



Tkinter Window

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.

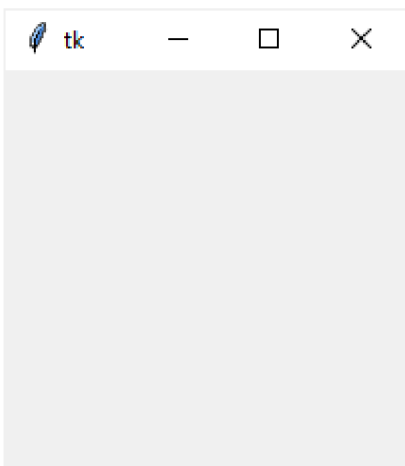
Summary: in this tutorial, you'll learn how to manipulate various attributes of a Tkinter window.

Let's start with a simple program that consists of a window:

```
import tkinter as tk

root = tk.Tk()
root.mainloop()
```

Output:



The root window has a title that defaults to `tk` . It also has three system buttons including Minimize, Maximize, and Close.

Let's learn how to change the attributes of the root window.

Changing the window title

To change the window's title, you use the `title()` method like this:

```
window.title(new_title)
```

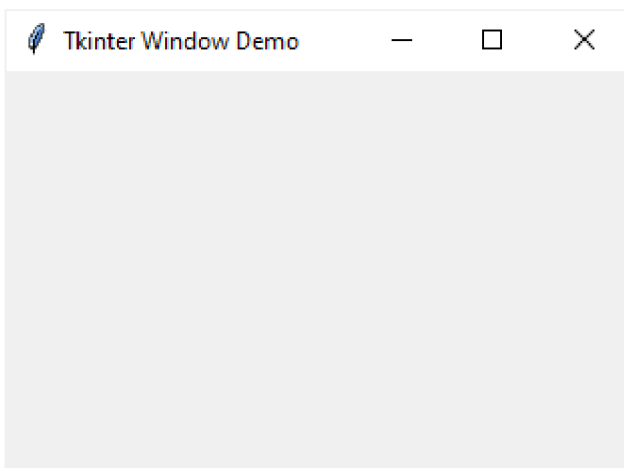
For example, the following changes the title of the root window to `'Tkinter Window Demo'` :

```
import tkinter as tk

root = tk.Tk()
root.title('Tkinter Window Demo')

root.mainloop()
```

Output:



To get the current title of a window, you use the `title()` method with no argument:

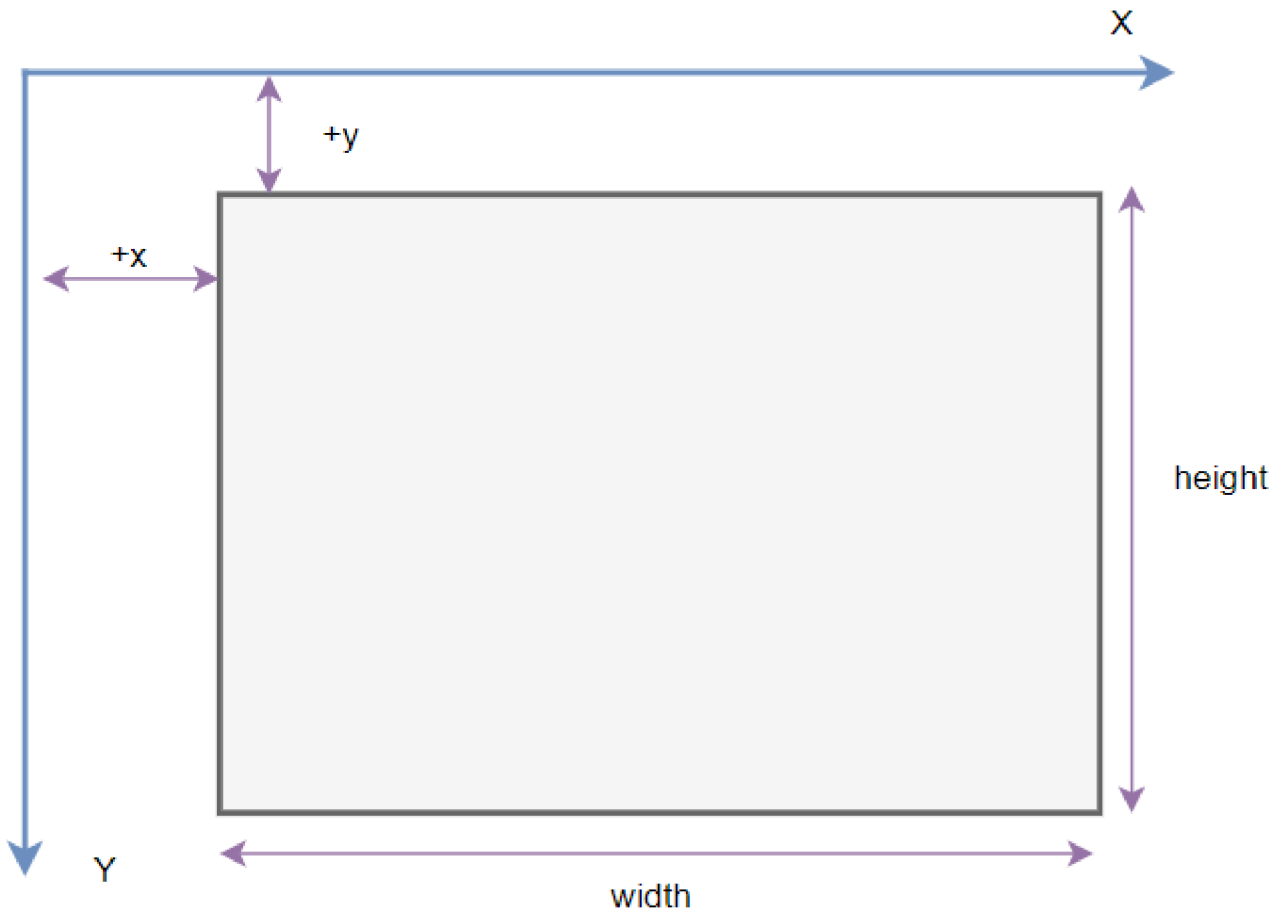
```
title = window.title()
```

Changing window size and location

In Tkinter, the position and size of a window on the screen is determined by *geometry*.

The following shows the geometry specification:

`widthxheight±x±y`



`window.geometry('widthxheight±x±y')`

In this specification:

- The **width** is the window's width in pixels.
- The **height** is the window's height in pixels.
- The **x** is the window's horizontal position. For example, **+50** means the left edge of the window should be 50 pixels from the left edge of the screen. And **-50** means the right edge of the window should be 50 pixels from the right edge of the screen.
- The **y** is the window's vertical position. For example, **+50** means the top edge of the window should be 50 pixels below the top of the screen. And **-50** means the bottom edge of the window should be 50 pixels above the bottom of the screen.

To change the size and position of a window, you use the **geometry()** method:

```
window.geometry(new_geometry)
```

The following example changes the size of the window to **600x400** and the position of the window to 50 pixels from the top and left of the screen:

```
import tkinter as tk

root = tk.Tk()
root.title('Tkinter Window Demo')
root.geometry('600x400+50+50')

root.mainloop()
```

Sometimes, you may want to center the window on the screen. The following program illustrates how to do it:

```
import tkinter as tk

root = tk.Tk()
root.title('Tkinter Window - Center')

window_width = 300
window_height = 200

# get the screen dimension
screen_width = root.winfo_screenwidth()
screen_height = root.winfo_screenheight()

# find the center point
center_x = int(screen_width/2 - window_width / 2)
center_y = int(screen_height/2 - window_height / 2)

# set the position of the window to the center of the screen
```

```
root.geometry(f'{window_width}x{window_height}+{center_x}+{center_y}')

root.mainloop()
```

How it works.

- First, get the screen width and height using the `winfo_screenwidth()` and `winfo_screenheight()` methods.
- Second, calculate the center coordinate based on the screen and window width and height.
- Finally, set the geometry for the root window using the `geometry()` method.

If you want to get the current geometry of a window, you can use the `geometry()` method without providing any argument:

```
window.geometry()
```

Resizing behavior

By default, you can resize the width and height of a window. To prevent the window from resizing, you can use the `resizable()` method:

```
window.resizable(width,height)
```

The `resizable()` method has two parameters that specify whether the width and height of the window can be resizable.

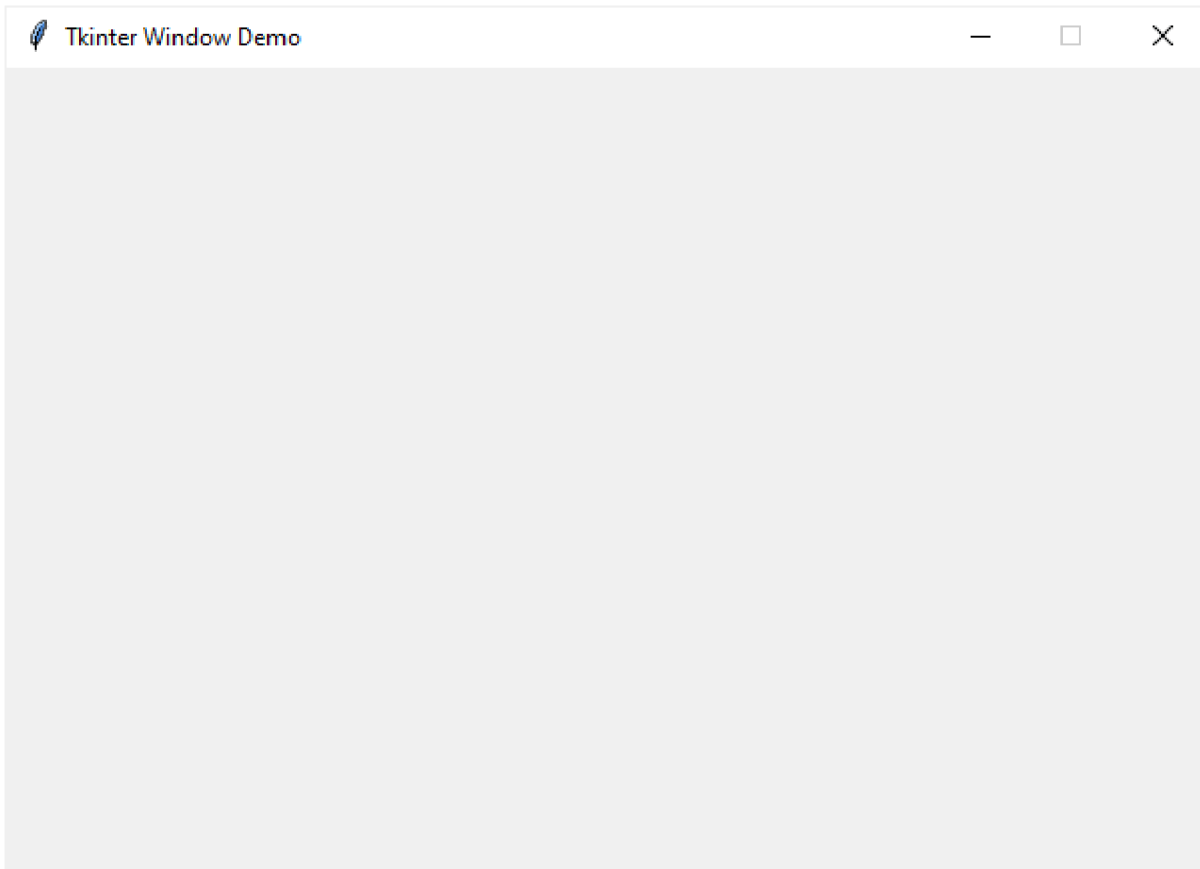
The following shows how to make the window with a fixed size:

```
import tkinter as tk

root = tk.Tk()
root.title('Tkinter Window Demo')
root.geometry('600x400+50+50')
root.resizable(False, False)
```

```
root.mainloop()
```

Output:



When a window is resizable, you can specify the minimum and maximum sizes using the `minsize()` and `maxsize()` methods:

```
window.minsize(min_width, min_height)
window.maxsize(min_height, max_height)
```

Transparency

Tkinter allows you to specify the transparency of a window by setting its alpha channel ranging from 0.0 (fully transparent) to 1.0 (fully opaque):

```
window.attributes('-alpha',0.5)
```

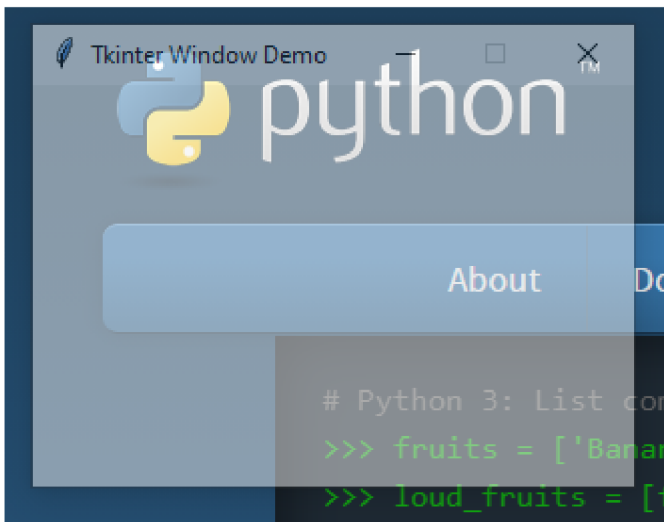
The following example illustrates a window with 50% transparent:

```
import tkinter as tk

root = tk.Tk()
root.title('Tkinter Window Demo')
root.geometry('600x400+50+50')
root.resizable(False, False)
root.attributes('-alpha', 0.5)

root.mainloop()
```

Output:



Window stacking order

The window stack order refers to the order of windows placed on the screen from bottom to top. The closer window is on the top of the stack and it overlaps the one lower.

To ensure that a window is always at the top of the stacking order, you can use the `-topmost` attribute like this:

```
window.attributes('-topmost', 1)
```

To move a window up or down of the stack, you can use the `lift()` and `lower()` methods:

```
window.lift()
window.lift(another_window)
```

```
window.lower()
window.lower(another_window)
```

The following example places the root window on top of all other windows. In other words, the root window is always on top:

```
import tkinter as tk

root = tk.Tk()
root.title('Tkinter Window Demo')
root.geometry('300x200+50+50')
root.resizable(0, 0)
root.attributes('-topmost', 1)

root.mainloop()
```

Changing the default icon

Tkinter window displays a default icon. To change this default icon, you follow these steps:

- Prepare an image in the `.ico` format. If you have the image in other formats like `png` or `jpg`, you can convert it to the `.ico` format. There are many online tools that allow you to do it quite easily.
- Place the icon in a folder that can be accessible from the program.
- Call the `iconbitmap()` method of the window object.

The following program illustrates how to change the default icon to a new one:

```
import tkinter as tk

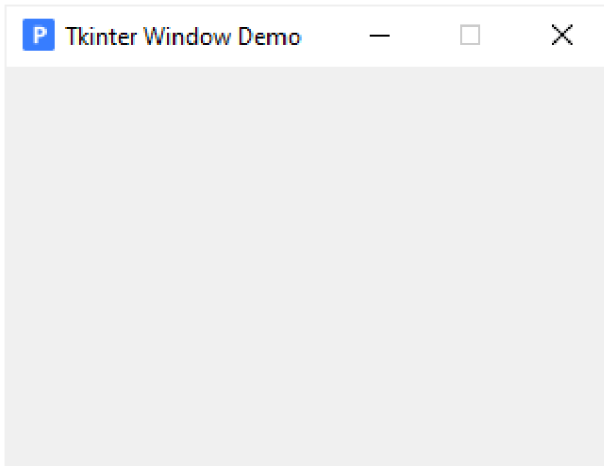
root = tk.Tk()
root.title('Tkinter Window Demo')
root.geometry('300x200+50+50')
```



```
root.resizable(False, False)
root.iconbitmap('./assets/pythontutorial.ico')

root.mainloop()
```

Output:



If you want to use the above icon, you can download it to your computer:



Summary

- Use the `title()` method to change the title of the window.
- Use the `geometry()` method to change the size and location of the window.
- Use the `resizable()` method to specify whether a window can be resizable horizontally or vertically.
- Use the `window.attributes('-alpha', 0.5)` to set the transparency for the window.
- Use the `window.attributes('-topmost', 1)` to make the window always on top.
- Use `lift()` and `lower()` methods to move the window up and down of the window stacking order.
- Use the `iconbitmap()` method to change the default icon of the window.