Computer Programming with MATLAB

MM1CPM - Lecture 4

Computer hardware, Screen output, strings and keyboard input

Dr. Roderick MacKenzie
roderick.mackenzie@nottingham.ac.uk

Autumn 2014

www.mm1cpm.com

Overview of this lecture

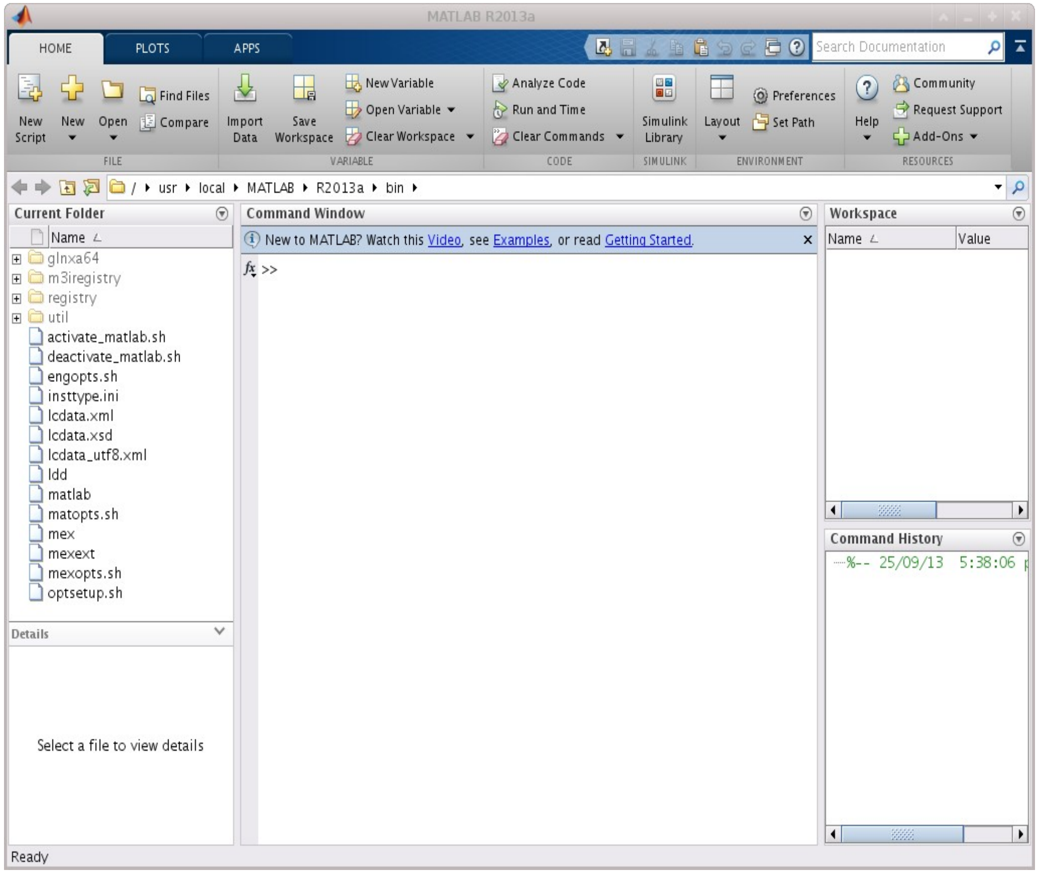
- **Computer fundamentals**
  - **What's in a computer?**
  - Types of computers
  - ASCII code

- Writing to the screen

- Reading text from the keyboard

- Strings in depth

Things I like about the MATLAB programming language



New to MATLAB? Watch this Video, see Examples, or read Getting Started.

```
fx >>
```

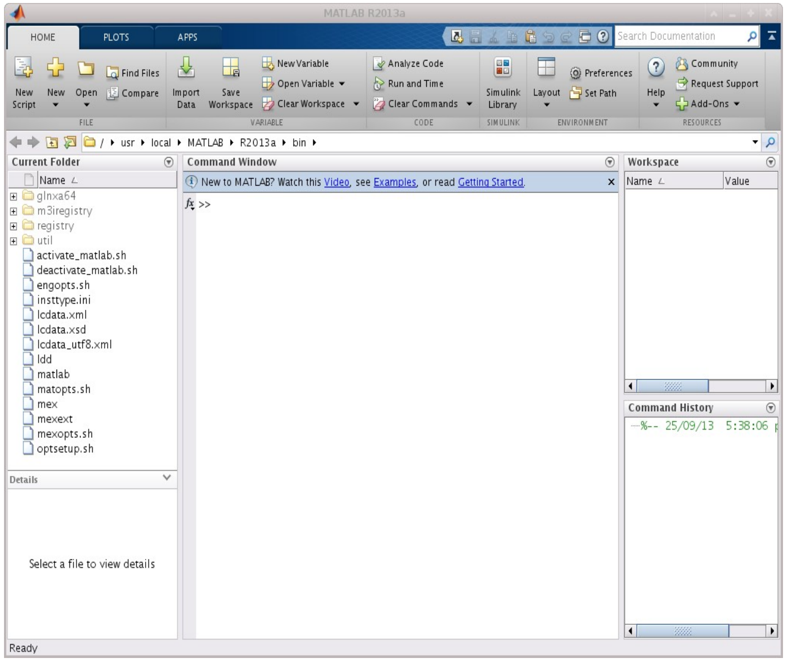
```
%-- 25/09/13 5:38:06
```

Select a file to view details

Ready

- It's very good at handling arrays
- It's good with complex numbers
- It's quite a simple language
- Produces quite nice 3D plots.
- It's generally very good for Engineering

Things I don't like about MATLAB



New to MATLAB? Watch this Video, see Examples, or read Getting Started.

```
fx >>
```

```
--%-- 25/09/13 5:38:06 p
```

Select a file to view details

Ready

- It's not free (python would be free) :(

- It's not as fast as other languages such as C.

- But the main thing I don't like about it is that it gives you this interface and gives you the ***impression*** it is a 'package'.

- But MATLAB is more than a package, **it gives you real control over what the computer is doing.** To really harness this power you need to know a little bit about how a computer works.

What is inside a a typical computer?

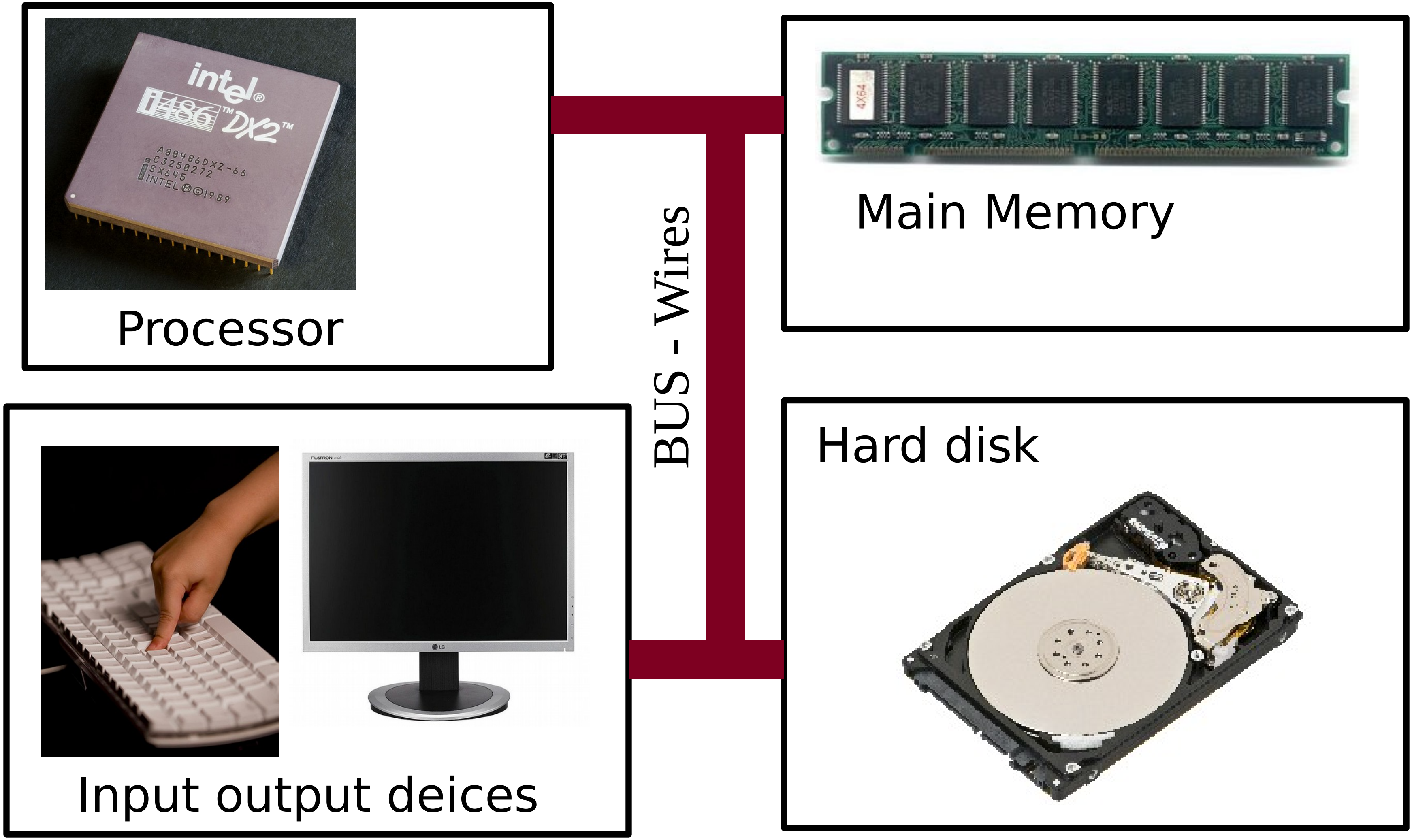
•To really use this power you need to understand how a computer works.

•In the next few slides you are going to learn what the **key components are and what they do.**

•We will be using the **typical PC** as an example because the **components are big but all computers have these basic components**.

The components of a computer

- Let's look at the components one by one.



Processor

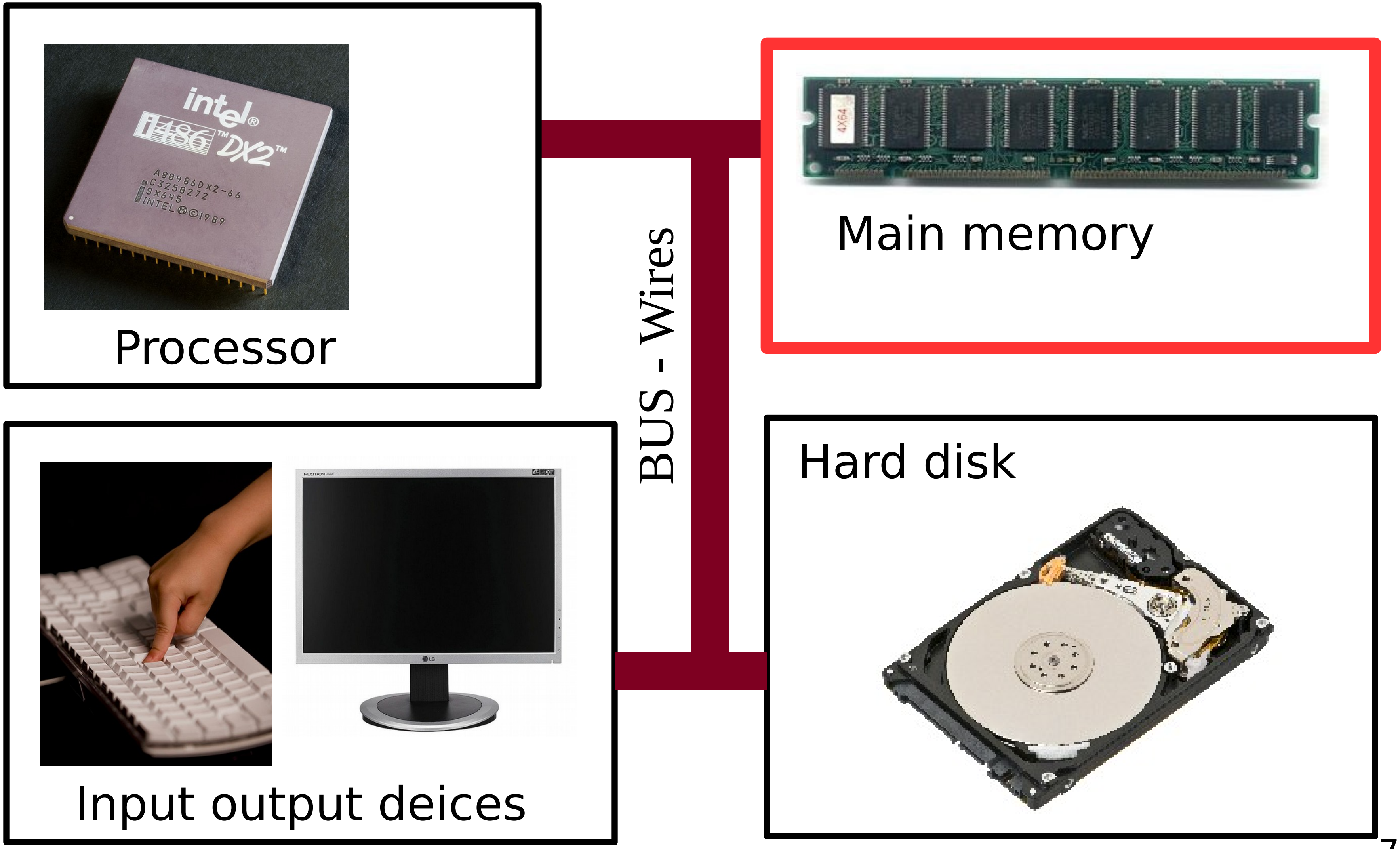
Main Memory

BUS - Wires

Input output deices

Hard disk

The components of a computer:



Processor

BUS - Wires

Main memory

Hard disk

Input output deices

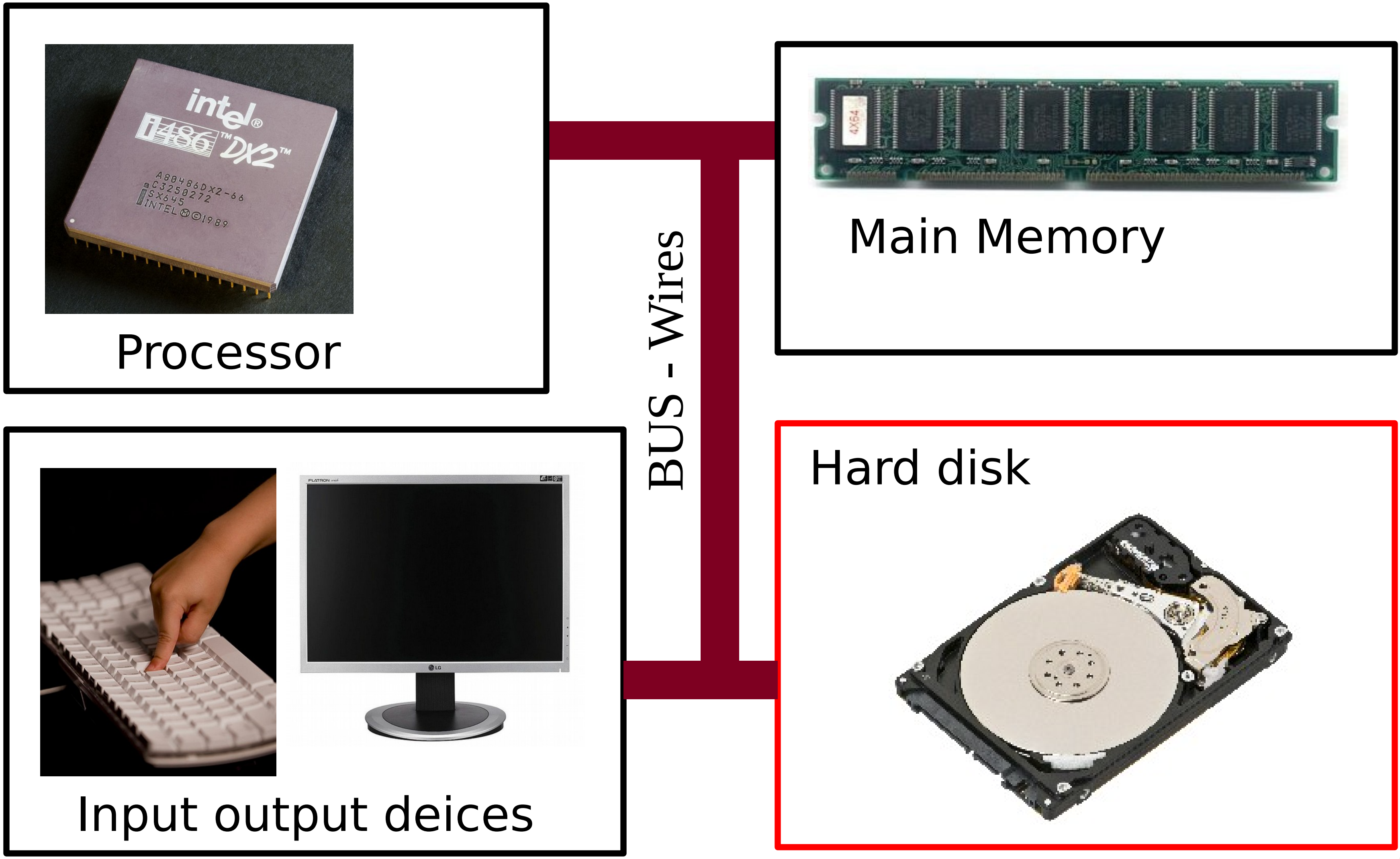
Main memory chips

- Store all information the computer **is currently using**.

- The **computers memory is very fast (1 ns)** but very expensive per Mb of stored information
- The **computer's memory** will only store information whilst the power is on – if you switch off the power it looses all information.

- Any **arrays** or **variables** you define will be stored in the memory.
- The memory also stores your programs/scripts whilst they are running.

The components of a computer:



Processor

Main Memory

BUS - Wires

Hard disk

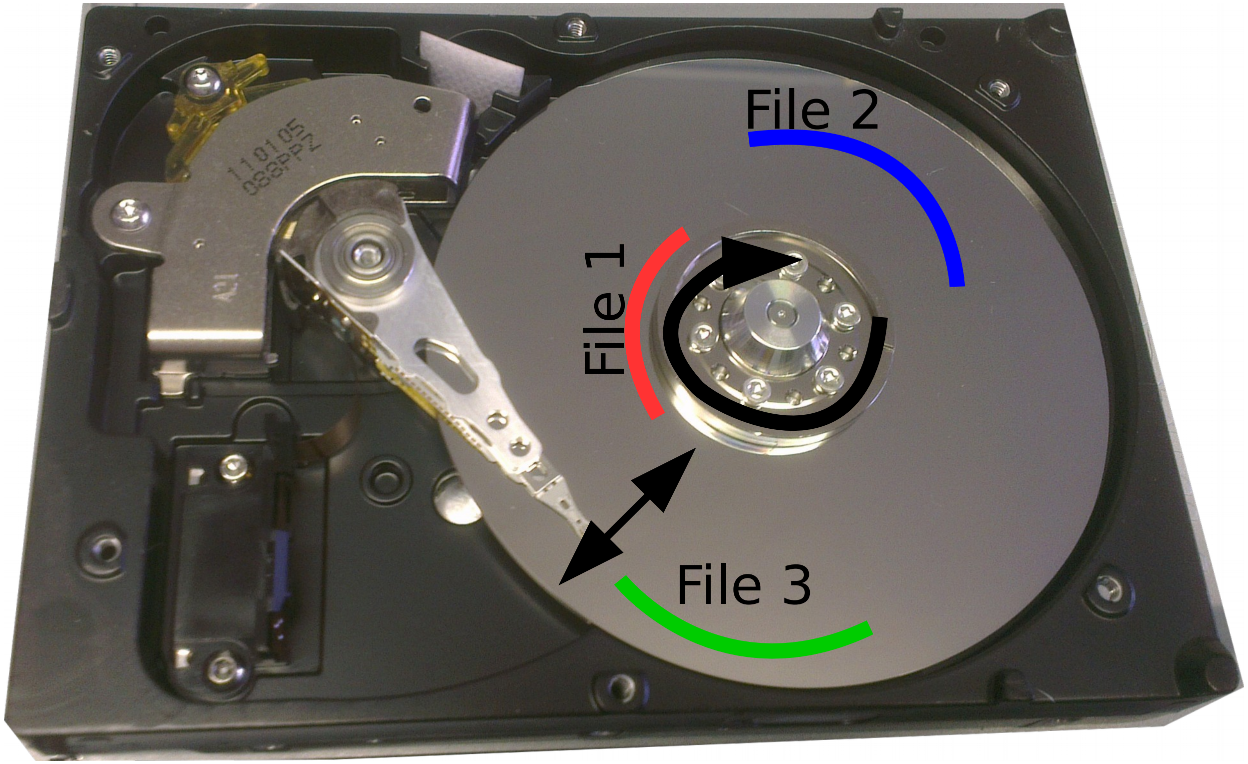
Input output deices

Hard disk

- This can hold a lot of information information while the computer is switched off – programs, word documents etc...
- Hard disks offer very low cost per Mb stored but **very very slow**
- 1 ms access time - 1x10^6 times slower than main memory).

- But why are hard disks so slow?

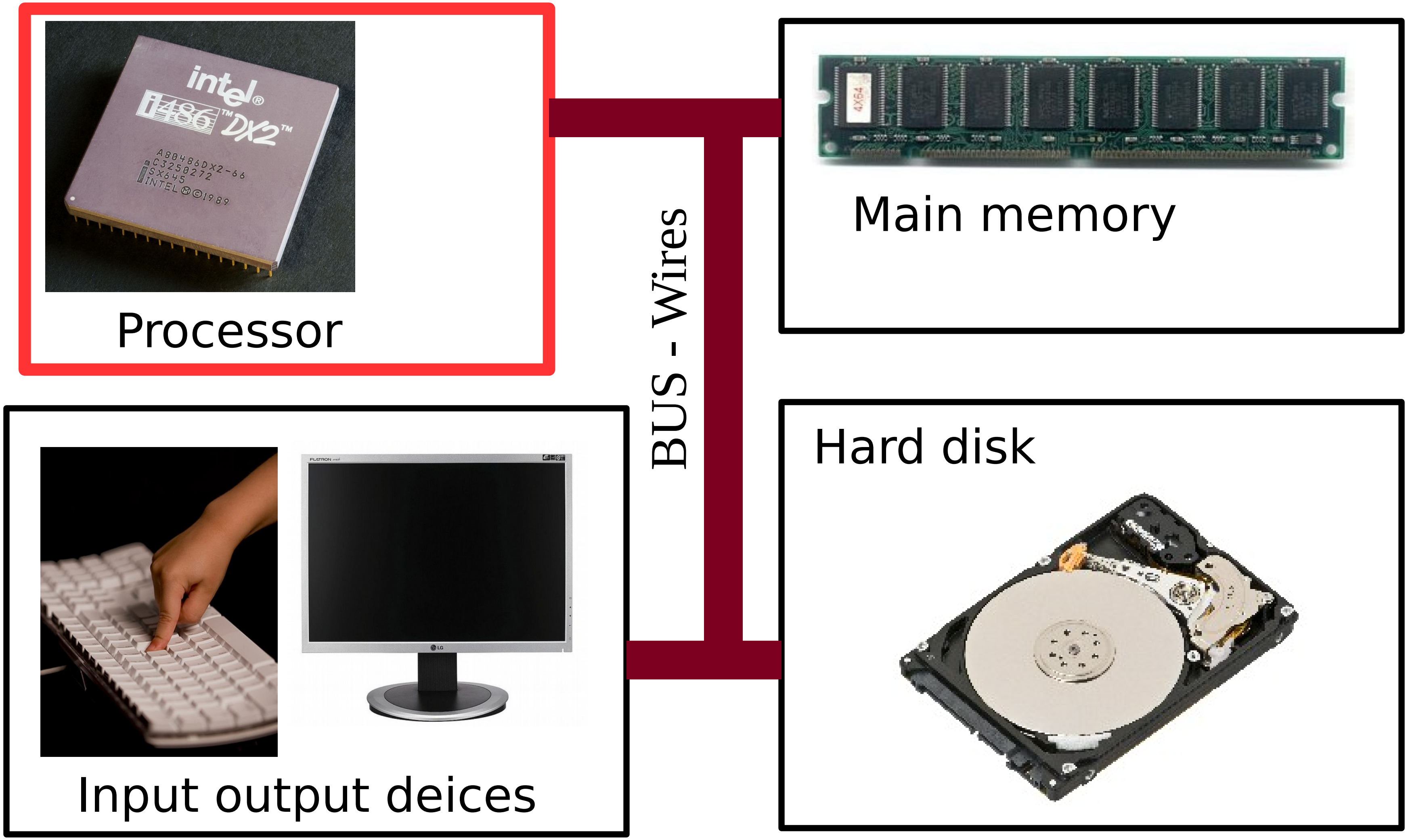
Why is a hard disk slow?



- The files are stored on a rotating magnetic disk – a bit like a record player
- For the computer to read the files, the head must physically move, this takes time.

- **Top programming tip:** If your program is running slowly you are probably using the hard disk to much.

The components of a computer



Processor

BUS - Wires

Main memory

Hard disk

Input output deices

The Processor

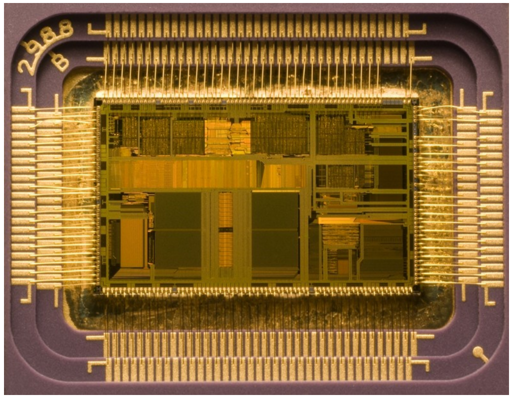
- This is the chip that:
  - Performs all **mathematical operations**
  - **Runs** and understands your **programs line-by-line**.

- When you type anything into the MATLAB:

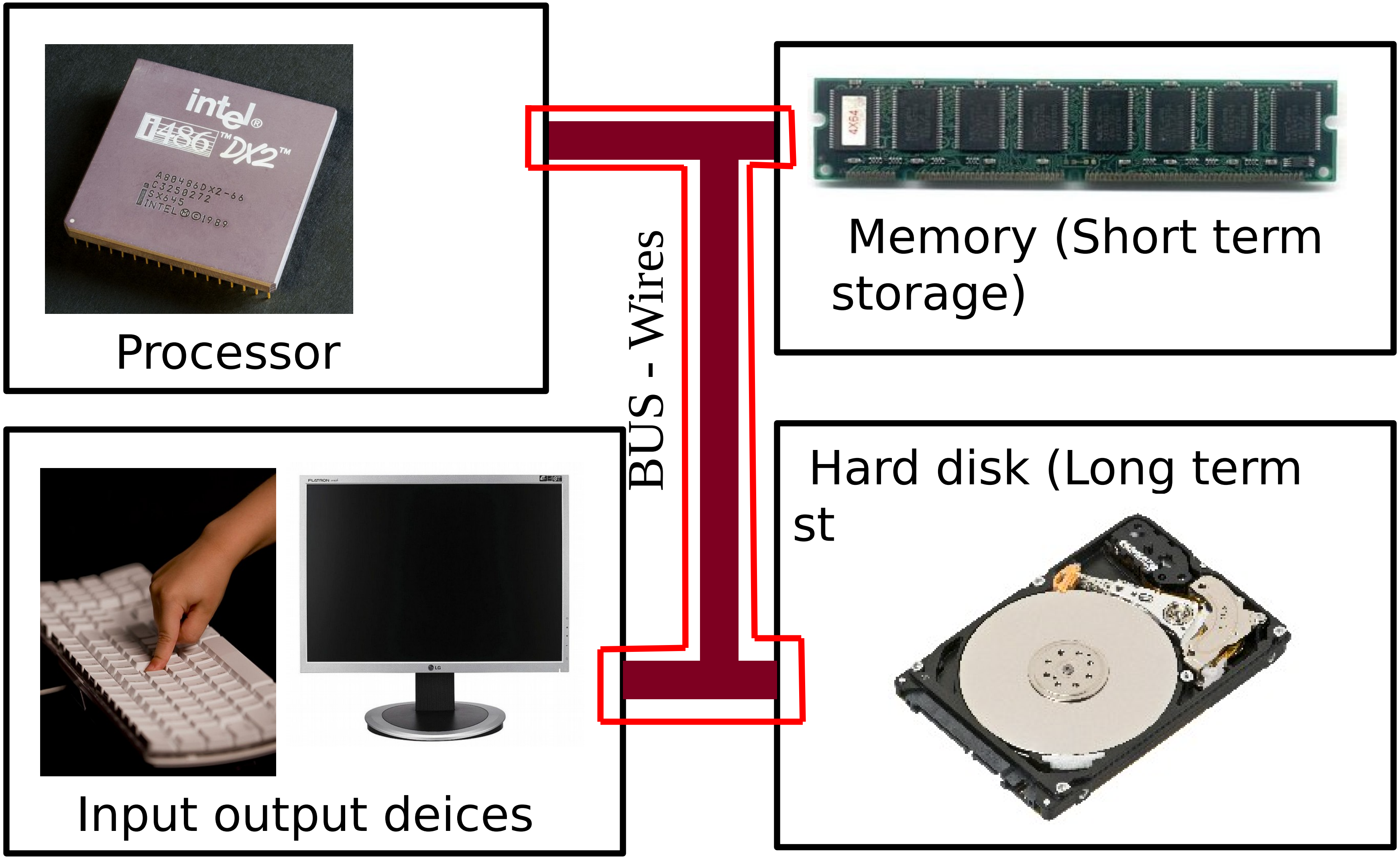
```
> (1+2)*(3+4)/7
```

- The processor **is the chip that works out the answer.**

- Processor speed is measured in **Operations per second**.



The components of a computer:



Processor

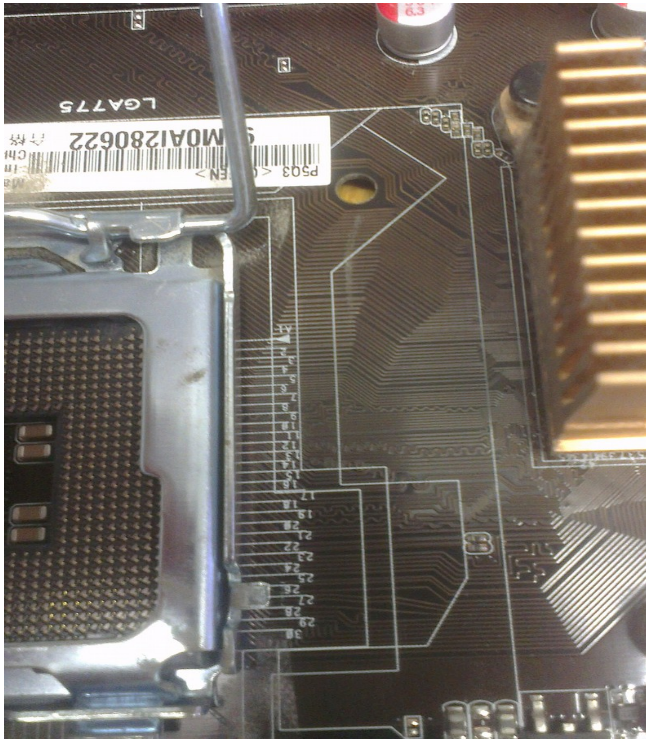
Memory (Short term storage)

BUS - Wires

Input output deices

Hard disk (Long term st

The bus

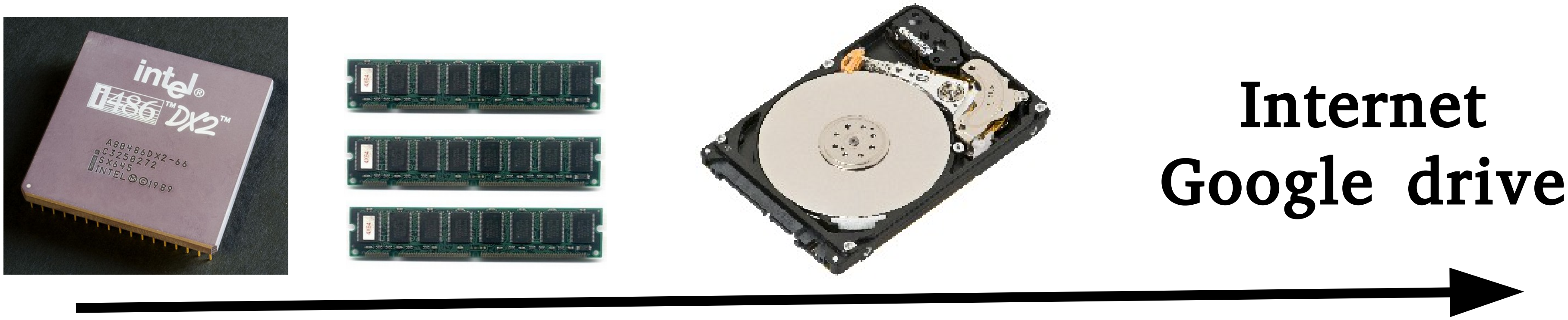


- The bus is a set of wires which connects the **processor**, **memory** and **storage devices together**.

- The bus is used to transfer information between components in the computer.
- It's a bit like an information highway.

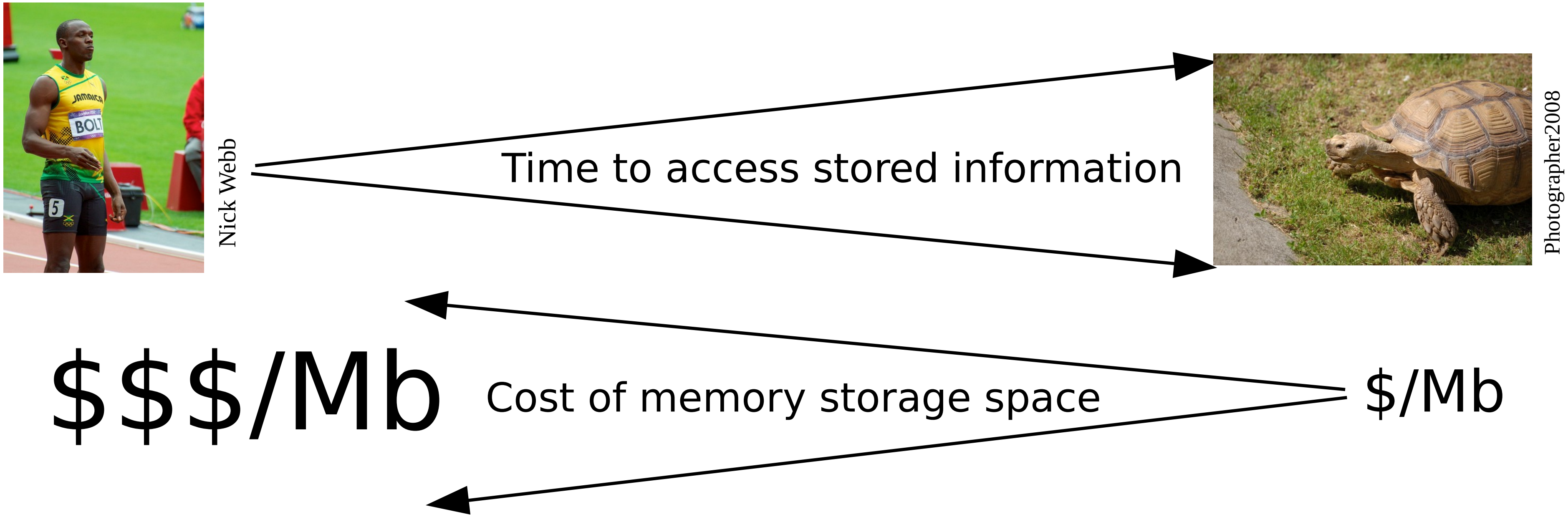
- In the computer circuit board I am handing around you can see it as a brown set of wires – these are the bus.

Access speed v.s. cost



Internet
Google drive

Distance from processor



Nick Webb

Time to access stored information

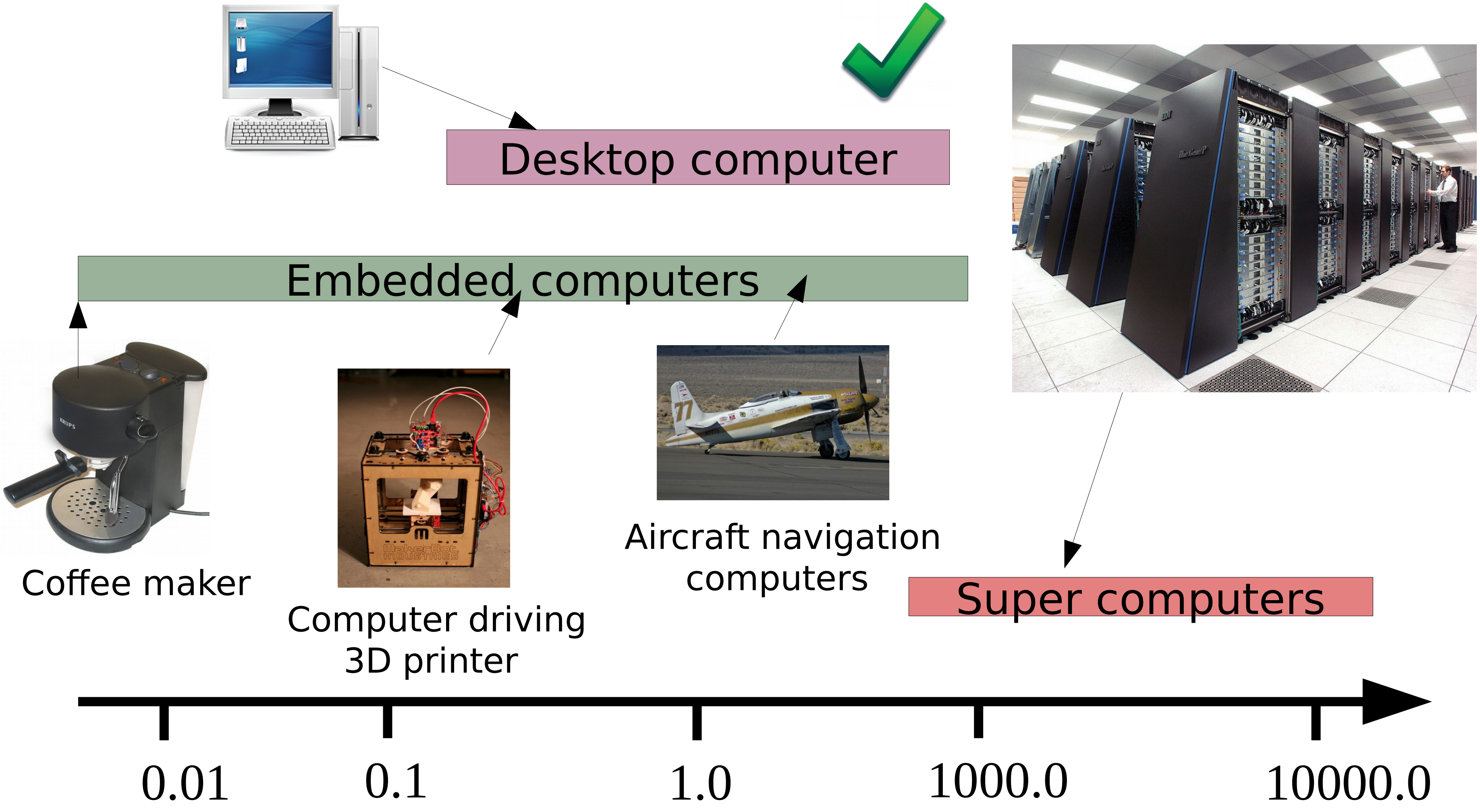
Photographer2008

$$$/Mb

Cost of memory storage space

$/Mb

Types of computer and their computing power



✓

Desktop computer

Embedded computers

Coffee maker

Computer driving 3D printer

Aircraft navigation computers

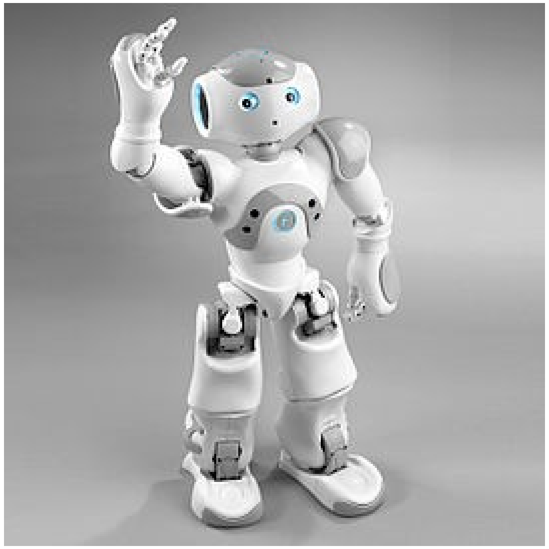
Super computers

Computing power

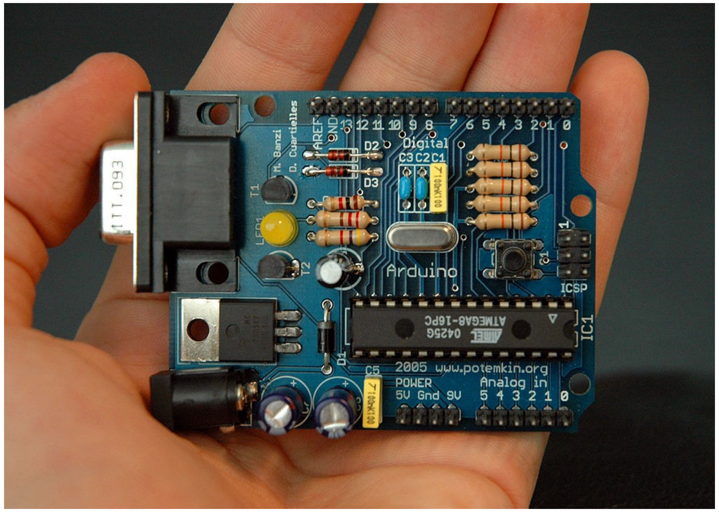
1.0= the power of a standard desktop PC

Embedded computers

- **Embedded computers** are:
  - computers embedded in an object – like computers embedded in a robot.
  - These are the most likely sort of computer you will come across.



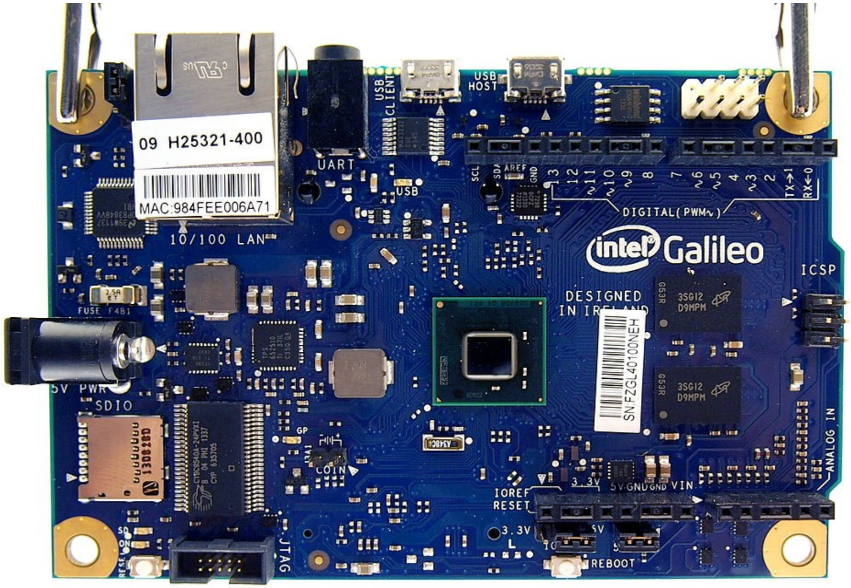
- **Computers on a chip**
  - All components (memory, processor and some storage) are integrated onto a single chip.



Internet of things – more embedded devices

- We will soon be living in a world where everything is online – even your fridge.

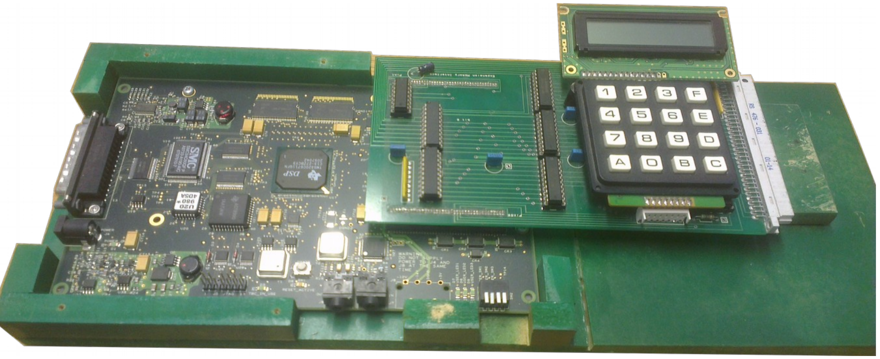
- This is Intel's kit for developing this type of product.



- It has everything that a normal computer would have.

Embedded computers

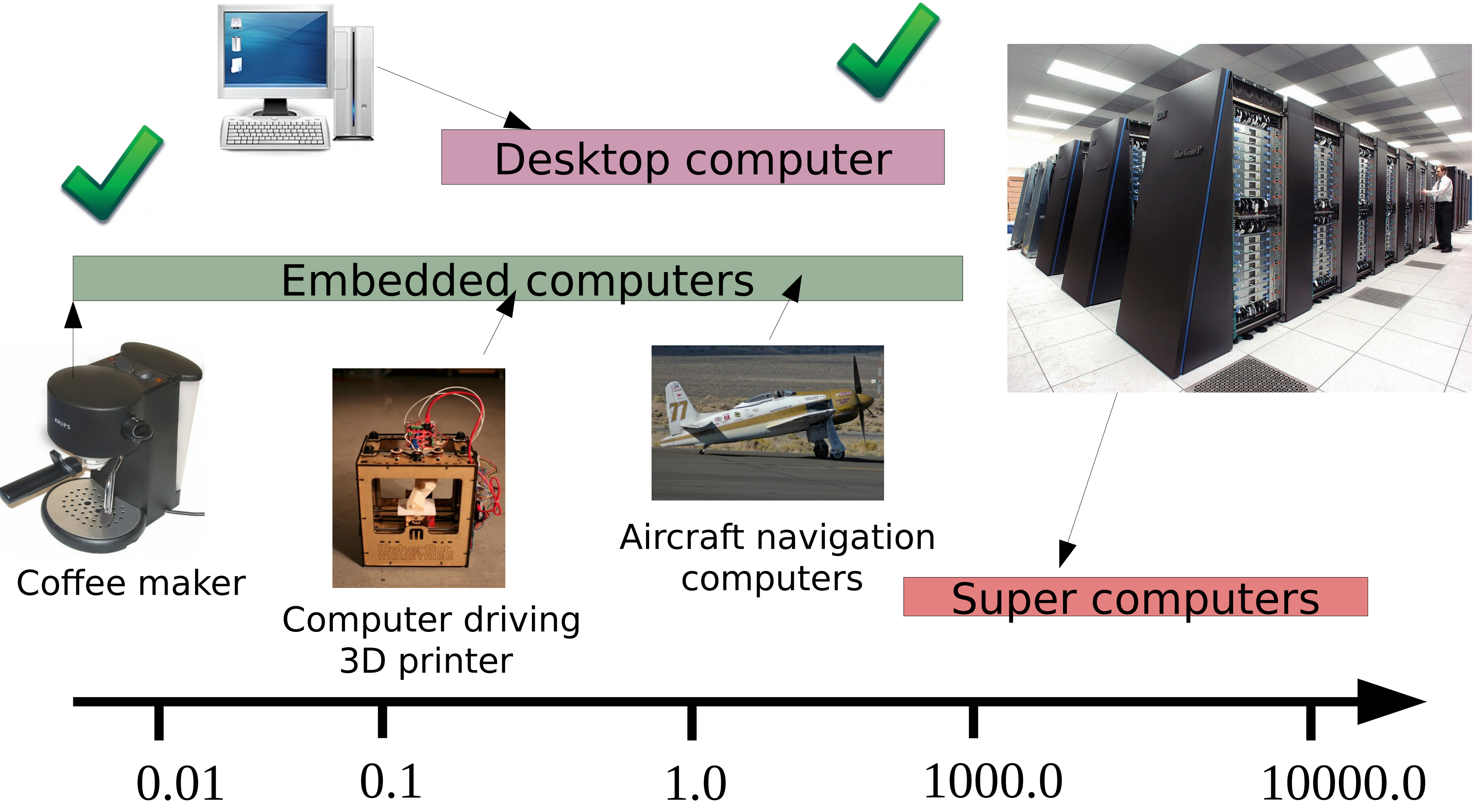
- **Digital Signal Processors (DSP)**
  - This type of computer is specially optimized to process real time data streams



- They widely used to optimize fuel/air mixtures in car engines in response to changing engine conditions.

- They are also used to process audio, and video streams.

Types of computer



Desktop computer

Embedded computers

Coffee maker

Computer driving 3D printer

Aircraft navigation computers

Super computers

Computing power

1.0= the power of a standard desktop PC

Supercomputers

•These computers are very powerful computers typically 1000-100000 more powerful than your desktop computer.

•Engineers use them all the time to solve very complex problems.

•Design of **airplane wings**, **optimizing rocket engines**, in general solving very **difficult problems**.

Supercomputers

- In your professional life there is a good chance you will use a supercomputer.
- Supercomputers could fill a whole lecture. I have therefore organized a special lecture on Supercomputers on **Wednesday 29th October at 2pm** in this room.

- **Dr. Colin Bannister** who runs the university of Nottingham supercomputer facility will be the guest lecturer.
- This is optional and the content will not be in the exam, but it should be interesting (and fun!).

Overview of this lecture

- **Computer fundamentals**
    - What's in a computer?
    - Types of computers
    - **ASCII code**

- Writing to the screen

- Reading text from the keyboard

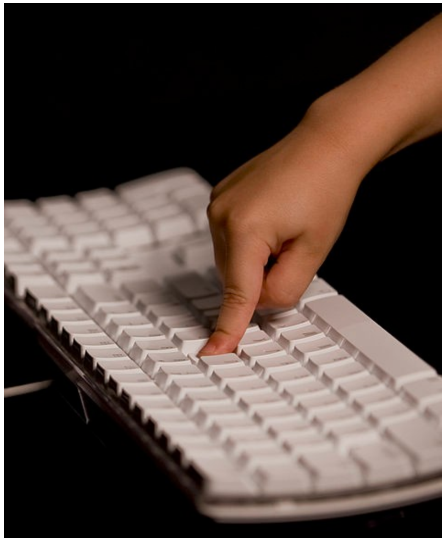
- Strings in depth

How do computers store information?

- Did you know that computers store and transmit ***all*** text as numbers from 0 to 255?

- For example:

a	A	b
97	65	98



- This code is called ASCII code (American Standard Code for Information Interchange)

Here is the full character list (ASCII code)

Number	Char	Number	Char	Number	Char	Number	Char
0	[NULL]	32	[SPACE]	64	@	96	`
1	[START OF HEADING]	33	!	65	A	97	a
2	[START OF TEXT]	34	"	66	B	98	b
3	[END OF TEXT]	35	#	67	C	99	c
4	[END OF TRANSMISSION]	36	$	68	D	100	d
5	[ENQUIRY]	37	%	69	E	101	e
6	[ACKNOWLEDGE]	38	&	70	F	102	f
7	[BELL]	39	'	71	G	103	g
8	[BACKSPACE]	40	(	72	H	104	h
9	[HORIZONTAL TAB]	41	)	73	I	105	i
10	[LINE FEED]	42	*	74	J	106	j
11	[VERTICAL TAB]	43	+	75	K	107	k
12	[FORM FEED]	44	,	76	L	108	l
13	[CARRIAGE RETURN]	45	-	77	M	109	m
14	[SHIFT OUT]	46	.	78	N	110	n
15	[SHIFT IN]	47	/	79	O	111	o
16	[DATA LINK ESCAPE]	48	0	80	P	112	p
17	[DEVICE CONTROL 1]	49	1	81	Q	113	q
18	[DEVICE CONTROL 2]	50	2	82	R	114	r
19	[DEVICE CONTROL 3]	51	3	83	S	115	s
20	[DEVICE CONTROL 4]	52	4	84	T	116	t
21	[NEGATIVE ACKNOWLEDGE]	53	5	85	U	117	u
22	[SYNCHRONOUS IDLE]	54	6	86	V	118	v
23	[ENG OF TRANS. BLOCK]	55	7	87	W	119	w
24	[CANCEL]	56	8	88	X	120	x
25	[END OF MEDIUM]	57	9	89	Y	121	y
26	[SUBSTITUTE]	58	:	90	Z	122	z
27	[ESCAPE]	59	;	91	[	123	{
28	[FILE SEPARATOR]	60	<	92	\	124	|
29	[GROUP SEPARATOR]	61	=	93	]	125	}
30	[RECORD SEPARATOR]	62	>	94	^	126	~
31	[UNIT SEPARATOR]	63	?	95	_	127	[DEL]

ASCII numbers to text example

This is how the computer stores my name:

Dr. MacKenzie

Name = [68 114 46 32 77 97 99 75 101 110 122 105 101]

But why should I care about this?

- **All computers store/transmit/read all information using this code.**

  - When you later (in mechatronics) try to make your computer talk to a **3D printer**, **data capture card** or **robot** it will **expect** commands composed of **ASCII** numbers from you.

  - For example if you send this robot the command ***PowerOn*** you would actually send [80 111 119 101 114 79 110] in ASCII code.

Converting from numbers to characters using char

In MATLAB if we wanted to tell the computer to convert this list of numbers back to a human readable string we would type:

```
>name = [68 114 46 32 77 97 99 75 101 110 122 105 101]

>char(name)

    Dr. MacKenzie
```

Your go!

- This set of numbers is sent to a robot:

  [80 79 87 69 82 79 78 0 82 111 116 97 116 101 57 48 100 101 103]

- Using the ASCII table on the previous slide convert this command into human readable text.

- What does this command tell the robot to do?

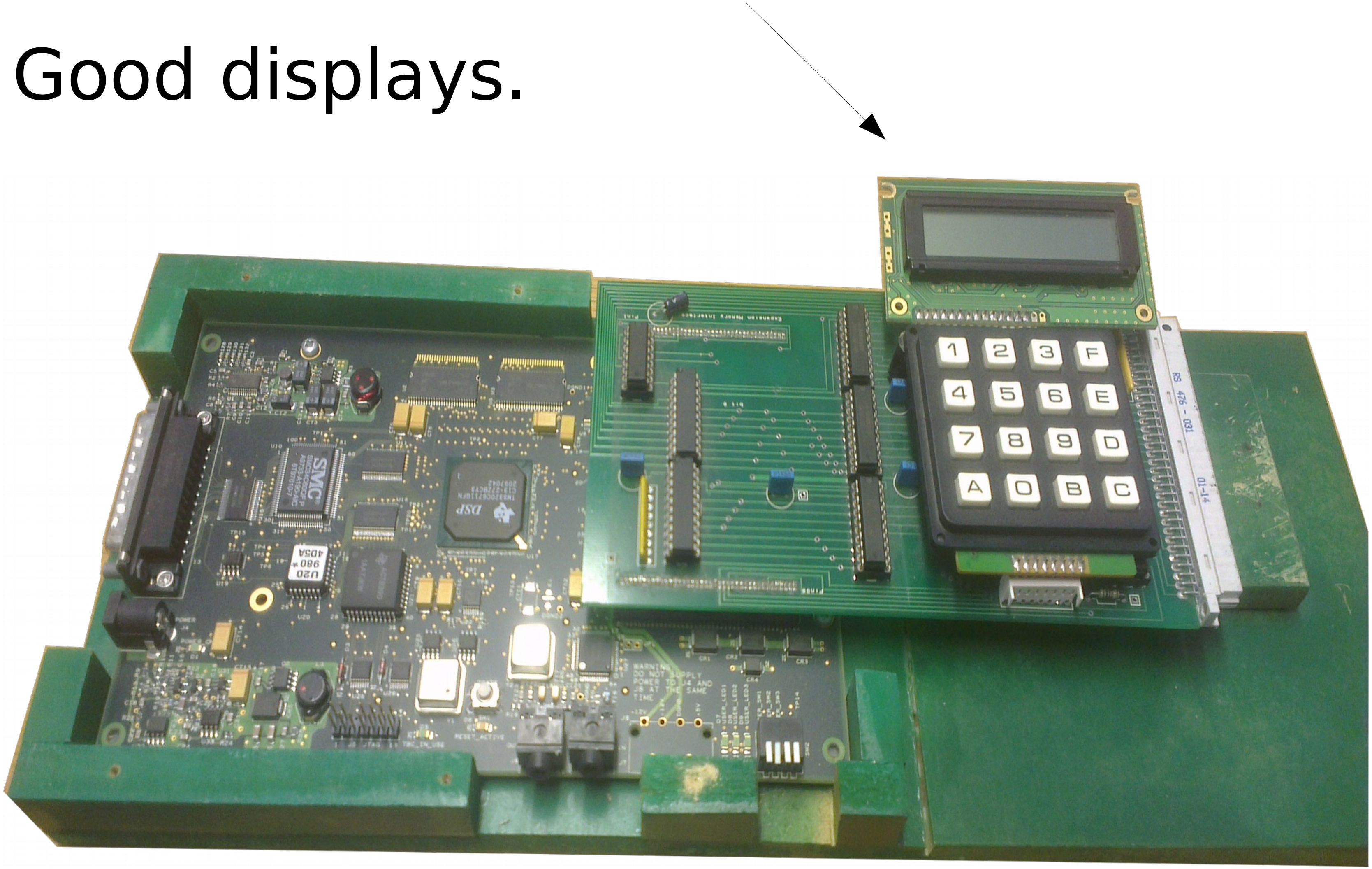
- Hint: The 0 is used by the robot to separate commands.

Question?

What did all the computers in today's lecture not have?

Answer:

Good displays.



Controlling output to the screen

•Most the computers you will work with be embedded in products like cars or jet engines.

•It is very important to be able to display text on these screens (graphs and pretty graphics are not an option!)

Allo002

•We therefore need to know how to control text output accurately..... back to MATLAB

Overview of this lecture

- Computer fundamentals
    - What's in a computer?
    - Types of computers
    - ASCII code
- **Writing to the screen**
- Reading text from the keyboard
- Strings in depth

Controlling output to the screen in MATLAB

- So far, our only option for controlling output to the screen is has been putting a '**;**' at the end of the line.

- This stopped MATLAB printing to the screen:

```
>x=1+2
     x=3

>x=1+2;
>
```

- So we need a better method to control output if we want to control screens like this:

Displaying text using the ***disp*** command

- Two simple examples of the **disp** command:

```
>disp('Hello world!')

 Hello world!
```

Notice the single quotes.

or

```
>disp('Learning how to program a computer will make me rich!')

Learning how to program a computer will make me rich!
```

You can remember this command
by thinking of a computer *disp*lay

Displaying numbers with the ***disp*** command

- ***disp*** command can display **numbers** OR **sentences**
  - But not both at the same time

```
%Program to print the speed of light

disp('The speed of light is ');
disp(3e8);
disp('m/s');
```

```
The speed of light is
3e8
m/s
```

Variables that can hold text

- Before we can get more control over the output we have to learn about '***strings***'

- **Strings** are a special type of variable that can hold text. Examples are:

```
message='The speed of light is '
name='Rod'
day_of_week='Monday'
name_of_university='Nottingham'
```

Single quote

disp can print variables .....

- The **disp** command also works with variables:

```
%Program to print the speed of light
message='The speed of light is ';
speed_of_light=3e8;
units='m/s';

disp(message);
disp(speed_of_light);
disp(units);
```

```
The speed of light is
3e8
m/s
```

- But notice we still don't have much control over how our text is printed.

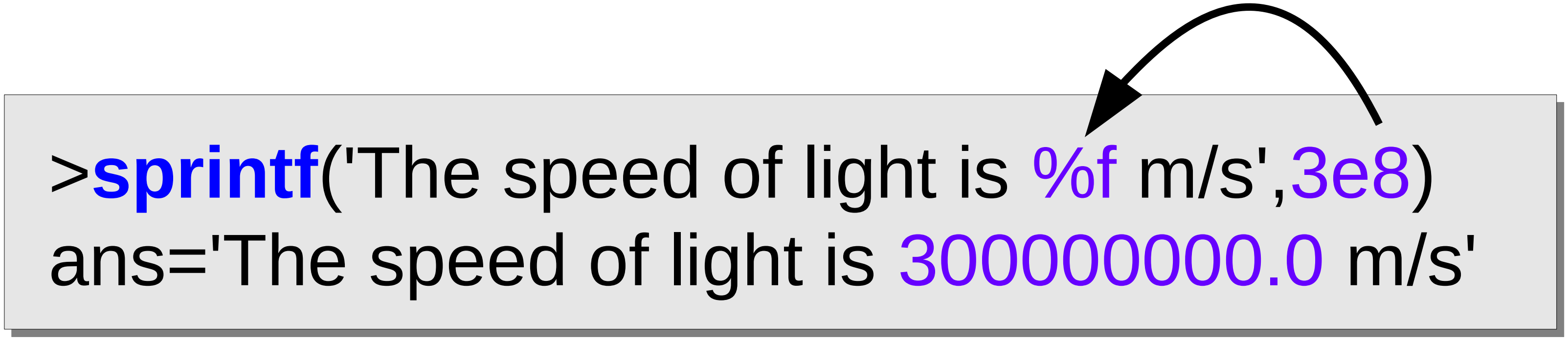
- **What is wrong with this output?**

- For this we need another command.

More control over ***disp*** with ***sprintf***

- Imagine I wanted to print

  “The speed of light is 300000000.0 m/s”



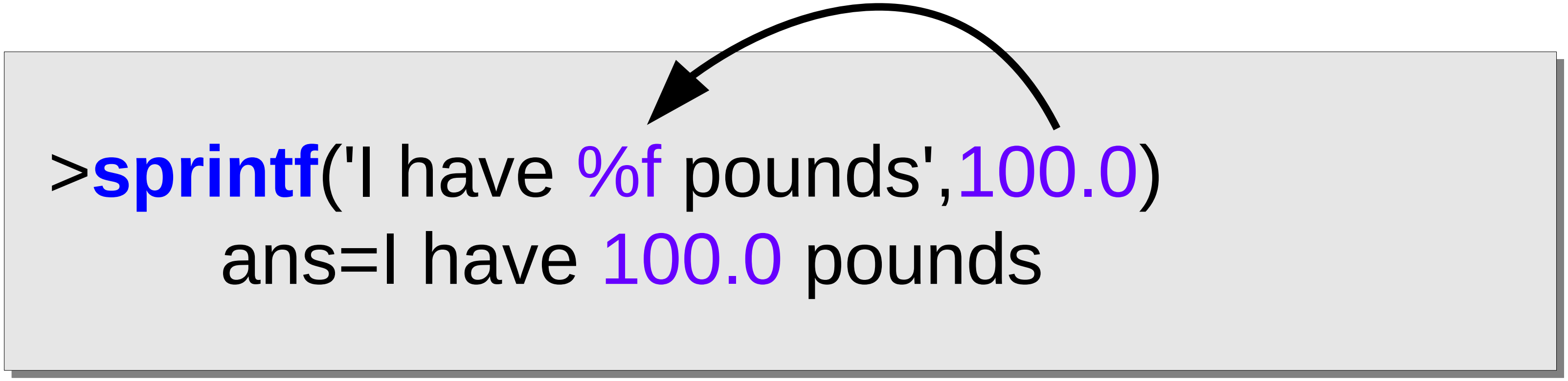
```
>sprintf('The speed of light is %f m/s',3e8)
ans='The speed of light is 300000000.0 m/s'
```

- **%f** is called a format specifier.

sprintf another example

- Imagine I wanted to print

  “I have 100.0 pounds”

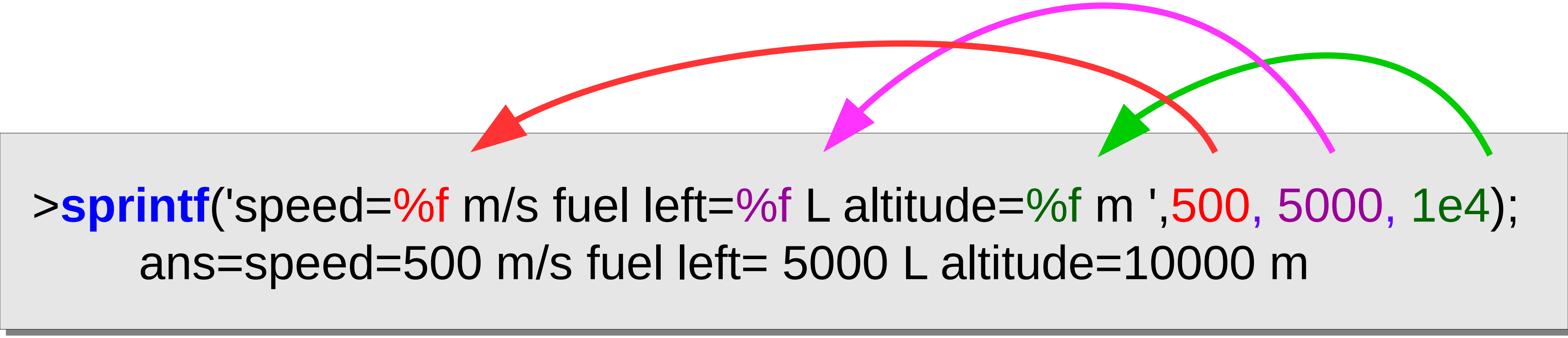


```
>sprintf('I have %f pounds',100.0)
        ans=I have 100.0 pounds
```

sprintf in depth

- Another example imagine we wanted to print:

  speed=500 m/s fuel left= 5000 L altitude=10000 m



```
>sprintf('speed=%f m/s fuel left=%f L altitude=%f m ',500, 5000, 1e4);
       ans=speed=500 m/s fuel left= 5000 L altitude=10000 m
```

- **sprintf** replaces anything beginning with a '%' with the corresponding number.

More control over output of strings with ***sprintf***

- You can specify the number of decimal places a number should be printed to

```
a=sprintf('The value of pi is %.10f',pi);
disp(a);
>>The value of pi is 3.1415926536
```

%.10f

- Number of decimal places

```
a=sprintf('The value of pi is %.5f  %.10f',pi,pi);
disp(a);
>> The value of pi is 3.14159  3.1415926536
```

sprintf is not limited to decimal numbers

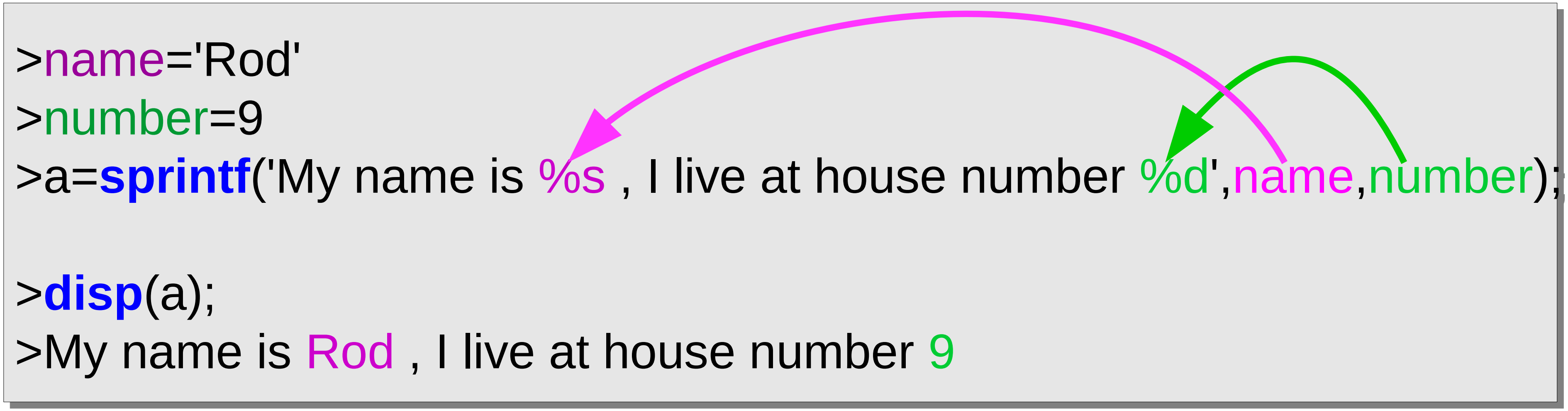
Type	Significance	Example
%f	Floating point (decimal place)	1.11111
%e	Scientific notation	1e-1
%d	decimal	100
%s	string	'hello'
%c	Single character	A

✓

Mixing numbers and text with ***sprintf***

Imagine we wanted to print

- **'My name is Rod , I live at house number 9'** where Rod and 9 are stored in variables.
- We could do it like this.



```
>name='Rod'
>number=9
>a=sprintf('My name is %s , I live at house number %d',name,number);

>disp(a);
>My name is Rod , I live at house number 9
```

Special characters ***sprintf*** accepts

- ***sprintf*** also has some special sequences of characters which can be used to further format the string:

Note this is x2 single quotation marks

Character	Significance
\n	New line
\t	tab
\\	backslash
\''	Single '
\%	Percent

- This is because ***sprintf*** understands % and ' as having a special meaning.

Special characters ***sprintf*** accepts

Oh this shiny new computer -
There just isn't nothin' cuter.
It knows everything the world ever knew.
And with this great computer I don't need no writin' tutor,
'Cause there ain't a single thing that it can't do.

by Shel Silverstein

```
 %A computer poem
 a=sprintf('Oh this shiny new computer - \n There just isn\"t
nothin\" cuter.\n It knows everything the world ever knew. \n And
with this great computer I don\"t need no writin\" tutor, \n \"Cause
there ain\"t a single thing that it can\"t do.\n ');
 disp(a);
```

Overview of this lecture

- Computer fundamentals
  - What's in a computer?
  - Types of computers
  - ASCII code

- Writing to the screen

- **Reading text from the keyboard**

- Strings in depth

Input from the keyboard

- No matter which computer you spend your working day programming it will almost always have a keyboard.

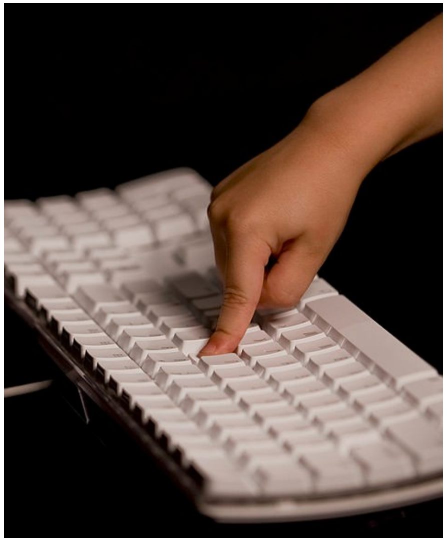
- The next part of the lecture deals with getting text from the keyboard into MATLAB variables.

PresLoiLoi

Flight management system from Boeing 737.

Neitram

ATM



PC - keyboard

Keyboard ***input***

Often your program needs to ask the user a question which requires a numeric answer:

How much fuel is needed?

In MATLAB we would do this with the input command

```
answer=input ('How much fuel is needed?');
```

A simple example

```
% Program to evaluate a quadratic
x=input('What value of x do you want to solve the equation for?')
y=(2*x*x+3*x+1)*cos(x)*sin(x);
disp('The answer is:')
disp(y)
```

```
What value of x do you want to solve the equation for?
```

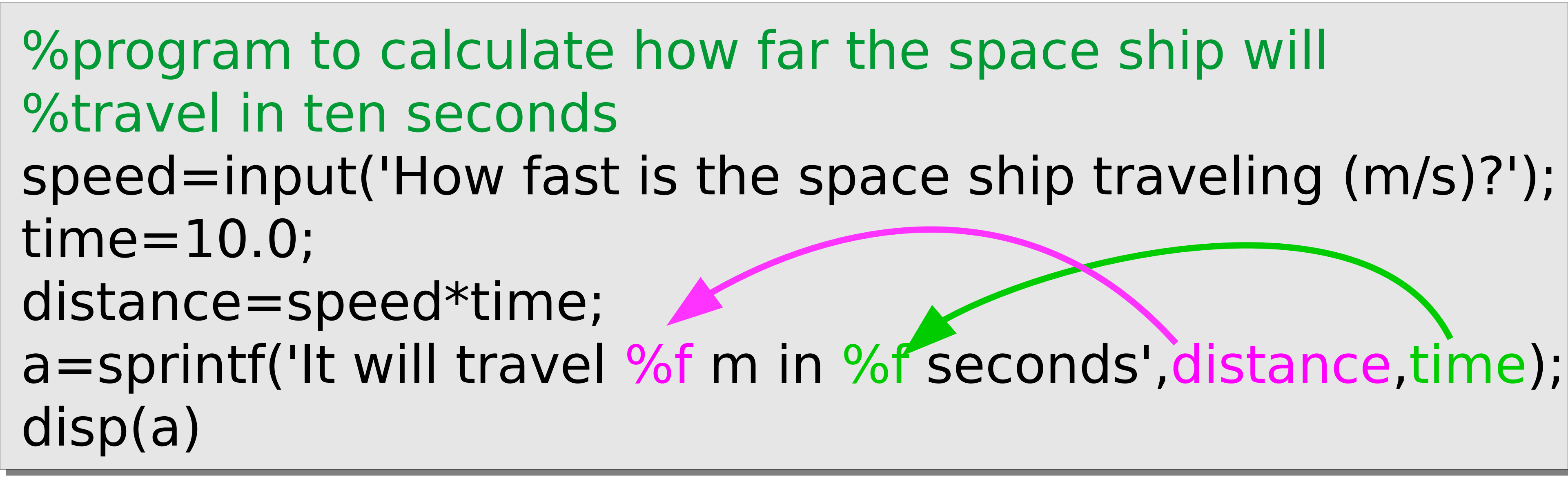
A simple example

```
% Program to evaluate a quadratic
x=input('What value of x do you want to solve the equation for?')
y=(2*x*x+3*x+1)*cos(x)*sin(x);
disp('The answer is:')
disp(y)
```

```
What value of x do you want to solve the equation for? 1.0
The answer is:
2.7279
```

Using ***input*** and ***sprintf*** together

Calculating how far the space ship will travel in ten seconds:



```
%program to calculate how far the space ship will
%travel in ten seconds
speed=input('How fast is the space ship traveling (m/s)?');
time=10.0;
distance=speed*time;
a=sprintf('It will travel %f m in %f seconds',distance,time);
disp(a)
```

```
How fast is the space ship traveling (m/s)?
```

Using ***input*** and ***sprintf*** together

Calculating how far the space ship will travel in ten seconds:

```
%program to calculate how far the space ship will
%travel in ten seconds
speed=input('How fast is the space ship traveling (m/s)?');
time=10.0;
distance=speed*time;
a=sprintf('It will travel %f m in %f seconds',distance,time);
disp(a)
```

```
How fast is the space ship traveling (m/s)? 1000.0
It will travel 10000.0 m in 10.0 seconds.
```

Keyboard *input*

It is also common to need to get text from the keyboard in response to a question:

```
Launch the rocket [yes/no]?
```

You need to append an 's' (for string) at the end of the input command....

```
answer=input ('Launch the rocket [yes/no]?','s');
```

Overview of this lecture

- Computer fundamentals
    - What's in a computer?
    - Types of computers
    - ASCII code

- Writing to the screen

- Reading text from the keyboard

- **Strings in depth**

Strings in depth

- In the first part of the lecture we learnt that strings of text can be stored in a variable.

```
% A string example
a='My name is Rod';
disp(a);
      My name is Rod
```

- But strings are ***really just arrays*** of letters and we can use all the tricks we learnt to deal with strings in the first three lectures to play with strings....

For example

- If we defined the string

```
a='My name is Rod'
```

- We could find out what the 2nd character is by doing

```
>a(2)
     ans='y'
```

- Or we could swap out the 14th character for a b

```
>a(14)='b';
>disp(a)
     My name is Rob
```

- Sometimes it's handy to think of strings as arrays.

Overview of this lecture

- Computer fundamentals
  - What's in a computer?
  - Types of computers
  - ASCII code
- Writing to the screen
- Reading text from the keyboard
- Strings in depth