

# NUMPY

Sea “a” y “b” dos arreglos numpy, sea “x” un elemento cualquiera, y sea “n” y “m” dos numeros enteros positivos cualesquiera

ACCION	NOMBRE
Crear un arreglo numpy	<code>a = np.array([“arreglo”], “tipo_de_datos”)</code> ej: <code>np.array([1,2,3,4,5], float)</code>
Crea un arreglo numpy lleno de ceros, de dimensiones m x n.	<code>np.zeros(m,n)</code>
Crea un arreglo numpy lleno de unos, de dimensiones m x n.	<code>np.ones(m,n)</code>
Crea un arreglo numpy lleno de ceros, con las mismas dimensiones del arreglo “a”	<code>np.zeros_like(a)</code>
Crea un arreglo numpy identidad (matriz identidad) de dimensión n.	<code>np.identity(n, dtype = “tipo_de_datos”)</code>
Devuelve el valor más pequeño de un arreglo numpy.	<code>np.min(a)</code>
Devuelve el valor más grande de un arreglo numpy.	<code>np.max(a)</code>
Conseguir las dimensiones del arreglo “a” (devuelve una tupla de dos elementos).	<code>a.shape</code>
Devuelve una copia del arreglo en un espacio de memoria diferente.	<code>a.copy()</code>
Llena todas las posiciones del arreglo con el elemento “x”.	<code>a.fill(x)</code>
Llevar un arreglo (de ser posible) a diferentes dimensiones n x m	<code>a.resize((n,m))</code> <code>a.reshape((n,m))</code>
Multiplicación de matricial.	<code>np.dot(a,b)</code>
Media aritmética de los valores de una matriz.	<code>np.mean(a)</code>
Conseguir una fila de un arreglo. Conseguir una columna de un arreglo.	<code>a[fila,:]</code> <code>a[:,columna]</code>
Transformar un arreglo numpy en una lista.	<code>a.tolist()</code>
Devuelve la diagonal de un arreglo numpy.	<code>np.diagonal(a)</code>
Devuelve el determinante de la matriz “a”.	<code>np.linalg.det(a)</code>
Devuelve la matriz inversa a “a”	<code>np.linalg.inv(a)</code>
Devuelve la matriz transpuesta de “a”	<code>np.transpose(a)</code>
Devuelve la matriz triangular inferior de “a” Devuelve la matriz triangulas superior de “a”	<code>np.tril(a)</code> <code>np.triu(a)</code>
Devuelve posición del valor mínimo de a Devuelve la posición del valor máximo de a	<code>a.argmin()</code> <code>a.argmax()</code>
Devuelve la suma de los elementos de a Devuelve una lista con la suma de los elementos de las columnas de “a” Devuelve una lista con la suma de los elementos de las filas de “a”	<code>np.sum(a)</code> <code>np.sum(axis=0)</code> <code>np.sum(axis=1)</code>
Función booleana que devuelve True si TODOS los elementos de “a” son verdadero (>=1)	<code>a.all()</code>
Función booleana que devuelve True si al menos un elemento de “a” es verdadero (>=1)	<code>a.any()</code>

## COLECCIONES

*Sea "l" una lista cualquiera, "dict" un diccionario cualquiera, "c" un set(conjunto) cualquiera, "cadena" un string cualquiera y "x" un elemento cualquiera*

ACCION	NOMBRE
Agregar un elemento a una lista	<code>l.append(x)</code>
Agregar un elemento a un conjunto	<code>c.add(x)</code>
Ordenar elementos de una lista en orden ascendente (de menor a mayor) $10 > 0$ ; $A > Z$	<code>l.sort()</code>
Contar cuantos elementos "x" hay en una lista. Devuelve un numero entero	<code>l.count()</code>
Devuelve la primera posición de la lista donde se ubica "x", si no se encuentra el elemento devuelve -1	<code>l.index(x)</code>
Verifica si "x" se encuentra en un conjunto "c"	<code>c.contains(x)</code>
Da la vuelta a una lista (es decir el ultimo para a ser el primero, el penúltimo el segundo, etc)	<code>l.reverse()</code>
Remover un valor "x" de una lista (en caso de "x" aparecer varias veces, se remueve solo UNO)	<code>l.remove(x)</code> <code>l.pop(x) ← devuelve "x"</code>
Remover un elemeno de un conjunto	<code>c.remove(x)</code>
Mínimo de una lista. Máximo de una lista. Sumatoria de los elementos una lista.	<code>min(l)</code> <code>max(l)</code> <code>sum(l)</code>
Devuelve la longitud de la lista	<code>len(l)</code>
Devuelve la longitud de un conjunto	<code>c.size()</code>
Verifica si un conjunto esta vacío	<code>c.isEmpty()</code>
Elimina todos los elementos de una lista	<code>l.clear()</code>
Elimina todos los elementos de un conjunto	<code>c.clear()</code>
Transformar una lista en un conjunto. Devuelve el nuevo conjunto.	<code>set(l)</code>
Obtener una lista de las claves de un diccionario.	<code>dict.keys()</code>
Obtener una lista de los valores de un diccionario.	<code>dict.values()</code>
Recorrer una lista.	<code>for elemento in l:</code>
Recorrer un set.	<code>for elemento in c:</code>
Recorrer un diccionario -> En cada iteración "k" toma el valor de una clave, y "v" toma el valor asociado a esa clave.	<code>for k,v in dict.items():</code>
Recorrer un String.	<code>for letra in cadena:</code>
Convertir un String todo a minúsculas	<code>cadena.lower()</code>
Convertir un String todo a mayúscula	<code>cadena.upper()</code>
Reemplaza un carácter "x" por el carácter "y" en un String.	<code>cadena.replace(x,y)</code>
Elimina los espacios o saltos de línea al final o al inicio de una cadena.	<code>cadena.strip()</code>
Encontrar el índice de la primera aparición de "x". En caso de no ser encontrado retorna -1	<code>cadena.find(x)</code>
Determina si toda la cadena son caracteres. Determina si toda la cadena son dígitos. Determina si toda la cadena son espacios. Determina si toda la cadena son signos de puntuación.	<code>cadena.isalpha()</code> (devuelven True o False) <code>cadena.isdigit()</code> ... <code>cadena.isspace()</code> ... <code>cadena.ispunctuation()</code> ...

Retorna una lista con elementos que son subcadenas de "cadena" separados por "x"	<code>cadena.split("x")</code>
Une los elementos de una lista convirtiéndolos en una cadena, utilizando "x" entre cada elemento	<code>"x".join(l)</code>
Carácter de escape:	<code>\</code>
Caracteres Especiales (algunos):	<code>\\</code> <code>backslash</code> <code>\'</code> <code>comilla simple</code> <code>\''</code> <code>comillas dobles</code> <code>\b</code> <code>backspace</code> <code>\n</code> <code>enter (salto de línea)</code>
Slicing de Strings	<code>subcadena = cadena[comienzo:final:salto]</code> <b>comienzo:</b> índice del primer carácter <b>final:</b> índice del carácter posterior al ultimo <b>salto (OPCIONAL):</b> cuantos caracteres saltar cada vez, hasta llegar al final