Tkinter Cheat Sheet

The most popular GUI creation tool for Python, Tkinter provides a number of widgets and methods you can use to create a user interface for your application.

Tkinter Widgets

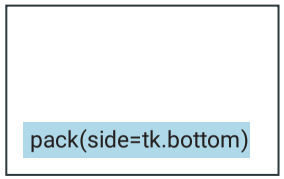
Code	from tkinter import *	from tkinter import * from tkinter.ttk import *

Widgets	Button
	instance = Button(root, text="Click me!", ...)
	
	Checkbutton
	instance = tk.Checkbutton(parent, option, ...)
	
	Entry
	instance = tk.Entry(master, option, ...)
	
	Frame
	instance = Frame(parent, option, ...)
	
	Label
	instance = tk.Label(text="some text")
	
	LabelFrame
	instance = LabelFrame(master, option, ...)
	
	Menubutton
	instance = Menubutton (master, options, ...)
	
	PanedWindow
	instance = PanedWindow(master, options, ...)
	
	Radiobutton
	instance = Radiobutton(master, options, ...)
	
	Scale
	instance = Scale (master, option, ...)
	
	Scrollbar
	instance = Scrollbar (master, options, ...)
		Combobox
		instance = ttk.Combobox(master, option=value, ...)
		
		Notebook
		instance = ttk.Notebook(container, options, ...)
		
		Progressbar
		instance = Progressbar(parent, options, ...)
		
		Separator
		# orient options are 'horizontal' or 'vertical':
		instance = ttk.Separator(container,orient='horizontal')
		
		Sizegrip
		instance = ttk.Sizegrip(master, options, ...)
		
		Treeview
		instance = ttk.Treeview(master, options, ...)

Position Widgets using `pack()`, `place()` or `grid()`

`pack()` organizes widgets in horizontal and vertical boxes that are limited to left, right, top, bottom positions. Each box is offset and relative to each other.

```
root.geometry('200x100')
test = tk.Label(root, text="pack(side=tk.bottom)", bg="teal")
test.pack(side=tk.bottom)
```

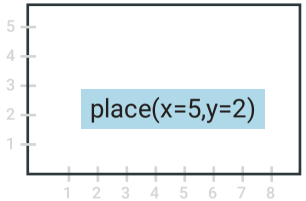


Options:

`padx` pads externally along the x axis
`pady` pads externally along the y axis
`ipadx` pads internally along the x axis
`ipady` pads internally along the y axis

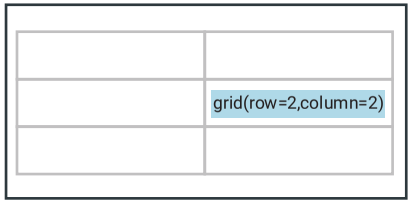
`place()` places widgets in a two dimensional grid using x and y absolute coordinates.

```
root.geometry('200x100')
Label(root, text="place(x=5, y=2)",
bg="#A3DBE0").place(x=5, y=2)
```



`grid()` locates widgets in a two dimensional grid using row and column absolute coordinates.

```
root.geometry('200x100')
Label(root, text="grid(row=2, column=2)",
width=12).grid(row=2, column=2)
```



Tkinter Images with Pillow

```
# Pillow is imported as PIL
from PIL import ImageTk, Image

image1 = Image.open("<path/image_name>")
test = ImageTk.PhotoImage(image1)

label1 = tkinter.Label(image=test)
label1.image = test

# Position image as the background image
label1.place(x=1, y=1)
# Resize image to fit on button
photoimage = photo.subsample(1, 2)
# Position image on button
Button(root, image = photoimage,).pack(side = BOTTOM)
```

Tkinter Cheat Sheet

The most popular GUI creation tool for Python, Tkinter provides a number of widgets and methods you can use to create a user interface for your application.

Tkinter Widgets

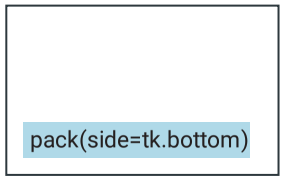
Code	from tkinter import *	from tkinter import * from tkinter.ttk import *

Widgets	Button
	instance = Button(root, text="Click me!", ...)
	
	Checkbutton
	instance = tk.Checkbutton(parent, option, ...)
	
	Entry
	instance = tk.Entry(master, option, ...)
	
	Frame
	instance = Frame(parent, option, ...)
	
	Label
	instance = tk.Label(text="some text")
	
	LabelFrame
	instance = LabelFrame(master, option, ...)
	
	Menubutton
	instance = Menubutton (master, options, ...)
	
	PanedWindow
	instance = PanedWindow(master, options, ...)
	
	Radiobutton
	instance = Radiobutton(master, options, ...)
	
	Scale
	instance = Scale (master, option, ...)
	
	Scrollbar
	instance = Scrollbar (master, options, ...)
		Combobox
		instance = ttk.Combobox(master, option=value, ...)
		
		Notebook
		instance = ttk.Notebook(container, options, ...)
		
		Progressbar
		instance = Progressbar(parent, options, ...)
		
		Separator
		# orient options are 'horizontal' or 'vertical':
		instance = ttk.Separator(container,orient='horizontal')
		
		Sizegrip
		instance = ttk.Sizegrip(master, options, ...)
		
		Treeview
		instance = ttk.Treeview(master, options, ...)

Position Widgets using `pack()`, `place()` or `grid()`

`pack()` organizes widgets in horizontal and vertical boxes that are limited to left, right, top, bottom positions. Each box is offset and relative to each other.

```
root.geometry('200x100')
test = tk.Label(root, text="pack(side=tk.bottom)", bg="teal")
test.pack(side=tk.bottom)
```

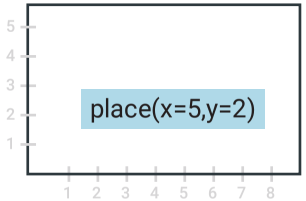


Options:

`padx` pads externally along the x axis
`pady` pads externally along the y axis
`ipadx` pads internally along the x axis
`ipady` pads internally along the y axis

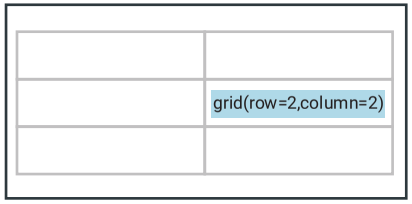
`place()` places widgets in a two dimensional grid using x and y absolute coordinates.

```
root.geometry('200x100')
Label(root, text="place(x=5, y=2)",
bg="#A3DBE0").place(x=5, y=2)
```



`grid()` locates widgets in a two dimensional grid using row and column absolute coordinates.

```
root.geometry('200x100')
Label(root, text="grid(row=2, column=2)",
width=12).grid(row=2, column=2)
```



Tkinter Images with Pillow

```
# Pillow is imported as PIL
from PIL import ImageTk, Image

image1 = Image.open("<path/image_name>")
test = ImageTk.PhotoImage(image1)

label1 = tkinter.Label(image=test)
label1.image = test

# Position image as the background image
label1.place(x=1, y=1)
# Resize image to fit on button
photoimage = photo.subsample(1, 2)
# Position image on button
Button(root, image = photoimage,).pack(side = BOTTOM)
```