COMPUTER SKILLS

LESSON 3

Valeria Cardellini
cardellini@ing.uniroma2.it

A.Y. 2015/16

Objectives of this lesson

We'll start studying MATLAB. We'll discuss:

- Variables and assignment statements
- Characters and encoding
- Numerical data types
  - Integer
  - Floating point
- Numerical expressions

Octave: open source alternative to MATLAB

- GNU Octave www.octave.org
- High-level interpreted programming language, primarily intended for numerical computations
- Provides capabilities for numerical solution of linear and nonlinear problems, and for performing other numerical experiments
- Provides extensive graphics capabilities for data visualization and manipulation
- Normally used through its interactive command line interface
- **The Octave language is quite similar to MATLAB** so that most programs are easily portable from MATLAB to Octave and viceversa

Variables and assignment statements

- "Once a programmer has understood the use of variables, he has understood the essence of programming"

  (E. Dijkstra)

- **Variable**: a memory location (i.e., a piece of your computer's memory) which is given a symbolic ***name*** (and a type), and can store a quantity of information referred to as a ***value***
  - We use variables to store the results of a computation and use those results later in our program
  - Variables in computer programming are frequently given long names to make them relatively descriptive of their use
- **Assignment statement**: a statement that stores a value into a variable's memory location

Variables and assignment statements

- **Assignment statement**

```
variablename = expression
```

= is the assignment operator; it does not mean equal

Syntax similar to conventional algebra but a different meaning! Consider:

```
z = x + y
z = 4*x - y
```

Algebraic interpretation: implied relationship between x and y by addition and simplification:

```
2y = 3x
```

Programming: I changed my mind about what I wanted to store in the variable `z` (`x` and `y` must have values)

Variable name

- Must always be on the left and the expression on the right
- Can be any combination of uppercase and lower case alphabetic letters, digits and the special characters _ and $ but it cannot have a space
- Must begin with a letter of the alphabet

```
myfirstvariable
myFirstVar
my_1st_var
1stvarofmine
```

NO

- MATLAB is case sensitive: variables `mynum`, `MYNUM` and `Mynum` are different!
- Reserved words or keywords cannot be used as variable names

Useful commands for variables

- `namelengthmax`
  - maximum variable name length
- `who`
  - currently defined variables
- `whos`
  - detailed information on currently defined variables
- `clear`
  - remove all variables from the workspace
- `clear variablename`
  - remove the variable from the workspace
- `clear variablename1 variablename2 …`
  - remove the listed variables from the workspace

Data typing

- How a programming language treats data stored in a variable
  - untyped
  - typed, e.g. C, Fortran, Java
    - both name and type must be specified
    - weak typing
    - strong typing

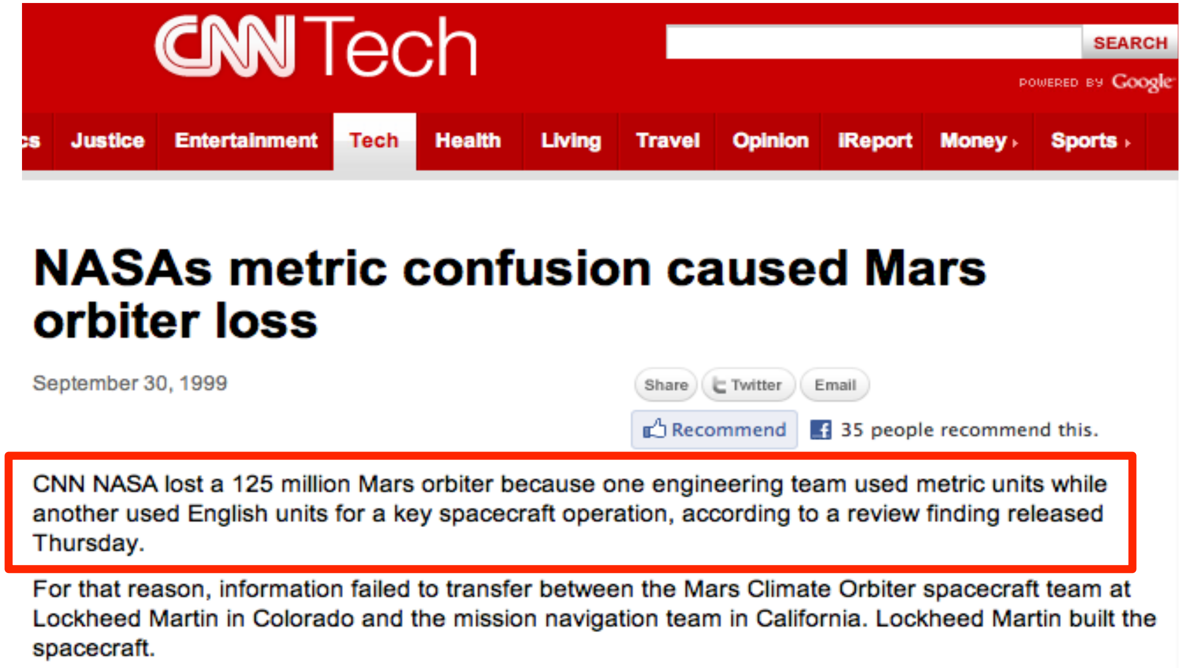
a Meter;

b Second;

c,d MeterPerSecond;

a+b, b+a,c+a **KO**

c=a*b + d **OK**



NASAs metric confusion caused Mars orbiter loss

September 30, 1999

CNN NASA lost a 125 million Mars orbiter because one engineering team used metric units while another used English units for a key spacecraft operation, according to a review finding released Thursday.

For that reason, information failed to transfer between the Mars Climate Orbiter spacecraft team at Lockheed Martin in Colorado and the mission navigation team in California. Lockheed Martin built the spacecraft.

Data typing

- MATLAB is both typed and untyped
- Untyped case

```
>> a = [2 3]     % a is a vector with two elements
a =
     2     3
>> x = 0.5*a
x =
   1.0000e+00   1.5000e+00
```

- Typed case

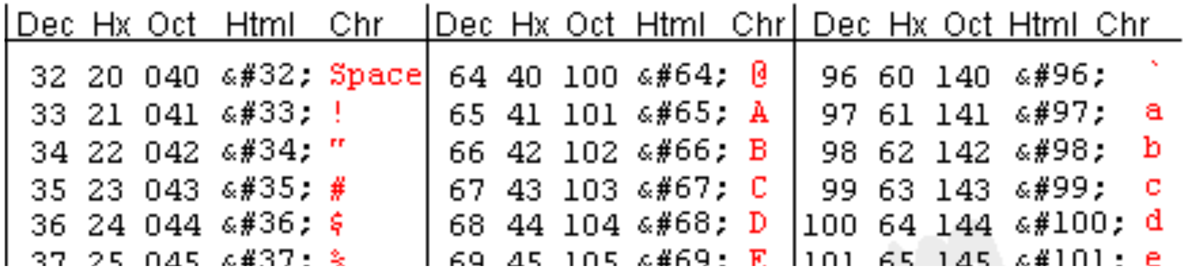
```
>> ai = uint8(a) %For more info on the uint8 function
                 %type «help uint8» on the console
                 % or browse the online documentation
ai =
    2    3
>> x = 0.5*ai
x =
    1    2
```

Untyped use of MATLAB

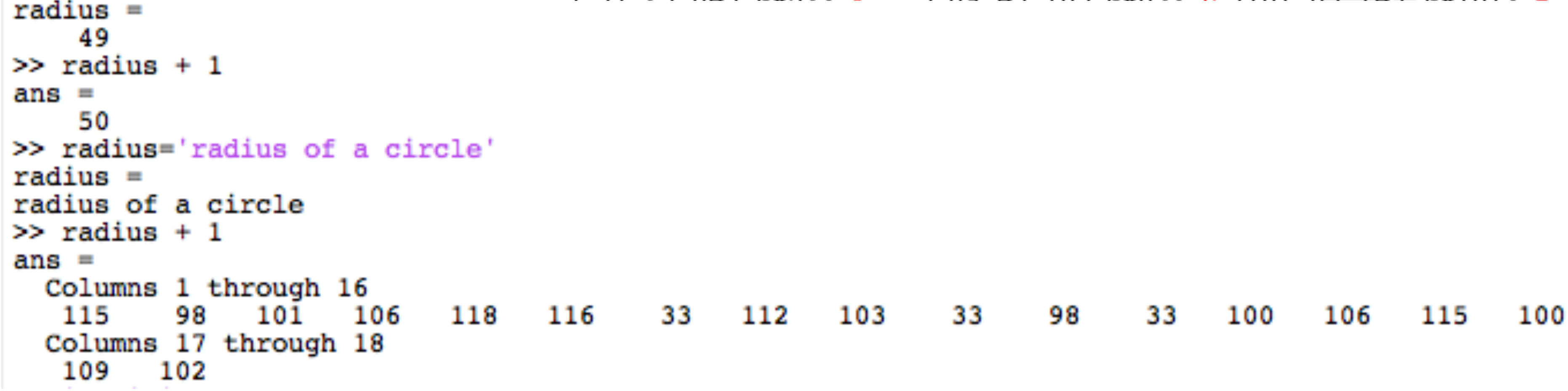
```
>> letter = ‘A’

letter = A

>> letter = letter+1

letter =  66
```

letter is char (character)

letter is double



Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
32	20	040	&#32;	Space	64	40	100	&#64;	@	96	60	140	&#96;	`
33	21	041	&#33;	!	65	41	101	&#65;	A	97	61	141	&#97;	a
34	22	042	&#34;	"	66	42	102	&#66;	B	98	62	142	&#98;	b
35	23	043	&#35;	#	67	43	103	&#67;	C	99	63	143	&#99;	c
36	24	044	&#36;	$	68	44	104	&#68;	D	100	64	144	&#100;	d
37	25	045	&#37;	%	69	45	105	&#69;	E	101	65	145	&#101;	e

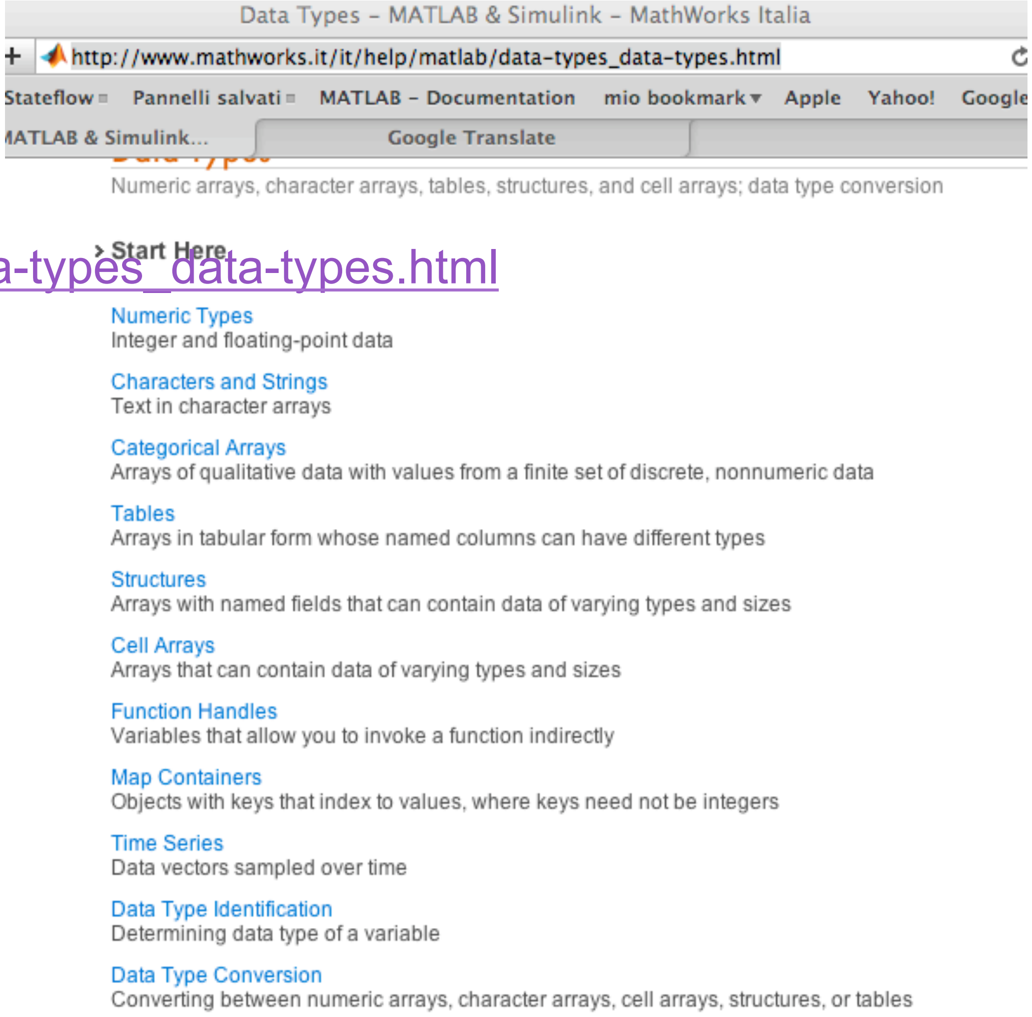


```
>> radius=49
radius =
    49
>> radius + 1
ans =
    50
>> radius='radius of a circle'
radius =
radius of a circle
>> radius + 1
ans =
  Columns 1 through 16
   115    98   101   106   118   116    33   112   103    33    98    33   100   106   115   100
  Columns 17 through 18
   109   102
```

MATLAB data types

Let’s play with types…

www.mathworks.it/it/help/matlab/data-types_data-types.html


Data Types - MATLAB & Simulink - MathWorks Italia
http://www.mathworks.it/it/help/matlab/data-types_data-types.html
Stateflow
Pannelli salvati
MATLAB - Documentation
mio bookmark
Apple
Yahoo!
Google
ATLAB & Simulink...
Google Translate

Numeric arrays, character arrays, tables, structures, and cell arrays; data type conversion

› Start Here

Numeric Types
Integer and floating-point data

Characters and Strings
Text in character arrays

Categorical Arrays
Arrays of qualitative data with values from a finite set of discrete, nonnumeric data

Tables
Arrays in tabular form whose named columns can have different types

Structures
Arrays with named fields that can contain data of varying types and sizes

Cell Arrays
Arrays that can contain data of varying types and sizes

Function Handles
Variables that allow you to invoke a function indirectly

Map Containers
Objects with keys that index to values, where keys need not be integers

Time Series
Data vectors sampled over time

Data Type Identification
Determining data type of a variable

Data Type Conversion
Converting between numeric arrays, character arrays, cell arrays, structures, or tables

Default data type

- By default, MATLAB stores all numeric variables as double-precision floating-point values
- Additional data types store text, integer or single-precision values, or a combination of related data in a single variable

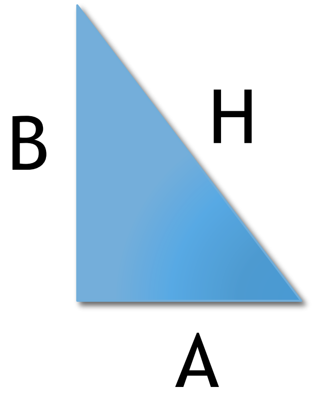
```
>> thisNumber = 42.0 %Variable: instance of an object storing a value
thisNumber =
    42
>> whos thisNumber
  Name                Size                Bytes  Class     Attributes
  thisNumber          1x1                     8  double
```

The first script

Write a script that computes the hypotenuse of a right triangle applying the Pythagora’s theorem:

$$H^2 = A^2 + B^2$$

where A and B are the sides adjacent to the right angle, and H is the hypotenuse



```
clear
clc %clear the terminal screen

A = 3; % the first side of a triangle
B = 4; % the second side of a triangle
hypSq = A^2 + B^2; % the square of the hypotenuse
H = sqrt(hypSq) % the answer
```

Characters and encoding

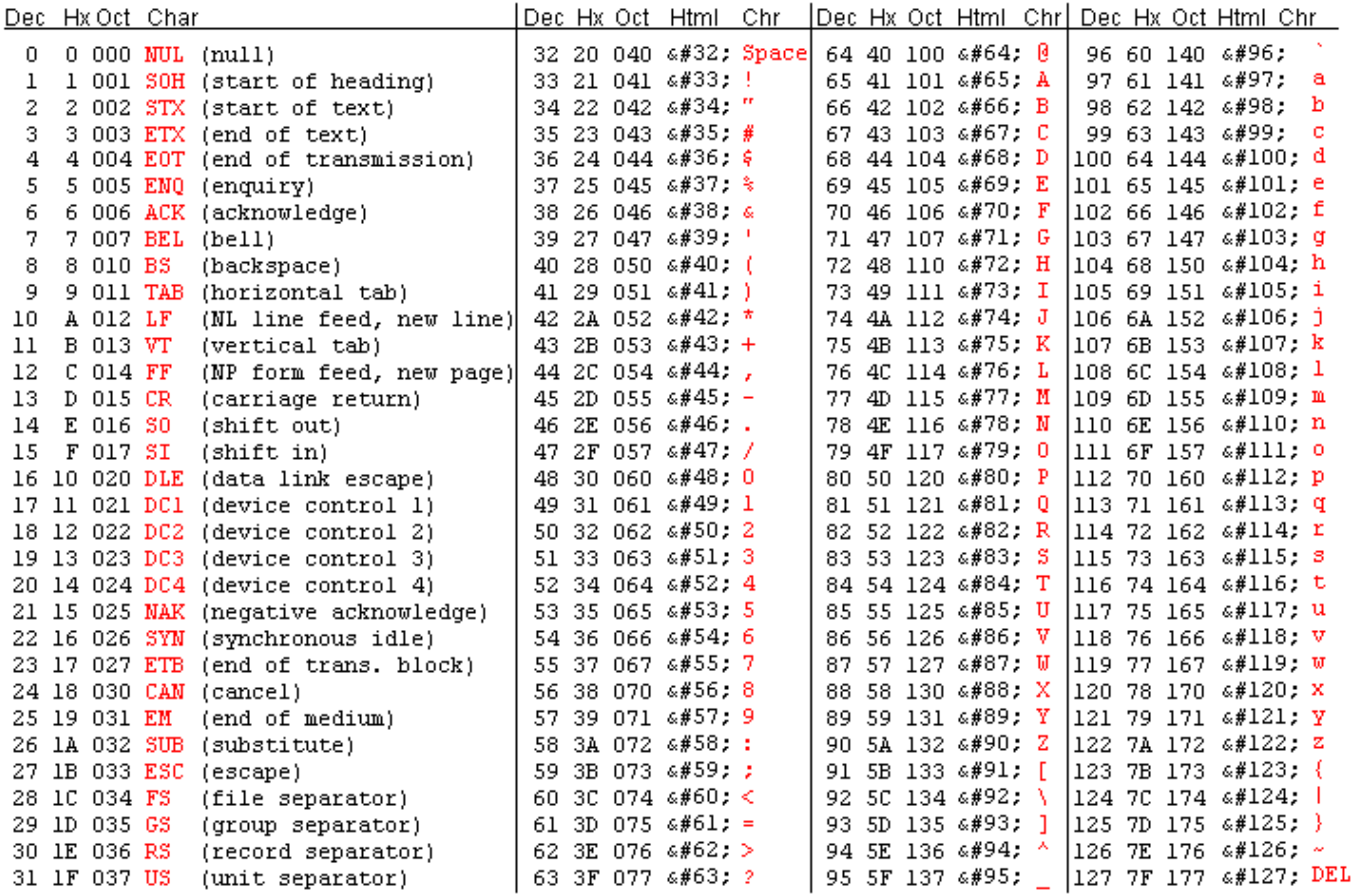
- A character in MATLAB is represented using single quotes
  - `'a', 'x', '3'`
- Characters are ordered using a character encoding
- The most common character encoding is ASCII
- In MATLAB there are numeric functions to convert a character to its equivalent numerical value

```
>> numequiv = double('a')
numequiv =  97
```

Additional reading: the ASCII code

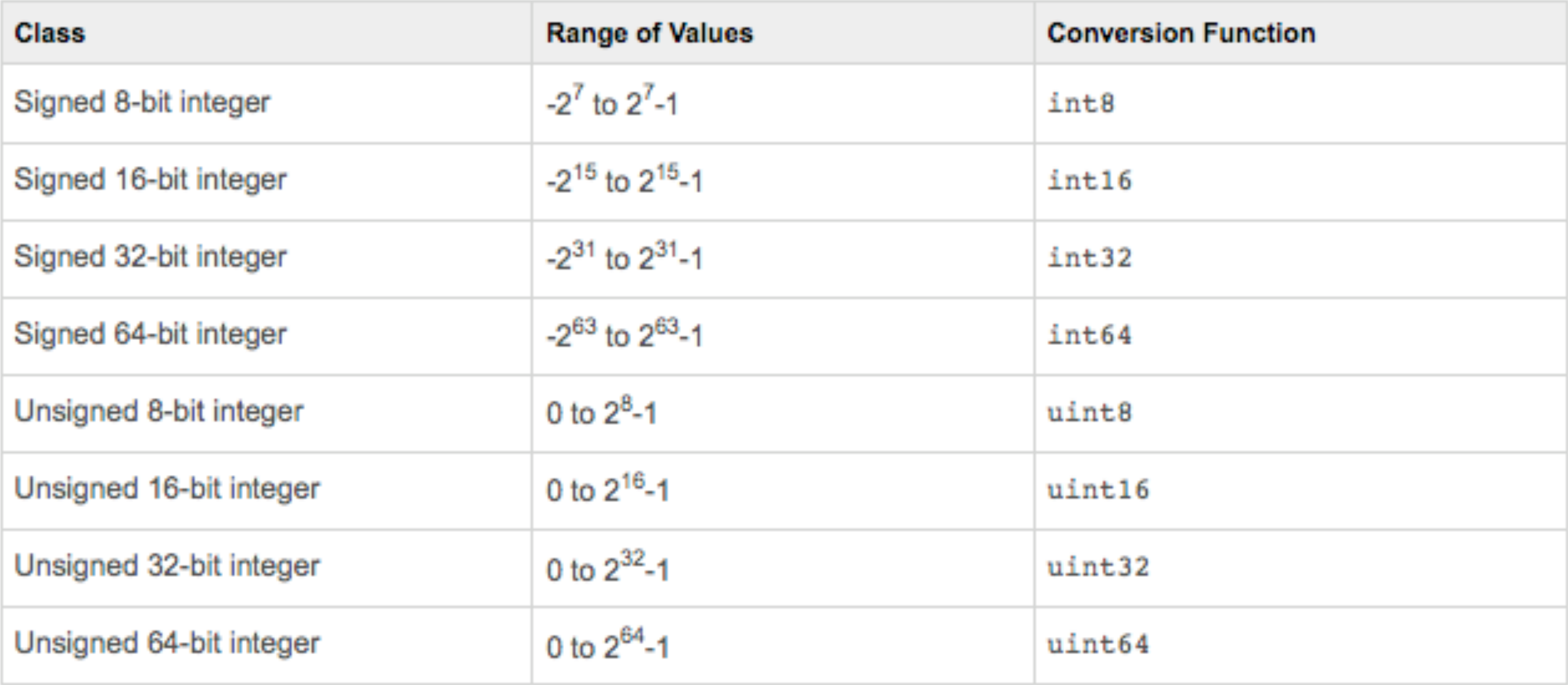
- ASCII stands for American Standard Code for Information Interchange
- Computers can only understand numbers, so an ASCII code is the numerical representation of a character such as 'a' or '@’
- ASCII was developed a long time ago and now the non-printing characters are rarely used for their original purpose
- On the next slide there is the ASCII character table, which includes descriptions of the first 32 non-printing characters. ASCII was actually designed for use with teletypes and so the descriptions are somewhat obscure. If someone says they want your CV however in ASCII format, all this means is they want 'plain' text with no formatting such as tabs, bold or underscoring - the raw format that any computer can understand. This is usually so they can easily import the file into their own applications without issues. Using Windows, notepad.exe creates ASCII text, or in MS Word you can save a file as 'text only'

Additional reading: the ASCII table



Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	&#32;	Space	64	40	100	&#64;	@	96	60	140	&#96;	`
1	1	001	SOH (start of heading)	33	21	041	&#33;	!	65	41	101	&#65;	A	97	61	141	&#97;	a
2	2	002	STX (start of text)	34	22	042	&#34;	"	66	42	102	&#66;	B	98	62	142	&#98;	b
3	3	003	ETX (end of text)	35	23	043	&#35;	#	67	43	103	&#67;	C	99	63	143	&#99;	c
4	4	004	EOT (end of transmission)	36	24	044	&#36;	$	68	44	104	&#68;	D	100	64	144	&#100;	d
5	5	005	ENQ (enquiry)	37	25	045	&#37;	%	69	45	105	&#69;	E	101	65	145	&#101;	e
6	6	006	ACK (acknowledge)	38	26	046	&#38;	&	70	46	106	&#70;	F	102	66	146	&#102;	f
7	7	007	BEL (bell)	39	27	047	&#39;	'	71	47	107	&#71;	G	103	67	147	&#103;	g
8	8	010	BS (backspace)	40	28	050	&#40;	(	72	48	110	&#72;	H	104	68	150	&#104;	h
9	9	011	TAB (horizontal tab)	41	29	051	&#41;	)	73	49	111	&#73;	I	105	69	151	&#105;	i
10	A	012	LF (NL line feed, new line)	42	2A	052	&#42;	*	74	4A	112	&#74;	J	106	6A	152	&#106;	j
11	B	013	VT (vertical tab)	43	2B	053	&#43;	+	75	4B	113	&#75;	K	107	6B	153	&#107;	k
12	C	014	FF (NP form feed, new page)	44	2C	054	&#44;	,	76	4C	114	&#76;	L	108	6C	154	&#108;	l
13	D	015	CR (carriage return)	45	2D	055	&#45;	-	77	4D	115	&#77;	M	109	6D	155	&#109;	m
14	E	016	SO (shift out)	46	2E	056	&#46;	.	78	4E	116	&#78;	N	110	6E	156	&#110;	n
15	F	017	SI (shift in)	47	2F	057	&#47;	/	79	4F	117	&#79;	O	111	6F	157	&#111;	o
16	10	020	DLE (data link escape)	48	30	060	&#48;	0	80	50	120	&#80;	P	112	70	160	&#112;	p
17	11	021	DC1 (device control 1)	49	31	061	&#49;	1	81	51	121	&#81;	Q	113	71	161	&#113;	q
18	12	022	DC2 (device control 2)	50	32	062	&#50;	2	82	52	122	&#82;	R	114	72	162	&#114;	r
19	13	023	DC3 (device control 3)	51	33	063	&#51;	3	83	53	123	&#83;	S	115	73	163	&#115;	s
20	14	024	DC4 (device control 4)	52	34	064	&#52;	4	84	54	124	&#84;	T	116	74	164	&#116;	t
21	15	025	NAK (negative acknowledge)	53	35	065	&#53;	5	85	55	125	&#85;	U	117	75	165	&#117;	u
22	16	026	SYN (synchronous idle)	54	36	066	&#54;	6	86	56	126	&#86;	V	118	76	166	&#118;	v
23	17	027	ETB (end of trans. block)	55	37	067	&#55;	7	87	57	127	&#87;	W	119	77	167	&#119;	w
24	18	030	CAN (cancel)	56	38	070	&#56;	8	88	58	130	&#88;	X	120	78	170	&#120;	x
25	19	031	EM (end of medium)	57	39	071	&#57;	9	89	59	131	&#89;	Y	121	79	171	&#121;	y
26	1A	032	SUB (substitute)	58	3A	072	&#58;	:	90	5A	132	&#90;	Z	122	7A	172	&#122;	z
27	1B	033	ESC (escape)	59	3B	073	&#59;	;	91	5B	133	&#91;	[	123	7B	173	&#123;	{
28	1C	034	FS (file separator)	60	3C	074	&#60;	<	92	5C	134	&#92;	\	124	7C	174	&#124;	|
29	1D	035	GS (group separator)	61	3D	075	&#61;	=	93	5D	135	&#93;	]	125	7D	175	&#125;	}
30	1E	036	RS (record separator)	62	3E	076	&#62;	>	94	5E	136	&#94;	^	126	7E	176	&#126;	~
31	1F	037	US (unit separator)	63	3F	077	&#63;	?	95	5F	137	&#95;	_	127	7F	177	&#127;	DEL

Integer data type



Class	Range of Values	Conversion Function
Signed 8-bit integer	-2^{7} to 2^{7}-1	`int8`
Signed 16-bit integer	-2^{15} to 2^{15}-1	`int16`
Signed 32-bit integer	-2^{31} to 2^{31}-1	`int32`
Signed 64-bit integer	-2^{63} to 2^{63}-1	`int64`
Unsigned 8-bit integer	0 to 2^{8}-1	`uint8`
Unsigned 16-bit integer	0 to 2^{16}-1	`uint16`
Unsigned 32-bit integer	0 to 2^{32}-1	`uint32`
Unsigned 64-bit integer	0 to 2^{64}-1	`uint64`

- To know the range of an integer data type, use the `intmin` and `intmax` built-in functions, e.g., `intmin(‘int16’)`
- MATLAB stores numeric data as double by default: to store data as integer, you need to convert from double to the desired integer type
  - Use one of the conversion functions shown above

From double to integer

- Example: store 325 as a 16-bit signed integer assigned to variable x

```
>> x = int16(325);
```

- If the number being converted to an integer has a fractional part, MATLAB rounds to the nearest integer
- If the fractional part is exactly 0.5, then from the two equally nearby integers, MATLAB chooses the one for which the absolute value is larger in magnitude

```
>> x = 325.499;
>> int16(x)
ans = 325

>> x = x + .001;
>> int16(x)
ans = 326
```

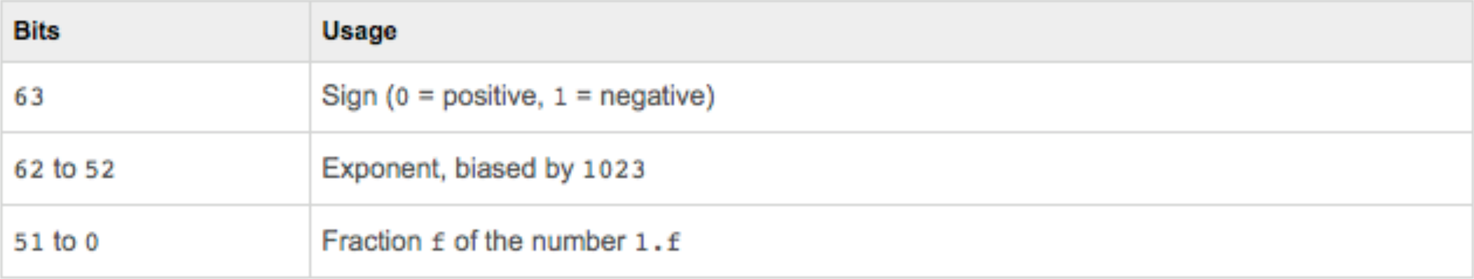
From double to integer: overriding rounding scheme

- If you need to round a number using a rounding scheme other than the default, MATLAB provides four rounding functions:
  - `round`: towards nearest integer
  - `fix`: towards zero
  - `floor`: towards negative infinity
  - `ceil`: towards positive infinity

```
>> x = 325.9;
>> int16(fix(x))
ans = 325
```

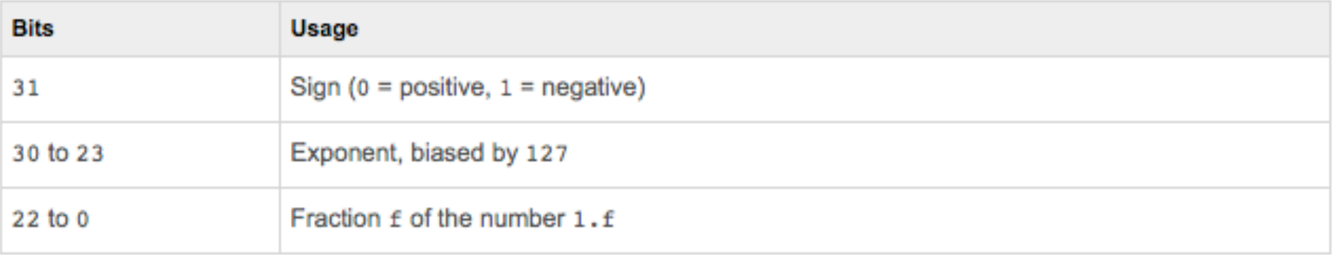
Floating point data type

- Double-precision (or double) data type: IEEE Standard 754 for double precision; any value stored as a double requires 64 bits



Bits	Usage
63	Sign (0 = positive, 1 = negative)
62 to 52	Exponent, biased by 1023
51 to 0	Fraction f of the number 1.f

- Single-precision (or single) data type: IEEE Standard 754 for single precision; any value stored as a single requires 32 bits



Bits	Usage
31	Sign (0 = positive, 1 = negative)
30 to 23	Exponent, biased by 127
22 to 0	Fraction f of the number 1.f

Numbers in single precision are represented with less precision than numbers in double precision

Numerical expressions

- Expressions can be created using values, variables, operators, built-in functions and parentheses

```
>> 2 * sin(1.4)
```

- Operators
  - Two kinds: unary and binary
- Unary operators: operate on a single value
- Binary operators: operate on two values or operands
- Some common operators with numerical expressions:

+	addition
-	negation, subtraction
*	multiplication
/	division (divided by)
\	division (divided into)
^	exponentiation

www.mathworks.it/it/help/matlab/operators-and-elementary-operations.html

Numerical expressions

- Some operators have precedence over others
- Using parentheses can change the precedence

```
>> 4 + 5 * 3
>> (4 + 5) * 3
```

- Operator precedence (from highest to lowest)

()	parentheses
^	exponentiation
-	negation
*, /, \	all multiplication and division
+, -	addition and subtraction

Examples of operator precedence

- `1\2`
- `-5^2`
- `(-5)^2`
- `10-6/2`
- `5*4/2*3`

Relational (or logical) expressions

- Expressions that are true or false (*Boolean* or logical expressions)

>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to
==	equality
~=	inequality

- 0 for false, 1 for true

```
>> 'd'<'g'                  >> ~(2<3)
ans = 1                     ans = 0
```

Review: Boolean algebra

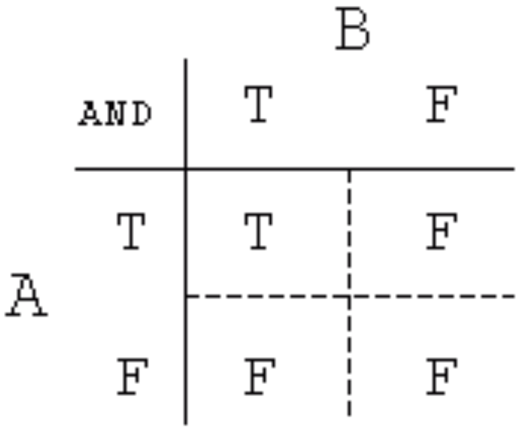
- In mathematics and mathematical logic, Boolean algebra is the subarea of algebra in which the values of the variables are:
  - the truth values *true* (T) and *false* (F), also denoted by 1 and 0 respectively

- The basic operations of Boolean algebra are:
  - the logical conjunction (AND) denoted by ∧
  - the logical disjunction (OR) denoted by ∨
  - the logical negation (NOT) denoted by ¬

- The logical exclusive disjunction OR (XOR) is a logical operator that outputs true whenever both inputs differ

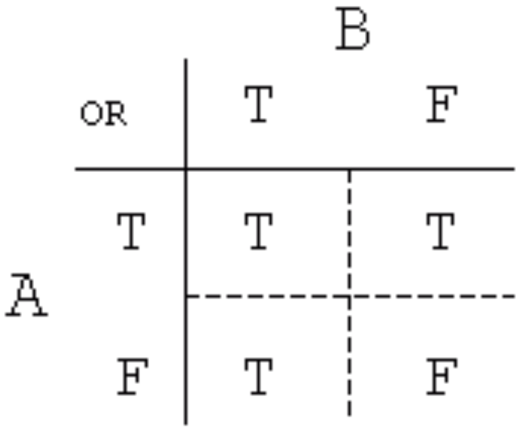
Review: logical operators

AND `&&`



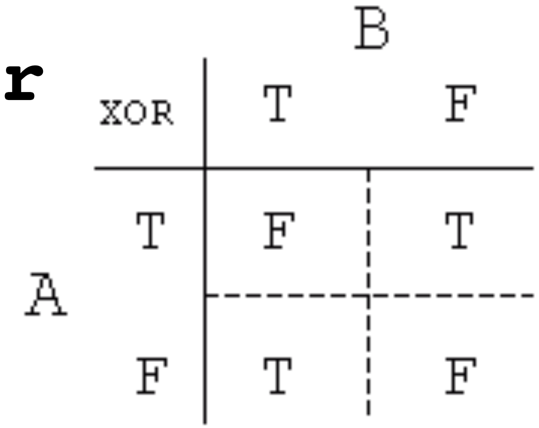
		B	
	AND	T	F
A	T	T	F
	F	F	F

OR `||`



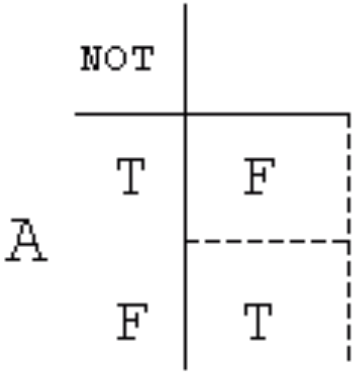
		B	
	OR	T	F
A	T	T	T
	F	T	F

XOR `xor`



		B	
	XOR	T	F
A	T	F	T
	F	T	F

NOT `~`



	NOT	
A	T	F
	F	T

Relational (or logical) expressions in MATLAB

- Logical operators

`||`	or
`&&`	and
`~`	not

```
>> 2 < 4 || 'x' == 'y'
ans = 1
```

- Built-in function `xor`

```
>> xor('c' > 'b', 5 > 2)
ans = 0
```