NUMPY BASICS

1) numpy\_using\_float

def attempt\_float(x):

 try:return float(x)

 except:

  return x

print(attempt\_float('1.2345'))

2) numpy\_using\_mathimatical\_operator

import numpy as np

data = np.random.randn(2, 3)

print(data)

3) numpy\_iterator

import numpy as np

some\_dict = {'a': 1, 'b': 2, 'c': 3}

dict\_iterator = iter(some\_dict)

print(dict\_iterator)

4) numpy\_iterator2

import numpy as np

some\_dict = {'a': 1, 'b': 2, 'c': 3}

dict\_iterator = iter(some\_dict)

print(list(dict\_iterator))

5) set\_is\_subset or\_subset

import numpy as np

a\_set = {1, 2, 3, 4, 5}

print({1, 2, 3}.issubset(a\_set))

6) numpy\_iterable

import numpy as np

some\_dict = {'a': 1, 'b': 2, 'c': 3}

for key in some\_dict:

    print(key)

 7) numpy\_itertools

import itertools

first\_letter = lambda x: x[0]

names = ['Alan', 'Adam', 'Wes', 'Will', 'Albert', 'Steven']

for letter, names in itertools.groupby(names, first\_letter):

    print(letter, list(names))

8) Bollean\_indexing

import numpy as np

names = np.array(['Bob', 'Joe', 'Will', 'Bob', 'Will', 'Joe', 'Joe'])

data = np.random.randn(7, 4)

print(names)

9) boolean\_indexing2

import numpy as np

names = np.array(['Bob', 'Joe', 'Will', 'Bob', 'Will', 'Joe', 'Joe'])

data = np.random.randn(7, 4)

print(data)

10) binary\_or\_unary\_functions

import numpy as np

x = np.random.randn(8)

print(x)

11)  binary\_or\_unary\_functions2

import numpy as np

x = np.random.randn(8)

y = np.random.randn(8)

print(np.maximum(x, y))

12) ndarray

import numpy as np

arr = np.array([[1,2,3,4],[5,6,7,8]])

print(arr.ndim)

print(arr.shape)

print(arr.dtype)

13) ndarray2

import numpy as np

print(np.zeros(15))

print(np.zeros((3, 6)))

14) ndarray3

import numpy as np

print(np.arange(15))

15) numpy\_4x4\_matrices

import numpy as np

arr = np.array([[1,2,3,4],[5,6,7,8],[9,10,11,12],[13,14,15,16]])

print(arr)

print(arr.shape)

16) array\_function

import numpy as np

arr = np.arange(10)

print(np.exp(arr))

17) arranging\_objects

import numpy as np

strings = ['a', 'as', 'bat', 'car', 'dove', 'python']

print([x.upper() for x in strings if len(x) > 2])

18) numpy\_arranging\_objects

import numpy as np

strings = ['a', 'as', 'bat', 'car', 'dove', 'python']

loc\_mapping = {val : index for index, val in enumerate(strings)}

print(loc\_mapping)

19) numpy\_flattended\_function

import numpy as np

some\_tuples = [(1, 2, 3), (4, 5, 6), (7, 8, 9)]

flattened = [x for tup in some\_tuples for x in tup]

print(flattened)

20) numpy\_using\_tuples

import numpy as np

some\_tuples = [(1, 2, 3), (4, 5, 6), (7, 8, 9)]

flattened = []

for tup in some\_tuples:

    for x in tup:

     (flattened.append(x))

     print([[x for x in tup] for tup in some\_tuples])