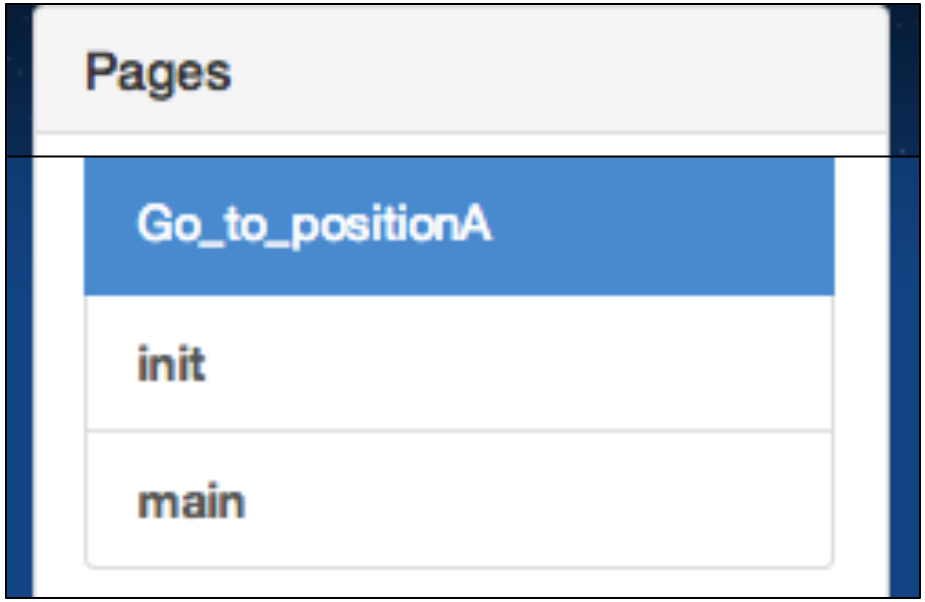
ZERO ROBOTICS

ISS PROGRAMING CHALLENGE

Creating Functions

Goals

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



Functions

- What is a Function?
  - Programmers find it convenient 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 piece of code called a *function*. This function can then be "called" and used in their program.
    - The void init() and void loop() sections in your programs are both functions called by the main SPHERES controller.
  - Creating functions:
    - Helps to organize the code and makes it easier to keep track of what is 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.
- The next tutorial, "Functions and the Step Counter Model" will demonstrate how functions can be used to simplify and organize a complicated program.

Create a new project using "Save As"

- 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** and select **Free Mode**
- Before we get started we will simplify this program so the function you create can also be used in the next tutorial
  - In the "else" statement, change the variable in the **`setPositionTarget`** block to "positionA" as shown.
  - We no longer need the variable "positionB," so we can remove it from the global variables section and void init().

```
void loop(){
  //This function is called once per second.
  //Use it to control the satellite.

  api.getMyZRState(myZRState);

  target[0] = 0.97;

  if (myZRState[0] < target[0]){
    api.setPositionTarget(positionA);
  }

  else{
    api.setPositionTarget(positionA);
  }
}
```

```
1  //Declare any variables shared between functions here
2  float myZRState[12];
3  float positionA[3];
4  float target[3];
5
6  void init(){
7    //This function is called once when your code is first loaded.
8
9    //IMPORTANT: make sure to set any variables that need an initial value.
10   //Do not assume variables will be set to 0 automatically!
11
12 positionA[0] = 1;
13 positionA[1] = 0;
14 positionA[2] = 0;
15
16 }
```

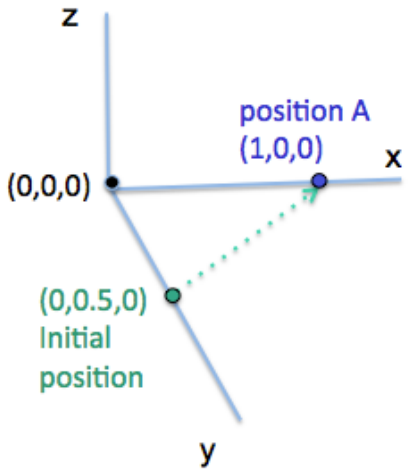
Create a new program

- 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
    - Telling the SPHERES to stay at positionA after it reaches its target
- 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:

  Go_to_positionA

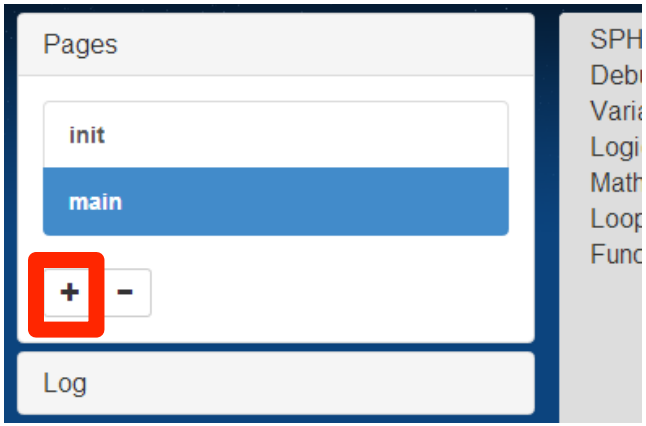
1
2

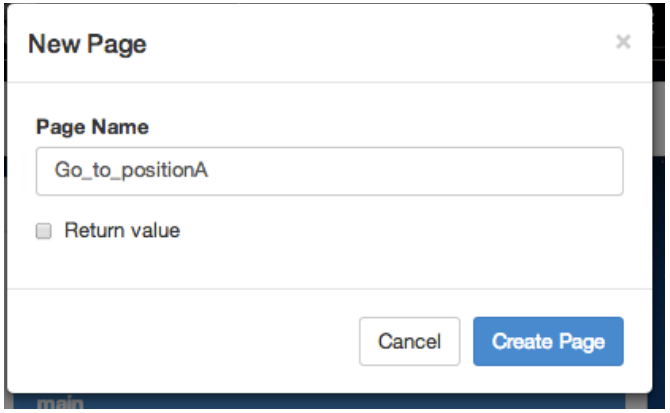
```
void loop(){
  //This function is called once per second.
  //Use it to control the satellite.

  api.getMyZRState(myZRState);

  target[0] = 0.97;

  if (myZRState[0] < target[0]){
    api.setPositionTarget(positionA);
  }


  else{
    api.setPositionTarget(positionA);
  }

}
```



Creating a function

- Click on + (new page) button under Pages
  - For Page Name, type Go_to_positionA (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 "Go_to_positionA"



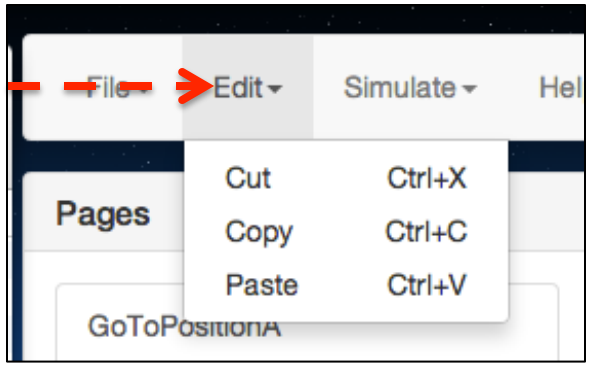



```
1- void Go_to_positionA() {
2  }
```

Creating a function (cont.)

- 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 page, and then delete it from the main loop as follows:
  - Highlight the text in void loop() starting from "target[0]=0.97" until the end of the "else" statement.
  - Once the text is highlighted, go to menu bar (at the top of the ZR IDE) and click the "Copy" icon, or right-click and select "Copy."

```
15 void loop(){
16   //This function is called once per second.
17   //Use it to control the satellite.
18
19   api.getMyZRState(myZRState);
20
21   target[0] = 0.97;
22
23   if (myZRState[0] < target[0]){
24     api.setPositionTarget(positionA);
25   }
26
27
28   else{
29     api.setPositionTarget(positionA);
30   }
31
32 }
33
```



Creating a function (cont.)

- Click to open the page **Go_to_positionA**

- On the **Go_to_positionA** page, we will need to declare the function. The typical structure of a function is shown at right:

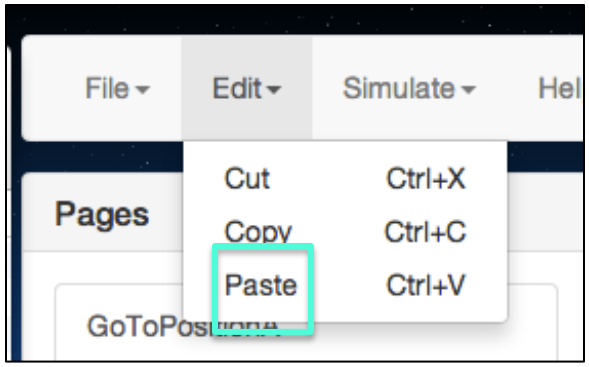
```
type functionName(optionalArguments) {

}
```

- Our function **Go_to_positionA** has no return type (**void**) and takes no arguments, so to initialize the function, your page should look like this:
  - If you don't know the terms "return type" and "arguments," don't worry about them for now.

```
1 void Go_to_positionA() {
2
3 }
```

Creating a function (cont.)

- Click inside the **Go_to_positionA** function and select "Paste"



- The code you copied will appear on the **Go_to_positionA** page as shown
- Make sure the code has been pasted into the **Go_to_positionA** function
- Click on **main** to return to the main loop page (shown on next slide)

```
1  void go_to_positionA(){
2
3    target[0] = 0.97;
4
5    if (myZRState[0] < target[0]){
6      api.setPositionTarget(positionA);
7    }
8
9    else{
10     api.setPositionTarget(positionA);
11   }
12
13 }
```

Creating a function (cont.)

- Delete the code that was previously dragged out of the loop. Your void loop() should look like this:

```
void loop(){
  //This function is called once per second.
  //Use it to control the satellite.

  api.getMyZRState(myZRState);

}
```

- Now you need to *call* (run) your function from the main loop. To do this:
  - Make sure you are in the main page.
  - At the end of void loop(), type **Go_to_positionA();**
  - Note that you don't have to type "api." in front of a function you made yourself.

```
void loop(){
  //This function is called once per second.
  //Use it to control the satellite.

  api.getMyZRState(myZRState);

  Go_to_positionA();
}
```

View simulation

- Compile, Simulate
  - Load settings: Tutorial _90
  - View simulation (the SPHERE should move to position A and stop there)
- The final C code for the pages **main** and **Go_to_positionA** is shown below:

```
1  float myZRState[12];
2  float positionA[3];
3  float target[3];
4
5  void init(){
6    postionA[0] = 1;
7    postionA[1] = 0;
8    postionA[2] = 0;
9  }
10
11 void loop(){
12   api.getMyZRState(myZRState);
13 Go_to_postionA();
14 }
```

```
1 void go_to_positionA(){
2   target[0] = 0.97;
3   if (myZRState[0]< target[0]){
4     api.setPositionTarget(positionA);
5   }
6   else{
7     api.setPositionTarget(postionA);
8   }
9 }
```

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