TI-Nspire™ Software Script

Creating a program (Tutorial E4)

http://www.atomiclearning.com/k12/en/movie/35635/play_window?type=Tutorial&sid=1674

[00:00:00.00]
You can write your own functions and programs using the Program Editor in the Calculator application. I'll insert the Calculator

[00:00:08.00]
application by clicking Insert and choosing Calculator. The TI-Nspire™ software considers Functions and Programs very similar

[00:00:17.00]
things. One difference is that Functions can return a value that can be graphed or entered in a table, just like sin(x). In this

[00:00:26.00]
case, I want to write a simple set of instructions that will randomly generate a number, and then display the word "heads" or "tails"

[00:00:34.00]
depending on which number was generated. I'll write this as a program. I'll go up to the Application Tool bar and click on Functions

[00:00:43.00]
& Programs, then Program Editor, New. I'll call this program coins by typing "c-o-i-n-s", and I'll make sure that the Type is set

[00:00:57.00]
to Program. I'll leave the Library Access set to the default, and then click OK to move into the Program Editor. I'll enter into

[00:01:09.00]
a split screen view, with my calculator entry on the left, and my Program Editor on the right. As you can see, the program framework

[00:01:16.00]
has already been entered. I'll press the down arrow key once to move to the field below the Program statement. I'll be using

[00:01:26.00]
the variable x in my program, but I need to be careful whenever I do that, because other programs and functions might be sharing

[00:01:32.00]
the variable named “x”. This is called a global variable. I don’t
want that type of variable; I want to tell the software to make

[00:01:40.00]
a version of the variable x that only applies within my program.
This is called a local variable, and I can create that by clicking

[00:01:49.00]
Define Variables, then choosing Local. Now, I’ll type my variable
name, “x”. To start a new line, I’ll the press the Enter key on

[00:02:00.00]
the keyboard. Now, I need to tell the program to generate my random
number and store it in my local x variable. I’ll open the Catalog

[00:02:09.00]
by pressing the Catalog key, and then I’ll press “R” on the keyboard
to move to the r’s. Now I’ll scroll down until I highlight the

[00:02:19.00]
random Integer function, randInt(). Notice that in the gray bar
on the bottom of the catalog there’s a hint that shows me what

[00:02:27.00]
this function requires. According to the hint, I’ll need to supply
both lower and upper bounds, separated by a comma. I’ll press

[00:02:37.00]
Enter to copy randInt() to the program line, and then I’ll enter
my bounds. I only want it to generate a 1 or a 2. 1 will represent

[00:02:47.00]
heads, and 2 will represent tails. I’ll type “1” then a comma,
and then “2” so randInt can only return a 1 or a 2. Those are

[00:02:58.00]
the only possibilities. I’ll press the right-arrow key to move
the cursor past the close parenthesis. Once a number is generated,

[00:03:07.00]
I’ll need to store it in my local x variable so I can test it.
I’ll open the Catalog again, and this time I’ll press the “z” key

[00:03:15.00]
to move to the bottom of the alphabet. Now, I'll scroll down to
the right-arrow symbol, and press Enter. This symbol allows me

[00:03:25.00]
to store an expression or value in a variable. Then type "x". When
I run the program later, the line "randInt(1,2) -> x" will generate

[00:03:36.00]
a random value of 1 or 2, and store it in my local variable, X.
Now I'll press Enter to start a new line. To test the stored x

[00:03:49.00]
value, I'll need a programming statement called "If...Then...Else".
This is called a conditional statement, because it executes commands

[00:03:57.00]
based on conditions. In this case, we want to tell the software
If x=1, then display the word 'heads.' Otherwise, display the

[00:04:06.00]
word 'tails.'" Another template helps you enter this statement;
just click Control, and then "If...Then...Else...EndIf". After

[00:04:19.00]
the template is inserted, I'll leave the cursor right where it's
sitting, just to the right of the word "If," and I'll type "x=

[00:04:25.00]
1". Then I'll press the down arrow key to start a new line, and
I'll click I/O, and then choose Disp. Now I'll type: ("heads"). This

[00:04:44.00]
means that if x=1, the software will display the word "Heads". Now
I'll press the down-arrow key twice to move to the blank line

[00:04:52.00]
under "Else," and I'll click I/O, and then choose Disp. Now I'll
type: ("tails"). Since x can only be 1 or 2, I know that if it's

[00:05:07.00]
not 1, it must be 2, so if x doesn't equal 1, then I want my program
to display the word "tails." Now, I'll click Check Syntax & Store,

[00:05:16.00]
and then choose the Check Syntax & Store option. If I've entered the program correctly, I'll see a dialog box letting me know the

[00:05:24.00]
syntax is correct and that it's been stored. If there's a problem with syntax, the software will display an Error dialog box instead.

[00:05:32.00]
I'll click OK. To run the program, I'll click the work area on the left to make it active, and in the Calculator entry line I'll

[00:05:43.00]
right-click and choose Variables. Now I'll highlight my "coins" program, and then click to choose it. I'll enter a set of empty

[00:05:54.00]
parentheses, and then press Enter. I can run the program repeatedly by just continuing to press Enter. Notice that it randomly generates

[00:06:04.00]
either the word "heads" or "tails" and displays one of those words every time I press Enter.

[00:06:12.00]