Arrays
- The fundamental unit of data in MATLAB is the **array**.
- An array is a collection of data values organized into rows and columns and is known by a specified name.
- Individual data values within an array are accessed by a name and subscripts of indexes.
  - For example: abc(1,2)

Arrays²
- A single value is called a **scalar**
  - Matrix with only one row and one column
- This is a 4x6 **matrix**, containing 24 elements.
  
| 3 0 0 4 4 0 |
| 0 3 0 4 4 0 |
| 1 1 3 4 4 1 |
| 0 0 1 1 1 0 |
- [1, 2, 3, 4] is a 1x4 array, known as a **row vector**.
- A 4x1 array is called a **column vector**.

Arrays³
- The size of an array is specified by the number of rows and the number of columns.
- [ ] creates an empty array.
- Data in an array are listed in **row order**, i.e. the items in each row are listed from left to right, with the topmost row first.
  - Syntax: `var(row, col)`

Variables
- A MATLAB **variable** is a region of memory containing an array, which is known by a user-specified name.
- Variable names must start with a letter.
- They may contain letters, numbers and the underscore ( _ ).
- Names are case sensitive.
- The effective length of a name is 63 characters.

Variables²
- There are certain keywords you can’t use
  - Use `iskeyword` function to check whether a name is a system reserved name.
Variables

- MATLAB will let you use built-in function names as variables – but it’s a really bad idea.
  - What if you accidentally overwritten a function to a variable name?

Variables

- Use command `clear` (check Chap 1)
- Then how to clear an accidentally created variable named as `clear`?

Naming Variables

- test
- Test
- if
- mybook
- my_book
- Thisisoneverylongnamebutitisstillallowed?
- 1stgroup
- group_one
- zzaAbc
- zS4wAwy12#
- sin  bad
- log  idea

Basic Data Types

- The most common types of MATLAB variables are `double` and `char`.
- A variable of type `double` consists of 64-bit double-precision floating-point values.
- Variable of type `char` consist of scalars or arrays of 16-bit values.

Default Data Type

- By default, a variable of type `double` is automatically created whenever a numerical value is assigned to a name.
  - It can be positive or negative number in the range of $10^{-386}$ to $10^{386}$, with 15 to 16 significant decimal digits.

Assignment Operator

- `=` is the assignment operator
  - Syntax: `var_name = expression;`
- An expression is a value, an array, or a combination of values, other variables, and mathematical operations.
- To define a variable `a` we might type `a=10.5`
  - which should be read as: `a` is assigned a value of 10.5
Data Initialization

- MATLAB variables are automatically created when they are initialized.
  - \( a = 10.5 \)
  - \( b = 'abc', c = 4 + 3i \)
  - \( d = [1 2 3 4] \)

Creating Variables

- The size of an array is specified by the number of rows and the number of columns.
- \([\ ]\) creates an empty array.
- In MATLAB, transpose matrix \( MT \) is denoted as \( M' \), \( ' \) is a transpose operator
  - The transpose operator changes rows to columns or vice versa.
  - The transpose operator makes it easy to create tables

Creating Variables²

Creating Variables³
Creating Tables

- The transpose operator makes it easy to create tables.

Creating Arrays

- Evenly spaced numerical sequence can be created by shortcut expressions.
- The general form of a colon operator:
  - `first:increment:last`
  - This expression creates an array of `[first, first+incr, first+incr*2, ...]` until the value is greater than the last.
- If the stepping increment is one, then the middle argument can be ignored:
  - `first:last`

Shortcut Expressions

- Color operator can be used to represent a range of indexes, i.e. a sub-region inside an array.
  - `n:m` represents indexes from `n` to `m`.
  - `:` can be used to represent all the indexes.
  - `[p q ...]` represents positions of `p`, `q`, and etc.
Build-In Functions

- **zeros, ones, and eye** can be used to create all zeros, all ones, and identity matrices of any size.
- If the function has a single argument, it will produce a squared array of the given value.
  
  - For example: 
    
    \[ \text{eye}(2) \text{ is the same as } \text{eye}(2, 2) \]

- **length** returns the length of a vector, or the longest dimension of a matrix.
- **size** returns all the dimensions of a variable.
- Nested function: the output of an inner function is used as a corresponding argument of an outer function.
  
  - For example: to create a new array \( b \) by duplicating the size of another array \( a \):
    
    \[ b = \text{zeros} \left( \text{size}(a) \right) \]
Keyboard Input

- To allow a M-script to prompt a user for input interactively.
- The `input` function displays a prompt string in the command window and then waits for the user to type in a response at the keyboard.
  - `input(prompt_string)`
  - `input(prompt_string, string_option)`
- If no receiving variable is specified, the default variable `ans` is used.

Keyboard Input²

- If string option is used, then whatsoever entered will be treated as characters.

Homework Assignment #2

- Quiz 2.1
  - Page 36: 2, 3, 4, 5, 6, 7
- This assignment is due by next week.
- Late submission will be penalized.
- Hand-written homework will not be accepted or graded.
- Hand in a hardcopy of your homework, no electronic submission will be accepted.

Multidimensional Arrays

- Sometimes you may want to store data in multidimensional arrays
  - Rows
  - Columns
  - Pages
  - Additional dimensions are possible.
Multidimensional Arrays

- A multidimensional array can also be formed by adding 2D pages.

- We say that a matrix is stored in row major order if it is stored row by row. The entire first row is stored first. On the other hand, it is in column major order if it is stored column by column.
  - For example: consider the matrix
  
  \[
  A = \begin{bmatrix}
  1 & 2 & 3 \\
  4 & 5 & 6 \\
  7 & 8 & 9 \\
  \end{bmatrix}
  \]
  - In the row-major order layout:
  
  \[
  \begin{bmatrix}
  1 & 2 & 3 \\
  4 & 5 & 6 \\
  7 & 8 & 9 \\
  \end{bmatrix}
  \]
  - In the column-major order layout:
  
  \[
  \begin{bmatrix}
  1 & 4 & 7 \\
  2 & 5 & 8 \\
  3 & 6 & 9 \\
  \end{bmatrix}
  \]

Memory Allocations

- In MATLAB, elements of an array are allocated in column-major order.

- To access the memory directly, use single indexing, such as A(3).

Memory Allocations

- We can modify data by applying the modifications directly to the memory.

Sub-Arrays

- Use shortcut indexing `first:incr:last` to reference to a subarray.
  - For example: consider the matrix
    
    \[
    A = \begin{bmatrix}
    1 & 2 & 3 \\
    4 & 5 & 6 \\
    7 & 8 & 9 \\
    \end{bmatrix}
    \]
    
    \[A(1:2, :) = \begin{bmatrix} 1 & 2 & 3; 7 & 8 & 9 \end{bmatrix}\]
    
    \[
    A(6) = 8 \text{ and } A(8) = 6
    \]
The end Function

- The keyword **end** returns the highest index in an array subscript.
  - For example: consider the following matrix A
    
    $\begin{pmatrix}
    0 & 0 & 0 & 2 & 3 & 4 \\
    0 & 1 & 1 & 3 & 4 & 5 \\
    1 & 1 & 2 & 3 & 4 & 9
    \end{pmatrix}$
    
    $A(\text{end}, \text{end})$ is referenced to $A(4, 6)$

Reference to Sub-Arrays

- Predefined Keywords
  
<table>
<thead>
<tr>
<th>Keyword</th>
<th>Predefined value</th>
</tr>
</thead>
<tbody>
<tr>
<td>pi</td>
<td>3.141592653589793</td>
</tr>
<tr>
<td>i, j</td>
<td>$\sqrt{-1}$</td>
</tr>
<tr>
<td>inf</td>
<td>$\infty$, infinite</td>
</tr>
<tr>
<td>NaN</td>
<td>Not-a-Number</td>
</tr>
<tr>
<td>clock</td>
<td>current date and time</td>
</tr>
<tr>
<td>date</td>
<td>current date in string form</td>
</tr>
<tr>
<td>eps</td>
<td>smallest difference between numbers</td>
</tr>
<tr>
<td>ans</td>
<td>the variable stores the unassigned result</td>
</tr>
</tbody>
</table>

Homework Assignment #3

- Quiz 2.2
  - Page 44: 1, 2, 3
- This assignment will due next week.
- Late submission will be penalized.
- Hand-written homework will not be accepted or graded.
- Hand in a hardcopy of your homework, no electronic submission will be accepted.