

Funciones

En el módulo anterior aprendimos a realizar un *script*.


Ejecutar un *script* es totalmente equivalente a ejecutar desde la línea de comandos cada una de las líneas del *script*

Todas las variables que se creen dentro del *script* permanecen en el espacio de trabajo. Son variables globales.


Disponemos de un tipo de programa diferente a los *scripts*, que se denomina función (*function*).

Las funciones se caracterizan por admitir argumentos de entrada y salida y por disponer de su propio espacio de trabajo. Variables locales

Lo primero que vamos hacer es transformar un *script* a una función

```
graficar.m   
1 a=1;  
2 b=10;  
3 h=0.1;  
4 x=a:h:b;  
5 y=cos(x);  
6 plot(x,y)
```



```
graficar.m   
1 function salida=graficar(a,b,h)  
2 %esta funcion grafica el coseno entre a y b  
3 %con paso h.  
4 %salida=graficar(a,b,h)  
5  
6 %a=1;  
7 %b=10;  
8 %h=0.1;  
9 x=a:h:b;  
10 y=cos(x);  
11 salida=y;  
12 plot(x,y)  
13 end
```

```
>>graficar
```

```
>>valory=graficar(1,10,0.1)
```

graficar.m 


```
1  function salida=graficar(a,b,h)
```

Variable de salida

Nombre de la función.

Es totalmente recomendable que coincida con el nombre del archivo

Variable de entrada

graficar.m 

```
1 function salida=graficar(a,b,h)
2     %esta funcion grafica el coseno entre a y b
3     %con paso h.
4     %salida=graficar(a,b,h)
5
6     %a=1;
7     %b=10;
8     %h=0.1;
9     x=a:h:b;
10    y=cos(x);
11    salida=y;
12    plot(x,y)
13 end
```

comentarios

comentarios

Variables locales

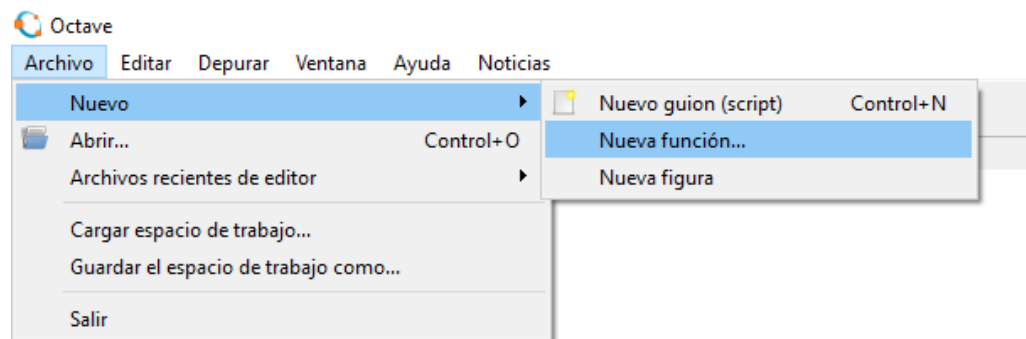
```
>> help graficar
```

```
'graficar' is a function from the file C:\Users'
```

```
esta funcion grafica el coseno entre a y b  
con paso h.
```

```
salida=graficar(a,b,h)
```

Para generar una función lo más común es utilizar directamente el editor de Octave



 Nueva función ? X

Nombre para la nueva función:

Aceptar

Cancelar

nuevafuncion.m

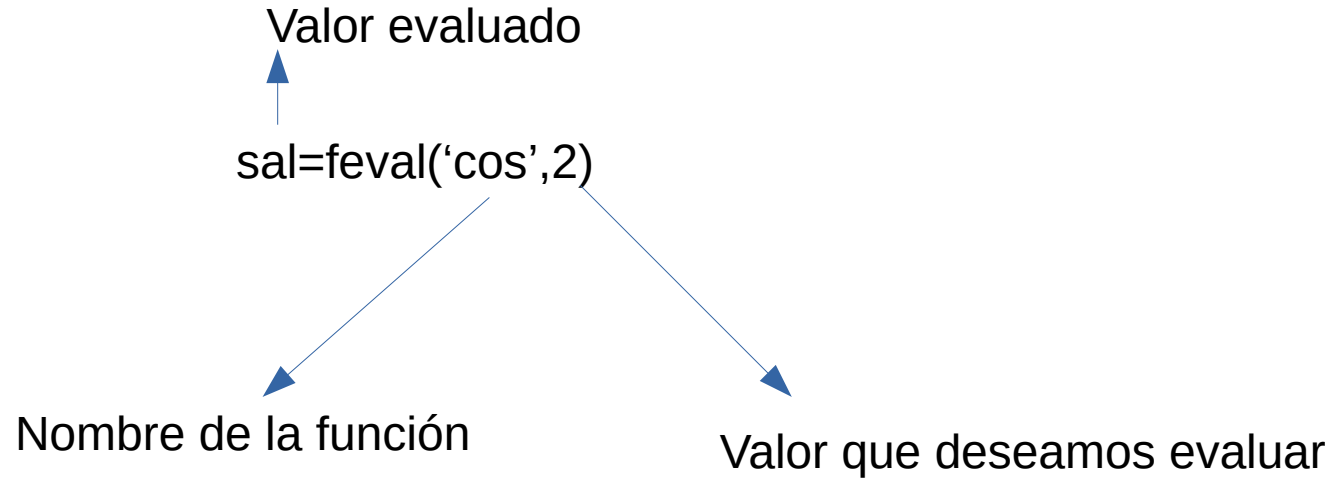
```
1  ## Copyright (C) 2020 DFR
2  ##
3  ## This program is free software: you can redistribute it and/or modify it
4  ## under the terms of the GNU General Public License as published by
5  ## the Free Software Foundation, either version 3 of the License, or
6  ## (at your option) any later version.
7  ##
8  ## This program is distributed in the hope that it will be useful, but
9  ## WITHOUT ANY WARRANTY; without even the implied warranty of
10 ## MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
11 ## GNU General Public License for more details.
12 ##
13 ## You should have received a copy of the GNU General Public License
14 ## along with this program. If not, see
15 ## <https://www.gnu.org/licenses/>.
16
17 ## -*- texinfo -*-
18 ## @deftypefn {} {@var{retval} =} nuevafuncion (@var{input1}, @var{input2})
19 ##
20 ## @seealso{}
21 ## @end deftypefn
22
23 ## Author: DFR <DFR@DESKTOP-4K4QGO4>
24 ## Created: 2020-08-18
25
26 function retval = nuevafuncion (input1, input2)
27
28 endfunction
29
```

▶ end

Como ingresar una función como parámetro de entrada a una función creada por nosotros

```
>>sal=feval('cos',2)
```

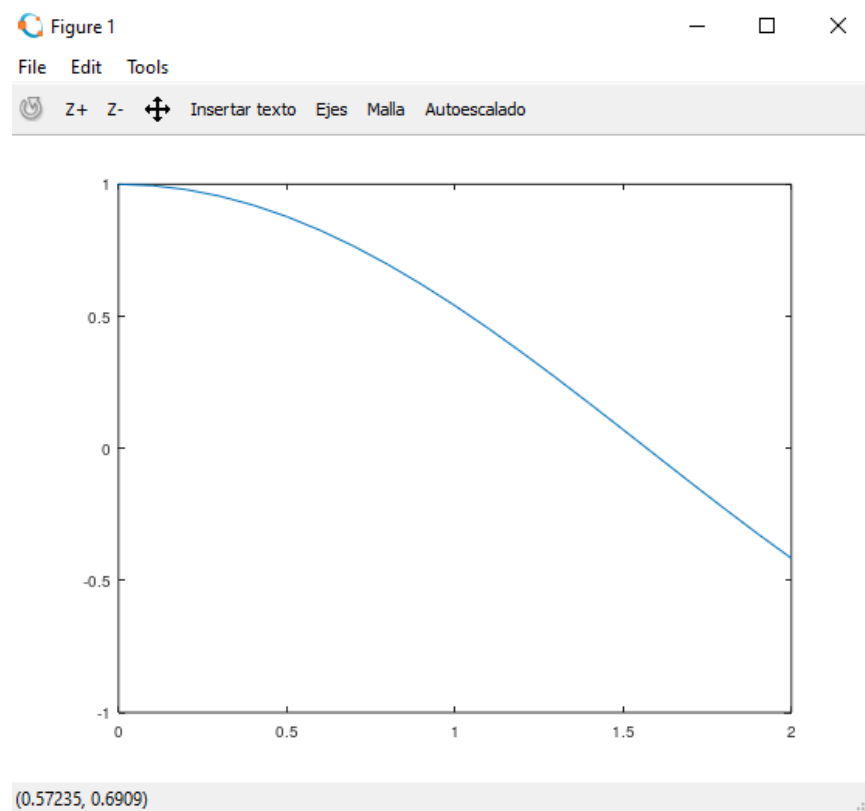
```
sal= -0.41615
```



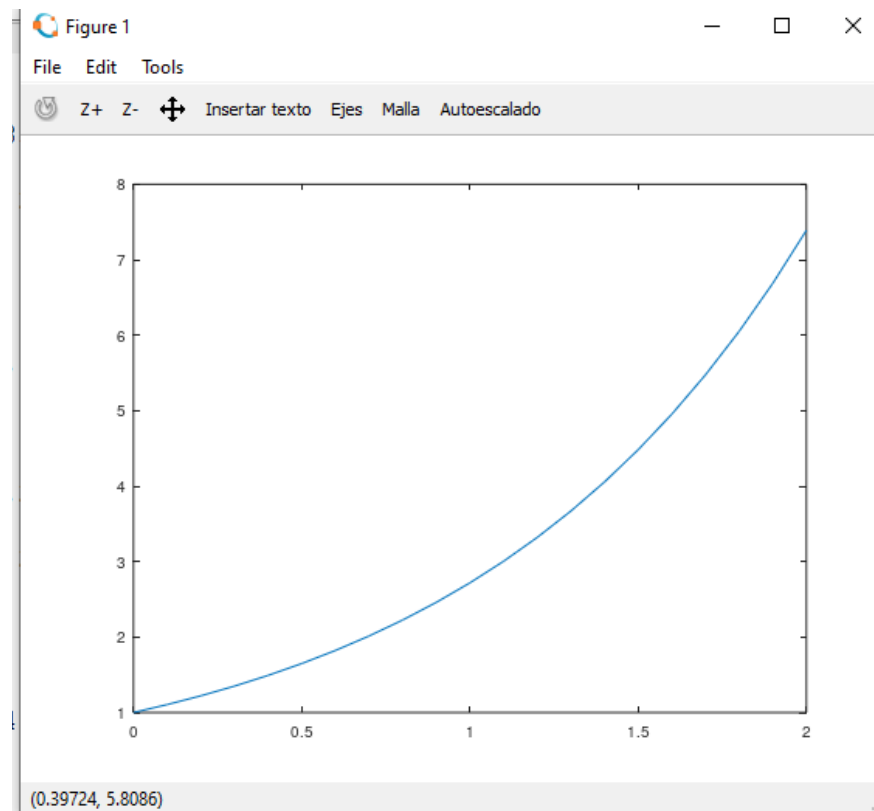
graficar.m

```
1 function salida=graficar(f,a,b,h)
2     %esta funcion grafica la funcion f entre a y b
3     %con paso h.
4     %salida=graficar(f,a,b,h)
5     %salida: vector de los valores evaluados entre ay b con paso h
6
7     x=a:h:b;
8     y=feval(f,x);
9     salida=y;
10    plot(x,y)
11 end
```

```
>>salida=graficar('cos',0,2,0.1)
```



```
>>salida=graficar('exp',0,2,0.1)
```



Actividad 1

Construir una función que tenga como nombre (areatriangulo) y que tenga como parámetros de entrada la longitud de sus tres lados (a,b,c) y como parámetro de salida el área del triángulo:

$$\text{área del triangulo} = \sqrt{s(s-a)*(s-b)*(s-c)}$$

$$s = \frac{(a+b+c)}{2}$$

Actividad 2

Construir una función que tenga como nombre (convertirFaC) y que tenga como parámetro de entrada la temperatura en grados Fahrenheit (T_f) y como parámetro de salida entrega la temperatura en grados Celsius

$$T_C = \frac{(T_F - 32) * 50}{90}$$

Actividad 3

Construir una función que tenga como nombre (convertirvectorFaC) y que tenga como parámetro de entrada un vector con temperaturas en grados Fahrenheit (T_f) y como parámetro de salida entrega un vector con la temperatura en grados Celsius

$$T_C = \frac{(T_F - 32) * 50}{90}$$

Actividad 4

Construir una función que tenga como nombre (ajustelineal), que tenga como parámetro de entrada dos vectores (x) e (y) del mismo tamaño. Como parámetro de salida entregue un vector Y

$$\bar{x} = \frac{\sum_{i=1}^n x(i)}{n} \quad \bar{y} = \frac{\sum_{i=1}^n y(i)}{n} \quad \bar{xy} = \frac{\sum_{i=1}^n x(i)y(i)}{n} \quad \bar{x^2} = \frac{\sum_{i=1}^n x(i)x(i)}{n}$$

$$Y(i) = \bar{y} + \frac{\bar{xy} - \bar{x} \bar{y}}{\bar{x^2} - (\bar{x})^2} (x(i) - \bar{x})$$

Ayuda: Utilice las funciones *sum* y *length*

Actividad 5

Construir una función que tenga como nombre (*graficarajuste*), que tenga como parámetro de entrada dos vectores (x) e (y) del mismo tamaño. Como parámetro de salida entregue un vector Y (ver actividad 4).

También debe graficar y en función de x e Y en función de x en la misma figura.

Tener una leyenda que indique el gráfico que se corresponde a cada vector.

- * una grilla
- * un título la figura (“Ajuste lineal”)
- * indique sobre el eje x (“x”)
- * indique sobre el eje y (“y”)

Ayuda: Utilice la función *ajustelineal* obtenida en la actividad 4