



Tkinter Combobox

If this Python Tutorial saves you hours of work, please **whitelist it in your ad blocker** 🙏 and

Donate Now

(<https://www.pythontutorial.net/donation/>)

to help us ❤️ pay for the web hosting fee and CDN to keep the

website running.

Summary: in this tutorial, you'll learn to create a Tkinter combobox widget that allows users to select one value from a set of values.

Introduction to the Tkinter Combobox widget

A combobox is a combination of an [Entry](https://www.pythontutorial.net/tkinter/tkinter-entry/) widget and a [Listbox](https://www.pythontutorial.net/tkinter/tkinter-listbox/) widget. A combobox widget allows you to select one value in a set of values. In addition, it allows you to enter a custom value.

Create a combobox

To create a combobox widget, you'll use the `ttk.Combobox()` constructor. The following example creates a combobox widget and links it to a [string variable](https://www.pythontutorial.net/tkinter/tkinter-stringvar/):

```
current_var = tk.StringVar()
combobox = ttk.Combobox(container, textvariable=current_var)
```

The `container` is the [window](https://www.pythontutorial.net/tkinter/tkinter-window/) or [frame](https://www.pythontutorial.net/tkinter/tkinter-frame/) on which you want to place the combobox widget.

The `textvariable` argument links a variable `current_var` to the current value of the combobox.

To get the currently selected value, you can use the `current_var` variable:

```
current_value = current_var.get()
```

Alternatively, you can use the `get()` method of the combobox object:

```
current_value = combobox.get()
```

To set the current value, you use the `current_var` variable or the `set()` method of the combobox object:

```
current_value.set(new_value)  
combobox.set(new_value)
```

Define value list

The combobox has the `values` property that you can assign a list of values to it like this:

```
combobox['values'] = ('value1', 'value2', 'value3')
```

By default, you can enter a custom value in the combobox. If you don't want this, you can set the `state` option to `'readonly'` :

```
combobox['state'] = 'readonly'
```

To re-enable editing the combobox, you use the `'normal'` state like this:

```
combobox['state'] = 'normal'
```

Bind events

When a select value changes, the combobox widget generates a `'<<ComboboxSelected>>'` virtual event. To handle the event, you can use the `bind()` method like this:

```
combobox.bind('<<ComboboxSelected>>', callback)
```

In this example, the callback function will execute when the selected value of the combobox changes.

Set the current value

To set the current value, you use the `set()` method:

```
combobox.set(self, value)
```

Also, you can use the `current()` method:

```
current(self, newindex=None)
```

The `newindex` specifies the index of values from the list that you want to select as the current value.

If you don't specify the `newindex`, the `current()` method will return the index of the current value in the list of values or `-1` if the current value doesn't appear in the list.

Python Tkinter combobox example

The following program illustrates how to create a combobox widget:

```
import tkinter as tk
from tkinter import ttk
from tkinter.messagebox import showinfo
from calendar import month_name

root = tk.Tk()

# config the root window
root.geometry('300x200')
root.resizable(False, False)
root.title('Combobox Widget')
```

```
# Label
label = ttk.Label(text="Please select a month:")
label.pack(fill=tk.X, padx=5, pady=5)

# create a combobox
selected_month = tk.StringVar()
month_cb = ttk.Combobox(root, textvariable=selected_month)

# get first 3 letters of every month name
month_cb['values'] = [month_name[m][0:3] for m in range(1, 13)]

# prevent typing a value
month_cb['state'] = 'readonly'

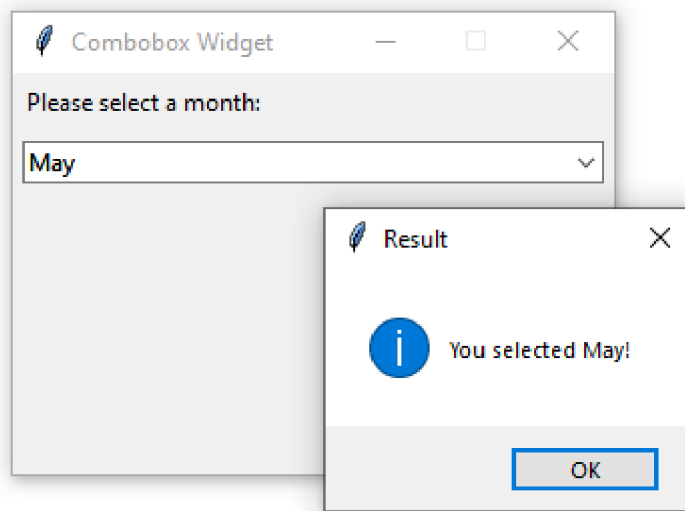
# place the widget
month_cb.pack(fill=tk.X, padx=5, pady=5)

# bind the selected value changes
def month_changed(event):
    """ handle the month changed event """
    showinfo(
        title='Result',
        message=f'You selected {selected_month.get()}!'
    )

month_cb.bind('<<ComboboxSelected>>', month_changed)

root.mainloop()
```

Output:



The following program shows the same month combobox widget and uses the `set()` method to set the current value to the current month:

```
import tkinter as tk
from tkinter import ttk
from tkinter.messagebox import showinfo
from calendar import month_name
from datetime import datetime

root = tk.Tk()

# config the root window
root.geometry('300x200')
root.resizable(False, False)
root.title('Combobox Widget')

# Label
label = ttk.Label(text="Please select a month:")
label.pack(fill=tk.X, padx=5, pady=5)

# create a combobox
selected_month = tk.StringVar()
month_cb = ttk.Combobox(root, textvariable=selected_month)

# get first 3 letters of every month name
```

```
month_cb['values'] = [month_name[m][0:3] for m in range(1, 13)]

# prevent typing a value
month_cb['state'] = 'readonly'

# place the widget
month_cb.pack(fill=tk.X, padx=5, pady=5)

# bind the selected value changes
def month_changed(event):
    """ handle the month changed event """
    showinfo(
        title='Result',
        message=f'You selected {selected_month.get()}!'
    )

month_cb.bind('<<ComboboxSelected>>', month_changed)

# set the current month
current_month = datetime.now().strftime('%b')
month_cb.set(current_month)

root.mainloop()
```

Summary

- Use `ttk.Combobox(root, textvariable)` to create a combobox.
- Set the `state` property to `readonly` to prevent users from entering custom values.
- A combobox widget emits the `'<<ComboboxSelected>>'` event when the selected value changes.