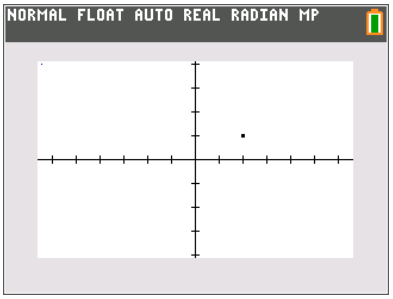
Unit 5: Graphics | **Skill Builder 2: Points and Pixels**

In this second lesson for Unit 5 you will learn about plotting points and the difference between the Pt-On and Pxl-On commands.

Objectives:

- Use the point and pixel plotting statements.
- Develop formulas to utilize graphics in programs.

Point Command



See the [DRAW] POINTS menu.

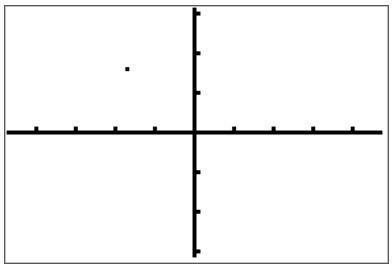
Pt-On(x,y) plots a point according to the graph WINDOW settings. It means 'turn the point on'. **Pt-On(2,1)** plots the point in the first quadrant in the screen to the right. **Pt-On(x,y,style,color)** has optional arguments: style (1 to 4) and color. See the syntax help by pressing [+] while on the command in the [DRAW] menu. **Pt-On(100,100)** will plot a point even if it's out of the current viewing window.

Pixel Command

Pxl-On(uses the screen's pixels and ignores the WINDOW settings. **Pxl-On(2,3)** plots the tiny pixel in the upper left corner of the same screen at row 2, column 3. It is hard to see! The coordinates are not in the usual (x,y) order: they're 'backwards' because they refer to (row#, column#). **Pxl-On(x,y,color)** has only an optional color argument. There are also corresponding **–Off(,** –**Change(, and -Test** commands that we'll not deal with here.

Pixels

Depending on your TI 84 Plus family calculator, your screen has a set number of pixel columns and rows: TI-84 Plus: 96 columns x 64 rows and TI-84 Plus C/CE: 265 columns x 165 rows. Rows are horizontal (across) and columns are vertical (up and down). Split screen settings affect the number of rows and/or columns depending on the setting. The **TI-84 Plus** graph screen does not use rightmost column or bottom row for *graphing* so that there are an odd number of points on the graph area. This ensures that there is a 'center' (origin) point. The upper left pixel is (0,0). **Pxl-On(0,0)** turns on the pixel at column 0, row 0. The rows and columns are numbered starting with zero, not one.



*This **TI-84 Plus** graph screen has larger pixels so they are easier to see. This screen shows the result of **Pxl-On(15,30)**. That's row 15, column 30 of the screen. The bottom right pixel is (63,95).*

Teacher Tip: Pixel-related statements can be confusing because the order of the coordinates is reversed: the *y*-value (row number) is first and the *x*-value (column number) is second. Also, the *y*-direction is upside down: row 0 is at the top and row 165 (or 63 on a TI-84 Plus) is at the bottom of the screen. This is similar to the orientation of the HOME screen grid when using the **Output**(statement (line#, column#, with line 1 at the top). The **Text**(statement that draws text on the graph screen (in a later lesson) also uses pixels for positioning the text and not points.

The **–Test** statements are used as conditions in **If** statements or loops to see if a point or pixel is 'on'. Pt-Test(0,0) will be true when the point (0,0) is on. Note that a point can be 'on' and white so it does not appear on the screen!

On a TI-84 Plus pixels and points are the same size. On the TI-84 Plus C/CE points are

larger than pixels.

Programming with Points

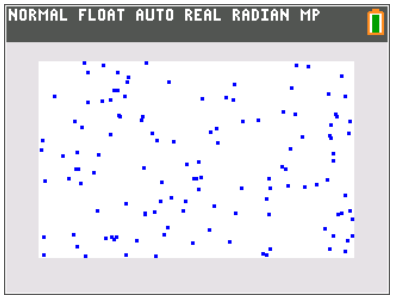
Let's write a program that randomly fills the GRAPH screen with POINTS. This program will have an infinite loop so press [ON] to 'break' out of the program.

We'll need an ***algorithm*** (formula) to get a random point on the current GRAPH screen within the window ranges:
rand is found on the [MATH] PROB menu and generates a random number between 0 and 1. **rand*(Xmax-Xmin)** generates a random number between 0 and Xmax-Xmin so we'll add **Xmin**.
rand*(Xmax-Xmin)+Xmin generates a random number between **Xmin** and **Xmax.**
…and we write a similar formula for the Y-coordinate.
And this is why **mathematics** is so important to programming!

Note:
AxesOff *is on the [FORMAT] screen*.
FnOff *is on the [VARS] Y-VARS On/Off… menu*.
PlotsOff *is on the [STAT PLOT] menu.*
ClDraw *is on the [DRAW] menu*.
While 1 *creates an infinite loop since True is represented as* ***1*** *in the calculator.*
The random values are stored in **A** *and* **B** *for plotting*.
Remember to press [ON] to 'break' the program.
This program works the same on any TI-84 Plus Family device!

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:FILLPTS
:AxesOff
:FnOff
:PlotsOff
:ClrDraw
:While 1
:rand*(Xmax-Xmin)+Xmin→A
:rand*(Ymax-Ymin)+Ymin→B
:Pt-On(A,B)
:End
```

This program works on all TI-84s.



Enhancing prgmFILLPTS with Color

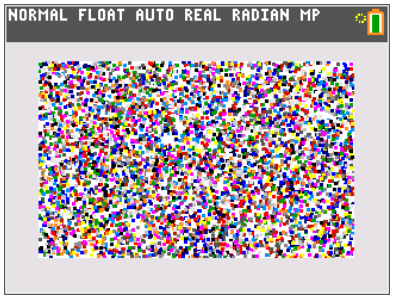
On the **TI-84 C/CE** you can add random COLOR to the **Pt-On(** statement. The lowest color number is 10 and the highest is 24.

Write a statement that generates a random integer from 10 to 24 using **randInt( )** and add that as the *third* argument to the **Pt-On(** statement.

Add the following statement before **Pt-On(A,B).** Change **Pt-On(A,B)** to **Pt-On(A,B,C).**

<your color generator> → **C**

Note: There are two optional arguments to **Pt-On***: style and color. Style can be from 1 to 4 and color can be from 10 to 24. So if the third argument is a value from 1 to 9 then it's a style. If it's between 10 and 24 it's a color. Other values cause an error.*



Multi-colored points

Answer: randint(10,24) →C