Unit 1: Getting Started With Python

Skill Builder : Introducing the Python App

In this lesson, you will be introduced to the Python programming system on the **TI-84 Plus CE Python** edition graphing calculator. You will take a tour of the menus, explore the changes to the keypad, and work with a program to see some of the key features of using Python.

Objectives:

- Python overview
- Use Editor and Shell
- Run a program
- Edit a program
- Entering expressions in the Shell

1. Welcome to the Python experience on your **TI-84 Plus CE Python graphing calculator**. These lessons require no prior knowledge of programming and will help you get started with coding in Python in a friendly, easy-to-use platform in the palm of your hand. Get ready to tap the power of programming in Python.

Teacher Tip: The **Python App** is a complete programming system that is separate from TI-Basic programming. There is an Editor and a Shell that work together to give a complete Python programming experience similar to computer text-based programming platforms. The toolbar at the bottom of the screen is the main menu for all things Python. The keypad has many Python-dedicated keys, including lowercase characters.

Python files are stored as **AppVars** in the calculator's memory. Using the TI Connect™ CE software application, you can transfer these **AppVars** to your computer as *.8xv files *or* as a *.py files which can be edited in your favorite Python IDE on your computer.

There is an online eGuide available.

2. Let's start learning about the Python environment. Press **[prgm]** and select **Python App** from the menu.

Teacher Tip: Python is a separate app in the calculator. If it is deleted, then this Python system will not work.

[mem] > Memory Management > Apps screen

3. The Python App is now active. You are looking at the **FILE MANAGER** showing the Python files the that are stored in RAM in your calculator. As you add programs, their filenames will be added to this short list alphabetically.

   On the bottom of the screen is a custom menu bar ("soft keys"). Access these menus by pressing the corresponding **top-row key** on your keypad. In these lessons you will see

   … *select* **<menu_name>** …

   and not the corresponding key name. For example:

   > To run the HELLO program, *press* the **[down arrow]** key to point to the word HELLO and *select* **<Run>** or *press* **[enter]**.

   Select **<Run>** by pressing **[y=]**.

   Note that the on-screen menu item <Softkey> is preceded by "select…" and a keypad key is preceded by "press…". This distinction is used throughout all these lessons and the corresponding top-row key will not be stated in the text. Also note that all the on-screen menus begin with a capital letter (but not menu items), and keypad keys are all lowercase.

Teacher Tip: As the list of files grows, it can get cumbersome, and you could even run out of room (limit is 100 files or 50 KB RAM).

Memory limit screen

To manage memory, use the **[mem]** tool on the calculator to *archive* the Python files rather than delete them. The Python files are **AppVars** and are indicated with a "**PY**" notation on the right side of the screen.

Another memory constraint: Lists for transferring between the Python environment and the TI-84 Plus CE environment are limited to 100 elements max.

4. Running the HELLO program brings you to the **Python Shell**, the place where Python programs are run. The Shell prompt is **>>>.** The program displays a **TI**p and the question "What is your name?" You must type your name and press **[enter]**.

   Note: **TI***p is no accident. After all, this is* **T***exas* **I***nstruments!*

5. To type letters, press **[2nd] [alpha]** to turn on lowercase alpha-lock, then type the letters. If you want capital letters, press the **[alpha]** key again.

   Tip: *The* **[alpha]** *key toggles between lowercase and uppercase modes. See the current state at the top of the screen. In this image, lowercase alpha-lock is active.*

6. At the end of your name, press **[enter]** to see what the program produces ("Hello *name*" and another **TI**p).

Press **[2nd] [alpha]** to turn alpha-lock off.

When you see **>>> |** the program has ended. The symbol **>>>** is the Shell prompt and the vertical bar (**|**) is the cursor.

At the Shell prompt **>>>** you are free to type some Python expressions. The Shell is similar to the calculator HOME screen, but it only understands Python expressions.

Try some calculations (press **[enter]** on each line to see the effect):

```
>>>2 + 3
>>>5 * 5
>>>6 * (4 - 2)
>>>3 ** 5        (raise to a power: 3^5; use [^] for **)
>>>x = 13        (assign 13 to x; use the [sto] key for =)
>>>2 * x + 4
```

After each expression except **x=13** you will see the result displayed.

7. The keypad is very different when working in Python. The top (graphing) row is used for accessing the Python menus on the bottom of the screen. Python supports both lowercase and uppercase characters. Many keys are now "python-friendly" (like **[^]** and **[sto]**). The cursor is a vertical bar in the Shell and a blinking underscore (insert mode only) in the Editor. Pressing **[del]** performs a *backspace* operation, deleting the character to the left of the cursor. Other keys take on new meanings as you will see.

   In the next step you will have a look at the Python **[Editor]** containing the HELLO program.

   Note that the **[X,T,θ,n]** *key now produces the either lowercase* x *or uppercase* X *depending on the alpha mode.*

8. Select **<Editor>** (press the **[trace]** key). You are now using the Python Editor which is where programs are written and edited. The menu at the bottom of the screen is slightly different: **[Run]** replaces **[Editor].**
   The Editor is color-enhanced.
   The program contains **print( )** statements and one **input( )** function:

   n = input ("What is your name?")

   The top line is a comment: **# Version 2 EN**. It begins with the **#** sign. Comments are useful to add notes to your code that the Python interpreter ignores when the program is run.

   The **"green"** stuff is text in quotation marks.
   The red **\n** is a special "newline" character.

   Note: Feel free to edit the program but be aware that all changes you make are automatically saved. You will write your own code in the next lesson.

9. Explore the other menus at the bottom of the screen, using the top row of keys on the calculator.

   [Fns…] contains seven sub-menus across the top of the screen. These sub-menus contain all the Python programming commands.

   The **Func** menu shown has just two statements but the others have many more.

   In the lower left corner is the **<Esc>** option which takes you back to the Editor or Shell when you press the **[y=]** key.

   We will explore the other sub-menus as we begin to write our own programs.

   Select **<Esc>** at any time to return to the Editor.

Teacher Tip: Here are the other six sub-menus.

10. Did you press the **[y=]** key? *Congratulations!* From now on we will use the language Select ***<menu option>*** to refer to the menu bar at the bottom of the screen and you will press the corresponding top-row key.

    Select **<Files>**.

11. The **FILE MANAGER** displays the Python files that are stored in RAM on your calculator. The menu bar displays:
    - **<Run>** — run the program you are pointing at
    - **<Edit>** — load the program into the Editor
    - **<New>** — create a new Python file
    - **<Shell>** — bring up an empty Shell
    - **<Manage>** — manage your files (Copy, Delete, Rename)

The calculator comes with two other demo programs: GRAPH and LINREGR. Feel free to run them at any time but be careful about editing them.

12. To **quit** using Python and return to your calculator's home screen, press **[2nd] [mode]** ([quit]) or, in the FILE MANAGER, select **<Manage>** and choose Quit Python.

    You will see this "**Are you Sure?"** screen. Select **<Ok>** or press **[enter]** to quit.

    Note: Any file that you are working on is automatically saved as you type. There is no need to save your Python file.

    Tip: When done with Python, do not leave the **Editor** *on the screen as it immediately responds to key presses and can mess up the program that is on the screen. You can leave it on the* **Shell** *or* **FILE MANAGER** *or you can quit the Python App completely.*

13. If you have written programs using TI-Basic, these are some of the **BIG** differences between TI-Basic and Python:
    - Lowercase is the norm in Python and Python *is* case-sensitive
    - Character-by-character typing *is* allowed; rather than selecting **print()** from a menu, you can simply type it in
    - Python relies heavily on proper *indentation* to determine code blocks like **loops** and **if…** blocks rather than keywords or symbols; the Editor has special indentation symbols (gray diamond characters) to help manage indentation

    In the next lesson, you will write your own program.

Teacher Tip: Python coding works like TI-Basic — the filename is entered first and all editing is saved "on the fly." On the Editor **<Tools>** menu there are handy editing options that are discussed in subsequent lessons. The **FILE MANAGER** (the place where a Python session starts, or select **<Files>**) supports deleting, renaming and replicating Python files and is the place where all **<New>** Python files are created.

Find a complete keypad description here.