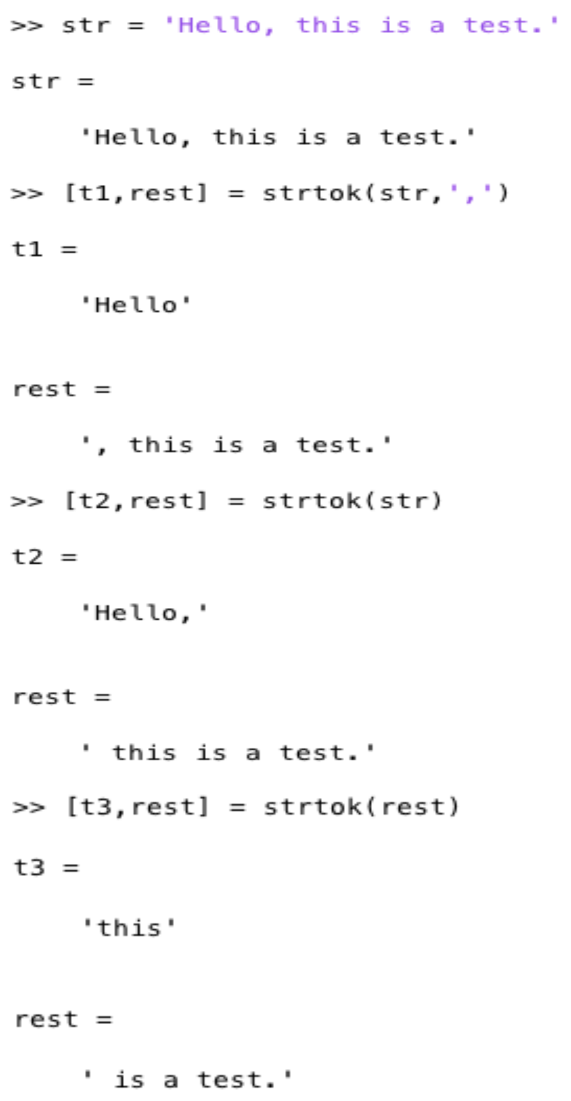
CS 1371
Spring 2020 Section A03
Week 2 Summary

1. **Vectors**
   a. A vector is a data structure, is a way of storing things
   b. All items in a vector need to be of the same data type (i.e. homogenous)
   c. Creating a vector
      i. Direct entry: --> vec = [1 2 3 4 5]
      ii. Colon operator:
         1. vec = start:step:end
         2. End is not guaranteed to be included in the vector
            a. vec1 = 1:2:4 --> vec1 = [1 3]
      iii. Functions
         1. vec = linspace(start,end,# of elements) -->evenly spaced
            a. vec2 = linspace(0,10,3) --> vec2 = [0 5 10]
         2. zeros()/ones()
            a. returns a vector of 0s and 1s
            b. Ex. ones(1,3) --> [1 1 1]
   d. Numerical Indexing
      i. A way of accessing values in a vector
      ii. Index --> position, spot
      iii. vec = [1 5 7 2] --> values
      iv. Index: 1 2 3 4 -->indices (plural of index)
      v. 2 ways of using indicies:
         1. Right hand side, access the VALUE
            a. x = vec(2) --> x is assigned the value 5
         2. Left hand side, select the SPOT
            a. vec(2) = 20 --> vec turns into [1 20 7 2]
      vi. We can also use vector to index, which will gives back multiple values
         1. vecM = vec(2:end) --> vecM becomes [5 7 2]
         2. vecM = vec([1 2 2 2]) --> vec M becomes [1 5 5 5]
      vii. To delete a value, index it on LEFT hand side and assign [] to it
         1. vec(3) = [] --> vec becomes [1 5 2]
      viii. To swap 2 values, use a temporary variable to store one of the two values
      ix. To insert a value in a spot, concatenate everything before the spot, the value that's going to be inserted, and everything after the spot
      x. length(vec) --> returns the number of elements in a vector
   e. Vector Operations
      i. Doubles
         1. Math works the same way, element by element

a. + - .* ./ .^ (the dot is IMPORTANT for vector math operations
b. Functions
   - i. sum(vec), prod(vec), mean(vec), max(vec),min(vec)
   - ii. mod(a,b) -->gives back the remainder of a/b, think of the “mod clock” we talked in recitation

2. We can use logical operators too
   - a. Elements being compared should have same dimensions
   - b. The only exception is when you compare a scalar to a vector
     - i. vec > 2 returns you a vector of logicals of the same length as vec

ii. Chars
1. A vector of chars is called a string
2. “String” is NOT a data type
3. Functions:
   - a. strcmp(str1, str2) -->compares 2 strings, returns a logical
   - b. strcmpi(str1, str2) --> same thing but case insensitive
   - c. upper(), lower() --> makes everything upper/lower case
   - d. strfind(str, pattern)
     - i. Looks for a pattern in the string
     - ii. Returns a vector of all the starting INDICIES where the pattern occurs
       1. Ex. str = ‘1371371371’
       2. f1 = strfind(str,’3’) --> returns [2 5 8]
       3. f2 = strfind(str,’1371’) --> returns [1 4 7]
   - e. [token,rest] = strtok(str,delim)
     - i. Looks for the delimiter in the string
     - ii. token becomes everything that comes before the delimiter
     - iii. Rest is the delimiter and everything after
     - iv. If str begins with one or more delimiter, those are “automatically deleted”/ignored



```
>> str = 'Hello, this is a test.'

str =

    'Hello, this is a test.'

>> [t1,rest] = strtok(str,',')

t1 =

    'Hello'


rest =

    ', this is a test.'

>> [t2,rest] = strtok(str)

t2 =

    'Hello,'


rest =

    ' this is a test.'

>> [t3,rest] = strtok(rest)

t3 =

    'this'


rest =

    ' is a test.'
```

2. **Casting**
    a. Casting = changing the data type of a variable
    b. Number as strings
        i. Chars are understood as number in MATLAB (reference to ASCII value)
        ii. ‘12’ and 12 are DIFFERENT
            1. ‘12’ is a string of length 2
            2. 12 is a double of length 1
        iii. char()
            1. Takes in a number or vector, gives back the char equivalent with respect to the ASCII table
            2. Ex. char([97,98,99])   --> ‘abc’
        iv. double()
            1. Takes in a char or string, gives back a double or vector of doubles equivalent with respect to the ASCII table
            2. Ex. double(‘abc’)   --> [97,98,99] [length 3 stays length 3]
            3. Ex. double(‘123’)   --> [49,50,51] [length 3 stays length 3]

v. num2str()

1. Ex. num2str(123) --> ‘123’ [length 1 becomes length 3]

vi. str2num()

1. Ex. str2num(‘123’) --> 123 [length 3 becomes length 1]