

tkinter Journeyman Reference

this doc does not consider ttk | color codes: explanatory, option, example, note, syntax, ↪ = "results in"/"yields"

Import: from tkinter import * (or import tkinter as tk - may require using "tk." prefix to commands)

Establish root: root=Tk() Define root or toplevel Geometry: root.Geometry(str(width)+"x"+str(height)+"+" +str(xoffset)+"+" +str(yoffset)); find screen dimensions with =root.info_screenwidth() or =root.infoScreenheight(); in TCL8.4: root.wm_attributes("-fullscreen", True) ↪ Observe syntax

Carefully To position root or toplevel windows use window.lift() or .lower(), or call on the window manager: top1.wm_attributes("-topmost", True)

Create Widgets: widget_name = widget_type(attributes (a.k.a options), and command callbacks); use any callable object as a callback: functions (most common), bound methods, lambda expressions, commands. No event information provided. Syntax:

↙ Your widget name ↗ option (ex: background color) function to bind when ↗ auto event occurs
b1=Button(parent, bg = 'light blue', text= 'Push me!', command= myCallbackFunction)
widget ↗ type ↗ parent window name ↗ option (ex: label text) button click binding is automatic

Bindings - events can bind at 4 levels: Application, Toplevel, Class, Instance (most common)

ex: label1.bind("Button-1", callback function name) **Manual Bindings** - To respond to an event not auto bound to a widget, use the .bind method to connect a callback. **widget.bind**

(event, callback) This binding delivers ←→ event objects to the callback if the function explicitly accepts it: def callback(event)

Common Manual Bindings

Bound event sequence : Comment

Button events:

<Button-1> : leftmost button, <1> is alias
<Button-2> : middle button if available
<Button-3> : rightmost mouse button
<ButtonRelease1> : button up
<Leave> : mouse pointer left widget
<B1-Motion>: movement with button down
<DoubleButton1> : double click
<Enter> : mouse pointer entered widget

Keyboard events:

<FocusIn> : keyboard focus moved to w
<FocusOut> : keyboard focus moved away
<Return> : the keyboard enter key
<Key> : w.bind("<Key>", cback) any keypress
"x" : a letter ex: frame.bind("x", callback)

Options, values, processes, and constants

Compass points: 'n', 'ne', 'e', 'se', 's', 'sw', 'w', 'nw', 'center'.
Bitmaps: 'error', 'gray75', 'gray50', 'gray25', 'gray12', 'hourglass', 'info', 'questhead', 'question', 'warning'



Colors: can be given as the names of colors in the rgb.txt file (downloaded with tkinter); also hex definitions #rgb, #rrggb, or #rrrgggbbb

Distance: Pixels → numeric; absolute distances → strings with a trailing character denoting units: c-centimeters, i-inches, m-millimeters, p-printer's points - these vary with font used.

FONTS: Ex: font=("Verdana 10 bold"). Font sizes with positive numbers measured in **points**; negative numbered sizes are measured in **pixels**.

Justify: "left", "center", "right", "fill" *include quotes*

Region: 4 space-delimited distances "3i 2i 4.5i 2i"

Relief: "raised", "sunken", "flat", "groove", "ridge"



Wrap: "none", "char", "word"

Cursors: many available such as: "arrow" "circle" "clock" "cross" "dotbox" "exchange" "fleur" "heart" "man" "mouse" "pirate" "plus" "shuttle" "sizing" "spider" "spraycan" "star" "trek" "watch"

Images: B&W id constructor: myBWpic = tk.BitmapImage(file=myimagefile.xbm); **Color:** myphotoimage = PhotoImage(file=myimagefile.{.gif, .pgm, .ppm formats})

Control Variables - shared by multiple widgets, updates all users automatically: constructors are =tk.DoubleVar(), =tk.IntVar(), =tk.StringVar(), =tk.BooleanVar(); use: v.get() and v.set(value)

Type of Events Listed above are keyboard and mouse or pointer events. Other types of events (not detailed here) are crossing, focus, exposure, configuration, and colormap

Special key bindings Cancel (the Break key), BackSpace, Tab, Return (the Enter key), Shift_L (any Shift key), Control_L (any Control key), Alt_L (any Alt key), Pause, Caps_Lock, Escape, Prior (Page Up), Next (Page Down), End, Home, Left, Up, Right, Down, Print, Insert, Delete, F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, Num_Lock, and Scroll_Lock; each encased in "< >"; ex: "<Tab>"

Other bindings: **Class:** .bind_class()

ex: self. Bind_class(w_type, "<event>", function)

Application: .bind_all('<some event>', function)

ex: self. bind_all('<Key-Print>', self.__printScreen)

Using lambda to pass simple values in a binding Ordinarily no variables are passed in an automatic binding and only event data in manual bindings. A lambda statement can be used to overcome this limitation with simple data - for example regular or control variables. Auto binding example:

command=lambda: button1callback(myvar) or

lambda: button1callback(cvar.get()), the callback function must specify a receiving variable container. For manual bind:

w.bind("<event>", lambda event, arg=variable: callback (event,arg)) *callback has 2 variables such as (event, myvar)*

comments and suggestions appreciated:
oakey.john@yahoo.com

WWW.WIKIPYTHON.COM

Event sequence tkinter term for an event name. It is a **string** containing one or more **event patterns**, as follows:

quote and angle bracket enclosure

" < [optional modifier(s)]... type [optional detail] >"

Alt, Control, Double, ⌘ like ⌘ Button ⌘ like 1, 2, 3
Lock, Shift, or Triple like ⌘ Return

Second Bindings if an automatic binding is already set to a **callback** (using the **command=callback** option), another binding can be manually set by including *args in the **callback** parens: **callback(*args)**. **callback** will now work for both but only the event you bind sends back event data and using it causes a built-in event error.

3 Geometries: Pack, Place, Grid - widget installation and formatting tools| Options and Attributes:

Geometry Options	Pack	Grid	Place	Default or Requirement	Options	Note
after	●			other window		name of another window
anchor	●		●	center	compass points	
before (other)	●			other window		name of another window
bordermode			●	inside	outside / ignore	border influence on slave placement
column		●		0	integer	column number; columns start with 0
columnspan		●		1	integer	number of columns spanned
expand	●			FALSE	true/false: 0/1	assign more space, distribute among widgets
fill	●			"none"	"x", "y", "both"	take up entire space, may need expand=
height			●	size	distance	outer dimension of window plus border
in_(target)	●	●	●	n/a	widget name	pack inside the target widget
ipadx	●	●		0	distance	internal padding pixels/distance; vertical
ipady	●	●		0	distance	internal padding pixels/distance; horizontal
padx	●	●		0	distance	external padding pixels/distance; vertical
pady	●	●		0	distance	external padding pixels/distance; horizontal
relheight			●	fp	fp 0 to 1.0	height relative to master; modified by - height
relwidth			●	fp	fp 0 to 1.0	width relative to master; modified by - width
relx			●	fp % * 100	0.0(left)-1.0(right)	anchor point x coordinates in master window
rely			●	fp % * 100	0.0(left)-1.0(right)	anchor point y coordinates in master window
row		●		first empty	row number	row number; rows start with 0
rowspan		●		1	integer	number of rows spanned
side	●			top / left/ right/ bottom		to pack against
sticky (glue widget to cell border)		●		centered	compass points	W+E stretch horiz; W+E+N+S - fill all; use a string="wens" or constants =W+E+N+S
width			●	size	distance	outer dimension of window plus border
x			●	0	distance	anchor point x coordinate in master window
y			●	0	distance	anchor point y coordinate in master window

Methods: Universal Geometry Methods

(x=widget name.a geometry name)

x_forget() remove from manager but do not destroy, can reuse **ex: lab1.grid_forget()**, retrieve by repeating the original grid command

x_info() ↳ a dictionary of options

x_slaves() returns list of sub widgets as tkinter widget references

x_configure(options) same as .pack()

Geometry Specific Methods

place: has no other Methods.

pack and grid:

x_propagate(flag) ; True/False; enables resizing of child widgets if too small

grid:

w.grid_bbox(column=None, row=None, col2=None, row2=None)

w.grid_size() tuple with number of columns and rows

w.grid_location(x,y) ↳ tuple with indexes

w.grid_remove() removes widget from manager; but the widget is available for reuse

To change the following, you must call these on a widget's **parent**:

grid_columnconfigure(index, options)

grid_rowconfigure(index, options)

index = column number

options: minsize=, pad=, weight=, uniform=

tkinter Widget Options or Attributes

tkinter Widget Options or Attributes	Button	Canvas	Checkbutton	Entry	Frame	Label	Labelframe	Listbox	Menu	Menubutton	Message	Panedwindow	Radiobutton	Scale	Scrollbar	Spinbox	Text	Toplevel	Value Type	Default Value
	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		
activebackground	●																		color	system
activeborderwidth																			distance value	
activeforeground	●	●	●		●					●									color	system
activerelief																			relief	raised
activestyle								●											dotbox, none,	dotbox-
anchor	●		●			●				●	●								compass points or "center"	usually center
aspect											●								positive integer aspect ratio	150
autoseparators														●					boolean	
background or -bg	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		color	system
bigincrement													●						fp	.1
bitmap	●		●			●					●								(see predefined list)	
blockcursor														●					boolean	false
borderwidth or -bd	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		distance value	2 pixels
buttonbackground																			color	
buttoncursor																			cusor name	default cursor
buttondownrelief																			relief	
buttonuprelief																			relief	
class						●													class for window - ● determines options	
closeenough	●																		fp value	1.0
colormap						●		●											'new' or another ● window	screen default
command	●			●															callback function name	none
compound	●			●		●					●			●	●	●	●		bottom, top, left, right, center	none
confine		●																	boolean	True
container				●															boolean	false
cursor	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		cursor name	system
default	●																		normal, active, disabled	
digits																			integer	
direction										●									above, below, left, right, flush	
disabledbackground					●														color	system
disabledforeground	●		●	●		●		●		●			●						color	
elementborderwidth														●					pixels	=borderwidth?
endline																			next-after-last line ● index integer	
exportselection				●				●							●	●			1 or 0	1
font	●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		font 3 tuple	
foreground or -fg	●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		color	

tkinter Widget Options or Attributes

tkinter Widget Options or Attributes	Button	Canvas	Checkbutton	Entry	Frame	Label	Labelframe	Listbox	Menu	Menubutton	Message	Panedwindow	Radiobutton	Scale	Scrollbar	Spinbox	Text	Toplevel	Value Type	Default Value
from														●		●			fp - lowest value	
handlepad														●					distance value	
handlesize														●					distance value	
height	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	text-lines, image-distance value	1
highlightbackground	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	color	system
highlightcolor	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	color	system
highlightthickness	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	distance value	1 pixel
image	●	●	●	●	●	●	●	●	●	●	●	●	●						bit file: gif, pgm, ppm	
inactivebackground																		●	color	
increment																	●		fp(can be + or -)	
indicatoron																		●	boolean	false
insertbackground	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	color	black
insertborderwidth	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	distance value	0
insertofftime	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	milliseconds	300
insertontime	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	milliseconds	600
insertunfocussed																	●		none,(no cursor), hollow, solid	none
insertwidth	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	distance value	2 pixels
invalidcommand or -invcmd																	●		script to evaluate	
jump																	●		boolean F-smooth T-update at release	
justify	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	see justify choices	center
label																	●		string	
labelanchor																	●		compass points	"nw"
labelwidget																	●		widget	
length																	●		distance value	
listvariable														●					name of a global variable	
maxundo																	●		integer	
menu																	●		name of associated menu	
offrelief			●															●	relief	
offvalue			●															●	value	0
onvalue			●														●		value	1
opaqueresize																	●		boolean	
orient																	●	●	'horizontal', 'vertical'	
overrelief	●	●	●																relief	"" (empty string)
padx	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	distance value	1 pixel
pady	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	distance value	1 pixel
proxybackground																		●	color	background color
proxyborderwidth																	●		pixels	

**tkinter
Widget
Options or
Attributes**

	Button	Canvas	Checkbutton	Entry	Frame	Label	Labelframe	Listbox	Menu	Menubutton	Message	Panedwindow	Radiobutton	Scale	Scrollbar	Spinbox	Text	Toplevel		Value Type	Default Value
readonlybackground				●										●					color	background	
relief	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	see relief choices	"sunken"	
repeatdelay	●													●	●	●			milliseconds before engage		
repeatinterval	●													●	●	●			milliseconds between execution		
resolution														●					real value	1 (integral)	
sashcursor														●					mouse cursor	horiz: sb_h_double_arrow, vert: sb_v_double_arrow	
sashpad														●					distance value		
sashrelief														●					relief		
sashwidth														●					distance value		
screen																	●	window	screen name for new window		
scrollregion	●																		coords: fp		
selectbackground	●		●						●							●	●		color		
selectborderwidth	●		●					●								●	●		distance value		
selectcolor		●	●											●					color		
selectforeground	●		●					●								●	●		text color		
selectimage		●																	image	ignored unless image option	
selectmode								●											single, browse, multiple, extended		
setgrid								●									●		boolean		
show			●																string - 1 character		
showhandle														●					boolean		
showvalue														●					boolean		
sliderlength														●					distance value		
sliderrelief														●					relief		
spacing1															●				distance value		
spacing2															●				distance value		
spacing3															●				distance value		
startline															●				integer index or ""		
state	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	Normal or Disabled	normal	
tabs															●				list of distances for	8 characters	
tabstyle															●				'tabular' or 'wordprocessor'	'tabular'	
takefocus	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	0 or 1	1	
text	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	a string		
textvariable	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	a string		
tickinterval														●					real number		
to														●		●			real number		
tristateimage		●																	button)		

tkinter
Widget
Options or Attributes

	Button	Canvas	Checkbutton	Entry	Frame	Label	Labelframe	Listbox	Menu	Menubutton	Message	Panedwindow	Radiobutton	Scale	Scrollbar	Spinbox	Text	Toplevel		Value Type	Default Value	Note
troughcolor													●	●						color (not on Windows)		
underline	●	●	●	●						●			●							integer		index of char to underline
undo														●						boolean		text; undo mechanism is active or not
use														●						hex string window identifier		toplevel; get from winfo id
validate				●												●				none, focus, focusin, focusout, key, or all	none	spinbox, entry; specifies validation mode
validatecommand or -vcmd				●												●				script to eval, {} disables	{}	spinbox, entry; must return boolean value
value													●							proper list		over from to
values													●							proper list		spinbox; content control
variable			●										●	●						name of global variable	SelectedButton value	checkbox, radiobutton, scale;
visual				●										●	info/complex					see TK_GetVisual for info/complex		
width	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	characters		
wrap													●	●						boolean		spinbox, text; wrap around values of data
wraplength	●		●	●				●					●							distance value	nowrap	max line length; 0-none
xscrollcommand	●		●					●						●		=scrollbar.set					scrollbar.config(command=w.xview)	
xscrollincrement	●																			distance		canvas; increment or horiz scroll
yscrollcommand	●					●										●				widget name + set		canvas, text, listbox;
yscrollincrement	●																			distance		canvas; increment or vert scroll

Window Manager Methods: `wm_method` *note: `w.wm_method` is the same as just `w.method`

attributes ex: <code>root.attributes("-fullscreen", True)</code>	<-NOTE QUOTES!	forget	window unmapped; can be remanaged
-alpha	transparency, 0.0-1.0	frame	returns window identifier
-fullscreen	NEW	geometry	widthxheight+woffset+yoffset (newGeometry cancels, reverts to native size)
-topmost	place this window on top	grid	only resize on boundries; use setGrid instead
-disabled	disables window	group	returns or sets widget leader of group of windows
-toolwindow	only shows close button	iconbitmap	(bitmap=); sets window icon
-transparentcolor	transparent color index of toplevel	iconify	iconify widget
-modified	for macs?	iconmask	(bitmap=); used with iconbitmap; 0-off 1-on
-notify	for macs?	iconname	(newName=) displayed inside widget icon
-titlepath		iconphoto	([default=None], [image]...) sets titlebar icon
-transparent	window area transparent	iconposition	(x,y); set icon position
-type	for x11 - see tcl.tk/man/tcl8.6	iconwindow	(widget); display name instead of icon
aspect	w/h ration min#/denom~max#/denom	manage	(widget); widget become standalone top level
client	(name=None); gets/stores WM_CLIENT_MACHINE	maxsize	(self, width=None,height=None) set max size
colormapwindows	store list of window names	minsize	(self, width=None,height=None) set min size
command	(value=); WM_COMMAND	overrideredirect	(boolean); undependable
deiconify	display normal form, set focus, raise	positionfrom	(who= program or user) too complex for practice
focusmodel	active passive (default); active claims focus	protocol	(name=name of atom for protocol)(command=)
forget	window unmapped; can be remanaged	resizable <small>topLevel only</small>	(width=boolean, height=boolean) can user resize
frame	returns window identifier	sizefrom	(who= program or user) too complex for practice
geometry	widthxheight+woffset+yoffset	stackorder	↳ toplevel list in order; (window1 [isabove]
	(newGeometry cancels, reverts to native size)	state	[isbelow] window 2) returns boolean result
grid	only resize on boundries; use setGrid instead	title	↳ one of normal, iconic, withdrawn, icon, zoomed
		transient	(master=); if specified, window is pull-down menu

Comments welcome @ www.wikipython.com
Widget Methods or Commands

Six widgets have only two methods or commands:

get (option) - retrieve opt value
configure (opts/values) - set options/values

Frame	Label	Labelframe
Menubutton	Message	Toplevel

BUTTON	CHECKBUTTON
cget (option)	cget (option)
configure (options/values)	configure (options/values)
flash	deselect
invoke callback	flash
MESSAGEBOX	
NONE	

PANEDWINDOW	
add	(child, option(s))
cget	(option)
configure	(options/values)
forget	(child)
identify	(xy coord)
proxy	(arg[s])
coord	↳ proxy xy
forget	
place	place @ xy
sash	
coord	index
dragto	index x y
mark	index x y
place	index x y
panecget	(window, option)
paneconfigure	(child, opt/vals)
after	(child)
before	(child)
height	size
hide	boolean
minsize	x y
padx	distance
pady	distance
sticky	nesw
stretch	always, first, last, middle, never
width	size
panes	↳ list of widgets
ENTRY	
bbox	index
cget	option
configure	option(s)/value(s)
delete	first, last
get	
icursor	index
index	index
insert	index string
scan	
mark	x
dragto	x
selection	
adjust	index
clear	
from	index
present	
range	start,end
index	
validate	
xview	
2 real fractions visible span	
index	to left edge
moveto	fraction
scroll	number.what
what=units or pages	
INDICES:	
number	character -0 1st
anchor	point
end	char after string
insert	next after cursor
sel.first	1st char
sel.last	char after select
@number	x coord

TEXT	
bbox	index x,y,width,height
cget	option
compare	index1 operator index2
configure	options/values
count	option, index1/2
chars	abbrev as "c"
displaychars	(visible chars)
displayindices	^+wins,images
displaylines	lines i1 to i2
indices	all objects
lines	abbrev as "l"
xpixels	hz pix to i2
ypixels	vt pix to i2
debug	boolean; internal ck
delete	index1/2
dlineinfo	index 5 ele area info
dump	default is all
all	all info
command	key, value, index
image	+ in dump reslt
mark	+ in dump reslt
tag	+ in dump reslt
text	+ in dump reslt
window	+ in dump reslt
edit	option
canredo	true if possilbe
canundo	true if possilbe
modified	boolean ; widget flag
redo	
reset	clears stacks
separator	
undo	undo last edit
get	[displaychars]
image	index1, index2
option	see embedded
image	image block for values
cget	index option value
configure	index option value
create	index option value
image	names
index	index
insert	index
mark	option, args
gravity	markName,direction
names	marks currently set
next	index
previous	index
set	markName, index
unset	markName...
peer	options args
create	newPathname, options
names	↳ list of peers
pendingsync	↳ 1 if not up to date, 0 if ok
replace	index1, index2
scan	(w/ chars/tags)
option	args
mark	x y
dragto	x y
search	switches, pattern, stop
these are the switches:	
	forwards
	backwards
	exact
	regexp

TEXT - continued	
nolinesstop	
nocase	
count	varName
all	
overlap	
strictlimits	
elide	(hidden)
--	(terminates switch list)
see	index (forces visible)
sync	(forces lines calcs)
tag	command (exe after line sync)
	options, args
add	tagName, indexes
bind	tagName, seq,script
cget	tagName, option/value
configure	tagName, option/value
delete	tagName
lower	tagName, belowThis
names	index
nextrange	tagName, indexes
prevrange	tagName, indexes
raise	tagName,aboveThis
ranges	tagName
remove	tagName, indexes
window	option, args
cget	index, option
configure	index, option/value
create	index, option/value
names	
xview	option, args
xview	
moveto	fraction
scroll	number,what
yview	args
yview	↳ pix1, pix after last
moveto	fraction
scroll	number,what
pickplace	index
number	obsolete
SCROLLBAR	
scrollbars are created and attached to other widgets	
Scrollbar methods	
activate	element
cget	(option)
configure	(options/values)
delta	Δx Δy
fraction	x y
get	
identify	x y
set	first, last
Scrolling methods	
prefix moveto	
	fraction 0=start
prefix scroll	# units (chr/Ins)
prefix scroll	# pages

Event Types	
syntax:	"<type>"
Key	
KeyRelease	
Button-x, B1-B5	
ButtonRelease	
Motion <B1-Motion>	
Enter	
Leave	
FocusIn	
FocusOut	
Expose	
Visibility	
Destroy	
Unmap	
Map	
Reparent	
Configure	
Gravity	
Circulate	
Property	
Colormap	
Activate	
Deactivate	
MouseWheel	
Safe set keysyms	
Character	Code
1-9	49-57
0	48
a-z	97-122
space	32
quotleft	96
minus	45
equal	61
bracketleft	91
bracketright	93
backslash	92
semicolon	59
quotright	39
comma	44
period	46
slash	47
plus	43
A Few Colors	
grey(1-99), gray, dim	gray
gray, slate gray, snow	(2-4), linen, ivory(2-4), red, red (2-4), misty rose, maroon(1-4), violet red, orange, DarkOrange(1-4), yellow, gold(2-4), goldenrod(1-4), green, SeaGreen(1-3), blue, SkyBlue(1-4), RoyalBlue(1-4), purple, DarkOrchid1, MediumPurple(1-4), violet, dark violet, VioletRed(1-4), burleywood(1-4), white, black

SPINBOX

see indices reference block	
bbox	index
cget	(option)
configure	(options/values)
delete	first, last
get	
icursor	index
identify	x y
index	index
insert	index string
invoke	element
scan	option args
mark	x y
dragto	x y
selection	to adj selection
adjust	sel end
clear	
element	element
from	index
present	
range	start, end
to	index
set	string
validate	↳ 0 or 1
xview	args chg hz pos
index	
moveto	fraction
scroll	number, what

LISTBOX

activate	index
bbox	index
cget	option
configure	(options/values)
curselection	
delete	first, last
get	first, last
index	index
insert	index, element
itemcget	index, option
itemconfigure	(options/values)
background	
foreground	
selectbackground	
selectforeground	
nearest	y
scan	
mark	x y
dragto	x y
see	index
selection	option
anchor	index
clear	first, last
includes	index
set	first, last
size	str no of elemnts
xview	2 real fractions visible span
index	
moveto	fraction
scroll	number,what

LISTBOX continued

yview	2 real fractions visible span	cget	(option)	get	(option)
index		configure	(options/values)	configure	(options/values)
moveto	fraction	deselect		coords	X Y list @ val point
scroll	number,what	flash		get	x y
select		invoke		identify	x y
select		select		set	value

CANVAS

addtag	above	tagOrId	postscript	option/value...
	all		channel	channelName
	below	tagOrId	colormap	varName
	closest	tagOrId	colormode	mode
	enclosed	x,y,halo,start	file	fileName
	overlapping	x1 y1 x2 y2	fontmap	varName
	withtag	tagOrId	height	size
bbox	tagOrId...		pageanchor	anchor
bind	tagOrId, sequence, command		pageheight	size
canvax	screenx, gridspacing		pagewidth	size
canvay	screeny, gridspacing		pagey	position
cget	(option)		rotate	boolean
configure	(options/values)		width	size
coords	tagOrId, x, y		x	position
coords	tagOrId, coordList		y	position
create	type x y, xy, option/value		raise	aboveThis
create	type coordList, option/value		rchars	tagOrId first last string
dchars	tagOrId, first, last		scale	tagOrId xOrigin yOrigin xScale yScale
delete	tagOrId, tagOrId...		scan	option, args
dtag	tagOrId.tagToDelete		mark	x y
find	searchCommand, arg...		dragto	x y, gain
focus	tagOrId		select	
gettags	tagOrId		adjust	tagOrId index
icursor	tagOrId, index		clear	tagOrId index
imove	tagOrId, index x y		from	tagOrId index
index	tagOrId, index		item	tagOrId index
insert	tagOrId beforeThis		to	tagOrId index
itemcget	tagOrId, option		type	
itemconfigure	tagOrId, option/value...		xview	tagOrId args
lower	tagOrId		moveto	fraction
move	tagOrId		scroll	number, what
moveto	tagOrId		yview	
scroll	tagOrId		moveto	fraction
scroll	tagOrId		scroll	number, what

Information Methods (winfo.x)

atom(name (a string), displayof=0) : unique integer mapped to string
atomname(id, displayof=0) : mapped to id
cells() : # of cells in colormap
children() : list of children
class() : widget class
colormapfull() :
containing(rootx, rooty, displayof=0) : widget @ this position
depth() : bit depth
exists() : true if widget exists
fpixels(distance,) : fp-# of screen pixels
geometry() : geometry sting
height() : widget height in pixels
id() : window identifier
intersp(displayof=0) : TCL interpreter mem list
ismapped() : boolean; check if window created
manager() : geo mgr: grid, pack, place, canvas, text
name() : widget name
parent() : widget parents full name
pathname(displayof=0) : full window name of id window
pixels(distance,) : convert to pixels
pointerx() : x coord of pointer on the root
pointery() : xy coords of pointer on the root
reqheight() : min size to display widget
reqwidth() : min size needed to display widget
rgb(color,) : an RGB 3 tuple - 0 to 65535

rootx() : left edge cood rel to screen
rooty() : left edge cood rel to screen
screen() : screen name as dec integer; always ":0.0" in Windows
screencells() : # of color cells
screendepth() : default bit depth
screenheight() : height of widget screen
screenmmheight() : screen height in mm
screenmmwidth() : screen width in mm
screenvisual() : "psuedocolor" or "truecolor"
screenwidth() : width of screen in pixels
server() : widget screen xserver window info
toplevel() : widget root
viewable() : True is widget chain is mapped
visual() : directcolor, grayscale, pseudocolor, staticcolor, staticgray, or truecolor
visualid() : X identifier for visual this widget
visualsavailable() : list of all visuals available for widget screen
vrootheight() : height of the virtual root window for widget
vrootwidth() : width of the virtual root window for widget
vrootx() : x offset of virtual root relative to root window of widget screen
vrooty() : yoffset of virtual root relative to root window of widget screen
width() : widget width, pixels (update_idletasks)
x() : upper corner coord
y() : upper corner coord