Tkinter cheatsheet (part 1)

Essential Tk commands:

`from Tkinter import *`: This imports the Tkinter library so that you can use it. This is the first line of any Tkinter program

`Tk()`: This creates a `Tk()` root . Everything that you put on the screen, whether it's a text box, button, or image is a widget and they must all be placed the root. You can think of it like a basket that you put all your widgets in. The first real command of a Tkinter program usually consists of creating the root and looks something like this:

```
>>>root = Tk()
```

`root.mainloop()`: This allows the root widget to actually do its thing. It starts something called an event loop. This line should appear at the end of your Tkinter program.

Fitting things into the root:

Once you make an object, you need to actually fit it into the Tkinter window. There are multiple ways to do this.

`foo.pack()`: This is the simplest command. It tells Tkinter to fit the widget foo into the window as a rectangular block. This is often ugly if you try to put multiple objects into the window, but for simple or small windows, `foo.pack()` is your best friend.

`foo.grid(...)`: This command places the widget foo inside of a table that you create. The " . . . " represents the various arguments that `foo.grid()` can take. Much like the range function which we have used for loops and unlike most of the functions you have written so far, foo.grid() can a different number of arguments depending on what you want to do. We will cover these later when we need them.

`foo.place(...)`: This command should be rarely used, if ever. It will put the foo widget at a specific point on the screen.

Basic Tkinter widgets:

Text boxes: The syntax for creating a text box is `Label(location,...)`. The location field tells Tkinter where to put text box. If you want to put it in the root, put root there. If you want to put it in the frame foo, put foo there. You can also add a text field by writing `text = foo`, where foo is a string. This is the proper syntax for accessing any of the additional arguments in any widget.

Example:

```
>>>from Tkinter import *
>>>root = Tk()
>>>CA = Label(root, text = “Phil”)
>>>CA.pack()
>>>root.mainloop()
```

This code will create a text box called "CA" which contains the word "foo" and will place it in the root.

Buttons: The syntax for making a button is `Button(location,...)`. The location field works just like it does for textboxes. The additional fields that we will worry about right now are `text`, which is the string that you want to appear on the button and `command`, which is the name of a function that you want to be called when the button is pressed. For now all the functions that we call will not take parameters.

Example:

```
>>>from Tkinter import *
>>>root = Tk()
>>>def foo():
>>>   print “Rodriguez”
>>>CA = Button(root, text = “Alex”, command = foo)
>>>CA.pack()
>>>root.mainloop()
```

This will make a button called “Alex,” which will print out “Rodriguez” when pressed.

Frames: The syntax for making a fame is Frame(location). A frame is like root in that you can think of it like a basket for widgets, but unlike the root it must first be packed inside of the root or another frame.

Example:

```
>>>from Tkinter import *
>>>root = Tk()
>>>frame1 = Frame(root)
>>>frame1.pack()
>>>frame2 = Frame(root)
>>>frame2.pack()
>>>def foo():
>>>   print “Purta”
>>>CA1 = Button(frame1, text = “Carrie”, command = foo)
>>>CA1.pack()
>>>CA2 = Label(farme2, text = “Rich”)
>>>CA2.pack()
>>>root.mainloop()
```

This will create two frames, one with a textbox called CA2, and the other with a button called CA1.

There are more arguments to all of these widgets than the ones that we have shown you. There are certainly more widgets than just these three. You can find a comprehensive list of all the widgets Tkinter supports and what arguments they can take online at:

http://www.pythonware.com/library/tkinter/introduction/