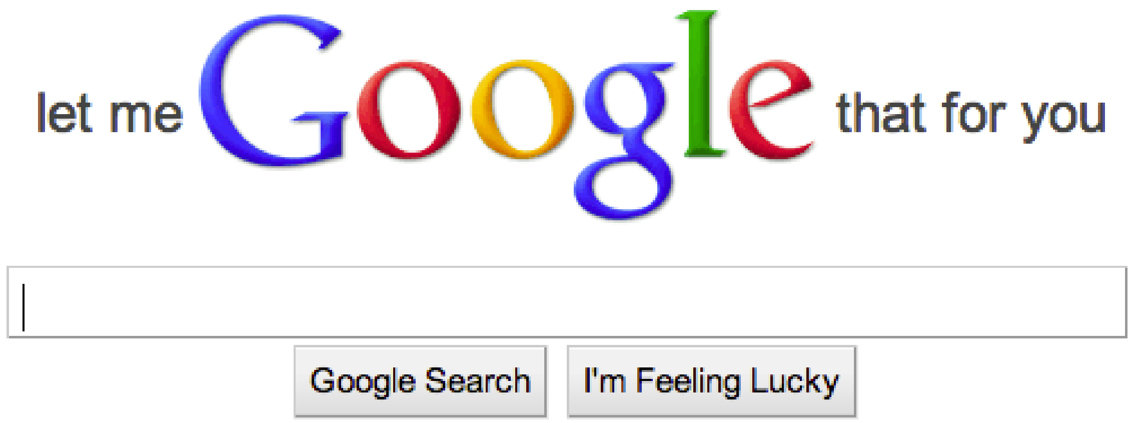
MATLAB BASICS

Amy Johnson

amy.johnson@mrc-cbu.cam.ac.uk

MRC CBU 2016

Some good news



Focus on the concepts, not the details… and google everything else

Some more good news

Script basics

A script is a list of commands that are executed almost as if you were typing them into the command window, line by line

Action:

- Open a new script
- create a variable, x, which is a list of 5 numbers
- save it as matlab_basics
- Run script ▶

Script basics

- '%' For bits you don't want to be run (titles, notes etc)

Add me

Top Tip Use these liberally!

- ';' To stop line printing in command window

Add me

- save('filename','variables')

```
save('test.mat', 'x')
```

Add me

- F5-run script, F9-run highlighted bit

Run save

Comparison operators

- Operators that tell us how two variables relate

Type 2>3 and run

- 1= true, 0 = false

- Can run on lists, 2D data and… any dimension of data

Type and run:
a = randi(100, 10)
a>= 50

Operator	Meaning
==	Is equal to
~=	Is not equal to
<	Is less than
<=	Is less than or equal to
>	Is greater than
>=	Is greater than or equal to

Combining Operators

Operator	Meaning
~	NOT/OPPOSITE
&	AND (need true AND true)
|	OR (need true OR true)

==	Is equal to
~=	Is not equal to
<	Is less than
<=	Is less than or equal to
>	Is greater than
>=	Is greater than or equal to

y = 5

y > 3| y ~= 5

True or False

= 1

What would be the answer to:

x = 8

y = 9

~(~(x < 3))&~(y >14 | y>10)

Conditional statements

- Comparison operators <, >, <=, >=, ==, ~=
- Combining operators and (&), or (|) and not(~)
- Conditional statements:
  - if, elseif, else

If

```
if this is true
     %Do whatever is in the middle
elseif this is true
     %Do whatever is in the middle
else
     %Do whatever is in the middle if
     neither above are true
end
```

If

```
a = 33;

if a < 30
    disp('small')
elseif a < 80
    disp('medium')
else
    disp('large')
end
```

Create an If statement

- X = 10, minVal = 2, maxVal = 6
- Write a script to print out (using ‘disp’):

a) ‘Value within range’ if x is within or equal to the range parameters

b) ‘Value exceeds maximum value’ if it’s larger than maxVal

c) ‘Value is below minimum value’ if it’s smaller than minVal

d) Test different x to check it’s working

Answer

```matlab
x = 10;
minVal = 2;
maxVal = 6;

if (x >= minVal) & (x <= maxVal)
    disp('Value within specified range.')
elseif (x > maxVal)
    disp('Value exceeds maximum value.')
else
    disp('Value is below minimum value.')
end
```

Repetitions: For loops

```
%General structure:
for index = values
     %Do whatever is in the middle
end
```

```
%Example:
data = [1 : 100];
n = length(data);
result = 0;
for k = 1 : n
result = result + data(k);
end
result_2 = result/n
```

Top Tip

Use variable names that describe what it is

Create a for loop

- Define an array with 5 numbers between 0 to 10 as you like. Each number represents the score of a subject in a test.
- For each subject, apply a correcting factor on the grades. Create a new variable which will contain the revised grades. The factor should be: x = x*1.2.
- If the revised grade is larger than 10, set it to 10.
- In the workspace, make sure you can see the two variables and that their values make sense.

Answer

```
score = [1, 5, 7, 9, 8];
n = length(score);
for ind = 1:n
    revised_score(ind) = score(ind)*1.2;
        if revised_score(ind) > 10
            revised_score(ind) = 10;
        end
end
```

Initialize arrays rather than growing with each loop
E.g. use revised_scores= zeros(size(score))

Functions

- You can run a script from the command line or from another script

- Put your for loop in a new script and save as my_for_loop
- Run your script by typing my_for_loop into the command window

- Want more flexibility? Functions…
- Like a script but you pass input values and return output values

Functions

```
function [outputs] = function_name(inputs)

%Put your script in here

end
```

Save the script as ‘function_name’

Create a function

- Want to revise score with any given factor (variable called ‘correct_factor’), not just *1.2
- Turn your for loop script into a function that takes inputs: ‘scores’ and ‘correct_factor’ and gives the revised scores as an output
- Run from the command line with a few different inputs to test