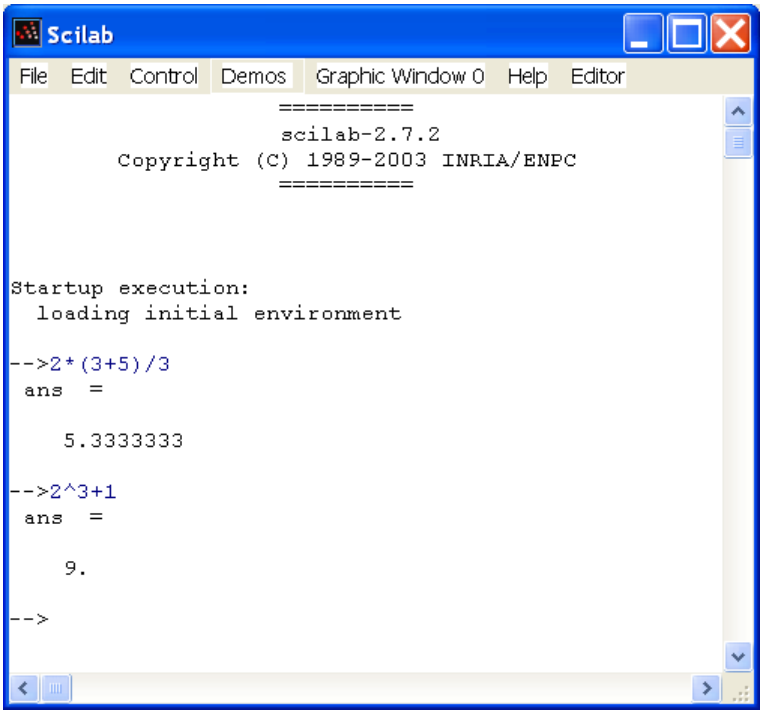
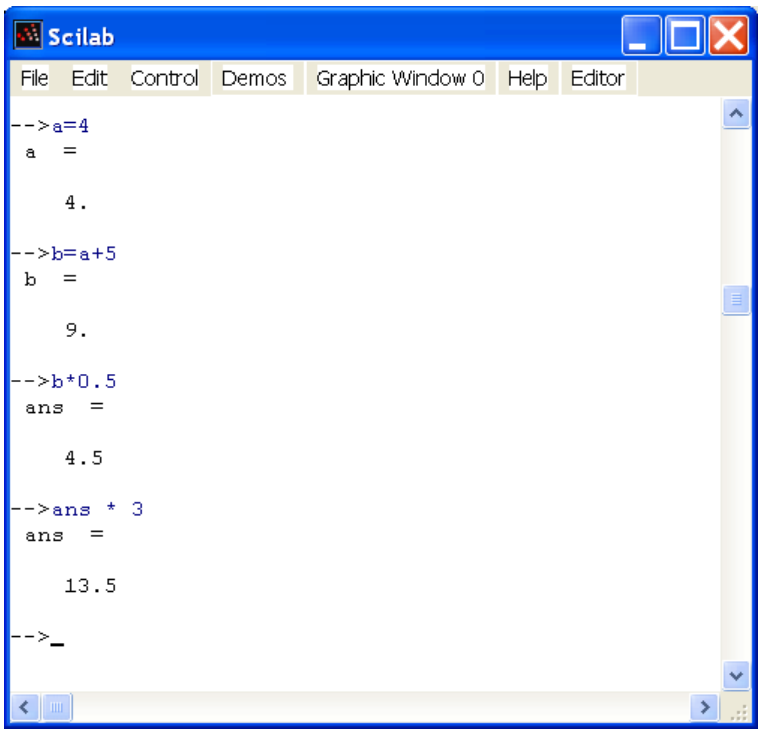
Introducción a Scilab

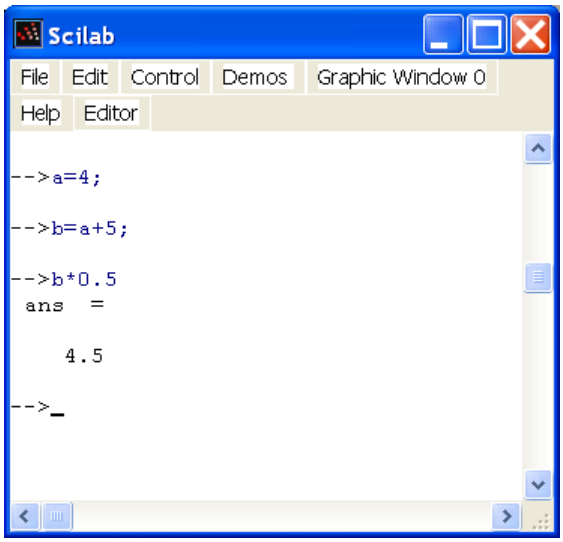
1. Scilab es básicamente una calculadora programable que podemos utilizar para realizar cálculos sencillos. Las operaciones básicas realizables son: suma (+), resta (-), producto (*), división (/) y potenciación (^):



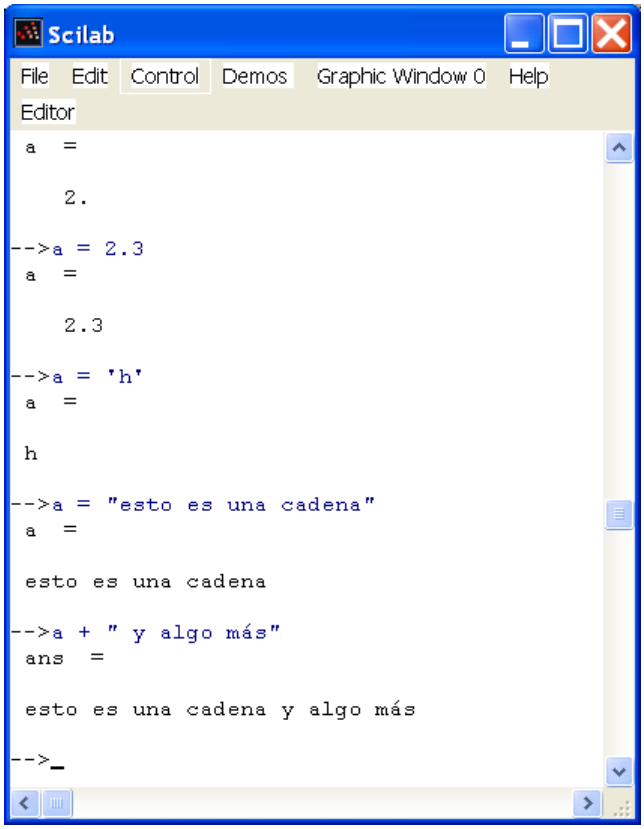
2. Podemos hacer uso de las variables. La asignación de valores a variables se realiza mediante el símbolo "=". Una vez se le haya asignado un valor a una variable, podemos utilizarla en expresiones posteriores. La variable *ans* contiene el valor de la última expresión evaluada:



3. La evaluación de una expresión genera un *eco*. Dicho eco puede eliminarse si la expresión finaliza con ";". Cuando programemos los algoritmos para Scilab, de no eliminar dicho eco, aparecerán por pantalla todos los cálculos intermedios que realiza nuestro algoritmo.



4. Las variables de Scilab no llevan asociado un único *tipo de dato*; pueden representar enteros, reales, complejos, caracteres y hasta cadenas (para el caso de cadenas las operaciones no tienen sentido, salvo la suma, que representa la concatenación):

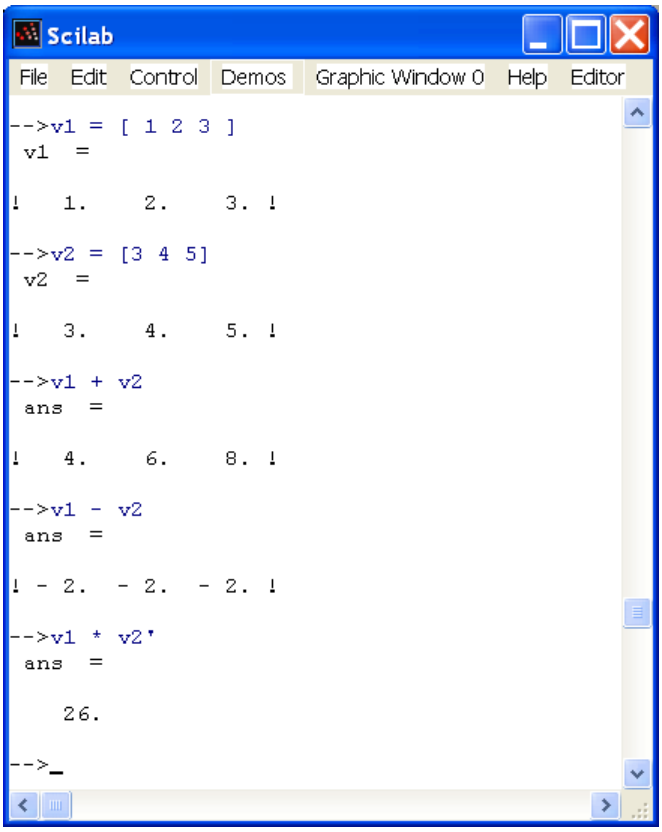


5. A pesar de que hasta ahora las variables han representado valores escalares, Scilab trata a las variables como matrices de dimensiones *n x m*. Las variables

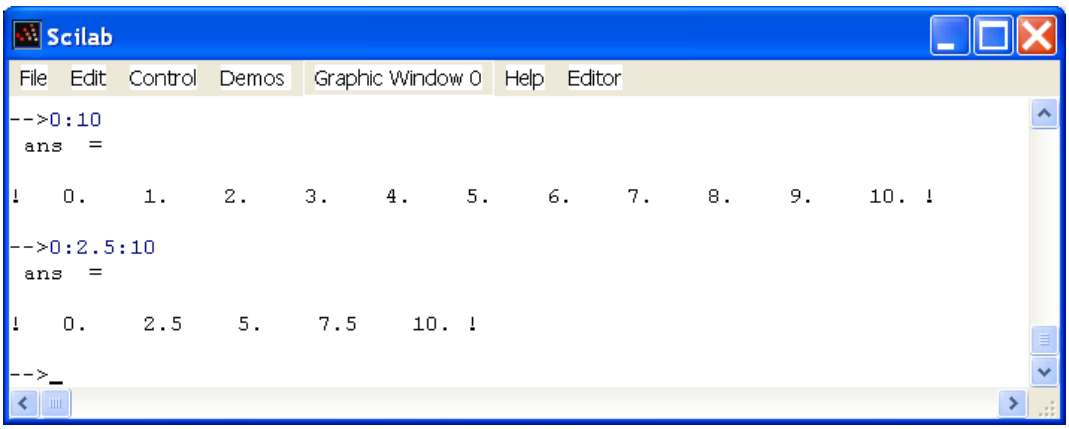
hasta ahora utilizadas equivalen a matrices de dimensión *1 x 1*. Una matriz se representa mediante la expresión:

$$[\ a_{11}\ a_{12}\ a_{13}\ \ldots\ a_{1m}\ ;\ a_{21}\ a_{22}\ a_{23}\ \ldots\ a_{2m}\ ;\ \ldots\ ;\ a_{n1}\ a_{n2}\ a_{n3}\ \ldots\ a_{nm}\ ]$$

Todas las operaciones anteriormente nombradas son aplicables a las matrices, eso si, teniendo en cuenta que estamos hablando de operaciones matriciales[1]. Además de los operadores previamente mencionados, existe el operador transposición ('):

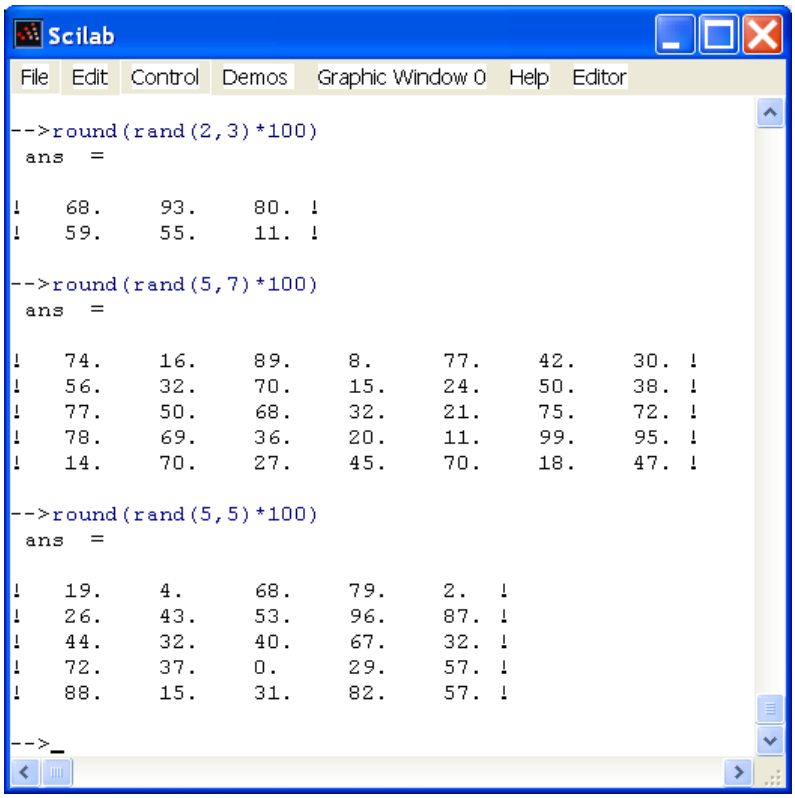


6. El operador ":"permite generar vectores cuyos valores forman una secuencia ascendente o descendente. La expresión *inicio:final* genera los números de *inicio* a *final* con intervalo *1*. Es posible también generar una secuencia con un intervalo diferente mediante la expresión *inicio:intervalo:final*.



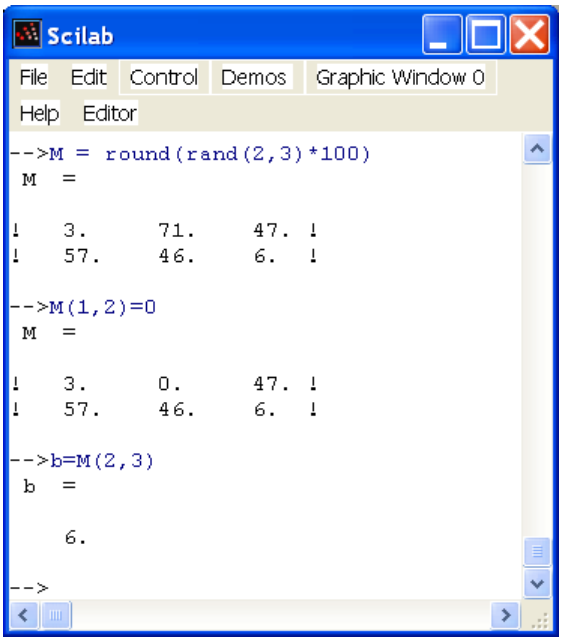
[1] Existen tres operadores adicionales: ".*" "./" y ".^". Estos operadores realizan la operación *"elemento a elemento"* en vez de de forma matricial.

7. Dado que en adelante trataremos operaciones sobre matrices, y para no perder mucho tiempo en la generación de las mismas, veremos una curiosa forma de obtener matrices con valores enteros aleatorios. La función *rand(n,m)* genera matrices aleatorias de dimensión *n x m* con valores aleatorios en el rango *[0 1]*, mientras que la función *round(x)* realiza el redondeo al valor entero más cercano. Podemos ya generar matrices de enteros automáticamente de la siguiente forma:

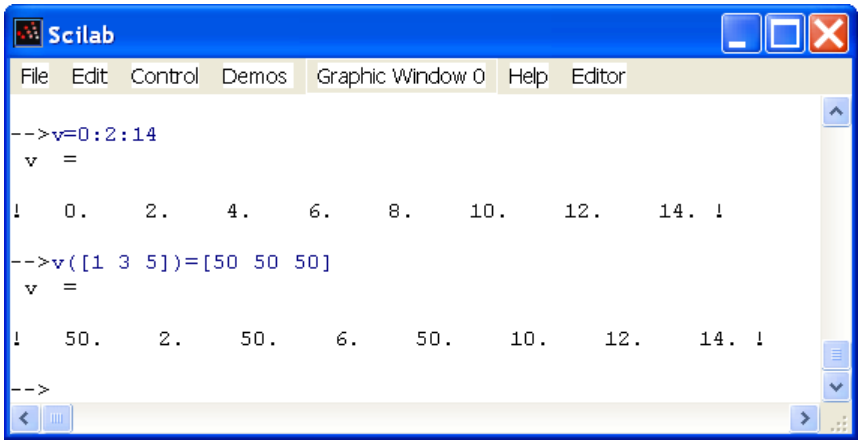


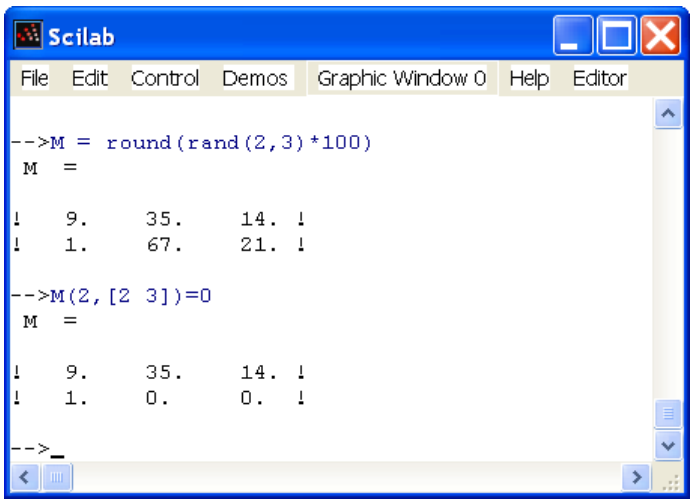
8. Es posible acceder a los elementos de una matriz tanto para usarlos como para modificarlos, para lo cual usaremos la expresión:

Nombre_vector(i) o Nombre_matriz(i,j)

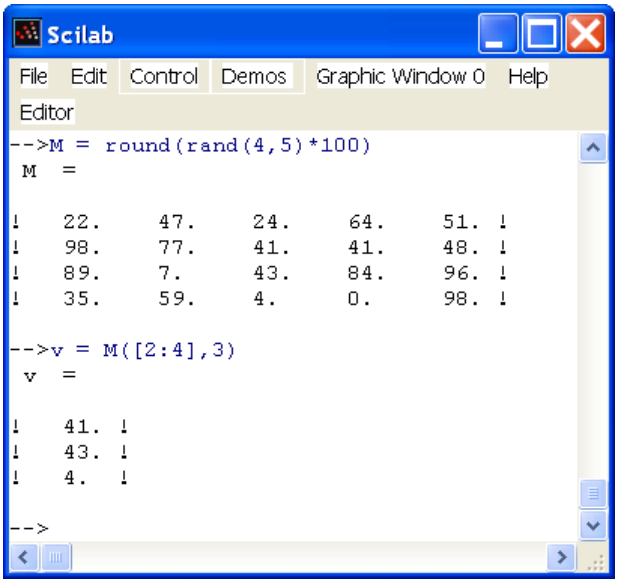


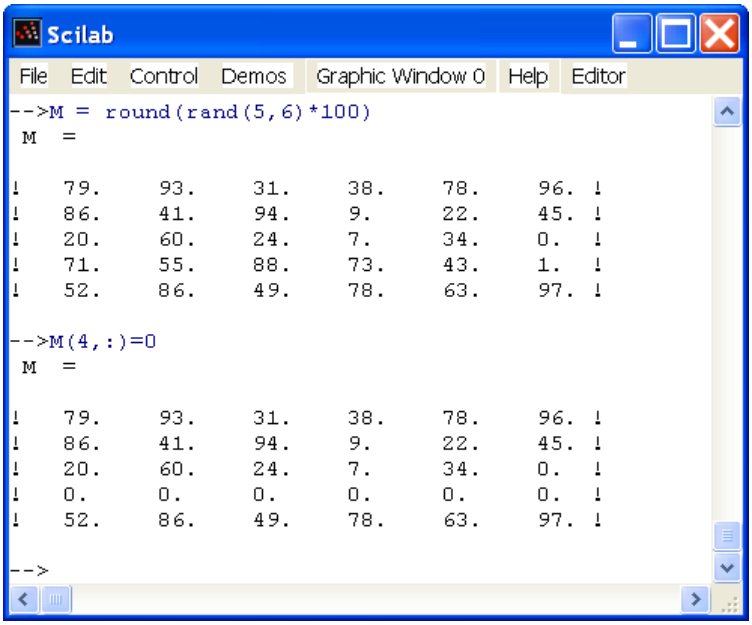
9. No estamos limitados a acceder a un solo elemento de una matriz; si reemplazamos los índices por vectores de índices, obtendremos un conjunto de elementos de la matriz:



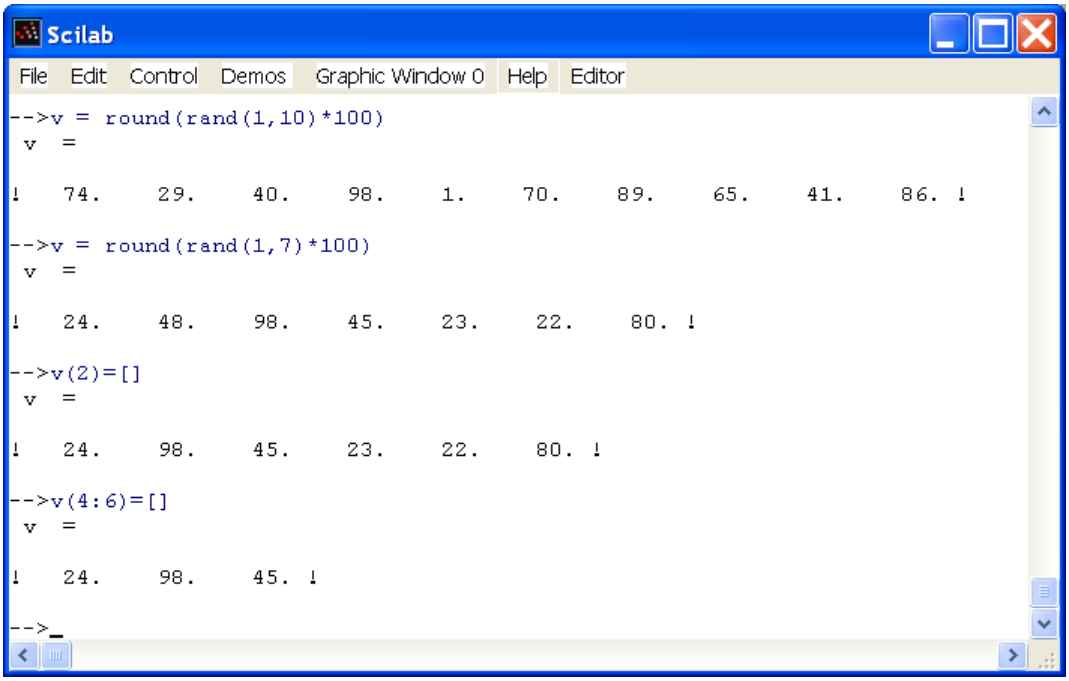


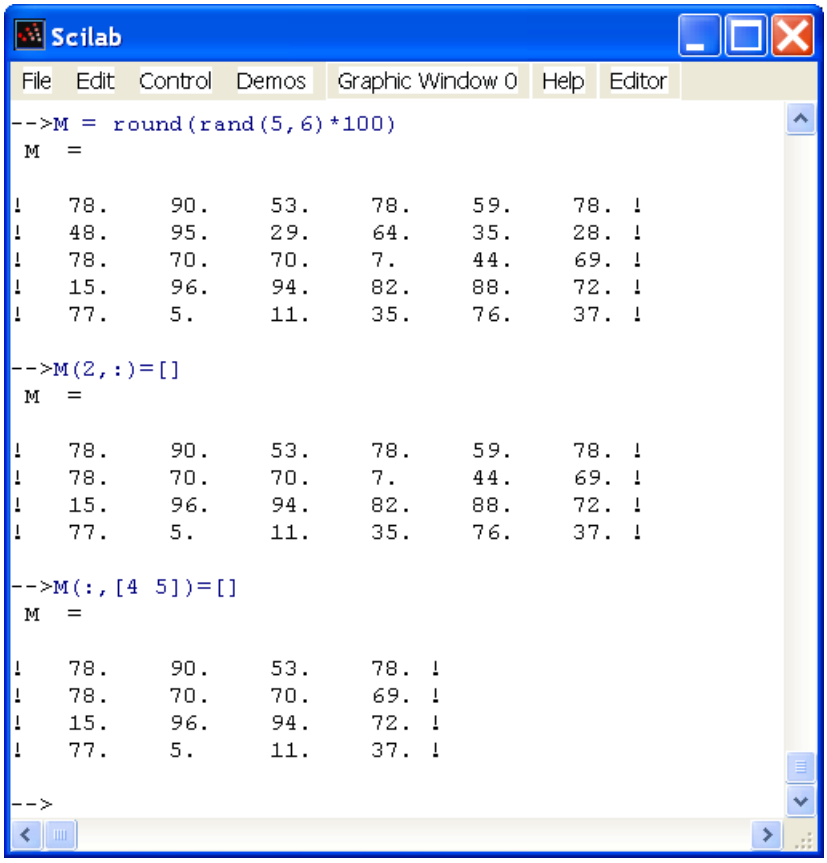
10. El uso del operador "*:*" es muy adecuado cuando se desea acceder a valores contiguos de una matriz, ya que no será necesario indicar todos los índices. De hecho, cuando no se indica *inicio* y *final*, dicho operador genera todo el rango posible de índices:



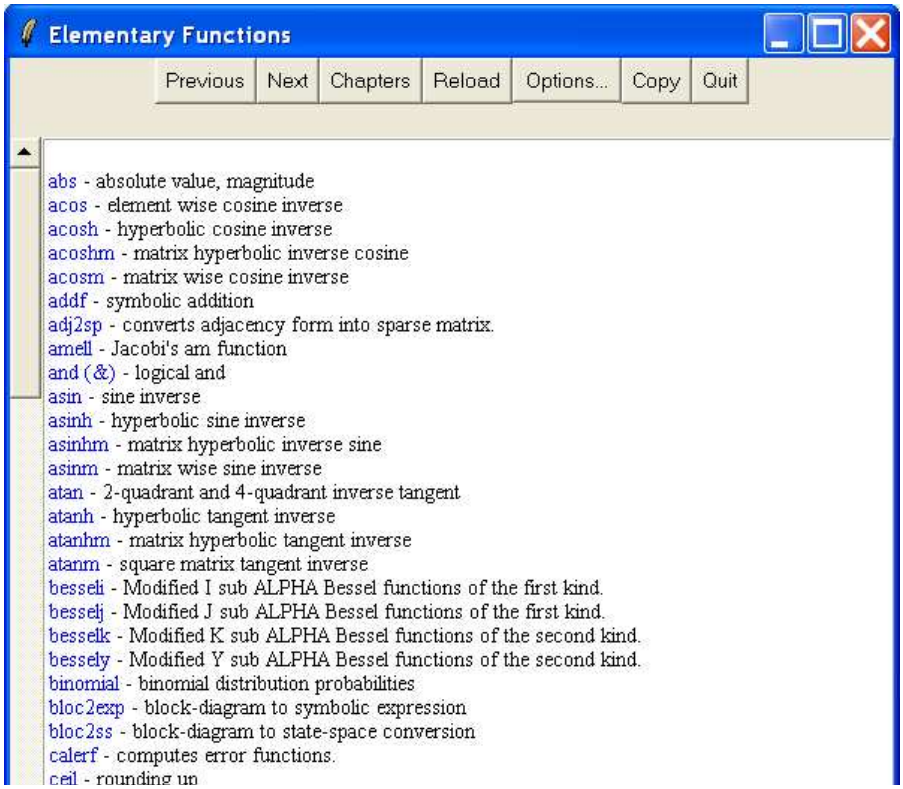


11. Existe también el *conjunto vacío*, representado por el símbolo *[ ]*. Cuando a un valor (o conjunto de valores) de un vector se le asigna el *conjunto vacío*, dicho elemento es eliminado del vector. De forma similar, cuando a una fila o columna (o conjunto de filas o columnas) de una matriz se le asigna el *conjunto vacío*, la fila o columna es eliminada:

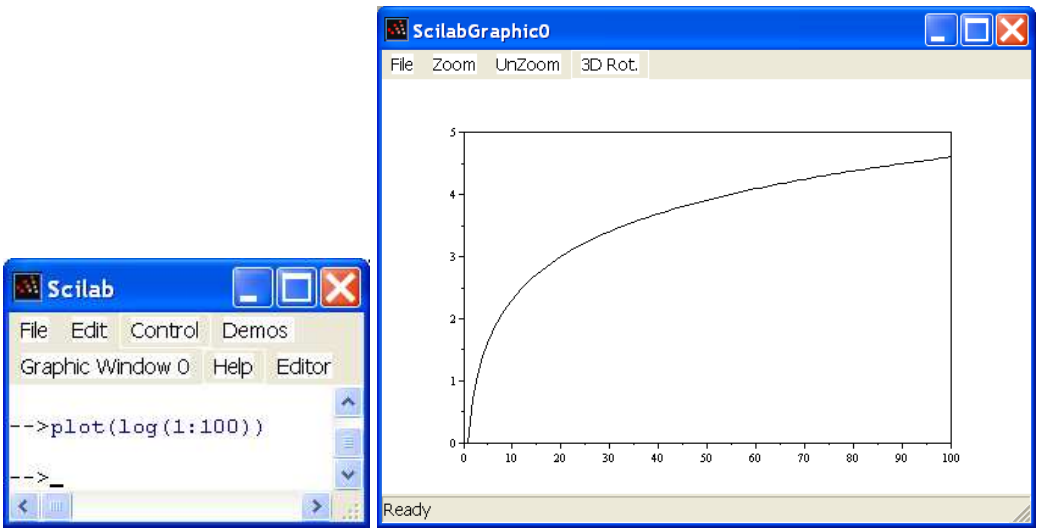




12. Existen infinidad de funciones matemáticas que podemos usar en nuestras expresiones. Podemos obtener un listado de las mismas pinchando sobre el botón de menú *help* y seleccionando *Elementary Functions*:



13. Para finalizar, estudiaremos brevemente una de las funciones de la librería gráfica, la función *plot()*. Esta función nos permite representar vectores y matrices de una forma gráfica. *plot()* no puede representar de una forma simbólica funciones, no obstante podemos calcular un vector que sea representativo del rango de la función que deseemos representar. Si la función es llamada con un solo argumento, dicho argumento es dibujado en el *eje y*, mientras que si es llamada con dos argumentos, toma el primero como *eje x* y el segundo como *eje y*:



Como práctica

1. Escribir el código en Scilab necesario para generar una gráfica de la función *seno(x)* en el intervalo *[0,2π]* (el valor π puede obtenerse mediante la expresión *%pi*). Asegurarse de que la escala del *eje x* es la correcta.