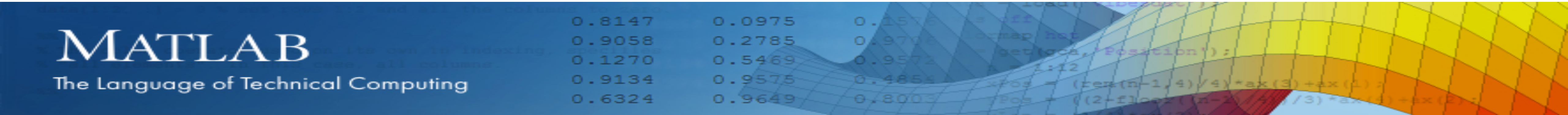
Getting Started with MATLAB

Hans-Petter Halvorsen

What is MATLAB?

- MATLAB is a tool for technical computing, computation and visualization in an integrated environment.
- MATLAB is an abbreviation for MATrix LABoratory
- It is well suited for Matrix manipulation and problem solving related to Linear Algebra, Modelling, Simulation and Control Applications
- Popular in Universities, Teaching and Research



MATLAB Syntax - Example

```
clear
clc
close all

x=[0, 1, 2, 3, 4 ,5];
y=[15, 10, 9, 6, 2 ,0];

for n=1:6 % n = model order

    p = polyfit(x,y,n)

    ymodel = polyval(p,x);

    subplot(3,2,n)
    plot(x,y,'o',x,ymodel)
    title(sprintf('Model order %d', n));

end
```

Defining Vectors

For Loop

Built-in Functions

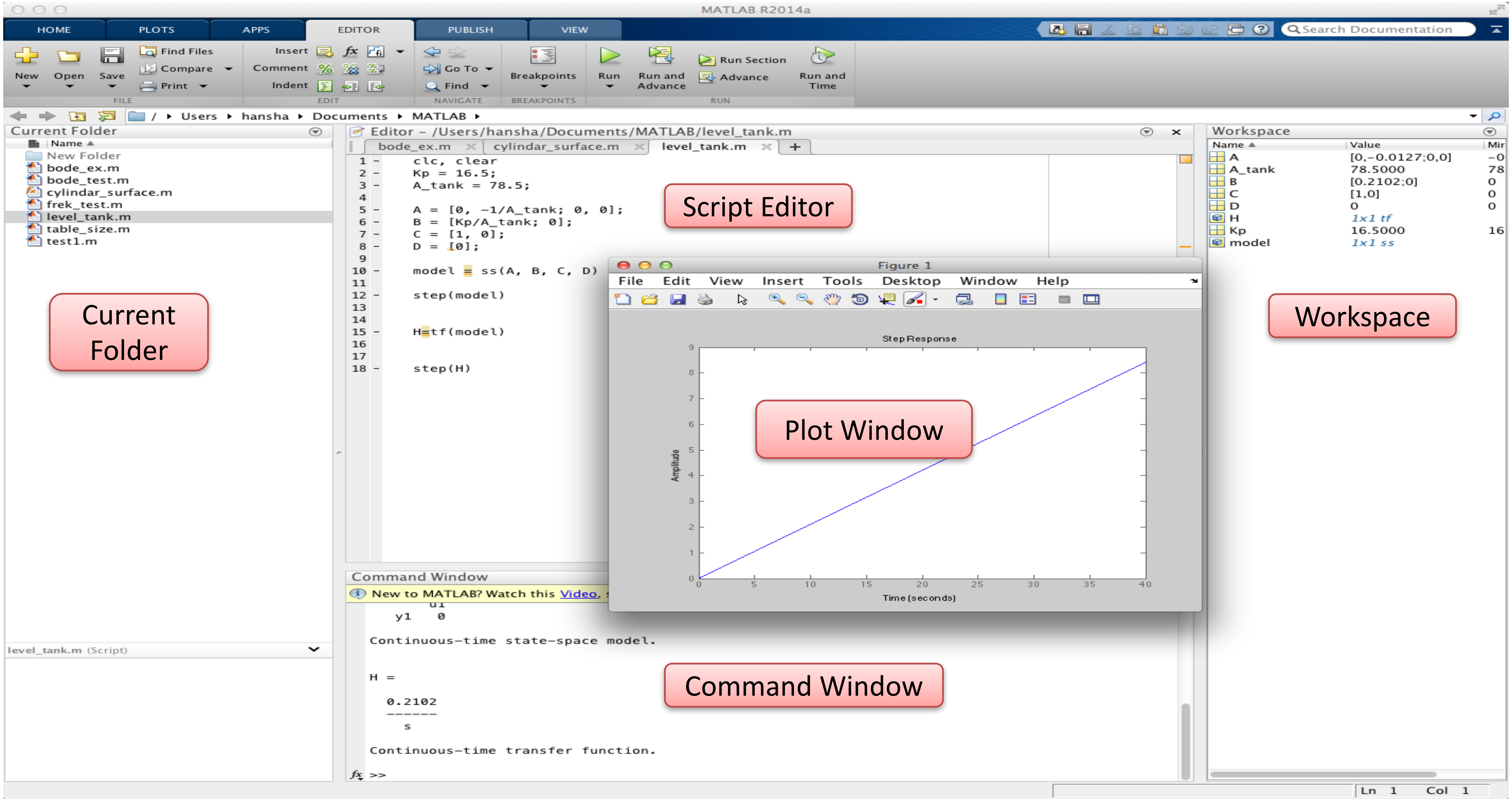
Topics

1. The MATLAB Environment (IDE)
2. MATLAB Basics
3. Vectors and Matrices
4. Plotting
5. Scripts (m-files)
6. User-defined Functions

MATLAB IDE

Hans-Petter Halvorsen

The MATLAB Environment (IDE)



Name ▲
New Folder
bode_ex.m
bode_test.m
cylindar_surface.m
frek_test.m
level_tank.m
table_size.m
test1.m

```
clc, clear
Kp = 16.5;
A_tank = 78.5;

A = [0, -1/A_tank; 0, 0];
B = [Kp/A_tank; 0];
C = [1, 0];
D = [0];

model = ss(A, B, C, D)


step(model)


H=tf(model)


step(H)
```

Name ▲	Value	Mir
A	[0,-0.0127;0,0]	-0
A_tank	78.5000	78
B	[0.2102;0]	0
C	[1,0]	0
D	0	0
H	1x1 tf	
Kp	16.5000	16
model	1x1 ss	

```
       u1
  y1    0

Continuous-time state-space model.


H =

   0.2102
   ------
     s

Continuous-time transfer function.

fx >>
```

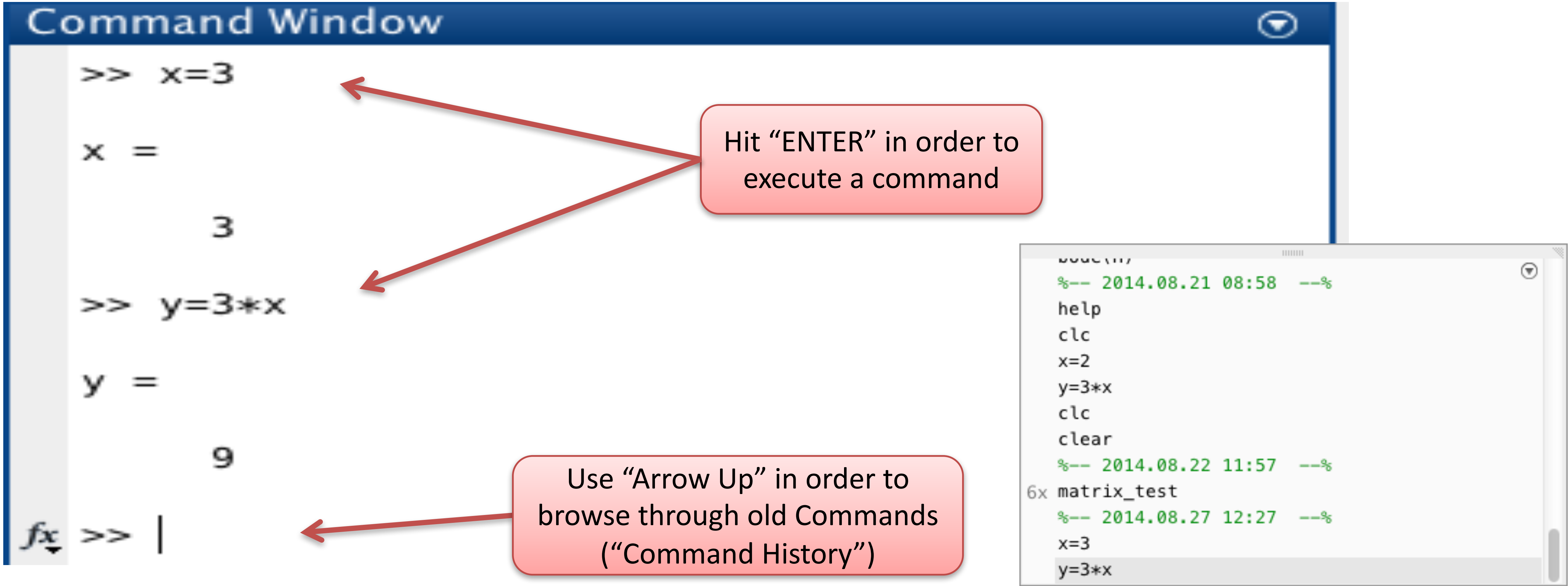
MATLAB Basics

Hans-Petter Halvorsen

MATLAB Basics

Command Window

The Command Window is the main window in MATLAB. Use the Command Window to enter variables and to run functions and M-files scripts (more about m-files later). Its like an advanced calculator!



```
>> x=3

x =

     3

>> y=3*x

y =

     9

fx >> |
```

Hit “ENTER” in order to execute a command

Use “Arrow Up” in order to browse through old Commands (“Command History”)

```
%-- 2014.08.21 08:58 --%
help
clc
x=2
y=3*x
clc
clear
%-- 2014.08.22 11:57 --%
6x matrix_test
%-- 2014.08.27 12:27 --%
x=3
y=3*x
```

MATLAB Basics

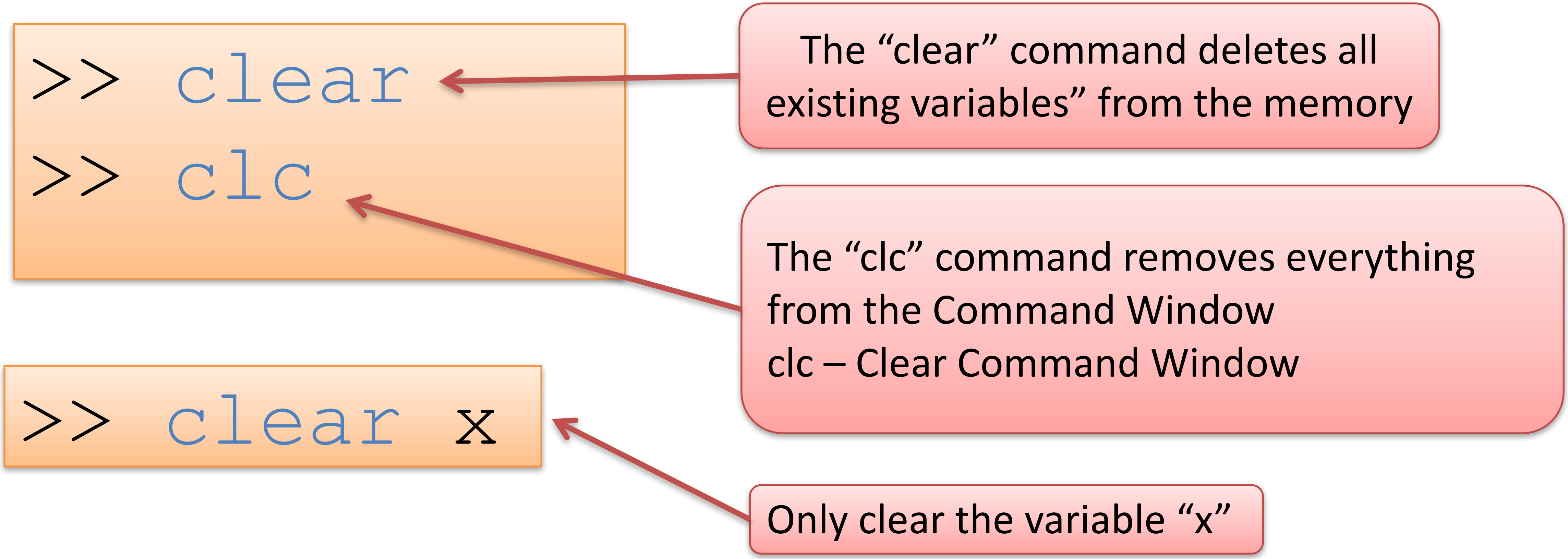
MATLAB is **case sensitive**! The variables x and X are not the same.

```
>> x=5;
>> X=6;
>> x+X

ans =
     11
```

```
>> x=3
x  =
       3


>> y=4;
>>
```

Unlike many other languages, where the semicolon is used to terminate commands, in MATLAB the semicolon serves to suppress the output of the line that it concludes.

MATLAB Basics



```
>> clear
>> clc
```

The “clear” command deletes all existing variables” from the memory

The “clc” command removes everything from the Command Window
clc – Clear Command Window

```
>> clear x
```

Only clear the variable “x”

MATLAB Basics

Built-in constants:

Name	Description
i, j	Used for complex numbers, e.g., z=2+4i
pi	π
inf	∞, Infinity
NaN	Not A Number. If you, e.g., divide by zero, you get NaN

Solutions:

MATLAB Basics

Name	Description
i, j	Used for complex numbers, e.g., z=2+4i
pi	π
inf	∞, Infinity
NaN	Not A Number. If you, e.g., divide by zero, you get NaN

```
>> r=5;
>> A=pi*r^2

A =
   78.5398
```

```
>> z1=3+3i;
>> z2=3+5i;
>> z = z1+z2
z =
   6.0000 + 8.0000i
```

```
>> a=2;
>> b=0;
>> a/b
```

Mathematical Expressions

	MATLAB
$\ln(x)$	`log(x)`
$\log_{10}(x)$	`log10(x)`
$\sqrt{x}$	`sqrt(x)`
e^x	`exp(x)`
x^2	`x^2`

Examples:

$$y(x) = \frac{3x+2}{2}$$

$$z = 3x^2 + \sqrt{x^2 + y^2} + e^{\ln(x)}$$

Solutions:

Mathematical Expressions

$$y(x) = \frac{3x + 2}{2} \qquad y(2) =?$$

```
>> x = 2;
>> y = (3*x+2)/2

y =
     4
```

Solutions:

Mathematical Expressions

$$z = 3x^2 + \sqrt{x^2 + y^2} + e^{\ln(x)}$$

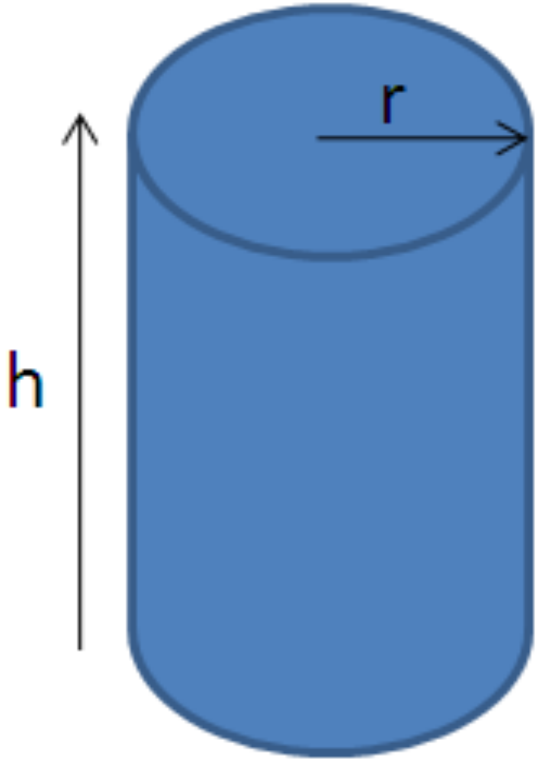
```
>> x=2;, y=2
>> z = 3*x^2 + sqrt(x^2 + y^2) + exp(log(x))

ans =
    16.8284
...
```

MATLAB Basics

We will use MATLAB in order to find the surface area (A) of a cylinder based on the height (h) and the radius (r) of the cylinder

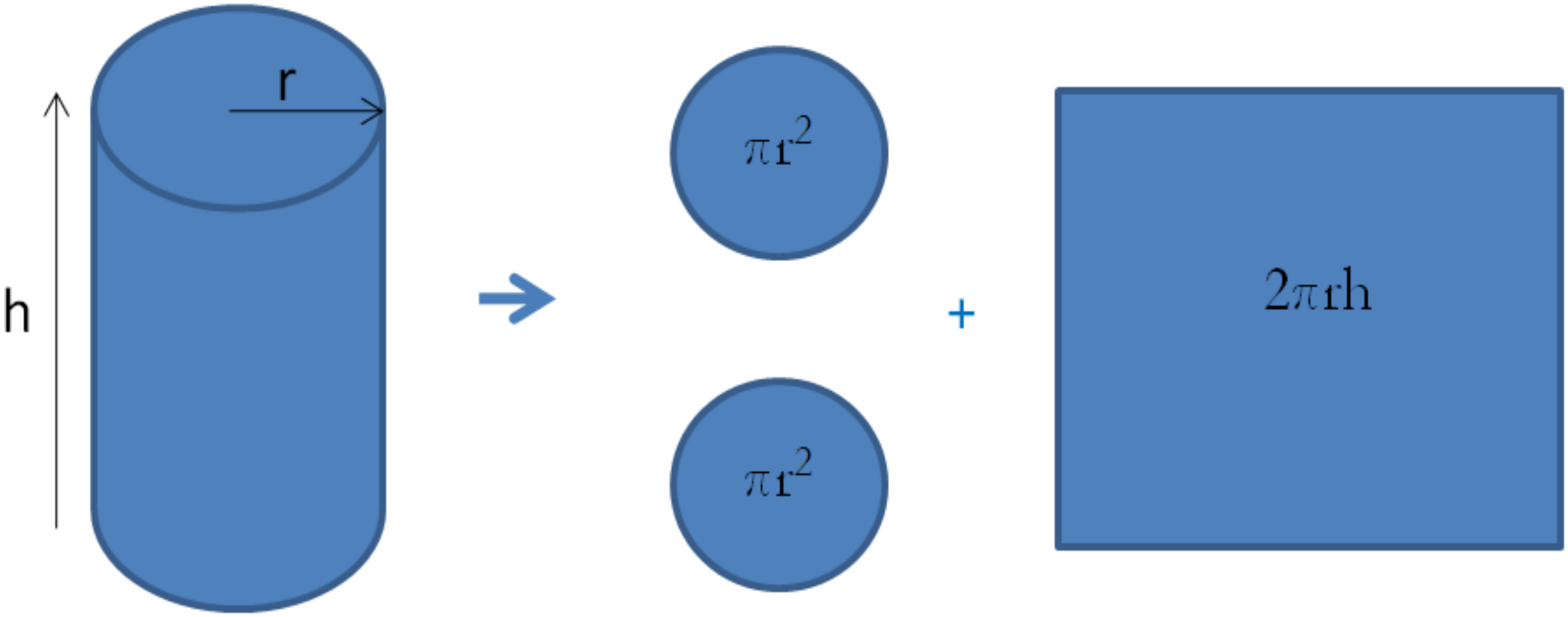
$$r = 3$$

$$h = 8$$



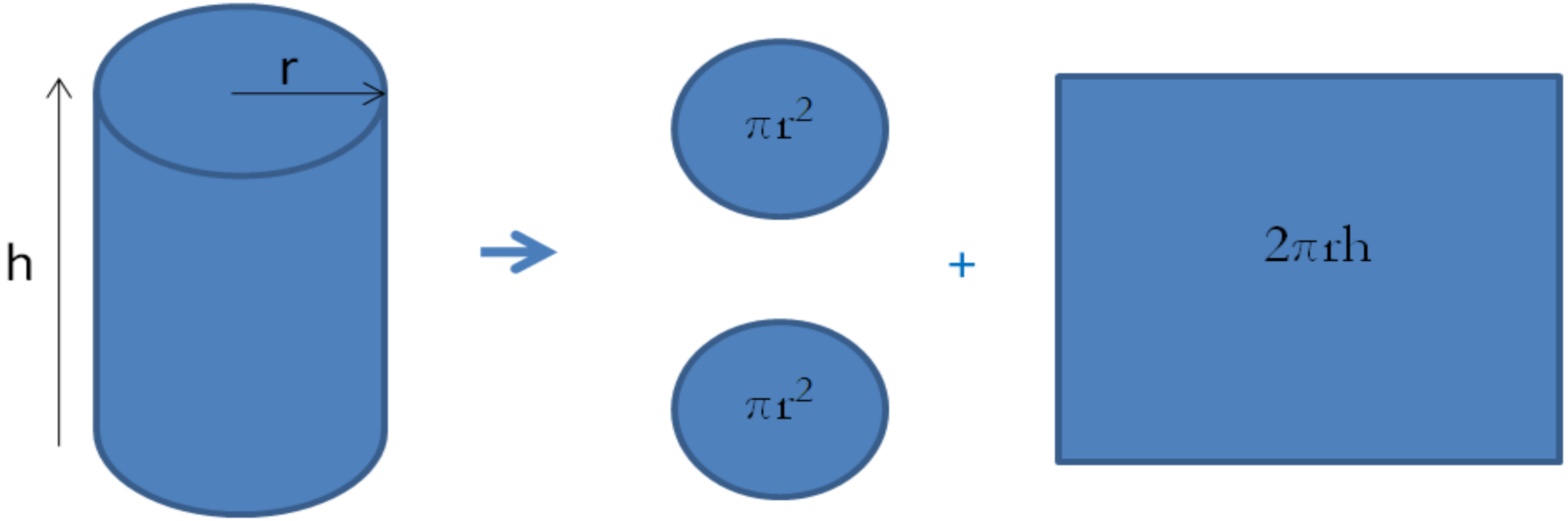
$$A =?$$

MATLAB Basics



Solutions:

MATLAB Basics



```
>> h=8
>> r=3
>> A = 2*pi*r^2 +2*pi*r*h;
A =
   207.3451
```

Vectors and Matrices in MATLAB

Hans-Petter Halvorsen

Vectors & Matrices

- Matrices and vectors (Linear Algebra) are the basic elements in MATLAB and also the basic elements in control design theory, etc.
- All variables in MATLAB is a matrix (but with different dimensions)
- So it is important you know how to handle vectors and matrices in MATLAB and in general

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nm} \end{bmatrix} \in R^{nxm}$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in R^{n}$$

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

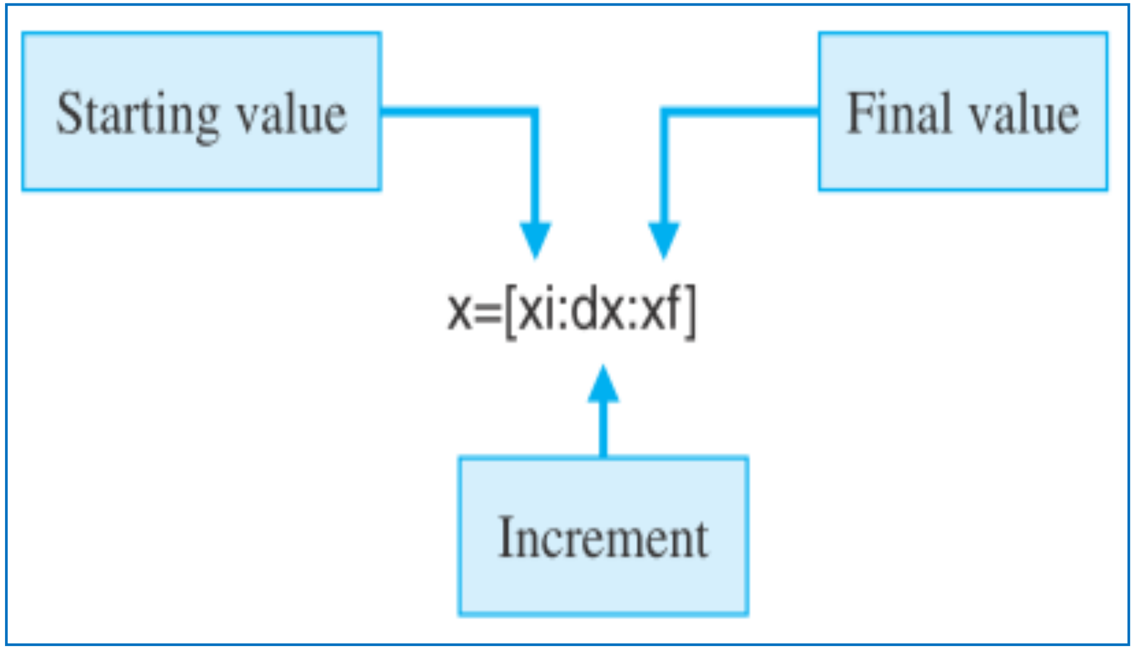
$$x = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

Vectors

Examples of different Rows and Columns vectors

```
>> x = [1, 2, 3]

>> y = [4; 5; 6]

>> z = [8, 9, 10]'
```

```
>> x*y
>> y*x
>> x*z
>> y*z
...
```



```
>> a = [1:10]

>> b = [1:2:10]

>> b = [1:0.5:4]
```

Vectors

Given the following Rain Data for a given Week (Monday to Sunday):

Day	Rain Amount
Monday	2,1 mm
Tuesday	10 mm
Wednesday	9,7 mm
Thursday	6,2 mm
Friday	2,5 mm
Saturday	0 mm
Sunday	8,3 mm

Vectors

Given the following Rain Data for a given Week (Monday to Sunday):

Day	Rain Amount
Monday	2,1 mm
Tuesday	10 mm
Wednesday	9,7 mm
Thursday	6,2 mm
Friday	2,5 mm
Saturday	0 mm
Sunday	8,3 mm

We define the Data in MATLAB like this:

```
>> x = [2.1, 10, 9.7, 6.2, 2.5, 0, 8.5]
```

If we are only interested in the Rain Amount on Monday:

```
>> x(1)
ans =     2.1000
```

Rain Amount on Friday:

```
>> x(5)
ans =     2.5000
```

Etc.

Vectors

Given the following Rain Data for a given Week (Monday to Sunday):

Day	Rain Amount
Monday	2,1 mm
Tuesday	10 mm
Wednesday	9,7 mm
Thursday	6,2 mm
Friday	2,5 mm
Saturday	0 mm
Sunday	8,3 mm

We define the Data in MATLAB like this:

```
>> x = [2.1, 10, 9.7, 6.2, 2.5, 0, 8.5]
```

What is the Average Rain Amount this Week?

In MATLAB we can use the ”mean” function:

```
>> mean(x)
ans =     5.5714
```

We can define a variable, e.g.:

```
>> mean_value_week = mean(x)
mean_value_week =     5.5714
```

Vectors

Given the following function:

$$y(x) = 2x^2 + 3x + 1$$

where: $-10 \le x \le 10$

```
>> x=-10:10
>> y=2.*x.^2 + 3.*x + 1
y =
   171    136    105      78
55      36      21      10       3
0       1       6      15      28
45      66      91     120     153
190     231
```

Note how we have used .* and .^

.* each element-wise Multiplication

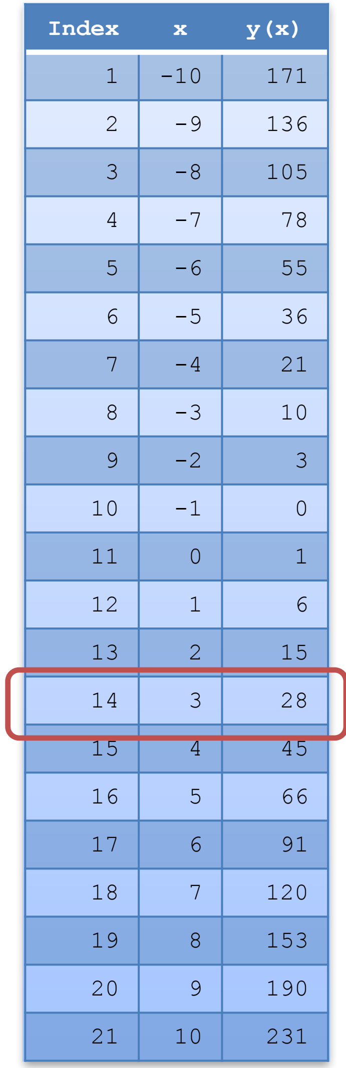
.^ each element-wise Power

What is $y(3) = ?$

```
>> y(14)
ans =      28
```

We can also do like this:

```
>> x = 3;
>> y = 2*x^2 + 3*x + 1
y =      28
```



Index	x	y(x)
1	-10	171
2	-9	136
3	-8	105
4	-7	78
5	-6	55
6	-5	36
7	-4	21
8	-3	10
9	-2	3
10	-1	0
11	0	1
12	1	6
13	2	15
14	3	28
15	4	45
16	5	66
17	6	91
18	7	120
19	8	153
20	9	190
21	10	231

Matrices

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

```
>> A = [1 2; 3 4]

A =   1        2
      3        4
```

or:

```
>> A = [1, 2; 3, 4]

A =   1        2
      3        4
```

$$B = \begin{bmatrix} 4 & 3 & 0 \\ 1 & -7 & 2 \\ 8 & 1 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} -1 & 3 & 0 \\ 4 & 7 & -2 \\ 2 & 0 & 9 \end{bmatrix}$$

```
>> B+C
>> B-C
>> B/C
>> B*C
>> B.*C
>> B'*C
...
```

Matrices

Given the following matrices:

$$n\left[\ \overset{m}{A}\ \right] m\left[\ \overset{p}{B}\ \right] = n\left[\ \overset{p}{C}\ \right]$$

$$A = \begin{bmatrix} 1 & 3 & 0 \\ 1 & -2 & 2 \\ 3 & 1 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 1 & 0 \\ -1 & 0 & 2 \end{bmatrix}$$

```
>>  A*B
>>  B*A
>>  A+B
>>  B'
>>  B'*C
>>  A*B'
>>  A'*B’
>>  A.*B
...
```

```
>>  A*(B*C)
>>  (A*B)*C
>>  (A+B)*C
>>  A*C + C*B
>>  (A+inv(B))*C

...
```

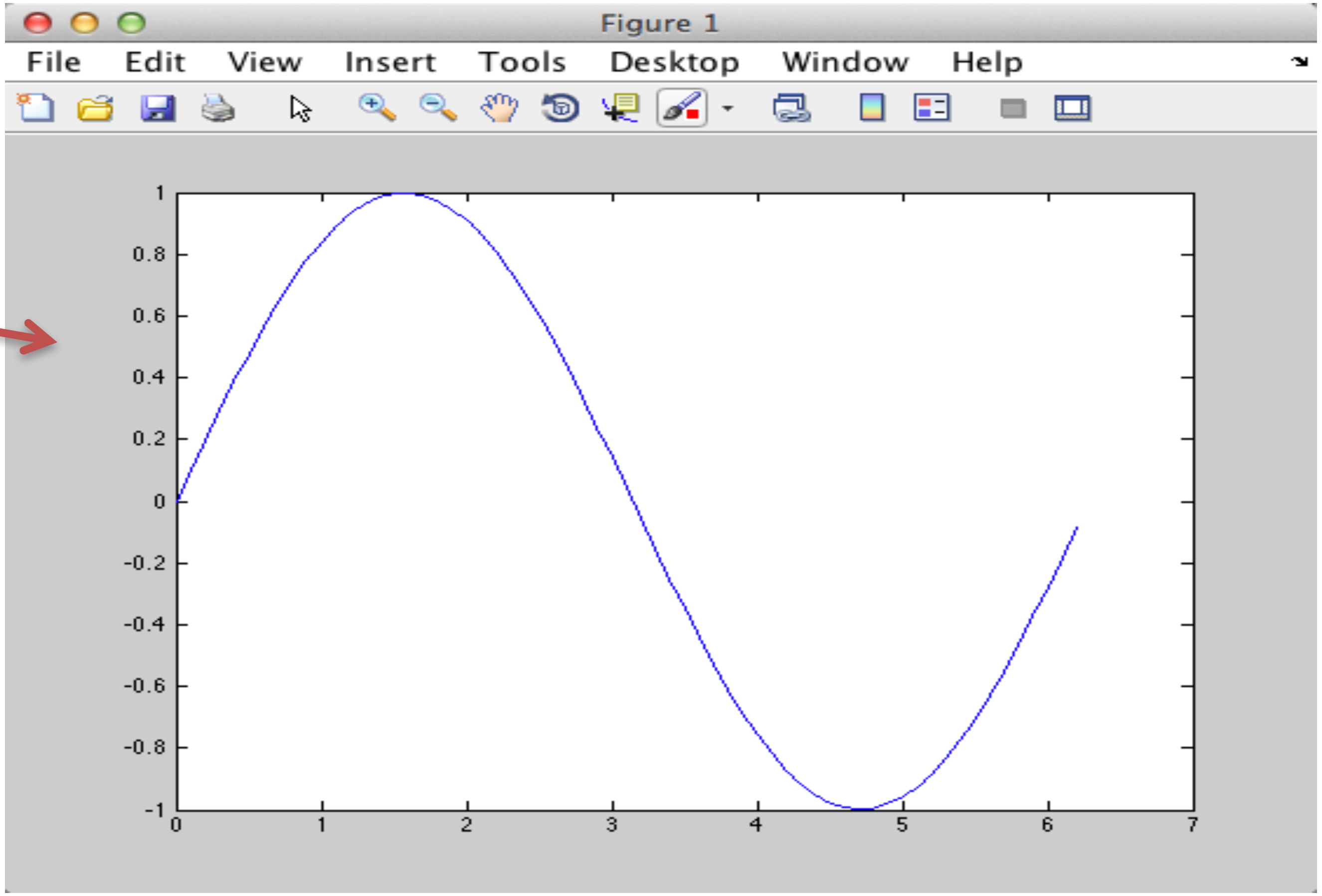
```
>>  rank(A)
>>  det(A)
>>  inv(A)
>>  inv(B)
>>  eig(A)
>>  inv(A)
>>  inv(B)
>>  diag(A)
>>  inv(A)*A
>>  A*inv(A)

...
```

Plotting in MATLAB

Hans-Petter Halvorsen

Plotting

```
>> x = 0:0.1:2*pi;
>> y = sin(x);
>> plot(x,y)
```



```
>> x = 0:0.1:2*pi;
>> y = sin(x);
>> y2 = cos(x);
>> plot(x,y, x,y2)
```

```
...
>> plot(x,y,'r*', x,y2,'g+')
```

Plotting

Plotting functions:

Name	Description
plot	Create a Plot
figure	Define a new Figure/Plot window
grid on/off	Create Grid lines in a plot
title	Add Title to current plot
xlabel	Add a Label on the x-axis
ylabel	Add a Label on the x-axis
axis	Set xmin,xmax,ymin,ymax
hold on/off	Add several plots in the same Figure
legend	Create a legend in the corner (or at a specified position) of the plot
subplot	Divide a Figure into several Subplots

```
>> x=0:0.1:2*pi;
>> y=sin(x);
>> plot(x,y)
>> title('Plot Example')
>> xlabel('x')
>> ylabel('y=sin(x)')
>> grid on
>> axis([0,2*pi,-1,1])
>> legend(’Temperature')
```

Plotting

Given the following Rain Data for a given Week (Monday to Sunday):

Day	Rain Amount
Monday	2,1 mm
Tuesday	10 mm
Wednesday	9,7 mm
Thursday	6,2 mm
Friday	2,5 mm
Saturday	0 mm
Sunday	8,3 mm

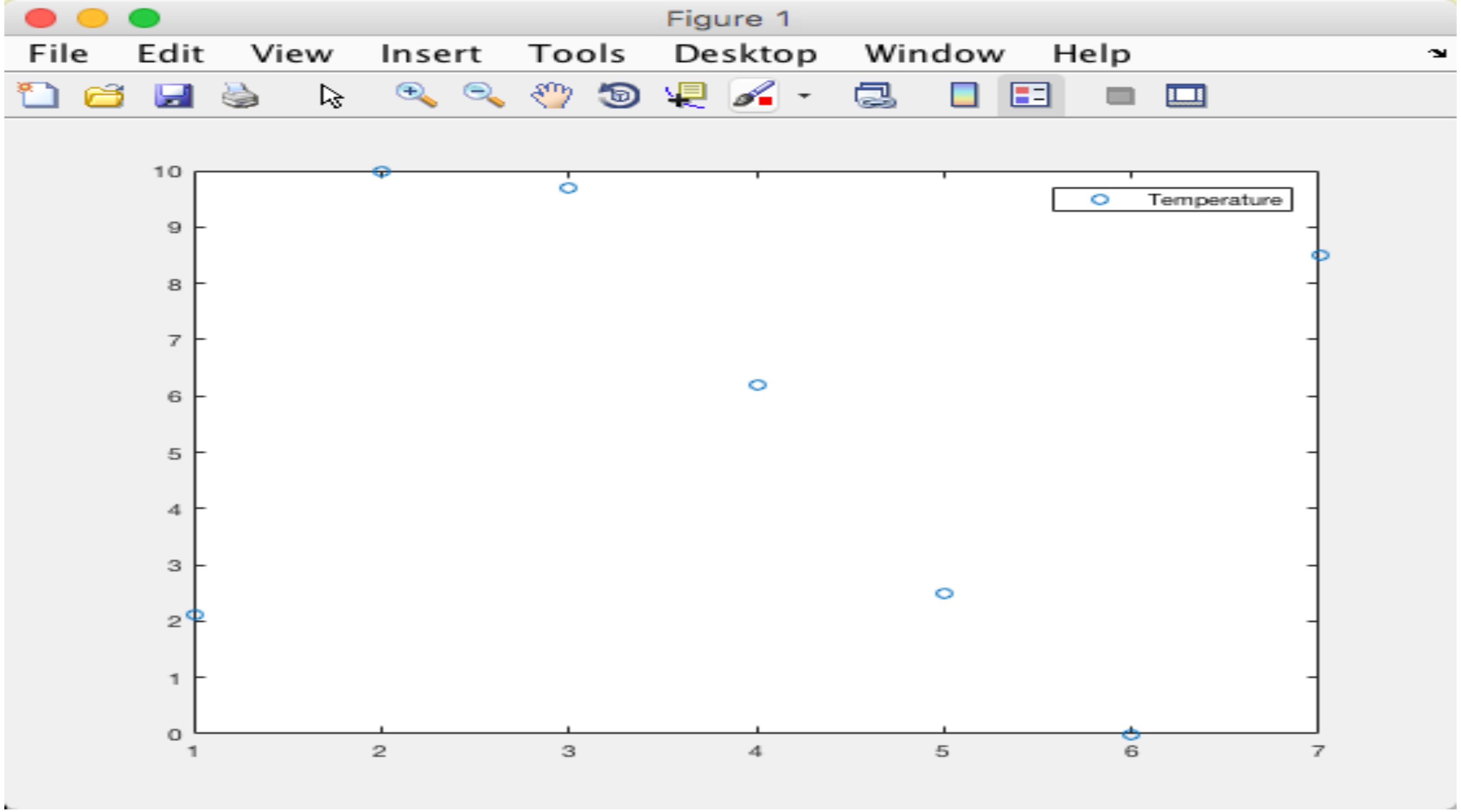
We will plot these values

Solutions:

Plotting

Day	Rain Amount
Monday	2,1 mm
Tuesday	10 mm
Wednesday	9,7 mm
Thursday	6,2 mm
Friday	2,5 mm
Saturday	0 mm
Sunday	8,3 mm

```
x = [2.1, 10, 9.7, 6.2, 2.5, 0, 8.5]
>> plot(x, 'o')
```



Plotting

Given the following function ($-10\ \leq x \leq 10$):

$$f(x) = 2x^2 + 3x + 1$$

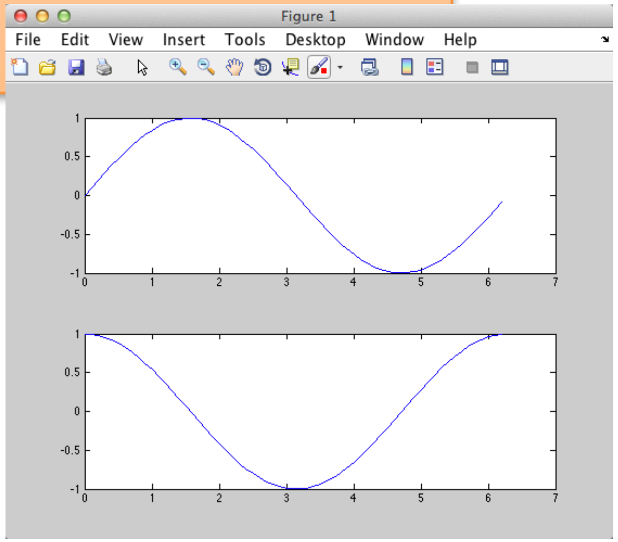
We will:

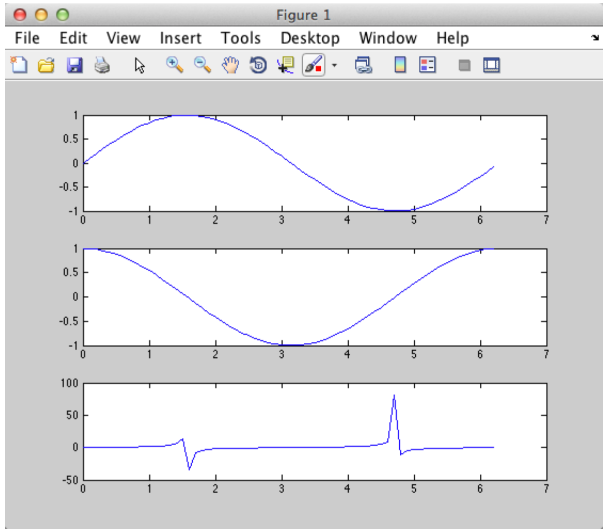
- Plot this function

- Use the Plot to find out:
  - For which value of x is $f(x) = 0$?
  - What is $f(5) =$?

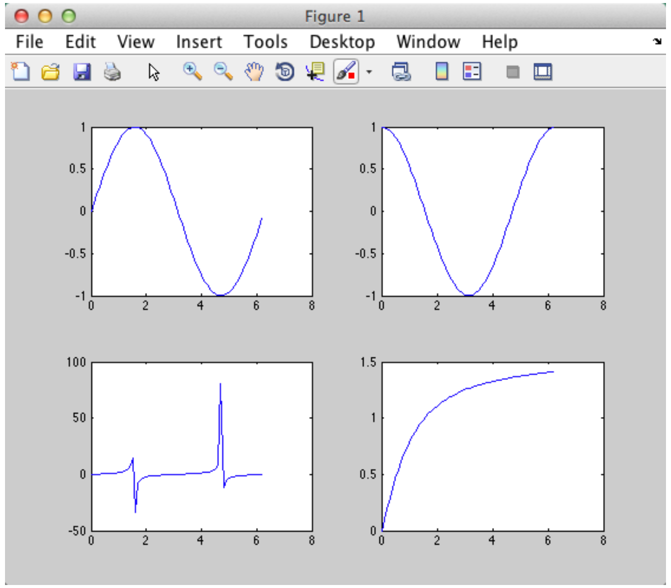
Subplots

```
>>  x=0:0.1:2*pi;
>>  y=sin(x);
>>  y2=cos(x);

>>  subplot(2,1,1)
>>  plot(x,y)

>>  subplot(2,1,2)
>>  plot(x,y2)
```





```
>>  x=0:0.1:2*pi;
>>  y=sin(x);
>>  y2=cos(x);
>>  y3=tan(x);

>>  subplot(3,1,1)
>>  plot(x,y)

>>  subplot(3,1,2)
>>  plot(x,y2)

>>  subplot(3,1,3)
>>  plot(x,y3)
```



```
>>  x=0:0.1:2*pi;
>>  y=sin(x);
>>  y2=cos(x);
>>  y3=tan(x);
>>  y4=atan(x);

>>  subplot(2,2,1)
>>  plot(x,y)

>>  subplot(2,2,2)
>>  plot(x,y2)

>>  subplot(2,2,3)
>>  plot(x,y3)

>>  subplot(2,2,4)
>>  plot(x,y4)
```

Scripts and User-defined Functions in MATLAB

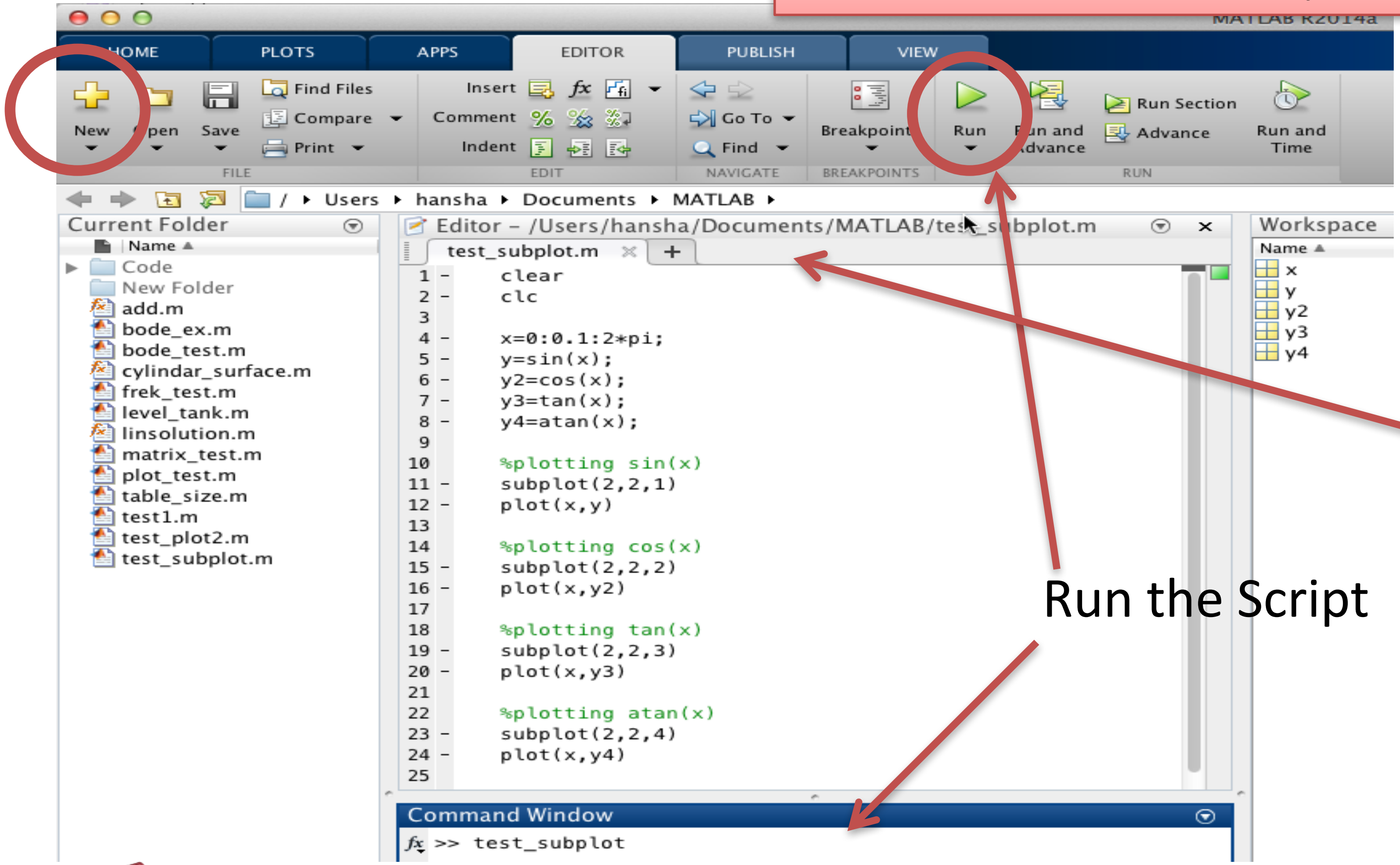
Hans-Petter Halvorsen

Scripts (m-files)

MATLAB Scripts are saved as so-called .m files (file extension is .m)

Script Editor

When using the Script Editor, you may create several lines of code and execute all in one batch. You can easily do changes in your code, create comments, etc.



```
clear
clc

x=0:0.1:2*pi;
y=sin(x);
y2=cos(x);
y3=tan(x);
y4=atan(x);

%plotting sin(x)
subplot(2,2,1)
plot(x,y)

%plotting cos(x)
subplot(2,2,2)
plot(x,y2)

%plotting tan(x)
subplot(2,2,3)
plot(x,y3)

%plotting atan(x)
subplot(2,2,4)
plot(x,y4)
```

```
>> test_subplot
```

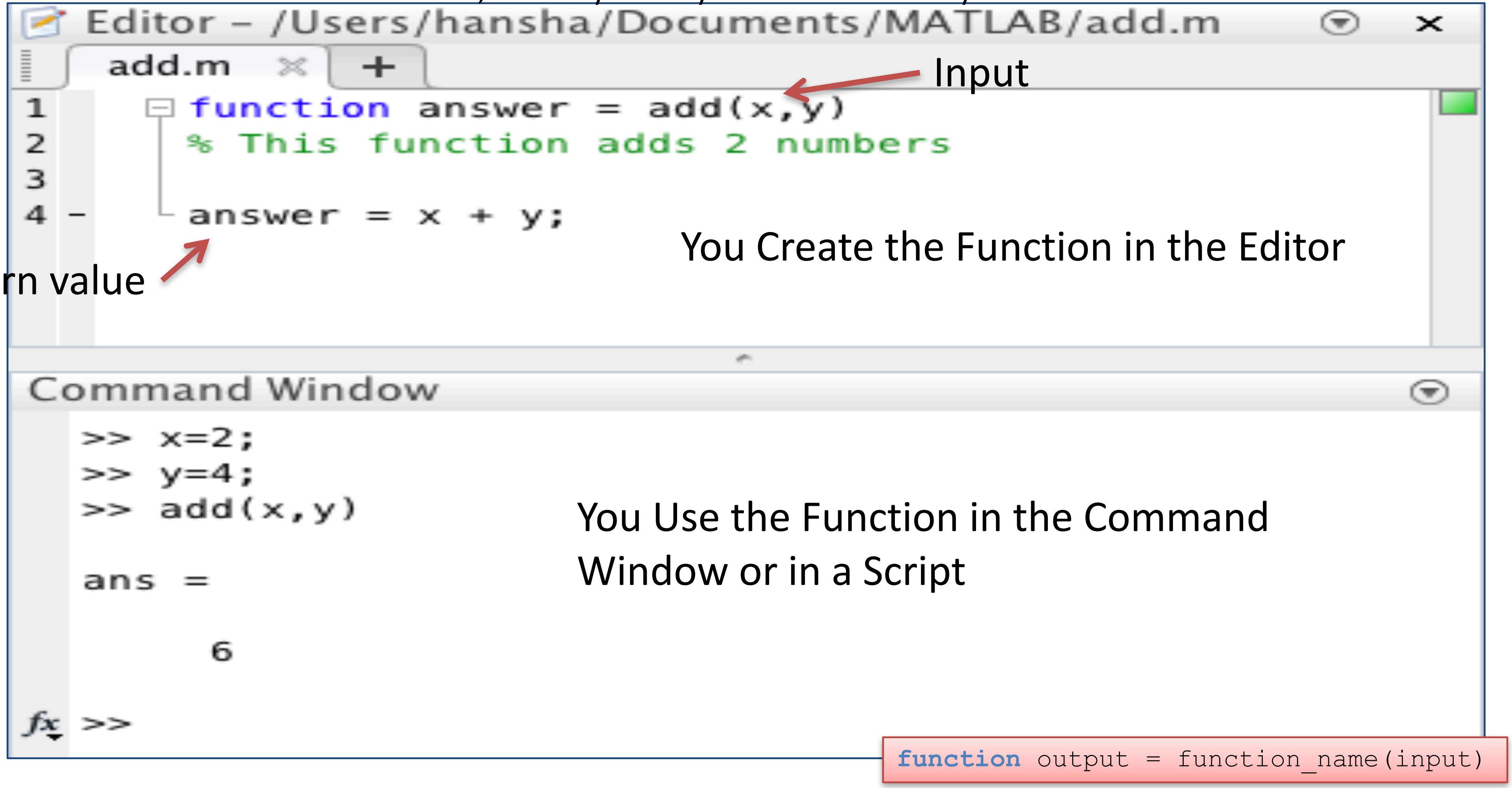
Run the Script

```
clear
clc

x=0:0.1:2*pi;
y=sin(x);
y2=cos(x);
y3=tan(x);
y4=atan(x);

%plotting sin(x)
subplot(2,2,1)
plot(x,y)

%plotting cos(x)
subplot(2,2,2)
plot(x,y2)

%plotting tan(x)
subplot(2,2,3)
plot(x,y3)

%plotting atan(x)
subplot(2,2,4)
plot(x,y4)
```

Students: Try this example

User-defined Functions

MATLAB contains hundreds of built-in functions, but very often you need to create your own functions



```
function answer = add(x,y)
% This function adds 2 numbers

answer = x + y;
```

Input

Return value

You Create the Function in the Editor

```
>> x=2;
>> y=4;
>> add(x,y)

ans =

     6

>>
```

You Use the Function in the Command Window or in a Script

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

User-defined Functions

Example: Convert from Celsius to Fahrenheit

$$T_F = \frac{9}{5}T_C + 32$$

We will create a User-defined Function that converts from Temperature in Celsius to Temperature in Fahrenheit

We can use the function like this in the Command Window:

```
>> Tc = 20;
>> Tf =
fahrenheit(Tc)

Tf =


    68
```

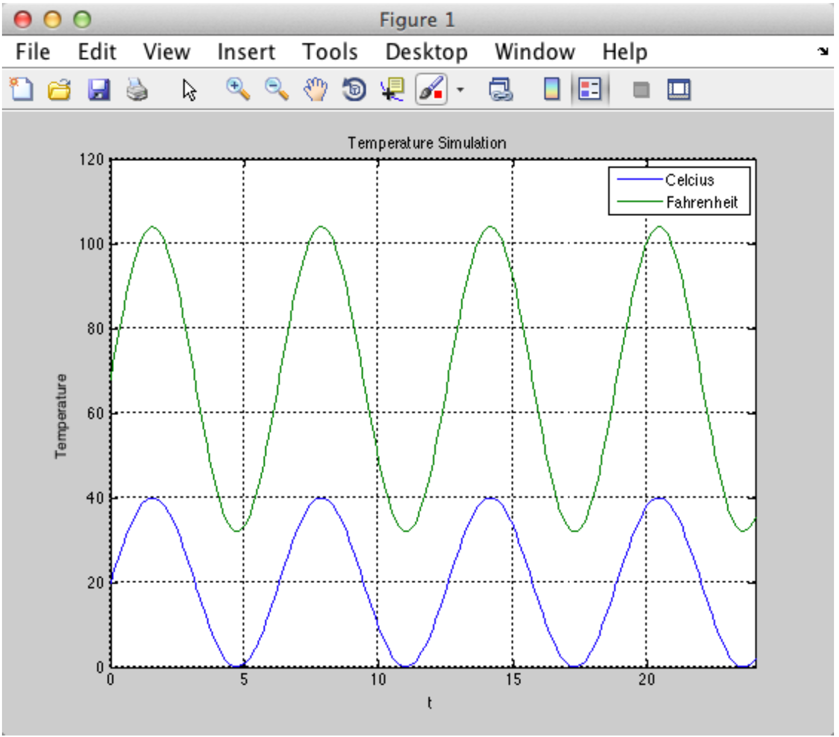
Solutions: Convert from Celsius to Fahrenheit

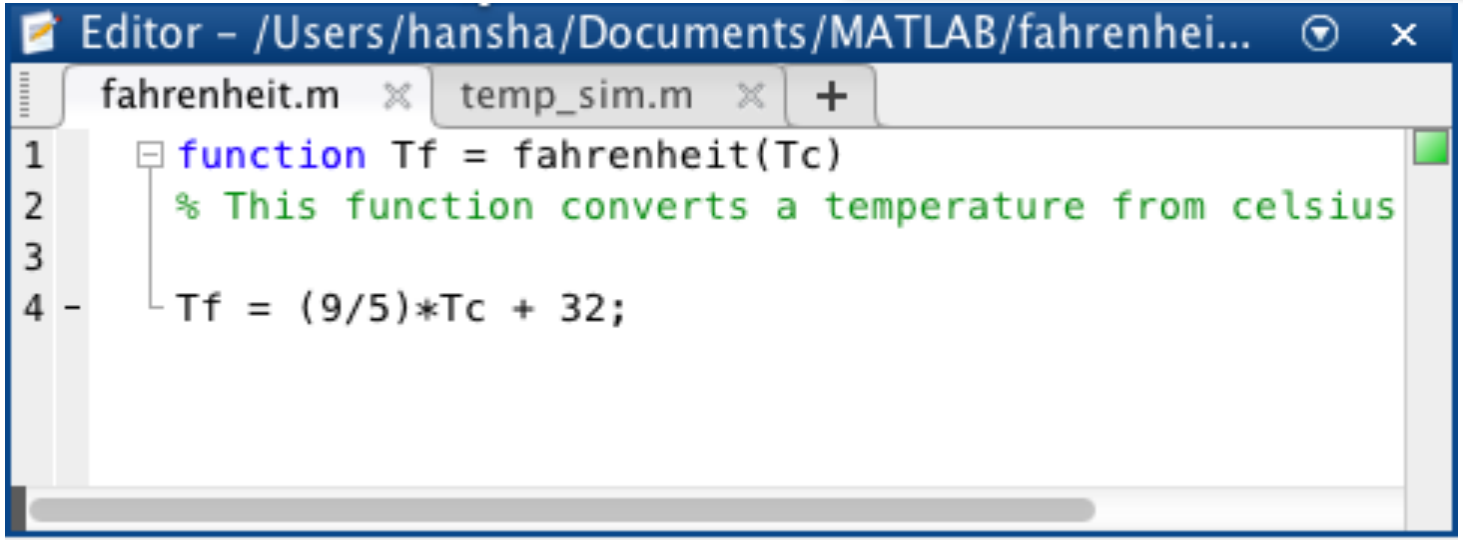
User-defined Functions

$$T_F = \frac{9}{5} T_C + 32$$

```
function Tf = fahrenheit(Tc)
% This function converts a temperature from celsius to
fahrenheit

Tf = (9/5)*Tc + 32;
```

```
clear
clc

t = 0:0.1:24;
Tc = (sin(t)+1)*20;
Tf = fahrenheit(Tc);

plot(t,Tc, t,Tf)

title('Temperature Simulation')
xlabel('t')
ylabel('Temperature')
grid on
axis([0,24, 0,120]);
legend('Celcius', 'Fahrenheit')
```





```
function Tf = fahrenheit(Tc)
% This function converts a temperature from celsius

Tf = (9/5)*Tc + 32;
```

Hans-Petter Halvorsen, M.Sc.

University College of Southeast Norway

www.usn.no

E-mail: hans.p.halvorsen@hit.no

Blog: http://home.hit.no/~hansha/