

PROJECT OVERVIEW

Mood Rings: How do they work?

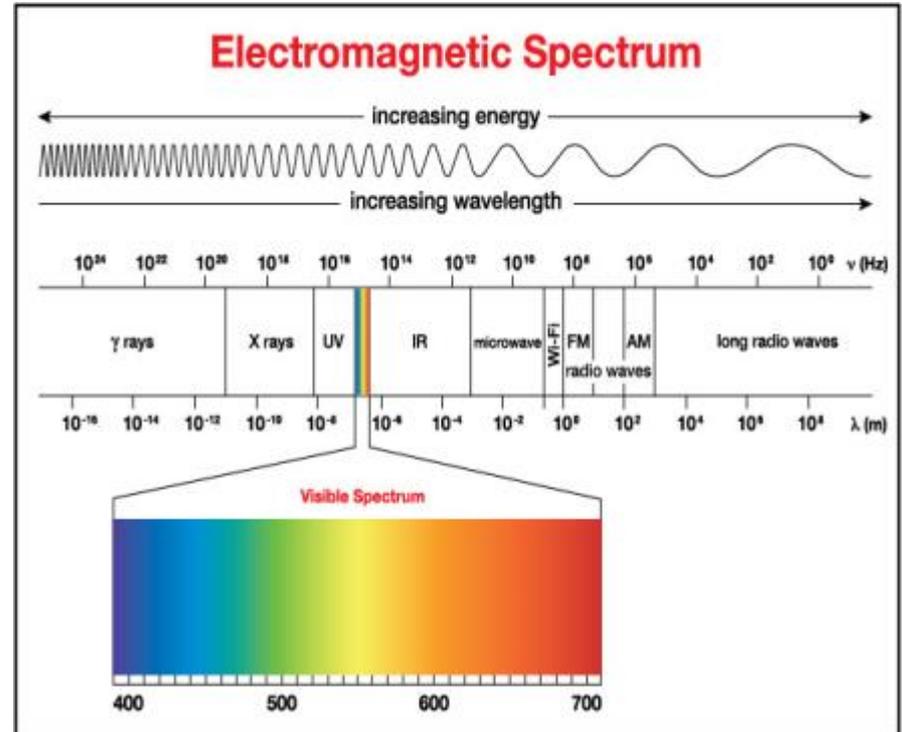
Background:

The mood ring was invented by Joshua Reynolds. Mood rings enjoyed fad popularity in the 1970s and are still around today. The stone of the ring changes color, supposedly according to the mood of the wearer. The 'stone' of a mood ring is really a hollow glass shell containing *thermotropic* liquid crystals. Modern mood rings are made from a flat plastic strip containing liquid crystals that is inserted into the hollow glass and mounted within the bezel of the mood ring.



Mood Rings: How do they work?

- The crystals respond to changes in temperature by twisting in a regular way. The twisting changes their molecular geometry, which alters the wavelengths of light that are reflected from the crystals.
- **Wavelengths** of light are another way of saying color, so when the temperature of the liquid crystals change, so does the color reflected from the stone. Thus, the ring changes color with the hand temperature of the wearer.



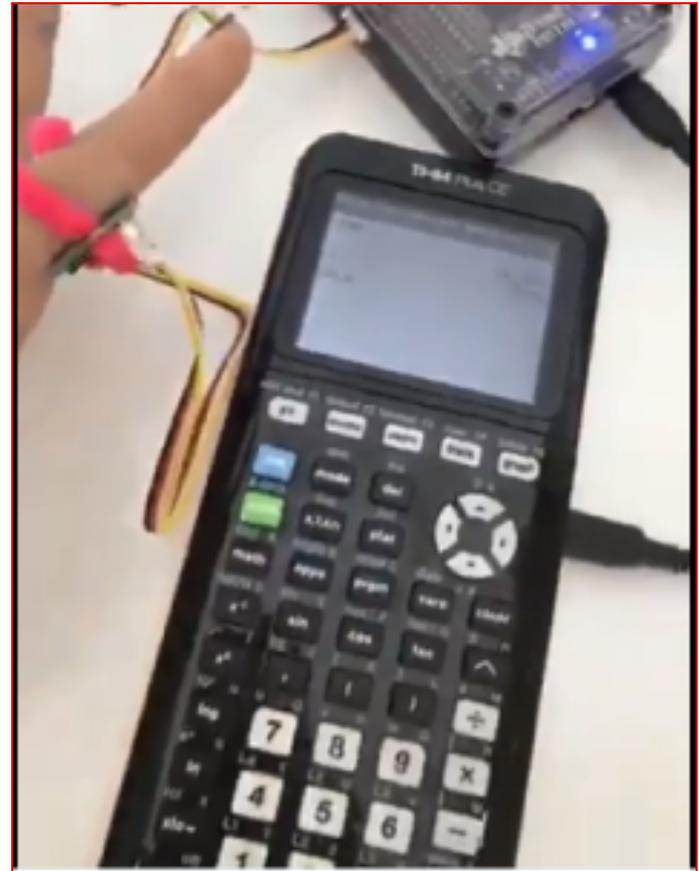
Time to DIY: Digital Mood Ring!

The Design Challenge:

Design and build a digital mood ring using the pipe cleaner that fits snugly on your finger, placement should be similar to any ring worn on the finger. Be sure the flat part of the temperature sensor is touching your skin.

In this project, you will investigate:

- temperature measurement using a temperature sensor and then build a digital mood ring that changes the color of the Hub's Red, Green, and Blue (RGB) light emitting diode (LED) as the temperature your finger changes.
- You can even display what mood you are in on your calculator!



Mood Ring Project Tasks:

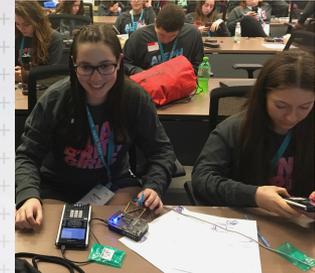
- ✓ Students are acquainted with the Mood Ring project and the graphing calculator platform. Create and execute a program to display “Hello World”.
- ✓ Students are acquainted with the TI-Innovator Hub. Create and execute a program using COLOR command to display different colors using the Red-Green-Blue LED.
- ✓ Students create a program that connects an external temperature sensor, reads values from the sensor and then displays values from the sensor. Students use the pipe cleaner to connect the TI-Innovator and temperature sensor to form a “ring.”
- ✓ Students build on the program from above to use if-then logic to control COLOR outputs based on temperature readings to create a digital mood ring.

PROJECT CONCEPTS

Digital Mood Ring (introduction to STEM and coding Projects for all levels)



Bring science and coding together (no coding experience necessary) while developing a mood ring! Explore the science of color mixing while determining the right body temperature thresholds.



This is an excellent first TI-Innovator STEM and coding project. The project includes variables, loops, conditional statements, Boolean operators and other fundamental programming concepts.



Digital Mood Ring Challenges

Challenge 1: Use SET COLOR to explore using the color LED. Try to find the RGB values of all the colors in the mood chart.

Challenge 2: Use DispAt command to display your name at several locations on the screen.

Challenge 3: Use a For..EndFor loop to display the numbers 1 through 10.

Challenge 4: Connect a temperature sensor to the TI-Innovator Hub and display the temperature on the calculator.

Challenge 5: Use a loop to read and display temperature.

Challenge 5 Extension (Optional): Use a loop to read, display and log into list arrays for time and temperature. Graph the result.

Challenge 6: Use a While..EndWhile loop along with getKey() command to monitor temperature and make a decision to display a message, “Hot” or “Cold”. Then modify your program to include a 3rd level between hot and cold, “Nice”.

Final Challenge: Build a mood ring to repeatedly read the temperature sensor, determine the mood of the person, display the temperature value and display the mood.

Summary of programming concepts in the project

- Control of a red-green-blue LED output
 - Send “SET COLOR 255 0 0”
- Display of text strings, variable values and expressions
 - DispAt 5, “Temperature=“t,”°C”
 - DispAt 4, “Time=“n*0.5,”sec”
- For loops to repeat a set of commands a specified number of times
 - For n, 1, 10
 commands within loop
 EndFor
- Connecting, reading and getting to a variable temperature sensor inputs
 - Send “CONNECT TEMPERATURE 1 TO IN 1”
 - Send “READ TEMPERATURE 1”
 - Get t
- Storing variables to variables using the assign operator, :=
 - f:=9/5 x t+32
- List variables of a set of values and accessing individual elements of the list using the position number.
 - time:={0.5, 1, 1.5, 2}
 - time[5]:=2.5
- True/False comparison operators used in conditional statements for program control: =, ≠, <, ≤, >, ≥
- Assignment of initial values to variables to give the program known starting points.
 - key:=“ “
 - time:={ }
- While loops to repeat a set of commands while a condition is true
 - While key≠“esc”
 commands within loop
 EndWhile
- Keyboard inputs using the TI-Nspire getKey command that returns a text string of the name of the last key pressed
 - key:=getKey()
- If-Then conditional statements that do a set of commands if a condition is true and are skipped if the condition is false
 - If t<22 Then
 commands within If-Then block
 EndIf
- Boolean operator *and* to require two conditional statements to be true for the entire statement to be true
 - If t≥22 and t<25 Then

Summary of TI-Nspire concepts and skills

- Home/On key to turn the calculator on and to return to the home screen, which is a jumping off point for managing the calculator.
- TI-Nspire document files combine a set of pages containing apps. Apps include calculator, program editor, grapher and others.
- Home screen document management includes
 - New document to start a new document file
 - My Documents to view and manage documents
 - Current returns to the document that is open
- Select from menus using arrow keys and enter or by the shortcut of the number or letter preceding the menu item
- The keyboard is divided into three sections:
 - Alpha entry (bottom)
 - Math entry (middle)
 - Navigation and document management (top)
- Press and release the ctrl key to access function labeled above the keys.
- esc key to back out of menus and dialogues
- ctrl-z to undo, ctrl-c to copy, ctrl-x to cut, ctrl-v to paste
- ctrl-s to save the document
- The delete key is a destructive backspace
- ctrl-doc key (+page) to add a page to a document
- Move from page to page in a document by using ctrl left arrow and ctrl right arrow or by using the touchpad to move the pointer to the page number tab and center clicking
- The menu key brings up selections that are available in each app
- Use the program editor app to create and edit programs.
- Use Check Syntax and Store to update changes made to programs before running the program.
 - ctrl-r and ctrl-b are useful shortcuts on the Check Syntax and Store menu.
 - ctrl-r, Run, checks syntax, stores and pastes the program name to a calculator page
- Programs are run on the calculator app
 - Type or paste the program name on a blank row, then press enter to run the program
- Use the var, variable, key to view and access the variables in a document
- Use the catalog (book symbol) key to see a menu of all the functions in the calculator

Standards

- NGSS 3-D Standards

- MS-PS1-1 Matter and its Interactions

- PS1-A Structure and Properties of Matter- solids may be formed from molecules, or they may be extended structures with repeating subunits (e.g., crystals)

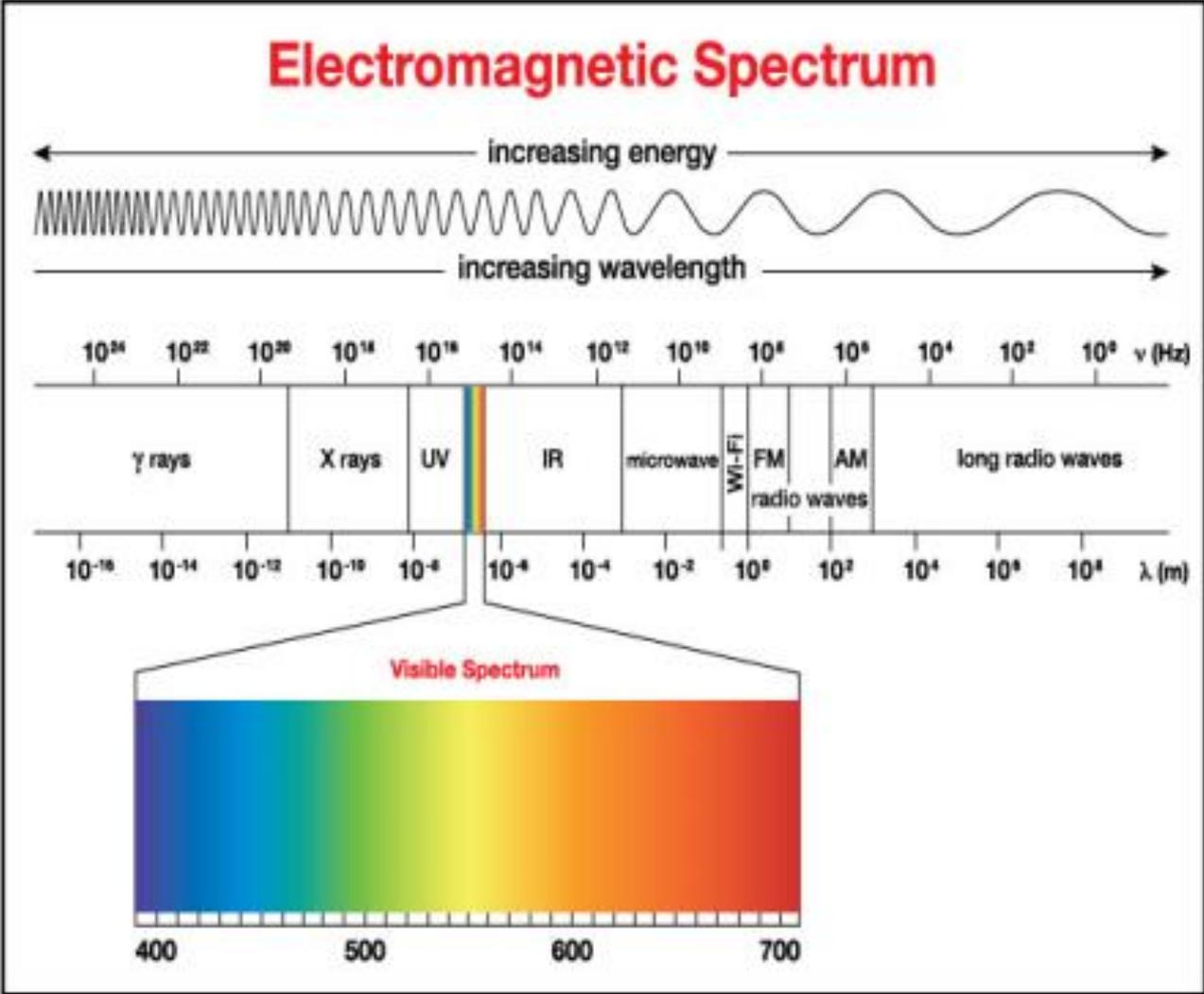
- Science and Engineering Practice:

- Developing and using Models
 - Develop a model to predict and/or describe a phenomenon

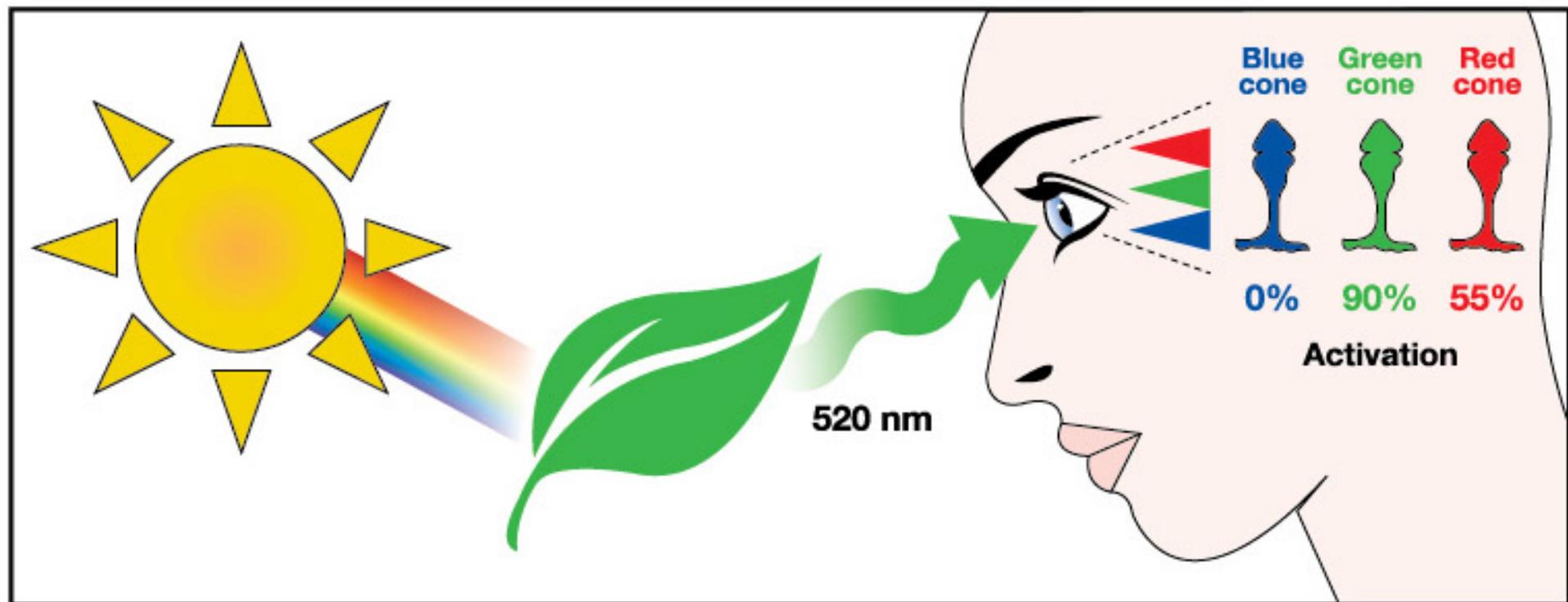
- Crosscutting Concept:

- Within a natural system, the transfer of energy drives the motion of matter

Background: The Electromagnetic Spectrum



How do we see color?



TI-Nspire CX Resources:

- Step-by Step YouTube videos designed for the teacher with a detailed, demonstration of each challenge. The videos are intended to guide those new programming in general, programming on the TI-84 Plus CE and programming with the TI-Innovator Hub.
https://www.youtube.com/playlist?list=PL17Fe0ZmhCR_j09b202PUI0wfaRYSLQka
- 10 Minutes of Code Teacher's Lounge (TI-Nspire CX): For a single download of all lessons for 10 Minute of Codes for the TI-Nspire CX in PDF's:
<https://education.ti.com/en/activities/ti-codes/nspire/teacher-lounge>
- TI-Innovator Technology eGuide
https://education.ti.com/html/webhelp/EG_Innovator/EN/index.html
- TI-Innovator Hub Commands
https://education.ti.com/html/webhelp/EG_Innovator/EN/content/eg_innovsys/resources/pdf/ti-innovator_hub_commands_en.pdf
- TI-Nspire CX Programming Editor Guide
https://education.ti.com/html/webhelp/EG_TINspireCode/EN/index.html
- YouTube Video (informal, but TI produced) gives you the basics of writing and running TI-Nspire programs that control the TI-Innovator.
<https://www.youtube.com/watch?v=2jjncu8Aba0>

PROJECT FINAL CHALLENGE

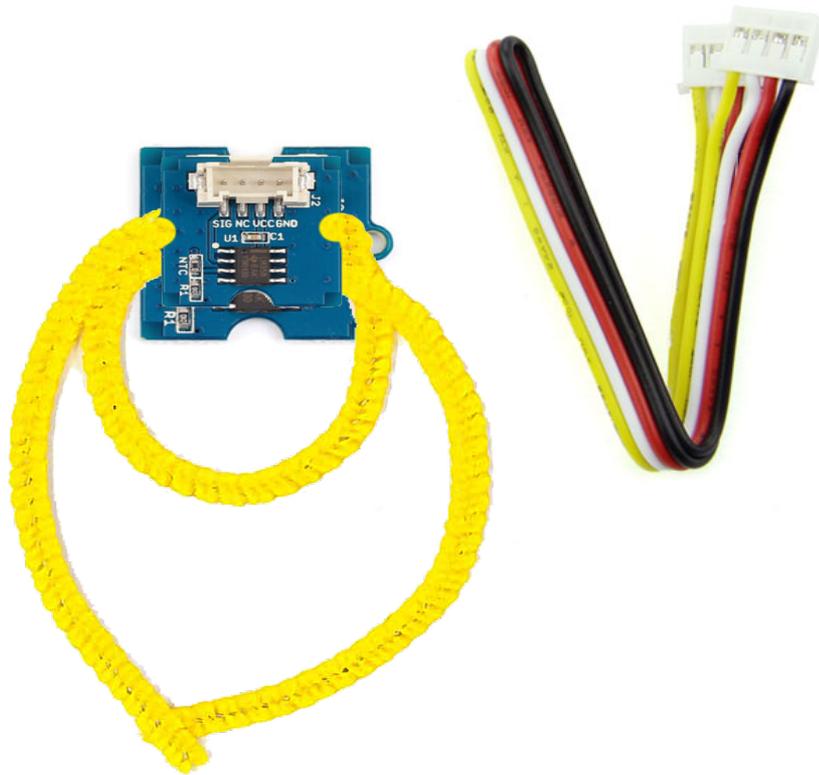


Final Challenge

Digital Mood Ring with TI-Nspire and TI-Innovator™ Hub student STEM project

Final Challenge: Build a mood ring to repeatedly read a temperature sensor, determine the mood of the person, display the temperature value and display the mood.

Chenille Wire (AKA “Pipe Cleaner”)



Defining your digital mood ring

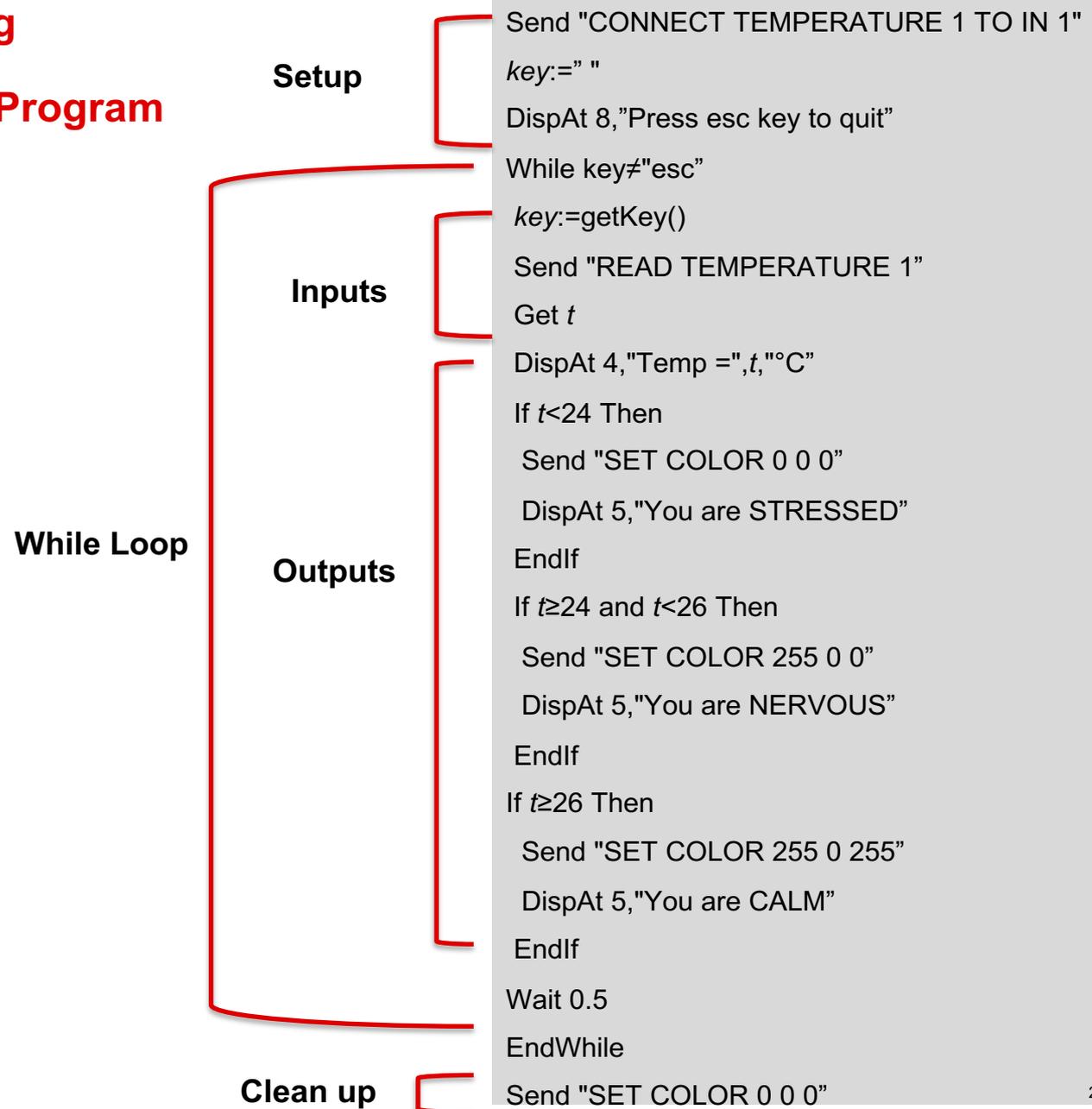
Temperature level definitions	Mood Message	Color Name or Description	Color RGB Code	Other Output (Optional) Sound? Vibration?...

Temperature level definitions	Mood Message	Color Name or Description	Color RGB Code	Other Output (Optional) Sound? Vibration?...
t<24	"You are Stressed"	off	0 0 0	
t≥24 and t<26	"You are Nervous"	red	255 0 0	
t≥26	"You are Calm"	magenta	255 0 255	

Digital Mood Ring for TI-Nspire Simple Example Program

```
Send "CONNECT TEMPERATURE 1 TO IN 1"  
key:= " "  
DispAt 8,"Press esc key to quit"  
While key≠"esc"  
  key:=getKey()  
  Send "READ TEMPERATURE 1"  
  Get t  
  DispAt 4,"Temp =",t,"°C"  
  If t<24 Then  
    Send "SET COLOR 0 0 0"  
    DispAt 5,"You are STRESSED"  
  EndIf  
  If t≥24 and t<26 Then  
    Send "SET COLOR 255 0 0"  
    DispAt 5,"You are NERVOUS"  
  EndIf  
  If t≥26 Then  
    Send "SET COLOR 255 0 255"  
    DispAt 5,"You are CALM"  
  EndIf  
  Wait 0.5  
EndWhile  
Send "SET COLOR 0 0 0"
```

Digital Mood Ring for TI-Nspire Simple Example Program



```
Send "CONNECT TEMPERATURE 1 TO IN 1"  
key:= " "  
DispAt 8,"Press esc key to quit"  
While key≠"esc"  
  key:=getKey()  
  Send "READ TEMPERATURE 1"  
  Get t  
  DispAt 4,"Temp =",t,"°C"  
  If t<24 Then  
    Send "SET COLOR 0 0 0"  
    DispAt 5,"You are STRESSED"  
  EndIf  
  If t≥24 and t<26 Then  
    Send "SET COLOR 255 0 0"  
    DispAt 5,"You are NERVOUS"  
  EndIf  
  If t≥26 Then  
    Send "SET COLOR 255 0 255"  
    DispAt 5,"You are CALM"  
  EndIf  
  Wait 0.5  
EndWhile  
Send "SET COLOR 0 0 0"
```

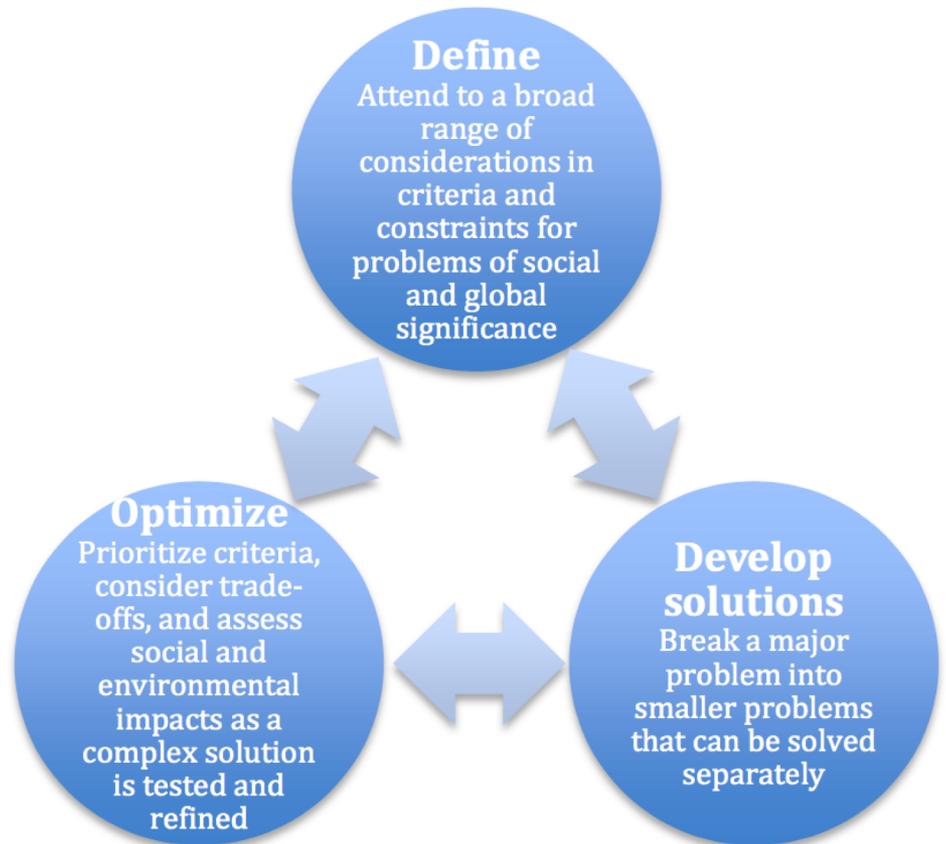
PROCESS OF ENGINEERING DESIGN AND COMPUTATIONAL THINKING

Engineering Design Process

- Define Problem
- Design a proposed solution
- Present group's design to class.
- Listen to feedback and make appropriate changes.
- Build and test the design.
- Use test results to make appropriate and informed changes to design.
- Present your solution to class.

Engineering Design Process

- **Defining** and delimiting engineering problems involves stating the problem to be solved as clearly as possible in terms of criteria for success, and constraints or limits.
- **Designing** solutions to engineering problems begins with generating a number of different possible solutions, then evaluating potential solutions to see which ones best meet the criteria and constraints of the problem.
- **Optimizing** the design solution involves a process in which solutions are systematically tested and refined and the final design is improved by trading off less important features for those that are more important.
- It is important to point out that these component ideas do not always follow in order, any more than do the “steps” of scientific inquiry. At any stage, a problem-solver can redefine the problem or generate new solutions to replace an idea that just isn’t working out.



Authors: NGSS Lead States

Title: Next Generation Science Standards: For States, By States – Engineering Design

Copyright Date: 2013

Source: Next Generation Science Standards

http://www.nextgenscience.org/sites/default/files/Appendix%20I%20-%20Engineering%20Design%20in%20NGSS%20-%20FINAL_V2.pdf

Computational Thinking

NGSS Science and Engineering Practices

Using Mathematics and Computational Thinking

<http://ngss.nsta.org/Practices.aspx?id=5>

Value of Pseudocode

• What is Pseudo code?

- Pseudo code is basically written instructions of what a program should do to complete an operation. The pseudo code is written in your speaking language. You simply write down the steps of an algorithm or procedure to make it easier for you or someone else to read it and convert it to a real programming language.

• Why was it made/is it used?

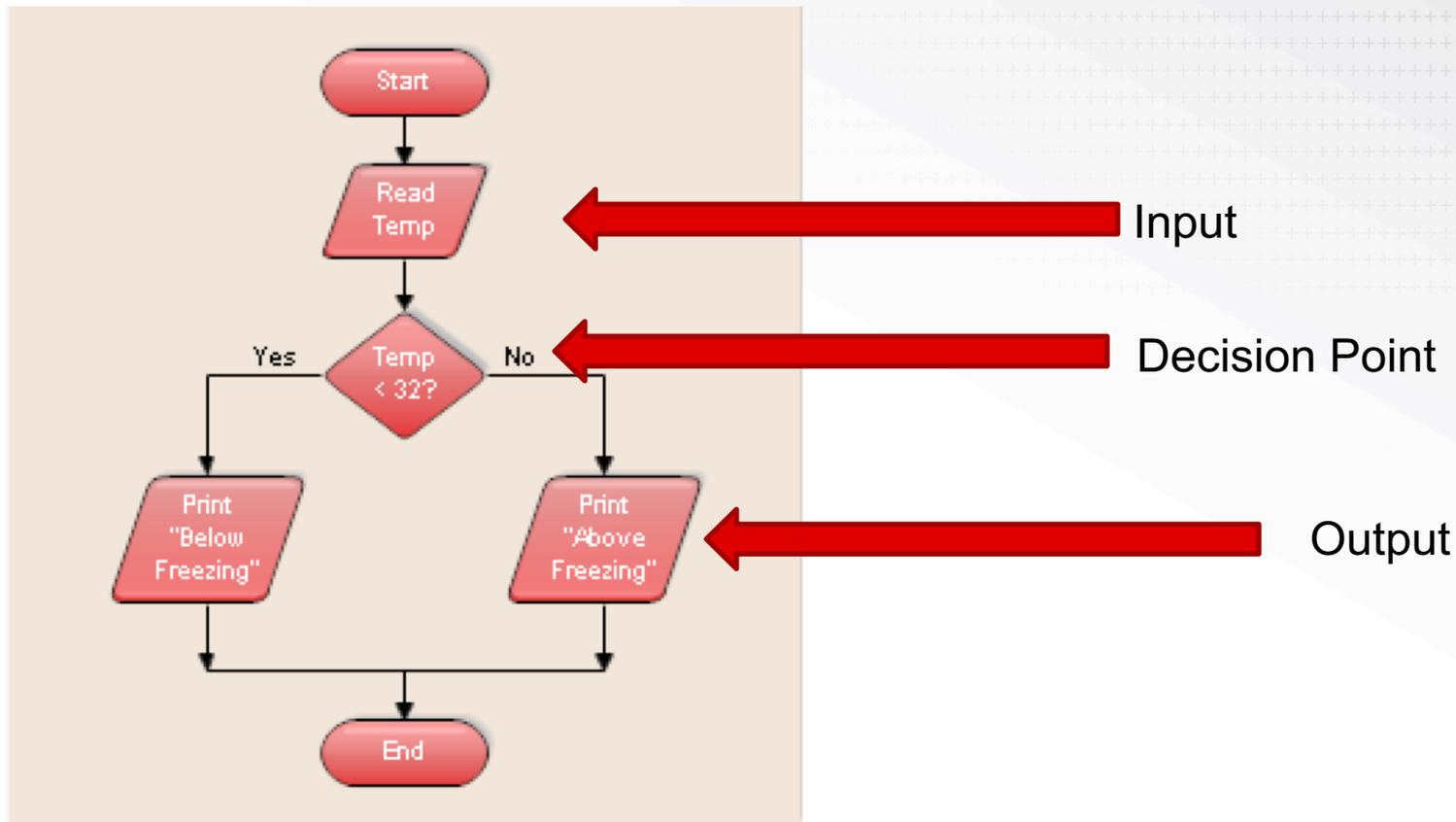
- It was made to help you specify the steps of a procedure, process or an algorithm without using a programming language. One possible alternative is to use flowcharts.

Planning your program

Important steps in your program	TI-Basic commands to implement steps in your program
1.	
2.	
3.	
4.	
5...	

Using a flowchart to plan:

Sample, simple flowchart:



QUICK CODE REFERENCE

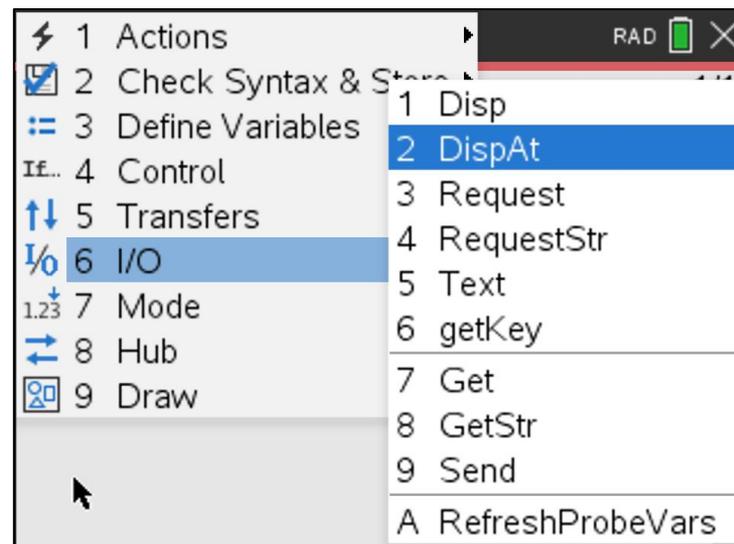
Background: Input/Output Commands for Displaying text strings and numeric values

The **DispAt** command allows you to control the text and values that are displayed on the rows of the Calculator application.

DispAt row value, text string or numeric value

Notes:

- Row values are integers from 1 to 8



DispAt 3,"Hello World"
DispAt 4, 2+3
DispAt 5,"Temperature= ",t

Displays string on row 3.

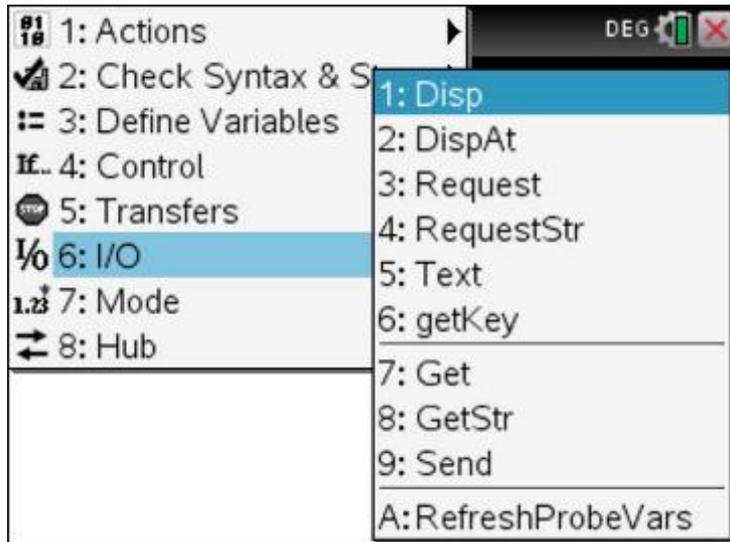
Displays calculation on row 4

Displays label and variable value on row 5



Background: Input/Output Commands for Displaying and Requesting text strings and numeric values

Program Editor Input/Output Menu



Display a string

Disp "hello world"

Disp 2+3

Display a numeric value

Request "Input a number",n

Disp n

Prompt

Variable for the input numeric value

Request a numeric value input

RequestStr "Input a string",s

Disp s

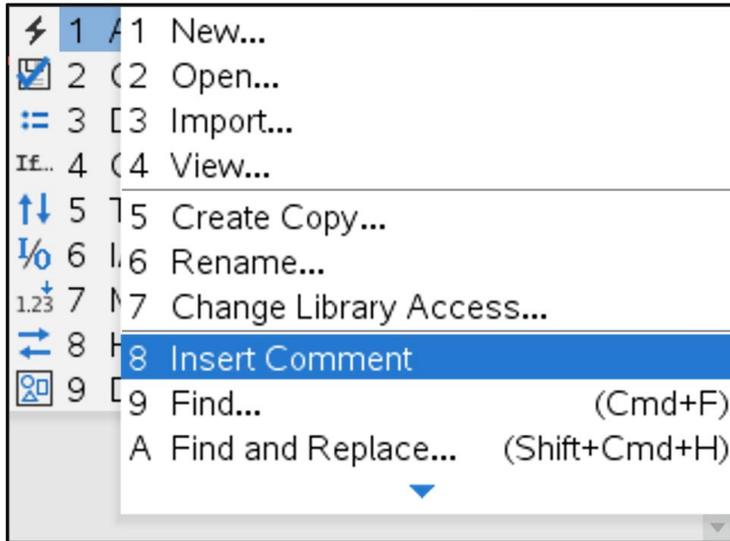
Prompt

Variable for the input text string

Request a string input

Background: Inserting a Comment from the Actions Menu

Program Editor Actions Menu



Comments are notes to explain your program. Commenting your code is a best practice for programming.

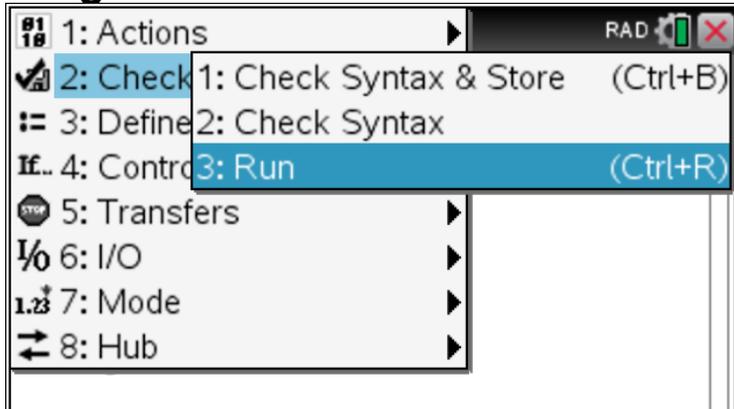
Insert a comment from the Program Editor Actions Menu. The Comment row is indicated by a ©.

When you Run the program comment rows are ignored.

© comment to explain the code

Background: For Loop

Program Editor Control Menu



Control variable
Initial Value
Ending Value

```
For c,1,3
```

```
  © Insert code to be repeated
```

```
EndFor
```

Background: Decision Making and Logic Statements

Decision Making Commands

If-Then command

If b>10 Then

© set of commands executed if statement is true

EndIf

If-Then-Else command

If b>10 Then

© set of commands executed if statement is true

Else

© set of commands executed if statement is false

EndIf

Logic functions

“and”

If b>10 and c>100 Then

© set of commands executed if both statements are true

EndIf

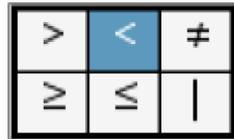
“or”

If b>10 or c>100 Then

© set of commands executed if either of the statements

EndIf

Note: Relational operators can be found on the keyboard using [=] and ctrl [=]. Logic functions, and, or, not, etc. can be typed or found in the catalog.



Background: While...EndWhile loop

The While loop repeats a set of commands as long as the While condition is true.

Command Syntax:

While *condition*

Commands in While block

EndWhile

```
b:=1
While b>.25
Send "READ BRIGHTNESS "
Get b
Disp b

EndWhile
```

Note: This program uses the BRIGHTNESS sensor as a control for the While loop. If you completely cover the sensor the While loop will stop.

Background: Advanced Version

Ending a loop using a keypress

Initialize default value.

Continually set the value of keys pressed.

```
key:=" "
```

```
While key≠"esc"
```

```
key:=getKey():
```

```
© set of commands executed while escape key isn't pressed
```

```
EndWhile
```

Background: Advanced Version

Storing Data in an Array

Declare an array variable

Inside a loop store a calculation

or

Store a value from a sensor

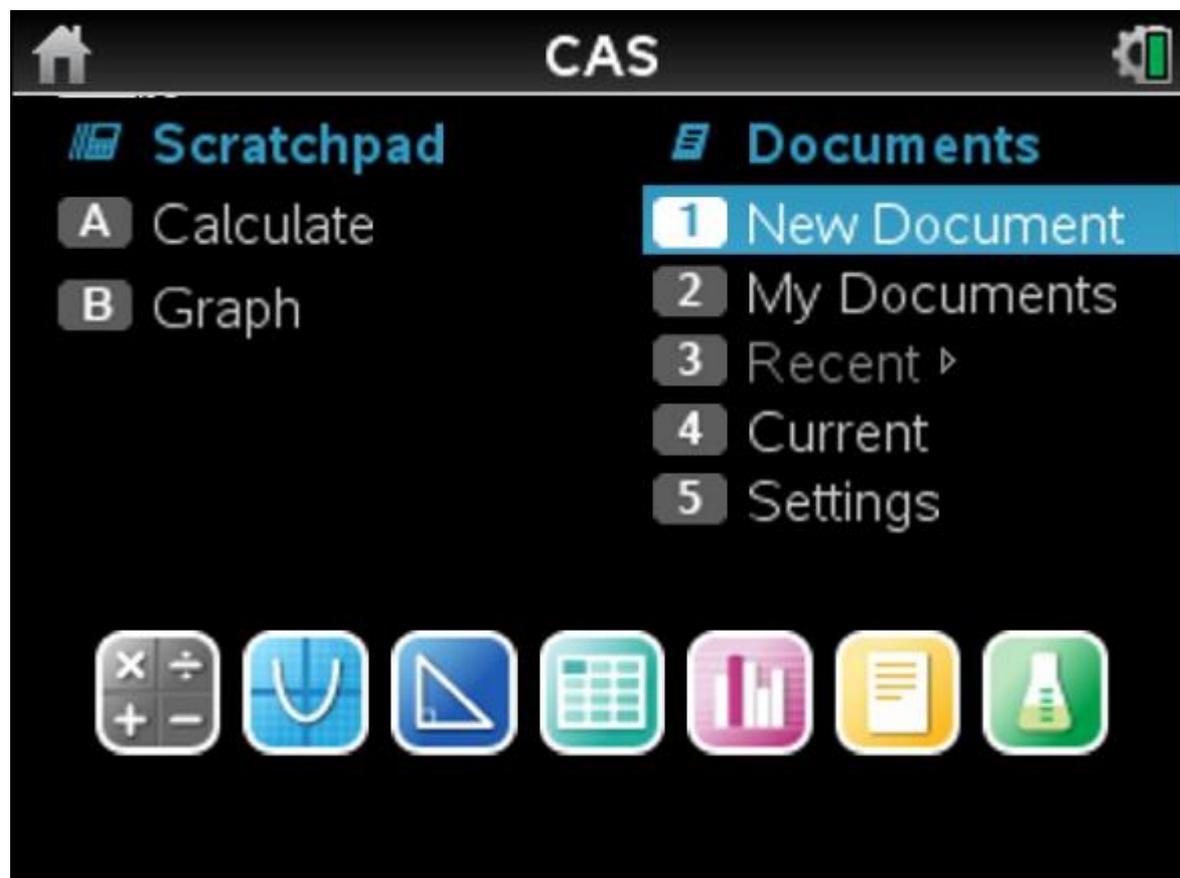
```
myarray:={}  
  
For c,1,3  
  
myarray[c]:=c*2:  
  
EndFor
```

```
myarray:={}  
  
For c,1,3  
  
Send "READ BRIGHTNESS "  
Get b  
myarray[c]:=b  
  
EndFor
```

Values can be plotted on a data and statistics page, graphs page, or used in other calculations.

CREATING, EDITING AND RUNNING PROGRAMS ON THE TI-NSPIRE

Create a new document



Press [Home] key

Select New Document from home menu

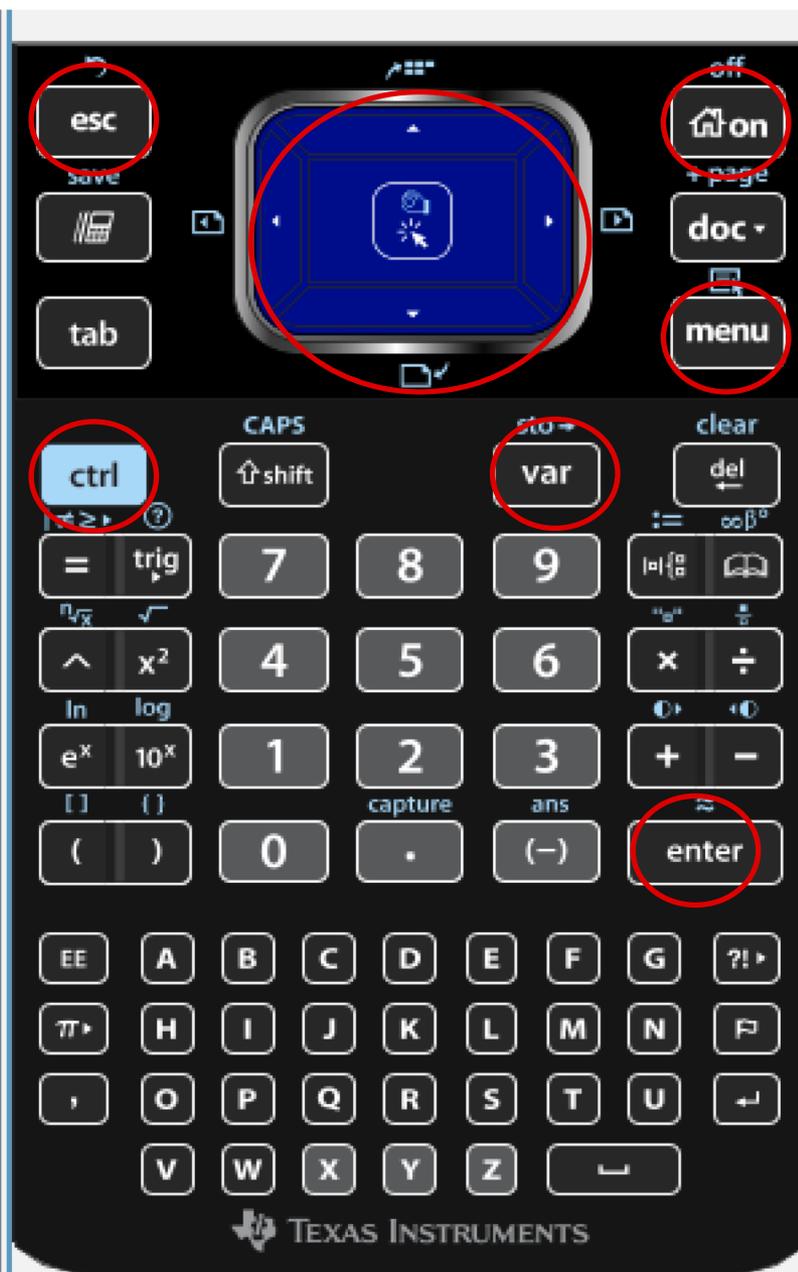
Important Keys

esc: Gets you out of what you are in

Ctrl z: is undo, just like on a computer

ctrl: Allows you to access the blue label functions printed above the keys.

touchpad: is like a mousepad, slide your finger around. Center click when finger icon is showing to select. You can also click the arrows to move the cursor space-by-space.



On: (House): Brings you to the Home screen (see next slide)

Ctrl doc (+page): adds a new page

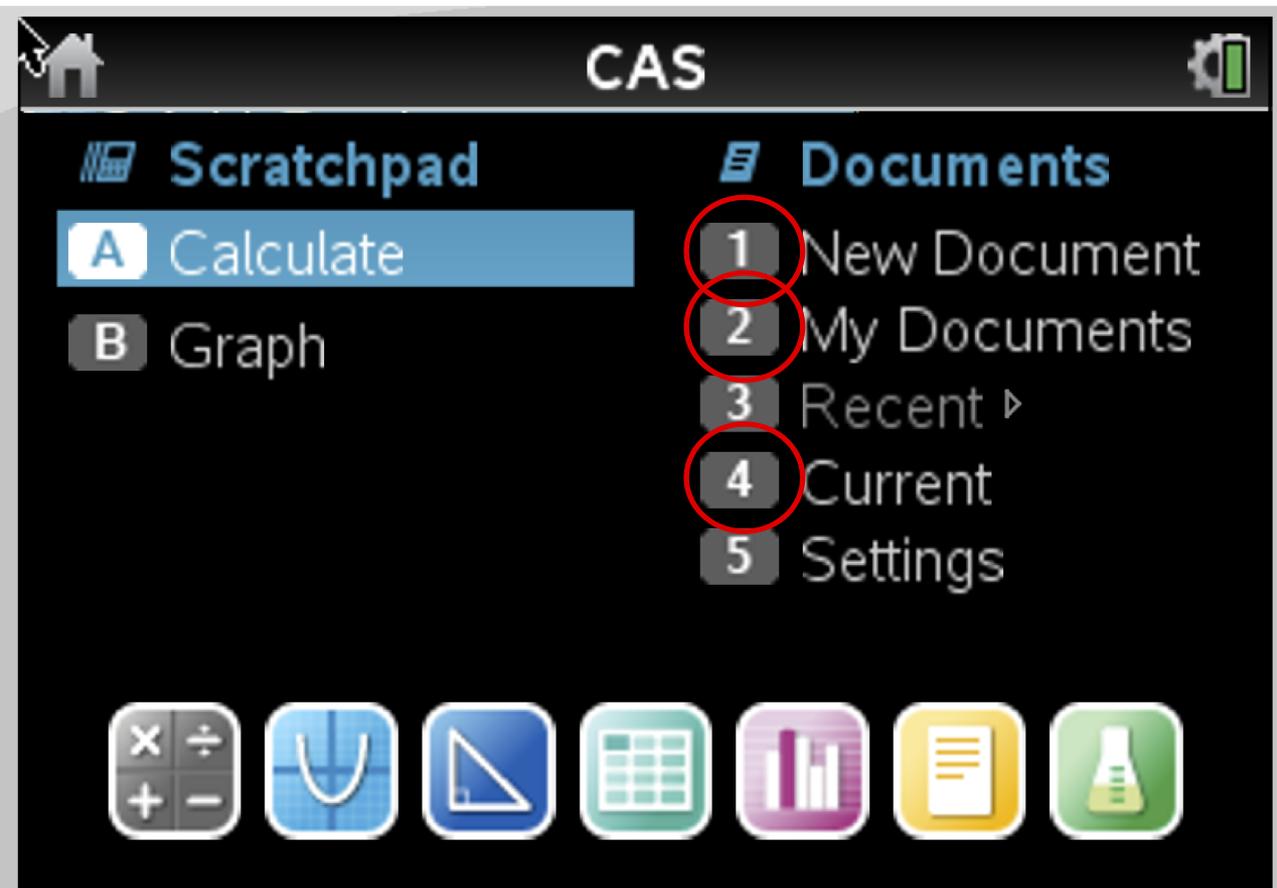
doc: has file commands. Save for example.

Menu: Gives you a list of commands that are relevant for your current application.

Var: gives you a list of variables, including programs to choose from to insert in your command.

enter: Completes and entry or selects a menu item or closes a dialogue box.

Home Menu

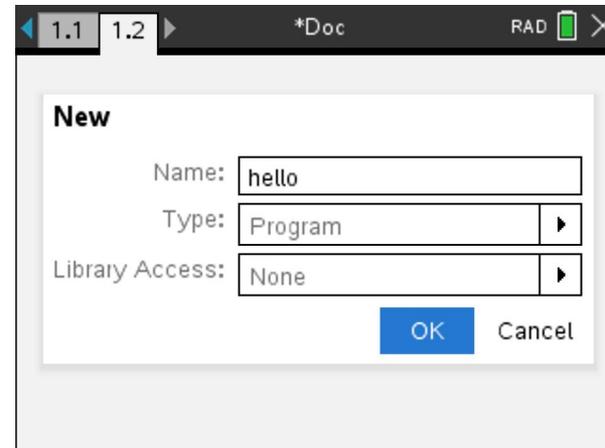
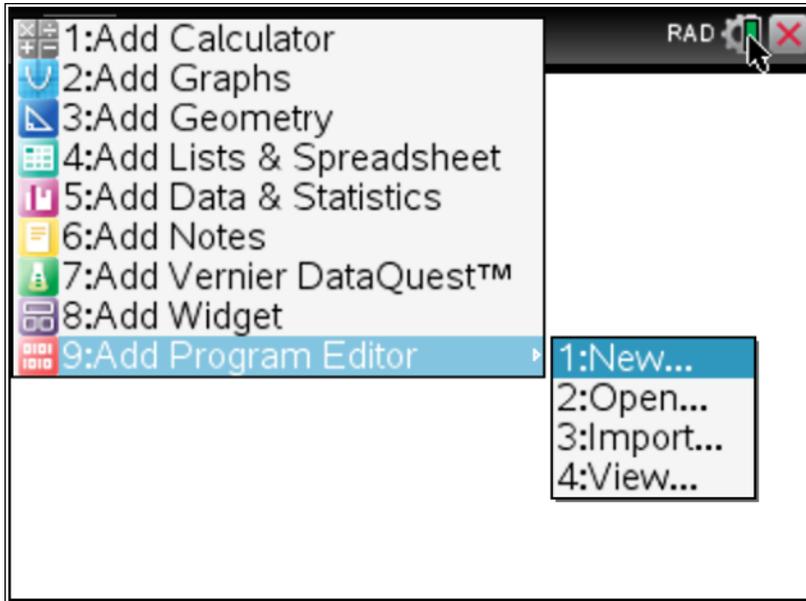


New Document: create a new document file.

My Documents: Brings you to folders with your stored files.

Current: Returns you to the document that you have open.

Set up a new Program Editor app

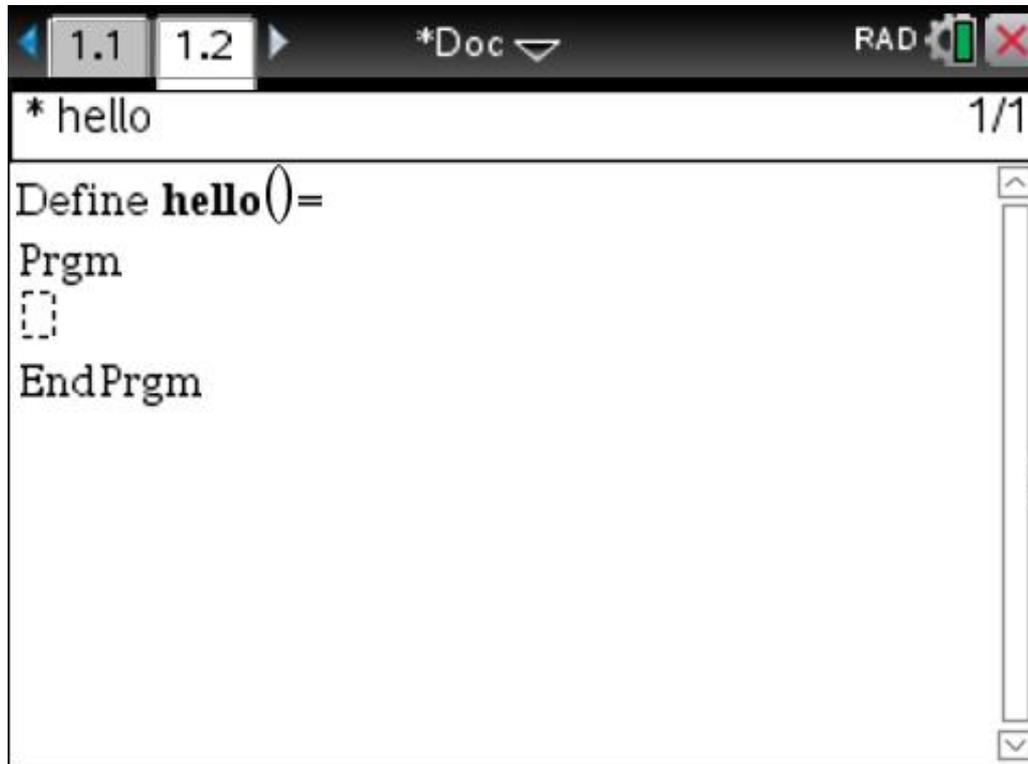


Press the menu key on a blank page or add a new page by pressing ctrl-doc (+page)

Select 9: Add Program Editor then 1: New. Enter your program name then press enter.

Note: Program names must begin with an alpha character.

Program Editor app

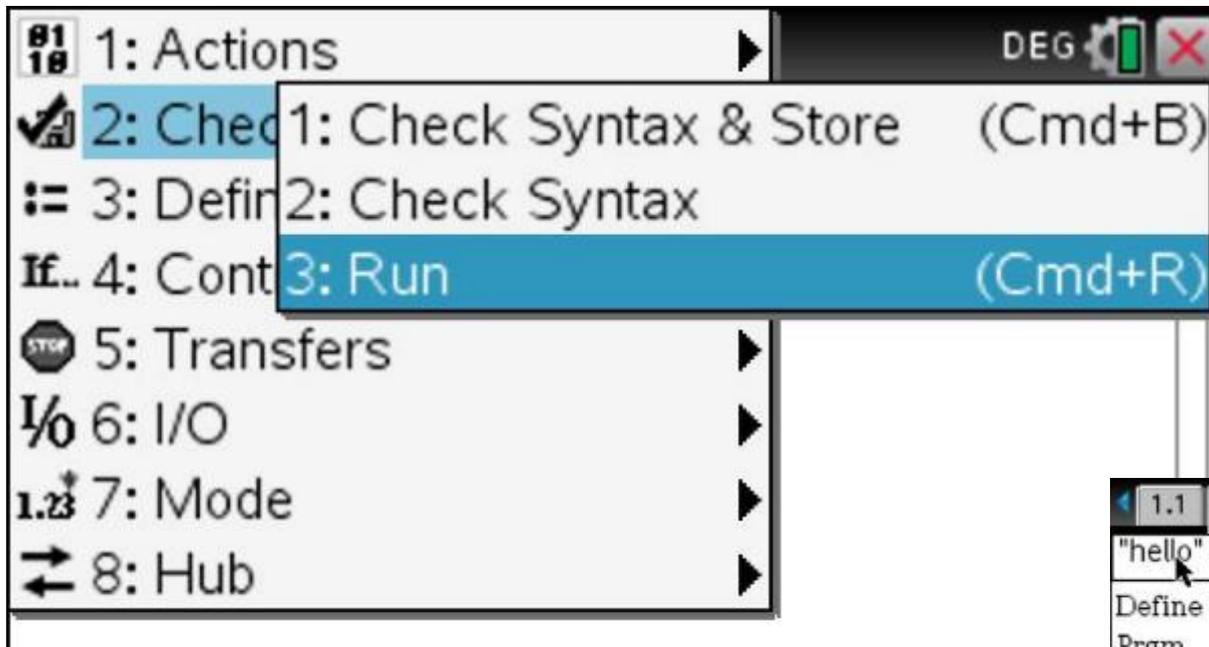


The screenshot shows a window titled "RAD" with a document icon and a close button. The window contains a code editor with the following text:

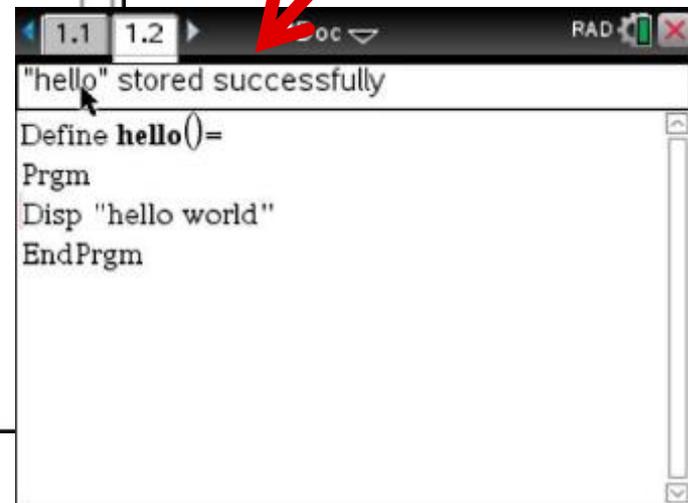
```
* hello 1/1  
Define hello()=  
Prgm  
[]  
EndPrgm
```

Remember to move cursor DOWN within the program

Store your program



Make sure the title says "stored successfully"



Use the Check Syntax & Store Menu to store your program. Look for the "stored successfully" message.

Use 1:Check Syntax & Store to store your program and remain in the editor. The shortcut is ctrl-b.

Use 3:Run to store and automatically paste the program name on a Calculator page. Then press Enter to run the program. The shortcut is ctrl-r.

Moving from page to page



You can move from page to page by pressing ctrl-LEFT arrow and ctrl-RIGHT arrow

or by using the touchpad to move the pointer to the desired page then click the center of the touchpad or press ENTER

Running your program from the Calculator app



1. Select the program name from the [var] menu.
2. Press [enter] to run the program.
3. "Done" indicates that the program is finished.

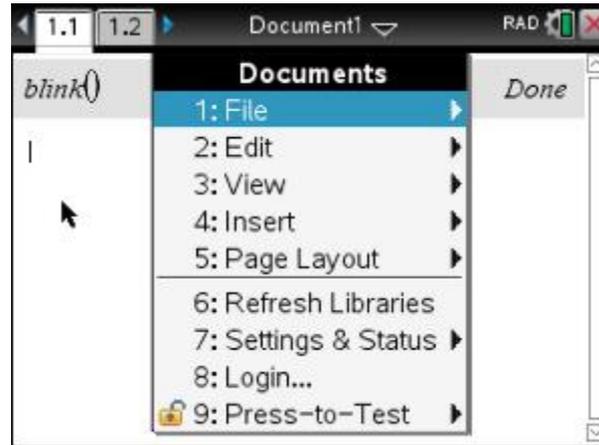
Note: The Calculator app "remembers" your last entry. Pressing [enter] again will re-run your last command.



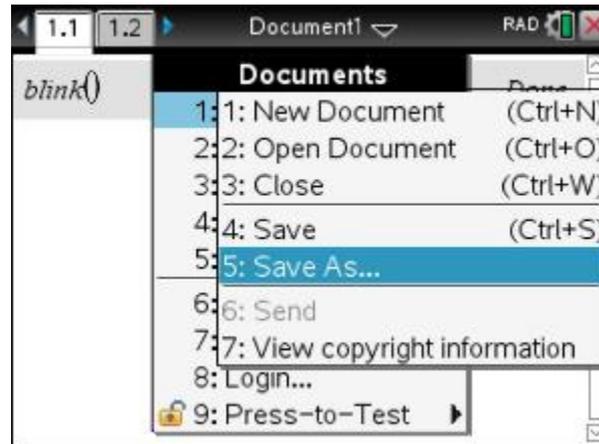
Save the TI-Nspire document file that contains your program



1. Press the [doc] key

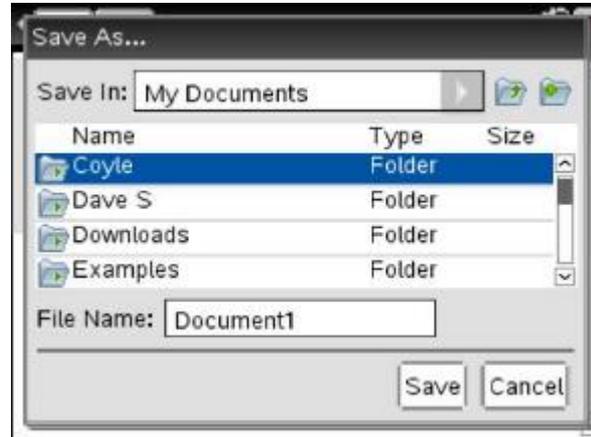
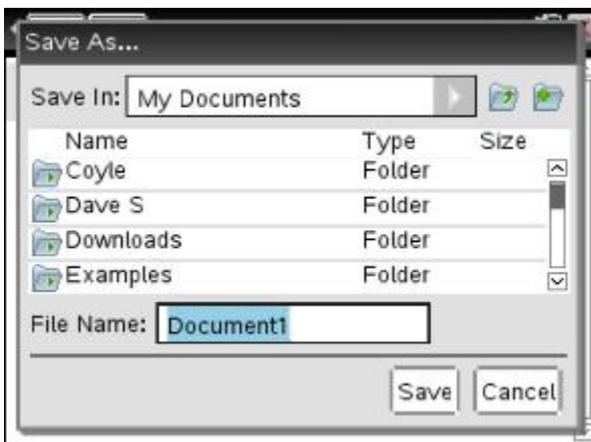


2. Select 1:File Menu

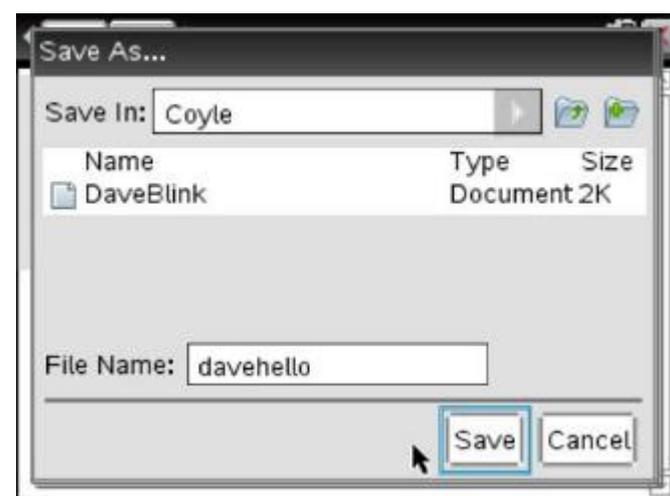


3. Select 5: Save As...

Save the TI-Nspire document with your program



1. Select a Folder with the touchpad and click



ctrl-s is a shortcut to save a file with the current name.

Have your students press ctrl-s periodically to assure that their work is saved.

2. Type in your file name - Press [enter] to save.