Fundamentos de Matlab Sesión 3-3.

Tabla de Contenido

Programación de Scripts:

Los Scripts de código en matlab se guardan econ extensión **.m**.Estos archivos (tambien conocidos como m files) son archivos de texto plano que contienen una secuencia ordenada de comandos MATLAB (instrucciones primitivas, funciones y/o llamadas a otros *scripts*) escritos tal y como se introducirían en la ventana de comandos y pueden ser creados en cualquier editor de texto pero para efectos de pruebas y depuración de códgo se recomineda trabajar con el editor de archivos de Matlab.

Para ejecutar un script de código basta con ejecutar en la consola de comandos el nombre del archivo.

Los scripts utilizan variables almacenadas en el *workSpace,* y las variables creadas en el xsript tambien se crean en el *workSpace.*

Comentarios:

Es recomendable comentar las códigos dentro de un Script, esto sirve para ayudar a nuevos usuarios a entender el codigo o para usos futuros del kismo. Los comentarios pueden ser de tres tipos:

- *Header commnent:* son comentarios al inicio del archivo generalmente empiezan con un doble signo de %, sirven como texto de ayuda en la consola de comandos (al escribir help... nombre del archivo.)
- *Function Comment:* son comentarios al inicio de funciones, se usan para describir los pareámetros y los valores que devuelve la función. Usan un solo simbolo de %
- *in line Comment:* Son comentarios en línea de una expresión se emplean para detallar la expresión.

Los comentarios multilnea se abren con ***%{*** y se cierran con ***%}***

```
%{
comentario
en varias
lineas
%}
```

Algunas Funciones Útiles:

- ***pause(n)****:* Pausa la ejecución un número x de segundos
- ***disp('texto')***: Muestra en la salida el texto entre comillas simples
- ***figure(n):*** abre un cuadro para graficas y le asigna el número n
- ***echo on / off :*** muestra las expresiones en la salida

- ***... (tres puntos):*** Se emplea para introducir para expresiones multilinea.
- ***clear:*** Borra las variables que se encuentren en memoria y el el workspace
- ***close:*** Cierra las ventanas de figuras que se encuentren abiertas
- ***save:*** Guarda las variables del workspace
- ***load:*** Carga las variables guardadeas o archivos de datos
- ***clc:*** Borra lel contenido de la consola de comandos.
- ***home:*** Regresa al inicio de página de la consola de comandos sin borrar su contenido.
- ***format:*** Selecciona el formato de visualizacion numérica en la consola de comandos

Ejemplo de un Script:

```
%% Ejemplo de código en un script

% Definición de variables

A = [2,4,5; 3,6,7];    % A es una matriz de 2X3
B  = magic (4);
C = pi + A;            % Suma la constante pi a cada valor dentro de la matriz A
```

Programación de Funciones:

Un archivo M tipo *function* también es una secuencia ordenada de comandos MATLAB pero, a diferencia de los *scripts*, para su ejecución necesitará que se le introduzcan argumentos de entrada y, como respuesta, generará argumentos de salida.

La sintaxis para declarar funciones es la siguiente:

```
% función que devuelve la suma y la resta de dos números
function [sum, res] = sum_rest(a,b)
sum = a+b;
res = a-b

end
```

El nombre del archivo con el que se debe guardar una función es el mismo nombre de la función.

en un archivo se pueden declarar varias funciones una a continuacion de otra pero el nombre del archivo debe ser el nombre de la primera función.

Argumentos de una función:

Las funciones en Matlab pueden recibir ninguno, uno o varios argumentos de entrada, Matlab ofrece la posibilidad de controlar el flujo de la función en respuesta al número de argumentos que se le presenten y paraesto cuenta con funciones especiales comok:

- ***nargin:*** Número de argumentos de entrada:
- ***nargout:*** Número de argumentos de salida
- ***varargin:*** Número de argumentos de entrada variable
- ***varargout:*** Número de argumentos de salida variable

- ***nargchk***: Prueba el numero de argumentos de entrada
- ***naraoutchk:*** Prueba el número de argumentos de salida:

Tipos de funciones:

- ***Funciones Anidadas***: Son funciones declaradas dentro de otra función siempres dse deben terminar con ***end***
- ***Funciones Locales:*** Son funciones creadas en la ventana de comandos (no se pueden anidar y solo tienen un argumento de salida).
- ***Funciones privadas:*** Son las funciones que están dentro de los directorios de nombre <*private*>. Sólo pueden ser vistas y llamadas por las funciones y *scripts* que están en el directorio inmediatamente superior al directorio <*private*>. Puesto que sólo son visibles por unas pocas funciones su nombre puede ser el mismo que otras funciones estándar de MATLAB, por ejemplo, *bode.m*.
- Las funciones en Matlab pueden recibir otras funciones como argumentos, para ello hace uso del signo símbolo de @ como manipulador de la función (*handler function*), para usarlo en el argumento se escribe el símbolo de @ seguido de la función que se quiera utilizar.
- ***Funciones anónimas***

Estructuras de Control:

Al igual que otros lenguajes de programación Matlab ofrece varias posibilidades para controlar el flujo de un script:

Bucles:

Los bucles se emplean para repetir un número determinado de veces un grupo de comandos (bucle ***for***) ó para repetirlo hasta que se cumpla cierta condicion (bucle ***while***).

- ***Bucle for:*** su sintaxis basica es:

```
for index == valor

    sentencias a ejecutar

end
```

index es el índice del bucle, mientras que *valor* puede tener varias formas

* *valor inicial**:**valor final. se ejecuta con incrementos de 1*

* *valor inicial**:incremento:**valor final.*

* *val array: El* valor es un arreglo de cualquier dimension,

* *val vector: E*l valor es un vector fila, el index toma cada valor del arreglo hasta el final del mismo.

- ***Bucle while:*** se ejecuta mientras la condición sea verdadera, su sintaxis básica es

```
while expresion
sentencias
end
```

Para salir de un bucle de manera arbitraria use la sentencia ***break*** en el sitio en donde quiera terminar la ejecución del bucle o si se cumple una condición irregular.

Condicionales:

Permiten ejecutar parte de un código si se cumple la condición., el condicional mas básico es la sentenciaa ***if else.***

- ***if*** el bloque de código se ejecuta si se cumple la condición. su sintaxis es:

```
if condicion
 sentencias   % si se cumple la condición
else
sentencias % si NO se cumple la condición
end
```

Se puede usar un i***f*** sin ***else*** y se pueden anidar tantos i***f*** como estime conveniente.

- ***switch case:*** Se utiliza cuando se require probar la ***igualdad*** de una variable frente a un número conocido de posibles valores. Su sintaxis ;:

```
switch nombrevariable
case valor1
sentencias      % se ejecutan si nombrevariable es igual a valor 1
case valor2
sentencias      % se ejecutan si nombrevariable es igual a valor 2
case valorn
sentencias      % se ejecutan si nombrevariable es igual a valor n
otherwise
sentencias      % se ejecutan si nombrevariable NO es igual a ninguno de los valores dentro del conjunto
end
```

Acá *nombrevariable* es el nombre de la variable del que se quiere comprobar su igualdad con un grupo de valores, en este caso, desde *valor 1* hasta *valor n.* la sentencia *otherwise* es opcional y puede omitirse.