

---

# ttkwidgets Documentation

*Release 0.12.0*

**ttwidgets developpers**

Aug 02, 2021

## Contents:

<b>1</b>	<b>Authors</b>	<b>1</b>
<b>2</b>	<b>Installation</b>	<b>3</b>
<b>3</b>	<b>Documentation</b>	<b>3</b>
<b>4</b>	<b>Examples</b>	<b>5</b>
<b>5</b>	<b>Index</b>	<b>16</b>
<b>6</b>	<b>License</b>	<b>16</b>
<b>7</b>	<b>Contributing</b>	<b>16</b>
<b>8</b>	<b>Issues</b>	<b>16</b>

---

A collection of widgets for Tkinter's ttk extensions by various authors

## 1 Authors

List of all the authors of widgets in this repository. Please note that this list only mentions the original creators of the widgets, and the widgets may have been edited and/or improved or otherwise modified by other authors.

- [RedFantom](#)
  - ScrolledFrame, based on an Unpythonic [idea](#)
  - ToggledFrame, based on an idea by [Onlyjus](#)
  - LinkLabel, based on an idea by [Nelson Brochado](#)
  - ScrolledListbox
  - FontChooser, based on an idea by [Nelson Brochado](#)

- FontSelectFrame
- Tooltip
- ItemsCanvas
- TimeLine
- The Python Team
  - Calendar, found [here](#)
- Mitja Martini
  - AutocompleteEntry, found [here](#)
- Russell Adams
  - AutocompleteCombobox, found [here](#)
- Juliette Monsel
  - CheckboxTreeview
  - Table
  - TickScale
  - AutoHideScrollbar based on an idea by [Fredrik Lundh](#)
  - All color widgets: askcolor(), ColorPicker, GradientBar and ColorSquare, LimitVar, Spinbox, AlphaBar and supporting functions in functions.py.
  - AutocompleteEntryListbox
- Dogeek
  - ValidatedEntry
  - Validator
  - RegexValidator
  - IntValidator
  - FloatValidator
  - PercentValidator
  - StringValidator
  - CapitalizedStringValidator
  - EmailValidator
  - PasswordValidator
  - IntEntry
  - FloatEntry
  - PercentEntry
  - LowerStringEntry
  - UpperStringEntry
  - CapitalizedStringEntry
  - EmailEntry

- PasswordEntry
- Multiple authors:
  - ScaleEntry (RedFantom and Juliette Monsel)

## 2 Installation

- With pip:

```
pip install ttkwidgets
```

- Ubuntu: ttkwidgets is available in the PPA [ppa:j-4321-i/ttkwidgets](#).

```
sudo add-apt-repository ppa:j-4321-i/ttkwidgets
sudo apt-get update
sudo apt-get install python(3)-ttkwidgets
```

- Archlinux: ttkwidgets is available in [AUR](#).

## 3 Documentation

---

**Note:** Only the widgets' specific options and methods are documented here, to get information about the options and methods inherited from standard tk/ttk widgets, consult [tkinter's documentation](#).

---

### 3.1 ttkwidgets

#### Classes

---

AutoHideScrollbar

---

Calendar

---

CheckboxTreeview

---

DebugWindow

---

ItemsCanvas

---

LinkLabel

---

ScaleEntry

---

ScrolledListbox

---

Table

---

TickScale

---

TimeLine

---

#### Modules

---

hook

---

tooltips

---

## 3.2 ttkwidgets.autocomplete

AutocompleteCombobox
AutocompleteEntry
AutocompleteEntryListbox

## 3.3 ttkwidgets.color

### Functions

### Classes

AlphaBar
ColorPicker
ColorSquare
GradientBar

## 3.4 ttkwidgets.font

### Functions

### Classes

FontChooser
FontFamilyDropdown
FontFamilyListbox
FontPropertiesFrame
FontSelectFrame
FontSizeDropdown

## 3.5 ttkwidgets.frames

### Classes

Tooltip
ScrolledFrame
ToggledFrame

## 3.6 ttkwidgets.validated\_entries

### Classes

ValidatedEntry
Validator
RegexValidator
IntValidator

Continued on next page

Table 7 – continued from previous page

FloatValidator
PercentValidator
StringValidator
CapitalizedStringValidator
EmailValidator
PasswordValidator
IntEntry
FloatEntry
PercentEntry
LowerStringEntry
UpperStringEntry
CapitalizedStringEntry
EmailEntry
PasswordEntry

#### Package structure

```
ttkwidgets
├── autocomplete
├── color
├── font
├── validated_entries
└── frames
```

## 4 Examples

Examples can be run in one of two ways:

- Run each example file as a stand alone script.
- Run the *run.py* script, which open a window with all reports at once represented each one by a button.

### 4.1 ttkwidgets

#### Example: AutoHideScrollbar

```
# -*- coding: utf-8 -*-

# Copyright (c) Juliette Monsel 2018
# For license see LICENSE

from ttkwidgets import AutoHideScrollbar
import tkinter as tk

window = tk.Tk()
listbox = tk.Listbox(window, height=5)
scrollbar = AutoHideScrollbar(window, command=listbox.yview)
listbox.configure(yscrollcommand=scrollbar.set)

for i in range(10):
    listbox.insert('end', 'item %i' % i)
```

(continues on next page)

```
tk.Label(window, text="Increase the window's height\nto make the scrollbar vanish.").
    ↳pack(side='top', padx=4, pady=4)
scrollbar.pack(side='right', fill='y')
listbox.pack(side='left', fill='both', expand=True)

window.mainloop()
```

## Example: Calendar

```
# -*- coding: utf-8 -*-

# Copyright (c) Juliette Monsel 2018
# For license see LICENSE

from ttkwidgets import Calendar
import tkinter as tk

def validate():
    sel = calendar.selection
    if sel is not None:
        label.configure(text='Selected date: %s' % sel.strftime('%x'))

window = tk.Tk()
calendar = Calendar(window, year=2015, month=3, selectforeground='white',
                    selectbackground='red')
calendar.pack()

tk.Button(window, text='Select', command=validate).pack()
label = tk.Label(window, text='Selected date:')
label.pack()
window.mainloop()
```

## Example: CheckboxTreeview

```
# -*- coding: utf-8 -*-

# Copyright (c) Juliette Monsel 2017
# For license see LICENSE

from ttkwidgets import CheckboxTreeview
import tkinter as tk

root = tk.Tk()

tree = CheckboxTreeview(root)
tree.pack()

tree.insert("", "end", "1", text="1")
tree.insert("1", "end", "11", text="11")
tree.insert("1", "end", "12", text="12")
tree.insert("11", "end", "111", text="111")
tree.insert("", "end", "2", text="2")
```

(continues on next page)

```
root.mainloop()
```

### Example: DebugWindow

```
# -*- coding: utf-8 -*-

# Copyright (c) RedFantom 2017
# Copyright (c) Juliette Monsel 2017
# For license see LICENSE

from ttkwidgets import DebugWindow
import tkinter as tk
from tkinter import ttk

root = tk.Tk()
ttk.Button(root, text="Print ok", command=lambda: print('ok')).pack()
DebugWindow(root)
root.mainloop()
```

### Example: ItemsCanvas

```
# -*- coding: utf-8 -*-

# Copyright (c) RedFantom 2017
# For license see LICENSE
from ttkwidgets import ItemsCanvas
import tkinter as tk
from tkinter import ttk

root = tk.Tk()

canvas = ItemsCanvas(root)
canvas.pack()

canvas.add_item("Example", font=("default", 13, "italic"), backgroundcolor="green",
→textcolor="darkblue",
    highlightcolor="blue")

root.mainloop()
```

### Example: LinkLabel

```
# -*- coding: utf-8 -*-

# Copyright (c) RedFantom 2017
# Copyright (c) Juliette Monsel 2018
# For license see LICENSE

from ttkwidgets import LinkLabel
```

(continues on next page)

(continued from previous page)

```
import tkinter as tk

window = tk.Tk()
LinkLabel(window, text="ttkwidgets repository",
           link="https://github.com/RedFantom/ttkwidgets",
           normal_color='royal blue',
           hover_color='blue',
           clicked_color='purple').pack()
window.mainloop()
```

### Example: ScaleEntry

```
# -*- coding: utf-8 -*-

# Copyright (c) Juliette Monsel 2018
# For license see LICENSE

from ttkwidgets import ScaleEntry
import tkinter as tk

window = tk.Tk()
scaleentry = ScaleEntry(window, scalewidth=200, entrywidth=3, from_=0, to=20)
scaleentry.config_entry(justify='center')
scaleentry.pack()
window.mainloop()
```

### Example: ScrolledListbox

```
# -*- coding: utf-8 -*-

# Copyright (c) Juliette Monsel 2018
# For license see LICENSE

from ttkwidgets import ScrolledListbox
import tkinter as tk

window = tk.Tk()
listbox = ScrolledListbox(window, height=5)

for i in range(10):
    listbox.listbox.insert('end', 'item {}'.format(i))

listbox.pack(fill='both', expand=True)
window.mainloop()
```

### Example: Table

```
# -*- coding: utf-8 -*-

# Copyright (c) Juliette Monsel 2018
```

(continues on next page)



```

# For license see LICENSE

from ttkwidgets import Table
import tkinter as tk
from tkinter import ttk

root = tk.Tk()

root.columnconfigure(0, weight=1)
root.rowconfigure(0, weight=1)

style = ttk.Style(root)
style.theme_use('alt')
sortable = tk.BooleanVar(root, False)
drag_row = tk.BooleanVar(root, False)
drag_col = tk.BooleanVar(root, False)

columns = ["A", "B", "C", "D", "E", "F", "G"]
table = Table(root, columns=columns, sortable=sortable.get(), drag_cols=drag_col.
    ↳get(),
                drag_rows=drag_row.get(), height=6)
for col in columns:
    table.heading(col, text=col)
    table.column(col, width=100, stretch=False)

# sort column A content as int instead of strings
table.column('A', type=int)

for i in range(12):
    table.insert('', 'end', iid=i,
                  values=(i, i) + tuple(i + 10 * j for j in range(2, 7)))

# add scrollbars
sx = tk.Scrollbar(root, orient='horizontal', command=table.xview)
sy = tk.Scrollbar(root, orient='vertical', command=table.yview)
table.configure(yscrollcommand=sy.set, xscrollcommand=sx.set)

table.grid(sticky='ewns')
sx.grid(row=1, column=0, sticky='ew')
sy.grid(row=0, column=1, sticky='ns')
root.update_idletasks()

# toggle table properties
def toggle_sort():
    table.config(sortable=sortable.get())

def toggle_drag_col():
    table.config(drag_cols=drag_col.get())

def toggle_drag_row():
    table.config(drag_rows=drag_row.get())

frame = tk.Frame(root)

```

(continues on next page)

(continued from previous page)

```
tk.Checkbutton(frame, text='sortable', variable=sortable, command=toggle_sort).
↳pack(side='left')
tk.Checkbutton(frame, text='drag columns', variable=drag_col, command=toggle_drag_
↳col).pack(side='left')
tk.Checkbutton(frame, text='drag rows', variable=drag_row, command=toggle_drag_row).
↳pack(side='left')
frame.grid()
root.geometry('400x200')

root.mainloop()
```

## Example: TickScale

```
# -*- coding: utf-8 -*-

# Copyright (c) Juliette Monsel 2017
# For license see LICENSE
from ttkwidgets import TickScale
import tkinter as tk
from tkinter import ttk

root = tk.Tk()
style = ttk.Style(root)
style.theme_use('clam')
style.configure('my.Vertical.TScale', sliderlength=50, background='white',
                foreground='red')
style.configure('my.Horizontal.TScale', sliderlength=10,
                font='TkDefaultFont 20 italic')
s1 = TickScale(root, orient='vertical', style='my.Vertical.TScale',
                tickinterval=0.2, from_=-1, to=1, showvalue=True, digits=2,
                length=400, labelpos='e')
s2 = TickScale(root, orient='horizontal', style='my.Horizontal.TScale',
                from_=0, to=10, tickinterval=2, resolution=1,
                showvalue=True, length=400)
s3 = TickScale(root, orient='horizontal', from_=0.25, to=1, tickinterval=0.1,
                resolution=0.1)

s1.pack(fill='y')
s2.pack(fill='x')
s3.pack(fill='x')

root.mainloop()
```

## Example: TimeLine

```
# -*- coding: utf-8 -*-

# Copyright (c) RedFantom 2017
# For license see LICENSE

import tkinter as tk
from ttkwidgets import TimeLine
```

(continues on next page)

```

window = tk.Tk()
timeline = TimeLine(
    window,
    categories={str(key): {"text": "Category {}".format(key)} for key in range(0, 5)},
    height=100, extend=True
)
menu = tk.Menu(window, tearoff=False)
menu.add_command(label="Some Action", command=lambda: print("Command Executed"))
timeline.tag_configure("1", right_callback=lambda *args: print(args), menu=menu,
    ↳foreground="green",
        active_background="yellow", hover_border=2, move_
    ↳callback=lambda *args: print(args))
timeline.create_marker("1", 1.0, 2.0, background="white", text="Change Color", tags=(
    ↳"1",), iid="1")
timeline.create_marker("2", 2.0, 3.0, background="green", text="Change Category",
    ↳foreground="white", iid="2",
        change_category=True)
timeline.create_marker("3", 1.0, 2.0, text="Show Menu", tags=("1",))
timeline.create_marker("4", 4.0, 5.0, text="Do nothing", move=False)
timeline.draw_timeline()
timeline.grid()
window.after(2500, lambda: timeline.configure(marker_background="cyan"))
window.after(5000, lambda: timeline.update_marker("1", background="red"))
window.after(5000, lambda: print(timeline.time))
window.mainloop()

```

## Example: tooltips

```

"""
Author: RedFantom
License: GNU GPLv3
Source: The ttkwidgets repository
"""
import tkinter as tk
from tkinter import ttk
# Import once, use everywhere
from ttkwidgets import tooltips

window = tk.Tk()
button = ttk.Button(window, text="Destroy", command=window.destroy, tooltip="This_
    ↳button destroys the window.")
button.pack()
x = lambda: button.configure(tooltip="This button no longer destroys the window",
    ↳command=lambda: print("Behaviour changed!"))
window.after(5000, x)
window.mainloop()

```

## 4.2 ttkwidgets.autocomplete

### Example: AutocompleteCombobox

```
# -*- coding: utf-8 -*-

# Copyright (c) Juliette Monsel 2018
# For license see LICENSE

from ttkwidgets.autocomplete import AutocompleteCombobox
import tkinter as tk

window = tk.Tk()
tk.Label(window, text="Combobox with autocompletion for the Tk instance's methods:").
    pack(side='left')
entry = AutocompleteCombobox(window, width=20, completevalues=dir(window))
entry.pack(side='right')
window.mainloop()
```

### Example: AutocompleteEntry

```
# -*- coding: utf-8 -*-

# Copyright (c) Juliette Monsel 2018
# For license see LICENSE

from ttkwidgets.autocomplete import AutocompleteEntry
import tkinter as tk

window = tk.Tk()
tk.Label(window, text="Entry with autocompletion for the Tk instance's methods:").
    pack(side='left')
entry = AutocompleteEntry(window, width=20, completevalues=dir(window))
entry.pack(side='right')
window.mainloop()
```

### Example: AutocompleteEntryListbox

```
# -*- coding: utf-8 -*-

# Copyright (c) Juliette Monsel 2019
# For license see LICENSE

import tkinter as tk
from ttkwidgets.autocomplete import AutocompleteEntryListbox

window = tk.Tk()
tk.Label(window, text="Entry + Listbox with autocompletion for the Tk instance's_
    methods:").pack()
entry = AutocompleteEntryListbox(window, width=20, completevalues=dir(window))
entry.pack()
window.mainloop()
```

## 4.3 ttkwidgets.color

## Example: askcolor

```
# -*- coding: utf-8 -*-

# Copyright (c) Juliette Monsel 2018
# For license see LICENSE

from ttkwidgets.color import askcolor
import tkinter as tk
from tkinter import ttk
from PIL import Image, ImageTk

def pick(alpha=False):
    global im # to avoid garbage collection of image
    res = askcolor('sky blue', parent=window, title='Pick a color', alpha=alpha)
    canvas.delete('image')
    if res[1] is not None:
        im = ImageTk.PhotoImage(Image.new('RGBA', (100, 100), res[1]), master=window)
        canvas.create_image(60, 60, image=im, tags='image', anchor='center')
    print(res)

window = tk.Tk()
canvas = tk.Canvas(window, width=120, height=120)
canvas.create_text(60, 60, text='Background', anchor='center')
canvas.pack()
ttk.Button(window, text="Pick a color (No alpha channel)", command=pick).pack(fill='x')
ttk.Button(window, text="Pick a color (With alpha channel)", command=lambda: pick(True)).pack(fill='x')
window.mainloop()
```

## 4.4 ttkwidgets.font

### Example: FontSelectFrame

```
# -*- coding: utf-8 -*-

# Copyright (c) Juliette Monsel 2018
# For license see LICENSE

from ttkwidgets.font import FontSelectFrame
import tkinter as tk
from tkinter import ttk

def update_preview(font_tuple):
    print(font_tuple)
    font = font_selection.font[0]
    if font is not None:
        label.configure(font=font)

window = tk.Tk()
label = ttk.Label(window, text='Sample text rendered in the chosen font.')
label.pack(padx=10, pady=10)
```

(continues on next page)

(continued from previous page)

```
font_selection = FontSelectFrame(window, callback=update_preview)
font_selection.pack()
window.mainloop()
```

### Example: askfont

```
# -*- coding: utf-8 -*-

# Copyright (c) Juliette Monsel 2018
# For license see LICENSE

from ttkwidgets.font import askfont
import tkinter as tk
from tkinter import ttk

def font():
    res = askfont()
    if res[0] is not None:
        label.configure(font=res[0])
    print(res)

window = tk.Tk()
label = ttk.Label(window, text='Sample text rendered in the chosen font.')
label.pack(padx=10, pady=10)
ttk.Button(window, text="Pick a font", command=font).pack()
window.mainloop()
```

## 4.5 ttkwidgets.frames

### Example: ScrolledFrame

```
# -*- coding: utf-8 -*-

# Copyright (c) Juliette Monsel 2018
# For license see LICENSE

from ttkwidgets.frames import ScrolledFrame
import tkinter as tk
from tkinter import ttk

window = tk.Tk()
frame = ScrolledFrame(window, compound=tk.RIGHT, canvasheight=200)
frame.pack(fill='both', expand=True)

for i in range(20):
    ttk.Label(frame.interior, text='Label %i' % i).pack()
window.mainloop()
```

## Example: ToggledFrame

```
# -*- coding: utf-8 -*-

# Copyright (c) RedFantom 2017
# For license see LICENSE

from ttkwidgets.frames import ToggledFrame
import tkinter as tk
from tkinter import ttk

window = tk.Tk()
frame = ToggledFrame(window, text="Value", width=10)
frame.pack()
button = ttk.Button(frame.interior, text="Button", command=window.destroy)
button.grid()
frame.toggle()
window.mainloop()
```

## Example: Tooltip

```
# -*- coding: utf-8 -*-

# Copyright (c) RedFantom 2017
# For license see LICENSE

from ttkwidgets.frames import Tooltip
import tkinter as tk

window = tk.Tk()
button = tk.Button(window, text="Button", command=window.destroy)
button.pack()
balloon = Tooltip(button)
window.mainloop()
```

## 4.6 ttkwidgets.hook

### Example: hook\_ttk\_widgets

```
# -*- coding: utf-8 -*-

# Copyright (c) RedFantom 2021
# For license see LICENSE

from tkinter import import ttk
import tkinter as tk
from ttkwidgets.hook import hook_ttk_widgets

if __name__ == '__main__':
    hook_ttk_widgets(lambda s, o, v: print(s, o, v), {"tooltip": "Default Value"})
    hook_ttk_widgets(lambda s, o, v: print(s, o, v), {"hello_world": "second_hook"})

    original_init = ttk.Button.__init__
```

(continues on next page)

```

def __init__(self, *args, **kwargs):
    print("User custom hook")
    original_init(self, *args, **kwargs)

    ttk.Button.__init__ = __init__

    window = tk.Tk()
    button = ttk.Button(window, text="Destroy", command=window.destroy, tooltip=
↪ "Destroys Window")
    button.pack()
    print([name for name in dir(button) if name.startswith("WidgetHook")])
    window.after(1000, lambda: button.configure(tooltip="Does not destroy window",
↪ command=lambda: None))
    window.mainloop()

```

## 5 Index

## 6 License

ttkwidgets: A collection of widgets for Tkinter's ttk extensions by various authors.

- Copyright (C) RedFantom 2017
- Copyright (C) The Python Team
- Copyright (C) Mitja Martini 2008
- Copyright (C) Russell Adams 2011
- Copyright (C) Juliette Monsel 2017

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

## 7 Contributing

If you have created a widget that you think is worth adding, then feel free to fork the [repository](#) and create a [Pull Request](#) when you've added the widget to your copy of the repository. You will be credited for your work, and you can add headers to your files. You will also be added to the [AUTHORS.md](#) file.

## 8 Issues

If you find any bugs or have any ideas, feel free to open an [issue](#) in the repository, and it will be looked at.