

MIT App Inventor Variable Blocks

There are five main types of variable blocks:

- [initialize global name to](#)
- [get](#)
- [set](#)
- [initialize local name to in \(do\)](#)
- [initialize local name to in \(return\)](#)

initialize global name to



This block is used to create global variables. It takes in any type of value as an argument. Clicking on *name* will change the name of this global variable. Global variables are used in all procedures or events so this block will stand alone.

Global variables can be changed while an app is running and can be referred to and changed from any part of the app even within procedures and event handlers. You can rename this block at any time and any associated blocks referring to the old name will be updated automatically.

get



This block provides a way to get any variables you may have created.

set



This block follows the same rules as `get`. Only variables in scope will be available in the dropdown. Once a variable *v* is selected, you can attach a block to give *v* a new value.

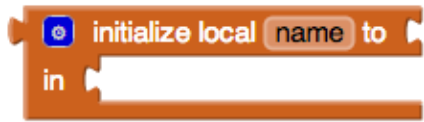
initialize local name to - in (do)



This block is a mutator that allows you to create new variables that are only used in the procedure you run in the DO part of the block. This way all variables in this procedure will all start with the same value each time the procedure is run. NOTE: This block differs from the block described below because it is a DO block. You can attach *statements* to it. Statements *do* things. That is why this block has space inside for statement blocks to be attached.

You can rename the variables in this block at any time and any corresponding blocks elsewhere in your program that refer to the old name will be updated automatically

initialize local name to - in (return)



This block is a mutator that allows you to create new variables that are only used in the procedure you run in the RETURN part of the block. This way all variables in this procedure will all start with the same value each time the procedure is run. NOTE: This block differs from the block described above because it is a RETURN block. You can attach *expressions* to it. Expressions *return* a value. That is why this block has a socket for plugging in expressions.

You can rename the variables in this block at any time and any corresponding blocks elsewhere in your program that refer to the old name will be updated automatically

© 2012-2024 Massachusetts Institute of Technology

This work is licensed under a Creative Commons Attribution-ShareAlike 3.0

Terms of Service and Privacy Policy

Support / Help
Other Inquiries
Twitter: @MITAppInventor
GitHub: mit-cml
Accessibility: accessibility.mit.edu