



In this third lesson for Unit 5 you will learn about drawing lines, text and color enhancements.

Objectives:

- Use the line, function, and text drawing statements.
- Use colors in graphics statements.
- Develop formulas to utilize graphics in programs.

Drawing Lines and Curves

Line(X,Y,W,Z) draws a *segment* between points (X,Y) and (W,Z). See CATALOG HELP for the optional features.

Vertical A draws the vertical line $X=A$.

Horizontal B draws the horizontal line $Y=B$.

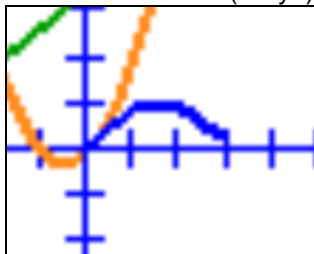
DrawF X^2+X draws the function. This is different than graphing the function.

See the examples to the right. Note the optional colors found on the **PRGM** COLOR menu. Color is not available on the TI-84 Plus.

Tip: to draw part of a function divide the function by the interval desired:

DrawF $\sin(X)/(X \geq 0 \text{ and } X \leq \pi)$ ■

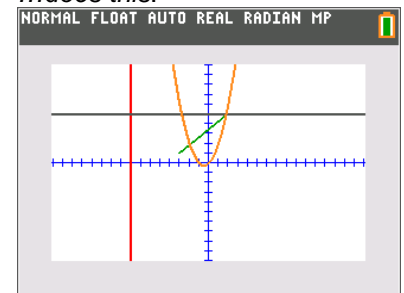
draws the blue curve seen here (Why?):



```
NORMAL FLOAT AUTO REAL Radian MP
PROGRAM:USS0
:FnOfff
:PlotsOfff
:ClrDraw
:ZStandard
:ZSquare
:Line(2.5,-3,1,GREEN)
:Vertical -8,RED
:Horizontal 5,DARKGRAY
:DrawF X^2+X,ORANGE
```

This program...

...does this:



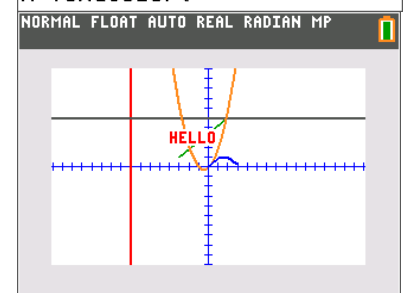
Text Drawing

The **Text(** drawing statement is unique because it uses *pixel* values rather than window (point) values for positioning the text. There is also a separate **TextColor(** statement that sets the color of the next text drawn.

Text(50,100,"HELLO") displays HELLO in the same spot on the screen regardless of the WINDOW settings. Row 50, column 100 of the pixels represents the *upper left corner* of the text to be drawn.

Note: Remember your screen's pixel dimensions: TI-84 Plus: 96 columns x 64 rows and TI-84 Plus C/CE: 265 columns x 165 rows.

```
NORMAL FLOAT AUTO REAL Radian MP
DRAW POINTS STO BACKGROUND
3↑Horizontal
4:Vertical
5:Tangent(
6:DrawF
7:Shade(
8:DrawInv
9:Circle(
0:Text(
A:TextColor(
```





10 Minutes of Code

TI-84 PLUS FAMILY

Programming with Line(and Algebra

This programming activity 'enhances' the **Line(** statement.

The **Line(** statement only draws a *segment* between two points. We'd like to see a *line through* the two points and extend all the way across the screen regardless of which two points are selected. This activity makes use of Algebra concepts, so be prepared!

1. Start a new program. We will call it **LINE**.
2. Add the usual graph setup statements to the beginning of the program.
3. Use two **Input** statements *without variables* to get the coordinates of two points on the screen. Input determines the values of **X** and **Y** so we need to store the first two values in other variables, **A** and **B**, so that we can get the second set of coordinates into **X** and **Y**.
4. Calculate and store the *slope* of the line.
5. Now we need the two points at the left and right sides of the screen for the line statement. The x-coordinates of these points are **Xmin** and **Xmax**.
6. We need to compute the y-coordinates.
7. The equation of the line is $y = M*(x - A)+B$ (*point-slope form*).

Your Task...

1. Substitute **Xmin** and **Xmax** (the *names* not the values!) into the equation for x and store the results in the two variables **Q** and **R** representing the y-coordinates.
2. Use the **Line(** statement to draw a line between the left and right side of the screen.

Extensions

1. Add a loop in this program to allow you to draw many lines without having to re-run the program which erases the screen.
2. This program fails when the line is vertical. Why? Incorporate an **If...** structure to handle this special case.

Programming Points to Pixels

Imagine this: you use the **Pt-On(** statement to draw a point (**A,B**) on the graph screen. You now would like to *label* the point with some text. Where do you draw the text?

Write two formulas (one for **C** and one for **D**) that convert WINDOW coordinates to pixel coordinates for the **Text(** statement. This table (TI-84 C/CE values) may help:

| <u>WINDOW</u> | <u>pixel</u> |
|---------------|--------------|
| Xmin | 0 |
| Xmax | 264 |
| A | ? |
| Ymax | 0 |
| Ymin | 164 |
| B | ? |

*Note: remember that in the **Text(** command the first argument is the ROW number which corresponds to the y-coordinate or the point!.*

UNIT 5: SKILL BUILDER 3

STUDENT ACTIVITY

```

NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:LINE
:FnOfff
:PlotsOff
:ZDecimal
:AxesOn
:ClrDraw
:Input
:X→A
:Y→B
:Input

```

: (Y-B)/(X-A)→M

:Line(Xmin,Q,Xmax,R)

Complete the ? so that the point is labelled P:

```

NORMAL FLOAT AUTO a+bi RADIAN MP
PROGRAM:POINT
: ?→C
: ?→D
:Pt-On(A,B)
:Text(C,D,"P")

```