

<https://www.halvorsen.blog>



Scripts and User-defined Functions with MATLAB

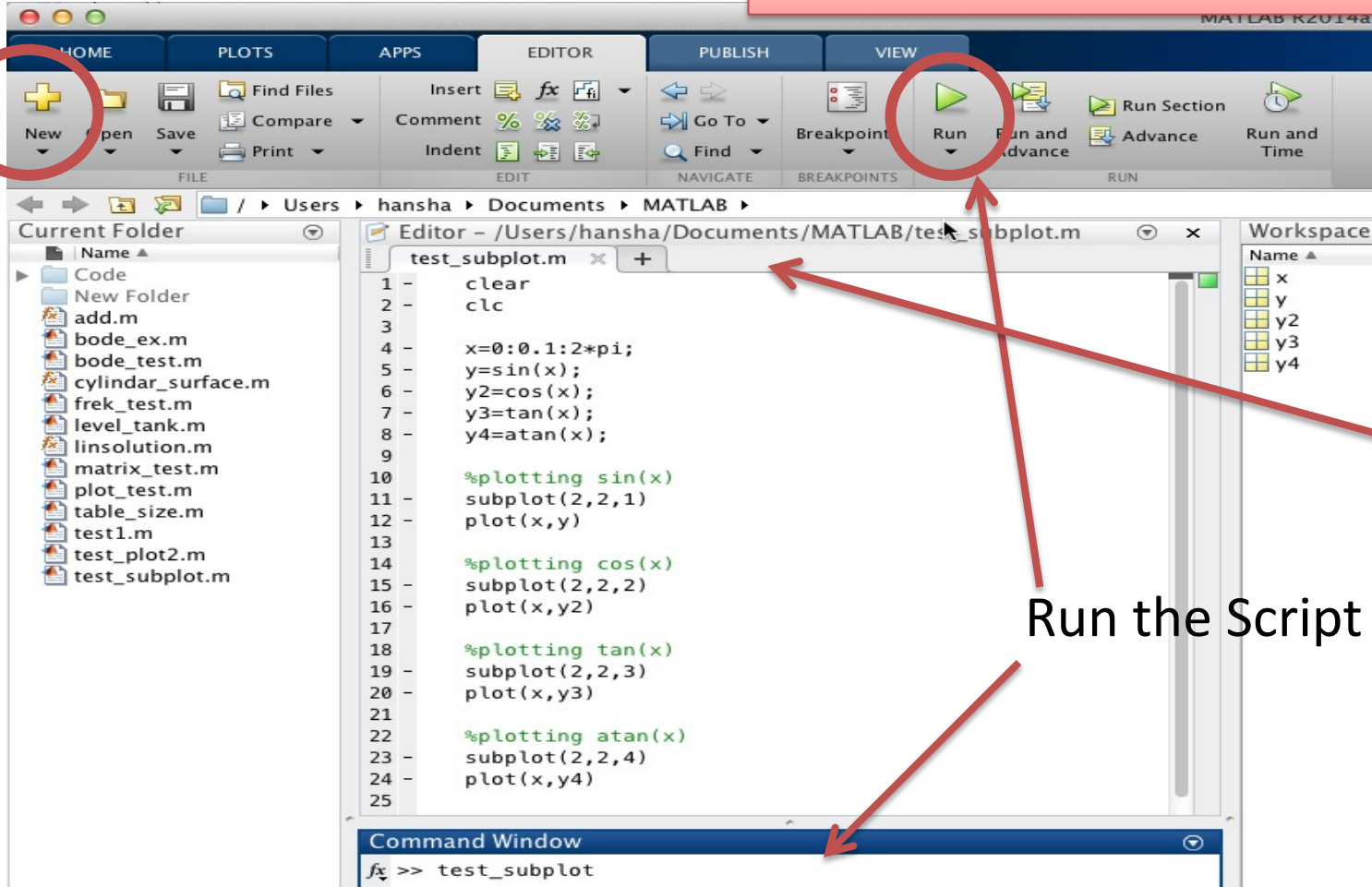
Hans-Petter Halvorsen

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.



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)
```

Script (M-file)

- Create a Script (M-file) where you create a vector with random data and find the average and the standard deviation
- Run the Script from the Command window.

```
x = rand(10, 1)
```

```
mean(x)
```

```
std(x)
```

```
x =      0.8147      0.9058
```

```
0.1270      0.9134
```

```
0.6324      0.0975
```

```
0.2785      0.5469
```

```
0.9575      0.9649
```

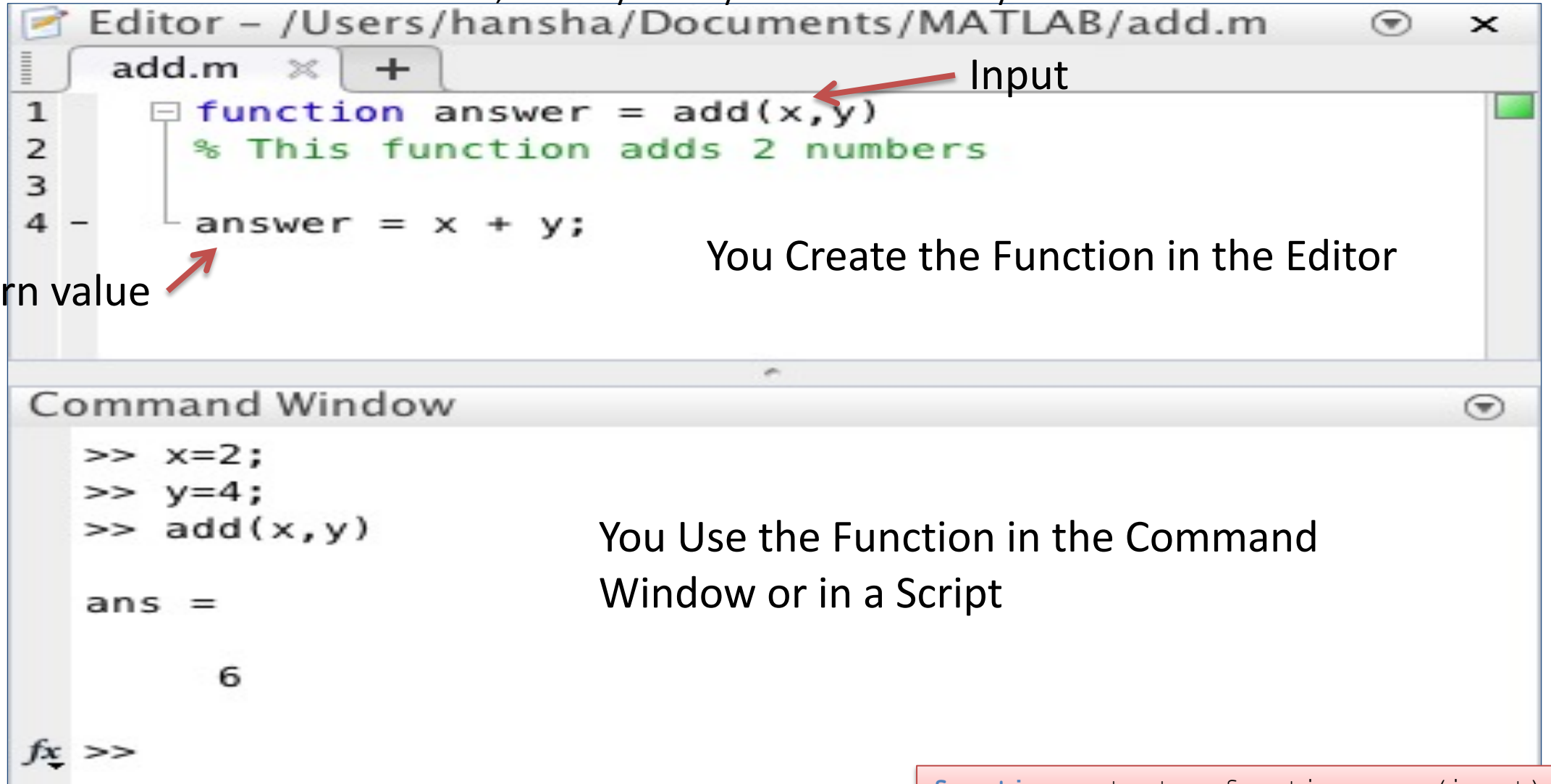
```
ans =      0.6239
```

```
ans =      0.3459
```



User-defined Functions

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



The screenshot shows the MATLAB Editor window with a file named 'add.m' open. The code in the editor is as follows:

```
1 function answer = add(x,y)
2 % This function adds 2 numbers
3
4 answer = x + y;
```

Annotations in the image include a red arrow pointing to the parameters 'x,y' in the function signature, labeled 'Input', and another red arrow pointing to the variable 'answer' in the assignment statement, labeled 'Return value'. Below the editor is the Command Window, which shows the following session:

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

ans =

     6

fx >>
```

Annotations in the image include the text 'You Create the Function in the Editor' pointing to the code in the editor, and 'You Use the Function in the Command Window or in a Script' pointing to the command window output.

`function` output = function_name(input)

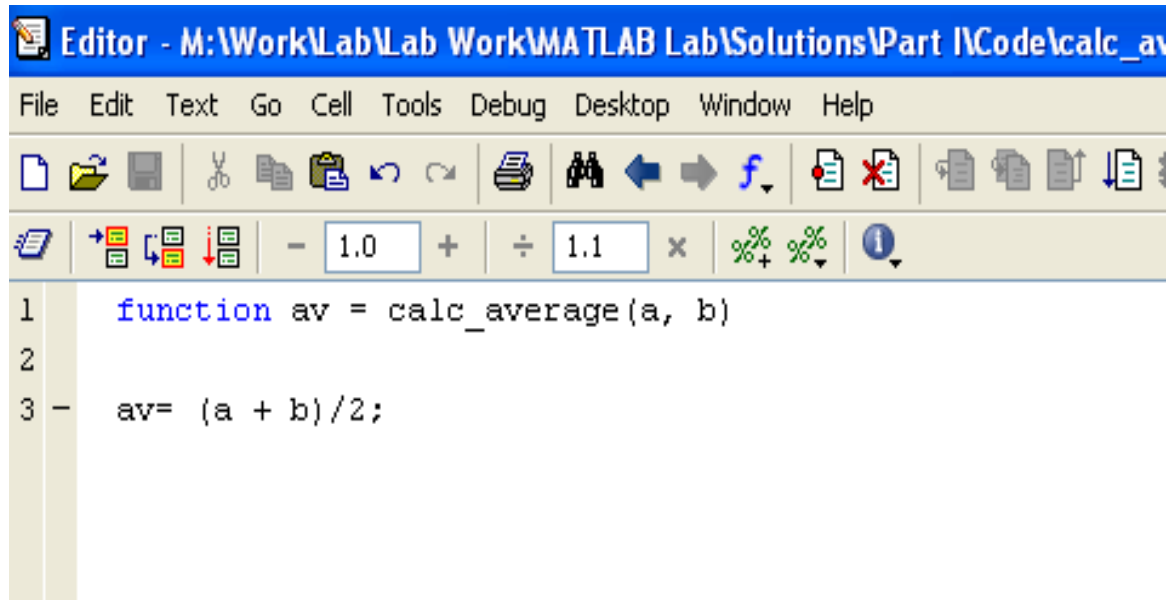
User-defined function

- Create a function **calc_average()** that finds the average of two numbers.
- Test the function afterwards in the Command window as follows:

```
>>x=2;
```

```
>>y=4;
```

```
>>z=calc_average(x,y)
```



The image shows a screenshot of a MATLAB editor window. The title bar reads "Editor - M:\Work\Lab\Lab Work\MATLAB Lab\Solutions\Part I\Code\calc_av". The menu bar includes "File", "Edit", "Text", "Go", "Cell", "Tools", "Debug", "Desktop", "Window", and "Help". The toolbar contains various icons for file operations and editing. Below the toolbar is a calculator-like interface with a minus sign, a text box containing "1.0", a plus sign, a text box containing "1.1", a multiplication sign, and percentage symbols. The main editor area shows the following code:

```
1 function av = calc_average(a, b)
2
3 - av= (a + b)/2;
```

We test the function in the Command window

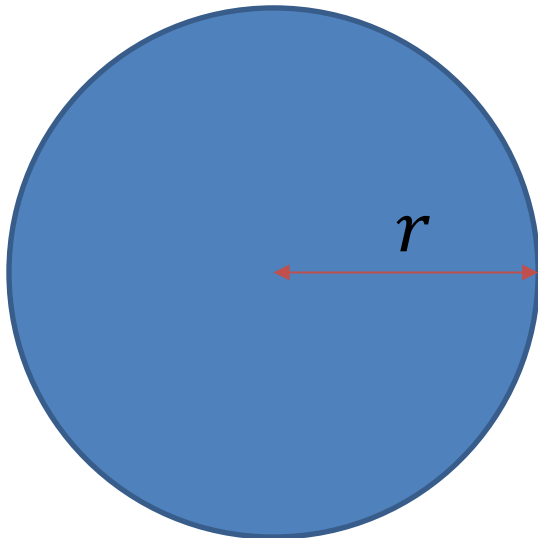
```
>> z=calc_average(x,y)
```

```
z =
```

```
3
```


User-defined function II

- Create a function **circle** that finds the area in a circle based on the input parameter r (radius).
- Run and test the function in the Command window.



$$A = \pi r^2$$

We define the function:

```
function A = circle(r)
```

```
A=pi*r*r;
```

Testing the function from the Command window:

```
>> circle(1)
```

```
ans =
```

```
3.1416
```

```
>> circle(2)
```

```
ans =
```

```
12.5664
```

```
>> r=4;
```

```
>> A=circle(r)
```

```
A =
```

```
50.2655
```



Hans-Petter Halvorsen



University of South-Eastern Norway

www.usn.no

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

Web: <https://www.halvorsen.blog>

