



## Extra syntax notes

- Semicolons suppress output of computations:

```
> a=4+5
a =
    9
> b=6+7;
>
```

- % starts a comment for the line (like // in C++)

- .\*, ./, .^ performs element-wise arithmetic

```
> c=[2 3 4]./[2 1 2]
> c =
    [1  3  1]
>
```

5

## Variables

- who, whos – list variables in environment

- Comparisons:

- Like C++: ==, <, >, <=, >=
- Not like C++: not ~, and &, or |

- Conditions:

- if(...), end;

- Loops:

- while(...), end;
- for x=a:b, end;

6

## Data: .mat files

- **save** filename variableNames

- **load** filename

- Confirm correct directories:

- pwd – show directory (**p**rint **w**orking **d**irectory)
- cd – change **d**irectory
- ls – **l**ist files in directory

```
import scipy.io
varIn = scipy.io.loadmat('file.mat')
X=varIn['matVarX']
```

7

## Define new functions: .m files

- Begin file with function header:

```
function output = function_name(input)
```

```
statement1;
```

```
statement2;
```

```
⋮
```

```
def func_name(input):
    statement1
    statement2
    ...
    return output
```

- Can allow multiple inputs/outputs

```
function [output1, output2] = function_name(input1, input2, input3)
```

8