Python 3 - Tkinter Canvas

The Canvas is a rectangular area intended for drawing pictures or other complex layouts. You can place graphics, text, widgets or frames on a Canvas.

Syntax

Here is the simple syntax to create this widget −

```
w = Canvas ( master, option = value, ... )
```

Parameters

- **master** − This represents the parent window.
- **options** − Here is the list of most commonly used options for this widget. These options can be used as key-value pairs separated by commas.

Sr.No.	Option & Description
1	**bd** Border width in pixels. Default is 2.
2	**bg** Normal background color.
3	**confine** If true (the default), the canvas cannot be scrolled outside of the scrollregion.
4	**cursor** Cursor used in the canvas like arrow, circle, dot etc.
5	**height** Size of the canvas in the Y dimension.
6	**highlightcolor** Color shown in the focus highlight.
7	**relief** Relief specifies the type of the border. Some of the values are SUNKEN, RAISED, GROOVE, and RIDGE.
8	**scrollregion** A tuple (w, n, e, s) that defines over how large an area the canvas can be scrolled, where w is the left side, n the top, e the right side, and s the bottom.
9	**width** Size of the canvas in the X dimension.
10	**xscrollincrement** If you set this option to some positive dimension, the canvas can be positioned only on multiples of that distance, and the value will be used for scrolling by scrolling units, such as when the user clicks on the arrows at the ends of a scrollbar.

11	**xscrollcommand** If the canvas is scrollable, this attribute should be the .set() method of the horizontal scrollbar.
12	**yscrollincrement** Works like xscrollincrement, but governs vertical movement.
13	**yscrollcommand** If the canvas is scrollable, this attribute should be the .set() method of the vertical scrollbar.

The Canvas widget can support the following standard items −

arc . Creates an arc item, which can be a chord, a pieslice or a simple arc.

```
coord = 10, 50, 240, 210
arc = canvas.create_arc(coord, start = 0, extent = 150, fill = "blue")
```

image . Creates an image item, which can be an instance of either the BitmapImage or the PhotoImage classes.

```
filename = PhotoImage(file = "sunshine.gif")
image = canvas.create_image(50, 50, anchor = NE, image = filename)
```

line . Creates a line item.

```
line = canvas.create_line(x0, y0, x1, y1, ..., xn, yn, options)
```

oval . Creates a circle or an ellipse at the given coordinates. It takes two pairs of coordinates; the top left and bottom right corners of the bounding rectangle for the oval.

```
oval = canvas.create_oval(x0, y0, x1, y1, options)
```

polygon . Creates a polygon item that must have at least three vertices.

```
oval = canvas.create_polygon(x0, y0, x1, y1,...xn, yn, options)
```

Example

Try the following example yourself −

```
# !/usr/bin/python3
from tkinter import *

from tkinter import messagebox
```

```
top = Tk()

C = Canvas(top, bg = "blue", height = 250, width = 300)

coord = 10, 50, 240, 210
arc = C.create_arc(coord, start = 0, extent = 150, fill = "red")
line = C.create_line(10,10,200,200,fill = 'white')
C.pack()
top.mainloop()
```

Result

When the above code is executed, it produces the following result −