MATLAB Functions

Signals and Systems Lab

#2 lab notes

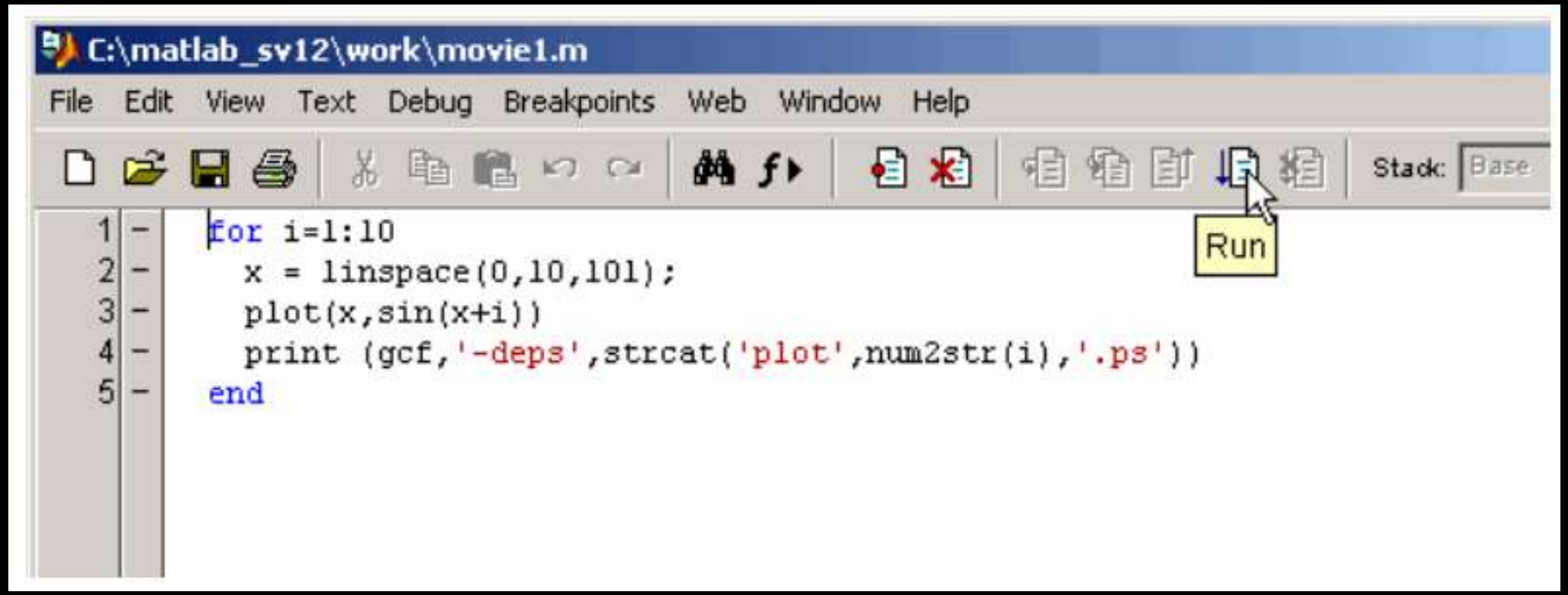
MATLAB Scripts

(a type of *.m files)

Scripts

- Computer languages:
  - Machine
  - Assembly
  - High-level
- Compiling vs. interpreting
  - Programs are compiled
  - Scripts are interpreted
- MATLAB scripts are stored in “m-files”

Script Example



Script that calculates area of circle

script1.m

```
radius = 5
area = pi * (radius^2)
```

Scripts can be run from the command window or directly from the editor window

```
>> script1
radius =
     5
area =
    78.5398
```

Write a script that calculates the circumference of a circle

Write a script that calculates the volume of a sphere

Write a script that calculates the area of a rectangle

Input and Output

```
>> rad = input('Enter the radius: ')
Enter the radius: 5
rad =
     5
```

```
>> mystr = input('Enter a string: ', 's')
Enter a string:       go
mystr =
      go

>> length(mystr)
ans =
      6
```

```
>> letter = input('Enter a char: ','s')
Enter a char: g
letter =
g
```

The same function **input()** can be used to input numbers, characters and strings

Write a script that calculates the area of a rectangle, with the two edges being input by the user at runtime

```
>> rad = input('Enter the radius: ')
Enter the radius: 5
rad =
     5
```

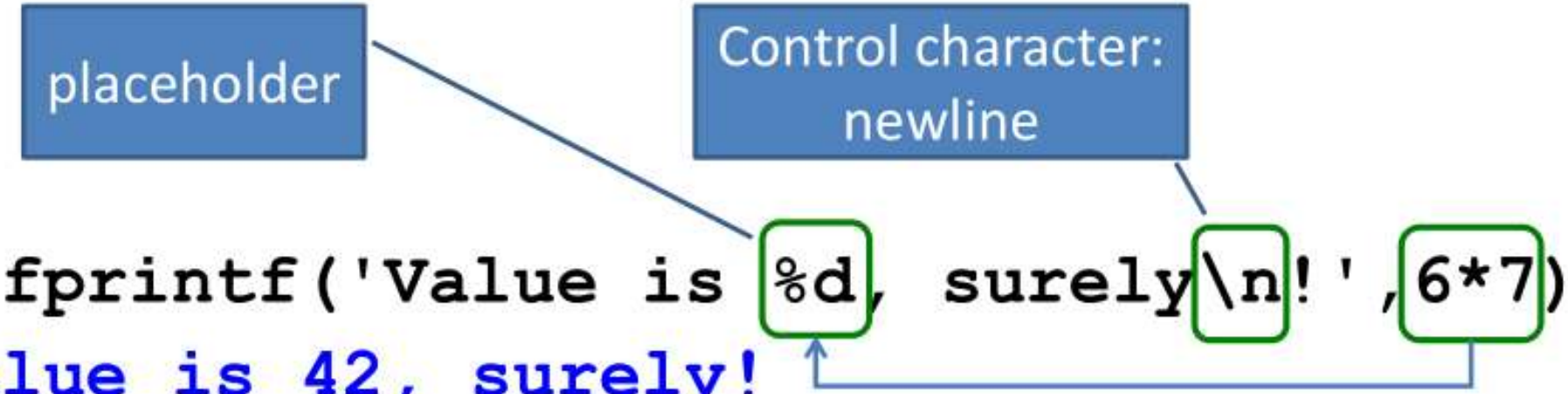
Output statements

- disp() → no formatting (simple!)
- fprintf() → with formatting (complex!)

```
>> disp('Hello!')
 Hello!
>> disp('6*7')
 42
```

The output is not assigned to the default variable **ans**

Output statements



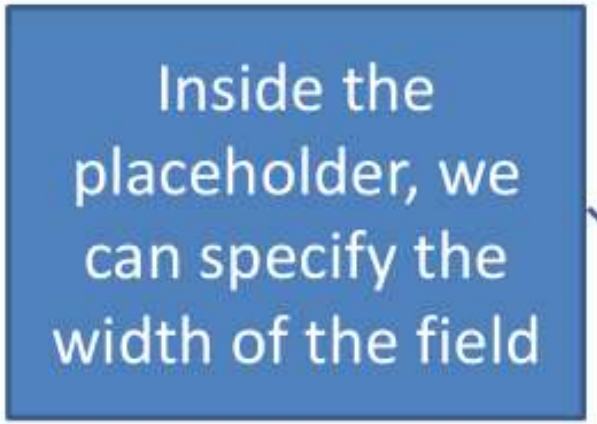

```
>> fprintf('Value is %d, surely\n!',6*7)
 Value is 42, surely!

>> fprintf('Int is %d, char is %c\n', ...
     45-3, 'a' )
 Int is 42, char is a
```

QUIZ: Output statements

The numbers 3 and 10 are stored in the variables a and b. Print this to the screen:

The sum of 3 and 3 is 13.




```
>> fprintf('Value is %5d!\n',6*7)
 Value is____42!
```

Use the width option to print the following 3 numbers in a column, aligned to the right: 42 1024 5
(do not cheat by putting spaces by hand!)

Printing vectors and matrices

```
>> vec = 2:5;
>> fprintf('%d\n', vec)
2
3
4
5
```

```
>> fprintf('%d', vec)
2345>>
```

In a script, we can fix this problem!

Printing vectors and matrices

printvec.m

```
% This demonstrates printing a vector

vec = 2:5;
fprintf('%d ',vec)
fprintf('\n')
```

```
>> printvec
2 3 4 5
>>
```

printvec.m

```
% This demonstrates printing a vector

vec = 2:5;
fprintf('%d ',vec)
fprintf('\n')
```

```
>> printvec
2 3 4 5
>>
```

Write a complete script that creates a row vector of all the prime numbers smaller than 10, and then prints them “nicely” on one line.
On the line above the numbers, there should be printed the word *Primes*.

Printing vectors and matrices

```
>> mat = [5 9 8; 4  1 10]
mat =
     5     9     8
     4     1    10
```

```
>> fprintf('%d\n', mat)
5
4
9
1
8
10
```

Remember: MATLAB stores matrices by columns!

Printing vectors and matrices

```
>> mat = [5 9 8; 4  1 10]
mat =
     5     9     8
     4     1    10
```

```
>> fprintf('%d %d %d\n', mat)
5 4 9
1 8 10
```

```
>> fprintf('%d %d %d\n', mat')
5 9 8
4 1 10
```

Finally!

Printing vectors and matrices … the painless way!

```
>> mat = [15 11 14; 7 10 13]
mat =

>> disp(mat)
    15    11    14
     7    10    13
```

Functions

calcarea.m

```
function area = calcarea(rad)
% calcarea calculates the area of a circle
% Format of call: calcarea(radius)
% Returns the area

area = pi * rad * rad;
end
```

Your turn!

Write a function that returns the volume of rectangular box

Selection a.k.a.

Decision

Relational expressions

```
>> 3<5

ans =

     1
```

true, logical(1)

```
>> 2>9

ans =

     0
```

false, logical(0)

Relational operators (logical)

Operator	Meaning
>	greater than
<	less than
>=	greater than or equals
<=	less than or equals
==	equality
~=	inequality

Operator	Meaning
||	or (for scalars)
&&	and (for scalars)
~	not

xor(op1, op2) exclusive or

Table 3.1 Truth Table for Logical Operators

x	y	~x	x || y	x && y	xor(x,y)
true	true	false	true	true	false
true	false	false	true	false	true
false	false	true	false	false	false

QUICK QUESTION!

Assume that there is a variable x that has been initialized. What would be the value of the expression

$3 < x < 5$

if the value of x is 4? What if the value of x is 7?

Hint: The expression is evaluated from Left to Right

Table 3.1 Truth Table for Logical Operators

x	y	~x	x || y	x && y	xor(x,y)
true	true	false	true	true	false
true	false	false	true	false	true
false	false	true	false	false	false

- X || 0 = ?
- X || 1 = ?
- X && 0 = ?
- X && 1 = ?
- xor(a, 1) = ?
- xor(a, 0) = ?

true/false can also be used in arithmetical expressions!

```
>> 3<5
ans =
     1
>> 42 + ans
ans =
    43
```

PRACTICE 3.1

Think about what would be produced by the following expressions.

```
4 > 3 + 1

'e' == 'd' + 1

3 < 9 - 2

(3 < 9) - 2

4 == 3 + 1 && 'd' > 'c'

3 >= 2 || 'x' == 'y'

xor(3 >= 2,'x' == 'y')

xor(3 >= 2,'x' ~= 'y')
```

QUIZ: In a script, we ask for a yes/no answer like this

```
letter = input('Choice (Y/N): ','s')
```

Are the following expressions the same?

```
letter == 'y' || 'Y'
```

```
letter == 'y' || letter == 'Y'
```

Table 3.2 Operator Precedence Rules

Operators	Precedence
parentheses: ( )	highest
transpose and power ', ^	
unary: negation (–), not (~)	
multiplication, division *,/,\	
addition, subtraction +, –	
colon operator :	
relational <, <=, >, >=, ==, ~=	
and &&	
or ||	
assignment =	lowest

Are the following expressions the same?

```
letter == 'y' || 'Y'
```

```
letter == 'y' || letter == 'Y'
```

if statement

```
>> num = -4;
>> if num < 0
       num = abs(num)
   end
num =
     4
```

sqrtifexamp.m

```
% Prompt the user for a number and print its sqrt
num = input('Please enter a number: ');
% If the user entered a negative number, change it
if num < 0
    num = abs(num);
end
fprintf('The sqrt of %.1f is %.1f\n',num,sqrt(num))
```

if statement

```
>> num = -4;
>> if num < 0
       num = abs(num)
   end
num =
     4
```

sqrtifexamp.m

```
% Prompt the user for a number and print its sqrt
num = input('Please enter a number: ');
% If the user entered a negative number, change it
if num < 0
    num = abs(num);
end
fprintf('The sqrt of %.1f is %.1f\n',num,sqrt(num))
```

QUIZ: What does this statement print?

```
>> if 5
       disp('Yes, this is true!')
   end
```

QUIZ: What does this program do?

```
A = rand()
B = rand()
if A > B
    A = B
    B = A
```

if-else statement

```
if rand < 0.5
   disp('It was less than .5!')
else
   disp('It was not less than .5!')
end
```

if-elseif-else statement

```
if   x < -1
     y = 1
elseif x <= 2
     y = x^2
else
     y = 4
end
```

There can be multiple **elseif** before the final **else**

switch statement

```
switch quiz
    case 10
        grade ='A'
    case 9
        grade ='A'
    case 8
        grade ='B'
    ...
    otherwise
        grade ='F'
    end
```

Can be combined as **case {10, 9}**

otherwise is optional … but strongly recommended!

Lab Work

Sources

Agapie, M. (2013), *CS 344 Class Notes* [used with permission]

Attaway, S. (2012). *MATLAB a practical introduction to programming and problem solving* (2nd ed.). Waltham, MA: Butterworth-Heinemann.