

Guía breve de introducción a Matlab

El comando más importante de Matlab:

HELP

Permite obtener ayuda de cómo se usa un comando o la definición del mismo. En la gran mayoría de los casos incluye ejemplos muy claros de la sintaxis del mismo. En las versiones más recientes (a partir de la 6 ó 6.5) la ayuda se puede manejar en una ventana aparte, accediendo a la misma donde dice HELP en la parte superior de la ventana principal.

```
>> help  
>> help fprintf
```

El símbolo >> indica la línea de comando, en la cual se escribe directamente lo que queremos hacer.

```
>> 5+3
```

Operadores aritméticos: +, -, *, /, ^,... (ver help +)

Principales Variables: (Matlab diferencia entre a y A “case sensitive”)

Reales o enteras: a=5	a=5.4
Vectores: v=[1 2 3]	v=[1,2,3]
Matrices: M=[1 2;3 4]	M=[1,2;3,4]
Carácter: c='c'	

Manejo de Variables:

```
v=[2 4 6]    => v(2)=4  
M=[1 3;5 7]  => M(2,1)= 5 = M(2)
```

Algunos comandos:

- **who:** lista las actuales variables
- **whos:** lista las actuales variables con su tamaño y tipo
- **clear:** borra todas las variables actuales (también admite clear var)
- **clc:** borra la ventana de comandos
- **fprintf:** imprime con formato
fprintf('Esto es un ejemplo'), fprintf('Ejemplo # %d',var)
- **load:** carga variables desde archivo al espacio de trabajo
- **save:** salva a archivo las variables del espacio de trabajo

- **display:** muestra en pantalla
display(var); display('Otro ejemplo')
- **input:** permite la introducción de datos a través de consola
a=input('Introduzca la edad')
- **if:** instrucción condicional
- **for:** permite repetir instrucciones un número determinado de veces
for i=1:10
....
end
- **while:** permite repetir instrucciones un número indeterminado de veces
while n ~=10
....
end
- **end:** termina ciclos y condicionales
- **break:** rompe con la ejecución de ciclos
- **plot:** grafica linealmente
x=[1:4];y=[4 2 5 7];
plot(x,y);
- **hold:** mantiene el gráfico actual
- **size:** calcula el tamaño de un arreglo
- **length:** calcula la longitud de un arreglo
- **abs:** valor absoluto
- **sort:** orden de un arreglo
- **sqrt:** raíz cuadrada
- **'** (comilla simple): transpuesta de un arreglo
- **& , | , ~:** operadores lógicos: and, or, not

Otros comandos: find, any, all, eye, zeros

Uso de ; : para evitar la impresión de cada instrucción o resultado de una operación se debe usar punto y coma(;) al final de cada instrucción.

Para escribir comentarios se usa porcentaje (%). En toda línea, a partir de un %, todo lo que esta a la derecha es comentario

script y function

Para trabajar cómodamente Matlab permite guardar una secuencia de instrucciones en un archivo de texto cuya extensión debe ser .m , esto se puede hacer desde cualquier editor de texto. Matlab tiene su propio editor, el cual permite ver en distintos colores las diferentes instrucciones y comandos.

Para ejecutar dichos archivos, sólo es necesario hacer el llamado desde el espacio de trabajo.

Ejemplo de **script**

```
% Esto es un script
a=5;
b=6;
fprintf('El resultado es %d \n',a+b)
% fin archivo prueba.m
```

Desde el espacio de trabajo hacer:

```
>> prueba
El resultado es 11
>>
```

Una **function** es un **script** pero con un encabezado obligatorio. Una **function** permite definir parámetros de entradas y de salida. Las variables dentro de una **function** son de tipo local, mientras que las de un **script** son de tipo global.

Ejemplo de una **function**

```
function salida=prueba2(entrada1,entrada2)
n=entrada1*entrada2;
salida=0;
for i=1:n
    salida=salida+1;
end
% fin de la function prueba2
```

Desde el espacio de trabajo hacer:

```
>> salida=prueba2(3,4)
```

```
salida =
```

```
12
```

```
>>
```

Importante: lo único que puede ir antes de una instrucción **function** son comentarios.