Drills - Setters, Getters, and Math

|  |  |  |  |
| --- | --- | --- | --- |
| Video  (English) | [Instructions](#_y7ss16yia2as)  [(English)](#_y7ss16yia2as) | Instructions (Spanish) | [Printable](https://docs.google.com/document/d/1H3K2ko6FcbzNRux6Q1mI-IgZExkBuV1xIdDOVURuVtI/preview) |

|  |  |
| --- | --- |
| Coding exercises involving ***setters***, ***getters***, and ***simple math***. The User Interface (UI) is given. For these drills, the app will get the user’s input from the Textbox (‘Type input here’), perform some kind of computation on the input, and display the result in the Label (‘Output shows here’).   CSP Learning Objectives:  * *Abstraction: 2.2.1 - Develop an abstraction (variables and expressions). [P2]* * *Programming: 5.1.1 - The student can explain how a program implements algorithms.* * *Programming: 5.2.1 - Use data abstraction (strings, concatenation, and integers). to manage complexity in programs. [P3]* * *Programming: 5.3.2 - The student can implement a correct program.* * *Programming: 5.4.1 - Employ appropriate mathematical and logical concepts in programming. [P1]* |  |

## 

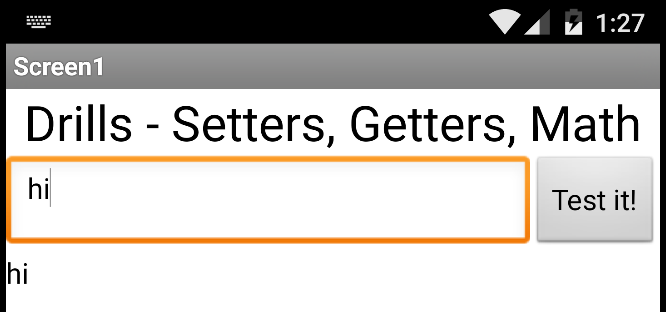
## Setup

Click here to open App Inventor and import the [Setters, Getters, and Math Drills Template](http://ai2.appinventor.mit.edu/?repo=templates.appinventor.mit.edu/trincoll/csp/drills2017/templates/DrillsSettersGettersMathTemplate/DrillsSettersGettersMathTemplate.asc). Use the *Save As* button to rename your project *DrillsSettersGetters* or something like that.

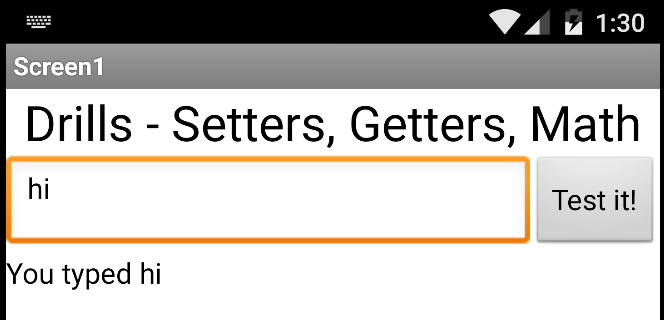
# Drills

For each of the following problems, code the solution in the Blocks Editor. Then run and test your solution. Keep running and testing until you are certain that it is correct. If you have trouble or questions, ask someone for help.  **For each exercise, you might want to do ‘SaveAs’ to preserve your individual solutions.**

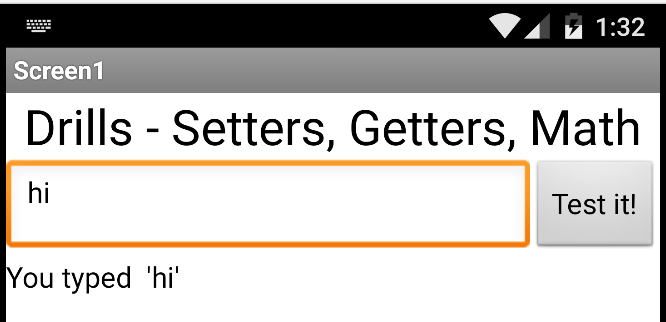
1. **Echo the input.**  Put together code in the blocks editor that will simply echo what the user has input when the ‘Test it!’ button is clicked. **Hint:** Use a ***global variable*** to store the input from the TextBox.



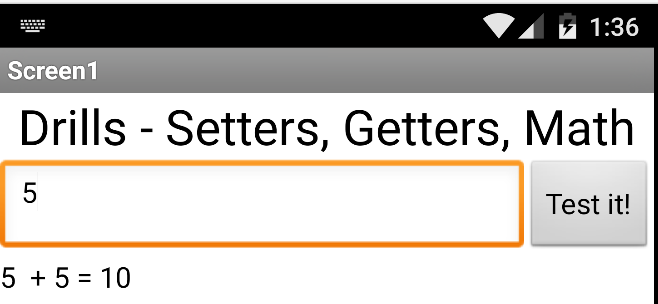
1. **Prettify the output.** Modify your solution to the previous exercise so that it ***joins*** ‘You typed ‘ to the user’s input and displays it. **Hint:** Use the join block in the Text drawer to ***concatenate*** the different parts of the output text.



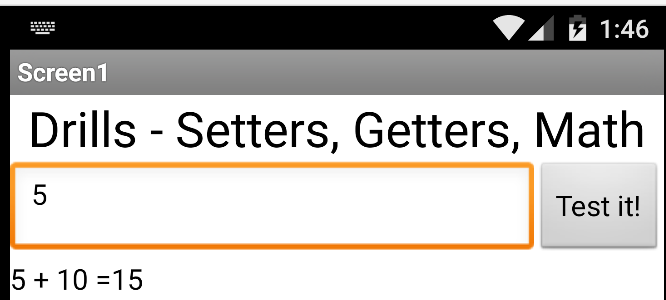
1. **Quotify the user’s input.** Modify your solution to the previous exercise so that it puts quote marks around the user’s input:



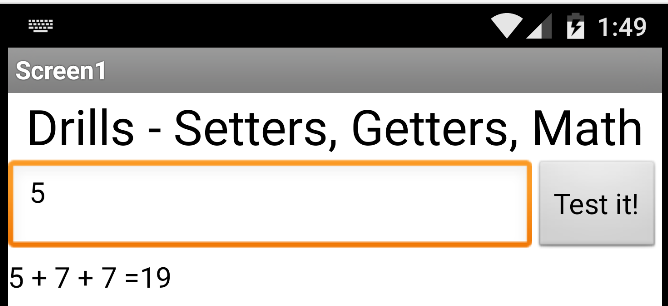
1. **Add 5 to the input number.** Modify the app so that the textbox is used to input



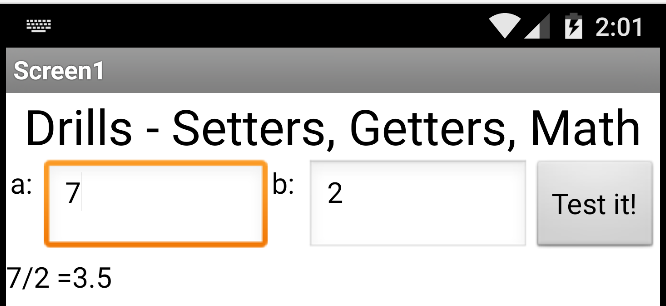
1. **Add a random value.**  Modify the previous exercise so that it adds a random number between 1 and 10 to the user’s input and displays the result as shown here. **HINT:** You’ll need a ***variable*** to store the random value for this problem.



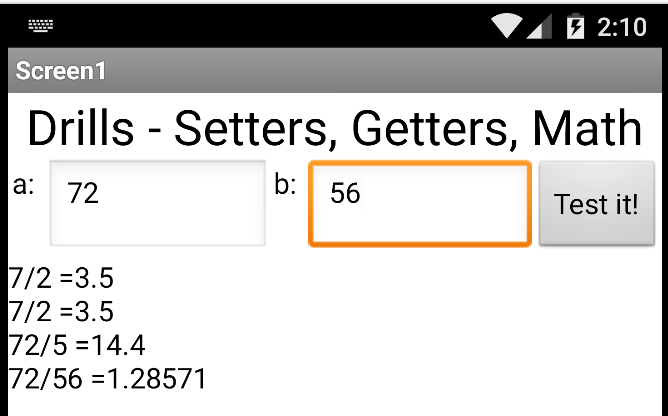
1. **Add a random value twice.** Modify your solution to the previous exercise so that it adds the random value twice -- e.g., 5 + 7 + 7 = 19



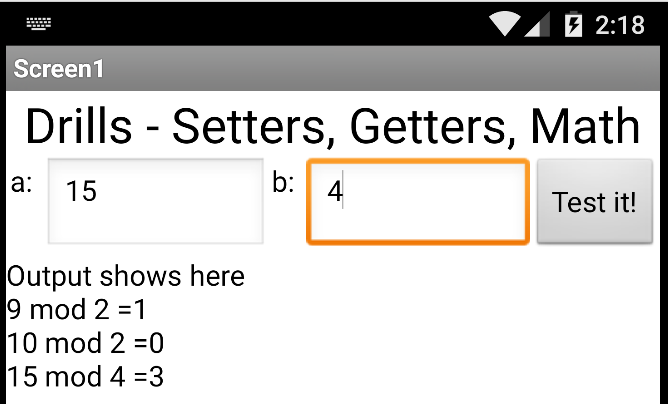
1. **Compute the quotient.**  (NOTE: For this exercise you’ll have to use the designer to modify the user interface so that it can input two values, *a* and *b.*) Input two values, *a* and *b* and compute and display the quotient, *a/b.* Don’t forget to give meaningful names to your components and variables. For this exercise, you should use two variables.



1. **Display Multiple Lines of Output.** Modify the solution to the previous exercise so that it displays multiple lines of output. **HINT:** Concatenate *LabelOutput’s* Text with the new result. To generate a new line after the LabelOutput’s Text, you can use the ***‘\n’ escape character*** (without the quotes) in your join block.



1. **Compute the remainder.**  Input two values, *a* and *b* and compute and display the remainder of *a* divided by *b*  -- i.e., [*a mod b*](https://en.wikipedia.org/wiki/Modulo_operation)*.* (HINT: Look for the *modulus/remainder* block in the Math drawer.)



1. **Calculate the Hypotenuse.**  Let the inputs *a* and *b* represent the sides of a right triangle. Then use the [Pythagorean formula](https://en.wikipedia.org/wiki/Pythagorean_theorem) for calculating the length of the hypotenuse where ***h = sqrt(a2 + b2)*** -- to calculate and report the hypotenuse as shown here. Two good test cases are 3, 4 and 5. That is, if the sides are 3 and 4, the hypotenuse will be 5. And 5, 12 and 13. If the sides are 5 and 12, the hypotenuse will be 13, as shown in the picture.

