

# Functions with Arguments

**Slide    Topic**

- 2**    Custom Functions (Quick Recap)
- 3**    Functions with Arguments
- 4**    Writing Custom Functions with Arguments
- 5**    Writing Your Own Functions with Multiple Arguments
- 6**    Arguments That Change Over Time

# Custom Functions (Quick Recap)

KISS  
INSTITUTE FOR  
PRACTICAL  
ROBOTICS

Botball®

When  
you call  
this  
function,  
how long  
will it run  
for?

## Source Code

```
1 #include <kipr/wombat.h>
2
3 void drive_forward(); // function prototype
4
5 int main()
6 {
7     drive_forward(); // function call
8     return 0;
9 }
10
11 void drive_forward() // function definition
12 {
13     motor(0, 80);
14     motor(3, 80);
15     msleep(4000);
16     ao();
17 }
```

Now, what if you don't want it to run for this long each time?

# Functions with Arguments

## Source Code

```
1 #include <kipr/wombat.h>
2
3 void drive_forward(int milliseconds); // function prototype
4
5 int main()
6 {
7     drive_forward(4000); // function call
8     return 0;
9 } // end main
10
11 void drive_forward(int milliseconds) // function definition
12 {
13     motor(0, 80);
14     motor(3, 80);
15     msleep(milliseconds);
16     ao();
17 }
```

- **Function arguments:** values you will set when you call the function

# Writing Custom Functions with Arguments

KISS  
INSTITUTE FOR  
PRACTICAL  
ROBOTICS

Botball®

## Source Code

```
1 #include <kipr/wombat.h>
2
3 void drive_forward(int milliseconds); // function prototype
4
5 int main()
6 {
7     drive_forward(4000); // function call
8     return 0;
9 } // end main
10
11 void drive_forward(int milliseconds) // function definition
12 {
13     motor(0, 80);
14     motor(3, 80);
15     msleep(milliseconds);
16     ao();
17 }
```

The value in the **function call**  
sets the value of the argument...

... which is then used in the **function definition**.

# Writing Your Own Functions with Multiple Arguments

KISS  
INSTITUTE FOR  
PRACTICAL  
ROBOTICS

Botball®

## Source Code

```
1 #include <kipr/wombat.h>
2
3 void drive_forward(int power, int milliseconds); // function prototype
4
5 int main()
6 {
7     drive_forward(80, 4000); // function call
8     return 0;
9 } // end main
10
11 void drive_forward(int power, int milliseconds) // function definition
12 {
13     motor(0, power);
14     motor(3, power);
15     msleep(milliseconds);
16     ao();
17 }
18
19
```

The diagram illustrates the flow of arguments from the function call in the main() function to the function definition. Red arrows point from the 'power' argument in line 7 to the 'power' parameter in line 11, and from the 'milliseconds' argument in line 7 to the 'milliseconds' parameter in line 11.

The value in the **function call**  
sets the value of the **argument...**

... which is then used  
in the **function definition.**

# Arguments That Change Over Time

## Source Code

```
1 #include <kipr/wombat.h>
2
3 void drive_forward(int power, int milliseconds); // function prototype
4
5 int main()
6 {
7     drive_forward(80, 4000); // function call
8     drive_forward(75, 2000);
9     return 0;
10 } // end main
11
12 void drive_forward(int power, int milliseconds) // function definition
13 {
14     motor(0, power);
15     motor(3, power);
16     msleep(milliseconds);
17     ao();
18 }
19
```

The diagram illustrates the flow of arguments from the second function call in the main() function to the parameters in the function definition. Red arrows point from the value 75 in line 8 to the parameter power in line 12, and from the value 2000 in line 8 to the parameter milliseconds in line 12.

The values in the SECOND function call  
are now 75 and 2000 respectively

... which is then used  
in the function  
definition.