



Getting Started with ti_draw

Circles

Computer graphics is the most fun of all programming projects. Here's an introduction the **ti_draw** module by making a collection of random colorful circles on the screen.

- 0. The **<Add On> ti_draw** module contains functions that let you produce colorful graphics like this on the screen.

*Note: Your calculator needs the additional **ti_draw.8xv** AppVar which is available in the OS and Apps bundle version 5.7 for the handheld. If you are using SmartView, you will have to transfer the module to SmartView and then you have to hand-type the module name(s) into import statements.*

- 1. In a new program add the three import statements:

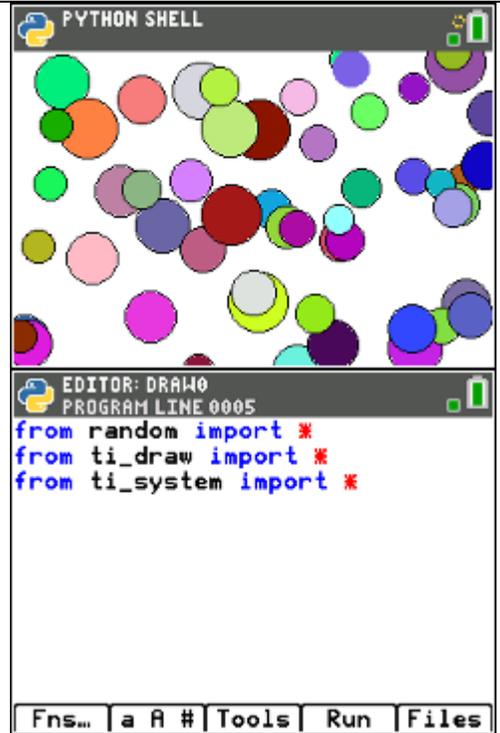
```
from random import *
from ti_draw import *
from ti_system import *
```

To get **ti_draw**, select **<Fns> Modul <Add Ons>**. **ti_draw** appears along with all other 'add-on' modules you may have installed.

- 2. When the statement **from ti_draw import *** is in the Editor, a new item, **ti_draw...**, appears on the **Modul** menu.

Use the **[math]** key as a shortcut to the **Modul** menu as seen here (but the **<Add Ons>** soft key does not appear).

This **ti_draw...** menu is where you find all the **Shape** drawing functions and **Control** tools on two sub-menus: **Shape** and **Control**.





3. The **Shape** menu:

Each of the Drawing functions on the **Shape** menu requires a certain number of arguments depending on the shape. The menu shows **blue** placeholders for the arguments, but they are not included in the function when inserted in the program.

```

EDITOR: DRAW0
Shape Control Ver
1:draw_line(x1,y1,x2,y2)
2:draw_rect(x,y,w,h)
3:fill_rect(x,y,w,h)
4:draw_circle(x,y,r)
5:fill_circle(x,y,r)
6:draw_text(x,y,"string")
7:draw_poly(x-list,y-list)
8:fill_poly(x-list,y-list)
9:plot_xy(x,y,sh)      sh=1-13
Esc

```

4. The **Control** menu contains functions that affect the screen and drawing state (**clear** the screen or a region of the screen, color, pen size, and window coordinates).

The function **show_draw()** is used at the end of a program (or to pause a program) to keep the image on the screen until the **[clear]** key is pressed.

```

EDITOR: DRAW0
Shape Control Ver
1:clear()
2:clear_rect(x,y,w,h)
3:set_color(r,g,b)      0-255
4:set_pen("size","style")
5:set_window(xmn,xmx,ymn,ymx)
6:show_draw()          [clear]
Esc

```

5. Select **clear()** from the **Control** menu. This will clear the Shell screen to prepare it for drawing shapes on a blank 'canvas'.

Get **while not escape()**: from **[math] ti_system....**

```

EDITOR: DRAW0
PROGRAM LINE 0009
from random import *
from ti_draw import *
from ti_system import *

clear()
while not escape():
Fns... a A # Tools Run Files

```

6. To make random circles you need a bunch of random values for the positions, radii and colors of the circles. Use **randint(,)** (from **[math] random...**) to assign *appropriate* values to the variables.

The screen is 318x212 pixels and the color values must be integers in the domain [0..255]. The pixel (0, 0) is in the upper left corner of the screen. It's OK to plot graphics that are off-screen, too).

x and **y** are the coordinates of the center of the circle.

r is the radius of the circle.

a, **b**, and **c** are the three color values, red, green and blue, required for **set_color(, ,)** coming up next...

```

EDITOR: DRAW0
PROGRAM LINE 0014
while not escape():
  x=randint(0,318)
  y=randint(0,212)
  r=randint(10,20)
  a=randint(0,255)
  b=randint(0,255)
  c=randint(0,255)
Fns... a A # Tools Run Files

```



10 MOC: Python Modules

TI-84 PLUS CE PYTHON

GETTING STARTED WITH TI_DRAW: CIRCLES

- Get `set_color(, ,)` from the [math] ti_draw... Control menu and use the variables **a**, **b**, and **c** for the three arguments controlling the red, green and blue channels of the pixels.

Note that you get the commas inside the parentheses, but not the blue argument placeholders seen in the menu.

```
EDITOR: DRAW0
PROGRAM LINE 0014
clear()
while not escape():
  x=randint(0,318)
  y=randint(0,212)
  r=randint(10,20)
  a=randint(0,255)
  b=randint(0,255)
  c=randint(0,255)
  set_color(a,b,c)
```

- Get `fill_circle()` from [math] ti_draw... Shape menu.

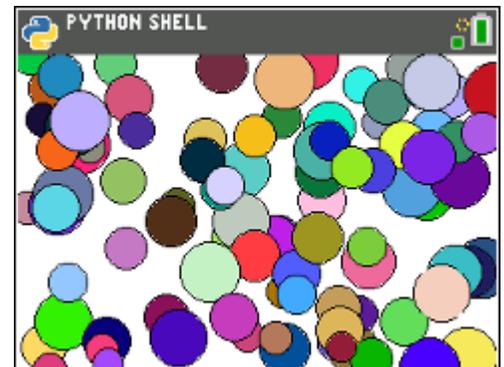
(**x**, **y**) is the center of the circle and **r** is the radius of the circle.

```
EDITOR: DRAW0
PROGRAM LINE 0015
clear()
while not escape():
  x=randint(0,318)
  y=randint(0,212)
  r=randint(10,20)
  a=randint(0,255)
  b=randint(0,255)
  c=randint(0,255)
  set_color(a,b,c)
  fill_circle(x,y,r)
```

- <Run> the program. Press [clear] to stop the program and add one more feature to the project...



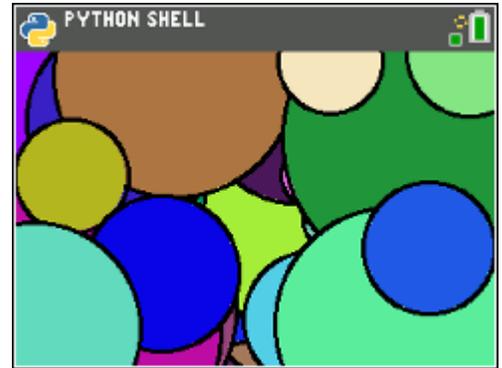
- Add a **black** border to each circle to help them stand out. Use `set_color(0, 0, 0)` and `draw_circle(x, y, r)` at the bottom of the loop (but still inside the loop).





11. `set_pen("size", "style")` has two arguments that you choose from special menus that determine the appearance of the drawing pen. This function is placed before the `draw_circle(, ,)` statement like the `set_color(, ,)` function and sets the pen state until another `set_pen(,)` command is issued.

In this image we have chosen “medium” for the size and “solid” for the pen style. We also changed the random radii of the circles.



12. Here's most of the final code.

Add `show_draw()` at the end of the program to keep the graphics on the screen until you press the [clear] key again.

The numbers used in the program are arbitrary. Feel free to change them to see the effects. Try some other **Shapes** found on the `ti_draw...` menu.

```
EDITOR: DRAW0
PROGRAM LINE 0008
y=randint(0,212)
r=randint(10,100)
a=randint(0,255)
b=randint(0,255)
c=randint(0,255)
set_color(a,b,c)
fill_circle(x,y,r)
set_pen("medium","solid")
set_color(0,0,0)
draw_circle(x,y,r)
show_draw()
Fns... | a A # | Tools | Run | Files
```