1. ~~Generators~~
2. ~~Exceptions~~
3. ~~Numpy~~
4. Pandas
5. Plotting
6. Classes

~~sdfasdfasdf~~

asdaasdf

asdfasdf

Generators

In [6]:
```
import random
students = [
    'Tzwrtzina',
    'Andreas',
    'Xristos',
    'Andromaxh',
    'Danah',
    'Antwnia',
    'Aris',
    'Maria',
    'Sofia',
    'Iwanna',
    'Aggelos',
]

def random_student():
    return random.choice(students)

rs = random_student
```

In [9]:
```
def f():
    return 5
    return 6
    return "mitsos"
```

In [10]:
```
f()
```

Out[10]: 5

In [13]:
```
def gen():
    yield 5
    yield 6
    yield "Mitsos"
```

In [14]:
```
g = gen()
```

In [16]:
```
next(g)
```

Out[16]: 5

In [17]:
```
next(g)
```

Out[17]: 6

In [18]:
```
next(g)
```

Out[18]: 'Mitsos'

In [19]:
```
next(g)
```

```
---------------------------------------------------------------------
---------
StopIteration                             Traceback (most recent c
all last)
<ipython-input-19-e734f8aca5ac> in <module>()
----> 1 next(g)

StopIteration:
```

In [20]:
```
def gen_prime():

    n = 1
    while True:
        n += 1
        prime = True
        for i in range(2, n):
            if n%i == 0:
                prime = False
                break
        if prime:
            yield n
```

In [21]:
```
prime_generator = gen_prime()
```

In [22]:
```
next(prime_generator)
```

Out[22]: 2

In [23]:
```
next(prime_generator)
```

Out[23]: 3

In [24]:
```
next(prime_generator)
```

Out[24]: 5

In [25]:
```
next(prime_generator)
```

Out[25]: 7

In [26]:
```
next(prime_generator)
```

Out[26]: 11

In [27]:
```
next(prime_generator)
```

Out[27]: 13

In [28]:
```
for i in range(500):
    print (next(prime_generator))
```

```
17
19
23
29
31
37
41
43
47
53
59
61
67
71
73
79
83
89
97
101
103
107
109
113
127
131
137
139
149
151
157
163
167
173
179
181
191
193
197
199
211
223
227
229
233
239
241
251
257
263
269
271
277
281
283
293
307
311
313
```

```
317
331
337
347
349
353
359
367
373
379
383
389
397
401
409
419
421
431
433
439
443
449
457
461
463
467
479
487
491
499
503
509
521
523
541
547
557
563
569
571
577
587
593
599
601
607
613
617
619
631
641
643
647
653
659
661
673
677
683
691
```

```
701
709
719
727
733
739
743
751
757
761
769
773
787
797
809
811
821
823
827
829
839
853
857
859
863
877
881
883
887
907
911
919
929
937
941
947
953
967
971
977
983
991
997
1009
1013
1019
1021
1031
1033
1039
1049
1051
1061
1063
1069
1087
1091
1093
1097
1103
```

```
1109
1117
1123
1129
1151
1153
1163
1171
1181
1187
1193
1201
1213
1217
1223
1229
1231
1237
1249
1259
1277
1279
1283
1289
1291
1297
1301
1303
1307
1319
1321
1327
1361
1367
1373
1381
1399
1409
1423
1427
1429
1433
1439
1447
1451
1453
1459
1471
1481
1483
1487
1489
1493
1499
1511
1523
1531
1543
1549
1553
```

```
1559
1567
1571
1579
1583
1597
1601
1607
1609
1613
1619
1621
1627
1637
1657
1663
1667
1669
1693
1697
1699
1709
1721
1723
1733
1741
1747
1753
1759
1777
1783
1787
1789
1801
1811
1823
1831
1847
1861
1867
1871
1873
1877
1879
1889
1901
1907
1913
1931
1933
1949
1951
1973
1979
1987
1993
1997
1999
2003
2011
```

```
2017
2027
2029
2039
2053
2063
2069
2081
2083
2087
2089
2099
2111
2113
2129
2131
2137
2141
2143
2153
2161
2179
2203
2207
2213
2221
2237
2239
2243
2251
2267
2269
2273
2281
2287
2293
2297
2309
2311
2333
2339
2341
2347
2351
2357
2371
2377
2381
2383
2389
2393
2399
2411
2417
2423
2437
2441
2447
2459
2467
```

```
2473
2477
2503
2521
2531
2539
2543
2549
2551
2557
2579
2591
2593
2609
2617
2621
2633
2647
2657
2659
2663
2671
2677
2683
2687
2689
2693
2699
2707
2711
2713
2719
2729
2731
2741
2749
2753
2767
2777
2789
2791
2797
2801
2803
2819
2833
2837
2843
2851
2857
2861
2879
2887
2897
2903
2909
2917
2927
2939
2953
```

```
2957
2963
2969
2971
2999
3001
3011
3019
3023
3037
3041
3049
3061
3067
3079
3083
3089
3109
3119
3121
3137
3163
3167
3169
3181
3187
3191
3203
3209
3217
3221
3229
3251
3253
3257
3259
3271
3299
3301
3307
3313
3319
3323
3329
3331
3343
3347
3359
3361
3371
3373
3389
3391
3407
3413
3433
3449
3457
3461
3463
```

```
3467
3469
3491
3499
3511
3517
3527
3529
3533
3539
3541
3547
3557
3559
3571
3581
3583
3593
```

In [29]:
```
def gen():
    for i in range(1,11):
        yield i
```

In [34]:
```
g = gen()
```

In [35]:
```
next(g)
```

Out[35]: 1

In [36]:
```
next(g)
```

Out[36]: 2

In [38]:
```
list(g)
```

Out[38]: [3, 4, 5, 6, 7, 8, 9, 10]

In [39]:
```
for i in gen():
    print (i)
```

```
1
2
3
4
5
6
7
8
9
10
```

In [37]:
```
rs()
```

Out[37]: 'Sofia'

In [40]:
```
range(1,11)
```

Out[40]:
```
range(1, 11)
```

In [41]:
```
list(range(1,11))
```

Out[41]:
```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

In [42]:
```
d = {1:5, 2:6, 3:7}
```

In [44]:
```
list(d.keys())
```

Out[44]:
```
[1, 2, 3]
```

In [45]:
```
l = [1,2,3,4,5,6]
def f(x):
    return x+1
```

In [46]:
```
map(f, l)
```

Out[46]:
```
<map at 0x107c191d0>
```

In [47]:
```
list(map(f,l))
```

Out[47]:
```
[2, 3, 4, 5, 6, 7]
```

In [48]:
```
a = ['a', 'b', 'c']
```

In [49]:
```
enumerate(a)
```

Out[49]:
```
<enumerate at 0x107c041b0>
```

In [50]:
```
list(enumerate(a))
```

Out[50]:
```
[(0, 'a'), (1, 'b'), (2, 'c')]
```

In [53]:
```
def g_1():
    print ('HELLO FROM GENERATOR 1')
    for i in range(1,11):
        yield i


def g_2():
    print ('HELLOG FROM GENERATOR 2')
    for i in ['a', 'b', 'c']:
        yield i
```

In [54]:
```
g_1()
```

Out[54]:
```
<generator object g_1 at 0x107bfa750>
```

In [ ]:
```

```

In [52]:
```
from itertools import chain
```

In [55]:
```
gen_1 = g_1()
gen_2 = g_2()

gen_3 = chain(gen_1, gen_2)
```

In [68]:
```
next(gen_3)
```

Out[68]:
```
'c'
```

Exceptions

In [74]:
```

```

In [77]:
```
a = 5
b = 0

try:
    c = a/b
except Exception:
    print ('oops')
    c = 0
```

```
oops
```

In [78]:
```
if b != 0:
    c = a/b
else:
    c = 0
```

In [86]:
```
fn = 'asdfasdfasdf'

import os

if os.path.exists(fn):
    f = open(fn)

#open(fn)
```

In [87]:
```
# Ask forgiveness not permission"
try:
    f = open(fn)
except Exception:
    f = None
    print ('FIle not found')
```

```
FIle not found
```

In [88]:
```
adfasdf
```

```
---------------------------------------------------------------------
---------
NameError                                 Traceback (most recent c
all last)
<ipython-input-88-ac65243ad0d1> in <module>()
----> 1 adfasdf

NameError: name 'adfasdf' is not defined
```

In [93]:
```
try:
    dfgsdfgsdfg
    a=1/0
except NameError:
    print ('oops')
```

```
oops
```

In [94]:
```
try:
    a=1/0
    dfgsdfgsdfg
except NameError:
    print ('oops')
```

```
---------------------------------------------------------------------
---------
ZeroDivisionError                         Traceback (most recent c
all last)
<ipython-input-94-12a4695f749a> in <module>()
      1 try:
----> 2     a=1/0
      3     dfgsdfgsdfg
      4 except NameError:
      5     print ('oops')

ZeroDivisionError: division by zero
```

In [98]:
```
try:
    #a=1/0
    #dfgsdfgsdfg
    open('asdfasdfasf')
except NameError:
    print ('oops den yparxei')
except ZeroDivisionError:
    print ('diaireses me to 0')
except FileNotFoundError:
    print ('Den yparxei to arxeio')
```

```
Den yparxei to arxeio
```

In [97]:
```
open('asdafsdf')
```

```
---------------------------------------------------------------------
---------
FileNotFoundError                         Traceback (most recent c
all last)
<ipython-input-97-cad831ffd63f> in <module>()
----> 1 open('asdafsdf')

FileNotFoundError: [Errno 2] No such file or directory: 'asdafsdf'
```

In [102]:
```
try:
    #a=1/0
    dfgsdfgsdfg
    #open('asdfasdfasf')
except (NameError, ZeroDivisionError):
    print ('oops den yparxei, ή διαίρεση με το 0')
except FileNotFoundError:
    print ('Den yparxei to arxeio')
```

```
oops den yparxei, ή διαίρεση με το 0
```

In [90]:
```
rs()
```

Out[90]:
```
'Sofia'
```

In [116]:
```
try:
    #a=1/0
    #dfgsdfgsdfg
    open('asdfasdfasf')
except Exception:
    print ('error')
except (NameError, ZeroDivisionError):
    print ('oops den yparxei, ή διαίρεση με το 0')
except FileNotFoundError:
    print ('Den yparxei to arxeio')
```

```
error
```

In [121]:
```
try:
    #a=1/0
    #dfgsdfgsdfg
    open('asdfasdfasf')
except (NameError, ZeroDivisionError):
    print ('oops den yparxei, ή διαίρεση με το 0')
except FileNotFoundError:
    print ('Den yparxei to arxeio')
except Exception:
    print ('error')
```

```
Den yparxei to arxeio
```

In [124]:
```
try:
    #a=1/0
    #dfgsdfgsdfg
    #open('asdfasdfasf')
    [1,2,3][5]
except (NameError, ZeroDivisionError):
    print ('oops den yparxei, ή διαίρεση με το 0')
except FileNotFoundError:
    print ('Den yparxei to arxeio')
except Exception:
    print ('error')
```

```
error
```

In [133]:
```
try:
    #a=1/0
    #dfgsdfgsdfg
    #open('asdfasdfasf')
    [1,2,3][5]
except Exception as e:
    print ('error')
    print (type(e))
    print (str(e))
```

```
error
<class 'IndexError'>
list index out of range
```

In [144]:
```
def f(a,b):
    '''
    a and b should be integers!!!!!
    '''

    if not type(a) is int  or  not type(b) is int :
        #raise Exception('parameters are not integer')
        raise TypeError('parameters are not integer')

    return a+b
```

In [ ]:
```

```

In [ ]:
```
def f(a,b):
    '''
    a and b should be integers!!!!!
    '''

    if not type(a) is int  or  not type(b) is int :
        raise Exception('parameters are not integer')

    return a+b
```

In [139]:
```
f(5,3)
```

Out[139]:
```
8
```

In [145]:
```
f(5 , 4.2)
```

```
---------------------------------------------------------------------
---------
TypeError                                 Traceback (most recent c
all last)
<ipython-input-145-65ca43e145bd> in <module>()
----> 1 f(5 , 4.2)

<ipython-input-144-6ca11cd0aab2> in f(a, b)
      6     if not type(a) is int  or  not type(b) is int :
      7         #raise Exception('parameters are not integer')
----> 8         raise TypeError('parameters are not integer')
      9
     10     return a+b

TypeError: parameters are not integer
```

In [142]:
```
try:
    f(5 , 4.2)
except Exception as e:
    print ('Something bad happened')
```

```
Something bad happened
```

In [146]:
```
class MyFabulousException(Exception):
    '''
    This should not happen! At All!
    '''
```

In [148]:
```
raise MyFabulousException('sdsdfsdf')
```

```
---------------------------------------------------------------------
---------
MyFabulousException                       Traceback (most recent c
all last)
<ipython-input-148-c25a7acbfccf> in <module>()
----> 1 raise MyFabulousException('sdsdfsdf')

MyFabulousException: sdsdfsdf
```

In [ ]:
```

```

In [131]:
```
dir(e)
```

```
---------------------------------------------------------------------
---------
NameError                                 Traceback (most recent c
all last)
<ipython-input-131-d4599e4c0f76> in <module>()
----> 1 dir(e)

NameError: name 'e' is not defined
```

In [ ]:
```
try:
```

In [109]:
```
try :
    eval(' sum([for i in range(15)]) ')
except SyntaxError:
    print ('oops')
```

```
oops
```

In [151]:
```
try:
    #a=1/0
    a=1/2
except Exception as e:
    print ('this should not happen')
finally:
    print ('I will fix it here')
```

```
I will fix it here
```

In [153]:
```
try:
    a=1/0
    #a=1/2
except Exception as e:
    print ('this should not happen')
else:
    print ('Well Done')
finally:
    print ('I will fix it here')
```

```
this should not happen
I will fix it here
```

In [154]:
```
try:
    #a=1/0
    a=1/2
except Exception as e:
    print ('this should not happen')
else:
    print ('Well Done')
finally:
    print ('I will fix it here')
```

```
Well Done
I will fix it here
```

In [155]:
```
def f(x):

    if x > 10:
        raise Exception('NOOOOOOOOO')

    return x
```

In [159]:
```
def g(x):
    return f(x/10)
```

In [160]:
```
try:
    g(1000)
except:
    print ('G raised an exception!')
```

```
G raised an exception!
```

In [161]:
```
g(1000)
```

```
---------------------------------------------------------------------
---------
Exception                                 Traceback (most recent c
all last)
<ipython-input-161-903a7cf01ceb> in <module>()
----> 1 g(1000)

<ipython-input-159-9b0d03ae8a9f> in g(x)
      1 def g(x):
----> 2     return f(x/10)

<ipython-input-155-6973942e0c67> in f(x)
      2
      3     if x > 10:
----> 4         raise Exception('NOOOOOOOOO')
      5
      6     return x

Exception: NOOOOOOOOO
```

In [163]:
```
try:
    g(1000)
except Exception as e:
    print ('G raised an exception!')
    raise e
```

```
G raised an exception!

---------------------------------------------------------------------
---------
Exception                                 Traceback (most recent c
all last)
<ipython-input-163-67c70e719f32> in <module>()
      3 except Exception as e:
      4     print ('G raised an exception!')
----> 5     raise e
      6
      7

<ipython-input-163-67c70e719f32> in <module>()
      1 try:
----> 2     g(1000)
      3 except Exception as e:
      4     print ('G raised an exception!')
      5     raise e

<ipython-input-159-9b0d03ae8a9f> in g(x)
      1 def g(x):
----> 2     return f(x/10)

<ipython-input-155-6973942e0c67> in f(x)
      2
      3     if x > 10:
----> 4         raise Exception('NOOOOOOOOO')
      5
      6     return x

Exception: NOOOOOOOOO
```

In [ ]:

In [158]:
```
rs()
```

Out[158]:
```
'Tzwrtzina'
```

In [ ]:

In [106]:
```
3+1
```

Out[106]:
```
4
```

Numpy

In [164]:
```
import math

math.sqrt(20)
```

Out[164]:
```
4.47213595499958
```

In [166]:
```
import numpy as np
```

In [173]:
```
a = [1,2,3,4,3,2]
b = [4,3,2,3,2,1]

[sum(x) for x in zip(a,b)]
```

Out[173]:
```
[5, 5, 5, 7, 5, 3]
```

In [174]:
```
a
```

Out[174]:
```
[1, 2, 3, 4, 3, 2]
```

In [175]:
```
a_np = np.array(a)
```

In [177]:
```
b_np = np.array(b)
```

In [179]:
```
a_np
```

Out[179]:
```
array([1, 2, 3, 4, 3, 2])
```

In [180]:
```
b_np
```

Out[180]:
```
array([4, 3, 2, 3, 2, 1])
```

In [181]:
```
a_np + b_np
```

Out[181]:
```
array([5, 5, 5, 7, 5, 3])
```

In [183]:
```
a_np - b_np
```

Out[183]:
```
array([-3, -1,  1,  1,  1,  1])
```

In [185]:
```
c = np.vstack((a_np, b_np))
```

In [187]:
```
c
```

Out[187]:
```
array([[1, 2, 3, 4, 3, 2],
       [4, 3, 2, 3, 2, 1]])
```

In [188]:
```
c.shape
```

Out[188]:
```
(2, 6)
```

In [212]:
```
a = np.random.random((4,3))
```

In [192]:
```
a
```

Out[192]:
```
array([[0.91437693, 0.25769378, 0.64537131],
       [0.67786047, 0.70912712, 0.97499621],
       [0.74118468, 0.25274784, 0.42444292],
       [0.81939891, 0.71757325, 0.91173435]])
```

In [194]:
```
a[ 1 , 2 ]
```

Out[194]:
```
0.9749962091160524
```

In [195]:
```
a[ 0:3 , 2 ]
```

Out[195]:
```
array([0.64537131, 0.97499621, 0.42444292])
```

In [196]:
```
a[ 0:3 , 1:3 ]
```

Out[196]:
```
array([[0.25769378, 0.64537131],
       [0.70912712, 0.97499621],
       [0.25274784, 0.42444292]])
```

In [197]:
```
a[ 0:3 , -1 ]
```

Out[197]:
```
array([0.64537131, 0.97499621, 0.42444292])
```

In [198]:
```
a[ 0:3 ,  ]
```

Out[198]:
```
array([[0.91437693, 0.25769378, 0.64537131],
       [0.67786047, 0.70912712, 0.97499621],
       [0.74118468, 0.25274784, 0.42444292]])
```

In [203]:
```
a[ : , 0:2]
```

Out[203]:
```
array([[0.91437693, 0.25769378],
       [0.67786047, 0.70912712],
       [0.74118468, 0.25274784],
       [0.81939891, 0.71757325]])
```

In [204]:
```
a
```

Out[204]:
```
array([[0.91437693, 0.25769378, 0.64537131],
       [0.67786047, 0.70912712, 0.97499621],
       [0.74118468, 0.25274784, 0.42444292],
       [0.81939891, 0.71757325, 0.91173435]])
```

In [205]:
```
type(a)
```

Out[205]:
```
numpy.ndarray
```

In [206]:
```
a = [[1,2,3], [4,5,6]]
```

In [210]:
```
a[1][1] = { 'a':1, 'b': set([5,6,7,8])}
```

In [211]:
```
a
```

Out[211]:
```
[[1, 2, 3], [4, {'a': 1, 'b': {5, 6, 7, 8}}, 6]]
```

In [ ]:
```
```

In [209]:
```
a
```

Out[209]:
```
[[1, 2, 3], [4, 5, 6]]
```

In [213]:
```
a = np.random.random((4,3))
```

In [214]:
```
a
```

Out[214]:
```
array([[0.95580993, 0.08631511, 0.67902571],
       [0.62001427, 0.80377634, 0.67783664],
       [0.84293146, 0.84259364, 0.4136305 ],
       [0.56984652, 0.97092873, 0.25443137]])
```

In [217]:
```
a [ :2  ,    ]
```

Out[217]:
```
array([[0.95580993, 0.08631511, 0.67902571],
       [0.62001427, 0.80377634, 0.67783664]])
```

In [221]:
```
a[ [0,2]  , [1]  ]
```

Out[221]:
```
array([0.08631511, 0.84259364])
```

In [222]:
```
a
```

Out[222]:
```
array([[0.95580993, 0.08631511, 0.67902571],
       [0.62001427, 0.80377634, 0.67783664],
       [0.84293146, 0.84259364, 0.4136305 ],
       [0.56984652, 0.97092873, 0.25443137]])
```

In [223]:
```
a[ [0,2]  , [1]  ]
```

Out[223]:
```
array([0.08631511, 0.84259364])
```

In [224]:
```
a[ [True, False, False, True]  ,    ]
```

Out[224]:
```
array([[0.95580993, 0.08631511, 0.67902571],
       [0.56984652, 0.97092873, 0.25443137]])
```

In [226]:
```
a[ [True, False, False, True]  ,  [ False, True, True ]  ]
```

Out[226]:
```
array([0.08631511, 0.25443137])
```

In [227]:
```
a
```

Out[227]:
```
array([[0.95580993, 0.08631511, 0.67902571],
       [0.62001427, 0.80377634, 0.67783664],
       [0.84293146, 0.84259364, 0.4136305 ],
       [0.56984652, 0.97092873, 0.25443137]])
```

In [228]:
```
a [ [0,2] ,  ]
```

Out[228]:
```
array([[0.95580993, 0.08631511, 0.67902571],
       [0.84293146, 0.84259364, 0.4136305 ]])
```

In [229]:
```
a [ [0,0,0,0,2,2,2] ,  ]
```

Out[229]:
```
array([[0.95580993, 0.08631511, 0.67902571],
       [0.95580993, 0.08631511, 0.67902571],
       [0.95580993, 0.08631511, 0.67902571],
       [0.95580993, 0.08631511, 0.67902571],
       [0.84293146, 0.84259364, 0.4136305 ],
       [0.84293146, 0.84259364, 0.4136305 ],
       [0.84293146, 0.84259364, 0.4136305 ]])
```

In [233]:
```
a [ [0,0,0,0,2,2,2] , 1:3 ]
```

Out[233]:
```
array([[0.08631511, 0.67902571],
       [0.08631511, 0.67902571],
       [0.08631511, 0.67902571],
       [0.08631511, 0.67902571],
       [0.84259364, 0.4136305 ],
       [0.84259364, 0.4136305 ],
       [0.84259364, 0.4136305 ]])
```

In [234]:
```
a
```

Out[234]:
```
array([[0.95580993, 0.08631511, 0.67902571],
       [0.62001427, 0.80377634, 0.67783664],
       [0.84293146, 0.84259364, 0.4136305 ],
       [0.56984652, 0.97092873, 0.25443137]])
```

In [236]:
```
b = [[True, False, False], [True, False, True], [False, False, Fals
e], [True, True, False]]
```

In [237]:
```
b = np.array(b)
```

In [238]:
```
a
```

Out[238]:
```
array([[0.95580993, 0.08631511, 0.67902571],
       [0.62001427, 0.80377634, 0.67783664],
       [0.84293146, 0.84259364, 0.4136305 ],
       [0.56984652, 0.97092873, 0.25443137]])
```

In [239]:
```
b
```

Out[239]:
```
array([[ True, False, False],
       [ True, False,  True],
       [False, False, False],
       [ True,  True, False]])
```

In [240]:
```
a[b]
```

Out[240]:
```
array([0.95580993, 0.62001427, 0.67783664, 0.56984652, 0.9709287
3])
```

In [241]:
```
a[b] = 7
```

In [243]:
```
a
```

Out[243]:
```
array([[7.        , 0.08631511, 0.67902571],
       [7.        , 0.80377634, 7.        ],
       [0.84293146, 0.84259364, 0.4136305 ],
       [7.        , 7.        , 0.25443137]])
```

In [244]:
```
a = np.random.random((4,3))
```

In [246]:
```
a
```

Out[246]:
```
array([[0.75406692, 0.68786308, 0.9918082 ],
       [0.75447414, 0.9260037 , 0.28090686],
       [0.92425588, 0.14333805, 0.60562478],
       [0.63416726, 0.83631617, 0.65823574]])
```

In [247]:
```
b = np.ones((2,2))
```

In [249]:
```
b = 7 * b
```

In [250]:
```
b
```

Out[250]:
```
array([[7., 7.],
       [7., 7.]])
```

In [251]:
```
a
```

Out[251]:
```
array([[0.75406692, 0.68786308, 0.9918082 ],
       [0.75447414, 0.9260037 , 0.28090686],
       [0.92425588, 0.14333805, 0.60562478],
       [0.63416726, 0.83631617, 0.65823574]])
```

In [252]:
```
b
```

Out[252]:
```
array([[7., 7.],
       [7., 7.]])
```

In [256]:
```
a[ 1:3 , 1:3 ] = b
```

In [257]:
```
a
```

Out[257]:
```
array([[0.75406692, 0.68786308, 0.9918082 ],
       [0.75447414, 7.        , 7.        ],
       [0.92425588, 7.        , 7.        ],
       [0.63416726, 0.83631617, 0.65823574]])
```

In [258]:
```
a = np.random.random((4,3))
```

In [259]:
```
a
```

Out[259]:
```
array([[0.708167  , 0.37660155, 0.7036449 ],
       [0.23996364, 0.89560585, 0.76374009],
       [0.5648255 , 0.62668968, 0.04869421],
       [0.036478  , 0.59991475, 0.2736219 ]])
```

In [260]:
```
a < 0.5
```

Out[260]:
```
array([[False,  True, False],
       [ True, False, False],
       [False, False,  True],
       [ True, False,  True]])
```

In [262]:
```
a[ a<0.5 ]
```

Out[262]:
```
array([0.37660155, 0.23996364, 0.04869421, 0.036478  , 0.2736219
])
```

In [263]:
```
a[ a<0.5 ] = 0
```

In [ ]:
```

```

In [264]:
```
a
```

Out[264]:
```
array([[0.708167  , 0.        , 0.7036449 ],
       [0.        , 0.89560585, 0.76374009],
       [0.5648255 , 0.62668968, 0.        ],
       [0.        , 0.59991475, 0.        ]])
```

In [267]:
```
a = np.random.random((4,3))
```

In [266]:
```
a
```

Out[266]:
```
array([[0.43371288, 0.22946899, 0.25804008],
       [0.24412603, 0.99910025, 0.00554241],
       [0.61658454, 0.42487768, 0.70431454],
       [0.72638369, 0.52883237, 0.56692966]])
```

In [ ]:
```
a<
```

In [261]:
```
a + 10
```

Out[261]:
```
array([[10.708167  , 10.37660155, 10.7036449 ],
       [10.23996364, 10.89560585, 10.76374009],
       [10.5648255 , 10.62668968, 10.04869421],
       [10.036478  , 10.59991475, 10.2736219 ]])
```

In [268]:
```
a = np.random.random((4,3))
```

In [269]:
```
a
```

Out[269]:
```
array([[0.083404  , 0.33272289, 0.98027558],
       [0.10288884, 0.72559567, 0.8127173 ],
       [0.61853214, 0.38826421, 0.84282034],
       [0.15888296, 0.07600264, 0.22165088]])
```

In [278]:
```
a [  ~ (a<0.5)  ]
```

Out[278]:
```
array([0.98027558, 0.72559567, 0.8127173 , 0.61853214, 0.8428203
4])
```

In [279]:
```
a
```

Out[279]:
```
array([[0.083404  , 0.33272289, 0.98027558],
       [0.10288884, 0.72559567, 0.8127173 ],
       [0.61853214, 0.38826421, 0.84282034],
       [0.15888296, 0.07600264, 0.22165088]])
```

In [281]:
```
# 0.3 < x < 0.7
```

In [283]:
```
(a>0.3) and (a<0.7)
```

```
---------------------------------------------------------------------
---------
ValueError                                Traceback (most recent c
all last)
<ipython-input-283-4164b66ab55e> in <module>()
----> 1 (a>0.3) and (a<0.7)

ValueError: The truth value of an array with more than one element
is ambiguous. Use a.any() or a.all()
```

In [286]:
```
(a>0.3) & (a<0.7)
```

Out[286]:
```
array([[False,  True, False],
       [False, False, False],
       [ True,  True, False],
       [False, False, False]])
```

In [287]:
```
a>0.3
```

Out[287]:
```
array([[False,  True,  True],
       [False,  True,  True],
       [ True,  True,  True],
       [False, False, False]])
```

In [288]:
```
a<0.7
```

Out[288]:
```
array([[ True,  True, False],
       [ True, False, False],
       [ True,  True, False],
       [ True,  True,  True]])
```

In [290]:
```
(a>0.3) or (a<0.7)
```

```
---------------------------------------------------------------------
---------
ValueError                                Traceback (most recent c
all last)
<ipython-input-290-a44ef58f71f2> in <module>()
----> 1 (a>0.3) or (a<0.7)

ValueError: The truth value of an array with more than one element
is ambiguous. Use a.any() or a.all()
```

In [293]:
```
(a<0.3) | (a>0.7)
```

Out[293]:
```
array([[ True, False,  True],
       [ True,  True,  True],
       [False, False,  True],
       [ True,  True,  True]])
```

In [292]:
```
a
```

Out[292]:
```
array([[0.083404  , 0.33272289, 0.98027558],
       [0.10288884, 0.72559567, 0.8127173 ],
       [0.61853214, 0.38826421, 0.84282034],
       [0.15888296, 0.07600264, 0.22165088]])
```

In [294]:
```
a
```

Out[294]:
```
array([[0.083404  , 0.33272289, 0.98027558],
       [0.10288884, 0.72559567, 0.8127173 ],
       [0.61853214, 0.38826421, 0.84282034],
       [0.15888296, 0.07600264, 0.22165088]])
```

In [302]:
```
l = []
for x in a:
    s = sum(x)
    l.append(s)
print (l)
```

```
[1.396402472025978, 1.6412018149617655, 1.8496166822080338, 0.4565
364891972742]
```

In [303]:
```
np.sum(a, axis=1)
```

Out[303]:
```
array([1.39640247, 1.64120181, 1.84961668, 0.45653649])
```

In [304]:
```
a
```

Out[304]:
```
array([[0.083404  , 0.33272289, 0.98027558],
       [0.10288884, 0.72559567, 0.8127173 ],
       [0.61853214, 0.38826421, 0.84282034],
       [0.15888296, 0.07600264, 0.22165088]])
```

In [305]:
```
a.T
```

Out[305]:
```
array([[0.083404  , 0.10288884, 0.61853214, 0.15888296],
       [0.33272289, 0.72559567, 0.38826421, 0.07600264],
       [0.98027558, 0.8127173 , 0.84282034, 0.22165088]])
```

In [306]:
```
a.dot(a.T)
```

Out[306]:
```
array([[1.07860096, 1.04669056, 1.00696864, 0.25581824],
       [1.04669056, 1.19758461, 1.03033756, 0.25163398],
       [1.00696864, 1.03033756, 1.24367722, 0.3145952 ],
       [0.25581824, 0.25163398, 0.3145952 , 0.08014931]])
```

In [308]:
```
(a.T).dot(a)
```

Out[308]:
```
array([[0.42536814, 0.35463553, 0.72190646],
       [0.35463553, 0.79371909, 1.25994731],
       [0.72190646, 1.25994731, 2.38092487]])
```

In [309]:
```
a
```

Out[309]:
```
array([[0.083404  , 0.33272289, 0.98027558],
       [0.10288884, 0.72559567, 0.8127173 ],
       [0.61853214, 0.38826421, 0.84282034],
       [0.15888296, 0.07600264, 0.22165088]])
```

In [310]:
```
a.reshape((2,6))
```

Out[310]:
```
array([[0.083404  , 0.33272289, 0.98027558, 0.10288884, 0.7255956
7,
        0.8127173 ],
       [0.61853214, 0.38826421, 0.84282034, 0.15888296, 0.0760026
4,
        0.22165088]])
```

In [312]:
```
a.reshape((6,2))
```

Out[312]:
```
array([[0.083404  , 0.33272289],
       [0.98027558, 0.10288884],
       [0.72559567, 0.8127173 ],
       [0.61853214, 0.38826421],
       [0.84282034, 0.15888296],
       [0.07600264, 0.22165088]])
```

In [313]:
```
a.reshape((6,3))
```

```
---------------------------------------------------------------------
---------
ValueError                                Traceback (most recent c
all last)
<ipython-input-313-005a9e3b3202> in <module>()
----> 1 a.reshape((6,3))

ValueError: cannot reshape array of size 12 into shape (6,3)
```

In [314]:
```
a
```

Out[314]:
```
array([[0.083404  , 0.33272289, 0.98027558],
       [0.10288884, 0.72559567, 0.8127173 ],
       [0.61853214, 0.38826421, 0.84282034],
       [0.15888296, 0.07600264, 0.22165088]])
```

In [316]:
```
a = np.vstack((a, [1,2,3]))
```

In [317]:
```
a
```

Out[317]:
```
array([[0.083404  , 0.33272289, 0.98027558],
       [0.10288884, 0.72559567, 0.8127173 ],
       [0.61853214, 0.38826421, 0.84282034],
       [0.15888296, 0.07600264, 0.22165088],
       [1.        , 2.        , 3.        ]])
```

In [323]:
```
a = np.hstack((a, np.array([5,6,7,8,9])  ))
```

```
---------------------------------------------------------------------
---------
ValueError                                Traceback (most recent c
all last)
<ipython-input-323-bc11a1ee0a7b> in <module>()
----> 1 a = np.hstack((a, np.array([5,6,7,8,9])  ))

~/anaconda3/lib/python3.7/site-packages/numpy/core/shape_base.py i
n hstack(tup)
    286         return _nx.concatenate(arrs, 0)
    287     else:
--> 288         return _nx.concatenate(arrs, 1)
    289
    290

ValueError: all the input arrays must have same number of dimensio
ns
```

In [320]:
```
a
```

Out[320]:
```
array([[0.083404  , 0.33272289, 0.98027558],
       [0.10288884, 0.72559567, 0.8127173 ],
       [0.61853214, 0.38826421, 0.84282034],
       [0.15888296, 0.07600264, 0.22165088],
       [1.        , 2.        , 3.        ]])
```

In [324]:
```
b = np.array([5,6,7,8,9])
b
```

Out[324]:
```
array([5, 6, 7, 8, 9])
```

In [326]:
```
np.hstack((a,b.T))
```

```
---------------------------------------------------------------------
---------
ValueError                                Traceback (most recent c
all last)
<ipython-input-326-7c3ededac04c> in <module>()
----> 1 np.hstack((a,b.T))

~/anaconda3/lib/python3.7/site-packages/numpy/core/shape_base.py i
n hstack(tup)
    286         return _nx.concatenate(arrs, 0)
    287     else:
--> 288         return _nx.concatenate(arrs, 1)
    289
    290

ValueError: all the input arrays must have same number of dimensio
ns
```

In [327]:
```
a
```

Out[327]:
```
array([[0.083404  , 0.33272289, 0.98027558],
       [0.10288884, 0.72559567, 0.8127173 ],
       [0.61853214, 0.38826421, 0.84282034],
       [0.15888296, 0.07600264, 0.22165088],
       [1.        , 2.        , 3.        ]])
```

In [328]:
```
b
```

Out[328]:
```
array([5, 6, 7, 8, 9])
```

In [330]:
```
b = np.array([[5], [6], [7], [8], [9]])
```

In [332]:
```
np.hstack((a,b))
```

Out[332]:
```
array([[0.083404  , 0.33272289, 0.98027558, 5.        ],
       [0.10288884, 0.72559567, 0.8127173 , 6.        ],
       [0.61853214, 0.38826421, 0.84282034, 7.        ],
       [0.15888296, 0.07600264, 0.22165088, 8.        ],
       [1.        , 2.        , 3.        , 9.        ]])
```

In [333]:
```
?np.tile
```

In [ ]:
```

```

In [344]:
```
b = np.array([5,6,7,8,9])
np.hstack((a, b.reshape((5,1))))
```

Out[344]:
```
array([[0.083404  , 0.33272289, 0.98027558, 5.        ],
       [0.10288884, 0.72559567, 0.8127173 , 6.        ],
       [0.61853214, 0.38826421, 0.84282034, 7.        ],
       [0.15888296, 0.07600264, 0.22165088, 8.        ],
       [1.        , 2.        , 3.        , 9.        ]])
```

In [345]:
```
a
```

Out[345]:
```
array([[0.083404  , 0.33272289, 0.98027558],
       [0.10288884, 0.72559567, 0.8127173 ],
       [0.61853214, 0.38826421, 0.84282034],
       [0.15888296, 0.07600264, 0.22165088],
       [1.        , 2.        , 3.        ]])
```

In [352]:
```
b = np.random.random((5,3))
```

In [353]:
```
a
```

Out[353]:
```
array([[0.083404  , 0.33272289, 0.98027558],
       [0.10288884, 0.72559567, 0.8127173 ],
       [0.61853214, 0.38826421, 0.84282034],
       [0.15888296, 0.07600264, 0.22165088],
       [1.        , 2.        , 3.        ]])
```

In [354]:
```
b
```

Out[354]:
```
array([[0.68227114, 0.21335234, 0.24030355],
       [0.91131236, 0.23860334, 0.97262899],
       [0.16363118, 0.26472247, 0.94543148],
       [0.28880624, 0.45008973, 0.21267164],
       [0.68967187, 0.67631857, 0.07650825]])
```

In [355]:
```
a < b
```

Out[355]:
```
array([[ True, False, False],
       [ True, False,  True],
       [False, False,  True],
       [ True,  True, False],
       [False, False, False]])
```

In [ ]:

In [342]:

In [ ]:

In [ ]:

In [337]:
```
b.reshape((1,-1))
```

Out[337]:
```
array([[5, 6, 7, 8, 9]])
```

In [339]:
```
b.reshape((5,1))
```

Out[339]:
```
array([[5],
       [6],
       [7],
       [8],
       [9]])
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [296]:
```
np.max(a)
```

Out[296]:
```
0.9802755794006675
```

In [297]:
```
np.max(a, axis=1)
```

Out[297]:
```
array([0.98027558, 0.8127173 , 0.84282034, 0.22165088])
```

In [299]:
```
np.max(a, axis=0)
```

Out[299]:
```
array([0.61853214, 0.72559567, 0.98027558])
```

In [300]:
```
np.sum(a, axis=1)
```

Out[300]:
```
array([1.39640247, 1.64120181, 1.84961668, 0.45653649])
```

In [ ]:

In [289]:
```
0.3<a<0.7
```

```
---------------------------------------------------------------------
---------
ValueError                                Traceback (most recent c
all last)
<ipython-input-289-5b82da8652fc> in <module>()
----> 1 0.3<a<0.7

ValueError: The truth value of an array with more than one element
is ambiguous. Use a.any() or a.all()
```

In [ ]:

In [284]:
```
5+3
```

Out[284]:
```
8
```

In [285]:
```
a + a
```

Out[285]:
```
array([[0.166808  , 0.66544579, 1.96055116],
       [0.20577768, 1.45119134, 1.62543461],
       [1.23706427, 0.77652841, 1.68564068],
       [0.31776592, 0.15200529, 0.44330177]])
```

In [ ]:

In [280]:
```
rs()
```

Out[280]:
```
'Antwnia'
```

In [ ]:

In [ ]:

In [253]:
```
rs()
```

Out[253]:
```
'Xristos'
```

In [ ]:

Generator Comprehensions

In [168]:
```
rs()
```

Out[168]:
```
'Andromaxh'
```

In [ ]:

In [ ]:
```
# ! pip install numpy
```