



Computación

Curso 2024 – Prof. Jorge Runco

Archivos

Guardar datos

- Copiar (al portapapeles) desde la Ventana de Comandos de Matlab y pegar en un archivo de texto.
- Guardar datos en un archivo binario o ASCII usando la función `save`.
- Guardar planillas de cálculo, datos científicos, imágenes o audio con funciones que vienen con las herramientas de Matlab.
- Guardar datos en un archivo utilizando las operaciones de Entrada/Salida que proporciona Matlab (`fwrite`, `fprintf`, etc.)

Recuperar datos

- Ingresar (manualmente) o pegar datos en la Ventana de Comandos.
- Crear un script para inicializar matrices u otras estructuras de datos.
- Cargar archivos binarios o ASCII utilizando load
- Cargar planillas de cálculo, datos científicos, imágenes o audio con funciones que vienen con las herramientas de Matlab.
- Cargar datos desde un archivo utilizando las operaciones de Entrada/Salida que proporciona Matlab (fread, fscanf, etc.).

Guardar datos usando save

- `save nombre_de_archivo`

Guarda todas las variables en el archivo especificado.

- `save nombre_de_archivo variables`

Guarda solamente las variables especificadas (separadas por espacios) en el archivo especificado.

- save nombre_de_archivo opciones

Guarda todas las variables en el archivo especificado, utilizando alguna de las siguientes opciones:

- append	Agrega variables nuevas a un archivo preexistente
- ascii	Guarda datos en formato ascii, números hasta 8 dígitos
- ascii - tabs	Idem anterior, columnas separadas por tabuladores
- ascii - double	Guarda datos en formato ascii, números hasta 16 dígitos
- ascii - double - tabs	Idem anterior, columnas separadas por tabulador
- mat	Guarda datos en formato binario (por defecto)



Cargar datos usando load

- Load nombre_de_archivo
Carga las variables desde el archivo especificado.
- Load nombre_de_archivo variables
Carga solamente las variables especificadas (separadas por espacios) desde el archivo especificado.
- Load -ascii nombre_de_archivo
Carga las variables desde el archivo especificado, tratándolo como si su contenido estuviera en ASCII.

E/S de “bajo nivel”

- fopen

Abre un archivo.

- fprintf

Da formato a los datos y los escribe en un archivo.

- fscanf

Lee datos con formato de un archivo.

- feof

Verifica que no se haya llegado al final del archivo.

- fclose

Cierra un archivo abierto.



Escribir un archivo con fprintf

- ➡ 1: Abro el archivo para escribirlo.

`fopen`

- ➡ 2: Mientras haya datos para guardar

- ➡ 2.1: Guardo los datos.

`fprintf`

- ➡ 3: Cierro el archivo.

`fclose`

Leer datos del archivo con fscanf

- ➡ 1: Abro el archivo para leerlo.

fopen

- ➡ 2: Mientras haya datos para leer,

feof

- ➡ 2.1: Leo los datos.

fscanf

- ➡ 2.2: Realizo operaciones con los datos.

- ➡ 3: Cierro el archivo.

fclose

Algunas consideraciones

- La operación `fopen` retorna un número (denominado handler) que representa al archivo abierto. Las demás operaciones utilizan ese handler para trabajar con el archivo.
- Una vez abierto el archivo, todas las operaciones `fprintf` agregan nuevos datos.
- Los archivos poseen un puntero de lectura. Cada vez que se invoca la operación `fscanf`, el puntero avanza tantos bytes como datos se hayan leído.

- 
- 
- Se puede consultar si el puntero llegó al final del archivo (end of file, o eof) utilizando la operación feof.
 - Hay que recordar siempre cerrar los archivos con fclose.



➤ handler = fopen(nombre_de_archivo,
tipo_de_acceso)

(fid) =

➤ Abre el archivo para el tipo de acceso especificado. Los tipos de acceso son:

'r'	Abrir el archivo para leerlo.
'w'	Abrir el archivo, o crearlo, para escribirlo. Si existe contenido, descartarlo
'a'	Abrir el archivo, o crearlo, para escribirlo. Agregar nuevo contenido al final del archivo

- 
- 
- ▶ `fprintf (handler, formato, A, ...)`
 - ▶ ¿Qué es el formato?
 - ▶ El formato es un texto (string) que contiene caracteres ordinarios y caracteres especiales, denominados caracteres de conversión.

- Caracteres de conversión

Se utilizan para controlar la notación, alineación, dígitos significativos, ancho del campo y otros aspectos de un valor escalar o una matriz. Comienzan siempre con %.

- Caracteres de escape

Se utilizan para representar teclas de control del teclado. Ejemplos de teclas de control: enter, tab, esc, etc. Comienzan siempre con \.



Formato de datos (carac. conversión)

- ❖ %d Enteros.
- ❖ %f Reales con punto fijo.
- ❖ %e Reales con formato exponencial.
- ❖ %g Utiliza uno de los formatos anteriores; el que dé la mayor precisión en el menor espacio.



Caracteres de escape

- `\n` Nueva línea (Enter).
- `\t` Tabulador (Tab).

Ejemplo

```
x = 0:1:1;
```

```
y = [x; exp(x)];
```

```
%
```

```
fid = fopen('exp.txt', 'w');
```

```
fprintf(fid, '%6.2f %12.8f\n', y);
```

```
fclose(fid);
```

```
%
```

```
fid = fopen('vamos.dat');
```

```
A = fscanf(fid, '%d %d', [2 inf]);
```

```
fclose(fid);
```



Otro ejemplo

- nombre='Jorge';
- edad=58;
- fprintf('Mi nombre es %s y tengo %d años', nombre, edad);
- ✓ La salida por defecto es la pantalla. Al no poner el identificador de archivo el texto aparece en pantalla.