

## 1. Lists

Look at the function below and test it out.

```
def sum_all(t):
    # t is a list of numbers
    total = 0
    for x in t:
        total += x
    return total
```

The function takes a list of numbers as a parameter (note how we specify, with a comment, what type of parameter to use), and returns the sum of all the numbers.

Note: a comment starts with # and does not get executed.

Now make a new function **cumulative\_sum** that modifies **sum\_all** so that instead of returning the sum of all the elements, it returns the cumulative sum; that is a new list where the *i*th element is the sum of the first *i* + 1 elements from the original list. For example, the cumulative sum of [1, 2, 3] is [1, 3, 6].

## 2. Strings

Write a function **pig\_latin** that lets the user enter in some English text, then converts the text to Pig-Latin. To review, Pig-Latin takes the first letter of a word, puts it at the end, and appends “ay”. The only exception is if the first letter is a vowel, in which case we keep it as it is and append “hay” to the end.

E.g. “hello” → “ellohay”, and “image” → “imagehay”.

It will be useful to define a list or tuple at the top of your code file called VOWELS. This way, you can check if a letter *x* is a vowel with the expression *x* in VOWELS.

## Understanding loops

For each of the following fragments of code, write what the output would be. Again, do this without running the code (although feel free to check yourself when you’re done).

1. num = 10

```
while num > 3:
    print num
    num = num - 1
```

2. divisor = 2

```
for i in range(0, 10, 2):
    print i/divisor
```

3. num = 10 while True:

```
if num < 7:
    break
print num
num -= 1
```

```
4. count = 0
   for letter in 'Snow!':
       print 'Letter #', count, 'is', letter
       count += 1
```

### **Buggy loop (aka Find The Bug!)**

Consider the following program (again, try to do this exercise without running the code in Jupiter):

```
n = 10
i = 10
while i > 0:
    print i
    if i % 2 == 0:
        i=i/2
    else:
        i=i+1
```

1. Draw a table that shows the value of the variables  $n$  and  $i$  during the execution of the program. Your table should contain two columns (one for each variable) and one row for each iteration. For each row in the table, write down the values of the variables as they would be at the line containing the print statement.

2. What is problematic about this program? Suggest one way to improve its behavior.