

---

# **tkStuff Documentation**

***Release 0.2***

**Simon Kennedy**

**Feb 22, 2018**



---

## Contents

---

<b>1</b>	<b>Gallery</b>	<b>1</b>
<b>2</b>	<b>Code Examples</b>	<b>7</b>
<b>3</b>	<b>Date Handling</b>	<b>9</b>
<b>4</b>	<b>Time Handling</b>	<b>11</b>
<b>5</b>	<b>Color Handling</b>	<b>13</b>
<b>6</b>	<b>Filesystem Entry Handling</b>	<b>17</b>
<b>7</b>	<b>Password Handling</b>	<b>19</b>
<b>8</b>	<b>Miscellaneous Stuff</b>	<b>21</b>
<b>9</b>	<b>License</b>	<b>25</b>
<b>10</b>	<b>Dependencies</b>	<b>27</b>
<b>11</b>	<b>Installation</b>	<b>29</b>
<b>12</b>	<b>Contact</b>	<b>31</b>
<b>13</b>	<b>Version History</b>	<b>33</b>
	<b>Python Module Index</b>	<b>35</b>



Shown below are examples of the widgets currently provided with tkStuff.

### 1.1 Date

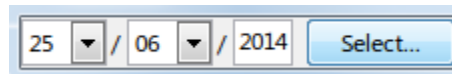


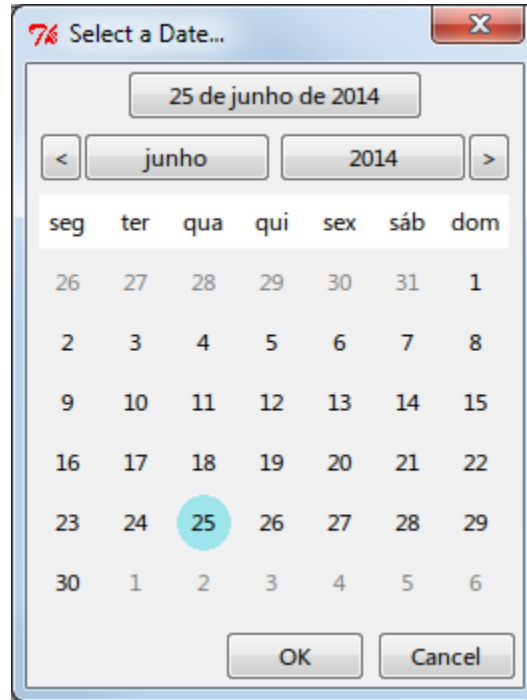
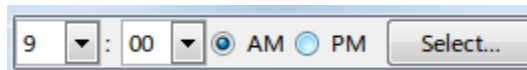
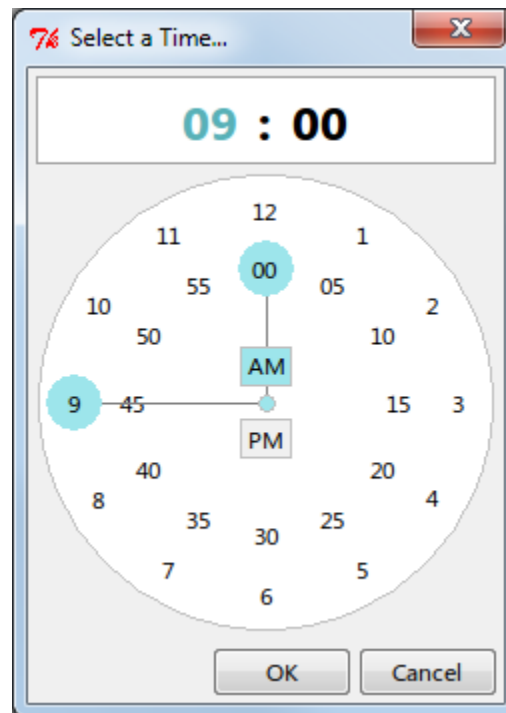
Fig. 1.1: Date Entry Widget (See the *DateEntry* class)

### 1.2 Time

The next 2 time selection dialog styles are chosen based on the Locale e.g. for the *en\_US* locale a 12 hour dialog is displayed.

### 1.3 Color

### 1.4 Passwords

Fig. 1.2: Date Selection Dialog (See the *DateDialog* class)Fig. 1.3: Time Entry Dialog (See the *TimeEntry* class)Fig. 1.4: 12 Hour Time Dialog Widget with AM/PM. (See the *TimeDialog* class)

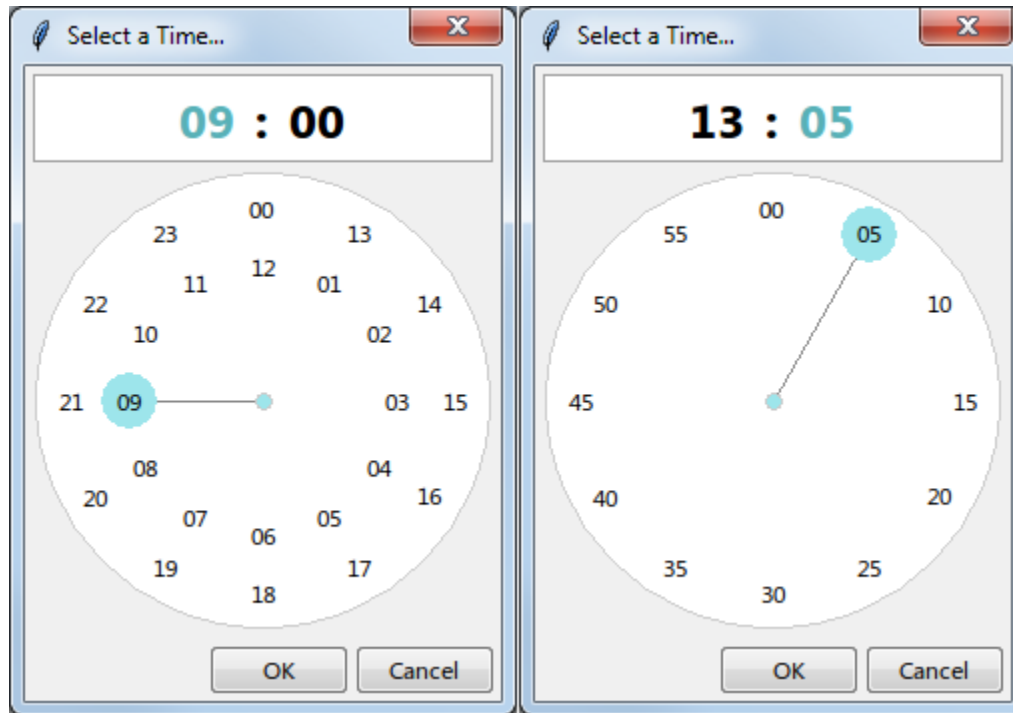


Fig. 1.5: 24 Hour Time Dialog Widget with Separate Minute Selection (See the *TimeDialog* class)



Fig. 1.6: Color Entry Widget (See the *ColorEntry* class)

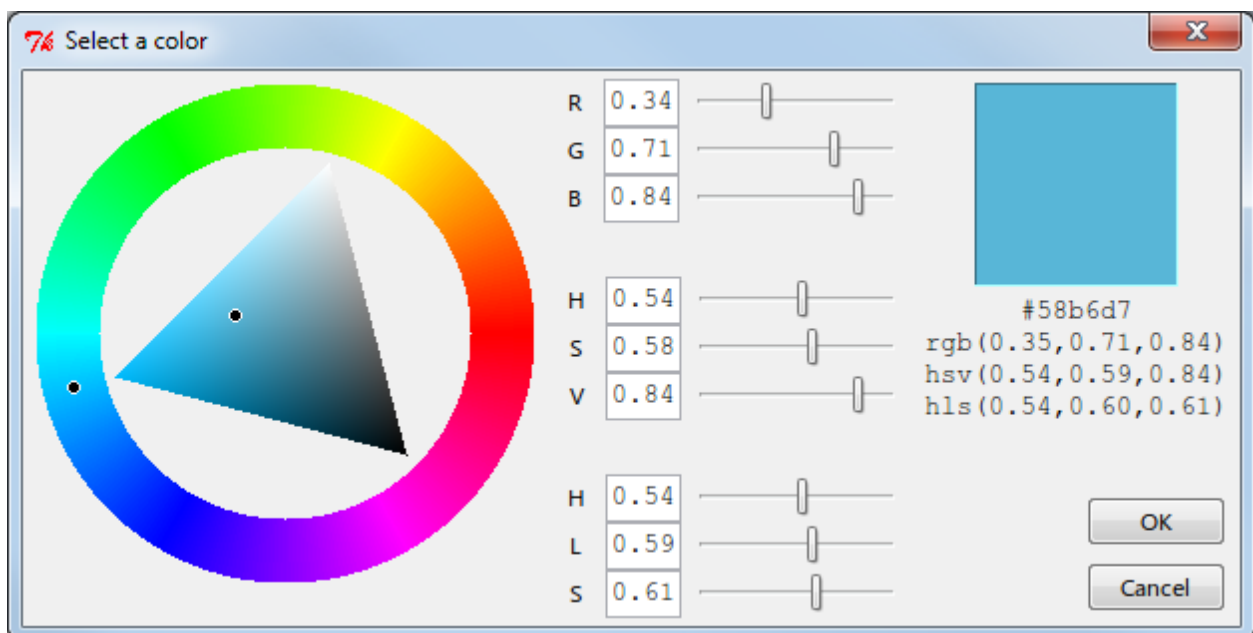


Fig. 1.7: Color Selection Dialog (See the *ColorDialog* class)

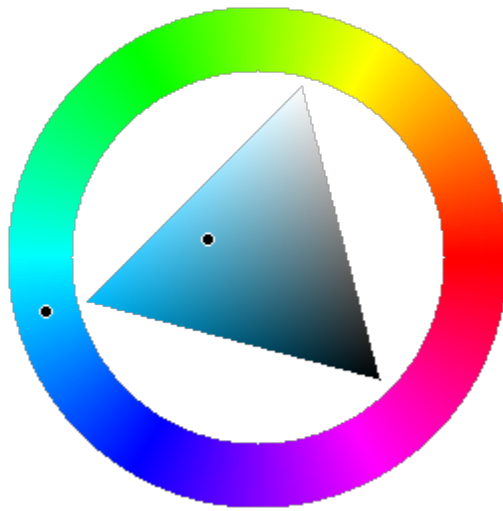


Fig. 1.8: Color Wheel Widget (See the *ColorWheel* class)

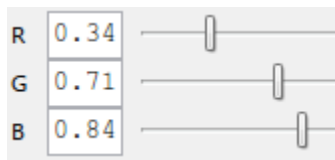


Fig. 1.9: Color Slider Widget (See the *ColorSlider* class and subclasses)

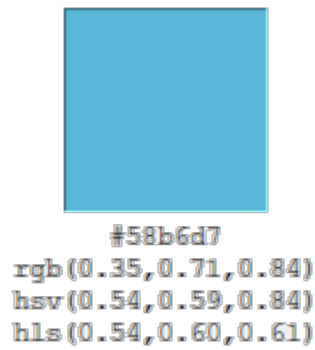


Fig. 1.10: Color Square Widget (See the *ColorSquare* class)





Fig. 1.11: Color Palette Selector Widget (See the *PaletteSelector* class)



Fig. 1.12: Tints and Shades Widgets (See the *ColorTint* and *ColorShade* classes)



Fig. 1.13: Password Entry Widget (See the *PasswordEntry* class)

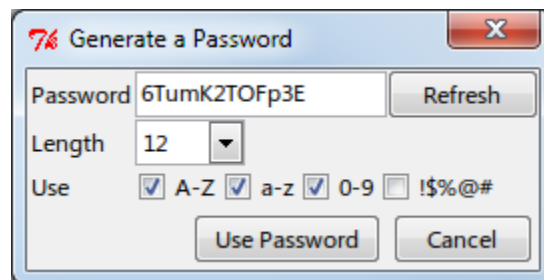


Fig. 1.14: Password Generation Dialog (See the *PasswordDialog* class)

## 2.1 A Date Entry

To display the date entry and dialog shown *in the gallery* the following code can be used.

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

from tks.dates import DateEntry

if __name__ == '__main__':
    root = tk.Tk()
    root.title('Date Test')
    entry = DateEntry(root, locale='pt')
    entry.grid(row=0, column=0, sticky=tk.EW)
    root.columnconfigure(0, weight=1)
    root.mainloop()
```

## 2.2 A Time Entry

To display the time entry and dialog shown *in the gallery* the following code can be used.

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

from tks.times import TimeEntry

if __name__ == '__main__':
```

```
root = tk.Tk()
root.title('Time Test')
entry = TimeEntry(root, locale='en_US')
entry.grid(row=0, column=0, sticky=tk.EW)
root.columnconfigure(0, weight=1)
root.mainloop()
```

## 2.3 A Color Entry

To display the color entry and dialog shown *in the gallery* the following code can be used.

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

from tks.colors import ColorEntry

if __name__ == '__main__':
    root = tk.Tk()
    root.title('Color Test')
    entry = ColorEntry(root, start_color='#fedcba')
    entry.grid(row=0, column=0, sticky=tk.EW)
    root.columnconfigure(0, weight=1)
    root.mainloop()
```

## 2.4 A Color Palette Widget

To display the color palette widget shown *in the gallery* the following code can be used.

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

from tks.color_palette import PaletteSelector

if __name__ == '__main__':
    root = tk.Tk()
    root.title('Color Palette')
    entry = PaletteSelector(root)
    entry.grid(row=0, column=0)
    root.mainloop()
```

`tk.date` provides 4 classes to obtain a date from a user.

**DateVar** A Tk variable which holds a date.

**DateEntry** Displays entry boxes for year, month and day as well as a button to display a date selection dialog.

**DateDialog** Displays a dialog window allowing the user to select a date.

**DateSelector** A widget which contains the date selection machinery.

**class** `tk.date.DateVar` (*master=None, value=None, name=None*)

A Tkinter variable which holds a `datetime.date`

**class** `tk.date.DateEntry` (*master, variable=None, locale='en', fonts=None*)

A date entry widget

Creates a frame which contains entry boxes for Year, Month and Day and a button to display a date selection dialog.

Dates will always consist of a 4 digit year and 2 digit months and days only the order and separator are determined by the *locale* parameter

#### Parameters

- **master** (`ttk.Frame`) – The master frame
- **variable** (`tk.date.DateVar`) – The variable which hold the date to display in the entry boxes.
- **locale** (str or `babel.Locale`) – Determines the order of the widgets in the entry. Either a locale name e.g. 'en' or a `babel.Locale` instance. If `babel` is not installed ISO 8601 format will be used.
- **fonts** (`DefaultFonts`) – Fonts to use.

#### value

The `date` represented by the entry.

```
class tks.dates.DateDialog(master, title, start_date=None, locale='en', target_type='circle',  
                           fonts=None, colors=None)
```

Display a dialog to obtain a date from the user

#### Parameters

- **master** (`ttk.Frame`) – The master frame
- **title** (`str`) – Dialog title
- **start\_date** (`datetime.date` or `datetime.datetime`) – The date to display in the entry boxes or None for today's date
- **locale** (`str` or `babel.Locale`) – Determines how today's name is displayed Either a locale name e.g. 'en' or a babel Locale instance. If babel is not installed ISO 8601 format will be used.
- **target\_type** (`TargetShape`) – `TargetShape.Square`, `TargetShape.Rectangle` or `TargetShape.Circle`
- **fonts** (`DefaultFonts`) – Font definitions to use
- **colors** (`DefaultColors`) – Colors to use.

```
class tks.dates.DateSelector(master, start_date, locale='en', target_type='circle', fonts=None,  
                             colors=None)
```

A date selection widget

#### Parameters

- **master** (`ttk.Frame`) – The master frame
- **start\_date** (`datetime.date` or `datetime.datetime`) – The date to display in the entry boxes or None for today's date
- **locale** (`str` or `babel.Locale`) – Determines how today's name is displayed Either a locale name e.g. 'en' or a babel Locale instance. If babel is not installed ISO 8601 format will be used.
- **target\_type** (`TargetShape`) – `TargetShape.Square`, `TargetShape.Rectangle` or `TargetShape.Circle`
- **fonts** (`tks.DefaultFonts`) – Font definitions to use
- **colors** (`DefaultColors`) – Colors to use.

```
class tks.dates.TargetShape
```

How to draw the target round a date

`tk.time` provides 4 classes to obtain a time from a user.

**TimeVar** A Tk variable which holds a time.

**TimeEntry** Displays entry boxes for hours, minutes and optionally seconds and an AM/PM selector. Also contains a button to display a time selection dialog.

**TimeDialog** Displays a dialog window allowing the user to select a time.

**TimeSelector** A widget which contains the time selection machinery.

**class** `tk.time.TimeVar` (*master=None, value=None, name=None*)

A Tkinter variable which holds a `datetime.time`

**class** `tk.time.TimeEntry` (*master, variable=None, locale='en', fonts=None, show\_seconds=False*)

A time entry widget

### Parameters

- **master** (`ttk.Frame`) – The master frame
- **variable** (`tk.time.TimeVar`) – The variable which hold the date to display in the entry boxes.
- **locale** (str or `babel.Locale`) – Determines the widgets in the entry. Either a locale name e.g. 'en' or a `babel.Locale` instance. If `babel` is not installed ISO 8601 format will be used i.e. 24 hours (no am/pm) and ':' as a separator.
- **fonts** (`DefaultFonts`) – Fonts to use
- **show\_seconds** (`bool`) – If True a seconds value can be entered.

### value

The `time` represented by the entry.

**class** `tk.time.TimeDialog` (*master, title, start\_time=None, locale='en', time\_position='top', show\_seconds=False, ampm=None, fonts=None*)

Display a dialog to obtain a time from the user

### Parameters

- **master** (`ttk.Frame`) – The master frame
- **title** (`str`) – Window title.
- **start\_time** (`datetime.time`) – The time to display in the entry boxes. If not provided or `None` then today's date will be used.
- **locale** (`str` or `babel.Locale`) – Determines the widgets in the entry. Either a locale name e.g. 'en' or a babel Locale instance. If `babel` is not installed ISO 8601 format will be used i.e. 24 hours (no am/pm) and ':' as a separator.
- **time\_position** – Controls if and where a text representation of the time is displayed. Can be one of the following  
  
None - Don't display tk.TOP - Display above the clock face tk.BOTTOM - Display below the clock face
- **show\_seconds** (`bool`) – If True a seconds value can be entered.
- **ampm** – If not None display a 12 hour clock face with am/pm selection where the 1st element of the tuple is the text for AM and the 2nd element for PM.
- **fonts** (`DefaultFonts`) – Fonts definitions to use

```
class tks.times.TimeSelector(master, start_time, locale='en', ampm=None, time_position='top',  
                             show_seconds=False, fonts=None, colors=None)
```

A time selection widget.

### Parameters

- **master** (`ttk.Frame`) – The master frame
- **locale** (`str` or `babel.Locale`) – Determines the widgets in the entry. Either a locale name e.g. 'en' or a babel Locale instance. If `babel` is not installed ISO 8601 format will be used i.e. 24 hours (no am/pm) and ':' as a separator.
- **ampm** – If not None display a 12 hour clock face with am/pm selection where the 1st element of the tuple is the text for AM and the 2nd element for PM.
- **time\_position** (`str`) – Controls where to display the selected time. If `None` the time is not displayed. If `tk.TOP` then it is displayed above the clock dial, if `tk.BOTTOM` then below.
- **show\_seconds** (`bool`) – If True then the selector can also be used to set seconds.
- **fonts** (`DefaultFonts`) – Font definitions to use
- **colors** (`DefaultColors`) – Colors to use.

Used by the `TimeDialog` class but can be used independently.

### time

The selected time



`tk.colors` provides 3 classes to obtain a color from a user.

**ColorVar** A Tk variable which holds an RGB color.

**ColorEntry** Displays an entry box to enter a color as well as a button to display a color selection dialog.

**ColorDialog** Displays a dialog window allowing the user to select a color using a color wheel or sliders.

**class** `tk.colors.ColorVar` (*master=None, value=None, name=None*)

A Tkinter Variable subclass to store an RGB color tuple.

**set** (*value*)

Set the color tuple to be stored.

**class** `tk.colors.ColorEntry` (*master, variable=None, color\_format='rgbhex', fonts=None, colors=None*)

Displays an entry to enter color information and a button to display a selection dialog.

### Parameters

- **master** – Tk master widget
- **variable** (`tk.colors.ColorVar`) – The variable which hold the color to display in the entry box.
- **color\_format** (*str*) – How to display the color in the entry box. One of the following `rgbhex`, `rgb`, `hsv` or `hls`
- **fonts** – Fonts to use

**rgb**

RGB representation of the selected color

**hsv**

HSV representation of the selected color

**hls**

HLS representation of the selected color

**class** `tk.colors.ColorDialog` (*master, title, start\_color=(0.5, 0.5, 0.5), fonts=None*)

Display a dialog to obtain an RGB value.

The color is returned as an (R, G, B) tuple where each component is between 0.0 and 1.0

#### Parameters

- **master** – The master widget
- **title** (*str*) – The window title
- **start\_color** (*tuple*) – The initial (R, G, B) tuple to display.

If any element of the tuple is greater than 1.0 then it is assumed that all values need to be scaled by 255.0 both when setting and obtaining the color value.

## 5.1 `tk.color_wheel`

Implements a color wheel with an outer ring to select a hue and a triangle within that where saturation and value can be selected

**class** `tk.color_wheel.ColorWheel` (*master, variable=None, radius=125*)

Displays an HSV color wheel.

## 5.2 `tk.color_slider`

3 element sliders to change RGB, HSV and HLS variables

**class** `tk.color_slider.RGBSlider` (*master, variable=None, fonts=None*)

Bases: `tk.color_slider.ColorSlider`

An RGB Color Slider

**class** `tk.color_slider.HSVSlider` (*master, variable=None, fonts=None*)

Bases: `tk.color_slider.ColorSlider`

An HSV Color Slider

**class** `tk.color_slider.HLSlider` (*master, variable=None, fonts=None*)

Bases: `tk.color_slider.ColorSlider`

An HLS Color Slider

**class** `tk.color_slider.ColorSlider` (*master, variable=None, fonts=None*)

Color slider base class.

**from\_rgb** (*rgb*)

Convert from RGB

**to\_rgb** (*value*)

Convert to RGB

## 5.3 `tk.color_square`

ColorSquare displays a solid square which changes color depending on the value of a variable. Also displays textual color information below.

```
class tks.color_square.ColorSquare (master, variable=None, mode='rw',
                                     color_info=('rgbhex', ), dnd_target=True,
                                     dnd_source=True, fonts=None)
```

Displays a colored rectangle and a text description of the color below it. A popup menu is provided to copy the hex, RGB, HSV and HLS representations of the color.

#### Parameters

- **variable** (ColorVar) – The RGB color to display
- **mode** (*str*) – One of *r*, *w*, *rw*. If *r* is specified then the widget responds to changes in the variable. If *w* is specified then the color is written to the variable on a left mouse click.
- **color\_info** – The color information to display under the square as a text representation of the color. The elements are specified as a tuple where the following strings can be provided.  
*rgbhex*, *rgb*, *hsv*, *hls*
- **dnd\_target** (*bool*) – If True then the square responds to colors being dropped on it.
- **dnd\_source** (*bool*) – If True the square works as a drag and drop source.

#### rgb

The RGB tuple to display. If None the the rectangle is cleared and the text set to the empty string.

#### dnd\_accept (source, event)

Indicate that we can handle a drag and drop operation.

#### dnd\_enter (source, event)

Called by the drag and drop machinery when the mouse enters the canvas.

#### dnd\_motion (source, event)

Called by the drag and drop machinery when the mouse moves within the canvas.

#### dnd\_leave (source, event)

Called by the drag and drop machinery when the mouse leaves the canvas.

#### dnd\_commit (source, event)

Called by the drag and drop machinery when the mouse is released over the canvas.

#### dnd\_end (target, event)

Called by the drag and drop machinery to end the operation.

## 5.4 tks.color\_tints\_and\_shades

Display tints and shades of a base color. This module provides 2 classes both of which display a number of ColorSquares enclosed in a LabelFrame along with a scale to adjust the percent between the colors.

**ColorTint** Displays a set of tints

**ColorShade** Displays a set of shades

```
class tks.color_tints_and_shades.ColorTint (master, variable, count=5, percent=(1, 5))
```

Displays a sequence of tints.

#### Parameters

- **variable** (tks.vars.ColorVar) – The RGB color to produce a set of tints from
- **count** (*int*) – The number of tints to display

- **percent** (*tuple*) – Determines the percent between the tints specified as a min and a max range.

**class** `tk.color_tints_and_shades.ColorShade` (*master, variable, count=5, percent=(1, 5)*)  
Displays a sequence of shades.

#### Parameters

- **variable** (`tk.vars.ColorVar`) – The RGB color to produce a set of shades from
- **count** (*int*) – The number of shades to display
- **percent** (*tuple*) – Determines the percentage percent between the shades specified as a min and a max range.

## 5.5 `tk.color_palette`

A Tkinter widget to select colors from a palette of colors

**class** `tk.color_palette.PaletteSelector` (*master, variable=None, height=400*)  
A widget to display a set of colors from a palette.

---

### Filesystem Entry Handling

---

Allows selection of filesystem entries (directories and files).

**DirEntry** Displays a text box and a button to select a directory using the system defined selection method.

**FileEntry** Displays a text box and a button to select a file using the system defined selection method.

**class** `tk.fs.DirEntry` (*master*, *variable=None*, *\*\*options*)  
Entry box and a button to select a directory.

**class** `tk.fs.FileEntry` (*master*, *variable=None*, *\*\*options*)  
Entry box and a button to select a file.



---

## Password Handling

---

`tk.passwords` provides 2 classes to obtain a password from a user

**PasswordEntry** Displays a widget which allows the user to enter a password

**PasswordDialog** Displays a dialog which allows the user to generate a random password.

```
class tk.passwords.PasswordEntry (master, variable=None, show_hide=True, compare=True,  
                                generate=True, fonts=None, colors=None)
```

A password entry widget

Creates a frame which contains an entry box for a password and optionally

- A checkbox to show and hide the password
- A second entry box to make sure the password was entered as the user thought.
- A button which displays a dialog to generate a random password.

### Parameters

- **master** (`ttk.Frame`) – The master frame
- **variable** (`StringVar`) – The variable which hold the date to display in the entry boxes.
- **show\_hide** (`bool`) – If True a show/hide checkbox is added to show the actual characters.
- **compare** (`bool`) – If True a second entry box is shown to as a comparison.
- **generate** (`bool`) – If True a button is shown which displays a password generator when clicked.
- **fonts** (`DefaultFonts`) – Fonts to use.
- **colors** (`DefaultColors`) – Fonts to use.

```
class tk.passwords.PasswordDialog (master, title, fonts=None, colors=None)
```

A Password Generator Dialog

### Parameters

- **master** (`ttk.Frame`) – The master frame
- **title** (`str`) – Dialog title



### 8.1 The tks Module

tkStuff - A collection of Tk widgets

**class** `tks.DefaultColors`  
A container for color names.

**class** `tks.DefaultFonts`  
A container for font information.

**class** `tks.PickleVar` (*master=None, value=None, name=None*)  
A Tkinter variable which stores values as pickled objects.

`tks.load_colors()`  
Load color definitions from the *.tksrc* file

`tks.load_fonts()`  
Load font definitions from the *.tksrc* file

`tks.parse_geometry` (*geom*)  
Return a width, height, x, y tuple from a Tk geometry.

`tks.rect_at` (*point, size, size\_y=-1*)  
Returns a rectangle centred at *point*. If only *size* is provided then the rectangle will be a square. The dimensions will be *size* \* 2.

`tks.rect_center` (*rect*)  
Return the centre of a rectangle as an (x, y) tuple.

### 8.2 Color Functions (tks.color\_funcs)

Various functions to manipulate RGB hex, RGB, HSV and HLS colors.

`tk.color_funcs.clamp(value)`  
Clamp a float between 0.0 and 1.0

`tk.color_funcs.clamped_tuple(value)`  
Clamps the values in a tuple between 0.0 and 1.0

`tk.color_funcs.color_string_to_color(value, allow_short_hex=True)`  
Convert a color string to a (color format, value) tuple where the color format is one of *rgbhex*, *rgb*, *hsv* or *hls*

`tk.color_funcs.color_string_to_rgb(value)`  
Return an RGB tuple from a color string.

`tk.color_funcs.color_string_to_tuple(value)`  
Convert a color string to a tuple of floats.

`tk.color_funcs.contrast_color(rgb)`  
Return either white or black whichever provides the most contrast

`tk.color_funcs.hex_string_to_rgb(value, allow_short=True)`  
Convert from a hex color string of the form *#abc* or *#abcdef* to an RGB tuple.

#### Parameters

- **value** (*str*) – The value to convert
- **allow\_short** (*bool*) – If True then the short of form of an hex value is accepted e.g. *#fff*

`tk.color_funcs.luminosity_transform(color, luminosity=0.05)`  
Transform an RGB color by a luminosity.

If luminosity is a tuple then the 3 elements are used to transform the red, green and blue values individually. If a float then the same value is used to transform all 3 elements.

`tk.color_funcs.rgb_intensity(rgb)`  
Convert an RGB color to its intensity

`tk.color_funcs.rgb_shade(rgb, percent=5)`  
Create a shade of the RGB color

#### Parameters

- **rgb** (*tuple*) – The RGB value for which to calculate the tint
- **percent** (*int*) – Determines the percent between the specified color and the shade

`tk.color_funcs.rgb_shades(rgb, base_percent, count, linear=True)`  
Produce a list of shades from the base color

#### Parameters

- **rgb** (*tuple*) – The RGB value for which to calculate the shades
- **base\_percent** (*float*) – Determines the factor between the returned colors
- **count** (*int*) – The number of shades to return

`tk.color_funcs.rgb_tint(rgb, percent=5)`  
Create a tinted version of the RGB color

#### Parameters

- **rgb** (*tuple*) – The RGB value for which to calculate the tint
- **percent** (*int*) – Determines the percent between the specified color and the tint

`tk.color_funcs.rgb_tints(rgb, base_percent, count, linear=True)`

Produce a list of tints from the base color

#### Parameters

- **rgb** (*tuple*) – The RGB value for which to calculate the tints
- **base\_percent** (*float*) – Determines the factor between the returned colors
- **count** (*int*) – The number of tints to return

`tk.color_funcs.rgb_to_hex_string(value)`

Convert from an (R, G, B) tuple to a hex color.

**Parameters** **value** (*tuple*) – The RGB value to convert

R, G and B should be in the range 0.0 - 1.0

`tk.color_funcs.rgb_to_hls_string(value, dp=3)`

Convert from an (R, G, B) tuple to an HLS string.

#### Parameters

- **value** (*tuple*) – The RGB value to convert
- **dp** (*int*) – Number of decimal places in the string

R, G and B should be in the range 0.0 - 1.0

`tk.color_funcs.rgb_to_hsv_string(value, dp=3)`

Convert from an (R, G, B) tuple to an HSV string.

#### Parameters

- **value** (*tuple*) – The RGB value to convert
- **dp** (*int*) – Number of decimal places in the string

R, G and B should be in the range 0.0 - 1.0

`tk.color_funcs.rgb_to_rgb_string(value, dp=3)`

Convert from an (R, G, B) tuple to an RGB string.

#### Parameters

- **value** (*tuple*) – The RGB value to convert
- **dp** (*int*) – Number of decimal places in the string

R, G and B should be in the range 0.0 - 1.0

## 8.3 Icon Functions (tk.icon)

`tk.icon.set_icon_from_data(root, image_data)`

Set the icon for a root window from image data.

#### Parameters

- **root** – Toplevel window for which to set the icon.
- **image\_data** (*bytes*) – The image data to use

`tk.icon.set_icon_from_file(root, filename)`

Set the icon for a root window from a file.

#### Parameters

- **root** – Toplevel window for which to set the icon.
- **filename** (*str*) – The filename to read the image data from

`tkstuff.icon.set_icon_from_resource (root, package, resource)`

Set the icon for a root window to a resource

#### Parameters

- **root** – Toplevel window for which to set the icon.
- **package** (*str*) – The package in which to find the resource
- **resource** (*str*) – The name of the resource

tkStuff is a collection of Tkinter widgets for dates, times and colors.

For each of these there is an entry widget and a dialog for obtaining values from the user.

For colors, each element of the dialog is also exposed as a class (classes exist for a *color wheel*, *3 element sliders*, *a color information square*.)

A *color palette* widget is also provided access to the set of colors defined by X11 and CSS3.

Finally there are 2 widgets to display *tints and shades* of a color.

Examples of these can be seen in the *Gallery* and some code examples in the *Code Examples* section.

## CHAPTER 9

---

### License

---

This module is licensed under the terms of the [Apache V2.0](#) license.



## CHAPTER 10

---

### Dependencies

---

tkStuff has one required dependency and an optional one. Either [Pillow](#) or [PIL](#) are required and [Babel](#) is optional.

- [PIL](#) is used to generate the color wheel.
- [Babel](#) is used to provide improved date and time handling and if [Babel](#) not present dates are limited to the ISO 8601 YYYY-MM-DD format and times are limited to a 24 hour clock.





# CHAPTER 11

---

## Installation

---

Installation is performed from [PyPi](#) via **pip**

```
pip install tks
```

This doesn't automatically install Babel. To do this either install it separately or use the following command for the **improved date/time handling**

```
pip install tks[idth]
```



## CHAPTER 12

---

### Contact

---

Simon Kennedy <[sffjunkie+code@gmail.com](mailto:sffjunkie+code@gmail.com)>



## CHAPTER 13

---

### Version History

---

Version	Description
0.2	<ul style="list-style-type: none"><li>• Added <code>*Entry</code> classes for files, directories and passwords.</li><li>• Added functions to set a Tk window icon.</li><li>• <code>*Entry</code> classes now take a Tk variable subclass instead of a value.</li><li>• Colors and fonts to be used for interface elements can be specified using an <i>rc</i> file.</li><li>• Added a decimal places parameter to the color to string conversion functions.</li></ul>
0.1	Initial release



### t

- `tk`, [21](#)
- `tk.color_funcs`, [21](#)
- `tk.color_palette`, [16](#)
- `tk.color_slider`, [14](#)
- `tk.color_square`, [14](#)
- `tk.color_tints_and_shades`, [15](#)
- `tk.color_wheel`, [14](#)
- `tk.colors`, [13](#)
- `tk.dates`, [9](#)
- `tk.fs`, [17](#)
- `tk.passwords`, [19](#)
- `tk.times`, [11](#)





## C

clamp() (in module tks.color\_funcs), 21  
clamped\_tuple() (in module tks.color\_funcs), 22  
color\_string\_to\_color() (in module tks.color\_funcs), 22  
color\_string\_to\_rgb() (in module tks.color\_funcs), 22  
color\_string\_to\_tuple() (in module tks.color\_funcs), 22  
ColorDialog (class in tks.colors), 13  
ColorEntry (class in tks.colors), 13  
ColorShade (class in tks.color\_tints\_and\_shades), 16  
ColorSlider (class in tks.color\_slider), 14  
ColorSquare (class in tks.color\_square), 14  
ColorTint (class in tks.color\_tints\_and\_shades), 15  
ColorVar (class in tks.colors), 13  
ColorWheel (class in tks.color\_wheel), 14  
contrast\_color() (in module tks.color\_funcs), 22

## D

DateDialog (class in tks.dates), 9  
DateEntry (class in tks.dates), 9  
DateSelector (class in tks.dates), 10  
DateVar (class in tks.dates), 9  
DefaultColors (class in tks), 21  
DefaultFonts (class in tks), 21  
DirEntry (class in tks.fs), 17  
dnd\_accept() (tks.color\_square.ColorSquare method), 15  
dnd\_commit() (tks.color\_square.ColorSquare method), 15  
dnd\_end() (tks.color\_square.ColorSquare method), 15  
dnd\_enter() (tks.color\_square.ColorSquare method), 15  
dnd\_leave() (tks.color\_square.ColorSquare method), 15  
dnd\_motion() (tks.color\_square.ColorSquare method), 15

## F

FileEntry (class in tks.fs), 17  
from\_rgb() (tks.color\_slider.ColorSlider method), 14

## H

hex\_string\_to\_rgb() (in module tks.color\_funcs), 22  
hls (tks.colors.ColorEntry attribute), 13

HLSSlider (class in tks.color\_slider), 14  
hsv (tks.colors.ColorEntry attribute), 13  
HSVSlider (class in tks.color\_slider), 14

## L

load\_colors() (in module tks), 21  
load\_fonts() (in module tks), 21  
luminosity\_transform() (in module tks.color\_funcs), 22

## P

PaletteSelector (class in tks.color\_palette), 16  
parse\_geometry() (in module tks), 21  
PasswordDialog (class in tks.passwords), 19  
PasswordEntry (class in tks.passwords), 19  
PickleVar (class in tks), 21

## R

rect\_at() (in module tks), 21  
rect\_center() (in module tks), 21  
rgb (tks.color\_square.ColorSquare attribute), 15  
rgb (tks.colors.ColorEntry attribute), 13  
rgb\_intensity() (in module tks.color\_funcs), 22  
rgb\_shade() (in module tks.color\_funcs), 22  
rgb\_shades() (in module tks.color\_funcs), 22  
rgb\_tint() (in module tks.color\_funcs), 22  
rgb\_tints() (in module tks.color\_funcs), 22  
rgb\_to\_hex\_string() (in module tks.color\_funcs), 23  
rgb\_to\_hls\_string() (in module tks.color\_funcs), 23  
rgb\_to\_hsv\_string() (in module tks.color\_funcs), 23  
rgb\_to\_rgb\_string() (in module tks.color\_funcs), 23  
RGBSlider (class in tks.color\_slider), 14

## S

set() (tks.colors.ColorVar method), 13  
set\_icon\_from\_data() (in module tks.icon), 23  
set\_icon\_from\_file() (in module tks.icon), 23  
set\_icon\_from\_resource() (in module tks.icon), 24

## T

TargetShape (class in tks.dates), 10

- time (tk.time.TimeSelector attribute), [12](#)
- TimeDialog (class in tk.time), [11](#)
- TimeEntry (class in tk.time), [11](#)
- TimeSelector (class in tk.time), [12](#)
- TimeVar (class in tk.time), [11](#)
- tk (module), [21](#)
- tk.color\_funcs (module), [21](#)
- tk.color\_palette (module), [16](#)
- tk.color\_slider (module), [14](#)
- tk.color\_square (module), [14](#)
- tk.color\_tints\_and\_shades (module), [15](#)
- tk.color\_wheel (module), [14](#)
- tk.colors (module), [13](#)
- tk.dates (module), [9](#)
- tk.fs (module), [17](#)
- tk.passwords (module), [19](#)
- tk.time (module), [11](#)
- to\_rgb() (tk.color\_slider.ColorSlider method), [14](#)

## V

- value (tk.dates.DateEntry attribute), [9](#)
- value (tk.time.TimeEntry attribute), [11](#)