جـــامـــعـــة دمـــشـــق
كـلـيـة الـهـنـدسـة الـمـدنـيـة

تـصـمـيـم بـمـعـونـة الـحـاسـب
Computer Aided Design

Lecture 1

MATLAB Introduction

11/11/2025

Wael Darwich

Damascus University

Civil Engineering Faculty

- Download material later from:
  - http://damascusuniversity.edu.sy/civil/index.php
  - https://cad.el.sy/

- Contact:

  w@el.sy

Content

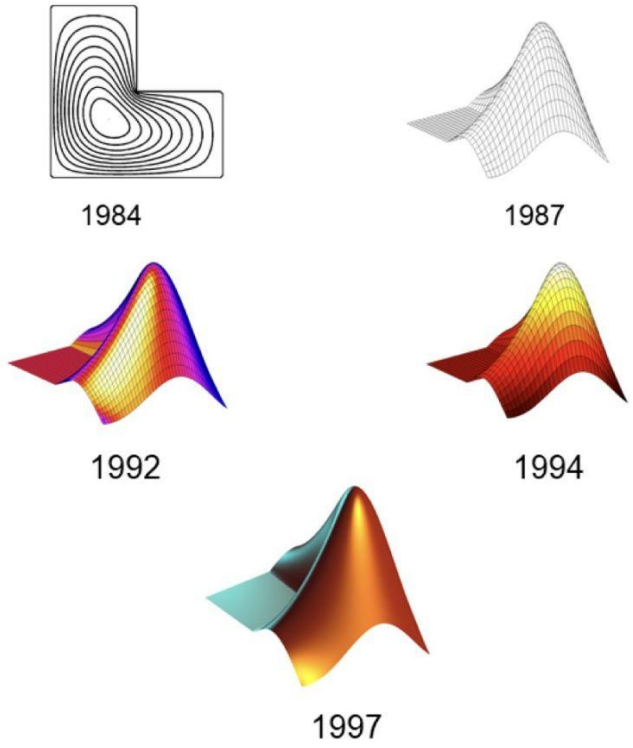
- Introduction
- Memory and Variables
- Operators
- Scripts vs. Functions
- Matrix Calculator

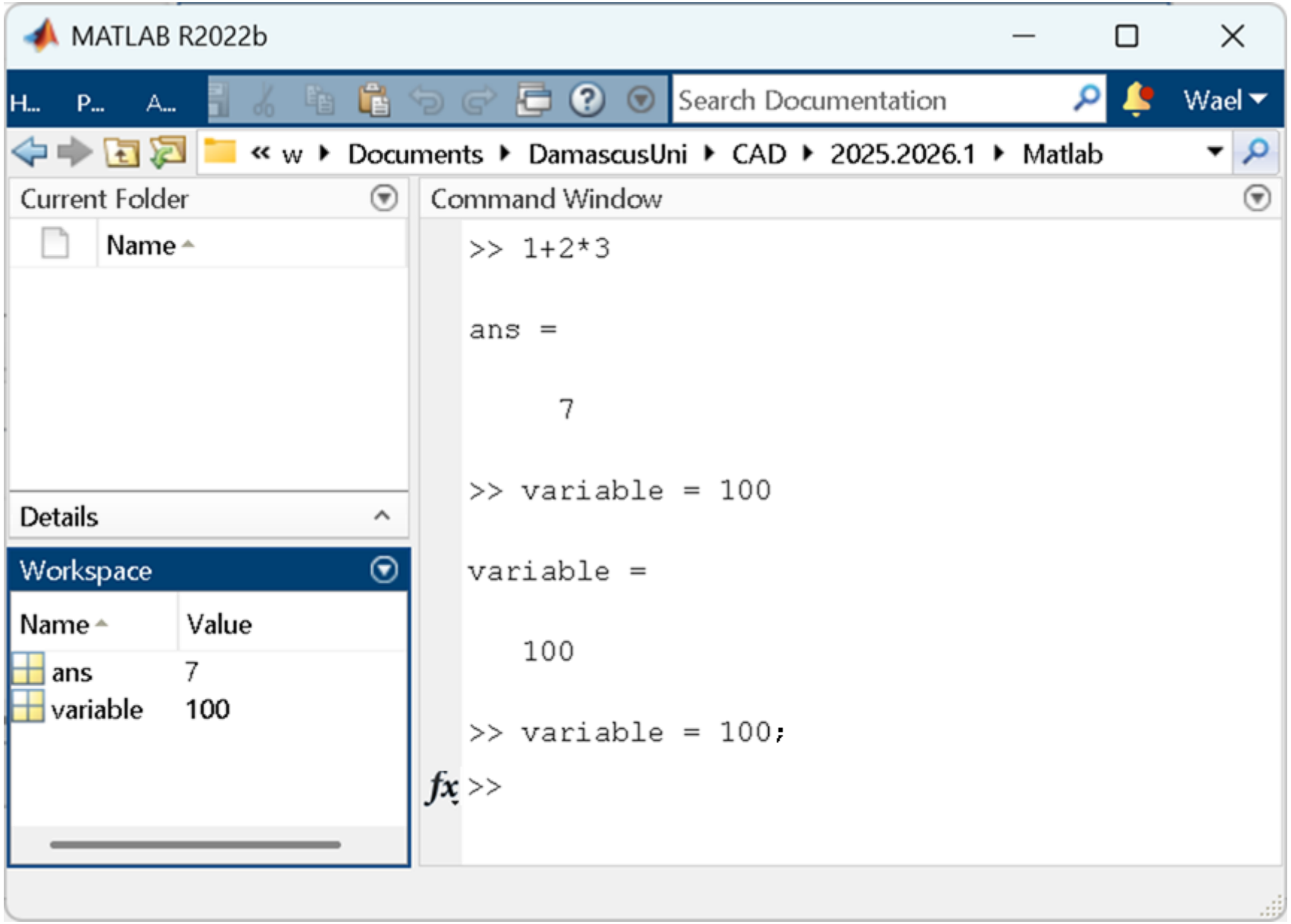
Programming Languages

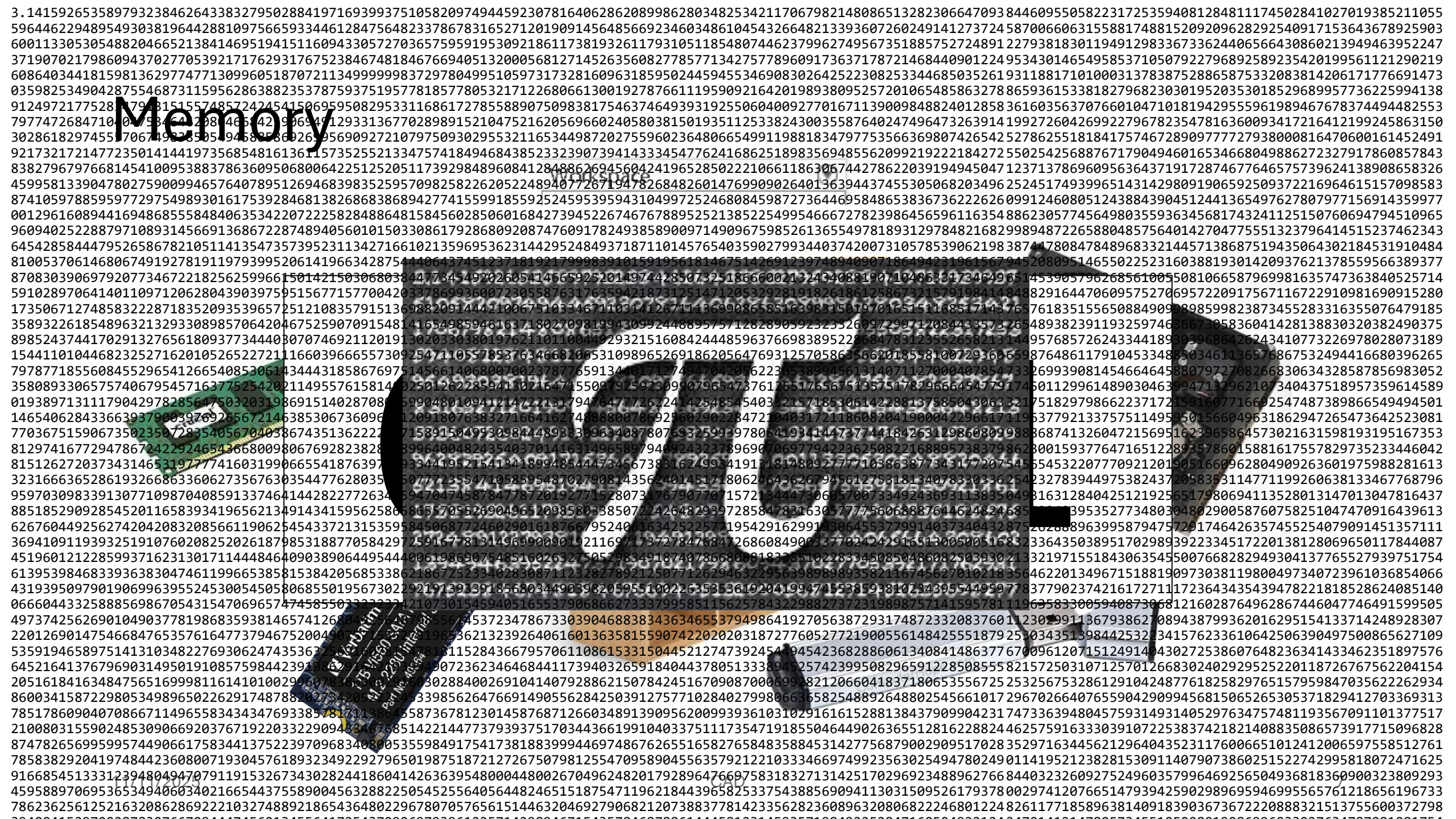
• Fortran	Formula Translation, 1957
• BASIC	Beginner's All-purpose Symbolic Instruction Code, 1964
• C/C++	1972/1985
• MS Visual Basic	1991
• Python	1991
• Java	1995
• C#	2000
• Swift	2010

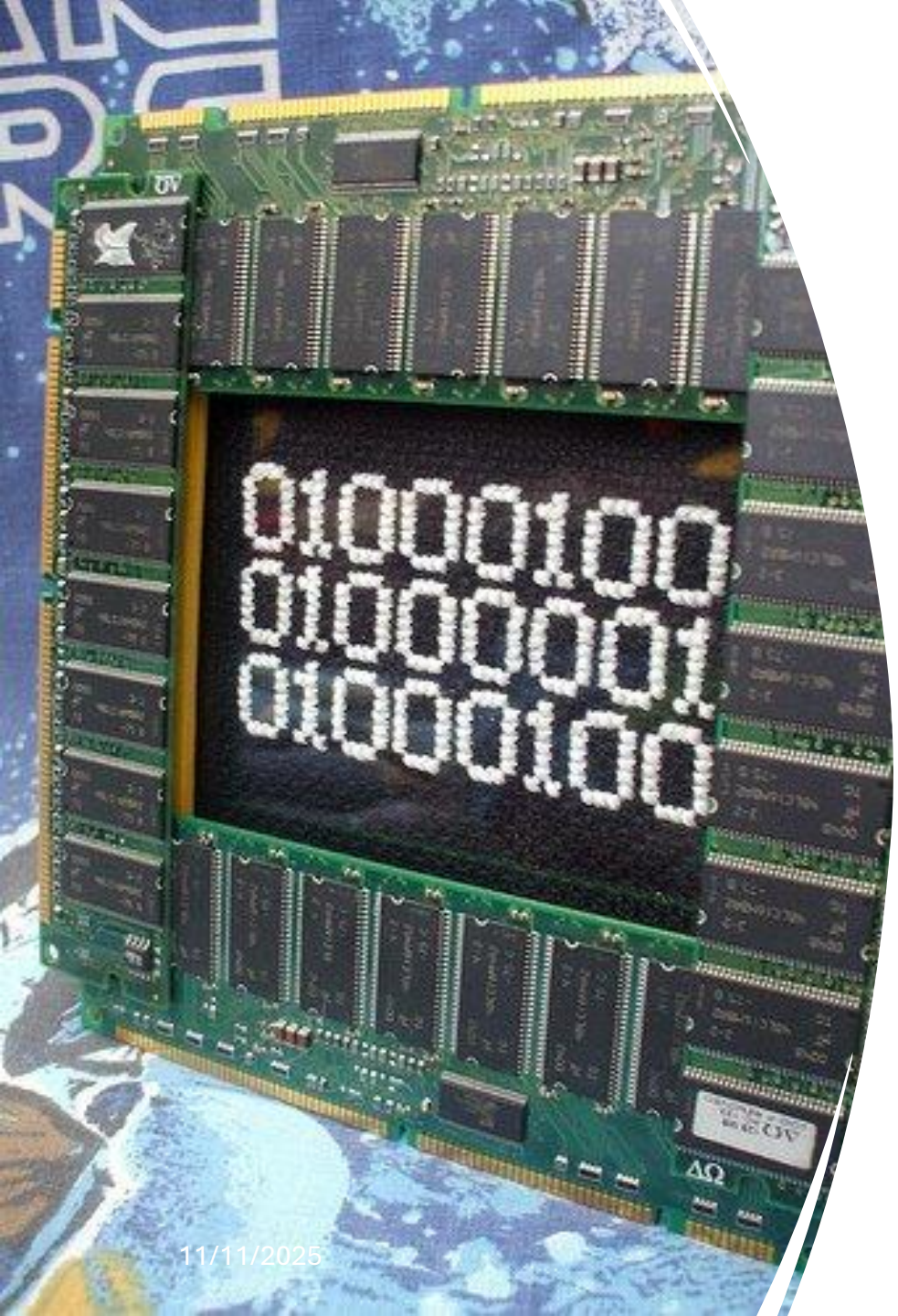
MATLAB

- Matrix Laboratory, 1970s
- Developed by MathWorks
- Matrix manipulations
- Algorithms
- Graph plotting
- Numerical and symbolic engine




MATLAB R2022b
Search Documentation
Wael
« w ▸ Documents ▸ DamascusUni ▸ CAD ▸ 2025.2026.1 ▸ Matlab
Current Folder
Name
Details
Workspace
Name
Value
ans
7
variable
100
Command Window
>> 1+2*3
ans =
7
>> variable = 100
variable =
100
>> variable = 100;
>>


Memory
Workspace
CAD



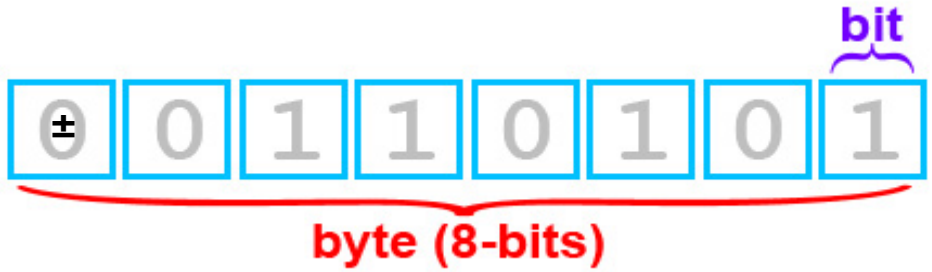
Bit vs Byte

- Bit or binary digit 0/1
- 2 Bit 0/1 0/1
- 3 Bit 0/1 0/1 0/1

2^3 values = 8

0:	**0**	**0**	**0**
1:	**0**	**0**	**1**
2:	**0**	**1**	**0**
3:	**0**	**1**	**1**
4:	**1**	**0**	**0**
5:	**1**	**0**	**1**
6:	**1**	**1**	**0**
7:	**1**	**1**	**1**

Numbers in Memory



- 1 Byte variable
  - Unsigned Integer 0 ~ 255 (2^8=256)

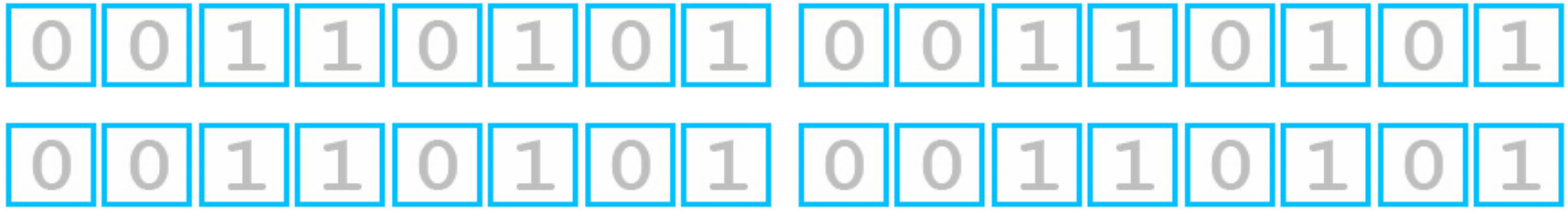
    1111 1111 Bin = 255 Dec
  - Signed integer -128 ~ +127 (2^7=128)

    111 1111 Bin = +127 Dec

Numbers in Memory

- 2 Byte integer (short)
  - Unsigned Integer 0 ~ 65535 (2^{16}=65536)
  - Signed Integer –32768 ~ 32767 (2^{15}=32768)

Numbers in Memory



- 32-bit signed integer
- `intmax = +2’147’483’647`

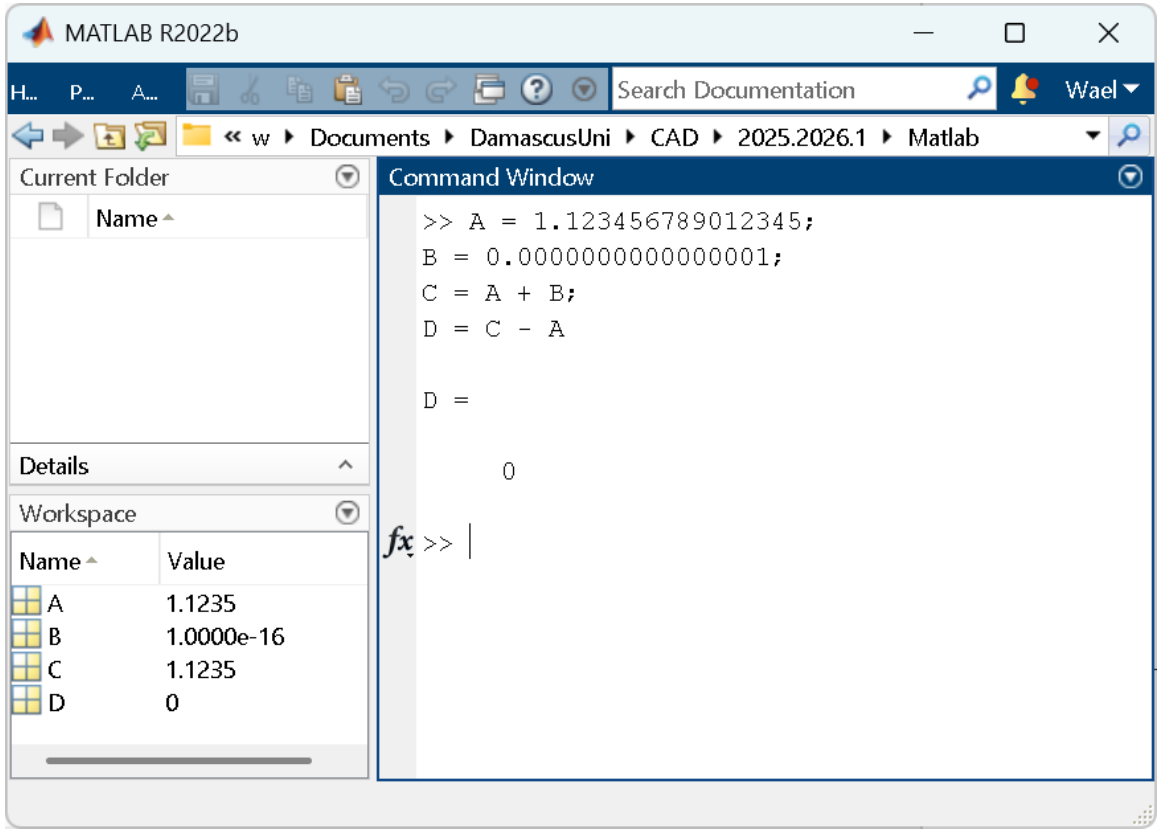
Numbers in Memory

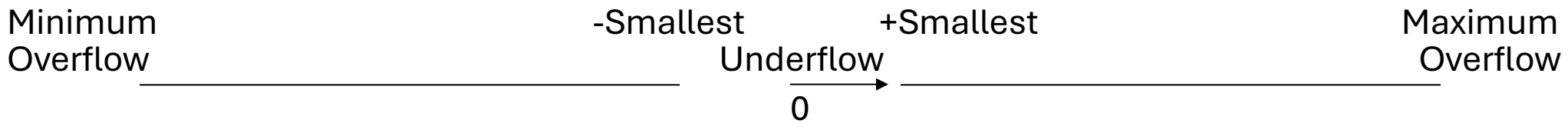
8 Byte double (default in MATLAB)

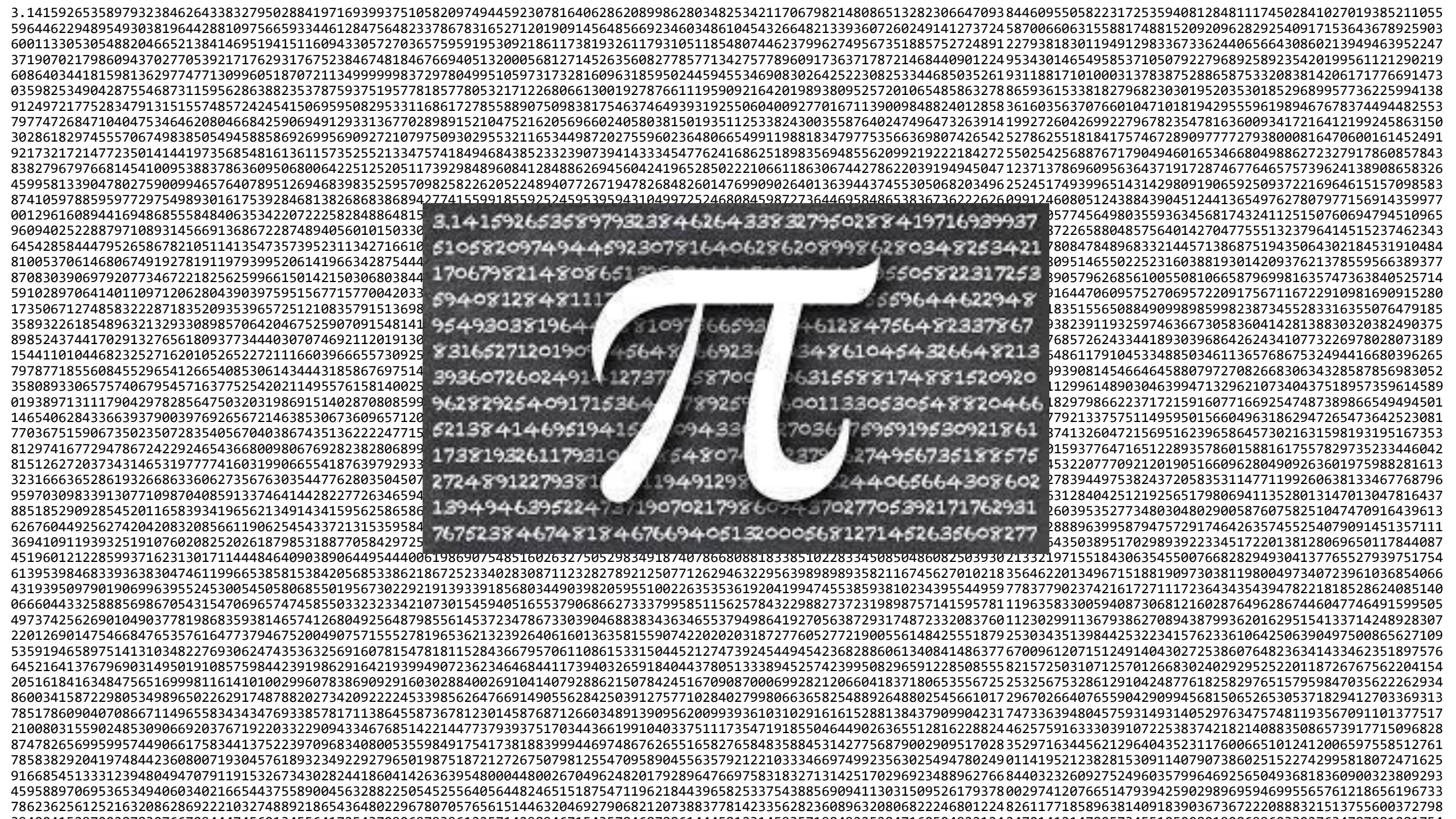
- Accuracy 16 digits:

```
A = 1.123456789012345
B = 0.0000000000000001   (1E-16)
C = A + B      = A
D = C - A      = A - A
D = ?          = 0
```

- Maximum/Minimum ±1.8 E+308
- Smallest ±2.2 E−308





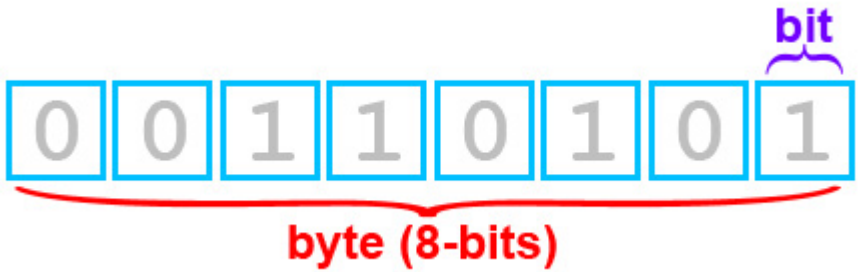


Memory

- Input ‘`Hello`’ to memory
- `Hello = H + e + l + l + o`

ASCII

- The **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange
- `1 byte = 8 bit`
- $2^8 = 256$
- `256` symbols



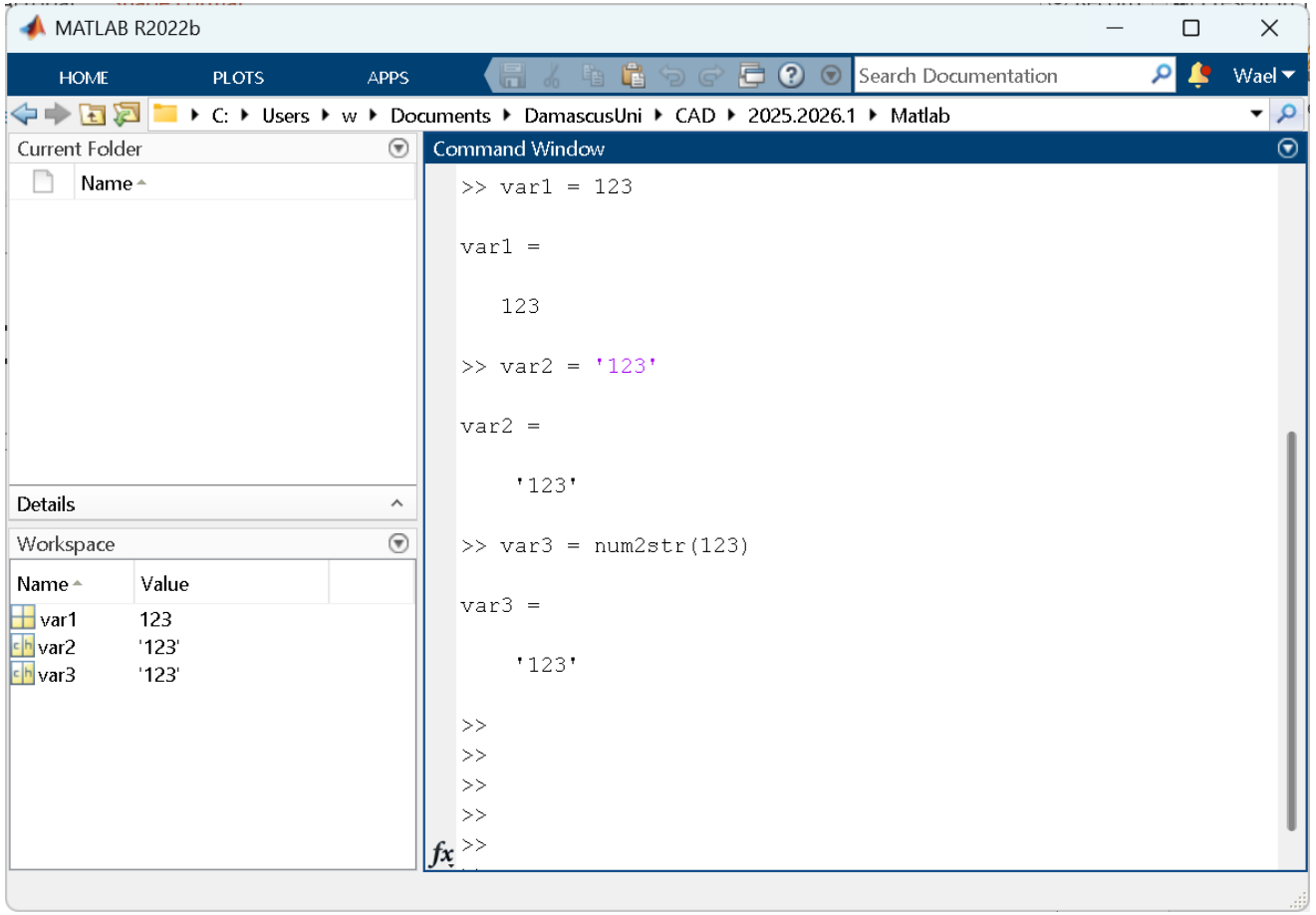
DEC	Symbol	DEC	Symbol	DEC	Symbol	DEC	Symbol	DEC	Symbol	DEC	Symbol	DEC	Symbol	DEC	Symbol
0	NUL	32		64	@	96	`	128	€	160		192	À	224	à
1	SOH	33	!	65	A	97	a	129		161	¡	193	Á	225	á
2	STX	34	"	66	B	98	b	130	‚	162	¢	194	Â	226	â
3	ETX	35	#	67	C	99	c	131	ƒ	163	£	195	Ã	227	ã
4	EOT	36	$	68	D	100	d	132	„	164	¤	196	Ä	228	ä
5	ENQ	37	%	69	E	101	e	133	…	165	¥	197	Å	229	å
6	ACK	38	&	70	F	102	f	134	†	166	¦	198	Æ	230	æ
7	BEL	39	'	71	G	103	g	135	‡	167	§	199	Ç	231	ç
8	BS	40	(	72	H	104	h	136	ˆ	168	¨	200	È	232	è
9	HT	41	)	73	I	105	i	137	‰	169	©	201	É	233	é
10	LF	42	*	74	J	106	j	138	Š	170	ª	202	Ê	234	ê
11	VT	43	+	75	K	107	k	139	‹	171	«	203	Ë	235	ë
12	FF	44	,	76	L	108	l	140	Œ	172	¬	204	Ì	236	ì
13	CR	45	-	77	M	109	m	141		173	-	205	Í	237	í
14	SO	46	.	78	N	110	n	142	Ž	174	®	206	Î	238	î
15	SI	47	/	79	O	111	o	143		175	¯	207	Ï	239	ï
16	DLE	48	0	80	P	112	p	144		176	°	208	Đ	240	ð
17	DC1	49	1	81	Q	113	q	145	‘	177	±	209	Ñ	241	ñ
18	DC2	50	2	82	R	114	r	146	’	178	²	210	Ò	242	ò
19	DC3	51	3	83	S	115	s	147	“	179	³	211	Ó	243	ó
20	DC4	52	4	84	T	116	t	148	”	180	´	212	Ô	244	ô
21	NAK	53	5	85	U	117	u	149	•	181	µ	213	Õ	245	õ
22	SYN	54	6	86	V	118	v	150	–	182	¶	214	Ö	246	ö
23	ETB	55	7	87	W	119	w	151	—	183	·	215	×	247	÷
24	CAN	56	8	88	X	120	x	152	˜	184	¸	216	Ø	248	ø
25	EM	57	9	89	Y	121	y	153	™	185	¹	217	Ù	249	ù
26	SUB	58	:	90	Z	122	z	154	š	186	º	218	Ú	250	ú
27	ESC	59	;	91	[	123	{	155	›	187	»	219	Û	251	û
28	FS	60	<	92	\	124	|	156	œ	188	¼	220	Ü	252	ü
29	GS	61	=	93	]	125	}	157		189	½	221	Ý	253	ý
30	RS	62	>	94	^	126	~	158	ž	190	¾	222	Þ	254	þ
31	US	63	?	95	_	127		159	Ÿ	191	¿	223	ß	255	ÿ

Memory

- Input ‘Hello’ to memory
- Hello = H + e + l + l + o
- 01001000 01100101 01101100 01101100 01101111

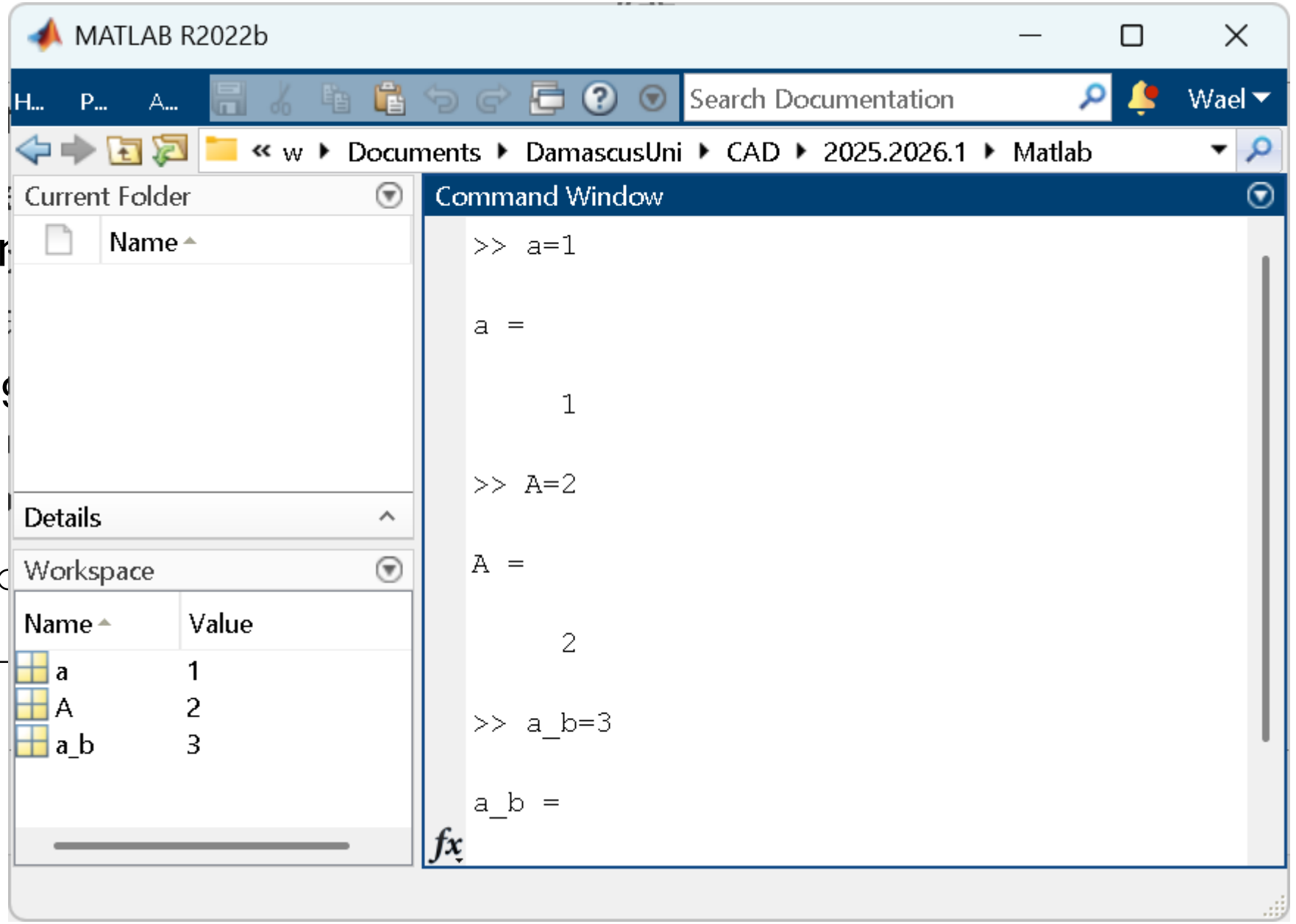
Numbers vs. Strings

- `1`
- `'1'`
- `num2str`



Variables

- Reserved locations ir
- Names:
  - Including a-z, A-z, 0-9
  - Starting a-z, A-z
  - Case sensitive
  - CamelCase: `Sectio`
  - No space `section_`



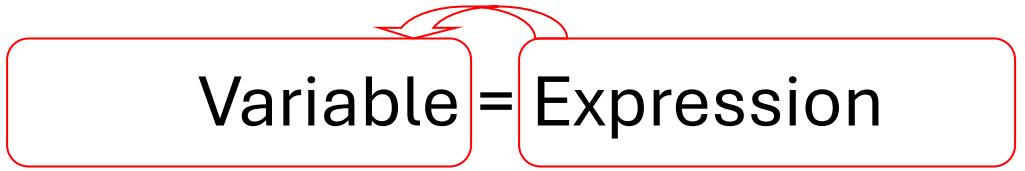
Special Variables

- ans
- pi
- inf
- NaN
- i (or j)
- realmin = 2.2251e-308
- realmax = 1.7977e+308
- intmax = 2147483647

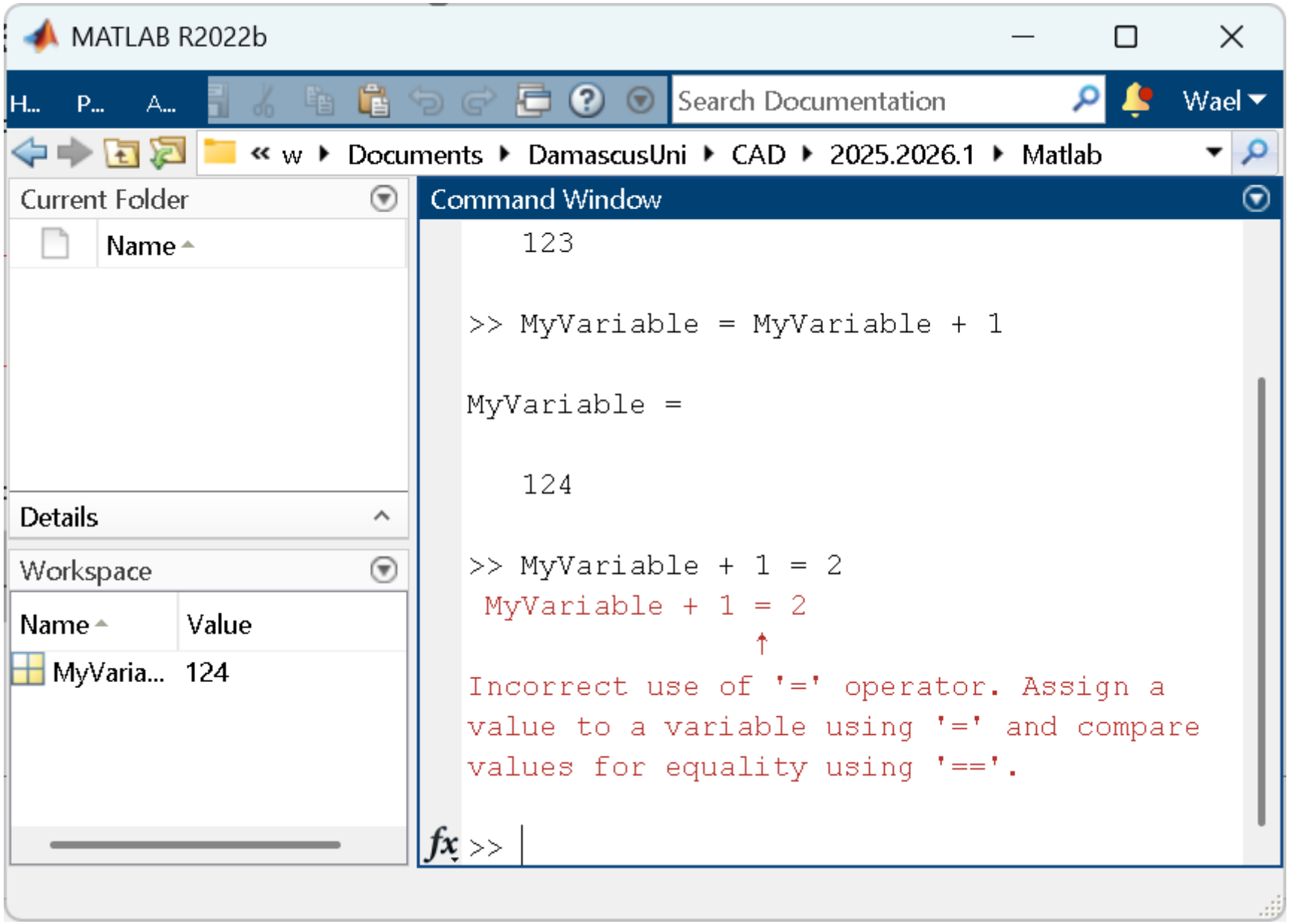
```
>> pi

ans =

    3.1416

>> 1/0

ans =

   Inf

>> -1/0

ans =

  -Inf

>> 0/0

ans =

   NaN

>> i + 2i

ans =

   0.0000 + 3.0000i
```

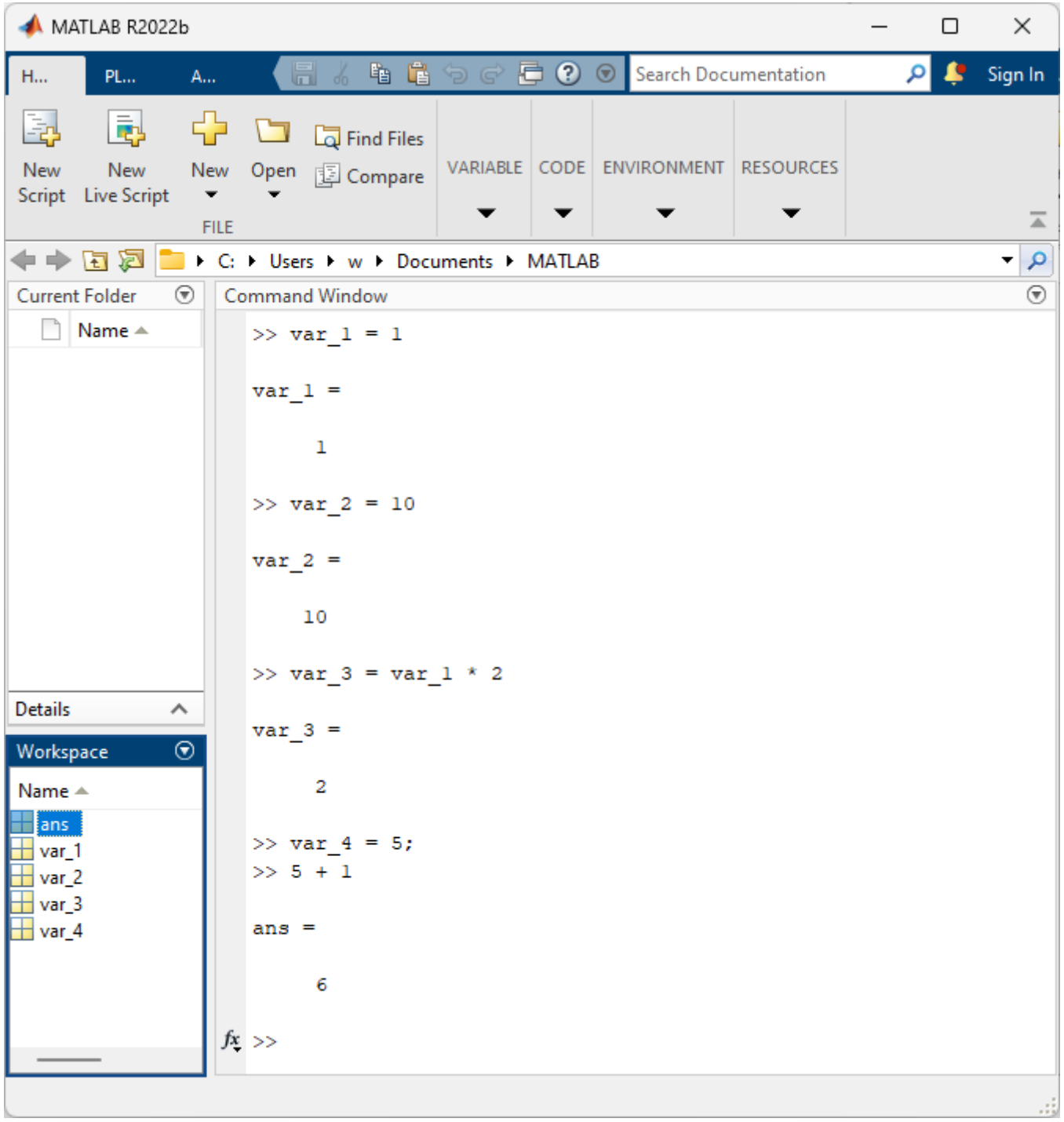
Operators

- Assignments operator =
- `MyVariable = 123`



- `MyVariable = MyVariable + 1`
- `MyVariable + 1 = 2`
- Assignments vs. Equal


MATLAB R2022b
Search Documentation
Wael
« w ▸ Documents ▸ DamascusUni ▸ CAD ▸ 2025.2026.1 ▸ Matlab
Current Folder
Name
Details
Workspace
Name
Value
MyVaria...
124
Command Window
123
>> MyVariable = MyVariable + 1
MyVariable =
124
>> MyVariable + 1 = 2
MyVariable + 1 = 2
↑
Incorrect use of '=' operator. Assign a value to a variable using '=' and compare values for equality using '=='.
>>


MATLAB R2022b
H...
PL...
A...
Search Documentation
Sign In
New Script
New Live Script
New
Open
Find Files
Compare
FILE
VARIABLE
CODE
ENVIRONMENT
RESOURCES
C: ▸ Users ▸ w ▸ Documents ▸ MATLAB
Current Folder
Name
Details
Workspace
Name
ans
var_1
var_2
var_3
var_4
Command Window
>> var_1 = 1
var_1 =
1
>> var_2 = 10
var_2 =
10
>> var_3 = var_1 * 2
var_3 =
2
>> var_4 = 5;
>> 5 + 1
ans =
6
>>

MATLAB Notes

- `clc` Clear command window
- `clear` Clear memory
- `who` List current variables
- `whos` List current variables, long form
- `help` Help for functions in Command Window
- `doc` Reference page in Help browser
- `lookfor` Search for keyword

Important Notes

```
>> help clc
 clc    Clear command window.
    clc clears the command window and homes the cursor.

    See also home.

    Documentation for clc

>> help clear
 clear  Clear variables and functions from memory.
    clear removes all variables from the workspace.
    clear VARIABLES does the same thing.
    clear GLOBAL removes all global variables.
    clear FUNCTIONS removes all compiled MATLAB and MEX-functions.
    Calling clear FUNCTIONS decreases code performance and is usually unnecessary.
    For more information, see the clear Reference page.
```

Arithmetic Operators

- + − × / ^
- Right division (forward slash, slash /)
- Left division (backward slash, backslash \)
    - $B / A = B \times A^{-1}$
    - $A \setminus B = A^{-1} \times B$

    - $A \times X = B$
    - $X = A \setminus B$

Scripts

- M file
- Commands
- Comments
- Input
- Display

MATLAB R2022b

HO... PL... APPS EDIT... PUB... VIEW

Search Documentation

Wael

C: ▸ Users ▸ w ▸ Documents ▸ DamascusUni ▸ CAD ▸ 2025.2026.1 ▸ Matlab

Current Folder

Name

script_file.m

Editor - C:\Users\w\Documents\DamascusUni\CAD\2025.2026.1\Matlab\script_file.m

script_file.m

```
1   var1 = 1;
2   var2 = var1 + 1;
3
4   % increment
5   var2 = var2 + 1;
```

Details

Workspace

Name	Value
var1	1
var2	3

Command Window

```
>> script_file
>>
```

Zoom: 175% UTF-8 CRLF script Ln 5 Col 17

```
1  % MATLAB script
2  a = 3;
3  b = 4;
4  c = sqrt(a^2 + b^2);
5  disp(c)
```

```
a = 3; b = 4;
```

```
a = 3, b = 4
```

Command Window

```
>> script_ex1
     5
```

```
1  a = input('First lenght? ');
2  b = input('Second lenght? ');
3  c = sqrt(a^2 + b^2);
4  disp('Result = ')
5  disp(c)
```

Command Window

```
First lenght? 3
Second lenght? 4
Result =
     5
```

Functions

- Set of steps
- Input argument(s) (optional)
- [Output argument(s)] (optional)

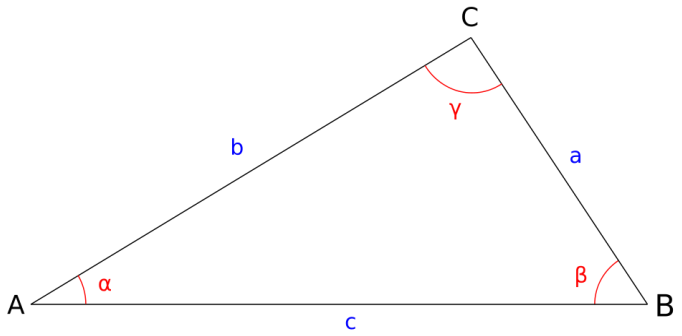
Function Syntax

```
function [output1, output2] = func(input1, input2)
% Comment
code
```

- Function name, same as .m file name.
- Current directory
- Comments for help
- Set output

Function Example

$$c^2 = a^2 + b^2 - 2 \times a \times b \times \cos(\gamma)$$



- Define inputs
- Calculate output

cosinerule_ex.m

```
1  function c = cosinerule_ex(a, b, gamma)
2  % Law of cosines
3  % x = cosinerule_ex(3, 4, pi/2)
4
5  c = sqrt(a^2 + b^2 - 2*a*b*cos(gamma));
6
```

$$c^2 = a^2 + b^2 - 2 \times a \times b \times \cos(\gamma)$$

```
function [output1 output2] = func(input1, input2)
```

Command Window

```
>> cosinerule_ex(3,4,pi/2)

ans =

     5
```

Built-in Functions

• `abs(x)`	Absolute value
• `sqrt(x)`	Square root
• `round(x)`	Rounds towards nearest integer
• `floor(x)`	Rounds towards minus infinity
• `ceil(x)`	Rounds towards plus infinity
• `fix(x)`	Truncates toward 0
• `sign(x)`	Signum function
• `rem(x,y)`	Remainder after division
• `exp(x)`	Exponential
• `log(x)`	Natural logarithm
• `log10(x)`	Common (base 10) logarithm
• `log2(x)`	Base 2 logarithm

Trigonometry

- `sin, cos, tan, asin…` input angles in radians
- 180 degree = pi radians
- 1 degree = pi/180 radians
- `angle_radians = angle_degrees * (pi/180)`
- `angle_degrees = angle_radians * (180/pi)`
- `rad2deg, deg2rad`
- `sind, cosd`

Matrices

- Every variable is a matrix
- `[1]`
- `[1 2]`
- `[1 2 3; 4 5 6]`
- `[1:3; 4:6]`

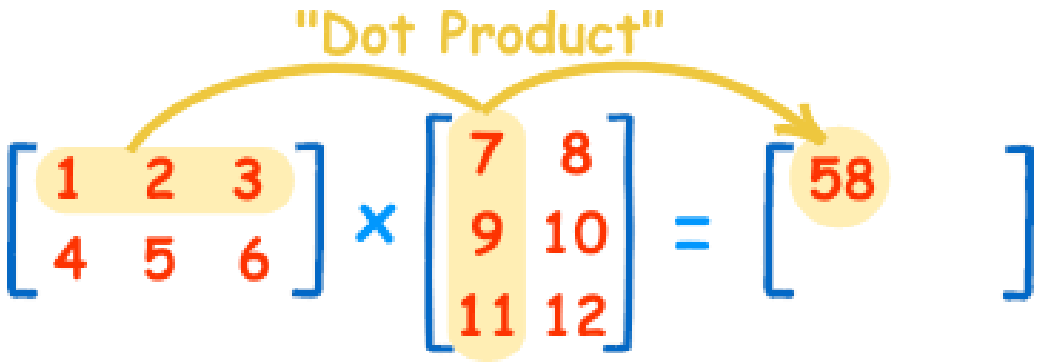
Matrices

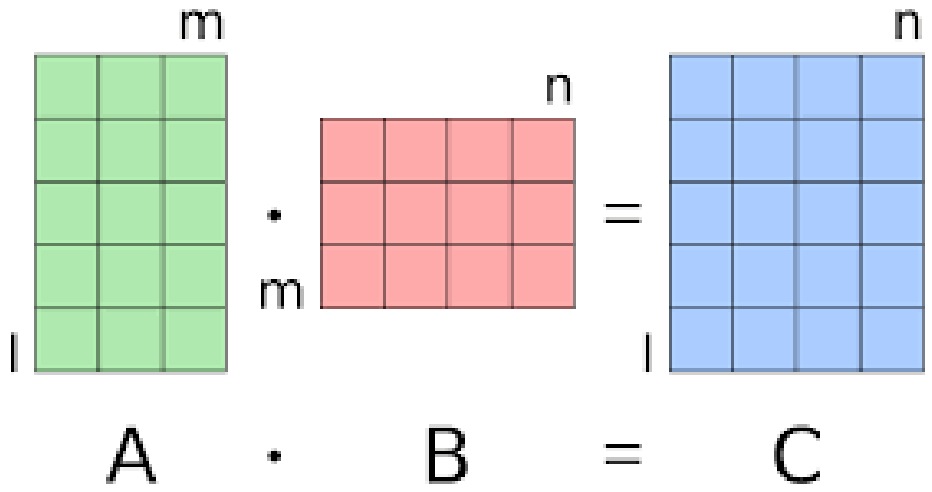
```
A=[1 2 3; 4 5 6; 7 8 9]
```

```
A =
     1     2     3
     4     5     6
     7     8     9
```

```
B = A(1,1)
C = A(2,1:3) = A(2,:)
D = A(3,[2 3]) = A(3,[2:3]) = A(3,2:3)
A(1, 1) = 4
```

Matrices






```
1   A = [1:3;4:6];
2   B = [7 8;9 10;11 12];
3   disp(A*B);
4
```

Command Window

```
>> matrices_ex1
    58    64
   139   154
```