Tkinter Spinbox

Summary: in this tutorial, you'll learn how to create Tkinter Spinbox widgets.

Introduction to the Tkinter Spinbox widget

A Spinbox widget allows you to select a value from a set of values. The values can be a range of numbers.

A Spinbox has an area for showing the current value and a pair of arrowheads.

When you click the upward-pointing arrowhead, the Spinbox advances the current value to the next higher value in the sequence. If the current value reaches the maximum value, you can set it to the minimum value.

On the other hand, if you click the downward-pointing arrowhead, the Spinbox advances the current value to the next lower value in the sequence. If the current value reaches the lowest value, you can set it to the maximum value.

Also, you can enter a value directly into the Spinbox widget as if it were an Entry (https://www.pythontutorial.net/tkinter/tkinter-entry/) widget.

To create a Spinbox widget, you use the `ttk.Spinbox` constructor. Here is a typical options:

```
ttk.Spinbox(container, from_, to, textvariable, wrap)
```

In this syntax:

- The `container` is the parent component of the Spinbox widget.
- The `from_` is the minimum value.
- The `to` is the maximum value.
- The `textvariable` specifies a `tk.StringVar` object that holds the current value of the Spinbox.
- The `wrap` is a Boolean value. If `wrap` equals `True`, when the current value reaches the maximum value, it's set to the lowest value if you click the upward-pointing arrowhead and vice versa. In case `wrap` equals `False`, it's set to the maximum value if you click the downward-pointing arrowhead.

Note that the `ttk.Spinbox` has been available since Python 3.7. If you use the lower version, you need to use the `tk.Spinbox`.

Getting the current value

To get the current value of the Spinbox, you can access the textvariable. For example:

```
current_value = tk.StringVar(value=0)
spin_box = ttk.Spinbox(
    container,
    from_=0,
    to=30,
    textvariable=current_value,
    wrap=True)
```

In this example, the `current_value` holds the current value of the Spinbox. And you can get it by calling the get method:

```
current_value.get()
```

Also, you can use the `get()` method of the Spinbox object:

```
spin_box.get()
```

Executing a function

To execute a function when the value of the Spinbox changes, you can assign that function to the `command` option. For example:

```
def value_changed():
    print(current_value.get())

current_value = tk.StringVar(value=0)

spin_box = ttk.Spinbox(
    container,
    from_=0,
    to=30,
    textvariable=current_value,
    command=value_changed)
```

In this example, the `value_changed` function will execute automatically whenever the value of the Spinbox changes.

Setting discrete steps

To set a list of discrete steps for a Spinbox, you assign a tuple of discrete numbers to the `values` option like this:

```
ttk.Spinbox(
    ...
    values=tuple
    ...
)
```

Tkinter Spinbox widget examples

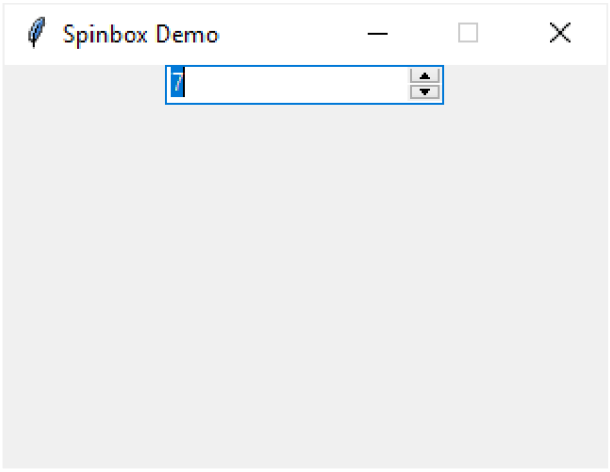
Let's take some example of creating a Spinbox widget.

1) A simple Tkinter Spinbox widget example

The following program illustrates how to use the Spinbox:

```
import tkinter as tk
from tkinter import ttk



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

# Spinbox
current_value = tk.StringVar(value=0)
spin_box = ttk.Spinbox(
    root,
    from_=0,
    to=30,
    textvariable=current_value,
    wrap=True)

spin_box.pack()

root.mainloop()
```

Output:



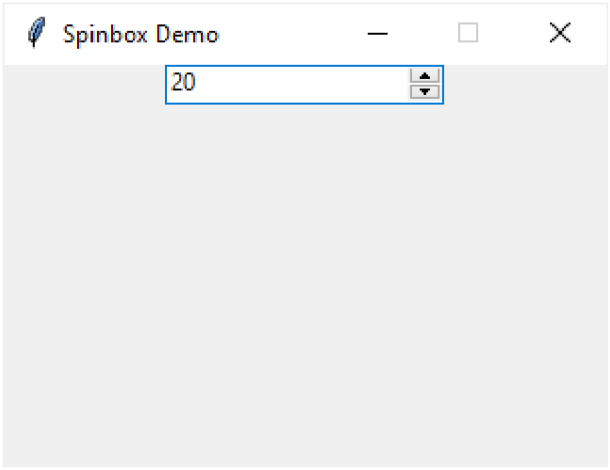
2) Tkinter Spinbox with discrete steps

The following example shows how to create a Spinbox with discrete steps:

```
import tkinter as tk
from tkinter import ttk


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


# spinbox
current_value = tk.StringVar()
spin_box = ttk.Spinbox(
    root,
    from_=0,
    to=50,
    values=(0, 10, 20, 30, 40, 50),
    textvariable=current_value,
    wrap=True)

spin_box.pack()
```

```
root.mainloop()
```

Output:



Summary

- Use `ttk.Spinbox(container, **options)` to create a Spinbox.
- Set `wrap=True` to set the current value to the minimum value when it reaches the maximum value and vice versa.