

Fundamentos de Matlab:

Tabla de Contenido

Variables en Matlab.....	1
Comandos iniciales.....	1
Definición de variables:.....	1
Vectores equiespaciados	2
Creación de Arreglos mediante Funciones de matlab:.....	3
Acceso a valores en un arreglo.....	3

Variables en Matlab

```
a = 19
```

```
a =  
19
```

Comandos iniciales

En la ventana de comandos se pueden ingresar operaciones y comandos propios del lenguaje, algunos comandos útiles son

clc : Borra los elementos de la ventana de comandos

clear all Borra las variables creadas en la actual sesión (workspace)

save NombreArchivo: Guarda las variables del Workspace en un archivo llamado NombreArchivo con extensión .mat

load NombreArchivo: Carga al espacio de trabajo las variables almacenadas en el archivo NombreArchivo.mat si este existe. en el path actual

Definición de variables:

Para Matlab **todas** las variables son arreglos, Matlab por defecto asigna de manera dinámica el espacio que ocupan las variables. las variables pueden contener datos numéricos, funciones, objetos.

para crear variable se debe definir un nombre y a él asignarle el contenido mediante el operador de asignación de matlab "="

Las variables deben comenzar por **una letra** no pueden contener espacios y algunos caracteres que no soporte el sistema:

práctica defina una variable de nombre miCasa y asígnele un valor.

```
miCasa = 2.5
```

```
miCasa =  
2.5000e+00
```

Los Arreglos se definen mediante el uso de corchetes [], los elementos de un arreglo pueden ser separados por comas o por espacios:

ejemplo

`x = [1]` es un arreglo de 1X1

`y = [65 23]` es una arreglo de 1X2

Las filas de una arreglo estan deben estar separadas por un *punto y coma (;)*

`A = [3 4 5 ; 6 7 8]` es en arreglo de 2 X 3

Práctica... cree un arreglo de 4 x 4

```
% Escriba acá las sentencias
```

Vectores equiespaciados

Para definir vectores equiespaciados se usa la notación de *dospuntos (:)*

`a = 1:10` crea un vector fila con numeros entre 1 y 10

```
clear all
a = 1:10
```

```
a = 1×10
    1     2     3     4     5     6     7     8     9    10
```

Para definir los incrementos personalizados se incluye el incremento entre *dos puntos* así

`b = 3:0.5:10` crea un arreglo entre 3 y 10 con incrementos de 0.5

práctica...

```
% Escriba acá las sentencias
b = 3:0.5:10
```

```
b = 1×15
    3.0000e+00    3.5000e+00    4.0000e+00    4.5000e+00    5.0000e+00    5.5000e+00    ...
```

Tambien se puede crear un arreglo equiespaciado si se conocen los valores extremos del vector y el numero de elemntos. para eso se hace uso de la funcion ***linspace***. su formato es:

`linspace(primerElemento,UltimoElemnto,númeroelemntos)`, así;

`x = linspace(5,20,8)`: crea un vector fila equiespaciado entre 5 y 20, el cual contiene 8 elementos

práctica...

```
clear all
x = linspace(5,20,8)
```

```
x = 1×8
    5.0000e+00    7.1429e+00    9.2857e+00    1.1429e+01    1.3571e+01    1.5714e+01 ...
```

Creación de Arreglos mediante Funciones de matlab:

Matlab incorpora algunas funciones especiales para para crear matrices, algunas de ellas:

ones(param1,param2): crea una matriz llena de unos de la dimensión especificada en los parametros.

ones (3,2) :crea una matriz de 3X2 donde todos sus elementos son 1

zeros(param1,param2): crea una matriz llena de ceros de la dimensión especificada en los parametros.

eye (n) crea una matriz cuadrada nXn donde los elementos de su diagonal principal todos son unos

práctica...

```
clear all
ones (3,2)
```

```
ans = 3×2
     1     1
     1     1
     1     1
```

```
eye(4)
```

```
ans = 4×4
     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
```

Acceso a valores en un arreglo.

Los valores de una rreglo son accesibles mediante su indice, se puede acceder a un solo elemento o a un grupo de el elementos haciendo uso de comodines de la siguiente manera, dado el arreglo:

```
A =
4    10    13
22     5     4
17    11    15

a = A(1,3) devuelve el elemento de la fila 1 y la columna 3
b = A(:,2) devuelve todas las filas de la columna 2
c = A(1,:) devuelve todas las columnas de la fila 1
d = A(1:2, 3) devuelve un vector columna con los elementos 1 y 2 de la columna 3
```

```
clear all
A = randi(15,3)
```

con el indicador *end*, se puede acceder a la última fila o a la última columna de un arreglo de la siguiente manera

por ejemplo dada la matriz A.

```
A =
1     8     5     7     4     7     9
6     7     7     5     2    13    10
5    10     9     2     4     9     1
3    10     6     8     6    13     4
3     9    11    11     7     4     3
6     1    10     6     6     9     9
2     1    13     2    12     4    11
```

se puede acceder a su última fila de la siguiente manera

```
v = A(end,:) devuelve
```

```
v =
2     1    13     2    12     4    11
```

o acceder al último elemento así

`c = A(end,end)` devuelve

`c = 11`

Práctica

```
v = A(end,:)
```

```
v = 1×3  
    9     3     8
```

```
c = A(end,end)
```

```
c =  
    8
```