Widget Methods Table

Most of this table is a consolidtion of the Shipman document at http://infohost.nmt.edu/tcc/help/pubs/tkinter/web/index.html with some modifications from info on: https://www.tcl.tk/man/tcl8.5/

also helpful: http://effbot.org/tkinterbook/widget.htm (Note: ↬ means 'returns' or 'yields')

Note: Toplevel methods listed separately.

Method	Comment	IN TB4	Type of Method
.lift(aboveThis=None)	w window moved to top of the stack	x	Manager
.lower(belowThis=None)	w window moved to bottom of the stack	x	Manager
.mainloop()	***SEE NOTE**	x	Manager
.quit()	This method exits the main loop.	x	Manager
.clipboard_append(text)	append selected text to clipboard		Action
.clipboard_clear()	clear clipboard		Action
.after(delay_ms, callback, *args)	timed callback after delay; id is returned		Callback
.after_cancel(id)	cancels a timed callback		Callback
.after_idle(func, *args)	callback waits till the system is idle		Callback
.bind(event, function, add=None)	bind event to callback; + for multiple bindings	x	Callback
.bind_all(sequence=None, func=None,add=None)	binds to all widgets in the entire app	x	Callback
.bind_class(className, sequence=None, func=None, add=None)	connects all widget of a certain class to an event (sequence) to be handled by a single function.	x	Callback
.bindtags(tagList=None)	↬ binding tags as tuple of strings		Callback
.unbind(sequence, funcid=None)	removes event bind; remove funcid	x	Callback
.unbind_all(sequence)	remove all bindings for an event	x	Callback
.unbind_class(className, sequence)	remove all binding for a class of widgets		Callback
.column_configure()	apply to parent of grided widget	x	Configure
.config(option=value, ...)	same as .configure().		Configure
.configure(option=value, ...)	set option values; See Shipman	x	Configure
.event_add(virtual event name strng, event triggers)	See Shipman		Configure
.event_delete(virtual event name strng, event to remove)	See Shipman		Configure
.option_add(pattern, value, priority= None)	See Shipman		Configure
.option_clear()	resets options to default	x	Configure
.option_readfile(fileName, priority= None)	file where users can put their preferred options, using the same format as the .Xdefaults file.		Configure
w.option_add(pattern, value, priority=None)	adds default option value to the option db. See Shipman		Configure
.destroy()	destroys w and all its children.	x	Environment
.focus_force()	forces input focus to w; "impolite"(?)	x	Environment
.winfo_containing(rootX, rootY, displayof = 0)	find window with X,Y - See Shipman		Environment
.event_generate(event to be triggered, **kw)	kw: field name in event=value		Execution
.focus_set()	moves focus to w IF w's app has focus	x	Execution
.grab_current()	↬ identifier or "none"	x	Execution
.grab_release()	release if grab in force	x	Execution
.grab_set()	grabs all events for w's app; replaces other grabs	x	Execution
.grab_set_global()	grab all events for entire screen	x	Execution
.register(function)	See Shipman: http://infohost.nmt.edu/tcc/help/pubs/tkinter/web/entry-validation.html		Execution
.selection_clear()	clear any selection w has	x	Execution
.selection_get()	selected text or if none tk.TclError	x	Execution
.selection_own()	See Shipman		Execution
.selection_own_get()	See Shipman		Execution
.update()	forces display update; unpredictable;	x	Execution
.update_idletasks()	for update of deferred tasks		Execution
.wait_variable(variable)	local wait loop for v to be set; app cont	x	Execution
.wait_visibility(w)	wait until widget *w* (typically a Toplevel) is visible.		Execution
.wait_window(w)	Wait until window *w* is destroyed.		Execution
.tk_focusFollowsMouse()	force MOUSE focus versus keyboard	x	Executon
.grid_forget()	w disappears-not destroyed-forgets options	x	Grid
.grid_propagate()	force w size regarless of content		Grid
.grid_remove()	like forget but remembers options	x	Grid
.image_names()	↬ all image names in ap	x	Grid
.rowconfigure()	grid management - call on the w parent	x	Grid
.bell()	beep		Information
.cget(option)	↬ option value AS A STRING REGUARDLESS	x	Information
.event_info()	↬ all current virtual event names		Information
.focus_displayof()	↬ name of window with input focus, "none"	x	Information
.focus_get()	↬ widget with focus or "none"	x	Information
.focus_lastfor()	see Shipman		Information
.grab_status()	↬ 'local', 'global', or 'none'	x	Information
.keys() [list = widget.keys()]	↬ options for w as a sequence of strings		Information
.nametowidget(name)	↬ w whose path name is name		Information
.option_get(instance key, classname)	↬ current value		Information
.tk_focusNext()	returns next w in normal sequence	x	Information
.tk_focusPrev()	return to last w in normal sequence		Information
.winfo_children()	↬ list w children low to high		Information
.winfo_class()	↬ w's class name (e.g., 'Button').		Information
.winfo_depth()	↬ the number of bits per pixel in w's display.		Information
.winfo_fpixels(number)	↬ w height pixels; update idle tasks	x	Information
.winfo_geometry()	w's geometry() string and location; must update idle tasks first		Information
.winfo_height()	↬ w height pixels; update idle tasks	x	Information
.winfo_id()	↬ an integer; needed for .winfo_pathname()	x	Information
.winfo_ismapped()	true or false; gridded, placed or packed all the way to parent		Information
.winfo_manager()	↬ "", or 'grid','pack', 'place', 'canvas', or 'text'.		Information
.winfo_name()	w's name relative to its parent; **See Shipman**		Information
.winfo_parent()	w's parent's pathname		Information
.winfo_pathname(id, displayof=0)	See Shipman		Information
.winfo_pixels(number)	↬ that distance in pixels on w's display; integer		Information
.winfo_pointerx()	↬ x relative to root, -1-1 if mouse on different screen		Information
.winfo_pointerxy()	A tuple x,y per root or -1-1if mouse on different screen	x	Information
.winfo_pointery()	↬ y relative to root, -1-1 if mouse on different screen		Information
.winfo_reqheight()	↬ min h for all of w's content		Information
.winfo_reqwidth()	↬ min w for all of w's content		Information
.winfo_rgb(color)	↬ 3-tuple (r,g,b) of integers for "color"		Information
.winfo_rootx()	↬ left side x of w's root rel to parent	x	Information
.winfo_rooty()	↬ top side y of w's root rel to parent	x	Information
.winfo_screenheight()	height of screen in pixels		Information
.winfo_screenmmheight()	height of screen in millimeters		Information
.winfo_screenmmwidth()	width of screen in millimeters		Information
.winfo_screenvisual()	color rendition of display; 'truecolor' or 'pseudocolor' (256-color displays)		Information
.winfo_screenwidth()	width of screen in pixels	x	Information
.winfo_toplevel()	↬ top-level window containing w.		Information
.winfo_viewable()	a true value if viewable		Information
.winfo_width()	w in pixels; use .winfo_reqwidth() instead	x	Information
.winfo_x()	w's left x relative to its parent; outer border edge		Information
.winfo_y()	w's top side y relative to parent; outer border edge		Information

Shipman: http://infohost.nmt.edu/tcc/help/pubs/tkinter/web/index.html
tcl/tk manual pages: https://www.tcl.tk/man/tcl8.5/

* Can also set an option value with w[option]=value

* .mainloop() This method must be called, generally after all the static widgets are created, to start processing events. You can leave the main loop with the .quit() method (below). You can also call this method inside an event handler to resume the main loop.

Method Type	Type Description
Action	an auxillary method to affect pheriphal activity
Callback	part of the process of going to a function to handle a real or contrived event
Configure	establish objects or values of objects, methods or attributies
Environment	alters widgets or the processes they generate
Execution	triggers, controls, alters, extends or implements a GUI process
Grid	call a grid method
Information	returns information about the process or environment
Manager	manipulates the tkinter root or toplevel windows