Contents

Easytk Docs

Rafugafu

Note: You can always combine easytk with normal tkinter.
Note: These are the Easytk docs, not the tkinter docs. Anything of tkinter will not be explained. You can find the tkinter docs here.
Note: Most of the examples are not full code. They usually need to be preceded by `import {modules used} ; root = easytk.win()` and succeeded by `root.show()`

1 Initialization

To make a window, use

```
import easytk
root = easytk.win()
```

This is equivalent to

```
import tkinter
root = tkinter.Tk()
```

To show the window at the end, you can use `root.show()` instead of `root.mainloop()`. If you want to make a subwindow (toplevel) of an already existing window `root`, use `subwin = root.subwin()` instead. Then you can use the `subwin` normally.

2 Widgets

These are all the widgets:

- button
- text
- textbox
- entry
- dropdown
- droptype
- label

- check
- multiplechoice
- image
- menu
- tabs
- frame
- slider
- progressbar
- scroll
- listbox
- spinbox
- separator

Note: Some of these widgets have the same name as a different one in normal tkinter. Do not get confused.
The widgets are part of the window class itself, so use `w = root.{widget}({args})`, not `w = easytk.{widget}({args})`. You also do not need to specify the master. If you want to put the widget inside a frame inside the window, you can specify the master to be the frame.
Example:

```
frame = root.frame()
w = root.{widget}(master = frame, {args})
```

Then you can `grid`, `pack`, or `place` it normally.
You can treat any widget, once made, exactly like a normal tkinter widget.
Example:

```
b = root.button(text = 'Click Me!',
command = lambda: b.config(state = 'disabled'))
b.grid()
```

Now here are specifications and examples for each one:

2.1 Button

Use `b = root.button({args})` to make a button in a window called `root`.
Example:

```
b = root.button(text = 'Click Me!')
b.grid()
```

There are two extra args: `image` and `imsize`. If you want to put an image in the button, you can use the `image` and `imsize` args like this:
`b = root.button(image = 'abc.png')`. Using the `imsize` arg is not necessary. The default is `(10, 10)`.
Example:

```
b = root.button(image = 'abc.png', imsize = (10, 10))
b.grid()
```

Note: The higher the values given to the `imsize`, the smaller the image becomes.

2.2 Text

Use `t = root.text({args})` to write some text in a window called `root`.
Example:

```
t = root.text(text = 'Hello!')
t.grid()
```

There are two extra args: `image` and `imsize`. If you want to put an image in the text, you can use the `image` and `imsize` args like this:
`t = root.text(image = 'abc.png')`. Using the `imsize` arg is not necessary. The default is `(10, 10)`.
Example:

```
t = root.text(image = 'abc.png', imsize = (10, 10))
t.grid()
```

Note: The higher the values given to the `imsize`, the smaller the image becomes.
Note: This `Text` is the same as the normal tkinter `Label`.

2.3 Textbox

Use `tb = root.textbox({args})` to make a textbox in a window called `root`.
Example:

```
tb = root.textbox(wrap = 'word')
tb.grid()
```

The only extra arg is `scrolled`. It is `False` by default, but if it is set to `True` the textbox will have a scrollbar.
Example:

```
tb = root.textbox(scrolled = True)
tb.grid()
```

Note: This `Textbox` is the same as the normal tkinter `Text`.

2.4 Entry

Use `e = root.entry({args})` to make an entry widget in a window called `root`.
Example:

```
e = root.entry()
e.grid()
```

The args are the same as normal tkinter.

2.5 Dropdown

Use `d = root.dropdown({args})` to make a dropdown in a window called `root`.
Example:

```
s = root.stringvar()
d = root.dropdown(options = ['Option 1', 'Option 2', 'Option 3'],
stringvar = s, showdefault = 'Option 1')
d.grid()
```

There are a few differences in the args: showdefault and options. The `showdefault` option sets the default text to be displayed on the dropdown before you change it. It does not need to be part of the options. The `options` arg takes a `list` or a `tuple` and sets the possible options in the dropdown menu.
Example:

```
s = root.stringvar()
d = root.dropdown(options = ['Option 1', 'Option 2', 'Option 3'],
stringvar = s, showdefault = 'Click Me!')
d.grid()
```

Note: This `Dropdown` is the same as the normal tkinter `OptionMenu`.

2.6 Droptype

Use `d = root.droptype({args})` to make a droptype in a window called `root`.
Example:

```
d = root.droptype(options = [1, 2, 3])
d.grid()
```

There is one extra arg: `command`. The `command` option triggers a given function when the value of the droptype is changed, *only by clicking*. If you change it with the keyboard, the function will not be called.
Example:

```
d = root.droptype(options = [1, 2, 3], command = lambda: print(d.get()))
d.grid()
```

Note: This `Droptype` is the same as the normal tkinter `Combobox`.

2.7 Label

This is actually not a widget in normal tkinter. This is a label that pops up when you hover your mouse over the widget it is attached to.
Use `root.label({widget}, {text}, {args})` to attach a label to the given `widget`.
Example:

```
b = root.button(text = 'Click Me!')
root.label(b, 'This is a button.')
b.grid()
```

The args are: `color`, `border`, and `master`.
The `color` arg sets the color of the label when it shows. The default is light yellow. It can be in any format tkinter recognizes.
The `border` arg sets how the border looks. It can be either of (`flat`, `raised`, `sunken`, `groove`, `ridge`). The default is `raised`.
The `master` arg sets the window it will appear in. It is very important to set it to be the same as the widget you are attaching it to.
Example:

```
f = root.frame()
b = root.button(master = f, text = 'Click Me!')
root.label(b, 'This is a button.', color = 'blue',
border = 'sunken', master = f)
f.grid()
b.grid()
```

Note: If you copy this code, the frame will only be big enough for the button. You will manually have to expand the frame to leave enough space for the label.

2.8 Check

Use `c = root.check({args})` to make a checkbutton in a window called `root`.
Example:

```
b = root.booleanvar()
c = root.check(text = 'Click Me!', variable = b,
command = lambda: print(b.get()))
c.grid()
```

The args are the same as normal tkinter.

2.9 Multiplechoice

Use `m = root.multiplechoice({args})` to make a multiplechoice option in a window called `root`.
Example:

```
v = root.stringvar()
root.multiplechoice(text = 'Option 1', var = v,
command = lambda: print(v.get(), 'is selected')).grid()
root.multiplechoice(text = 'Option 2', var = v,
command = lambda: print(v.get(), 'is selected')).grid()
```

There are a few differences in the args from normal tkinter: `variable` is renamed to `var`, and the `value` option does not exist. The `value` is automatically set to the `text`. The rest of the args are the same.
Note: This `Multiplechoice` is the same as the normal tkinter `Radiobutton`.

2.10 Image

This is actually not a widget in normal tkinter. This is an image that you can put anywhere in a window.
Use `i = root.image({args})` to make an image in a window called `root`.
Example:

```
root.image(image = 'abc.png', imsize = (1, 1)).grid()
```

The args are: `image`, and `imsize`. The `image` arg specifies the filename of the image. The `imsize` arg is not neccesary.
Note: The higher the values given to the `imsize`, the smaller the image becomes.

2.11 Menu

Use `m = root.menu({args})` to make a menu in a window called `root`.
Example:

```
m = root.menu()
a = root.menu(m)
m.add_cascade(menu = a, label = 'Menu A')
a.add_command(label = 'Hello!', command = None)
root.config(menu = m)
```

The args are the same as normal tkinter.

2.12 Tabs

Use `tabs = root.tabs({args})` to make a menu in a window called `root`.
Example:

```
tabs = root.tabs()
t1 = root.frame(tabs)
t2 = root.frame(tabs)
root.text(master = t1, text = 'Tab 1.').grid()
root.text(master = t2, text = 'Tab 2.').grid()
```

```
tabs.add(t1, text = 'Tab 1')
tabs.add(t2, text = 'Tab 2')
tabs.grid()
```

The args are the same as normal tkinter.
Note: This `Tabs` is the same as the normal tkinter `Notebook`.

2.13 Frame

Use `f = root.frame({args})` to make a frame in a window called `root`.
Example:

```
f = root.frame({args})
root.button(master = f, text = 'Click Me!').grid()
f.grid()
```

The args are the same as normal tkinter.

2.14 Slider

Use `s = root.slider({args})` to make a slider in a window called `root`.
Example:

```
val = root.text(text = 'Value: 0')
val.grid()
s = root.slider(range_ = (0, 100),
command = lambda v: val.config(text = f'Value: {round(float(v))}'))
s.grid()
```

There are a few differences in the args: `from_` and `to` has become `range_`. The `range_` arg takes a `list` or a `tuple` in the form `(n_1, n_2)` and creates a slider ranged from `n_1` to `n_2`. The rest of the args are the same.
Note: This `Slider` is the same as the normal tkinter `Scale`.

2.15 Progressbar

Use `p = root.progressbar({args})` to make a progressbar in a window called `root`.
Example:

```
def setp(val):
        p['value'] = val

p = root.progressbar(length = 100)
p.grid()
s = root.slider(range_ = (0, 100), command = setp)
s.grid()
```

The args are the same as normal tkinter.

2.16 Scroll

Use `s = root.scroll({args})` to make a scrollbar in a window called `root`.
Example:

```
s = root.scroll()
s.pack(side = 'right', fill = 'y')
l = root.listbox(contents = [f'Item {i}' for i in range(1, 26)],
yscrollcommand = s.set)
l.pack(side = 'left', fill = 'both')
s.config(command = l.yview)
```

The args are the same as normal tkinter.
Note: This `Scroll` is the same as the normal tkinter `Scrollbar`.

2.17 Listbox

Use `l = root.listbox({args})` to make a listbox in a window called `root`.
Example:

```
l = root.listbox(contents = [f'Item {i}' for i in range(1, 26)])
l.grid()
```

There is one extra arg: `contents`. This takes a `list` or a `tuple` and makes a listbox with the given contents already made.

2.18 Spinbox

Use `s = root.spinbox({args})` to make a spinbox in a window called `root`.
Example:

```
s = root.spinbox(range_ = (0, 100))
s.grid()
```

There are a few differences in the args: `from_` and `to` has become `range_`. The `range_` arg takes a `list` or a `tuple` in the form `(n_1, n_2)` and creates a spinbox ranged from `n_1` to `n_2`. The rest of the args are the same.

2.19 Separator

Use `s = root.separator({args})` to make a separator in a window called `root`.
Example:

```
s = root.separator()
s.grid()
```

There is one difference in the args: `orient` has become `way`.
Example:

```
s = root.separator(way = 'vertical')
s.grid()
```

3 Functions

These are all the extra or modified functions in easytk:

- getfonts
- style
- themes
- error
- info
- warning
- ask
- askstring
- askcolor
- stringvar
- intvar
- booleanvar
- seticon
- getdir
- openfile
- savefile
- sizablefalse
- sizabletrue
- show
- gettheme
- getgeo

Now here are specifications and examples for each one:

3.1 Getfonts

Use `root.getfonts()` to get a list of all the installed fonts that can be used with the widgets.
Example:

```
root.text(text = 'The installed fonts are:').grid()
root.listbox(contents = root.getfonts()).grid()
```

3.2 Style

The `style` function can be used to set a theme, or return a ttk `Style` or ttkthemes `ThemedStyle` object. It takes two args: `style`, and `master`. If the `master` arg is not given, it will be set to the window you are calling the function with.
Example:
If you are calling the `style` function with a window called `root`: `root.style()`, and the `master` arg is not given, it will default to `root`.
If the `style` arg is not given and you have installed `ttkthemes`, it will return a ttkthemes `ThemedStyle` object. If you have not installed `ttkthemes`, it will return a ttk `Style` object.
If the `style` arg is given, the function will set the theme of the window and all the widgets to the given `style`.
Example:
`root.style(style = 'clam')` will set the theme of the window to `clam`.
Note: The `style` function can set the theme of the window from `ttkthemes` as well, if you have it installed.

3.3 Themes

Use `root.themes()` to get a list of all the usable themes.
Example:

```
root.text(text = 'Theme:').grid(column = 0, row = 0)
s = root.stringvar()
root.dropdown(options = root.themes(), stringvar = s,
showdefault = root.gettheme(),
command = lambda x: root.style(x)).grid(column = 1, row = 0)
```

The `themes` function can also find themes from `ttkthemes` if you have it installed.

3.4 Error

Use `root.error({args})` to show an error message from a window called `root`.
Example:

```
root.error(title = 'Error!', message = 'There was an error:\n{error}')
```

The args are: `title` and `message`.
Note: This is equivalent to
`from tkinter import messagebox as mb; mb.showerror({args})`, but with a style.

3.5 Info

Use `root.info({args})` to show an info dialogue box from a window called `root`.

Example:

```
root.info(title = 'Info!', message = 'Finished process successfully!')
```

The args are `title` and `message`.
Note: This is equivalent to
`from tkinter import messagebox as mb; mb.showinfo({args})`, but with a style.

3.6 Warning

Use `root.warning({args})` to show a warning message from a window called `root`.
Example:

```
root.warning(title = 'Warning!', message = 'Warning:\n{warning}.')
```

The args are `title` and `message`.
Note: This is equivalent to
`from tkinter import messagebox as mb; mb.showwarning({args})`, but with a style.

3.7 Ask

Use `root.ask({args})` to show a dialogue box asking something from a window called `root`. The args are: `title`, `question`, and `options`. The `title` and `message` args take an `str` as an input and make a dialogue box with the given title and message. The `options` arg takes a `list` or a `tuple` as an input with the contents: `('yes', 'no')`, `('ok', 'cancel')`, `('yes', 'no', 'cancel')`, or `('retry', 'cancel')`. The rest of the args are the same as `tkinter.messagebox.askyesno` or whatever corresponds to the given `options`.
Example:

```
root.ask(title = '', question = "The document '{doc}' contains\
unsaved changes.\n Do you want to save before closing?",
options = ('yes', 'no', 'cancel'))
```

The function returns the same output as the corresponding function in normal tkinter.

3.8 Askstring

Use `root.askstring({args})` to show a dialogue box asking for a string input. The args are: `title` and `prompt`.
Example:

```
root.askstring(title = '', prompt = 'What is your name: ')
```

This will show a dialogue box asking for your name with the buttons `Ok` and `Cancel`. If you click `Cancel` or close the window, it will return `None`. If you fill in the entry and click `Ok`, it will return your answer.

3.9 Askcolor

Use `root.askcolor({args})` to show a color picker.
Example:

```
print(root.askcolor())
```

If you select a color and click on `Okay`, it will return the color you chose in a hex format. If you close the window or click `Cancel`, it will return `None`.

3.10 Stringvar

Use `root.stringvar({args})` to get a `tkinter.StringVar({args})`. The `master` arg, if not given, will default to `root` in the `tkinter.StringVar` that is returned.
Example:

```
s = root.stringvar()
d = root.dropdown(options = ['Option 1', 'Option 2', 'Option 3'],
stringvar = s, showdefault = 'Click Me!')
d.grid()
```

3.11 Intvar

Use `root.intvar({args})` to get a `tkinter.IntVar({args})`. The `master` arg, if not given, will default to `root` in the `tkinter.StringVar` that is returned.
Example:

```
i = root.intvar()
s = root.spinbox(range_ = (0, 100), textvariable = i)
s.grid()
```

3.12 Booleanvar

Use `root.booleanvar({args})` to get a `tkinter.BooleanVar({args})`. The `master` arg, if not given, will default to `root` in the `tkinter.BooleanVar` that is returned.
Example:

```
b = root.booleanvar()
c = root.check(text = 'Click Me!', variable = b,
command = lambda: print(b.get()))
c.grid()
```

3.13 Seticon

Use `root.seticon({image})` to set the icon of a window called `root`.
Example:

```
root.seticon('abc.png')
```

You do not have to give it a `tkinter.PhotoImage` as an input. Instead, it takes the filename of an image as an input.

3.14 Getdir

Use `root.getdir({args})` to ask for a directory from the user. The args are the same as `tkinter.filedialog.askdirectory`.
Example:

```
curdir = os.getcwd()
def chdir():
        global curdir
        newdir = root.getdir()
        if newdir:
                curdir = newdir
                refresh()

def refresh():
        l.delete(0, 'end')
        for file in os.listdir(curdir):
                l.insert('end', file)

root.button(text = 'Change Directory', command = chdir).grid()
root.text(text = 'Files:').grid()
l = root.listbox(contents = os.listdir(curdir))
l.grid()
```

3.15 Openfile

Use `root.openfile({types})` to open a file picker. You can filter the types of files it is allowed to open by giving the optional `types` arg. The `types` arg takes a list or a tuple. The default is `('all')`. If you have more than one, you will get a dropdown menu in the bottom selecting the filters.
Example:
If you have given `root.openfile(types = ('py', 'txt'))`, then you will get a file picker with the default filter being `*.py`, but you will be able to select `*.txt` as well. If you have selected `'all'`, then there will be no filters.
If you select a file and click on `Okay` or double click on a file, it will return the file name. If you close the window or click on `Cancel`, it will return `None`.
Example:

```
print(root.openfile(types = ('py', 'txt')), 'is your file')
```

3.16 Savefile

Use `root.savefile({types})` to open a file save dialogue. You can filter the types of files it is allowed to return by giving the optional `types` arg. The `types` arg takes a list or a tuple. The default is `('all')`. If you have more than one, you will get a dropdown menu in the bottom selecting the filters.
Example:
If you have given `root.savefile(types = ('py', 'txt'))`, then you will get a file save dialogue with the default file extension being `*.py`, but you will be able to select `*.txt` as well. If you have selected `'all'`, then you will manually have to type the file extension as well.

3.17 Sizablefalse

Use `root.sizablefalse()` to make the window unresizable.
Example:

```
def change():
        global i
        i = not i

i = 0
root.button(text = 'Click Me!',
command = lambda: [exec(('root.sizablefalse()',
'root.sizabletrue()')[i]), change()]).grid()
```

3.18 Sizabletrue

Use `root.sizabletrue()` to make the window sizable again.
Example:

```
def change():
global i
i = not i

i = 0
root.button(text = 'Click Me!',
command = lambda: [exec(('root.sizablefalse()',
'root.sizabletrue()')[i]), change()]).grid()
```

3.19 Show

Use `root.show()` to show the window after making it. This is the exact same as `root.mainloop()`.
Example:

```
root = easytk.win()
root.title('')
```

3.20 Gettheme

Use `root.gettheme()` to get the name of the theme currently in use in a window called `root`.
Example:

```
root.text(text = f'Theme: {root.gettheme()}').grid()
```

3.21 Getgeo

Use `root.getgeo()` to get the dimensions of a window called `root`. When called, the function returns a tuple in the form `({width}, {height})`.
Example:

```
def config():
        d = root.getgeo()
        t.config(text = f'Dimensions: {d[0]}x{d[1]}')

t = root.text(text = '')
t.grid()
root.bind('<Configure>', lambda e: config())
```