Introduction to MATLAB

Week 2 Lecture 3

Polina Cherepanova:
cherepanova@princeton.edu
Polina Iamschinina:
pi7127@princeton.edu

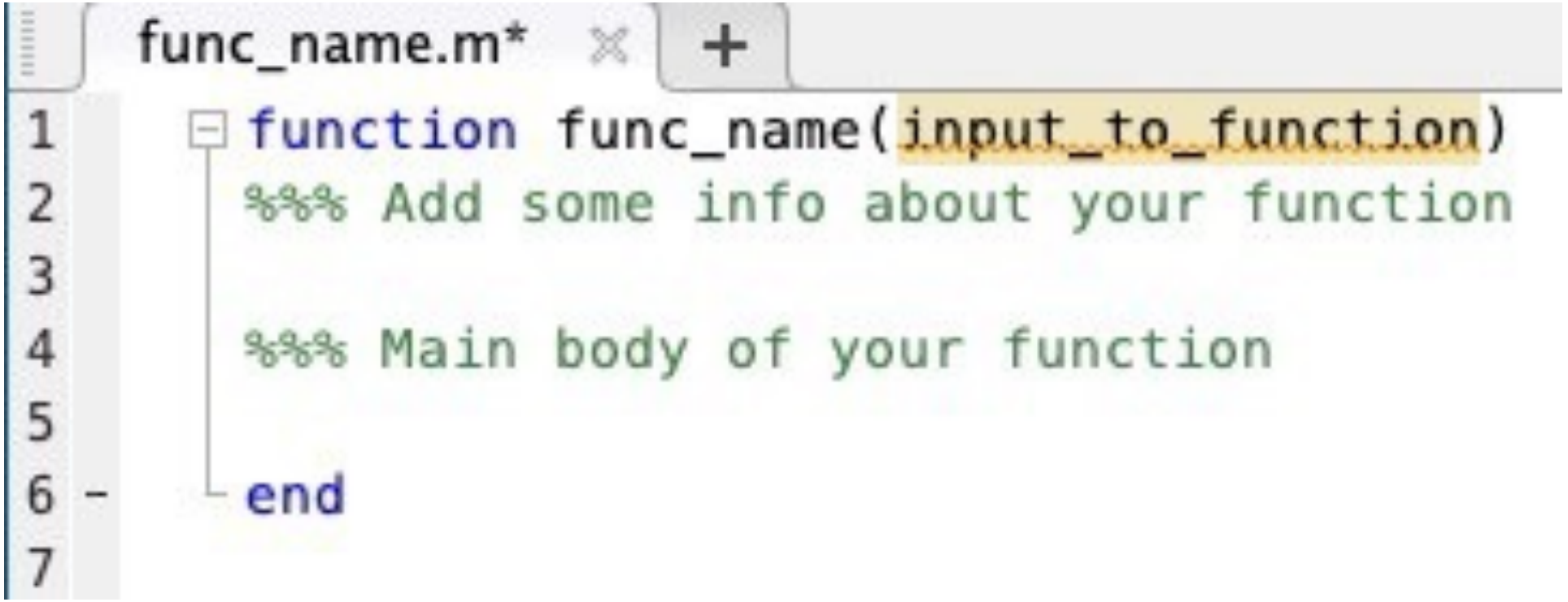
Functions in MATLAB

- You have already used many in-built MATLAB functions such as:

- sort
- length
- mean
- unique
- disp

- You can also make your own functions in MATLAB pretty easily

Syntax for functions



Important:

1) Rules for naming a function is same as rules for naming a variable
2) Name of the function and the name of the function file must be SAME
3) One function per file

Example: basic_average

basic_avg.m

```
1   function basic_avg(array)
2 -     avg=mean(array);
3 - end
```

Drawback: The output of this function is not stored as a separate variable, and so we can't access it after running the function.

Store output of the function as a variable

better_avg.m

```
1   function avg=better_avg(array)
2 -     avg=mean(array);
3 - end
```

```
>> avg_result = better_avg([9 2 4 6 7 10 11 1 0]);
```

Exercise 1: Write a function to calculate the circumference of a circle

$$\text{circumference} = 2 * \pi * \text{radius}$$

Make a function that takes in the radius of a circle and returns its circumference

```
>> circumference = calc_circ(4.6);
>> circumference

circumference =

        28.90
```

Custom functions can call built-in MATLAB functions

```
1  function avgx3 = mean_times_three(array)
2
3      avg = mean(array);
4      avgx3 = avg * 3;
5
6  end
```

Exercise 2: Write the following function:

Make a function that takes in a vector of elements and returns the sum of its unique elements

```
>> output = sum_unique([1 2 2 3 5 1 10 9]);
>> output

output =

          30
```

Exercise 3: Write your own MATLAB mean function

Write a function that takes in a vector of elements and returns its mean (without using the built in MATLAB mean function)

```
>> avg = my_mean([1 2 2 3 5 1 10 9]);
>> avg

avg =

        4.125
```

Functions can generate multiple outputs

min_max.m

```
1   function [min_num,max_num]=min_max(array)
2 -     min_num=min(array);
3 -     max_num=max(array);
4 -   end
```

```
>> [min_n, max_n] = min_max([1 2 3 5 6 1 10 7]);

>> [~, max_n] = min_max([1 2 3 5 6 1 10 7]);
```

Exercise 4: Write a function to return the results of five operations

Make a function that takes in a vector of elements and returns its mean, standard deviation, min, max, and unique elements, all as separate variables.

```
>> [avg, stand_dev, mini, maxi, uniq] = operations([1 2 2 3 5 1 10 9]);
>> stand_dev

mini =

     1
```

Exercise 5: Temperature conversion

$$C = \frac{5}{9}(F - 32)$$

$$K = C + 273.15$$

Write a function that takes in a temperature in Fahrenheit and returns two outputs: the converted temperature in Celsius and the converted temperature in Kelvin

Custom functions can call other custom functions

```
1  function avgx3 = mean_times_three(array)
2
3      avg = mean(array);
4      avgx3 = avg * 3;
5
6  end
```

```
1  function avgx3_plus5 = mean_times_three_plus_five(array)
2
3      avgx3 = mean_times_three(array);
4      avgx3_plus5 = avgx3 + 5;
5
6  end
```

Remember: one function per file!

Exercise 6:

Write a function that takes in an array of temperatures in Fahrenheit and returns both the average and the range of these temperatures in Celsius.

Hint: call the temperature conversion function you just wrote when coding this new function

```
>> [avg_C, range_C] = temp_operations([65 22 89]);
```