

Functions, Parameters and Local Variables

A supplemental lesson after Mission 9

Part 2



FIRIA LABS

Warm-up

Functions, parameters and local variables - Part 2



FIRIA LABS

Remember when ...



- Answer the warm-up questions on the assignment.



Review

Functions, parameters and local variables - Part 2



FIRIA LABS

Review: What is a function?

- **Function:** a named set of instructions that accomplishes a task

Reusable chunks of code

A function is a named chunk of code you can run anytime just by calling its name!

In other programming languages functions are sometimes called **procedures**. Functions can also be bundled with *objects*, where they're referred to as **methods**. Whatever you call them, they are a good way to package up useful sections of code you can use over and over again!



When to use a function?

- When you first started using functions, you identified places in your code that were repeated.
- You created a function for the repeated code
 - Gave it a name
 - Coded the function
 - Called the function
- You can also create a function for code that is similar (and duplicated) that just needs a parameter



Functions and parameters

- The first example does the exact same thing every time – changes all four pixel LEDs a random color
- By adding a parameter, you can call the function many times, but it can change the color from 1-4 pixel LEDs (flexibility)
- The information you give the function is called a **parameter**

```
def turn_on():  
    for lite in range(4):  
        color = random.choice(COLOR_LIST)  
        pixels.set(lite, color)
```

```
def turn_on(num_pixels):  
    for lite in range(num_pixels):  
        color = random.choice(COLOR_LIST)  
        pixels.set(lite, color)
```



Functions and parameters

- You use the parameter in the function code to complete the task

```
def turn_on(num_pixels):  
    for lite in range(num_pixels):  
        color = random.choice(COLOR_LIST)  
        pixels.set(lite, color)
```

- When you call a function with a parameter, you must give the value for the parameter
- This is called an argument

```
turn_on(3)
```



Functions and parameters

- An argument can be a literal value

```
turn_on(3)
number = 2
turn_on(number)
```

In the first call, the value passed to the parameter is 3, so pixels 0, 1, 2 will turn on.

In the second call, the value passed to the parameter is 2, so pixels 0 and 1 will turn on.

- An argument can be a variable
- The name of the variable does not have to be the same as the parameter
- The **value** of the variable is passed to the parameter



Functions with parameters

So how do you determine what is a parameter and what is a local variable?

Here are some standard rules for parameters:

- If a variable is used in a calculation
- If a variable is used in a condition (if statement)
- If a variable is used in a condition (loop)

Here are some standard rules for local variables:

- If the variable is being calculated
 - It may be returned at the end of the function
- If the variable is the counter in a loop



Functions, parameters and local variables

Activity #1
more than one parameter



FIRIA LABS

A. Functions with parameters



- Look at the code:
- You decide to make the circled code into a function.
- What would you call the function?
- What are the variables it needs?
- What are the parameters?
- What are the local variables?
- Does it need a return?

```
while True:
    # Start game with button B
    if buttons.was_pressed(BTN_B):
        # Reset the board for each game
        reset()
        # Select first random number
        num1 = random.randrange(6) + 1
        if num == 1:
            one_roll()
        elif num == 2:
            two_roll()
        elif num == 3:
            three_roll()
        elif num == 4:
            four_roll()
        elif num == 5:
            five_roll()
        else:
            six_roll()
        sleep(delay)
```



B. Functions with parameters



- Look at the code:
- You decide to make the code in the middle into a function.
- What would you call the function?
- What are the variables it needs?
- What are the parameters?
- What are the local variables?
- Does it need a return?

Note: lists are global. They do not need to be parameters or local variables.

```
if set_list == "a":
    my_image = a_list[choice]
else:
    my_image = b_list[choice]

if type(my_image) == tuple:
    display.fill(my_image)
else:
    display.show(my_image)

if buttons.was_pressed(BTN_R):
    choice = choice + 1
    if choice > LAST_INDEX:
        choice = 0
```



C. Functions with parameters



- Look at the code:
- You decide to make the code in the middle into a function.
- What would you call the function?
- What are the variables it needs?
- What are the parameters?
- What are the local variables?
- Does it need a return?

```
while True:
    num = random.randrange(sides) + 1
    if buttons.was_pressed(BTN_A):
        display.clear()
        display.draw_text("Rolling", scale=3, x=35, y=80)
        sleep(delay-0.7)
        display.draw_text("Rolling", scale=3, x=50, y=120)
        sleep(delay-0.4)
        display.clear()
        display.draw_text(str(num), scale=20, color=GREEN)
        sleep(delay)
        display.clear()
```



Functions, parameters and local variables

Activity #2

Different kinds of arguments



FIRIA LABS

Functions with parameters

As you can see, a function can have more than one parameter.
What about three or four parameters!

Sure! However, there is something you have to be careful about:
Arguments must be in the same order as parameters

Function definition:

```
def display_roll(num, delay):  
    if num == 1:  
        one_roll()  
    elif num == 2:
```

Function call:

```
display_roll(num, delay)
```



Functions with multiple parameters

More examples:

```
def see_image(set_list, choice):  
    if set_list == "a":  
        my_image = a_list[choice]  
    else:  
        my_image = b_list[choice]  
  
    if type(my_image) == tuple:  
        display.fill(my_image)  
    else:  
        display.show(my_image)  
  
see_image(set_list, choice)
```

```
def rolling_dice(delay, num):  
    display.clear()  
    display.draw_text("Rolling", scale=3,  
sleep(delay-0.7)  
display.draw_text("Rolling", scale=3,  
sleep(delay-0.4)  
display.clear()  
display.draw_text(str(num), scale=20,  
sleep(delay)  
display.clear()  
  
rolling_dice(delay, num)
```



Functions with parameters

You are comfortable with integers as variables, and even strings.
Any data type can be a variable:

```
message = "Click the A button"  
button = BTN_A  
the_image = pics.TSHIRT  
number = random.randrange(len(a_list))
```

And ... any variable can be used as a parameter and/or argument



Functions and parameters

- Parameters let you do more with less
- Look at the example from Mission 4 again
- This program has four functions:
 - option_A()
 - option_B()
 - option_L()
 - option_R()
- But they all do the same thing, with just a different button
- Can we create ONE function and use parameters?

```
def option_A():  
    display.show("Hold Button A")  
    sleep(1)  
    pressed = buttons.is_pressed(BTN_A)  
    if pressed:  
        pixels.set(0, GREEN)  
    else:  
        pixels.set(0, RED)  
    sleep(1)
```

```
def option_B():  
    display.show("Hold Button B")  
    sleep(1)  
    pressed = buttons.is_pressed(BTN_B)  
    if pressed:  
        pixels.set(1, GREEN)  
    else:  
        pixels.set(1, RED)  
    sleep(1)
```

```
def option_L():  
    display.show("Hold Button L")  
    sleep(1)  
    pressed = buttons.is_pressed(BTN_L)  
    if pressed:  
        pixels.set(2, GREEN)  
    else:  
        pixels.set(2, RED)  
    sleep(1)
```

```
def option_R():  
    display.show("Hold Button R")  
    sleep(1)  
    pressed = buttons.is_pressed(BTN_R)  
    if pressed:  
        pixels.set(3, GREEN)  
    else:  
        pixels.set(3, RED)  
    sleep(1)
```



Functions and parameters

- Take a look at one of the functions.
- What information is used that can be variables?
- Is it the same information in every function?

```
def option_R():  
    display.show("Hold Button R")  
    sleep(1)  
    pressed = buttons.is_pressed(BTN_R)  
    if pressed:  
        pixels.set(3, GREEN)  
    else:  
        pixels.set(3, RED)  
    sleep(1)
```



Example: Functions and parameters

- On your assignment document, answer the questions about this example
- In CodeSpace, either start a new program, or open the starter code for this program and Save As.
- Call the program “**Display2**”

```
def option_R():  
    display.show("Hold Button R")  
    sleep(1)  
    pressed = buttons.is_pressed(BTM_R)  
    if pressed:  
        pixels.set(3, GREEN)  
    else:  
        pixels.set(3, RED)  
    sleep(1)
```



Example: Functions and parameters

- Create a single function that can be used for any of the six buttons
- Call the function at least once (four times to play the game), passing arguments in the correct order

```
def option_R():  
    display.show("Hold Button R")  
    sleep(1)  
    pressed = buttons.is_pressed(BTN_R)  
    if pressed:  
        pixels.set(3, GREEN)  
    else:  
        pixels.set(3, RED)  
    sleep(1)
```

Hints:

- If the value of an argument doesn't change, like delay, you don't have to reassign it to a variable before you use it as an argument.
- Sometimes it is easiest to just use a literal value for the argument, like the pixel to turn on.



Wrap-up

Functions, parameters and local variables - Part 2



FIRIA LABS

When to use parameters and local variables?



- Answer the reflection questions on the assignment.

