## Practica 0: python, numpy, matplot

		Imprimir "hola mundo"
In	[181]:	
		Hello world
		Imprimir "hola mundo" diez veces
In	[182]:	
		Hello world
		Hacer una funcion que imprima un mensaje, n veces. Invocar la funcion.
In	[183]:	
		Hello World Hello World Hello World
		Crear una lista con los diez primeros numeros pares
In	[184]:	
		[0, 2, 4, 6, 8, 10, 12, 14, 16, 18]
		Imprimir la longitud de la lista
In	[185]:	
		10

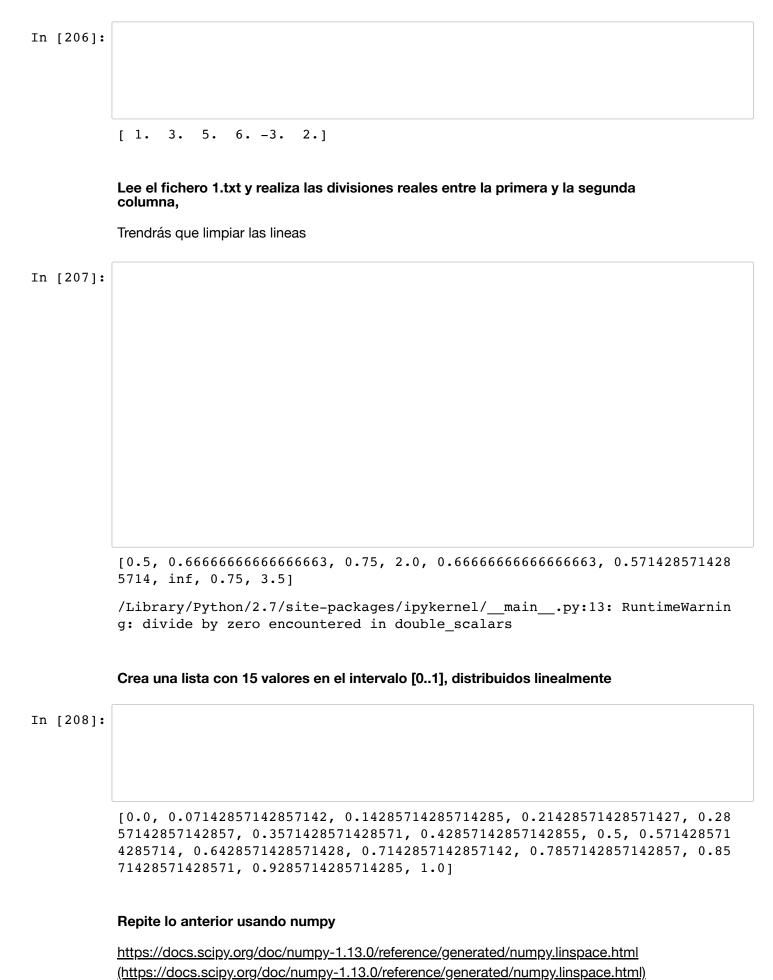
## Imprimir el ultimo elemento de la lista

```
In [186]:
Out[186]: 18
           Imprimir los tres últimos elementos de la lista
In [187]:
Out[187]: [14, 16, 18]
           Imprimir los tres primeros elementos de la lista
In [188]:
Out[188]: [0, 2, 4]
           Imprimir los cuatro elementos centrales de la lista
In [189]:
           [6, 8, 10, 12]
           Convertir la lista anterior a floats
In [190]:
           [0.0, 2.0, 4.0, 6.0, 8.0, 10.0, 12.0, 14.0, 16.0, 18.0]
           Desordena los elementos (usa shuffle de la librería random)
In [228]:
               2.
                                          14.
                                                 6. 12. 10.
                   18.
                         16.
                                0.
                                                                  4.]
           Vuelve a ordenarios
In [193]:
           [0.0, 2.0, 4.0, 6.0, 8.0, 10.0, 12.0, 14.0, 16.0, 18.0]
```

Sumar todos los elementos de la lista

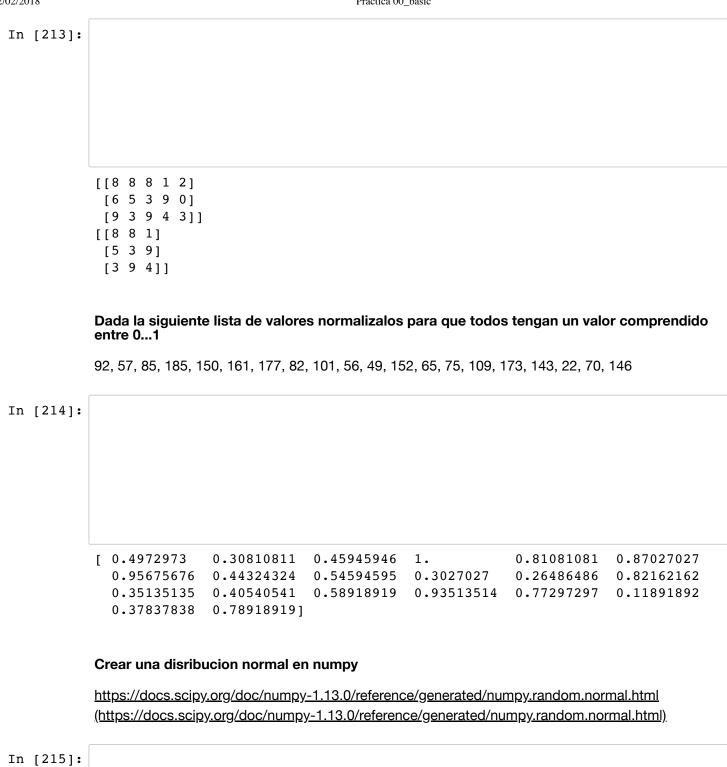
In	[194]:	
		90.0
		Utilizando numpy, repite las operaciones (generar lista, convertir a float y sumar los valores)
		https://s3.amazonaws.com/assets.datacamp.com/blog_assets/Numpy_Python_Cheat_Sheet.pdf (https://s3.amazonaws.com/assets.datacamp.com/blog_assets/Numpy_Python_Cheat_Sheet.pdf)
In	[195]:	
In	[196]:	
		[ 0 2 4 6 8 10 12 14 16 18] [ 0. 2. 4. 6. 8. 10. 12. 14. 16. 18.] 90.0
		Crear una lista de 20 enteros aleatorios entre 0 y 100
In	[197]:	
In	[198]:	
		[73, 86, 53, 99, 53, 73, 84, 27, 56, 36, 99, 91, 1, 97, 28, 14, 3, 11, 8 5, 27]
		Encontrar el maximo valor y el indice en el que esta
In	[199]:	
		99

In [200]:	
	99 3 99
	Hacer lo mismo con numpy
In [201]:	
	99 5
	Reemplaza todas las 'a's por 'e's en la siguiente frase:
	"Pepe Pecas pica papas con un pico, con un pico pica papas Pepe Pecas."
In [202]:	
Out[202]:	'Pepe Peces pice pepes con un pico, con un pico pice pepes Pepe Peces.'
	<b>-</b>
	Transforma la cadena a minuscula
In [203]:	
	Cuenta el numero de 'P's
In [204]:	
Out[204]:	14
040[201].	
	Parsea como enteros la siguiente cadena de caracteres:
	"1,3, 5, 6, -3,2,"
	.,-, -, -, -,-,
In [205]:	
	['1', '3', '5', '6', '-3', '2']



```
In [209]:
Out[209]: array([ 0.
                                   0.07142857,
                                                  0.14285714,
                                                                 0.21428571, 0.28571429,
                     0.35714286,
                                   0.42857143,
                                                                 0.57142857,
                                                                                0.64285714,
                                                  0.5
                     0.71428571,
                                    0.78571429,
                                                  0.85714286,
                                                                 0.92857143,
                                                                                1.
                                                                                            ])
           Crea dos np arrays con 20 numeros randoms y réstalas.
           Filtra el resultado eliminando todos los numeros menores de cero
In [210]:
            [0 0 0 1 3 0 6 7 5 5]
           Cea dos arrays aleatorias de 10 valores, concatenalas y selecciona los 10 valores
           centrales.
In [211]:
           Cea dos arrays aleatorias de 5 valores, apilalas en una matrix y selecciona la columna
           central.
           help(np.vstack)
In [212]:
            [7 5 8 0 0] [0 2 0 1 1]
            [[7 5 8 0 0]
             [0 2 0 1 1]]
            [8 0]
```

Crea tres arrays aleatorias de 5 valores, apilalas en una matrix y selecciona la submatrix 3x3 central.



Pintalos con matplotlib, recuerda usar "%matplotlib inline" para que jupyter lo pinte en el notebook.

Utiliza plt.hist

https://matplotlib.org/xkcd/users/pyplot\_tutorial.html (https://matplotlib.org/xkcd/users/pyplot\_tutorial.html)

```
In [216]:
```

```
2.,
                                                         15.,
                                                                        29.,
                                                                               30.,
Out[216]: (array([
                      1.,
                             2.,
                                            8.,
                                                   6.,
                                                                16.,
                     37.,
                                   71.,
                                         103.,
                                                         79.,
                                                                        77.,
                            52.,
                                                 101.,
                                                                85.,
                                                                               79.,
                     63.,
                            47.,
                                   28.,
                                          24.,
                                                  19.,
                                                         13.,
                                                                 6.,
                                                                         2.,
                                                                                1.,
                      2.,
                             0.,
                                    2.]),
           array([-0.62874274, -0.58504106, -0.54133937, -0.49763768, -0.453936
                   -0.41023431, -0.36653262, -0.32283094, -0.27912925, -0.23542756,
                   -0.19172588, -0.14802419, -0.10432251, -0.06062082, -0.01691913,
                    0.02678255, 0.07048424, 0.11418593, 0.15788761,
                                                                          0.2015893 ,
                    0.24529099,
                                 0.28899267,
                                               0.33269436,
                                                            0.37639605,
                                                                          0.42009773,
                    0.46379942,
                                 0.50750111,
                                               0.55120279,
                                                            0.59490448,
                                                                          0.63860617,
                    0.68230785]),
           <a list of 30 Patch objects>)
```

