Middle School Programming

Teacher Resource Documents

Developed by:
Kelly Strick
strickkr@gmail.com
Summer 2016

TABLE OF CONTENTS

1. INTRODUCTION

The materials contained herein would be appropriate to use with the Middle School Programming Syllabus. Any material within this document may be printed or edited.

2. STEM 1 ASSESSMENTS

The following assessments would be appropriate for use with the STEM 1 Scratch lessons. The materials are a work in progress and will be added to after the course is taught during the 2016-2017 school year.

2.1. SCRATCH FORMATIVE ASSESSMENT 1

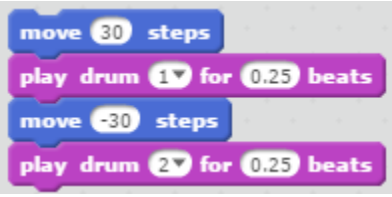
Name _____

1. What is a sprite?

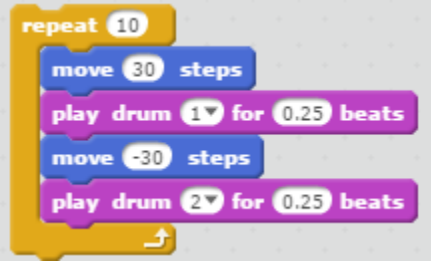
2. If you don't want the cat in your Scratch project, how can you get rid of it?

3. When drawing your own sprite, how are bitmap and vector mode different?

4. In Scratch, what would the following code do?



5. How would the following code differ from the code in #4 above?



2.2. SCRATCH FORMATIVE ASSESSMENT 2

Name _____

1. What is meant by costume?

2. If you sprite ends up upside-down when using the 

block, how can you fix it?

3. What would you have to do to this block to get your sprite to say “Go Bulldogs!”? (Be specific.)

4. Name or describe three different methods of getting your project to run.

5. Why would you use negative numbers in motion blocks?

3. STEM 2 ASSESSMENTS

The following formative assessments may be used to determine progress of students and to plan instruction with the STEM 2 syllabus. These documents are a work in progress and will be added to as the courses are taught during the 2016-2017 school year.

3.1. PYTHON FORMATIVE ASSESSMENT 1

Name _____

1. Write a line of code that will print "*Go Bulldogs!*".

2. How do you enter a comment in Python?

3. A) What do the following lines of code output?

   print(2 * 3)

   print(2 ** 3)

   B) Why do they give a different answer?

4. Why doesn't this code work?

   A = 22

   print(a)

5. Correct the following code. Write the correct code underneath the given code. (There may be more than one mistake.)

   Print("Hello. What is your name?)"

3.2. PYTHON FORMATIVE ASSESSMENT 2

Name _____

1. Why would this code not work?

   print(a)

   a = 45

2. What do the following lines of code output? Explain why the outputs are different.

   print(19 – 12)

   print(“19 – 12”)

3. What would the following line of code output?

   print(“This \nis \nmy \nsample.”)

4. Write a Python program to create a variable, assign a value to that variable, and print the value stored in that variable.

5. Circle the legal variable names in Python.

my_name	super bad	144xy
color	first number	variable_fun

3.3. PYTHON FORMATIVE ASSESSMENT 3

Name _____

1. Try this example Python code first:

   print("Don't is short for do not")

   Then, try this code:
   print('Isn't this going to work?')

   Did the second code work? Why not? What error message did it give you?

   Can you fix it? Compare it with the example that worked. Write your code below.

2. Try this example Python program:

   ```
   zombies = 5
   zombies = 10
   ghosts = 3
   ghosts = ghosts + 2
   bad_guys = zombies + ghosts
   print("There are", bad_guys, "bad guys.")
   ```

   When you ran this program, what result did it print? Explain the result.

3. Examine and try the following program:

   ```
   children = 30
   candy = 5
   total = children*candy
   print(total)
   ```

   The above program calculates the total number of pieces of candy a class of 30 children would need for each child to receive 5 pieces. How many total pieces of candy is required? _____

   Write a program to calculate the total number of brownies required for a class party if we know the number of children attending the party and the number of brownies each child should receive. Choose your own number of children and how many brownies each should get!

3.4. PYTHON SUMMATIVE ASSESSMENT

Name _____

1. Which of these is not a Python data type?

    a. int
    b. float
    c. iter
    d. str

2. How can we create a new line (move down) in a print function?

3. Which operator is used to compare two numbers for equality?

4. What is the difference between 12 + 13 and “12” + “13”?

5. Explain the difference between an integer, a float, and a string.

6. How do you enter a comment in a program?

7. What is the difference between == and =

8. Correct the following lines of code:

    a. (print”How are you?”)
    b. print(‘Go Team”)
    c. print(‘Kelly’s team is winning’)

4. PYTHON QUICK REFERENCE GUIDE

Adapted from iNTERFACEWARE

This Quick Reference Guide was created with the intent that students would have it readily available to refer to when programming. It is also a work in progress and could be added to as the course is taught during the 2016-2017 school year. Instructors may alter the document as necessary for their classes.

4.1. PRINT COMMAND

print displays values on the screen

Table 1 print Function Examples

Input	Output	Input	output
print("Hello")	Hello	print("2 + 3")	2 + 3
print(2 + 3)	5	print(Hello)	SyntaxError

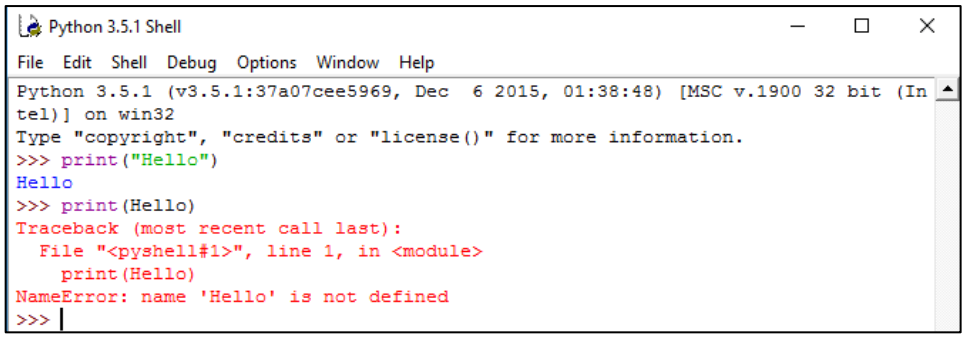


Figure 1 print Function Examples

4.2. ESCAPE CODES

Table 2 Escape Codes

Escape Code	Meaning	Escape Code	Meaning
\\	Backslash	\'	Single quote
\"	Double quote (useful in strings enclosed in double quotes)	\n	Line feed (move down)
\r	Carriage return (move to the left)	\t	Tab

4.3. CALCULATIONS AND VARIABLES

A **variable** can be any combination of letters, digits, and underscore characters. Variables in Python are ***case sensitive*** (variable and VARIABLE are not the same.) A variable cannot be a digit. Multi-word variable names should be separated by underscores (see more examples in Table 3).

Table 3 Variable Name Examples

Legal Variable Names	Illegal Variable Names	Legal, but not Proper Variable Names
first_name	first name	FirstName
distance	9ds	firstName
ds9	%correct	X

Python supports the standard arithmetic operations on integers and floating point numbers. Calculations follow the order of operations.

Table 4 Arithmetic Operators

Symbol	Meaning	Symbol	Meaning
+	Addition	*	Multiplication
-	Subtraction	/	Division
%	Remainder from division	**	Power

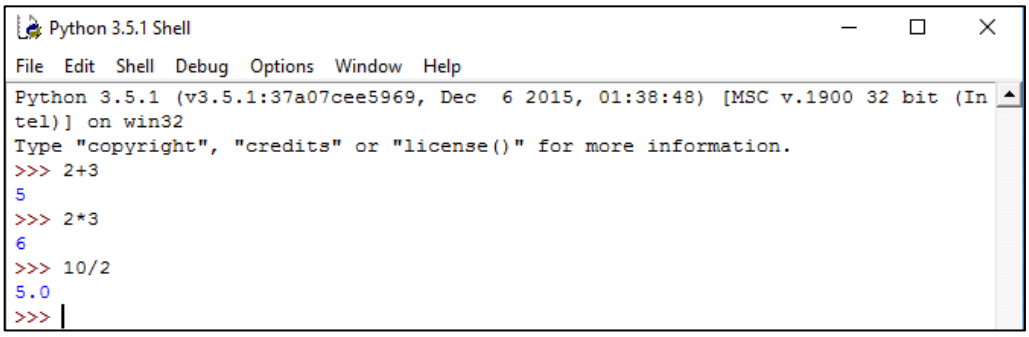


Figure 2 Arithmetic Operation Examples

Python also uses comparison operators.

Table 5 Comparison Operators

Symbol	Meaning	Symbol	Meaning
==	Equal to	<=	Less than or equal to
!=	Not equal to	>	Greater than
<	Less than	>=	Greater than or equal to

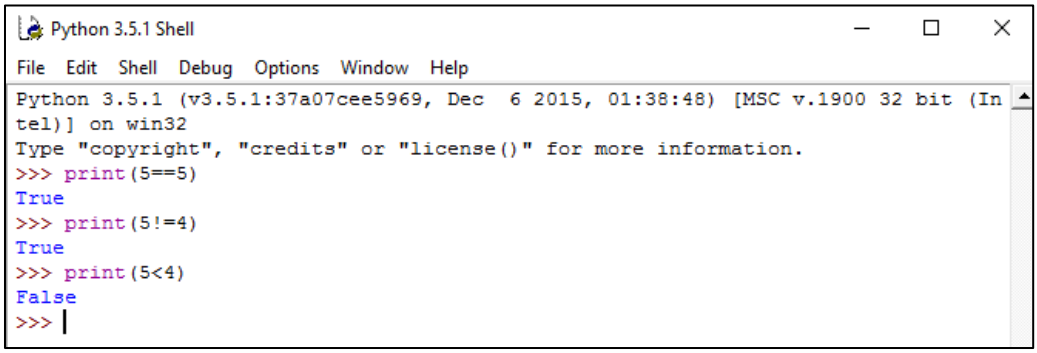


Figure 3 Comparison Operator Examples

4.4. DATA TYPES

Numbers can be *integers* (int) or *floating* point values (float).

42 3.14159

Strings (str) are enclosed in quotes, and can contain any printable character:

“test” “Hello, world!”

4.5. COMMENTS

Comments in Python are for the programmer or other humans to read. Everything after the # character is ignored.

```
#This is a comment.
print("Hello") #This is also a comment
```

Multi-line comments use three single quotes in a row.

```
'''
This is a
multi-line comment.
'''
```

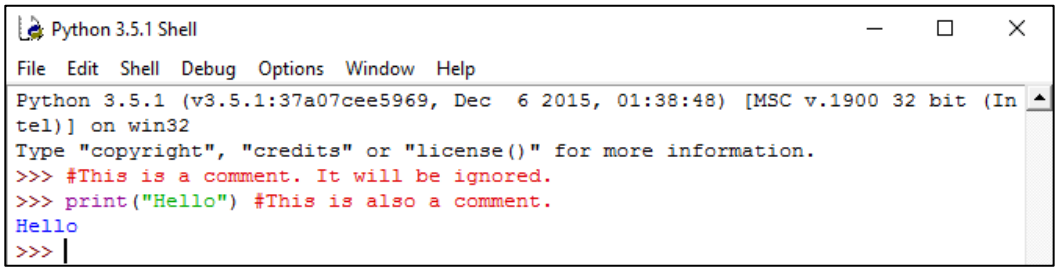


Figure 4 Comment Example

4.6. COLOR CODES

Table 6 Color Codes

Code	Color	Code	Color
Strings	Green	Comments	Red
Output	Blue	Reserved words	purple