

Overview

A **combobox** widget combines an entry with a list of choices. This lets users either choose from a set of values you've provided (e.g., typical settings), but also put in their own value (e.g., for less common cases).

Important Information

Creating a Combobox

Comboboxes are created using the `ttk.Combobox` class. In this example, we will create a combobox that allows the user to select a country:

```
# Create a variable to track the selected value from the user
country = StringVar()
# Create a combobox with the textvariable linked to country
# parent represents whatever container you want to put it in. This can be
# the root, or a frame.
country_box = ttk.Combobox(parent, textvariable=country)
```

Predefined Values

You can provide a list of values that users can choose from using the `values` configuration option:

```
# Add a list of values to the combobox for the user to select from
country_box['values'] = ['USA', 'Canada', 'Australia']
```

If set, the `readonly` state flag will restrict users to making choices only from the list of predefined values. **If you do not set the read only flag, users may type their own text into the combobox, but it won't change the value of the variable if what the user typed is not in the list you created.**

```
# Ensure that users can't type their own text into the combobox
country_box.state(['readonly'])
```

Watching For Combobox Changes

A combobox will generate a `<<ComboboxSelected>>` virtual event that you can bind to whenever its value changes.

```
# Create a function to bind to the combobox
def combobox_changed(*args):
    print("Value Changed!")

# Bind the combobox to the function when it is changed
country_box.bind('<<ComboboxSelected>>', combobox_changed)
```

You can also get the current value using the `get()` method and change the current value using the `set()` method (which takes a single argument, the new value).

```
# Set the value by changing the variable directly
country.set("Canada")
# Get the value by checking the variable directly
current_country = country.get()
```

To complement the `get()` and `set()` methods, you can also use the `current()` method to determine which item in the predefined values list is selected. If you call `current()` with no arguments it will return a 0-based index into the list or `-1` if the current value is not in the list.

```
# Return the index of the current selected element
# Return -1 if none of the elements from the predefined list are selected
index = country_box.current()
```

You can select an item in the list by calling `current()` with a single 0-based index argument.

```
# Change the value to a specific index
country_box.current(1)
```

Copy, Change, Challenge

Copy

```
from tkinter import *
from tkinter import ttk

root = Tk()
root.title("Combobox Copy Example")

# Create a variable to track the selected value
country = StringVar()

# Create the combobox
country_box = ttk.Combobox(root, textvariable=country)
country_box['values'] = ["USA", "Canada", "Australia"]

# Restrict input to the predefined values
country_box.state(['readonly'])

# Place the combobox in the window
country_box.grid()

root.mainloop()
```

Change

Modify the program so that:

- A **label** is added to the left of the combobox that says "**Country:**"
- The default selected value is "**Canada**"
- The combobox allows users to **type their own value** instead of being restricted to the list

Challenge

Create a program that uses a combobox to select a **favourite programming language**:

- The combobox must include **at least five predefined options**
- Add a **label** that displays the currently selected language
- When the selection changes:
 - Update the label to show the new value
 - Print the **index** of the selected option to the terminal using `current()`
- If the selected value is **not in the predefined list**, display `-1` as the index