Tkinter Checkbox

Summary: in this tutorial, you'll learn about the Tkinter Checkbox widget and how to use it effectively.

Introduction to the Tkinter checkbox widget

A checkbox is a widget that allows you to check and uncheck.

A checkbox can hold a value and invoke a callback when it's checked or unchecked.

Typically, you use a checkbox when you want to ask users to choose between two values.

To create a checkbox, you use the `ttk.Checkbutton` constructor:

```
checkbox_var = tk.StringVar()

def check_changed():
    #...

checkbox = ttk.Checkbutton(container,
                text='<checkbox label>',
                command=check_changed,
                variable=checkbox_var,
                onvalue='<value_when_checked>',
                offvalue='<value_when_unchecked>')
```

The `container` argument specifies the window that you want to place the checkbox.

The `text` argument specifies the label for the checkbox.

The `command` is a callable that will be called once the checkbox is checked or unchecked.

The `variable` holds the current value of the checkbox. If the checkbox is checked, the value of the variable is 1. Otherwise, it is 0.

If you want other values than 0 and 1, you can specify them in the `onvalue` and `offvalue` options.

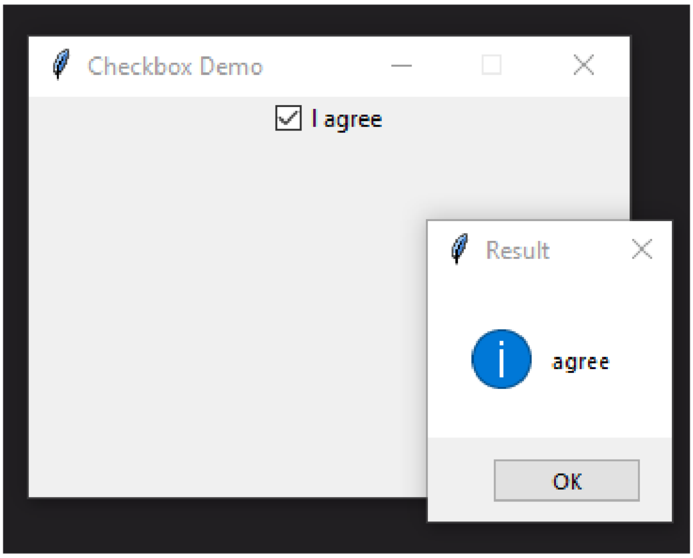
If the linked variable doesn't exist, or its value is neither the on value nor off value, the checkbox is in the indeterminate or tristate mode.

Tkinter checkbox example

The following program illustrates how to use a checkbox widget. Once you check or uncheck the checkbox, a message box will show the on value and the off value accordingly:

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

root = tk.Tk()
root.geometry('300x200')
root.resizable(False, False)
root.title('Checkbox Demo')

agreement = tk.StringVar()


def agreement_changed():
    tk.messagebox.showinfo(title='Result',
                        message=agreement.get())


ttk.Checkbutton(root,
                text='I agree',
                command=agreement_changed,
                variable=agreement,
                onvalue='agree',
                offvalue='disagree').pack()


root.mainloop()
```

Output:



How it works.

First, create a string variable that will hold the value of the checkbox:

```
agreement = tk.StringVar()
```

Second, define a function that will be called once the state of the checkbox changed. The function shows the value of the checkbox:

```
def agreement_changed():
    tk.messagebox.showinfo(title='Result',
                                message=agreement.get())
```

Third, create a checkbox widget and set its options accordingly:

```
ttk.Checkbutton(root,
                text='I agree',
                command=agreement_changed,
                variable=agreement,
                onvalue='agree',
                offvalue='disagree').pack()
```

Summary

- Use `ttk.Checkbutton(text, variable)` to create a checkbox; the variable is a `tk.StringVar()` .
- Use `command` argument to specify a function that executes when the button is checked or unchecked.
- Use the `onvalue` and `offvalue` to determine what value the `variable` will take.