Group: Usman Ahmed, Wayne Tobias, Ed Hawkins, Kirk Martin

Week 2 Day 1

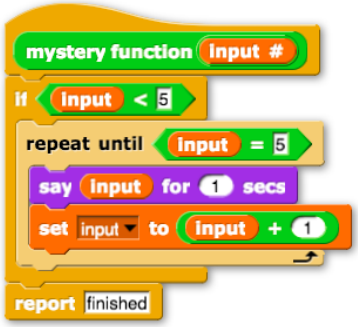
Aim: SWBAT create 2 projects that pass arguments into functions and call on those functions.

Standards:
9-12.CT.4 Implement a program using a combination of student-defined and third-party functions to organize the computation
9-12.CT.5 Modify a function or procedure in a program to perform its computation in a different way over the same inputs, while preserving the result of the overall program.
9-12.CT.9 Systematically test and refine programs using a range of test cases, based on anticipating common errors and user behavior.

Do Now:

What will the sprite say if you run the same function with the input 1?

- 1 2 3 4
- 5
- 4
- 1 2 3 4 5



Turn and Talk: Why would this be considered an abstraction.

Mini Lesson: In Snap Coding, whenever we created our own blocks, they were known as abstractions. But now they're known as functions.

What is the name of this function from the code written above? And what is being passed into it?

Answer: Function name is mystery function. The input of 1 is passed into it.

Notes:

A function is a block of code which only runs when it is called.

You can pass data, known as parameters, into a function.

A function can return data as a result.

Take a look at the image below. With your partner, answer the following two questions:

1) How do we create a function?
2) How do we call a function?

In Python a function is defined using the `def` keyword:

Example

```
def my_function():
  print("Hello from a function")
```

Calling a Function

To call a function, use the function name followed by parenthesis:

Example

```
def my_function():
  print("Hello from a function")

my_function()
```

Notes:

Parameters or Arguments?

The terms *parameter* and *argument* can be used for the same thing: information that are passed into a function.

From a function's perspective:

A parameter is the variable listed inside the parentheses in the function definition.

An argument is the value that is sent to the function when it is called.

Class Call Out:

What is the function name? What are the parameters? What are the arguments?

Example

```
def my_function(fname):
  print(fname + " Refsnes")

my_function("Emil")
my_function("Tobias")
my_function("Linus")
```

Example

This function expects 2 arguments, and gets 2 arguments:

```
def my_function(fname, lname):
  print(fname + " " + lname)

my_function("Emil", "Refsnes")
```

Notice we can have more than one parameter in our function.

Turn and Talk:
Why would this code return an error?

Example

This function expects 2 arguments, but gets only 1:

```
def my_function(fname, lname):
  print(fname + " " + lname)

my_function("Emil")
```

Activity:

1) Create a program that calls on a function named greeting that passes in parameters for your first name and prints out a greeting such as, “Hello Usman!”

2) Create a program that calls on a function named greeting that passes in parameters for your first name, last name and course and prints out a greeting such as “Hello Usman Ahmed from Computer Science Principles!”