

Multiple commands at once: writing a program to a .m file

Steps

1. change to a directory where you will be saving/writing your .m code.
 - Change this in “location bar” at top of command window
 - Use the `>>cd` command;
2. Type `>> edit`
3. Write code
4. Use `%` as comment – anything on same line ignored
5. Save it, say as `myfirstcode.m`
6. Use `>> ls` command to make sure the codes and files you need are actually in your current directory
7. Run it: `>> myfirstcode` (no .m here)

The for loop

Basic structure (see code `for_loop_demo_1.m`):

```
for j=1:3
    disp(j)
end
```

More for loop examples – for_loop_demo_1.m-for_loop_demo_4.m

```
for k=1:2:5
    disp(k)
end
```

```
for p=[4 6 67 -1]
    disp(p)
end
```

the command can be anything!

```
sum=0

for k=1:5
    sum=sum+k
end
```

Another important point: if statements can be *nested*:

e.g build a matrix with entries $A(n, m) = m * n$

```
nested_for_loops.m
```

LOGIC:

```
>> a=4; b=5;
```

```
>> a=b
```

```
a =
```

```
5
```

```
>> a==b
```

```
ans =
```

```
0
```

```
>
```

```
>=
```

```
<
```

```
<=
```

```
~=
```

(there are others as well).

E.g.

```
>> b>a
```

```
ans =
```

```
1
```

If statements

Simplest example:

```
if (logical statement)
(expressions to execute if ``true" (that is, 1))
end
```

Implementation: `simple_if_statement.m`

```
a=4 ;
b=5 ;

if a<b
    disp('b bigger than a')
end
```

Using else and elseif

```
if (logical statement)
    (expressions to execute)
elseif (logical statement)
    (expressions to execute)
elseif (logical statement)
    (expressions to execute)
else
    (expressions to execute)
end
```

Implementation: `else_demo.m`

Another use for logical statements: ejecting from a for loop

The current for loop will be exited when `break` is reached

Implementation: `break_demo.m`

```
for j=1:5
    disp(j)

    if j >= 3
        break
    end
end
```

Need reminder or more info about use of a command?

See a new command in lecture or class materials?

Remember the doc and help features, experiment, and ask.

It is your responsibility to keep up with the commands introduced in class and in the 301.pdf course notes.

e.g.

```
help break
```

Basic plotting:

Implementation: `first_plot_demo.m`

Go to Edit menu → axis properties

Use zoom feature (magnifying glass)

Explore basic GUI plot editing features. To find more on this, type `>>doc`, search for editing plots ...

Our first scientific computing algorithm! Finding zeros (roots) of a function via bisection method. Implementation: `bisection.m`