Python Tkinter.Toplevel() Examples

The following are *50* code examples for showing how to use *Tkinter.Toplevel()*. They are extracted from open source Python projects. You can vote up the examples you like or vote down the exmaples you don't like. You can also save this page to your account.

Example 1

Project: *BISIP* Author: *clberube* File: *GUI.py* (MIT License) *View Source Project* 10 votes

```
def popup_bivar(self, bivar_type):
        try: self.top_bivar.destroy()
        except: pass
        self.top_bivar = tk.Toplevel()
        self.top_bivar.title("Bivariate plotting")
        tk.Label(self.top_bivar, text="Select two different parameters: ").grid(row=0, column=0, sticky=tk.W+tk.E+tk.N, pady=(5,5))
        self.biv1, self.biv2 = tk.StringVar(), tk.StringVar()
        self.biv1.set(self.list_of_parameters[0])
        self.biv2.set(self.list_of_parameters[1])
        optionmenu1 = tk.OptionMenu(self.top_bivar, self.biv1, *self.list_of_parameters)
        optionmenu2 = tk.OptionMenu(self.top_bivar, self.biv2, *self.list_of_parameters)
        optionmenu1.grid(row=1, column=0, sticky=tk.W+tk.E+tk.S)
        optionmenu1.config(bg = "gray97", relief=tk.GROOVE)
        optionmenu2.grid(row=2, column=0, sticky=tk.W+tk.E+tk.S)
        optionmenu2.config(bg = "gray97", relief=tk.GROOVE)
        button = tk.Button(self.top_bivar, height=1, width=20, text="OK", command=lambda: self.plot_diagnostic(bivar_type), bg='gray97',
        button.grid(row=3, column=0, sticky=tk.S, pady=(10,10))

    # Diagnostics buttons
```

Example 2

Project: *Projects* Author: *it2school* File: *ImageTk.py* (license) *View Source Project* 8 votes

```
def _show(image, title):
    """Helper for the Image.show method."""

    class UI(tkinter.Label):
        def __init__(self, master, im):
            if im.mode == "1":
                self.image = BitmapImage(im, foreground="white", master=master)
            else:
                self.image = PhotoImage(im, master=master)
            tkinter.Label.__init__(self, master, image=self.image,
                                   bg="black", bd=0)

    if not tkinter._default_root:
        raise IOError("tkinter not initialized")
    top = tkinter.Toplevel()
    if title:
        top.title(title)
    UI(top, image).pack()
```


Related Functions

- sys.exit()
- sys.argv()
- re.compile()
- os.path()
- time.time()
- os.listdir()
- time.sleep()
- re.search()
- os.system()
- threading.Thread()
- math.sqrt()
- Tkinter.Tk()
- Tkinter.Label()
- Tkinter.Button()
- Tkinter.Frame()
- Tkinter.Entry()
- Tkinter.Canvas()
- Tkinter.StringVar()
- Tkinter.END
- Tkinter.PhotoImage()

Related Modules

- os
- sys
- re
- time
- logging
- datetime
- random
- string
- math
- subprocess
- threading
- copy
- json
- numpy

Example 3

Project: *BISIP* Author: *clberube* File: *GUI.py* (MIT License) *View Source Project* 6 votes

```
def plot_window(self, fig, name=""):
        top_fig = tk.Toplevel()
        top_fig.lift()
        top_fig.title(name)
        top_fig.rowconfigure(1, weight=1)
        def _quit():
            top_fig.destroy()
        canvas = FigureCanvasTkAgg(fig, master=top_fig)
        canvas.show()
        canvas.get_tk_widget().grid(row=1, column=0, columnspan = 3, pady=(15,15), padx=(25,25), sticky=tk.N+tk.S+tk.E+tk.W)
        button = tk.Button(top_fig, height=1, width=20, text="Dismiss", bg='gray97', command=_quit, relief=tk.GROOVE)
        button.grid(row=2, column=0, columnspan=1, sticky=tk.W, pady=(10,10), padx=(20,20))
        toolbar_frame = tk.Frame(top_fig)
        toolbar_frame.grid(row=0,column=0,columnspan=2, sticky=tk.W+tk.E)
        NavigationToolbar2TkAgg( canvas, toolbar_frame )
        top_fig.resizable(width=tk.FALSE, height=tk.FALSE)
```

Example 4

Project: *BISIP* Author: *clberube* File: *GUI.py* (MIT License) *View Source Project* 6 votes

```
def popup_bivar(self, bivar_type):
        try: self.top_bivar.destroy()
        except: pass
        self.top_bivar = tk.Toplevel()
        self.top_bivar.title("Bivariate plotting")
        tk.Label(self.top_bivar, text="Select two different parameters: ").grid(row=0, column=0, sticky=
        self.biv1, self.biv2 = tk.StringVar(), tk.StringVar()
        self.biv1.set(self.list_of_parameters[0])
        self.biv2.set(self.list_of_parameters[1])
        optionmenu1 = tk.OptionMenu(self.top_bivar, self.biv1, *self.list_of_parameters)
        optionmenu2 = tk.OptionMenu(self.top_bivar, self.biv2, *self.list_of_parameters)
        optionmenu1.grid(row=1, column=0, sticky=tk.W+tk.E+tk.S)
        optionmenu1.config(bg = "gray97", relief=tk.GROOVE)
        optionmenu2.grid(row=2, column=0, sticky=tk.W+tk.E+tk.S)
        optionmenu2.config(bg = "gray97", relief=tk.GROOVE)
        button = tk.Button(self.top_bivar, height=1, width=20, text="OK", command=lambda: self.plot_diag
        button.grid(row=3, column=0, sticky=tk.S, pady=(10,10))

    # Diagnostics buttons
```

Example 5

Project: *BISIP* Author: *clberube* File: *GUI.py* (MIT License) *View Source Project* 6 votes

```
def plot_window(self, fig, name=""):
        top_fig = tk.Toplevel()
        top_fig.lift()
        top_fig.title(name)
        top_fig.rowconfigure(1, weight=1)
        def _quit():
            top_fig.destroy()
        canvas = FigureCanvasTkAgg(fig, master=top_fig)
        canvas.show()
        canvas.get_tk_widget().grid(row=1, column=0, columnspan = 3, pady=(15,15), padx=(25,25), sticky=tk.N+tk.S+tk.E+tk.W)
        button = tk.Button(top_fig, height=1, width=20, text="Dismiss", bg='gray97', command=_quit, relief=tk.GROOVE)
        button.grid(row=2, column=0, columnspan=1, sticky=tk.W, pady=(10,10), padx=(20,20))
        toolbar_frame = tk.Frame(top_fig)
        toolbar_frame.grid(row=0,column=0,columnspan=2, sticky=tk.W+tk.E)
        NavigationToolbar2TkAgg( canvas, toolbar_frame )
        top_fig.resizable(width=tk.FALSE, height=tk.FALSE)
```

Example 6

Project: *workflows.kyoyue* Author: *wizyoung* File: *ImageTk.py* (MIT License) *View Source Project* 6 votes

- collections

Others in *Tkinter*

- .Tk()
- .Label()
- .Button()
- .Frame()
- .Entry()
- .Canvas()
- .Toplevel()
- .StringVar()
- .Listbox()
- .Scrollbar()
- .Menu()
- .Text()
- .IntVar()

```
def _show(image, title):
    """Helper for the Image.show method."""

    class UI(tkinter.Label):
        def __init__(self, master, im):
            if im.mode == "1":
                self.image = BitmapImage(im, foreground="white", master=master)
            else:
                self.image = PhotoImage(im, master=master)
            tkinter.Label.__init__(self, master, image=self.image,
                                   bg="black", bd=0)

    if not tkinter._default_root:
        raise IOError("tkinter not initialized")
    top = tkinter.Toplevel()
    if title:
        top.title(title)
    UI(top, image).pack()
```


Example 7

Project: *gobang_AI* Author: *colingogogo* File: *graphics.py* (license) *View Source Project* 6 votes

```
def __init__(self, title="Graphics Window",
                 width=200, height=200, autoflush=True):
        assert type(title) == type(""), "Title must be a string"
        master = tk.Toplevel(_root)
        master.protocol("WM_DELETE_WINDOW", self.close)
        tk.Canvas.__init__(self, master, width=width, height=height,
                           highlightthickness=0, bd=0)
        self.master.title(title)
        self.pack()
        master.resizable(0,0)
        self.foreground = "black"
        self.items = []
        self.mouseX = None
        self.mouseY = None
        self.bind("<Button-1>", self._onClick)
        self.bind_all("<Key>", self._onKey)
        self.height = int(height)
        self.width = int(width)
        self.autoflush = autoflush
        self._mouseCallback = None
        self.trans = None
        self.closed = False
        master.lift()
        self.lastKey = ""
        if autoflush: _root.update()
```

Example 8

Project: *Snake-Game* Author: *StarostaGit* File: *graphics.py* (license) *View Source Project* 6 votes

```
def __init__(self, title="Graphics Window",
                 width=200, height=200, autoflush=True):
        assert type(title) == type(""), "Title must be a string"
        master = tk.Toplevel(_root)
        master.protocol("WM_DELETE_WINDOW", self.close)
        tk.Canvas.__init__(self, master, width=width, height=height,
                           highlightthickness=0, bd=0)
        self.master.title(title)
        self.pack()
        master.resizable(0, 0)
        self.foreground = "black"
        self.items = []
        self.mouseX = None
        self.mouseY = None
        self.bind("<Button-1>", self._onClick)
        self.bind_all("<Key>", self._onKey)
        self.height = int(height)
        self.width = int(width)
        self.autoflush = autoflush
        self._mouseCallback = None
        self.trans = None
        self.closed = False
        master.lift()
        self.lastKey = ""
        if autoflush: _root.update()
```

Example 9

Project: *goreviewpartner* Author: *pnprog* File: *toolbox.py* (license) *View Source Project* 6 votes

```
def start_review(self):
                import dual_view
                app=self.parent
                screen_width = app.winfo_screenwidth()
                screen_height = app.winfo_screenheight()

                Config = ConfigParser.ConfigParser()
                Config.read("config.ini")

                display_factor=.5
                try:
                        display_factor=float(Config.get("Review", "GobanScreenRatio"))
                except:
                        Config.set("Review", "GobanScreenRatio",display_factor)
                        Config.write(open("config.ini","w"))

                width=int(display_factor*screen_width)
                height=int(display_factor*screen_height)
                # Toplevel()

                new_popup=dual_view.DualView(self.parent,self.filename[:-4]+".rsgf",min(width,height))
                new_popup.pack(fill=BOTH,expand=1)
                self.remove_app()
```

Example 10

Project: *ecel* Author: *ARL-UTEP-OC* File: *Pmw.py* (license) *View Source Project* 6 votes

```
def installedversions(alpha = 0):
    if alpha:
        return ()
    else:
        return (_VERSION,)


######################################################################
### File: PmwBase.py
# Pmw megawidget base classes.

# This module provides a foundation for building megawidgets.  It
```

```
# contains the MegaArchetype class which manages component widgets and
# configuration options.  Also provided are the MegaToplevel and
# MegaWidget classes, derived from the MegaArchetype class.  The
# MegaToplevel class contains a Tkinter Toplevel widget to act as the
# container of the megawidget.  This is used as the base class of all
# megawidgets that are contained in their own top level window, such
# as a Dialog window.  The MegaWidget class contains a Tkinter Frame
# to act as the container of the megawidget.  This is used as the base
# class of all other megawidgets, such as a ComboBox or ButtonBox.
#
# Megawidgets are built by creating a class that inherits from either
# the MegaToplevel or MegaWidget class.
```

Example 11

Project: *pyktrader2* Author: *harveywwu* File: *gui_agent.py* (license) *View Source Project* 6 votes

```
def position_view(self):
        params = self.app.get_agent_params(['Positions'])
        positions = params['Positions']
        pos_win   = tk.Toplevel(self)
        pos_frame = ScrolledFrame(pos_win)
        fields = ['gateway', 'inst', 'currlong', 'currshort', 'locklong', 'lockshort', 'ydaylong', 'ydayshort']
        for idx, field in enumerate(fields):
            row_idx = 0
            tk.Label(pos_frame.frame, text = field).grid(row=row_idx, column=idx)
            for gway in positions.keys():
                for inst in positions[gway]:
                    row_idx += 1
                    if field == 'inst':
                        txt = inst
                    elif field == 'gateway':
                        txt = str(gway)
                    else:
                        txt = positions[gway][inst][field]
                    tk.Label(pos_frame.frame, text = txt).grid(row=row_idx, column=idx)
```

Example 12

Project: *pyktrader2* Author: *harveywwu* File: *gui_agent.py* (license) *View Source Project* 6 votes

```
def tradepos_view(self):
        params = self.app.get_agent_params(['Risk.ppos'])
        res = params['Risk']
        sum_risk = {}
        sum_risk['total'] = res['total']
        strat_list = res['strats'].keys()
        for strat_name in strat_list:
            sum_risk[strat_name] = res['strats'][strat_name]
        pos_win   = tk.Toplevel(self)
        tp_frame = ScrolledFrame(pos_win)
        fields = ['inst', 'total'] + strat_list
        for idx, field in enumerate(fields):
            tk.Label(tp_frame.frame, text = field).grid(row=0, column=idx)
            for idy, inst in enumerate(sum_risk['total'].keys()):
                if field == 'inst':
                    txt = inst
                else:
                    inst_risk = sum_risk[field].get(inst, {})
                    txt = str(inst_risk.get('ppos', 0))
                tk.Label(tp_frame.frame, text = txt).grid(row=idy+1, column=idx)
```

Example 13

Project: *oil* Author: *oilshell* File: *dynOptionMenuWidget.py* (license) *View Source Project* 6 votes

```
def _dyn_option_menu(parent):  # htest #
    from Tkinter import Toplevel

    top = Toplevel()
```

```
    top.title("Tets dynamic option menu")
    top.geometry("200x100+%d+%d" % (parent.winfo_rootx() + 200,
                  parent.winfo_rooty() + 150))
    top.focus_set()

    var = StringVar(top)
    var.set("Old option set") #Set the default value
    dyn = DynOptionMenu(top,var, "old1","old2","old3","old4")
    dyn.pack()

    def update():
        dyn.SetMenu(["new1","new2","new3","new4"], value="new option set")
    button = Button(top, text="Change option set", command=update)
    button.pack()
```

Example 14

Project: *oil* Author: *oilshell* File: *ColorDelegator.py* (license) *View Source Project* 6 votes

```
def _color_delegator(parent):  # htest #
    from Tkinter import Toplevel, Text
    from idlelib.Percolator import Percolator

    top = Toplevel(parent)
    top.title("Test ColorDelegator")
    top.geometry("200x100+%d+%d" % (parent.winfo_rootx() + 200,
                  parent.winfo_rooty() + 150))
    source = "if somename: x = 'abc' # comment\nprint\n"
    text = Text(top, background="white")
    text.pack(expand=1, fill="both")
    text.insert("insert", source)
    text.focus_set()

    p = Percolator(text)
    d = ColorDelegator()
    p.insertfilter(d)
```

Example 15

Project: *oil* Author: *oilshell* File: *CallTipWindow.py* (license) *View Source Project* 6 votes

```
def _calltip_window(parent):  # htest #
    from Tkinter import Toplevel, Text, LEFT, BOTH

    top = Toplevel(parent)
    top.title("Test calltips")
    top.geometry("200x100+%d+%d" % (parent.winfo_rootx() + 200,
                  parent.winfo_rooty() + 150))
    text = Text(top)
    text.pack(side=LEFT, fill=BOTH, expand=1)
    text.insert("insert", "string.split")
    top.update()
    calltip = CallTip(text)

    def calltip_show(event):
        calltip.showtip("(s=Hello world)", "insert", "end")
    def calltip_hide(event):
        calltip.hidetip()
    text.event_add("<<calltip-show>>", "(")
    text.event_add("<<calltip-hide>>", ")")
    text.bind("<<calltip-show>>", calltip_show)
    text.bind("<<calltip-hide>>", calltip_hide)
    text.focus_set()
```

Example 16

Project: *python2-tracer* Author: *extremecoders-re* File: *dynOptionMenuWidget.py* (license) *View Source Project* 6 votes

```
def _dyn_option_menu(parent):  # htest #
    from Tkinter import Toplevel

    top = Toplevel()
    top.title("Tets dynamic option menu")
    top.geometry("200x100+%d+%d" % (parent.winfo_rootx() + 200,
                 parent.winfo_rooty() + 150))
    top.focus_set()

    var = StringVar(top)
    var.set("Old option set") #Set the default value
    dyn = DynOptionMenu(top,var, "old1","old2","old3","old4")
    dyn.pack()

    def update():
        dyn.SetMenu(["new1","new2","new3","new4"], value="new option set")
    button = Button(top, text="Change option set", command=update)
    button.pack()
```

Example 17

Project: *python2-tracer* Author: *extremecoders-re* File: *ColorDelegator.py* (license) *View Source Project*

6 votes

```
def _color_delegator(parent):  # htest #
    from Tkinter import Toplevel, Text
    from idlelib.Percolator import Percolator

    top = Toplevel(parent)
    top.title("Test ColorDelegator")
    top.geometry("200x100+%d+%d" % (parent.winfo_rootx() + 200,
                 parent.winfo_rooty() + 150))
    source = "if somename: x = 'abc' # comment\nprint\n"
    text = Text(top, background="white")
    text.pack(expand=1, fill="both")
    text.insert("insert", source)
    text.focus_set()

    p = Percolator(text)
    d = ColorDelegator()
    p.insertfilter(d)
```

Example 18

Project: *python2-tracer* Author: *extremecoders-re* File: *CallTipWindow.py* (license) *View Source Project*

6 votes

```
def _calltip_window(parent):  # htest #
    from Tkinter import Toplevel, Text, LEFT, BOTH

    top = Toplevel(parent)
    top.title("Test calltips")
    top.geometry("200x100+%d+%d" % (parent.winfo_rootx() + 200,
                 parent.winfo_rooty() + 150))
    text = Text(top)
    text.pack(side=LEFT, fill=BOTH, expand=1)
    text.insert("insert", "string.split")
    top.update()
    calltip = CallTip(text)

    def calltip_show(event):
        calltip.showtip("(s=Hello world)", "insert", "end")
    def calltip_hide(event):
        calltip.hidetip()
    text.event_add("<<calltip-show>>", "(")
    text.event_add("<<calltip-hide>>", ")")
    text.bind("<<calltip-show>>", calltip_show)
    text.bind("<<calltip-hide>>", calltip_hide)
    text.focus_set()
```

Example 19

Project: *PaleoView* Author: *GlobalEcologyLab* File: *paleo_view_v1_2.py* (license) *View Source Project* 6 votes

```
def viewGridPlots(self, parameter_data, region_masks) :
        #print 'TODO: viewGridPlots'

        # Remove existing climate data window
        if hasattr(self, 'view_climate_data_window') :
            self.view_climate_data_window.destroy()

        # Create the view plot window
        self.view_climate_data_window = tk.Toplevel(self)
        self.view_climate_data_window.title('View Map Grid Plots')
        self.view_climate_data_window.transient(self)
        self.view_climate_data_window.focus_set()
        self.current_view_climate_data_window_type = 'map'

        # Create grid plot contents
        self.ignore_grid_plot_config_events = 0
        self.createGridPlots(parameter_data, region_masks)

        # Bind window to resize/configure events
        self.view_climate_data_window.bind('<Configure>', self.__configureViewGridPlotWindow)

    # Step 7 Method: Configure View Grid Plot Window (Event)
```

Example 20

Project: *PaleoView* Author: *GlobalEcologyLab* File: *backend_tkagg.py* (license) *View Source Project* 6 votes

```
def showtip(self, text):
        "Display text in tooltip window"
        self.text = text
        if self.tipwindow or not self.text:
            return
        x, y, _, _ = self.widget.bbox("insert")
        x = x + self.widget.winfo_rootx() + 27
        y = y + self.widget.winfo_rooty()
        self.tipwindow = tw = Tk.Toplevel(self.widget)
        tw.wm_overrideredirect(1)
        tw.wm_geometry("+%d+%d" % (x, y))
        try:
            # For Mac OS
            tw.tk.call("::tk::unsupported::MacWindowStyle",
                       "style", tw._w,
                       "help", "noActivates")
        except Tk.TclError:
            pass
        label = Tk.Label(tw, text=self.text, justify=Tk.LEFT,
                         background="#ffffe0", relief=Tk.SOLID, borderwidth=1,
                         )
        label.pack(ipadx=1)
```

Example 21

Project: *MyKnowledge* Author: *guofei9987* File: *graphics.py* (license) *View Source Project* 6 votes

```
def __init__(self, title="Graphics Window",
                 width=200, height=200, autoflush=True):
        master = tk.Toplevel(_root)
        master.protocol("WM_DELETE_WINDOW", self.close)
        tk.Canvas.__init__(self, master, width=width, height=height)
        self.master.title(title)
        self.pack()
        master.resizable(0,0)
        self.foreground = "black"
        self.items = []
        self.mouseX = None
```

```
        self.mouseY = None
        self.bind("<Button-1>", self._onClick)
        self.bind_all("<Key>", self._onKey)
        self.height = height
        self.width = width
        self.autoflush = autoflush
        self._mouseCallback = None
        self.trans = None
        self.closed = False
        master.lift()
        self.lastKey = ""
        if autoflush: _root.update()
```

Example 22

Project: *aws_lambda_backup_s3* Author: *ogckw* File: *ImageTk.py* (license) *View Source Project* 6 votes

```
def _show(image, title):
    """Helper for the Image.show method."""

    class UI(tkinter.Label):
        def __init__(self, master, im):
            if im.mode == "1":
                self.image = BitmapImage(im, foreground="white", master=master)
            else:
                self.image = PhotoImage(im, master=master)
            tkinter.Label.__init__(self, master, image=self.image,
                                   bg="black", bd=0)

    if not tkinter._default_root:
        raise IOError("tkinter not initialized")
    top = tkinter.Toplevel()
    if title:
        top.title(title)
    UI(top, image).pack()
```

Example 23

Project: *Class_Schedule_Creater* Author: *alexander-hamme* File: *dialogWindow.py* (license) *View Source Project* 6 votes

```
def __init__(self, parent, numbclasses=5):
        self.answer1, self.answer2 = None, None  # so cancel will work
        self.days = None
        self.e_course = None
        self.e_rm_numb = None
        self.e_time = None
        self.e_class_days = None
        self.classes = []
        self.numb_classes = numbclasses
        self.top = tK.Toplevel(parent)

        self.e_courses = []
        self.e_rm_numbs = []
        self.e_start_times = []
        self.e_end_times = []
        self.e_days = []

        self.row_offset = 2

        self.setup(parent, numbclasses)
```

Example 24

Project: *maze-pathfinder* Author: *ivan-ristovic* File: *gui.py* (license) *View Source Project* 6 votes

```
def perform_process(self, function, message):
                # Popup initialization
                popup = Tkinter.Toplevel(self)
                popup.grab_set()
```

```
                popup.wm_title("Working...")
                if sys.platform.startswith('linux'):
                        img = Tkinter.PhotoImage(file = filepath.get_filepath("assets", "icon.png"))
                        popup.tk.call('wm', 'iconphoto', popup._w, img)
                else:
                        popup.iconbitmap(os.path.abspath(filepath.get_filepath("assets", "icon.ico")))
                popup.geometry(
                        '%dx%d+%d+%d' % (250, 50, self.winfo_x() + 50, self.winfo_y() + 100)
                )
                popup.resizable(False, False)
                info_label = Tkinter.Label(popup, text = message)
                info_label.place(relx = 0.5, rely = 0.5, anchor = Tkinter.CENTER)

                # Starting process
                thread = threading.Thread(target = function)
                thread.start()
                while thread.is_alive():
                        popup.update()
                        time.sleep(0.001)
                popup.grab_release()
                popup.destroy()
```

Example 25

Project: *PIEFACE* Author: *jcumby* File: *pieface_gui.py* (license) *View Source Project* 6 votes

```
def raise_message(log):
    if "Label(s) %s are not present" in log.msg:
        box = tk.Toplevel(root)
        box.title('Error')
        message = ttk.Label(box, text = log.msg % log.args)
        labels = {}
        for f in app.filenames:
            labels[os.path.basename(f)] = " ".join(sorted(multiCIF._alllabels(f)))
        advice = ttk.Label(box, text = "Valid labels are:\n{0}".format( "".join( ["{0:40s}: {1:30s}\n".format(p, labels[p]) for p in labe
        tip = ttk.Label(box, text="[ Tip: Regular expressions can also be used to centre labels ]")
        button = ttk.Button(box, text='OK', command= lambda: box.destroy())
        message.grid(row = 0, padx = 5, pady = 5)
        advice.grid(row = 1, padx = 5, pady = 5)
        tip.grid(row=2, padx=5, pady=5)
        button.grid(row = 3, padx = 5, pady = 5)
        root.wait_window(window=box)
    else:
        pass

    #tkMessageBox.showerror('Error',log.msg)
```

Example 26

Project: *serialplot* Author: *crxguy52* File: *cfgWindow.py* (license) *View Source Project* 6 votes

```
def aboutButton(self):
        toplvl = tk.Toplevel()
        toplvl.title('About')

        txt = ttk.Label(toplvl, wraplength=450, text= \
        "This program was written by Victor Zaccardo as a way to familiarize "
        "myself with Python, and also so I don't have to try and read a serial"
        " terminal every time I want to visualize data coming out of a "
        "microcontroller. It's written in Python 2.7, using tkinter, "
        "matplotlib, and pyserial. \n \n I hope it can be helpful with your "
        "embedded projects. If you have any questions or comments, feel free "
        "to contact me at victorzaccardo@gmail.com. Happy plotting!")
        txt.grid(row=0, column=0, padx=5, pady=5)

        closeButton = ttk.Button(toplvl, text='Close', command=toplvl.destroy)
        closeButton.grid(row=1, column=0, pady = 3)

        toplvl.update()
        scrwidth = toplvl.winfo_screenwidth()
```

```
        scrheight = toplvl.winfo_screenheight()
        winwidth = toplvl.winfo_reqwidth()
        winheight = toplvl.winfo_reqheight()
        winposx = int(round(scrwidth/2 - winwidth/2))
        winposy = int(round(scrheight/2 - winheight/2))
        toplvl.geometry('{}x{}+{}+{}'.format(winwidth, winheight, winposx, winposy))
```

Example 27

Project: *ITC110* Author: *SeattleCentral* File: *graphics.py* (license) *View Source Project* 6 votes

```
def __init__(self, title="Graphics Window",
                 width=200, height=200, autoflush=True):
        assert type(title) == type(""), "Title must be a string"
        master = tk.Toplevel(_root)
        master.protocol("WM_DELETE_WINDOW", self.close)
        tk.Canvas.__init__(self, master, width=width, height=height,
                           highlightthickness=0, bd=0)
        self.master.title(title)
        self.pack()
        master.resizable(0,0)
        self.foreground = "black"
        self.items = []
        self.mouseX = None
        self.mouseY = None
        self.bind("<Button-1>", self._onClick)
        self.bind_all("<Key>", self._onKey)
        self.height = int(height)
        self.width = int(width)
        self.autoflush = autoflush
        self._mouseCallback = None
        self.trans = None
        self.closed = False
        master.lift()
        self.lastKey = ""
        if autoflush: _root.update()
```

Example 28

Project: *ITC110* Author: *SeattleCentral* File: *graphics.py* (license) *View Source Project* 6 votes

```
def __init__(self, title="Graphics Window",
                 width=200, height=200, autoflush=True):
        assert type(title) == type(""), "Title must be a string"
        master = tk.Toplevel(_root)
        master.protocol("WM_DELETE_WINDOW", self.close)
        tk.Canvas.__init__(self, master, width=width, height=height,
                           highlightthickness=0, bd=0)
        self.master.title(title)
        self.pack()
        master.resizable(0,0)
        self.foreground = "black"
        self.items = []
        self.mouseX = None
        self.mouseY = None
        self.bind("<Button-1>", self._onClick)
        self.bind_all("<Key>", self._onKey)
        self.height = int(height)
        self.width = int(width)
        self.autoflush = autoflush
        self._mouseCallback = None
        self.trans = None
        self.closed = False
        master.lift()
        self.lastKey = ""
        if autoflush: _root.update()
```

Example 29

6 votes

Project: *temporal-planning* Author: *aig-upf* File: *txt2tags.py* (license) *View Source Project*

```
def scrollwindow(self, txt='no text!', title=''):
                # Create components
                win    = Tkinter.Toplevel() ; win.title(title)
                frame  = Tkinter.Frame(win)
                scroll = Tkinter.Scrollbar(frame)
                text   = Tkinter.Text(frame,yscrollcommand=scroll.set)
                button = Tkinter.Button(win)
                # Config
                text.insert(Tkinter.END, '\n'.join(txt))
                scroll.config(command=text.yview)
                button.config(text=_('Close'), command=win.destroy)
                button.focus_set()
                # Packing
                text.pack(side='left', fill='both', expand=1)
                scroll.pack(side='right', fill='y')
                frame.pack(fill='both', expand=1)
                button.pack(ipadx=30)
```

Example 30

Project: *HangmanGUI* Author: *devhid* File: *graphics.py* (license) *View Source Project* 6 votes

```
def __init__(self, title="Graphics Window",
                 width=200, height=200, autoflush=True):
        assert type(title) == type(""), "Title must be a string"
        master = tk.Toplevel(_root)
        master.protocol("WM_DELETE_WINDOW", self.close)
        tk.Canvas.__init__(self, master, width=width, height=height,
                           highlightthickness=0, bd=0)
        self.master.title(title)
        self.pack()
        master.resizable(0,0)
        self.foreground = "black"
        self.items = []
        self.mouseX = None
        self.mouseY = None
        self.bind("<Button-1>", self._onClick)
        self.bind_all("<Key>", self._onKey)
        self.height = int(height)
        self.width = int(width)
        self.autoflush = autoflush
        self._mouseCallback = None
        self.trans = None
        self.closed = False
        master.lift()
        self.lastKey = ""
        if autoflush: _root.update()
```

Example 31

Project: *fast-downward* Author: *danfis* File: *txt2tags.py* (license) *View Source Project* 6 votes

```
def scrollwindow(self, txt='no text!', title=''):
                # Create components
                win    = Tkinter.Toplevel() ; win.title(title)
                frame  = Tkinter.Frame(win)
                scroll = Tkinter.Scrollbar(frame)
                text   = Tkinter.Text(frame,yscrollcommand=scroll.set)
                button = Tkinter.Button(win)
                # Config
                text.insert(Tkinter.END, '\n'.join(txt))
                scroll.config(command=text.yview)
                button.config(text=_('Close'), command=win.destroy)
                button.focus_set()
                # Packing
                text.pack(side='left', fill='both', expand=1)
                scroll.pack(side='right', fill='y')
```

```
            frame.pack(fill='both', expand=1)
            button.pack(ipadx=30)
```

Example 32

Project: *CNCGToolKit* Author: *cineuse* File: *ImageTk.py* (license) *View Source Project* 6 votes

```
def _show(image, title):

    class UI(Tkinter.Label):
        def __init__(self, master, im):
            if im.mode == "1":
                self.image = BitmapImage(im, foreground="white", master=master)
            else:
                self.image = PhotoImage(im, master=master)
            Tkinter.Label.__init__(self, master, image=self.image,
                bg="black", bd=0)

    if not Tkinter._default_root:
        raise IOError, "tkinter not initialized"
    top = Tkinter.Toplevel()
    if title:
        top.title(title)
    UI(top, image).pack()
```

Example 33

Project: *face_rekognition* Author: *cnidus* File: *ImageTk.py* (license) *View Source Project* 6 votes

```
def _show(image, title):
    """Helper for the Image.show method."""

    class UI(tkinter.Label):
        def __init__(self, master, im):
            if im.mode == "1":
                self.image = BitmapImage(im, foreground="white", master=master)
            else:
                self.image = PhotoImage(im, master=master)
            tkinter.Label.__init__(self, master, image=self.image,
                                   bg="black", bd=0)

    if not tkinter._default_root:
        raise IOError("tkinter not initialized")
    top = tkinter.Toplevel()
    if title:
        top.title(title)
    UI(top, image).pack()
```

Example 34

Project: *Pong* Author: *MichaelGardiner97* File: *graphics.py* (license) *View Source Project* 6 votes

```
def __init__(self, title="Graphics Window",
                 width=200, height=200, autoflush=True):
        master = tk.Toplevel(_root)
        master.protocol("WM_DELETE_WINDOW", self.close)
        tk.Canvas.__init__(self, master, width=width, height=height)
        self.master.title(title)
        self.pack()
        master.resizable(0,0)
        self.foreground = "black"
        self.items = []
        self.mouseX = None
        self.mouseY = None
        self.bind("<Button-1>", self._onClick)
        self.height = height
        self.width = width
        self.autoflush = autoflush
        self._mouseCallback = None
        self.trans = None
```

```
        self.closed = False
        master.lift()
        if autoflush: _root.update()
```

Example 35

Project: *rig-remote* Author: *Marzona* File: *utility.py* (license) *View Source Project* 6 votes

```
def _show(self):
        if self._opts['state'] == 'disabled':
            self._unschedule()
            return
        if not self._tipwindow:
            self._tipwindow = tw = tk.Toplevel(self.master)
            # hide the window until we know the geometry
            tw.withdraw()
            tw.wm_overrideredirect(1)

            if tw.tk.call("tk", "windowingsystem") == 'aqua':
                tw.tk.call("::tk::unsupported::MacWindowStyle",
                           "style",
                           tw._w,
                           "help",
                           "none")

            self.create_contents()
            tw.update_idletasks()
            x, y = self.coords()
            tw.wm_geometry("+%d+%d" % (x, y))
            tw.deiconify()
```

Example 36

Project: *ProjectOfDataMining* Author: *IljaNovo* File: *graphics.py* (license) *View Source Project* 6 votes

```
def __init__(self, title="Graphics Window",
                 width=200, height=200, autoflush=True):
        master = tk.Toplevel(_root)
        master.protocol("WM_DELETE_WINDOW", self.close)
        tk.Canvas.__init__(self, master, width=width, height=height)
        self.master.title(title)
        self.pack()
        master.resizable(0,0)
        self.foreground = "black"
        self.items = []
        self.mouseX = None
        self.mouseY = None
        self.bind("<Button-1>", self._onClick)
        self.bind_all("<Key>", self._onKey)
        self.height = height
        self.width = width
        self.autoflush = autoflush
        self._mouseCallback = None
        self.trans = None
        self.closed = False
        master.lift()
        self.lastKey = ""
        if autoflush: _root.update()
```

Example 37

Project: *InstagramPosting* Author: *LeviParadis* File: *ImageTk.py* (license) *View Source Project* 6 votes

```
def _show(image, title):

    class UI(Tkinter.Label):
        def __init__(self, master, im):
            if im.mode == "1":
                self.image = BitmapImage(im, foreground="white", master=master)
            else:
```

```
            self.image = PhotoImage(im, master=master)
        Tkinter.Label.__init__(self, master, image=self.image,
            bg="black", bd=0)

    if not Tkinter._default_root:
        raise IOError, "tkinter not initialized"
    top = Tkinter.Toplevel()
    if title:
        top.title(title)
    UI(top, image).pack()
```

Example 38

Project: *ngx_status* Author: *YoYoAdorkable* File: *ImageTk.py* (license) *View Source Project* 6 votes

```
def _show(image, title):

    class UI(Tkinter.Label):
        def __init__(self, master, im):
            if im.mode == "1":
                self.image = BitmapImage(im, foreground="white", master=master)
            else:
                self.image = PhotoImage(im, master=master)
            Tkinter.Label.__init__(self, master, image=self.image,
                bg="black", bd=0)

    if not Tkinter._default_root:
        raise IOError, "tkinter not initialized"
    top = Tkinter.Toplevel()
    if title:
        top.title(title)
    UI(top, image).pack()
```

Example 39

Project: *code* Author: *ActiveState* File: *recipe-579107.py* (MIT License) *View Source Project* 5 votes

```
def textwindow(url):
    title = url
    h = html2text.HTML2Text()
    h.ignore_links = True
    h.ignore_images = True
    s = gethtml(url)
    s = h.handle(s)
    s = h.unescape(s)
    text = convert65536(s)
    top = Tkinter.Toplevel()
    top.geometry("+200+100")
    top.title(title)
    top.bind("<Escape>", lambda _ : top.destroy())
    S = Tkinter.Scrollbar(top)
    customFont = tkFont.Font(family="Arial", size=16)
    T = TextPlus(top,height=20,width=78,font=customFont,bg="lightgrey")
    S.pack(side=Tkinter.RIGHT,fill=Tkinter.Y)
    T.pack(side=Tkinter.LEFT,fill=Tkinter.Y)
    S.config(command=T.yview)
    T.config(yscrollcommand=S.set)
    T.insert(Tkinter.END,text)
```

Example 40

Project: *BISIP* Author: *clberube* File: *GUI.py* (MIT License) *View Source Project* 5 votes

```
def make_options(self, from_root=False):
        try: self.top_checkbox.destroy()
        except: pass

        # Select all or none functions
        def select_all(): # self expl
            for v in list(self.save_options.values()):
```

```
            v.set(True)
        def select_none(): # self expl
            for v in list(self.save_options.values()):
                v.set(False)
        if not from_root:
            self.top_checkbox = tk.Toplevel(self.master)
            self.top_checkbox.title("Save options")
            frame_checkbox = tk.LabelFrame(self.top_checkbox, text="Check items to save", width=200, height=4, font=fontz["bold"])
            frame_checkbox.grid(row = 0, column=0,sticky=tk.S+tk.W+tk.E+tk.N, padx=10, pady=10)
            tk.Button(frame_checkbox, height=1, width=15, text="Check all", command=select_all, bg='gray97', font=fontz["normal_small"],
            tk.Button(frame_checkbox, height=1, width=15, text="Check none", command=select_none, bg='gray97', font=fontz["normal_small"]
            button = tk.Button(frame_checkbox, height=2, width=20, text="OK", bg='gray97', font=fontz["bold"], command=self.top_checkbox.
            button.grid(row=len(self.save_options)+2, column=0, columnspan=2, sticky=tk.W+tk.E, pady=(10,10), padx=(20,20))
        for i, (k, v) in enumerate(sorted(self.save_options.items())):
            if from_root:
                self.save_options[k].set(self.root_ini[k])
            if not from_root:
                tk.Checkbutton(frame_checkbox, text=k, variable=v).grid(row=i+1, column=0, sticky=tk.W+tk.N+tk.S,padx=(10,10),pady=(5,0))

#============================================================================
# Plotting
#============================================================================
```

Example 41

Project: *BISIP* Author: *clberube* File: *GUI.py* (MIT License) *View Source Project* 5 votes

```
def popup(self, title, message, size=10):
        top = tk.Toplevel()
        top.title(title)
        about_message = (message)
        top.lift()
        msg = tk.Text(top, width=90, font=('courier', size, 'normal'))
        msg.grid(stick=tk.N, padx=(10,10), pady=(10,10))
        msg.insert("1.0", about_message)
        button = tk.Button(top, height=1, width=20, text="Dismiss", command=top.destroy, bg='gray97', relief=tk.GROOVE)
        button.grid(sticky=tk.S, pady=(0,10))
        s = tk.Scrollbar(top, width=20)
        s.grid(row=0, column=0, sticky=tk.E+tk.N+tk.S, padx=(0,10),pady=(10,10))
        s['command'] = msg.yview
        msg['yscrollcommand'] = s.set
        top.resizable(width=tk.FALSE, height=tk.FALSE)
```

Example 42

Project: *BISIP* Author: *clberube* File: *GUI.py* (MIT License) *View Source Project* 5 votes

```
def make_options(self, from_root=False):
        try: self.top_checkbox.destroy()
        except: pass

        # Select all or none functions
        def select_all(): # self expl
            for v in list(self.save_options.values()):
                v.set(True)
        def select_none(): # self expl
            for v in list(self.save_options.values()):
                v.set(False)
        if not from_root:
            self.top_checkbox = tk.Toplevel(self.master)
            self.top_checkbox.title("Save options")
            frame_checkbox = tk.LabelFrame(self.top_checkbox, text="Check items to save", width=200, height=4, font=fontz["bold"])
            frame_checkbox.grid(row = 0, column=0,sticky=tk.S+tk.W+tk.E+tk.N, padx=10, pady=10)
            tk.Button(frame_checkbox, height=1, width=15, text="Check all", command=select_all, bg='gray97', font=fontz["normal_small"],
            tk.Button(frame_checkbox, height=1, width=15, text="Check none", command=select_none, bg='gray97', font=fontz["normal_small"]
            button = tk.Button(frame_checkbox, height=2, width=20, text="OK", bg='gray97', font=fontz["bold"], command=self.top_checkbox.
            button.grid(row=len(self.save_options)+2, column=0, columnspan=2, sticky=tk.W+tk.E, pady=(10,10), padx=(20,20))
        for i, (k, v) in enumerate(sorted(self.save_options.items())):
            if from_root:
                self.save_options[k].set(self.root_ini[k])
```

```
            if not from_root:
                tk.Checkbutton(frame_checkbox, text=k, variable=v).grid(row=i+1, column=0, sticky=tk.W+tk.N+tk.S,padx=(10,10),pady=(5,0))

#==============================================================================
# Plotting
#==============================================================================
```

Example 43

Project: *BISIP* Author: *clberube* File: *GUI.py* (MIT License) *View Source Project* 5 votes

```
def popup(self, title, message, size=10):
        top = tk.Toplevel()
        top.title(title)
        about_message = (message)
        top.lift()
        msg = tk.Text(top, width=90, font=('courier', size, 'normal'))
        msg.grid(stick=tk.N, padx=(10,10), pady=(10,10))
        msg.insert("1.0", about_message)
        button = tk.Button(top, height=1, width=20, text="Dismiss", command=top.destroy, bg='gray97', relief=tk.GROOVE)
        button.grid(sticky=tk.S, pady=(0,10))
        s = tk.Scrollbar(top, width=20)
        s.grid(row=0, column=0, sticky=tk.E+tk.N+tk.S, padx=(0,10),pady=(10,10))
        s['command'] = msg.yview
        msg['yscrollcommand'] = s.set
        top.resizable(width=tk.FALSE, height=tk.FALSE)
```

Example 44

Project: *darkc0de-old-stuff* Author: *tuwid* File: *tooltip.py* (GNU General Public License v3.0) *View Source Project* 5 votes

```
def _show(self):
        if self._opts['state'] == 'disabled':
            self._unschedule()
            return
        if not self._tipwindow:
            self._tipwindow = tw = Tkinter.Toplevel(self.master)
            # hide the window until we know the geometry
            tw.withdraw()
            tw.wm_overrideredirect(1)
            self.create_contents()
            tw.update_idletasks()
            x, y = self.coords()
            tw.wm_geometry("+%d+%d" % (x, y))
            tw.deiconify()
```

Example 45

Project: *Scoary* Author: *AdmiralenOla* File: *GUI.py* (license) *View Source Project* 5 votes

```
def AboutScoary(self):
        """
        Placeholder button. Planned short information about the method
        """
        topwin = Tkinter.Toplevel(self)
        button = \
            Tkinter.Button(topwin,
            text=str("https://github.com/AdmiralenOla/Scoary"))
        button.pack()
```

Example 46

Project: *PythonGraphics* Author: *SethDamiani* File: *graphics.py* (license) *View Source Project* 5 votes

```
def __init__(self, title="Graphics Window", width=200, height=200, autoflush=True):
        assert isinstance(title, str), "Title must be a string"
        master = tk.Toplevel(_root)
        master.protocol("WM_DELETE_WINDOW", self.close)
        tk.Canvas.__init__(self, master, width=width, height=height, highlightthickness=0, bd=0)
```

```
        self.master.title(title)
        self.pack()
        master.resizable(0, 0)
        self.foreground = "black"
        self.items = []
        self.mouseX1 = None
        self.mouseY1 = None
        self.mouseX2 = None
        self.mouseY2 = None
        self.bind("<Button-1>", self._onClick1)
        self.bind("<Button-2>", self._onClick2)
        self.bind_all("<Key>", self._onKey)
        self.height = height
        self.width = width
        self.autoflush = autoflush
        self._mouseCallback = None
        self._mouseCallback2 = None
        self.trans = None
        self.closed = False
        master.lift()
        self.lastKey = ""
        if autoflush:
            _root.update()
```

Example 47

Project: *Circadia* Author: *hooyah* File: *edit.py* (license) *View Source Project* 5 votes

```
def aboutDlg(self):

        toplevel = tk.Toplevel()

        scrollbar = tk.Scrollbar(toplevel)
        scrollbar.pack(side='right', fill='y')

        aboutTxt = tk.Text(toplevel, height=40, width=100, yscrollcommand=scrollbar.set)
        aboutTxt.insert(tk.INSERT,editAbout.ABOUTTEXT)
        aboutTxt.config(state=tk.DISABLED)
        aboutTxt.pack()

        aboutTxt.master.title('Circadia Theme Editor - %s'%__version__)
        scrollbar.config(command=aboutTxt.yview)




##########################################################################################################################
##########################################################################################################################
```

Example 48

Project: *ecel* Author: *ARL-UTEP-OC* File: *Pmw.py* (license) *View Source Project* 5 votes

```
def __TkinterToplevelTitle(self, *args):
    # If this is being called from the constructor, include this
    #  Toplevel in the list of toplevels and set the initial
    # WM_DELETE_WINDOW protocol to destroy() so that we get to know
    # about it.
    if not _toplevelBusyInfo.has_key(self):
        _addToplevelBusyInfo(self)
        self._Pmw_WM_DELETE_name = self.register(self.destroy, None, 0)
        self.protocol('WM_DELETE_WINDOW', self._Pmw_WM_DELETE_name)

    return apply(Tkinter.Wm.title, (self,) + args)
```

Example 49

Project: *ecel* Author: *ARL-UTEP-OC* File: *Pmw.py* (license) *View Source Project* 5 votes

```
def __init__(self):

        self._errorQueue = []
        self._errorCount = 0
        self._open = 0
        self._firstShowing = 1

        # Create the toplevel window
        self._top = Tkinter.Toplevel()
        self._top.protocol('WM_DELETE_WINDOW', self._hide)
        self._top.title('Error in background function')
        self._top.iconname('Background error')

        # Create the text widget and scrollbar in a frame
        upperframe = Tkinter.Frame(self._top)

        scrollbar = Tkinter.Scrollbar(upperframe, orient='vertical')
        scrollbar.pack(side = 'right', fill = 'y')

        self._text = Tkinter.Text(upperframe, yscrollcommand=scrollbar.set)
        self._text.pack(fill = 'both', expand = 1)
        scrollbar.configure(command=self._text.yview)

        # Create the buttons and label in a frame
        lowerframe = Tkinter.Frame(self._top)

        ignore = Tkinter.Button(lowerframe,
                text = 'Ignore remaining errors', command = self._hide)
        ignore.pack(side='left')

        self._nextError = Tkinter.Button(lowerframe,
                text = 'Show next error', command = self._next)
        self._nextError.pack(side='left')

        self._label = Tkinter.Label(lowerframe, relief='ridge')
        self._label.pack(side='left', fill='x', expand=1)

        # Pack the lower frame first so that it does not disappear
        # when the window is resized.
        lowerframe.pack(side = 'bottom', fill = 'x')
        upperframe.pack(side = 'bottom', fill = 'both', expand = 1)
```

Example 50

Project: *ecel* Author: *ARL-UTEP-OC* File: *Pmw.py* (license) *View Source Project* 5 votes

```
def __init__(self, parent = None, **kw):
        # Define the megawidget options.

        optiondefs = (
            ('borderx',     10,    INITOPT),
            ('bordery',     10,    INITOPT),
        )
        self.defineoptions(kw, optiondefs)

        # Initialise the base class (after defining the options).
        Dialog.__init__(self, parent)

        # Create the components.
        interior = self.interior()
        aliases = (
            ('listbox', 'scrolledlist_listbox'),
            ('label', 'scrolledlist_label'),
        )
        self._list = self.createcomponent('scrolledlist',
                aliases, None,
                ScrolledListBox, (interior,),
                dblclickcommand = self.invoke)
        self._list.pack(side='top', expand='true', fill='both',
                padx = self['borderx'], pady = self['bordery'])
```

```
        if not kw.has_key('activatecommand'):
            # Whenever this dialog is activated, set the focus to the
            # ScrolledListBox's listbox widget.
            listbox = self.component('listbox')
            self.configure(activatecommand = listbox.focus_set)

        # Check keywords and initialise options.
        self.initialiseoptions()

    # Need to explicitly forward this to override the stupid
    # (grid_)size method inherited from Tkinter.Toplevel.Grid.
```