

5.4. Matrices

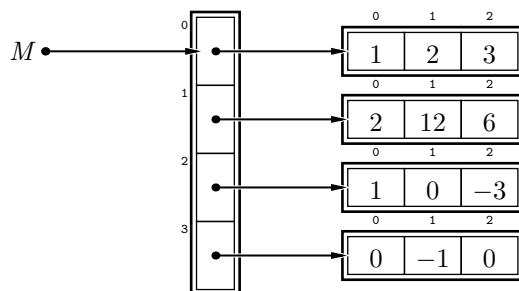
Las matrices son disposiciones bidimensionales de valores. En notación matemática, una matriz se denota encerrando entre paréntesis los valores, que se disponen en filas y columnas. He aquí una matriz M :

$$M = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 12 & 6 \\ 1 & 0 & -3 \\ 0 & -1 & 0 \end{pmatrix}$$

Esta matriz tiene 4 filas y 3 columnas, lo cual abreviamos diciendo que es una matriz de dimensión 4×3 .

Las listas permiten representar series de datos en una sola dimensión. Con una lista de números no se puede representar directamente una matriz, pero sí con una *lista de listas*.

```
>>> M = [ [1, 2, 3], [2, 12, 6], [1, 0, -3], [0, -1, 0] ] ↵
```



En la notación matemática el elemento que ocupa la fila i -ésima y la columna j -ésima de una matriz M se representa con $M_{i,j}$. Por ejemplo, el elemento de una matriz que ocupa la celda de la fila 1 y la columna 2 se denota con $M_{1,2}$. Pero si deseamos acceder a ese elemento en la matriz Python M , hemos de tener en cuenta que Python siempre cuenta desde cero, así que la fila tendrá índice 0 y la columna tendrá índice 1:

```
>>> M[0][1] ↵  
2
```

Observa que utilizamos una doble indexación para acceder a elementos de la matriz. ¿Por qué? El primer índice aplicado sobre M devuelve un componente de M , que es una lista:

```
>>> M[0] ↵  
[1, 2, 3]
```

Y el segundo índice accede a un elemento de esa lista, que es un entero:

```
>>> M[0][0] ↵  
1
```

..... EJERCICIOS

► 244 Una matriz nula es aquella que sólo contiene ceros. Construye una matriz nula de 5 filas y 5 columnas.

► 245 Una matriz identidad es aquella cuyos elementos en la diagonal principal, es decir, accesibles con una expresión de la forma $M[i][i]$, valen uno y el resto valen cero. Construye una matriz identidad de 4 filas y 4 columnas.

► 246 ¿Qué resulta de ejecutar este programa?

```
1 M = [ [1, 0, 0], [0, 1, 0], [0, 0, 1] ]  
2 print M[-1][0]  
3 print M[-1][-1]  
4 print '---'  
5 for i in range(0, 3):  
6     print M[i]  
7 print '---'  
8 for i in range(0, 3):  
9     for j in range(0, 3):  
10        print M[i][j]
```

► 247 ¿Qué resulta de ejecutar este programa?

```
1 M = [ [1, 0, 0], [0, 1, 0], [0, 0, 1] ]  
2 s = 0.0  
3 for i in range(0, 3):  
4     for j in range(0, 3):  
5         s += M[i][j]  
6 print s / 9.0
```