

Diplomado : Herramientas de Programación para Ciencias e Ingeniería

Módulo: MATLAB (Clase 4)

Docente: Juan Sebastián Salcedo Gallo

Universidad Nacional de Colombia Sede Manizales

Contenido

- Funciones
- Entrada y salida

¿Qué es una función?

En computación, una subrutina o subprograma (también llamada procedimiento, función o rutina), se presenta como un subalgoritmo que forma parte del algoritmo principal, el cual permite resolver una tarea específica.

Podemos decir que una función es una parte de un programa con un nombre, que puede ser invocada cuantas veces sea necesario, desde cualquier sección del código principal.

¿Qué es una función?

Una función permite organizar el código puesto que pueden “sectorizarse” tareas en scripts secundarios y llamar a la función para que retorne los parámetros de interés, cuando sea necesario dentro del código. Generalmente, una función recibe parámetros de entrada (opcional), se ejecutan las instrucciones dentro de la función, y se devuelve un valor.

No confundir con una clase. Puede que una “clase” tenga una estructura similar. Sin embargo, se usa en programación orientada a objetos para el desarrollo de aplicaciones de cálculo técnico complejas.

Veamos un ejemplo MUY simple

En el editor, veríamos algo así ...

```
function print_message(x)
    disp(x)
endfunction
```

Nombre de la función

Argumento de la función

Lo que retorna la función

Y en la ventana de comandos ...

```
>> print_message("¡Hola Mundo! Esta es mi primera función en MATLAB")
¡Hola Mundo! Esta es mi primera función en MATLAB
>> |
```

Función input de MATLAB

- input : solicitar entrada de usuario

Sintaxis

```
x = input(prompt)
str = input(prompt, 's')
```

Descripción

`x = input(prompt)` muestra el texto del *prompt* y espera a que el usuario introduzca un valor y pulse la tecla **Enter**.

El usuario puede introducir expresiones como `pi/4` o `rand(3)`, y puede utilizar variables del espacio de trabajo.

Función input de MATLAB

- Si el usuario pulsa la tecla Enter sin introducir nada, input devuelve una matriz vacía.
- Si el usuario ingresa una expresión no válida, MATLAB se lo hará saber.

```
>> %Vamos a asignar a la variable x el valor de 5.  
>> %usando input  
>> prompt = "Hola, escriba 5 para asignarlo a la variable x";  
>> x = input(prompt)  
Hola, escriba 5 para asignarlo a la variable x5  
x = 5  
>>
```

Función input de MATLAB

- Si queremos devolver el texto introducido, sin evaluar la entrada como una expresión.

```
>> prompt = "Esto se escribirá pero no se evaluará ";  
>> str = input(prompt, "s")  
Esto se escribirá pero no se evaluará "Hola Mami, este es el mensaje que quiero escribir"  
str = "Hola Mami, este es el mensaje que quiero escribir"  
>> % se asigna a la variable str, con clase char!!!;  
>>
```


Ejemplos función input

Solicite una entrada numérica y multiplique la entrada por 10.

```
prompt = 'What is the original value? ';  
x = input(prompt)  
y = x*10
```

En el símbolo del sistema, especifique un valor numérico o un array, como 42.

```
x =  
42
```

```
y =  
420
```

En el símbolo del sistema, escriba `magic(3)`.

```
x =  
      8      1      6  
      3      5      7  
      4      9      2
```

```
y =  
     80     10     60  
     30     50     70  
     40     90     20
```

Ejemplos función input

Solicite una respuesta de texto simple que no requiera evaluación.

```
prompt = 'Do you want more? Y/N [Y]: ';  
str = input(prompt, 's');  
if isempty(str)  
    str = 'Y';  
end
```

La función `input` devuelve el texto exactamente como se escribe. Si la entrada está vacía, este código asigna un valor predeterminado, 'Y', a `str`.

Ejemplo función input

- Calcule la suma de los n primeros términos de la sucesión

$$\sum_{i=1}^n ix^{(i-1)}$$

Ejemplo función input

```
>> sum_over_n
¿Cuantos terminos quieres sumar? 4
Dame el valor de la variable x 4
El valor que quieres calcular es:
  313
>>
```

Los introduces
tú!!!

Resultado

Ejercicio

Calcule el factorial de un número que usted pueda ingresar arbitrariamente con la función `input`. Compruébelo con el comando `factorial(n)`.

Solución

```
1 n=input('escribe el numero a que le quieres calcular su factorial: ');
2 f=1;
3 for i=1:n
4     f=f*i;
5 end
6 disp(f)
7
8
```

```
>> factorial_
escribir el valor del numero del que se quiere hacer el factorial: 5
120
```

Funciones en MATLAB

Sintaxis

```
function [y1,...,yN] = myfun(x1,...,xM)
```



Descripción

Puede guardar su función en un archivo que solo contenga definiciones de función. ¡El nombre del archivo debe coincidir con el nombre de la primera función del archivo!

Utilice la palabra clave **end**, para mejor legibilidad.

Ejemplo simple de una función en MATLAB

```
function y = square(x)  
    y = x*x;  
end
```

```
>> y = square(2)  
y = 4  
>>
```


Ejercicio

- Decidir si un número es par o no, cree una función para esta tarea. La función debe imprimir el veredicto.

Solución

```
function is_par_(x);  
    if (mod(x,2) == 0);  
        disp("El numero es par")  
    else  
        disp("El numero es impar")  
    end  
end
```

```
>> is_par_(2019)  
El numero es impar  
>> |
```

Ejercicio

- Defina una función en un archivo denominado promedio.m que acepte un vector de entrada, calcule el promedio de los valores y devuelva un solo resultado.

Solución

```
function y = promedio(x)
if ~isvector(x)
    error('Input debe ser un vector')
end
y = sum(x)/length(x);
end
```

```
>> x = (1:10);
>> promedio(x)
ans = 5.5000
>> |
```

Ejercicio

- Emplee una función para determinar si un estudiante pasa o no la asignatura, teniendo en cuenta que la nota numérica se encuentra entre 0 y 5.0, y que la nota mínima para aprobar es 3.0. Dé el veredicto para un número total de 10 estudiantes.

Solución

```
1 function pass = pass_or_not(nota)
2   if (nota < 0 || nota > 5.0);
3       pass = "not valid";
4   elseif (nota >= 3.0 && nota <= 5.0);
5       pass = true;
6   else
7       pass = false;
8   endif
9 endfunction
```

10
11
12

```
>> x = [0.0 1.0 3.3 4.3 5.0 4.4 4.3 4.1 4.0 2.1];
>> for i=1:length(x);
    pass = pass_or_not(x(i))
endfor;
pass = 0
pass = 0
pass = 1
pass = 1
pass = 1
pass = 1
pass = 1
pass = 1
pass = 1
pass = 0
>> |
```

Ejercicio

Cree un programa para calcular las raíces de un polinomio de orden 2. Usando una función para calcularlas, y después realice un script donde el usuario pueda ingresar el valor de los coeficientes a , b y c , y que este mismo script imprima los valores de las dos raíces.

Solución

```
roots_m  user.m
```

```
1 function [m, s] = roots_(a,b,c);
2     raiz = sqrt(b^2 - 4*a*c);
3     m = (-b + raiz)/(2*a);
4     s = (-b - raiz)/(2*a);
5 end
```

```
>> user
ingrese el valor de a: 1
ingrese el valor de b: 2
ingrese el valor de c: 4
la raiz 1 es:
-1.0000 + 1.7321i
la raiz 2 es:
-1.0000 - 1.7321i
>> |
```

```
roots_m  user.m
```

```
1 a_ = "ingrese el valor de a: ";
2 a = input(a_);
3 b_ = "ingrese el valor de b: ";
4 b = input(b_);
5 c_ = "ingrese el valor de c: ";
6 c = input(c_);
7
8 [x1, x2] = roots_(a, b, c);
9
10 disp("la raiz 1 es: ")
11 disp(x1)
12 disp("la raíz 2 es: ")
13 disp(x2)
14
15
```


¡Nos vemos en una próxima ocasión!