
tkcalendar Documentation

Release 1.5.0

Juliette Monsel

Sep 06, 2019

Contents:

| | | |
|----------|----------------------|-----------|
| 1 | Installation | 3 |
| 2 | Documentation | 5 |
| 3 | Example | 11 |
| 4 | HowTos | 13 |
| 5 | Changelog | 15 |
| 6 | Index | 19 |
| 7 | License | 21 |
| | Index | 23 |

tkcalendar is a python module that provides the Calendar and DateEntry widgets for Tkinter. The DateEntry widget is similar to a Combobox, but the drop-down is not a list but a Calendar to select a date. Events can be displayed in the Calendar with custom colors and a tooltip displays the event list for a given day. tkcalendar is compatible with both Python 2 and Python 3. It supports many locale settings (e.g. 'fr_FR', 'en_US', ..) and the colors are customizable.

Project page: <https://github.com/j4321/tkcalendar>

CHAPTER 1

Installation

Requirements

- Linux, Windows, OSX
- Python 2 or 3 with tkinter + ttk (default for Windows but not for Linux) and babel

Install

- Ubuntu: use the PPA [ppa:j-4321-i/ppa](#)

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

- Archlinux:

The package is available on [AUR](#)

- With pip:

```
$ pip install tkcalendar
```

Note: This documentation only addresses the methods specific to the *Calendar* and *DateEntry* widgets, but these widgets also possess the methods common to all tkinter widgets. and the one common to all ttk widgets. In addition, the *DateEntry* also have all the methods of the Entry widget.

2.1 Calendar

2.1.1 Class

class tkcalendar.**Calendar** (*master=None, **kw*)

Bases: tkinter.ttk.Frame

Calendar widget.

__init__ (*master=None, **kw*)

Construct a *Calendar* with parent master.

Standard Options

cursor [str] cursor to display when the pointer is in the widget

font [str or Tkinter Font] font of the calendar

borderwidth [int] width of the border around the calendar

state [str] “normal” or “disabled” (unresponsive widget)

Widget-specific Options

year [int] initially displayed year, default is current year.

month [int] initially displayed month, default is current month.

day [int] initially selected day, if month or year is given but not day, no initial selection, otherwise, default is today.

firstweekday [str] first day of the week: “monday” or “sunday”

weekenddays [list] days to be displayed as week-end days given as a list of integers corresponding to the number of the day in the week (e.g. [6, 7] for the last two days of the week).

mindate [datetime.date or datetime.datetime (default is None)] minimum allowed date

maxdate [datetime.date or datetime.datetime (default is None)] maximum allowed date

showweeknumbers [bool] whether to display week numbers (default is True).

showothermonthdays [bool] whether to display the last days of the previous month and the first of the next month (default is True).

locale [str] locale to use, e.g. ‘en_US’

date_pattern [str] date pattern used to format the date as a string. The default pattern used is `babel`’s short date format in the *Calendar*’s locale.

A valid pattern is a combination of ‘d’, ‘m’ and ‘y’ separated by non letter characters to indicate how and in which order the year, month and day should be displayed.

| | |
|---|---|
| d | ‘d’ for the day of month number without padding, ‘dd’ for a two-digit day |
| m | ‘m’ for the month number without padding, ‘mm’ for a two-digit month |
| y | ‘yy’ for the two last digits of the year, any other number of ‘y’ for the full year with an extra padding of zero if it has less digits than the number of ‘y’. |

Examples for `datetime.date(2019, 7, 1)`

- ‘y-mm-dd’ → ‘2019-07-01’
- ‘m/d/yy’ → ‘7/1/19’

selectmode [str] “none” or “day” (default): whether the user can change the selected day with a mouse click.

textvariable [StringVar] connect the currently selected date to the variable.

Style Options

background [str] background color of calendar border and month/year name

foreground [str] foreground color of month/year name

disabledbackground [str] background color of calendar border and month/year name in disabled state

disabledforeground [str] foreground color of month/year name in disabled state

bordercolor [str] day border color

headersbackground [str] background color of day names and week numbers

headersforeground [str] foreground color of day names and week numbers

selectbackground [str] background color of selected day

selectforeground [str] foreground color of selected day

disabledselectbackground [str] background color of selected day in disabled state

disabledselectforeground [str] foreground color of selected day in disabled state

normalbackground [str] background color of normal week days

normalforeground [str] foreground color of normal week days

weekendbackground [str] background color of week-end days

weekendforeground [str] foreground color of week-end days

othermonthforeground [str] foreground color of normal week days belonging to the previous/next month

othermonthbackground [str] background color of normal week days belonging to the previous/next month

othermonthweforeground [str] foreground color of week-end days belonging to the previous/next month

othermonthwebbackground [str] background color of week-end days belonging to the previous/next month

disableddaybackground [str] background color of days in disabled state

disableddayforeground [str] foreground color of days in disabled state

Tooltip Options (for calevents)

tooltipforeground [str] tooltip text color

tooltipbackground [str] tooltip background color

tooltipalpha [float] tooltip opacity between 0 and 1

tooltipdelay [int] delay in ms before displaying the tooltip

calevent_cget (*ev_id, option*)

Return value of given option for the event *ev_id*.

calevent_configure (*ev_id, **kw*)

Configure the event *ev_id*.

Keyword options: date, text, tags (see `calevent_create` options).

calevent_create (*date, text, tags=[]*)

Add new event in calendar and return event id.

Options:

date [datetime.date or datetime.datetime] event date

text [str] text to put in the tooltip associated to date.

tags [list] list of tags to apply to the event. The last tag determines the way the event is displayed. If there are several events on the same day, the lowest one (on the tooltip list) which has tags determines the colors of the day.

calevent_lower (*ev_id, below=None*)

Lower event *ev_id* in tooltip event list.

below [str] put *ev_id* below given one, if below is None, put it at the bottom of tooltip event list.

The day's colors are determined by the last tag of the lowest event which has tags.

calevent_raise (*ev_id, above=None*)

Raise event *ev_id* in tooltip event list.

above [str] put *ev_id* above given one, if above is None, put it on top of tooltip event list.

The day's colors are determined by the last tag of the lowest event which has tags.

calevent_remove (**ev_ids, **kw*)

Remove events from calendar.

Arguments: event ids to remove or 'all' to remove them all.

Keyword arguments: tag, date.

They are taken into account only if no id is given. Remove all events with given tag on given date. If only date is given, remove all events on date and if only tag is given, remove all events with tag.

configure (*cnf*={}, ***kw*)

Configure resources of a widget.

The values for resources are specified as keyword arguments. To get an overview about the allowed keyword arguments call the method `keys()`.

format_date (*date*=None)

Convert date (datetime.date) to a string in the locale.

get_calevents (*date*=None, *tag*=None)

Return event ids of events with given tag and on given date.

If only date is given, return event ids of all events on date. If only tag is given, return event ids of all events with tag. If both options are None, return all event ids.

get_date ()

Return selected date as string.

get_displayed_month ()

Return the currently displayed month in the form of a (month, year) tuple.

keys ()

Return a list of all resource names of this widget.

see (*date*)

Display the month in which date is.

date [datetime.date or datetime.datetime] date to be made visible

selection_clear ()

Clear the selection.

selection_get ()

Return currently selected date (datetime.date instance).

Always return None if selectmode == "none".

selection_set (*date*)

Set the selection to date.

date [datetime.date, datetime.datetime or str] date to be made visible. If given as a string, it should be in the format corresponding to the calendar locale.

Do nothing if selectmode == "none".

tag_cget (*tag*, *option*)

Return the value of the tag's option.

tag_config (*tag*, ***kw*)

Configure tag.

Keyword options: foreground, background (of the day in the calendar)

tag_delete (*tag*)

Delete given tag.

Delete tag properties and remove tag from all events.

tag_names ()

Return tuple of existing tags.

2.1.2 Virtual Events

- A <<CalendarSelected>> event is generated each time the user selects a day with the mouse.
- A <<CalendarMonthChanged>> event is generated each time the user changes the displayed month.

2.1.3 Calendar Events

Special events (e.g. birthdays, ..) can be managed using the `calevent_. . ()` methods. The way they are displayed in the calendar is determined with tags. An id is attributed to each event upon creation and can be used to edit the event (`ev_id` argument).

2.2 DateEntry

2.2.1 Class

class tkcalendar.**DateEntry** (*master=None, **kw*)

Bases: `tkinter.ttk.Entry`

Date selection entry with drop-down calendar.

__init__ (*master=None, **kw*)

Create an entry with a drop-down calendar to select a date.

When the entry loses focus, if the user input is not a valid date, the entry content is reset to the last valid date.

Keyword Options

- `ttk.Entry` options:
class, cursor, style, takefocus, xscrollcommand, exportselection, justify, show, state, textvariable, width.
- *Calendar* options: see the [documentation](#).

cget (*key*)

Return the resource value for a KEY given as string.

config (*cnf={}, **kw*)

Configure resources of a widget.

The values for resources are specified as keyword arguments. To get an overview about the allowed keyword arguments call the method `keys ()`.

configure (*cnf={}, **kw*)

Configure resources of a widget.

The values for resources are specified as keyword arguments. To get an overview about the allowed keyword arguments call the method `keys ()`.

destroy ()

Destroy this and all descendants widgets.

drop_down ()

Display or withdraw the drop-down calendar depending on its current state.

get_date ()

Return the content of the DateEntry as a `datetime.date` instance.

keys ()

Return a list of all resource names of this widget.

set_date (*date*)

Set the value of the DateEntry to date.

date can be a datetime.date, a datetime.datetime or a string in locale ‘%x’ format.

state (**args*)

Modify or inquire widget state.

Widget state is returned if statespec is None, otherwise it is set according to the statespec flags and then a new state spec is returned indicating which flags were changed. statespec is expected to be a sequence.

2.2.2 Virtual Events

A <<DateEntrySelected>> event is generated each time the user selects a date.

CHAPTER 3

Example

```
try:
    import tkinter as tk
    from tkinter import ttk
except ImportError:
    import Tkinter as tk
    import ttk

from tkcalendar import Calendar, DateEntry

def example1():
    def print_sel():
        print(cal.selection_get())

    top = tk.Toplevel(root)

    cal = Calendar(top, font="Arial 14", selectmode='day', locale='en_US',
                    cursor="hand1", year=2018, month=2, day=5)

    cal.pack(fill="both", expand=True)
    ttk.Button(top, text="ok", command=print_sel).pack()

def example2():

    top = tk.Toplevel(root)

    cal = Calendar(top, selectmode='none')
    date = cal.datetime.today() + cal.timedelta(days=2)
    cal.calevent_create(date, 'Hello World', 'message')
    cal.calevent_create(date, 'Reminder 2', 'reminder')
    cal.calevent_create(date + cal.timedelta(days=-2), 'Reminder 1', 'reminder')
    cal.calevent_create(date + cal.timedelta(days=3), 'Message', 'message')

    cal.tag_config('reminder', background='red', foreground='yellow')
```

(continues on next page)

(continued from previous page)

```
cal.pack(fill="both", expand=True)
ttk.Label(top, text="Hover over the events.").pack()

def example3():
    top = tk.Toplevel(root)

    ttk.Label(top, text='Choose date').pack(padx=10, pady=10)

    cal = DateEntry(top, width=12, background='darkblue',
                    foreground='white', borderwidth=2, year=2010)
    cal.pack(padx=10, pady=10)

root = tk.Tk()
ttk.Button(root, text='Calendar', command=example1).pack(padx=10, pady=10)
ttk.Button(root, text='Calendar with events', command=example2).pack(padx=10, pady=10)
ttk.Button(root, text='DateEntry', command=example3).pack(padx=10, pady=10)

root.mainloop()
```


4.1 Widget styling

Calendar

All styling is done using options, see the [documentation](#).

DateEntry

`DateEntry` inherits from `ttk.Entry` therefore the styling is done using a `ttk` style:

```
try:
    import tkinter as tk
    from tkinter import ttk
except ImportError:
    import Tkinter as tk
    import ttk

from tkcalendar import DateEntry

style = ttk.Style()
# style.theme_use('clam') # -> uncomment this line if the styling does not work
style.configure('my.DateEntry',
                fieldbackground='light green',
                background='dark green',
                foreground='dark blue',
                arrowcolor='white')

dateentry = DateEntry(style='my.DateEntry')
dateentry.pack()

tk.mainloop()
```

If the style of the *DateEntry* does not change, then it might be because of the used ttk theme. Changing the theme with `style.theme_use('clam')` should solve the issue.

4.2 PyInstaller

When bundling an application with *PyInstaller*, there seem to be an issue ([#32](#)) with the detection of the babel dependency of tkcalendar. This can be fixed by using the `--hidden-import` option:

```
$ pyinstaller --hidden-import babel.numbers myscript.py
```

or by editing the `.spec` file:

```
hiddenimports=["babel.numbers"]
```

4.3 Custom date formatting

When using the “en_US” locale, the default date formatting in the *DateEntry*, or when getting the selected date from the *Calendar* as a string is `M/d/yy`, i.e. July 4, 2019 will give “7/4/19”. If you want to get “07/04/2019” instead, you can pass “MM/dd/yyyy” to the `date_pattern` option of the *Calendar* or *DateEntry*.

```
try:
    import tkinter as tk
    from tkinter import ttk
except ImportError:
    import Tkinter as tk
    import ttk

from tkcalendar import DateEntry

DateEntry(locale='en_US').pack()
DateEntry(locale='en_US', date_pattern='MM/dd/yyyy').pack()

tk.mainloop()
```

5.1 tkcalendar 1.5.1

Bug fixes

- Fix calendar drop-down not in front issue if window has the `-topmost` attribute in Windows (#49)
- Make `Calendar.config()` and `DateEntry.config()` accept a dictionary like standard tkinter widgets
- Fix calendar not hiding when clicking again on `DateEntry` drop-down button in Windows (#51)
- Fix `maxdate` disabled while it could be the latest allowed date (#50)

5.2 tkcalendar 1.5.0

New features

- `Calendar.see()` method: make sure given date is visible
- Make `locale` option editable after the creation of the `Calendar`

New options

- `disabledforeground` and `disabledbackground`: colors of calendar border and month/year name in disabled state
- `maxdate` and `mindate`: set an allowed date range for date selection
- `weekenddays`: choose the days colored as week-end days (#37)
- `date_pattern`: customize the date format

Bug fixes

- Make `Calendar.selection_clear()` actually clear the selection
- Fix `ValueError` when retrieving default locale
- Fix date parsing error in Swedish locale and some others (#44)
- Improve compliance with ttk themes by make the `DateEntry` look like a `ttk.Combobox` (#42)
- Fix high CPU issues in Windows (#36)

5.3 tkcalendar 1.4.0

New features

- `<<CalendarMonthChanged>>` virtual event: event generated each time the user changes the displayed month
- `Calendar.get_displayed_month()` method: return the currently displayed month in the form of a (month, year) tuple

New options

- `showothermonthdays`: show/hide the last and first days of the previous and next months

Bug fixes

- Fix handling of `style` option in `DateEntry`
- Fix display of events for January days showing on December and conversely

5.4 tkcalendar 1.3.1

Bug fixes

- Fix bug in day selection when `firstweekday` is 'sunday' (#28)

5.5 tkcalendar 1.3.0

New features

- Add possibility to display special events (like birthdays, ..) in the calendar. The events are displayed with colors defined by tags and the event description is displayed in a tooltip (see *Calendar Events*) (#19)

New options

- `showwekknnumbers`: show/hide week numbers (#18)
- `firstweekday`: first week day ('monday' or 'sunday') (#25)

Bug fixes

- No longer set locale globally to avoid conflicts between several instances, use babel module instead (#15)
- Make *DateEntry* compatible with more ttk themes, especially OSX default theme (#16)

5.6 tkcalendar 1.2.1

Bug fixes

- Fix *ValueError* in *DateEntry* with Python 3.6.5 (#13)

5.7 tkcalendar 1.2.0

New options

- *textvariable*: connect the currently selected date to the given *StringVar* (#6)
- *state*: 'normal' or 'disabled'
- *disabledselectbackground*, *disabledselectforeground*, *disableddaybackground* and *disableddayforeground*: configure colors when *Calendar* is disabled

Bug fixes

- Fix *DateEntry* behavior in readonly mode
- Make *Calendar.selection_get()* always return a *datetime.date*

5.8 tkcalendar 1.1.5

Bug fixes

- Fix endless triggering of <<ThemeChanged>> event in *DateEntry* (#9)

5.9 tkcalendar 1.1.4

Bug fixes

- Fix error in January due to week 53
- Fix *DateEntry* for ttk themes other than 'clam' (#3)

5.10 tkcalendar 1.1.3

Bug fixes

- Make *DateEntry* support initialisation with partial dates (e.g. just year=2010)
- Improve handling of wrong year-month-day combinations

5.11 tkcalendar 1.1.2

Bug fixes

- Fix bug after destroying a *DateEntry*
- Fix bug in style and font

5.12 tkcalendar 1.1.1

Bug fixes

- Fix bug when content of *DateEntry* is not a valid date

5.13 tkcalendar 1.1.0

New widget

- *DateEntry*: date selection entry with drop-down calendar

New options

- *borderwidth*: width of the border around the calendar (integer)
- *othermonthbackground*: background color for normal week days belonging to the previous/next month
- *othermonthweforeground*: foreground color for week-end days belonging to the previous/next month
- *othermonthwebackground*: back

Bug fixes

- Fix display of the first days of the next month
- Increment year when going from December to January

5.14 tkcalendar 1.0.0

- Initial version

CHAPTER 6

Index

tkcalendar - Calendar and DateEntry widgets for Tkinter

Copyright (C) 2017-2018 Juliette Monsel with contributions from:

- [Neal Probert](#)
- [arahorn28](#)

tkcalendar 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.

tkcalendar 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/>.

Symbols

`__init__()` (*tkcalendar.Calendar* method), 5
`__init__()` (*tkcalendar.DateEntry* method), 9

C

Calendar (class in *tkcalendar*), 5
`calevent_cget()` (*tkcalendar.Calendar* method), 7
`calevent_configure()` (*tkcalendar.Calendar* method), 7
`calevent_create()` (*tkcalendar.Calendar* method), 7
`calevent_lower()` (*tkcalendar.Calendar* method), 7
`calevent_raise()` (*tkcalendar.Calendar* method), 7
`calevent_remove()` (*tkcalendar.Calendar* method), 7
`cget()` (*tkcalendar.DateEntry* method), 9
`config()` (*tkcalendar.DateEntry* method), 9
`configure()` (*tkcalendar.Calendar* method), 8
`configure()` (*tkcalendar.DateEntry* method), 9

D

DateEntry (class in *tkcalendar*), 9
`destroy()` (*tkcalendar.DateEntry* method), 9
`drop_down()` (*tkcalendar.DateEntry* method), 9

F

`format_date()` (*tkcalendar.Calendar* method), 8

G

`get_calevents()` (*tkcalendar.Calendar* method), 8
`get_date()` (*tkcalendar.Calendar* method), 8
`get_date()` (*tkcalendar.DateEntry* method), 9
`get_displayed_month()` (*tkcalendar.Calendar* method), 8

K

`keys()` (*tkcalendar.Calendar* method), 8
`keys()` (*tkcalendar.DateEntry* method), 9

S

`see()` (*tkcalendar.Calendar* method), 8
`selection_clear()` (*tkcalendar.Calendar* method), 8
`selection_get()` (*tkcalendar.Calendar* method), 8
`selection_set()` (*tkcalendar.Calendar* method), 8
`set_date()` (*tkcalendar.DateEntry* method), 10
`state()` (*tkcalendar.DateEntry* method), 10

T

`tag_cget()` (*tkcalendar.Calendar* method), 8
`tag_config()` (*tkcalendar.Calendar* method), 8
`tag_delete()` (*tkcalendar.Calendar* method), 8
`tag_names()` (*tkcalendar.Calendar* method), 8