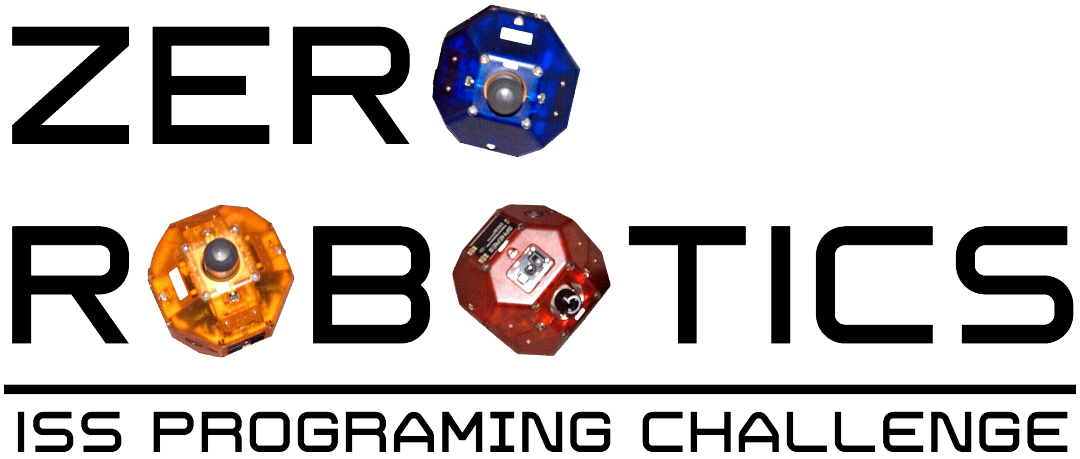
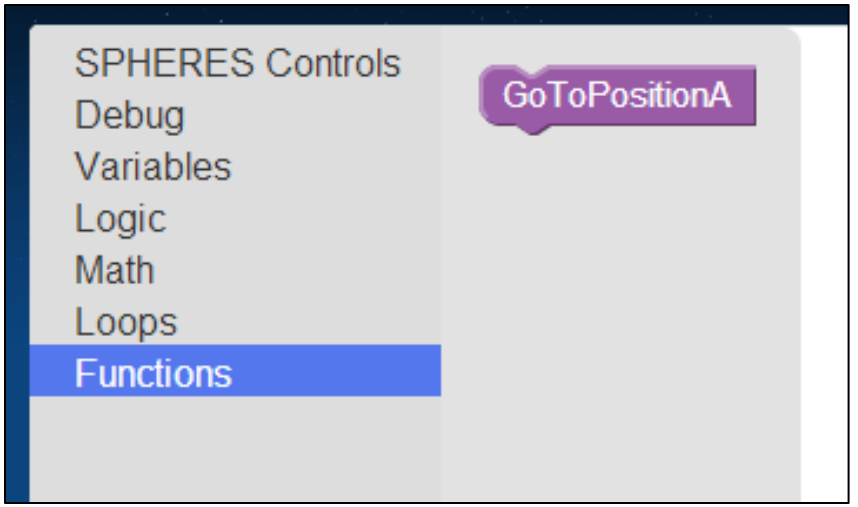


Creating Functions

Goals

In this tutorial you will learn how to create a procedural function to organize your program.

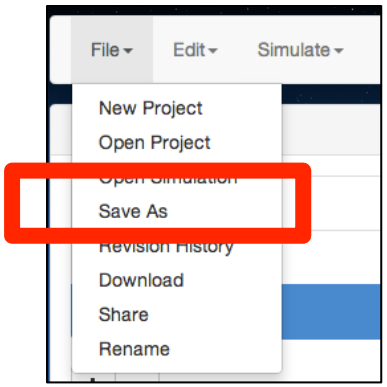


Review about functions

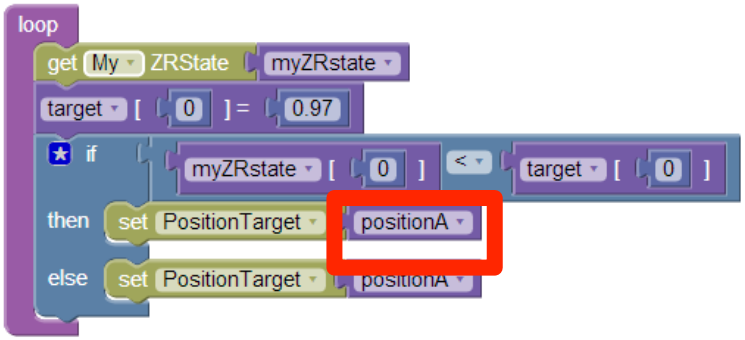
- What is a Function?
  - Programmers find it easier to break up their code into separate sections that perform different tasks.
  - A programmer puts a set of instructions for a particular task into a *function*. This function can then be "called" and used in their program.
  - Creating functions:
    - Helps to organize the code and makes it easier to keep track of what it happening
    - It is also useful to create a function for code that is used more than once in a program instead of duplicating/repeating the code in multiple places
- In this tutorial you will learn how to create a Procedural Function to help organize your program
- This tutorial focuses on: How to Create a Function
- The next tutorial "Functions and the Step Counter Model" will demonstrate how functions can be useful for simplifying and organizing a more complicated program

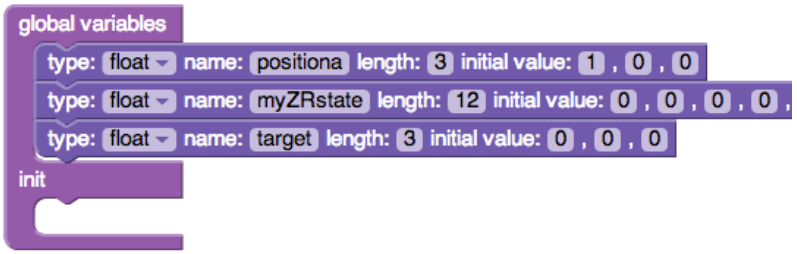
Create a new project using "Save As"

- First create a new program by re-naming your previous program as follows:
  - Open the ZR IDE
  - **Open Project 9** (from the *Applied Conditionals tutorial*)
  - On the menu bar select "File" and then "Save As" from the drop down menu.
  - Type in **Project 10,** select **Graphical Editor**, **Free Mode**



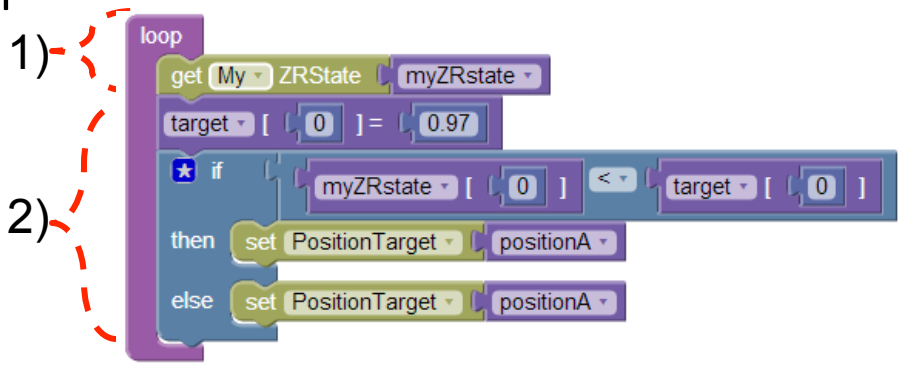
- Before we get started we will simplify this program so that the function you create can also be used in the next tutorial
  - In the "else" slot, change the variable in the **`setPositionTarget`** block to "positionA" shown.
  - In the Init page click on the positionB block delete it.
  - The init page should look like this





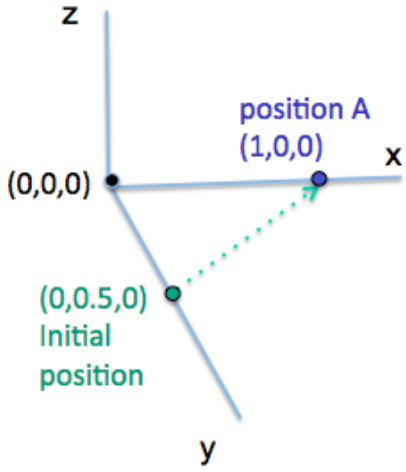
Create a new program

- For this tutorial we will divide the program into two sections:
  - 1) Get ZR State information
  - 2) Send the SPHERE to positionA by:
    - Setting a target value for deciding if the SPHERES has reached positionA
    - Using the State information to send the SPHERES to positionA until it reaches the target value
    - Then telling the SPHERES to stay at positionA after it reaches its target



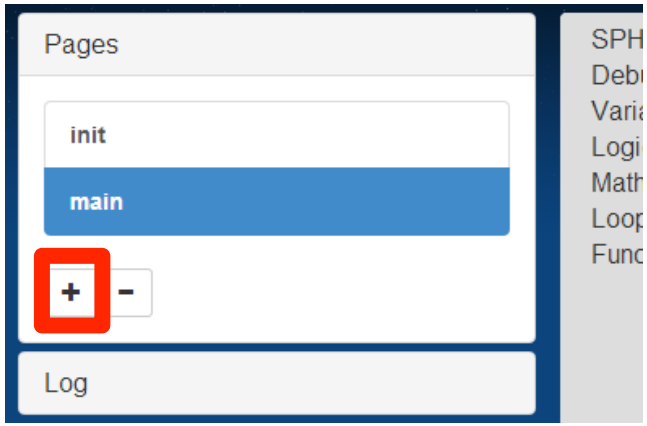
- Next we will create a separate function for the part of the program contained in section 2
- Since section 2 sends the SPHERE to positionA, we will name the function:

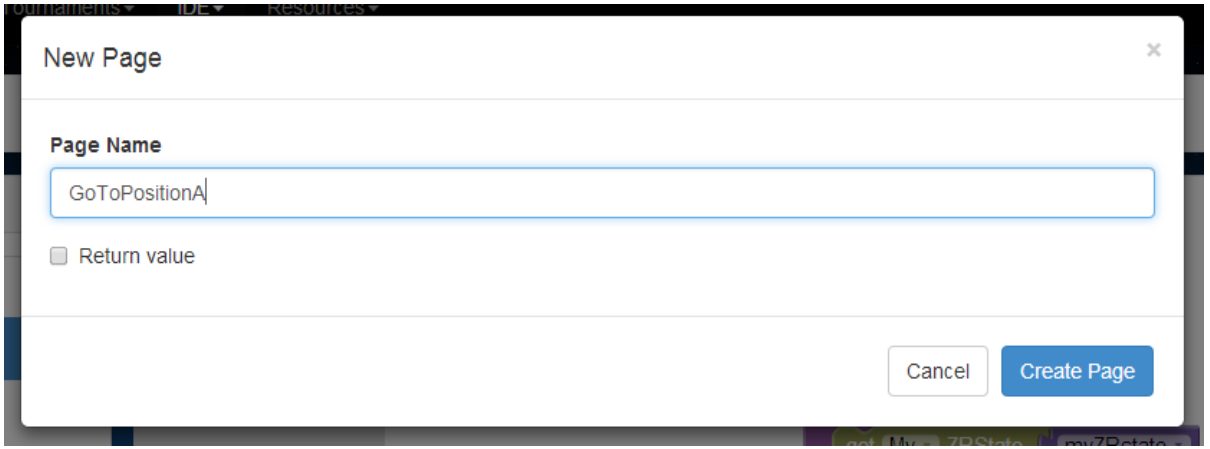
GoToPositionA



Creating a function

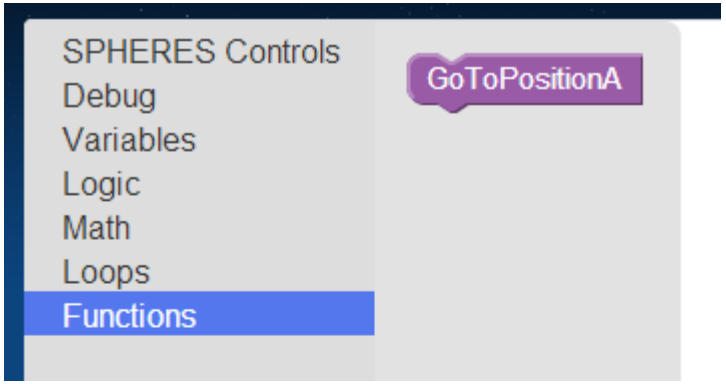
- Click on + (new page) button under Pages
  - For Page Name, type GoToPositionA (This will be the name of your function)
  - Leave the "Return value" box empty
  - Click the blue "Create Page" button
- Your new page will open to a loop called "GoToPositionA"

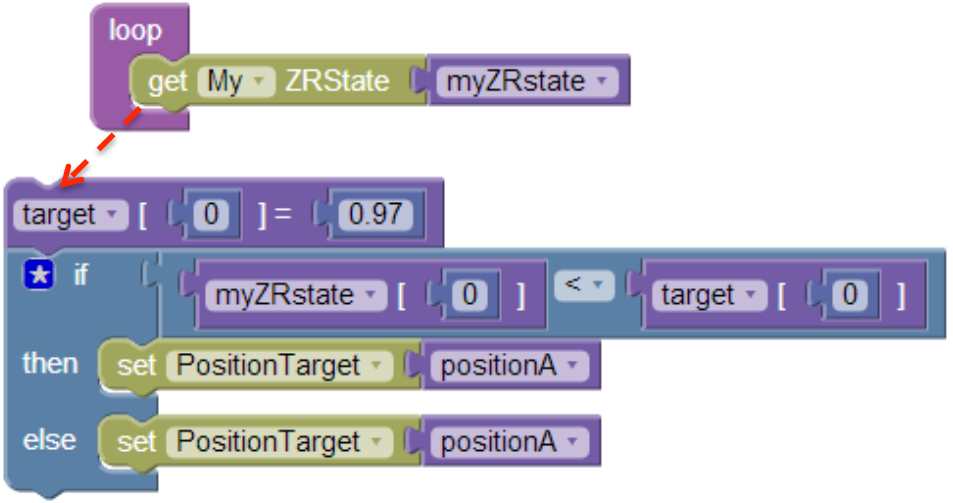




Creating a function (cont.)

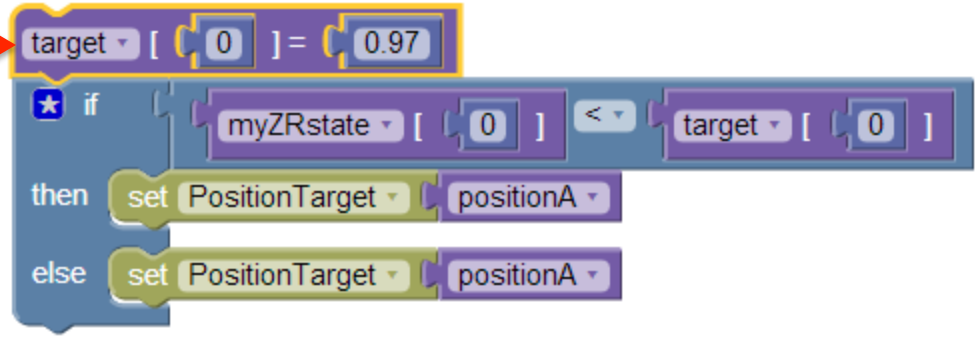
- Click on the "main" page to return to your main loop.
- On the loop page, open the User Function accordion (as shown)
  - You will see a block for your function: "GoToPositionA"
- Now you need to put code into your function (The code for your function will include everything in section 2 as shown earlier)
- Since the code is already written in the main loop you can copy it from the main loop, paste it into the function, and then delete it from the main loop as follows:
  - Click on the target[0]=0.97 block and drag everything out of your loop except for the getMyZRState block

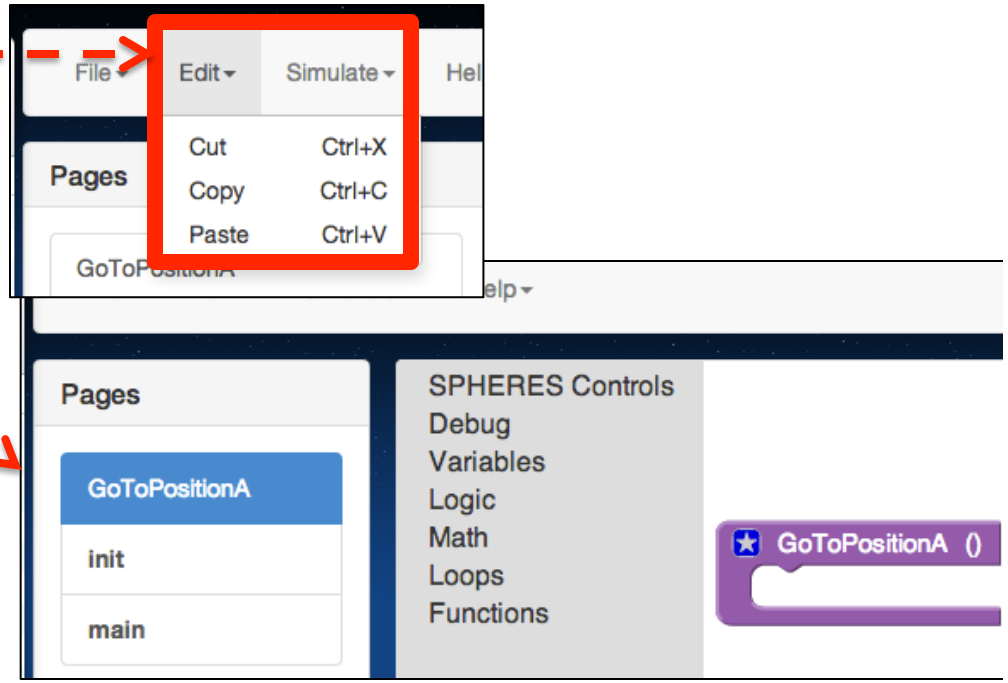




Creating a function (cont.)

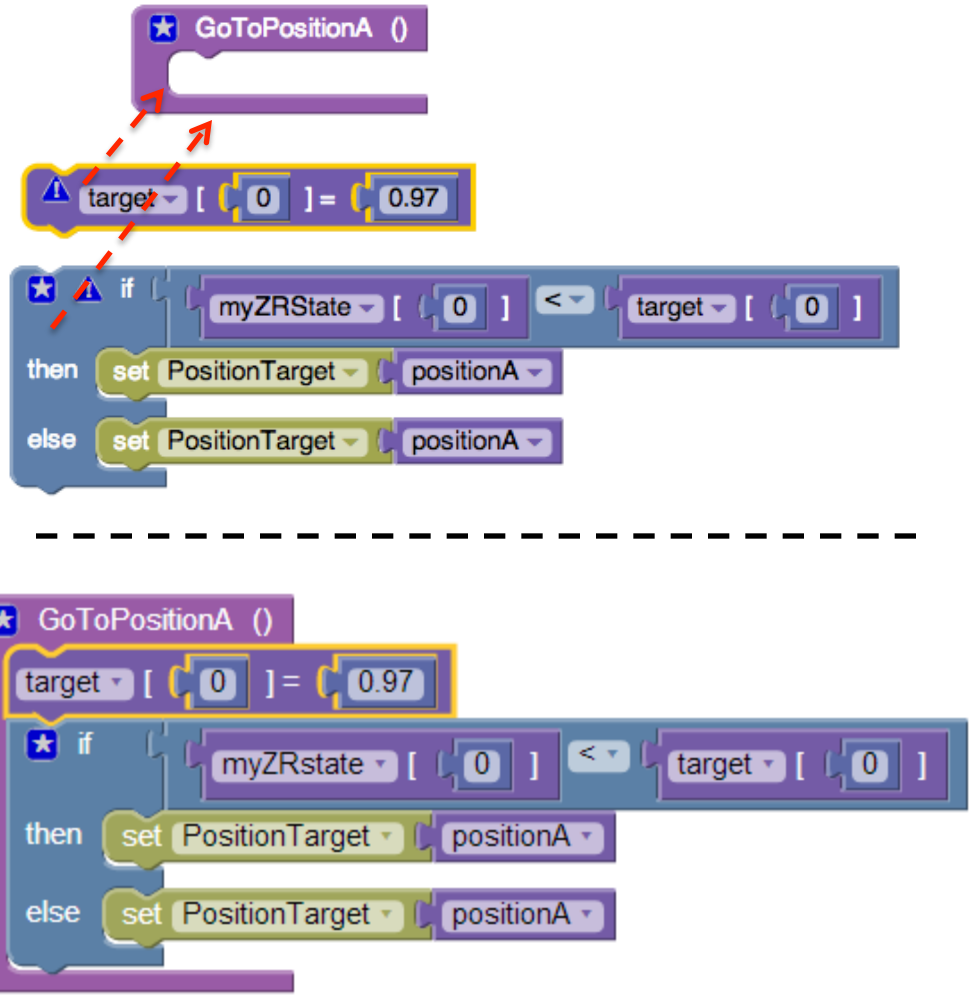
- Copy, cut and paste the **`target[0]= 0.97`** into the "GotopositionA" loop as follows:
  - Click on the **`target[0]= 0.97`** block
  - Go to the "Edit" menu and select "**Copy**", then select "**Cut**"
  - Open the "GoToPositionA" page on the Pages menu
  - Open the "Edit" menu and select "**Paste**" to paste the code onto the page
  - Return to the "main" page
- Copy, cut and paste the "**If-then-else**" block from the "main" page into the "GoToPositionA" page in the same way





Creating a function (cont.)

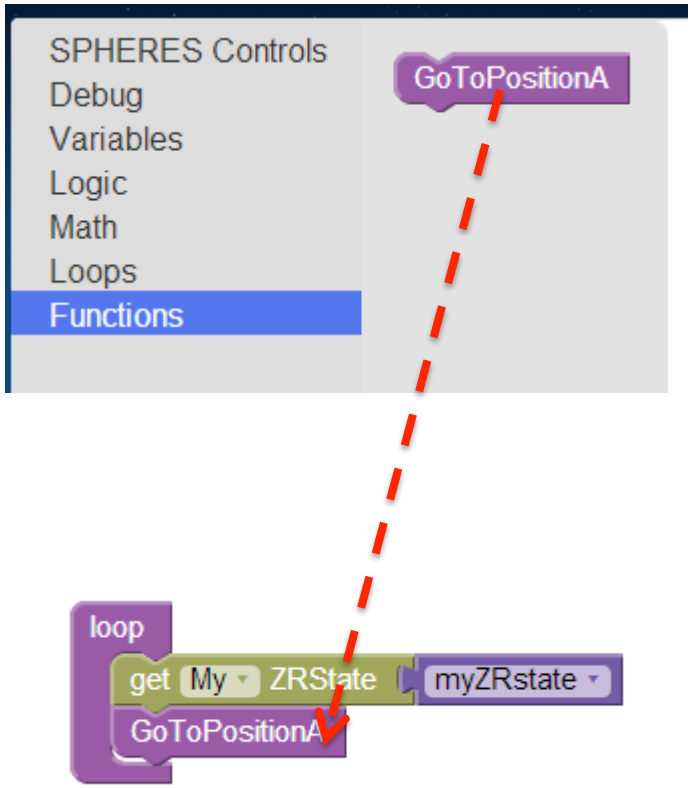
- The code you copied will appear on the “GoToPositionA” page
- Drag and drop the code into the “GoToPositionA” loop
- Click on the “main” page to return to the main loop (shown on next slide)



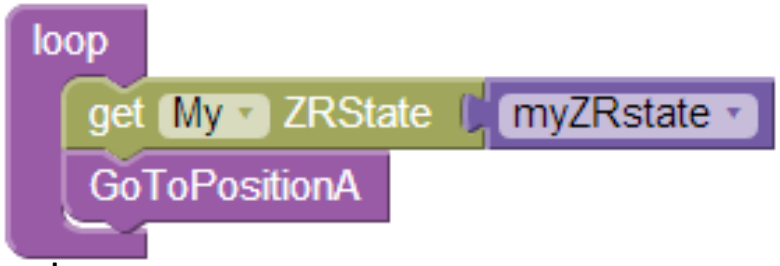
Creating a function (cont.)

Now you need to "call" your function from the main loop. To do this:

- Open the Functions accordion
- Drag the "GoToPositionA" block into the loop



View simulation



- Test your program!
- Compile, Simulate
  - Sets Maximum Time: 90 seconds
  - View simulation (the SPHERE should move to position A and stop there)
- Click “Back to Project”
- The C code for the pages: “main” and “GoToPositionA” is shown below:

```
1 void loop() {
2   api.getMyZRState(myZRstate);
3   GoToPositionA();
4 }
```

```
1 void GoToPositionA() {
2   target[0] = 0.97;
3   if (myZRstate[0] < target[0]) {
4     api.setPositionTarget(positionA);
5   } else {
6     api.setPositionTarget(positionA);
7   }
8 }
```

Review

Congratulations!

- You have learned how to create functions in your programs
- Continue to the next tutorial to see how creating functions can help you organize your program