**Usos de numpy**

**a2 = np.arange(10) *# np.arange es un metodo similar al range del tipo list***

**a3 = np.zeros((2,3)) *# crear un lista de dos dimensiones, pre-rellenada con zeros***

**a4 = np.ones((2,3)) *# crear un lista de dos dimensiones, pre-rellenada con unos***

**np.linspace(a,b,n) Es una función que permite crear arrays de una dimensión, de largo n, y que contienen puntos entre a y b, distanciados de forma regular. La distancia entre cada punto será de .(b-a)/(n-1)**

**a5 = np.linspace(0,1,11)**

**display(a5)**

**array([ 0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ])**

**Notita:**

display( ) lo que nos retorna los números y el tipo.

**Arrays es numpy**

“ **np.array\_equal(a, b )**  ” compara y retorna si 2 arreglos (a, b) son iguales.

La instrucción “**.shape**” nos retorna la forma del dato : (1 “fila”, 2”columnas)

La instrucción “**.reshape(a, b)** ” modifica el array a la dimensión (**a, b).**

“ *la* ***condición*** *es que la multiplicación de la nueva dimensión sea i****gual al número de elementos*** *que tenía el array que será modificado. “*

**SLICING**

**a = np.arange(10)**

**b = np.eye(3)**

**display(a);display(b)**

**array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])**

**array([ [ 1., 0., 0. ],**

**[ 0., 1., 0. ],**

**[ 0., 0., 1. ] ])**

**a[:5]**

**array([0, 1, 2, 3, 4])**

**b[0:3,1] # Los 3 elementos de la segunda columna**

**array([ 0., 1., 0.])**

**Operaciones sobre Arrays**

La operación “%” nos retorna el residuo de la división. “ a%b”

**Ufunctions**

**a = np.arange(4)**

**b = np.arange(1,5)**

**display(np.exp(a)) # exponencial**

**display(np.log(b)) # logaritmo natural**

**display(np.sqrt(a)) # raiz cuadrada**

**display(np.greater(a,b)) # superior o igual punto a punto: hace una**

**comparación de cada elemento entre los 2**

**arrays**

**array( [ 1. , 2.71828183, 7.3890561 , 20.08553692] )**

**array( [ 0. , 0.69314718, 1.09861229, 1.38629436] )**

**array( [ 0. , 1. , 1.41421356, 1.73205081] )**

**array( [False, False, False, False], dtype=bool )**

**Estadística y aleatoriedad**

np.random.random( 10 ) # nos retorna un array con 10 números aleatorios entre 0 y 1