



## Homework 3: Lightbot Moon Walk

**Goal:** The purpose of this exercise is to learn how to express a function symbolically.

**Abstract:** You will write three functions in this assignment; for the last one, the bot does the Moonwalk.

This assignment is a continuation of the last assignment when you programmed the Lightbot symbolically, that is, with text instructions instead of iconographic instructions. Recall the following association:



Step   Right   Left   Jump   Power   F.name

We also introduced iteration, that is, repeating operations, as in **4:Step**. (Check the earlier assignment if you have forgotten.)

### **Symbolic Function Definitions**

When we define functions symbolically, we use a special form. For the function *definition*, that is, saying how it works, we write the name, a pair of parentheses, the operation sequence, and a final period. (The purpose of the parentheses will be clear later.) For example, this is a function definition for a function that turns the bot around:

F.turn\_around( ) Right, Right.  
           name                    body

The *name* of the function is the part between the dot and the open parenthesis, and the part after the closing parenthesis to the period, is called the *body*. It defines how the function works.

When we use a function, it's called the *function call* or *function invocation*, we also have a special form. We give the name, followed by the parentheses, as in `turn_around( )`, which instructs the bot to do the instructions in the body of the function.

For example, to program the bot to “turn around and jump”, we would write the program

`turn.around( ), Jump.`

When the bot gets these instructions, it runs the function by checking the function definition, and following its instructions of the function body; when complete it returns to do the instructions following the call.

**Important.** Notice that a single function like `turn.around( )` has two roles: in one role it is defined; in the other role it is called. A function needs only one definition, because it's necessary to tell how it works only once. A function can be called many times, because every time we need the operation it performs, we call it.

### **Example: The Moon Walk**

Being a rock music listener and a fan of the late Michael Jackson, the Lightbot 2.0 likes to do the Moonwalk. If we tell the bot, when its standing on a raised block, to walk, its legs and arms move, but it doesn't go anywhere. This leads to the bot version of the Moonwalk. This is the Moonwalk definition:

```
F.moon_walk( ) 4:(Step, Right).
```

According to the function body, the bot's Moonwalk is four repeats of taking a step (and not going anywhere) and then turning right. So, it walks in each direction for one step.

To use the `moon_walk( )` function after jumping up two steps, for example, we could write

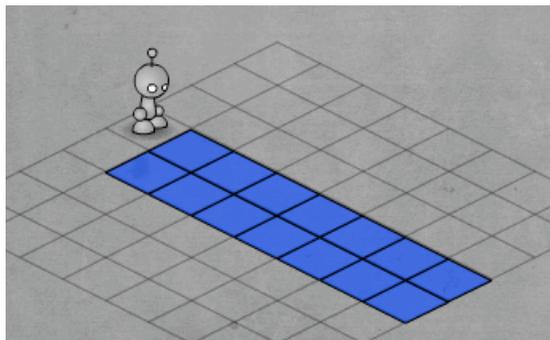
```
Jump, Jump, moon_walk( ).
```

This is an example of calling the `moon_walk( )` function. The program asks the bot to up twice, and then call the `moon_walk( )` function, which causes it to do the operations in the function's definition.

### **Exercises**

**Assignment Part A.** Consider a different solution to a problem (3a) from last time.

Suppose we have written the program `7:light_a_pair( )`. Define – that is, write out the function definition – for the `F.light_a_pair( )` function so that the command works for our program.

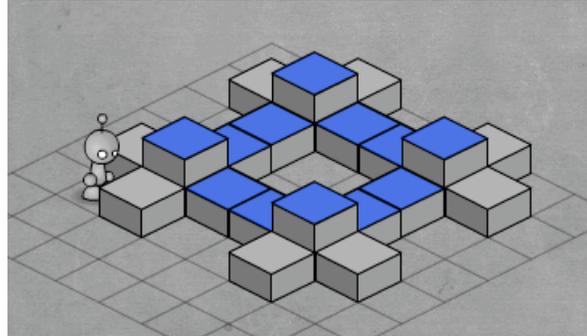


(3a)

**Assignment Part B.** The program for problem 3b ends with the instruction `4:light_a_side:`

Jump, Left, Jump, Right, 4:light\_a\_side.

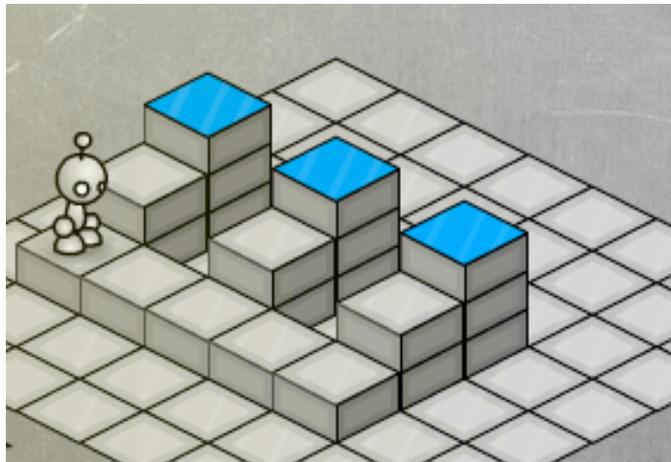
Write the F.light\_a\_side function definition so that the program works. Your function will probably take about seven instructions.



(3b)

### ***Moonwalk***

The Lightbot wants a new solution to the Basic Level 6 of the Lightbot 2.0. (You solved Level 6 in assignment 1. Find it at <http://armorgames.com/play/6061/light-bot-20>.) What the bot wants is to go up each riser (see Figure 3c) and do its version of a Moonwalk on the top before powering the light. (As explained above the bot's Moonwalk exploits the fact that the bot cannot walk forward if there are no tiles in front of it ... so it just walks in place.) You will need a different function definition for the Moonwalk than above.



(3c)

**Assignment Part C.** Solve the problem in 3c so that the bot does a Bot Moonwalk at the top of each riser *before* powering the light. (You will probably want to use another function to simplify your work.) Include a copy of your moon\_walk( ) definition with your solution.

### ***To Turn In***

The solutions to all three parts – written out in a document (Word, perhaps) – are to be submitted in the course homework drop box; see the calendar page.

### ***Wrap Up***

In this assignment you learned how to express functions symbolically. There is a function definition, with a three-part structure – name, parentheses, body – and a function call, also with a standard form with parentheses at the end. You wrote functions and a program to demonstrate your understanding of using these concepts.