



Tkinter Progressbar

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 about the Tkinter Progressbar widget.

Introduction to the Tkinter Progressbar widget

A Progressbar widget allows you to give feedback to the user about the progress of a long-running task. To create a Progressbar widget, you use the `ttk.Progressbar` class:

```
ttk.Progressbar(container, **options)
```

The following shows the typical parameters to create a Progressbar widget:

```
ttk.Progressbar(container, orient, length, mode)
```

In this syntax:

- The `container` is the parent component of the progressbar.
- The `orient` can be either `'horizontal'` or `'vertical'`.
- The `length` represents the width of a horizontal progress bar or the height of a vertical progressbar.

- The `mode` can be either `'determinate'` or `'indeterminate'` .

The indeterminate mode

In the `indeterminate` mode, the progressbar shows an indicator that bounces back and forth between the ends of the widget.

Typically, you use the `indeterminate` mode when you don't know how to accurately measure the time that the long-running task takes to complete.

The determinate mode

In the `determinate` mode, the progressbar shows an indicator from the beginning to the end of the widget.

If you know how to measure relative progress, you can use the `determinate` mode.

The important methods of a progressbar

The Progressbar has the following important methods:

- `start([interval])` – start moving the indicator every `interval` millisecond. The `interval` defaults to 50ms.
- `step([delta])` – increase the indicator value by `delta`. The `delta` defaults to 1 millisecond.
- `stop()` – stop moving the indicator of the progressbar.

Tkinter Progressbar examples

Let's take some examples of creating progressbar widgets.

1) Tkinter Progressbar in the indeterminate mode example

The following program illustrates how to create a progressbar in the `indeterminate` mode. If you click the `start` button, the progressbar starts moving the indicator. When you click the `stop` button, the progressbar stops moving the progress indicator:

```
import tkinter as tk
from tkinter import ttk
```

```
# root window
root = tk.Tk()
root.geometry('300x120')
root.title('Progressbar Demo')

root.grid()

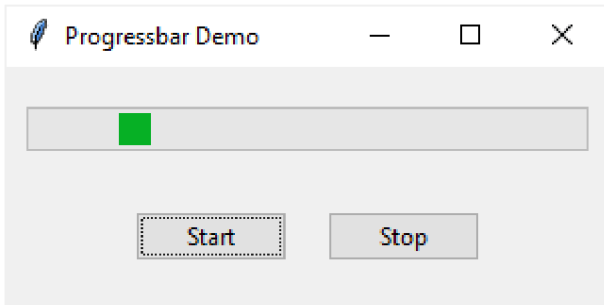
# progressbar
pb = ttk.Progressbar(
    root,
    orient='horizontal',
    mode='indeterminate',
    length=280
)
# place the progressbar
pb.grid(column=0, row=0, columnspan=2, padx=10, pady=20)

# start button
start_button = ttk.Button(
    root,
    text='Start',
    command=pb.start
)
start_button.grid(column=0, row=1, padx=10, pady=10, sticky=tk.E)

# stop button
stop_button = ttk.Button(
    root,
    text='Stop',
    command=pb.stop
)
stop_button.grid(column=1, row=1, padx=10, pady=10, sticky=tk.W)
```

```
root.mainloop()
```

Output:



How it works.

First, create a horizontal progressbar whose length is 280 pixels and mode is `'indeterminate'` :

```
pb = ttk.Progressbar(  
    root,  
    orient='horizontal',  
    mode='indeterminate',  
    length=280  
)
```

Second, pass the `Progressbar.start` method to the command of the `start` button:

```
start_button = ttk.Button(  
    root,  
    text='Start',  
    command=pb.start  
)
```

Third, pass the `Progressbar.stop` method to the command of the `stop` button:

```
stop_button = ttk.Button(  
    root,  
    text='Stop',
```

```
        command=pb.stop  
    )
```

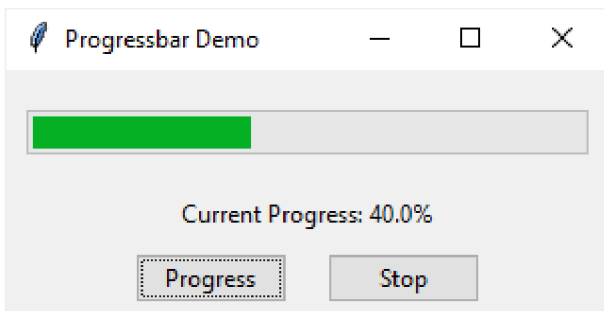
2) Tkinter Progressbar in the determinate mode example

The following program shows how to use a progressbar in the **determinate** mode:

```
from tkinter import ttk  
import tkinter as tk  
from tkinter.messagebox import showinfo  
  
# root window  
root = tk.Tk()  
root.geometry('300x120')  
root.title('Progressbar Demo')  
  
def update_progress_label():  
    return f"Current Progress: {pb['value']}%"  
  
def progress():  
    if pb['value'] < 100:  
        pb['value'] += 20  
        value_label['text'] = update_progress_label()  
    else:  
        showinfo(message='The progress completed!')  
  
def stop():  
    pb.stop()  
    value_label['text'] = update_progress_label()  
  
# progressbar
```

```
pb = ttk.Progressbar(  
    root,  
    orient='horizontal',  
    mode='determinate',  
    length=280  
)  
  
# place the progressbar  
pb.grid(column=0, row=0, columnspan=2, padx=10, pady=20)  
  
# Label  
value_label = ttk.Label(root, text=update_progress_label())  
value_label.grid(column=0, row=1, columnspan=2)  
  
# start button  
start_button = ttk.Button(  
    root,  
    text='Progress',  
    command=progress  
)  
start_button.grid(column=0, row=2, padx=10, pady=10, sticky=tk.E)  
  
stop_button = ttk.Button(  
    root,  
    text='Stop',  
    command=stop  
)  
stop_button.grid(column=1, row=2, padx=10, pady=10, sticky=tk.W)  
  
root.mainloop()
```

Output:



How it works.

First, create a progressbar in the `determinate` mode:

```
pb = ttk.Progressbar(  
    root,  
    orient='horizontal',  
    mode='determinate',  
    length=280  
)
```

Second, bind the `progress()` function to the click event of the `progress` button. Once the button is clicked, the value of the Progressbar is increased by 20% and the progress label is updated. Also, the program shows a message box indicating that the progress is completed if the value reaches 100:

```
def progress():  
    if pb['value'] < 100:  
        pb['value'] += 20  
        value_label['text'] = update_progress_label()  
    else:  
        showinfo(message='The progress completed!')
```

Third, bind the `stop()` function to the click event of the `stop` button. Also, the `stop()` function will update the progress label.

```
def stop():  
    pb.stop()  
    value_label['text'] = update_progress_label()
```

Summary

- Use the `ttk.Progressbar(container, orient, length, mode)` to create a progressbar.
- Use the `indeterminate` mode when the program cannot accurately know the relative progress to display.
- Use the `determinate` mode if you know how to measure the progress accurately.
- Use the `start()` , `step()` , and `stop()` methods to control the current value of the progressbar.