CS005 Introduction to Programming: Matlab

Eamonn Keogh
eamonn@cs.ucr.edu

Matlab Strings

```
EDU>> FirstName = 'eamonn'

FirstName =

eamonn
```

Here we are assigning a literal string to a variable, note the single quotes

```
EDU>> FirstName = HisAge
```

Contrast with this. Here we are assigning a variable to another variable, note the lack of single quotes

A Key Insight: String are Special Arrays

- This means that (almost) all the tricks we know about manipulating array work for strings.

```
EDU>> FirstName = 'eamonn'
FirstName =
eamonn
```

FirstName

3	inf	e	a	m	o	n	n	23	23

Displaying Strings

```
EDU>> FirstName = 'eamonn'
FirstName =
eamonn

EDU>> disp(FirstName)
eamonn
```

Getting String Length

```
EDU>> FirstName = 'eamonn' ;

EDU>> length(FirstName)
ans =
     6
```

```
FirstName = 'eamonn' ;

disp('there are ')
disp(length(FirstName))
disp(' letters in the name ')
disp(FirstName)
```

```
there are
     6
 letters in the name
eamonn
```

Indexing Strings

```
EDU>> FirstName = 'eamonn' ;
EDU>> FirstName(1)
ans =
e
EDU>> FirstName(2)
ans =
a
EDU>> FirstName(end)
ans =
n
EDU>>
```

```
EDU>> FirstName = 'eamonn' ;
EDU>> FirstName(1:3)
ans =
eam
EDU>> FirstName(3:5)
ans =
mon
```

```
FirstName = 'eamonn' ;

disp('Your initial is ')
disp(FirstName(1))
```

```
Your initial is
e
```

```
FirstName = 'eamonn' ;
```

We can change individual letters

```
EDU>> FirstName(1) = 'r'
FirstName =
ramonn
```

```
EDU>> FirstName = 'eamonn'
FirstName =
eamonn
EDU>> FirstName(end) = []
FirstName =
eamon
```

To delete a letter in a string, assign it the empty set

Review: Recall we can concatenate arrays

```
EDU>> X = [ [1 2 3] , [7 8 9] ]
X =
    1    2    3    7    8    9

EDU>> Boys = [2 3 4];
EDU>> Girls = [3 2 1];
EDU>> Kids = [Boys  Girls]
Kids =
    2    3    4    3    2    1
```

```
EDU>> FirstName = 'augustus'
FirstName =
augustus
EDU>> LastName = 'fink-nottle'
LastName =
fink-nottle

EDU>> disp(FirstName) , disp(LastName)
augustus
fink-nottle
```

```
EDU>> FirstName = 'augustus'
FirstName =
augustus
EDU>> LastName = 'fink-nottle'
LastName =
fink-nottle

EDU>> FullName = [ FirstName , LastName  ]
FullName =
augustusfink-nottle
```

```
EDU>> FullName = [ 'eamonn' , 'keogh'  ]
FullName =
eamonnkeogh

EDU>> FirstName = 'augustus'
FirstName =
augustus

FullName = [FirstName , ' caesar'  ]
FullName =
augustuscaesar
```

```
EDU>> FirstName = 'augustus' ;
EDU>> LastName = 'fink-nottle' ;
EDU>> FullName = [ FirstName , ' ', LastName  ]
FullName =
augustus fink-nottle

EDU>> FullName = [ 'eamonn' ,  ' '  , 'keogh'  ]
FullName =
eamonn keogh
```

```
EDU>> FirstName = 'eamonn' ;
EDU>> LastName = ‘keogh' ;
EDU>> MiddleName = 'john' ;
EDU>> disp([FirstName, ' ', MiddleName, ' ', LastName  ])
eamonn john keogh


EDU>> disp(['Dr. ', FirstName, ' ', LastName  ])
Dr. eamonn keogh
```

```
EDU>> FirstName = 'eamonn' ;
EDU>> LastName = ‘keogh' ;
EDU>> MiddleName = 'john' ;

% For some records we list, last name, then a comma,
   followed by the first name, then a comma,  then the initial
   of the middle name, followed by a period.

EDU>> disp([LastName,', ', FirstName, ', ' ,MiddleName(1),'.'  ])
keogh, eamonn, j.
```

Built in functions for Strings

```
EDU>> BookName = 'of mice and men'
BookName =
of mice and men

EDU>> BookName = upper( 'of mice and men')
BookName =
OF MICE AND MEN
```

The function upper takes in a string and returns an upper case version of it

```
EDU>> BookName = upper( '@ Swim 2 Birds')
BookName =
@ SWIM 2 BIRDS
```

The function upper ignores everything except a,b,c..x,y,z

```
EDU>> FirstName = 'eamonn' ;
EDU>> disp(FirstName)
eamonn

EDU>> FirstName(1) = upper( FirstName(1) )
FirstName =
Eamonn
```

This lets us format names correctly…

FirstName

3	inf	e	a	m	o	n	n	23	23

3	inf	E	a	m	o	n	n	23	2

```
EDU>> lower(' A b C d E +-&^%$#@!')
ans =
 a b c d e +-&^%$#@!
```

The function lower does exactly what you expect

```
EDU>> PrettyFormatProfName('eamonn','keogh')
ans =
Dr. Eamonn Keogh.

EDU>> PrettyFormatProfName('eAmonn','KEOGH')
ans =
Dr. Eamonn Keogh.

EDU>> PrettyFormatProfName('eAmoNN','kEOgH')
ans =
Dr. Eamonn Keogh.
```

Let us write a function that takes in the first and second name of a prof, and returns a formatted string with his title

```
function FormattedNameString = PrettyFormatProfName(FirstName,LastName )

FirstName      = lower(FirstName);
LastName       = lower(LastName);
FirstName(1) = upper(FirstName(1));
LastName(1)  = upper(LastName(1));

Title = 'Dr.';

FormattedNameString =[Title,' ',FirstName,' ',LastName,'.'];
```

```
EDU>> PrettyFormatProfName('eAmonn','KEOGH')
ans =
Dr. Eamonn Keogh.
```

```
EDU>> PrettyFormatName('Eamonn','Keogh',1)
ans =
Mr. Eamonn Keogh.

EDU>> PrettyFormatName('Peggy','Keogh',0)
ans =
Ms. Peggy Keogh.
```

Let us write a function that takes in the first and second name of a person, together with their sex, and returns a formatted string

```
function FormattedName = PrettyFormatName(FirstName,LastName,IsMale)

if IsMale
   FormattedName = ['Mr. ',FirstName,' ',LastName, '. '];
else
   FormattedName = ['Ms. ',FirstName,' ',LastName, '. '];
end
```

```
EDU>>
PrettyFormatName('Peggy','Keogh',0)
ans =
Ms. Peggy Keogh.
```

```
EDU>> FirstName = 'eamonn';
EDU>> LastName = 'keogh';
EDU>> PrettyFormatName(FirstName,LastName,1)
ans =
Mr. eamonn keogh
EDU>>
```

Getting Strings

```
EDU>> input('Enter your age : ')
Enter your age : 17

ans =
   17

EDU>> input('Enter your name : ','s')
Enter your name : Joe

ans =
Joe
```

When we a getting a string, we need the string flag 's'

Lab 6: Part I, Write a function DoStuff

As always, don't try to write the whole function at once, get something simple working, and build up to the full function. Note how the *case* problems need to be fixed

```
EDU>> FNameM = 'eamonn';
EDU>> LNameM = 'KEOGH';
EDU>> FNameF = 'michelle';
EDU>> LNameF = 'yEOh';
EDU>> MHasPHD = 0;
EDU>> DoStuff(FNameM, LNameM, FNameF, LNameF, MHasPHD );
```

When Michele Yeoh marries Mr. Eamonn Keogh she will become Mrs. Michele Keogh

```
EDU>> FNameM = ‘KERMIT';
EDU>> LNameM = ‘FROG';
EDU>> FNameF = ‘piGGY';
EDU>> LNameF = ‘lee';
EDU>> MHasPHD = 1;
EDU>> DoStuff(FNameM, LNameM, FNameF, LNameF, MHasPHD );
```

When Piggy Lee marries Dr. Kermit Frog she will become Mrs. Piggy Frog

Lab 6: Part 2, Write Function PluralWife

```
function WifeCount = PluralWife(FirstName,LastName)

WifeCount = 0;
AllowLoop = 1;

while AllowLoop >= 0
   resp      = input('Enter 1 to add wife, -1 to quit : ');
   if resp == 1
     FemaleName = input('Enter your name ','s');
     disp(['You are now Mrs. ', FemaleName,' ', LastName]);
     WifeCount = WifeCount + 1;
   else
      AllowLoop = -999;
   end
end
```

```
EDU>> PluralWife(FirstName,LastName)
Enter 1 to add wife, -1 to quit : 1
Enter your name Sue
You are now Mrs. Sue keogh
Enter 1 to add wife, -1 to quit : 1
Enter your name Joan
You are now Mrs. Joan keogh
Enter 1 to add wife, -1 to quit : 1
Enter your name Lisa
You are now Mrs. Lisa keogh
Enter 1 to add wife, -1 to quit : -1
ans =
    3
EDU>>
```

Continue to next slide for the next step

PluralWife

Fix the code..

Remove unused Firstname variable.

Make sure that no matter what name is passed in with the variable LastName, or typed by the ladies, it is written with lowercase, except the first letter which will be uppercase.

Add the summary statement at the end (for now, it will have to be written with 3 disp function calls)

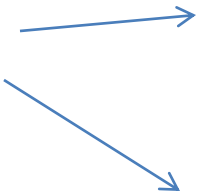
```
EDU>> LastName = 'kEoGH'
EDU>> PluralWife(LastName)

Enter 1 to add wife, -1 to quit : 1
Enter your name sue
You are now Mrs. Sue Keogh

Enter 1 to add wife, -1 to quit : 1
Enter your name JOAN
You are now Mrs. Joan Keogh

Enter 1 to add wife, -1 to quit : 1
Enter your name LisA
You are now Mrs. Lisa Keogh

Enter 1 to add wife, -1 to quit : -1

Keogh has
3
wives
```

Lab 6: Part 3, Write a Conversion Script

Let us write a script that converts numbers from kilometers to miles, while the user wants to continue..

We need a way for the user to signal that he/she is finished, let us use a negative number to signal this…

(for now, it will have to be written with 4 disp function calls)



```
EDU>> kilo2miles
I will convert kilometers to miles for you.
Enter number to convert or negative number to quit : 3
    3
 kilometers is
 1.864
 miles
Enter number to convert or negative number to quit : -1
Thank you for using my program
```