Name:_____ **Partner:** _____

Python Activity 13: Range

What if we want to repeat an action a certain <u>number</u> of times?

Learning Objectives

Students will be able to:

Content:

- Describe the syntax of the **range()** function
- Identify the values in a **range()**
- Predict when to use **range()** versus iterate directly over objects

Process:

- Write code that uses use **range()** vs. direct iteration approriately

Prior Knowledge

- For-each..loops, input, types, * str, \t, list(), sequences

Concept Model: **Do not spend more than 2 minutes on CM 1 & 2!**

Examine the partially completed code below, and its desired output (to the right):

Concept Model – Python Program

```
height = int(input("How tall to make the triangle? "))

# Your code goes here
```

```
How tall to make
the triangle? 5
*
**
***
****
*****
```

CM1. Using *only concepts we've learned so far*, summarize the code we'd have to write to get the output shown to the right:

_____

_____

_____

CM2. What might some problems with your approach? (*Hint: Will your solution work for heights other than 5?)*

_____

_____

Critical Thinking Questions:

1. Closely examine the Python program below, which generates the output shown in the Concept Model for all [valid] user inputs:

Range -- Python Program

```
height = int(input("How tall to make the triangle? "))
for index in range(height):
    print('*' * (index+1))
```

   a. Circle the **loop variable** in the code.

b. What are the *values* that the loop variable must represent to produce the displayed output? _____

c. Underline the **sequence** that the loop is iterating over.

d. What might the `range(height)` command represent?

_____

FYI: A *looping structure* for which you know the number of times it will execute is known as a **count-controlled** loop.

2. Observe the following interactive python session:

```
>>> range(0, 10)
range(0, 10)
>>> type(range(0, 10))
range
>>> list(range(0, 10))
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> list(range(10))
```

a. What **type** of object is returned by a call to the built-in function **range()**? _____

b. What do we have to do to see the values that **range()** contains?

_____

c. What do you expect will happen when the last line of code is executed?

_____

3. Observe the following code snippets (assume `a_str` is set to empty string between each):

```
a_str = ''
a.    for use_better_names in range(5):
         a_str = a_str + str(use_better_names) + ' '
b.    for not_good in range(1,5):
         a_str = a_str + str(not_good) + ' '
c.    num_iterations = 6
      for b in range(num_iterations):
         a_str = a_str + str(b) + ' '
d.    num_iterations = 6
      for c in range(1, num_iterations+1):
         a_str = a_str + str(c) + ' '
e.    for a in range(3,20,2):
         a_str = a_str + str(a) + ' '
```

Which of the above questions a-e produce the following outputs, when `a_str` is printed?

`0 1 2 3 4 5`	a	b	c	d	e	(Circle one)
`0 1 2 3 4`	a	b	c	d	e	
`1 2 3 4`	a	b	c	d	e	
`3 5 7 9 11 13 15 17 19`	a	b	c	d	e	
`1 2 3 4 5 6`	a	b	c	d	e	

f. After examining the five code fragments in #3, explain how the **range()** built-in function works. Include an explanation of the argument values.

_____

_____

_____

_____

FYI: The Python built-in function - **range()** - is used to define a *sequence* of numbers and can be used in a for..loop to determine the number of times the loop is executed. The syntax is similar to sequence slicing: `range(start_num, end_num_exclusive, optional_step)`

4. a. Complete the arguments in the following range function so that the code prints the even numbers between 100 and 200 inclusive.

```
for x in range(_____________________________):
    print(x)
```

b. Complete the arguments in the following range function so that the code prints: 5 4 3 2 1 0.

```
for x in range(_____________________________):
    print(x)
```

5. Examine the following code segment:

```
total = 0
for x in range(5):
     number = int(input("Enter a number: "))
     total = total + number
print("The total is: ", total)
```

a. Why is the variable **`total`** initialized to 0 in the first line of code?

_____

b. Predict what the following code might do:

```
for x in range(5):
```

_____

c. Explain what the following code does: **`total = total + number`**

_____

d. How many numbers does the program prompt for? _____

e What is the **accumulator** in the code segment? _____

f. What does this code do?

_____

Application Questions: Use the Python Interpreter to check your work

1. Write a code segment using a for..loop that *prints* multiples of 5 from 5 to 500, one on a line.

2. It's possible to write a for-each..loop as a for..loop using range(). Convert the following for-each..loop into a for..loop that uses `range(..)`:

```
day = input("What day is today? ")
for d in day:
   print(d + "aturday")
```