FUNDAMENTALS OF MATLAB

Or:

How Matlab made my life better.

Programming

- Different languages …
  … one basic skill.

- Key: Being as simple-minded as…
  … your computer.

Part I

- What is a computer?
- What is an algorhythm?
- What is a vector?
- Matrices and basic operations

Computers are

- Tame and stubborn.

- No matter how hard you try, you won’t BREAK it with your code.

Computers are

- Hardware + Software
- Hardware:
  - What you'll ruin with a brick, a hammer, a screwdriver (of both solid and liquid kind), or even a third-floor window …

Computers are

- Hardware + Software
- Hardware:
  - Memory:
    - dead (ROM, hard drive) or alive (RAM)
    - in a stick (flash) or in a disk (CD, DVD)
  - Monitor and graphics card (VRAM)
  - Speakers (and sound card, integrated)
  - Various peripherals:
    - Mouse, keyboard, eye tracker, EEG stuff.
  - ONE (or a few)
    ALL-MIGHTY microprocessor CPU
    - Central Processing Unit

(D)RAM and CACHE

- Dynamic Random Access Memory
  - Yummy.
  - Link.
- Get a TON!
- This will make you (and Matlab) happier.

So... software

- a.k.a. programs
  - Catalog of “behavioral” instructions.
- Programs are written in “programming languages”.
- CANNOT HARM your computer.
- So, feel free to crash your computer as much as you’d like!

Aha! Programming

- Tell the machine what to do.

- You'll need:
  - A language to communicate with the CPU's binary world (bits & bytes) and
  - Lots of patience.

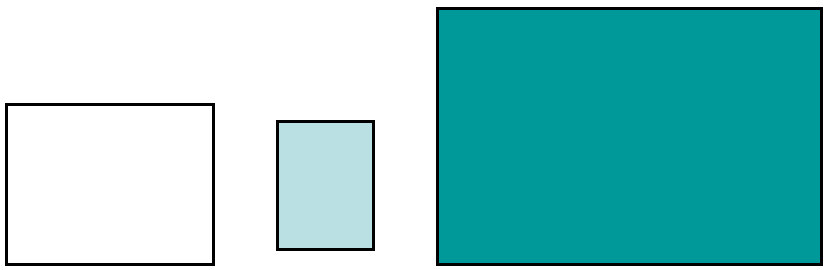
MATLAB

- MATrix LABoratory:
  - High-level language
    - slow to interpret by machine
    - easy to understand by humans
  - I will teach you “syntax”
    - Algorithm based
  - Our vocabulary will be MATRICES (and functions).

    MATLAB is optimized for **matrix-based** calculations

ALGORITHMS

- Describe in a piece of paper, the instructions required to achieve the following goal:
  - I want to stack these three boxes on top of each other.

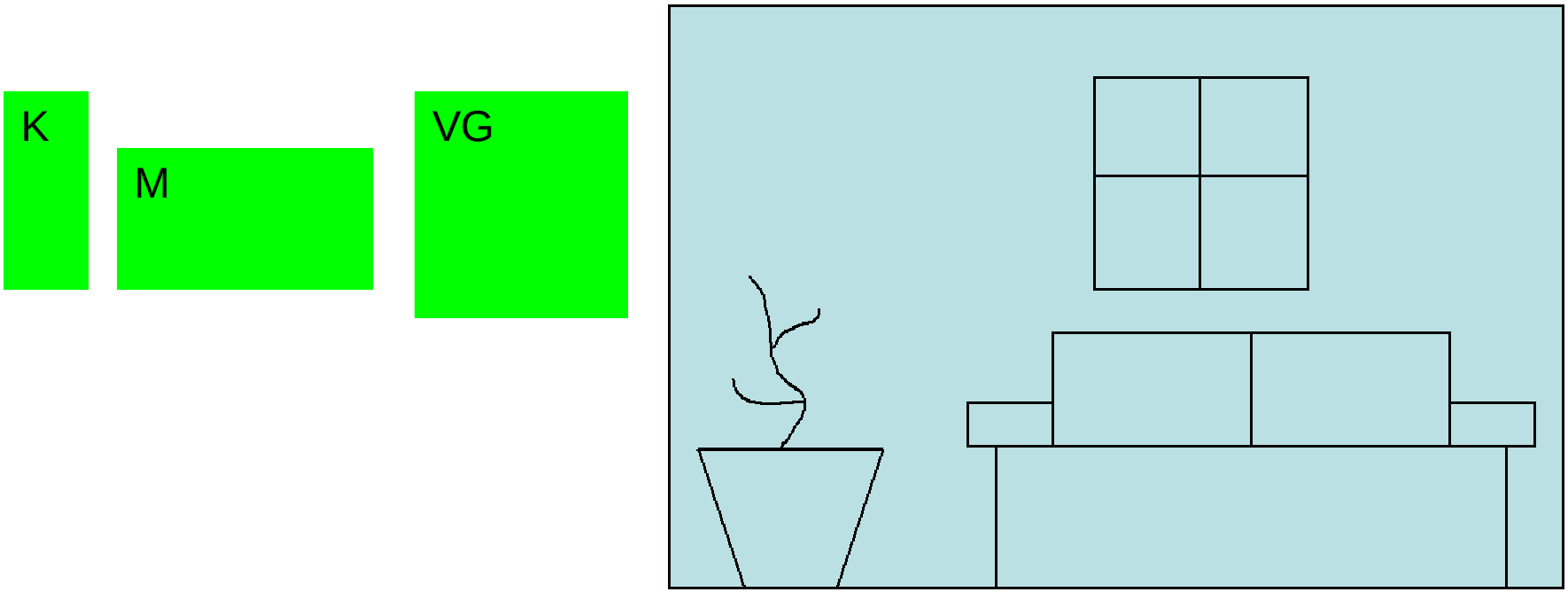


How would these instructions change if we wanted a stable structure?

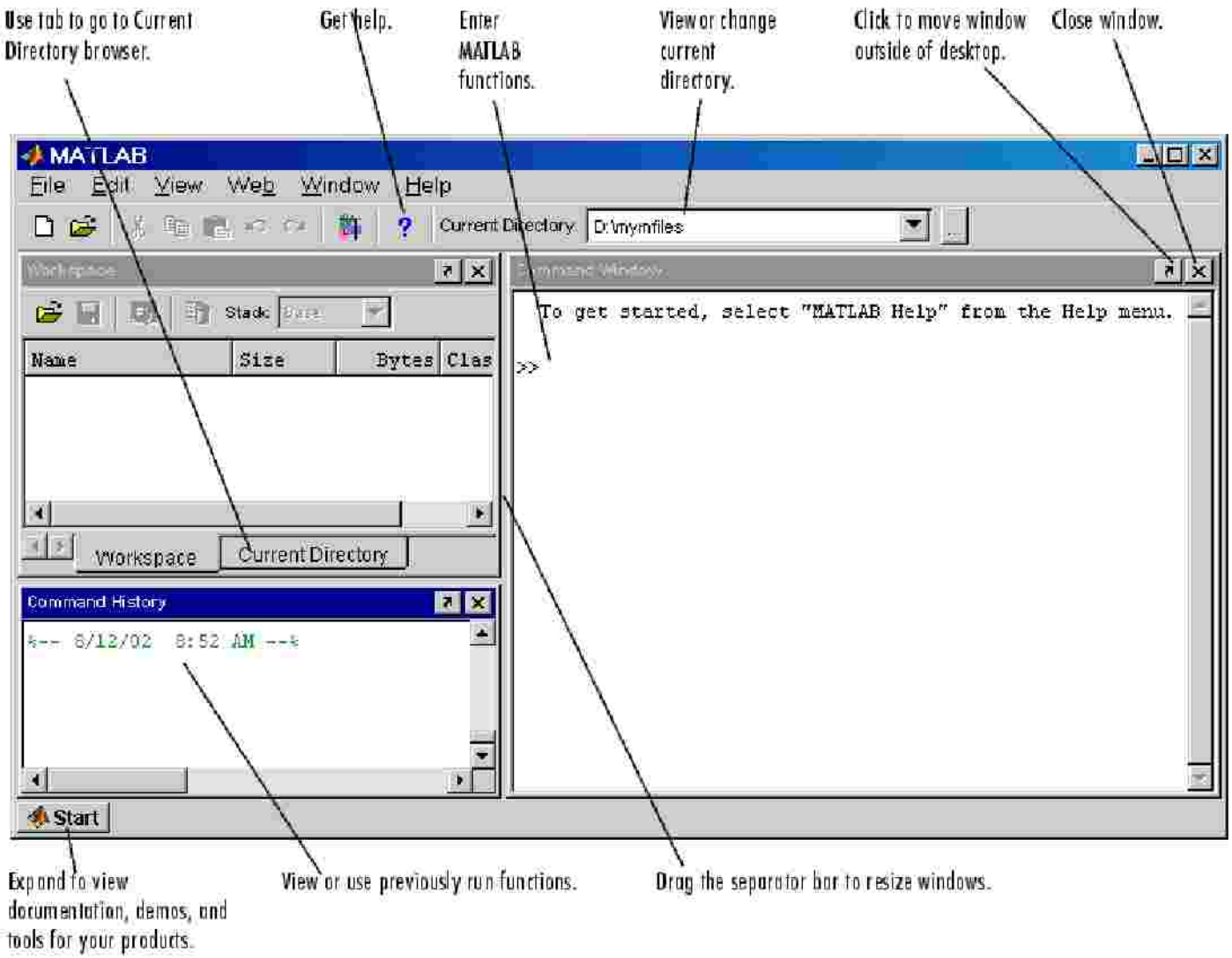
How would these instructions change if we had an indeterminate amount of boxes of different sizes?

ALGORITHMS

- Describe in a piece of paper, the instructions required to achieve the following goal:
  - I want to hang my Klimt, Monet, and VanGogh paintings on my wall.



START MATLAB!



A few important things

- **Create a directory where you will save your Matlab work**

- **SET PATH!**
  - **Window or in command line:**
  - **Setting the path in your code:**

```
>p=path;
>path(p,'c:\MatlabCourse');
```

A few important commands

- **pwd: Where are you?**
- **cd: Change Directory**
- **dir: lists contents of folder**
- **mkdir: makes a new directory**

- **who: lists of variables in workspace**
- **whos: who + size of variables**
- **clear X: erases X**
- **clear all: erases all.**

The most important commands

- **help *topic***
- **lookfor *string***
- **edit *nameoffile***

Intro to Matrices

- **Vector**:
  - Orderly array of stuff
  - Stuff = numbers, letters, words, bits

[1 2 3 4 5 6] is a 6 element vector of digits

[‘cat’ ‘dog’] is a 2 element vector of words (strings)

[c a t d o g] is a 6 element vector of characters

[0 1 0 1 1 1] is a 6 element vector of bits (binary digit)

MATRIX = “VECTOR” Of Vectors

Matrices

- Simple matrices will look like this: (horizontally stacked vectors)

```
[1 1 1 1
 2 2 2 2
 3 4 5 6]
```

What is the (2,3) element in the matrix?

This is a 3 rows by 4 column matrix referred to as a 3-by-4 matrix.

First mentioned dimension is **ALWAYS the number of rows**
Second dimension = number of columns

Matrices

- Matrices can have as many **dimensions** as you'd like.
  - Vectors have 1 dimension
    - Third element in vector V: V(3)
  - Most algebraic Matrices have 2 (one for the rows, and for the columns)
    - Third row, fourth column of M: M(3,4).
  - Matrices can be (m-by-n-by-o-by-p)
    - Elements are then referenced (i,j,k,l), One index for each dimension.

What can I put in a matrix?

- A note on memory allocation and types of values: link
  - numbers (int, double)
  - or letters (char)
  - or strings of letters (strings)
  - A matrix with a combination of different kinds of values is called a STRUCTURE (later in the semester).

- A **variable** is a piece of memory that holds whichever value you specify. In Matlab, a variable will hold the values of an entire matrix or structure. Size.

- A **pointer** is the address in memory of a piece of memory holding whichever value you specified.

Quick word on variables

A variable NAME has at least one letter + any number of letters, digits and underscores.

```
myage = [29]

myname  = ‘alejo’
```

What happens when I do?

```
myname = alejo
```

A WARNING on variables

Matlab is smart

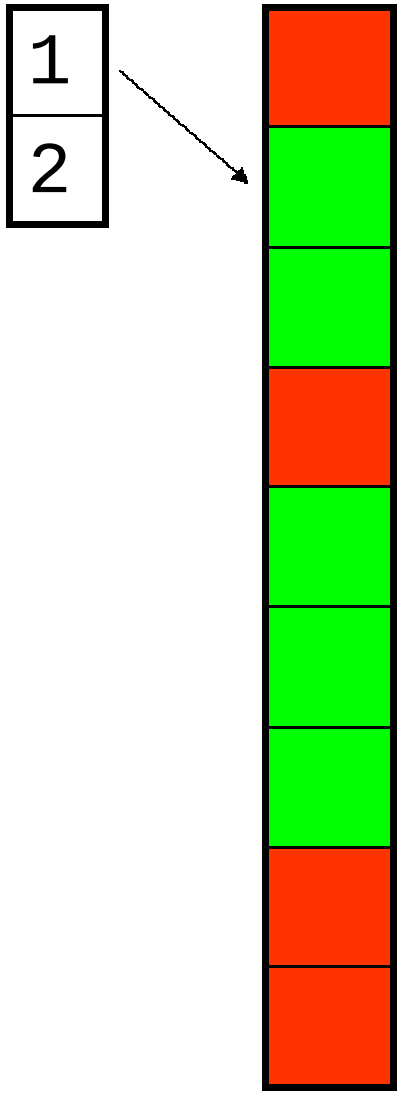
--> no preallocation necessary

Dangerous!

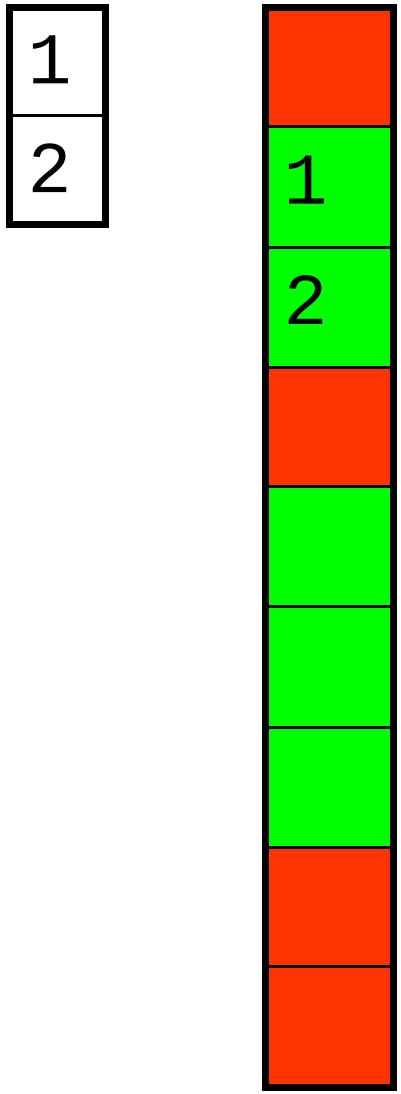
Besides, it is slow:

The #1 slow-down factor of Matlab code is dynamic preallocation.

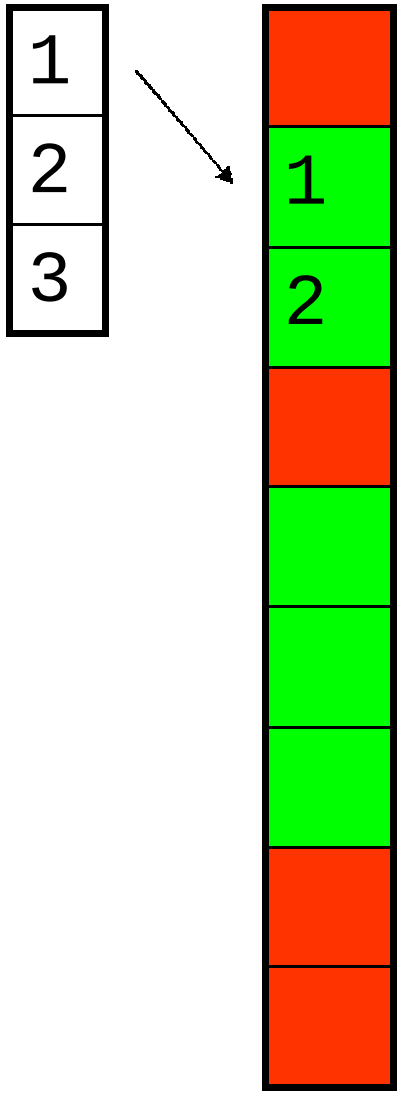
How does Matlab put information in memory?



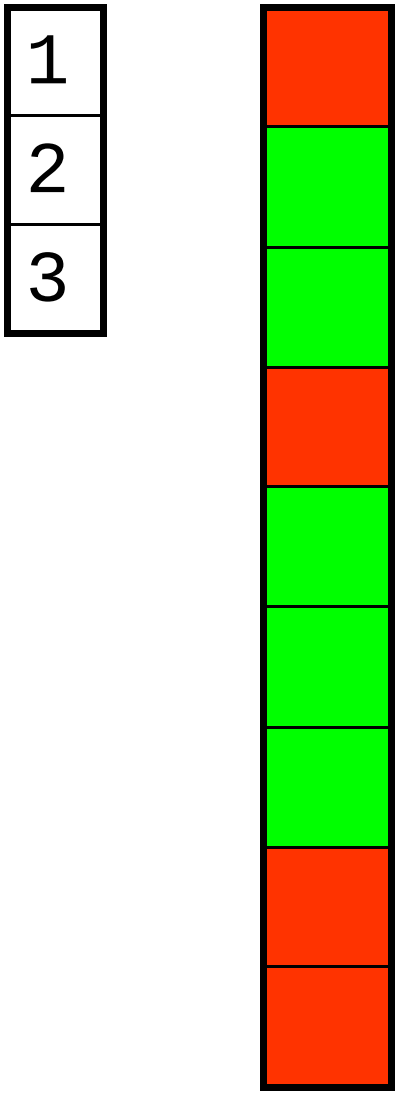
How does Matlab put information in memory?



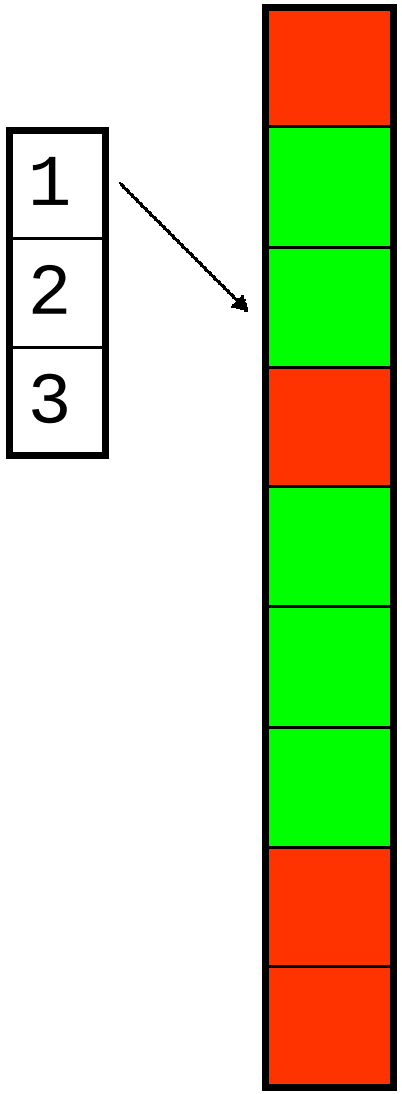
How does Matlab put information in memory?



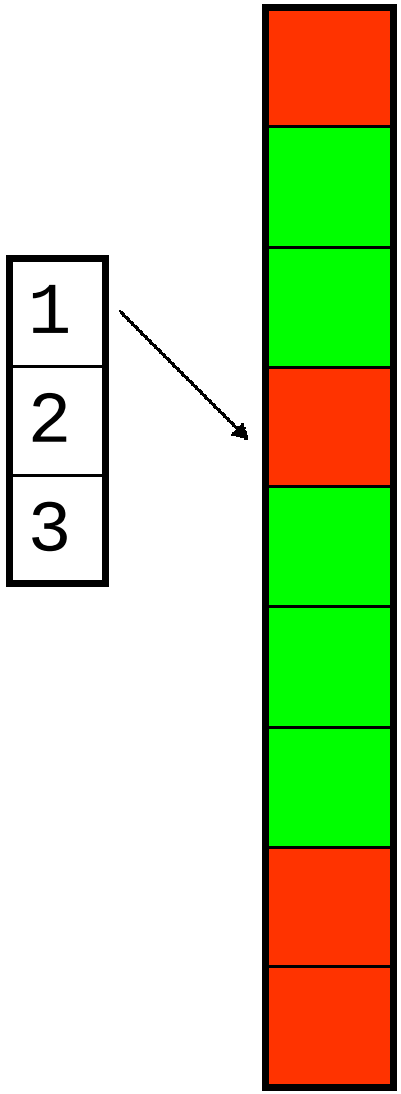
How does Matlab put information in memory?



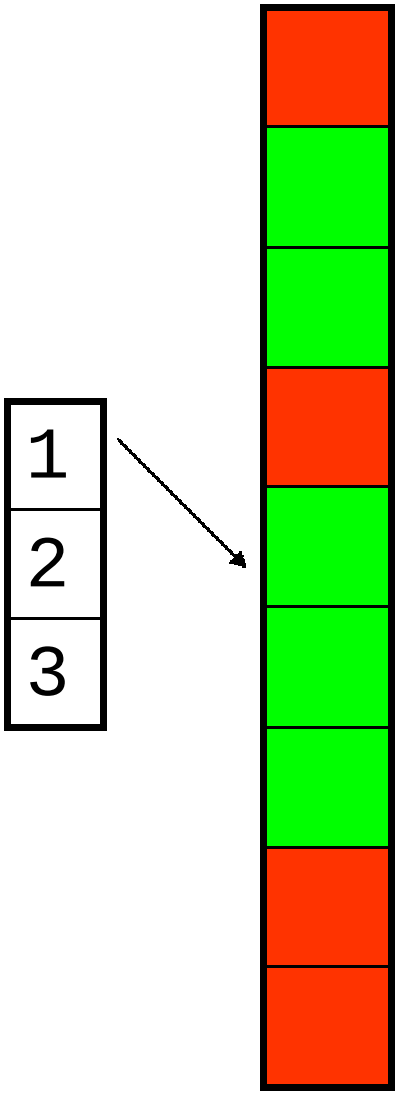
How does Matlab put information in memory?



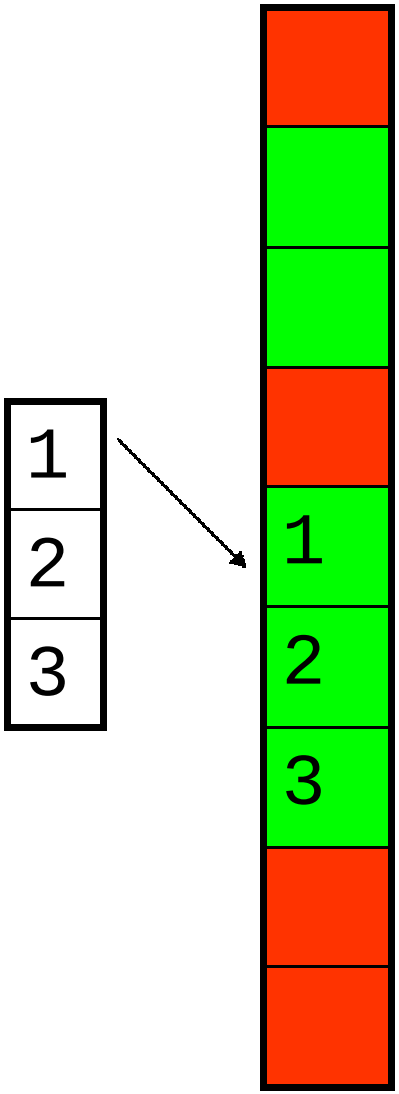
How does Matlab put information in memory?



How does Matlab put information in memory?



How does Matlab put information in memory?



And this repeats EVERY time you modify the SIZE of a variable.

THE POWER OF MATLAB RESIDES IN THE SEQUENTIAL ADDRESS OF ORDERED INFORMATION IN MEMORY.

Back to Matlab!

- Name your matrices always lowercase (because…)
- Create variable a:

```
> a = [ 1 2 3 4];
```

, delimitates elements within a row

Create matrix b:

```
> b = [ 1 2 3
4 5 6];
```

; delimitates rows

Or:

```
> c = [1,2,3;4,5,6];
```

c(i,j) is element i,j of c

Back to Matlab!

- Create a vector of characters:

```
> d = ‘Hello world’;
```

What is d(4)? d(5)? d(6)?

Create a matrix of characters:

```
> e = [‘Hello’;’world’]
```

Select some of the elements in the Matrix using ‘:’

```
> f = e(:,5)   %all elements of 5th column of e
> g = e(2,2:4) %elements 2, 3 and 4 of 2nd row.
```

Exploring…

- Question: what do empty brackets do in the expression?

```
> e(:,4)=[]
```

Exploring…

Question: without typing on Matlab, what happens in the next series of commands?

```
> m = e(2,:);
    % m is a vector = 2nd row of e
Øe(2,:) = e(1,:);
    % 2nd row of e = 1st row of e
> e(1,:) = m;
```

What we learned…

--> to swap the values of 2 variables…

…you need 3 variables.

Exploring…

- Question: can I simply use numbers, like in? (that is, without brackets)

```
> age = 29;

Type
> age
> age(1)
> age(2)
```

Anything bizarre?

Exploring…

- Question: what do you think would happen if I did the following?

```
> t = e(7);
```

Values are stacked!

e(1) = e(1,1)

e(2) = e(2,1)

e(7) = ‘o’

Stacking matrices...

e(k) = `e(1+mod(k-1,number of rows),ceil(k./y))`

mod(x,y): remainder of the division of x by y:

mod(10,3) = 1

mod(9,3) = 0

SUPER USEFUL COMMAND (counterbalancing conditions)

Huhhh…?

e(k) = `e(1+mod(k-1,number of rows),ceil(k./y))`

ceil.

Rounds towards plus infinity.

ceil(3.4) = 4

ceil(3.999) = 4

ceil(-3.4) = ?

Other useful functions.

Try:

round(3.4999)
round(3.5001)

round = nearest integer.

Without typing help… what does FLOOR round to?

minus infinity.

FIX?

Towards zero.

CLEAR ALL

How to **use** a matrix: sum.

- Add two vectors: C = A + B:

```
A = [ 1 0 3]
B = [-1 4 2]
C = [ 0 4 5]
```

Corresponding elements are added in each cell.

Therefore, you cannot add vectors (or matrices) of different dimensions!

How to **use** a matrix: sum.

Type:

```
D = A + 4        % what happened?
```

--> be sure you specify to which element of a you'd like to add 4.

```
a(1) = a(1) + 4
```

So, in

```
 q = a(1) + 4       What's q?
```

How to **use** a matrix: multiplication.

- There are two important forms of multiplication:

  .* which multiplies all the values of a matrix by the same number.

  Type:

  ```
  > clear all;
  > a = [ 1 1 2; 3 4 5]
  > 2as = a.*2;  % 2.*a also works
  ```

How to **use** a matrix: multiplication.

- There are two important forms of multiplication:

  * which multiplies two matrices together. Hard math. To multiply A times B (A*B), A needs to have the same number of columns as B has rows.

  Try:

  ```
  >C = A * A                %what happened?
  >B = A'                 %what happened?
  >D = A * B
  ```

Other stuff.

So, be careful how you multiply.

But you can also subtract, divide
(both ./ and /, though matrix
division is OUT of our league).
^ is expontential (for numbers)

What is a script file?

Files of the type “.m” are simply a collection of orderly command- line instructions
(scripting language)

type:

```
> edit moo.m
```

and save it in your working directory.

script files

Inside moo.m type the following sequence of commands:

```
>str=’this is my first program’;
```

Save your file, close it and on the command line, execute it by typing:

```
>moo
```

script files

What happened? Why?

Allow your program to show you its output and run it again.

script files

Erase everything on moo and type:

```
a=[…
1 2 3 6
4 5 6 7.811];
```

Save moo and execute it.

CHECK WORKSPACE

script files

Two types of m files:

- programs (aka scripts)
  - don’t take inputs.
  - don’t produce outputs.
  - but leave variables in workspace.

- functions (tomorrow).
  - can take input (passing variables)
  - can return outputs.
  - variables are internal.

BREAK!!

Be back in 10 minutes.

Part II

- What is a computer?
- What is an algorhythm?
- What is a vector?
- Matrices and basic operations