

The Building Blocks of Logo (Lynx)

Gary Stager, Ph.D.

Lynx procedures are a sequence of instructions with a name. They are defined in the procedures pane beginning with the word `TO` and ending with the word, `END`. The **procedures** built into a language like Lynx, a modern dialect of [Logo](#)¹, are called **primitives**. You execute the procedure by calling its name.

Programming languages typically feature two kinds of instructions, **commands** and **reporters**. In some languages, reporters are known as functions. Logo commands and reporters are procedures that may be combined to create **super procedures**. Other languages may use terms like routines and subroutines to mean the same thing.

Procedures are self-contained programs that perform a particular task and when combined with perform more complex actions. The procedures used in a larger program are called **subprocedures**.

In Lynx, procedures are created in the procedure pane and executed/run in the command center, buttons, turtles, or sliders.

Even when working with a text-based programming language, the block metaphor is useful in understanding how procedures work. Think of procedures and their built-in cousins, primitives, as little helpers used to

build a working computer program. Each little helper may stand alone or work together.

Commands perform a task or do a job. They may have any number of inputs, but do not output anything.

The values inputted into commands or reporters may be numbers, words, or lists. These values are sometimes called parameters or variables. Logo inputs are not typed, meaning that procedures do not require you to specify their type. However, you may end up with bugs related to using the wrong data type. For example, you cannot add Lemon and Pie like you would $3 + 2$. So, if your procedure expects numbers, or words, or lists, give the program what it expects.

Reporters report a result, hoping that result will be caught by a command. For this reason, using a reporter without a preceding command will generate an *I don't know what to do with...* error message. Reporters always report **one** thing/value and then are done with their work. Reporters never output more than one thing.


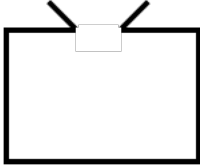
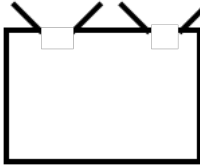
When writing your own programs, you may create procedures with any number of inputs. You want each of your “helpers” to be as smart and reusable as possible.²

¹ Read the [History of Logo](#) for extensive background on the design, evolution, and philosophy behind the Logo programming language. Solomon, C., Harvey, B., Kahn, K., Lieberman, H., Miller, M. L., Minsky, M., Papert, A., & Silverman, B.

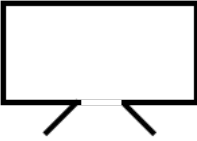
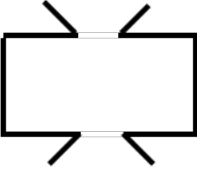
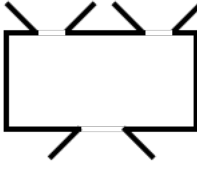
(2020). History of Logo. *Proceedings of the ACM on Programming Languages*, 4(HOPL), 1-66.

² Logo procedure and primitive names do not need to be capitalized. I do so here for purposes of clarity.

An Assortment of Commands

0 input commands	1 input commands	2 input commands
		
<i>examples</i>	<i>examples</i>	<i>examples</i>
CG CLEAN CT STOP	FORWARD # FD # RIGHT # RT # SETC # SHOW	REPEAT # [list of instructions]
<h1 style="color: lightblue; text-align: center;">What other primitives would you include in this table?</h1>		

An Assortment of Reporters

0 input reporters	1 input reporters	2 input reporters
		
<i>examples</i>	<i>examples</i>	<i>examples</i>
POS XCOR COLOR SHAPE HEADING	SQRT # RANDOM # PICK <i>thing</i> FIRST <i>thing</i> BUTFIRST <i>thing</i>	SUM # # Power # # WORD " <i>thing1</i> " " <i>thing2</i> " ITEM # <i>thing</i>
<h1 style="color: #4F81BD; text-align: center;">What other primitives would you include in this table?</h1>		

Challenge questions

1. Where would you place IF in the table of commands or reporters? What about IFELSE?
2. Where does SETPOS belong in the tables above?
3. What sort of procedure is OUTPUT? Where would you list OUTPUT in the tables?

But wait! There's more!

There is a second kind of reporter procedure in Logo/Lynx, called a **predicate**. Predicates report true or false. That is their entire job. They evaluate a situation and report either true or false. These predicate procedures are incredibly helpful in logic situations, like IF or IFELSE.

In Lynx, primitive predicates always end with a ? mark. EQUAL?, WORD?, LIST? MEMBER? Older dialects of Logo denote predicates by ending them with a P instead of ?. This is good to know if you wish to translate "classic" Logo programs into Lynx.

Note

Logo/Lynx is a prefix language, meaning that instructions come before inputs/variables/parameters. Exceptions are made for commonly used arithmetic functions like +, -, *, /, =, <, >. However, since the language prefers sum 2 3 to 2 + 3, it's often a good idea to put parentheses around infix operations. For example, (2 + 3). User created reporters must be in prefix form.

Always be sure to use spaces between symbols, inputs, and procedure/primitive names!

