

Getting Started with Lynx

Turtle Cheat Sheet

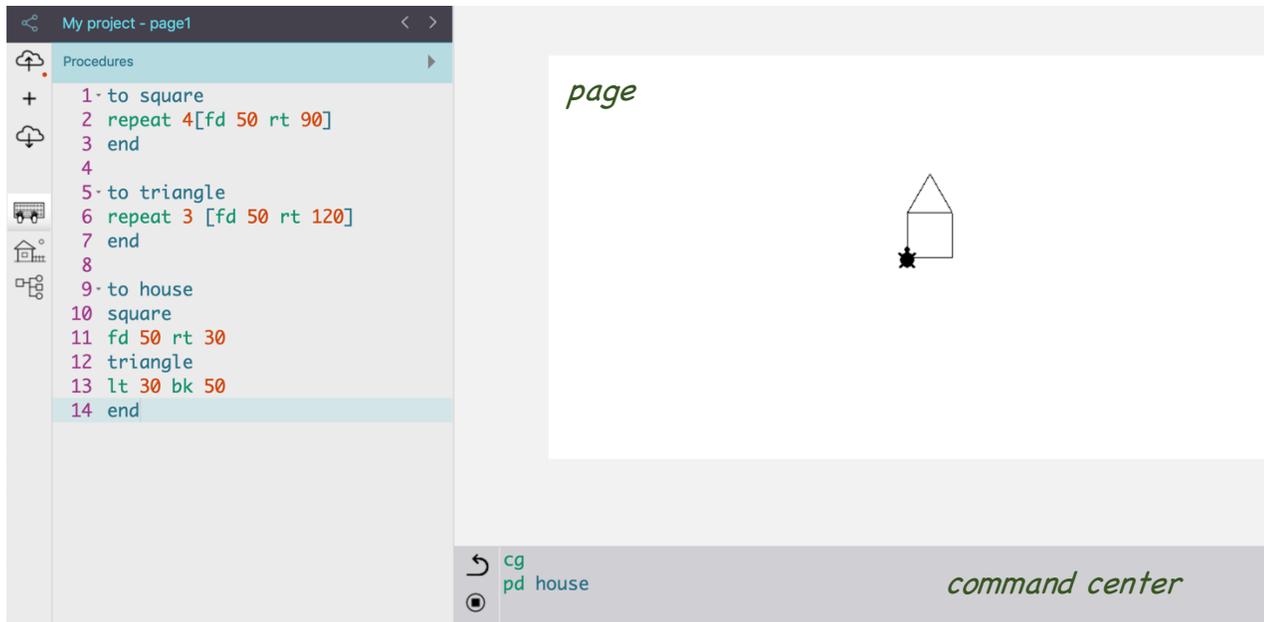
Notes:

- # is the sign for inserting a number as the input to a command
- Be sure to use spaces between words and numbers!

| | | | |
|--|---|--|----------------|
| Forward # FD # For example, <code>fd 50</code> | Back # BK # | Right # RT # | LEFT # LT # |
| CG clear graphics Clears the screen and puts the turtle at the center | Clean Clears the screen, but leaves the turtle where it is | PU Pen up | PD Pen down |
| REPEAT # [list of commands] For example, <code>repeat 4[fd 62 rt 90]</code> | | SETC # set color SETC 57 SETC "black SETC "red | |
| SETPOS [# #] For example: <code>setpos [10 20]</code> <code>setpos [-25 10]</code> <code>setpos [-10 -20]</code> <code>setpos [20 -25]</code> | | SHOW POS Displays the current position of the turtle (in coordinates) in the command center | |
| SHOW 3 *4 Shows the product of 3 and 4 in the command center. This is the same as asking the turtle to multiple 3 X 4 Show runs a reporter or operation and displays the result in the command center. | | | |

Writing and Running Procedures

A procedure is a list of instructions with a name. All procedures begin with `to` and end with `end`.



Procedures to Define

Enter the following procedures into the procedures area of Lynx.

You may test each procedure in the command center by typing its name.

```
to square
repeat 4[fd 50 rt 90]
end
```

```
to triangle
repeat 3 [fd 50 rt 120]
end
```

```
to house
square
fd 50 rt 30
triangle
lt 30 bk 50
end
```

Procedures with inputs

Just like `forward` or `right` are commands that require additional information as input, we can create our own procedures with inputs. Inputs allow us to use a procedure with different values.

`A :` with a letter or word next to it in the title line of a procedure creates an input. Procedures may have one or more input.

Change the procedures above in the procedures area to read like these.

```
to square :steps
repeat 4[fd :steps rt 90]
end
```

```
to triangle :side
repeat 3 [fd :side rt 120]
end
```

```
to house :x
square :x
fd :x rt 30
triangle :x
lt 30 bk :x
end
```

Test each procedure in the command center to see if it works. Don't forget to include a number as input!

For example:

```
square 30 square 50 square 15
triangle 25
house 10 + 50
```