

Tkinter doesn't suck, honest!

Richard Jones

richardj0n3s@gmail.com

Abstract

I hope to dispel the often-repeated myth that Tkinter just plain sucks. I propose three reasons it does not suck: it's simple, it's always there and it just works. There are two reasons why Tkinter might suck: it doesn't look that pretty and extending is a little tricky. I intend to show that it doesn't have to look that ugly these days with skinning, and that extending is actually quite easy.

1. Introduction

Often the question will appear in an online discussion forum, “what GUI toolkit should I use?” The responses may vary depending on the exact audience, but a common theme I have noticed is that invariably someone will say “whatever you do don't use Tkinter, it sucks!” and a bunch of people will chime in, agreeing.

I hope to dispel this often-repeated myth that Tkinter just plain sucks.

I propose three reasons why it does not, in fact, suck:

1. it's simple to use - a simple user interface may be created in mere minutes,
2. it's always there¹, and
3. it is mature, having been around for over a decade, and just works.

There are a number of good tutorials and references covering Tkinter. Two of those are particularly noteworthy:

<http://effbot.org/tkinterbook>

This is Fredrik Lundh's Tkinter tutorial which covers most of Tkinter at a good level of detail, including covering all of the basic widgets with common usage code examples. It also serves as a handy reference during application development, so keep it bookmarked.

http://www.ferg.org/thinking_in_tkinter/

A Creative Commons work, Thinking in Tkinter consists of a set of Python programs. Each program contains a long documentation string with text that explains certain concepts, followed by executable code that illustrates the concepts.

There is also a well-regarded book on the subject, “Python and Tkinter Programming” by John E Grayson Ph.D.

¹ you may need to install Tkinter manually on some Linux variants

After informally polling of developer acquaintances there are two reasons I was given for why Tkinter may suck:

1. it doesn't look very pretty, and
2. extending it is a little tricky.

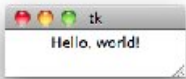

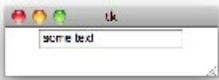

The first reason was certainly the case for many years. Recently (and "recent" encompasses years) Tk has been blessed with the ability to "skin" its widgets. This allows them to look rather like the native widgets users are familiar with. Python 3 has the skinning ability built in. The result is that it's still not the prettiest toolkit, but I believe this is a reasonable tradeoff for its other positive qualities.



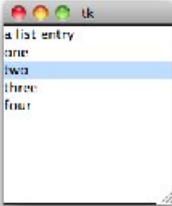

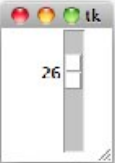
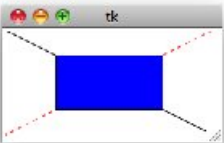
Extending Tkinter is possible and the references I give above cover the detail of creating your own widgets. The toolkit "Python Megawidgets" at <http://pmw.sourceforge.net/> provides high-level compound widgets.

2. A brief tour of Tkinter

The table below lists the basic widget types available in Tkinter. To aid clarity the same application code may be used for each example:

```
import Tkinter as tk
root = tk.Tk()
# code goes here
root.mainloop()
```

widget	"code goes here"
	<pre>tk.Label(root, text="Hello, world").pack()</pre>
	<pre>def pressed(): print 'You pressed me!' tk.Button(root, text='Press me!', command=pressed).pack()</pre>
	<pre>entry = tk.Entry(root) entry.pack() entry.insert(0, 'some text') value = entry.get()</pre>
	<pre>variable = tk.IntVar() tk.Checkbutton(root, text='Checked?', variable=variable).pack() value = variable.get()</pre>

widget	"code goes here"
	<pre>value = tk.IntVar() for n in range(4): tk.Radiobutton(root, value=n, text="Selection %d"% (n+1), variable=value).pack()</pre>
	<pre>value = tk.StringVar(value='One') tk.OptionMenu(root, value, 'One', 'Two', 'Three').pack()</pre>
	<pre>listbox = tk.Listbox(root) listbox.pack() listbox.insert(tk.END, "a list entry") for item in 'one two three four'.split(): listbox.insert(tk.END, item)</pre>
	<pre>text = tk.Text(root) text.pack() text.insert(tk.END, '''some text more text''')</pre>
	<pre>scale = tk.Scale(root, from_=0, to=100) scale.pack()</pre>
	<pre>w = tk.Canvas(root, width=200, height=100) w.pack() w.create_line(0, 0, 200, 100) w.create_line(0, 100, 200, 0, fill="red", dash=(4, 4)) w.create_rectangle(50, 25, 150, 75, fill="blue")</pre>