

Name: _____

Partner: _____

Python Activity 12: Looping Structures – FOR Loops

How do we look at each element in a sequence?

Learning Objectives

Students will be able to:

Content:

- Motivate the use of loops
- Explain the syntax of a **for-each loop**
- Explain the use of **accumulation variables**.

Process:

- **Trace** through for..loops, identifying how local variables are modified
- Write code that includes a **for-each loop**.
- Write code that uses **accumulation variables**.

Prior Knowledge

- Lists, conditionals, input, expressions

Conceptual Model

Closely examine the Python program below that uses only concepts we've learned so far.

Python Program

```
# Counts the occurrences of a string in a given list
wthr_lst = ['sun', 'partial', 'partial', 'rain', 'rain']
print("List is:", wthr_lst)
item = input("What item to count? ")

count = 0
if wthr_lst[0] == item:
    count = count + 1
if wthr_lst[1] == item:
    count = count + 1
if wthr_lst[2] == item:
    count = count + 1
if wthr_lst[3] == item:
    count = count + 1
if wthr_lst[4] == item:
    count = count + 1
print("Occurrences:", count)
```

CM1. When this code runs, and the user enters 'rain', what might the output of the program be?

CM2. What would we have to do if we wanted to accomplish the same task for a longer list, perhaps:

```
['sun', 'partial', 'partial', 'rain', 'rain', 'rain', 'partial', 'partial', 'rain', 'sun', 'rain']?
```






CM3. Approximately how many lines of code would your solution require? _____

Critical Thinking Questions


1. Closely examine the Python program below.


```
Python Program  
  
# Counts the occurrences of a string in a given list  
wthr_lst = ['sun', 'partial', 'partial', 'rain', 'rain']  
print("List is:", wthr_lst)  
item = input("What item to count? ")  
  
count = 0  
for wthr in wthr_lst:  
    if wthr == item: # is our current element item?  
        count = count + 1  
  
print("Occurrences:", count)
```

-  a. Circle the new keyword we haven't seen before. How many lines of code is this example?
-  b. The output from this program will be identical to that of the previous example, except it will work for a list of *any length* as well. Why might that be?

-  c. What might the `for` keyword do?

FYI: A **looping structure** allows a block of code to be repeated one or more times. A **for** loop is one of the two looping structures available in Python. This particular example of `for..loop` is known as a **for-each loop**, as it **iterates** (or loops over) each of the items in a sequence (e.g., each character in a string).

-  d. How does the Python interpreter know what lines of code belong to the loop body? (or, why doesn't Python print "Occurrences" several times?)

-  e. **Every for..loop structure requires certain syntax.** Identify the part of code in the Python program that corresponds to each of these syntax requirements.

- A keyword that indicates the beginning of the `for..loop`: _____

- A **loop variable** that represents each element in the sequence as python loops:

- A sequence to loop over :

- An operator that indicates the test condition between the loop variable and the sequence:

- An operator that indicates the end of the `for..loop` definition: _____

FYI: An **accumulation variable** "accumulates" values in a loop structure.

KEY f. What is the accumulation variable in the above example? _____

2. Examine the following interactive python interaction:

```
pixels_favs = ['cheese', 'whipped cream', 'ice cream', 'jerky']  
for food in pixels_favs:  
    print(food)
```



a. What kind of loop is this? _____
b. Identify the part of code in the Python program that corresponds to each of the for..loop syntax requirements.

▪ A keyword that indicates the beginning of the for..loop: _____

▪ A **loop variable** that represents each element in the sequence as python loops:

▪ A sequence to loop over :

▪ An operator that indicates the test condition between the loop variable and the sequence:

c. Is there an accumulation variable in the above example? _____

KEY d. What does the loop variable `food` represent?

Where is `food` assigned its value(s)?

3. Examine the following code:

```
Python Program  
items = ['choc chip', 'nilla', 'oreo', 'oatmeal', 'shortbread']  
  
count = 0  
for it in items:  
    count = count + 1  
    print(count, it, "AH AH AH")
```

KEY

a. What is the output of this program? Trace through the values as they change:

Before the loop: items count it displayed/print()

Iteration 1: _____ _____ _____ _____

Iteration 2: _____ _____ _____ _____

Iteration 3: _____ _____ _____ _____

Iteration 4: _____ _____ _____ _____

Iteration 5: _____ _____ _____ _____

