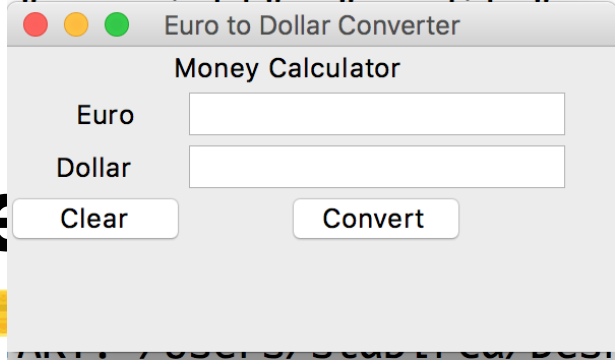
```
listbox.bind("<<ListboxSelect>>", self.onSelect)
```

listtest.py - /Users/stabirca/Desktop/listtest.py (3.5.2)

```
from tkinter import *

# setup the window
root = Tk()
root.geometry("200x300+200+200")
root.title("Label Test")

def helloCallback():
    items = nameList.curselection()[0]
    name = names[ids]
    labelVar.set("you selected " +name)
#end helloCallback

# setup the components
labelVar = StringVar()
nameLabel = Label(root, textvariable = labelVar)
nameLabel.grid(row = 0, column = 0, columnspan = 1)
labelVar.set("Choose an item from the list")

names = ["sabin", "sabina", "saby", "sabinus"]
nameList = Listbox(root)
for i in names:
    nameList.insert(END, i)
nameList.grid(row = 1, column = 0, rowspan = 1, columnspan = 1)

helloButton = Button(root, text = "Say Hello", command = helloCallback)
helloButton.grid(row = 3, column = 0, columnspan = 2)

# loop for events
root.mainloop()
```



Example 1 - Converte



Converter Euro to Dollar Interface:

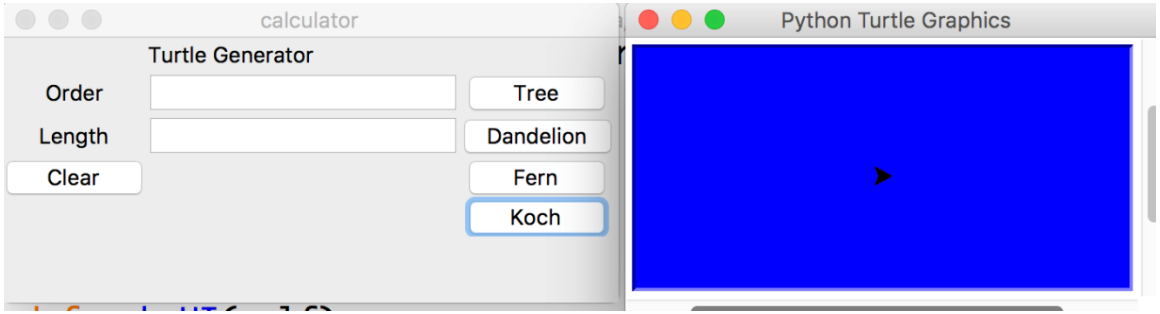
- widgets: 2 Label-s, 2 Entry-s and 2 Button-s
- Additional 2 StringVar to manipulate the Entry text
- All widgets with a simple grid geometry
- Callback functions for the buttons

Example 2 – Turtle Interface

Interface to control drawing Turtle figures.

Widgets to work with:

- ☒2 Label-s, 2 Entry-s for the figure order and length.
- ☒Additional 2 StringVar to manipulate their text.
- ☒A button clear to reset everything.
- ☒Some widgets, maybe buttons, to draw the required fractal



To do List

1. Solve the HW problems.
2. Read more about the Tkinter module