

Instrucciones if/else y while

```
if expresión lógica
    %instrucciones MATLAB
elseif expresión2 %opcional
    %instrucciones
... %otros elseif
else %opcional
    %instrucciones
end
while expresión lógica
    % instrucciones MATLAB
end
```

Funciones con varios resultados

```
%f23(x): calcula x al cuadrado y al cubo
function[xx,xxx] = f23(x)
xx = x.^2;
xxx = x.^3;
>>[ a b ] = f23(2)
a =
    4
b =
    8
>>f23(2)
ans =
    4
```

Operador	símbolo	prioridad
(expresión)	()	0
potencia	^ .^	1
unarios	+ - ~ (negación)	2
mult, división	* / \ .* ./ .\	3
Suma, resta	+ -	4
Operador :	:	5
relacional	< <= > >= == ~=	6
AND	&	7
OR		8
asignación	=	9

Funciones internas (locales)

```
%f23(x): calcula x cuadrado y x cubo
function[xx,xxx] = f23(x)
xx = x.^2;
xxx = cubo(x);

function y=cubo(x) %visible en archivo
y = x.^3;
```

Instrucción for

```
>>for i=1:10 %for variable=vector
    disp(i);    %instrucciones
end

>>a=1:10;
>>for i=a
    disp(i);
end

>>a=1:10;
>>for i=1:length(a)    %largo de arreglo
    disp(i);
end
```

Raices de ecuación de 2º grado

```
%raices(a,b,c): raíces de axx+bx+c=0
function[r1,r2] = raices(a,b,c)
d=b^2 - 4*a*c; %discriminante
if d < 0
    error('raices imaginarias'); %aborta
elseif d = 0
    warning('raices iguales');%muestra mensaje
end
r1=(-b + sqrt(d))/(2*a);
r2=(-b - sqrt(d))/(2*a);

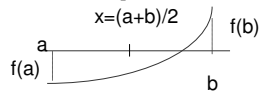
>>[x y]=raices(1,4,3) >>[x y]=raices(1,1,4)
x=-1 y=-3                raices imaginarias
```

Ejercicio: Calcular raíz por el método de búsqueda binaria

Algoritmo:

- determinar x como punto medio del intervalo
- si $f(x)$ tiene el mismo signo de $f(a)$ entonces repetir el proceso en intervalo $[x,b]$
- si $f(x)$ tiene el mismo signo de $f(b)$ entonces repetir el proceso en intervalo $[a,x]$

Nota. Las iteraciones se detienen cuando el tamaño del intervalo de búsqueda alcance un epsilon.



```
function r=raiz(a,b,eps)
```

“Escritura” formateada a un string

a=sprintf(formato,X,...);

Escribe el dato X en el string a bajo el control del string formato

Ejemplos:

```
>> a=sprintf('x=%g',pi)
```

```
a =
```

```
x=3.14159
```

```
%Jalisco: nunca pierde
```

```
n=input('nº? ');
```

```
disp(sprintf('gano yo con %g',n+1));
```

Notas

- %g se reemplaza por el valor de X
- %s se usa para strings

Solución iterativa

%raiz(a,b,eps): raíz de f en $[a,b]$ c/prec eps

function r=raiz(a,b,eps)

```
while b-a > eps
```

```
    x=(a+b)/2;
```

```
    if signo(f(x))==signo(f(a)) a=x; else b=x;
```

```
    end
```

```
end
```

```
r=(a+b)/2
```

%signo(x) devuelve -1, 0 o 1

function r=signo(x)

```
if x<0 r=-1; elseif x>0 r=1; else r=0; end
```

Ejemplo con strings

```
>>a=rand(2,3);
```

```
>>[filas cols]=size(a);
```

```
>>for i=1:filas
```

```
    linea=sprintf('fila %g: ',i);
```

```
    for j=1:cols
```

```
        linea=sprintf('%s%g ',linea, a(i,j));
```

```
    end
```

```
    disp(linea);
```

```
end
```

```
fila 1: 0.921813 0.176266 0.93547
```

```
fila 2: 0.738207 0.405706 0.916904
```

Solución recursiva

%raiz(a,b,eps): raíz de f en $[a,b]$ c/prec eps

function r=raiz(a,b,eps)

```
x=(a+b)/2;
```

```
if b-a <= eps
```

```
    r=x;
```

```
else
```

```
    if signo(f(x)) == signo(f(a))
```

```
        r=raiz(x,b,eps);
```

```
    else
```

```
        r=raiz(a,x,eps);
```

```
    end
```

```
end
```

Grabar y leer arreglo

%grabar arreglo

```
datos=zeros(2,100);
```

```
datos(1,:)=linspace(0,2*pi,100);%angulos
```

```
datos(2,:)=sin(datos(1,:)); %senos de angulos
```

save datos

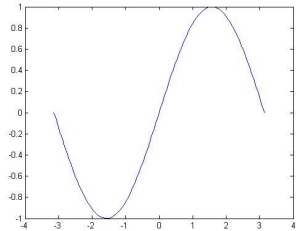
%leer arreglo

```
load datos
```

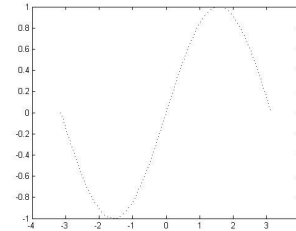
```
x=datos(1,:);
```

```
y=datos(2,:);
```

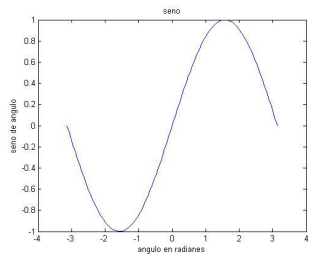
```
>>x=linspace(-pi,pi,100); plot(x,sin(x))
```



```
plot(x,y, 'k:'); %negro y puntos
```



```
title('...') xlabel('...') ylabel('...')
```



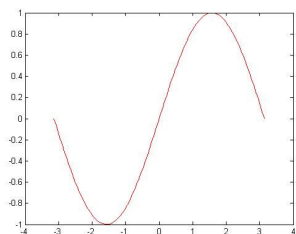
```
plot(x,y, '...')
```

- Various line types, plot symbols and colors may be obtained with
- `PLOT(X,Y,S)` where `S` is a character string made from one element
- from any or all of the following 3 columns:

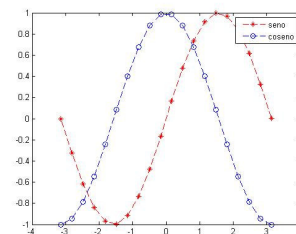
b	blue	.	point	-	solid
g	green	o	circle	:	dotted
r	red	x	x-mark	-.	dashdot
c	cyan	+	plus	--	dashed
m	magenta	*	star		(none) no line
y	yellow	s	square		
k	black	d	diamond		
		v	triangle (down)		
		^	triangle (up)		
		<	triangle (left)		
		>	triangle (right)		
		p	pentagram		
		h	hexagram		

- For example, `PLOT(X,Y,'c+')` plots a cyan dotted line with a plus
- at each data point; `PLOT(X,Y,'bd')` plots blue diamond at each data
- point but does not draw any line.

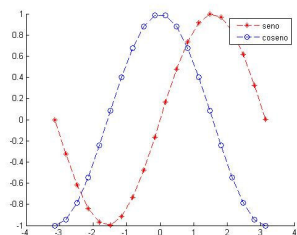
```
plot(x,y, 'r') %color red
```



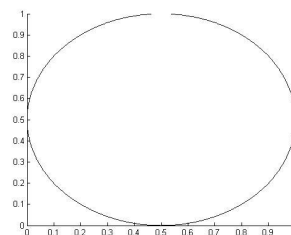
```
>> x=linspace(-pi,pi,20);
>> plot(x,sin(x), 'r-*', x,cos(x), 'b-o')
>> legend('seno', 'coseno')
```



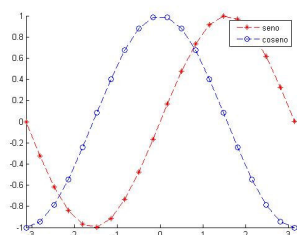
```
>> hold on
>> plot(x,sin(x),'r--*')
>> plot(x,cos(x),'b--o')
>> legend('seno','coseno')
```



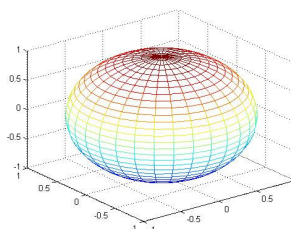
```
>> rectangle('Curvature',[1 1])
```



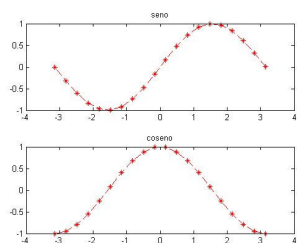
```
axis([-pi,pi,-1,1])
```



```
[x y z]=sphere(30);
mesh(x,y,z)
```



```
>> subplot(2,1,1), plot(x,sin(x),'r--*'), title('seno')
>> subplot(2,1,2), plot(x,cos(x),'r--*'), title('coseno')
```



```
surf(x,y,z)
```

