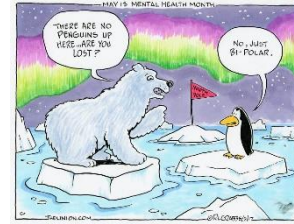


## ARRAYS & LOOPS & FUNCTIONS

File: ARRAY and LOOPS. The Bipolar Bear-V3.docx

Solutions: BipolarBear.py



Cartoon from: <https://images.app.goo.gl/EzjaD8uxx18F7nKD9>

Document Check List (for the Ex#2 Portfolio):

	EX#1	EX#2	EX#3	EX#4	EX#5	EX#6
Code						
Output						

In these exercises you will practice input, output, and processing of one-dimensional arrays. Under the umbrella of processing you will learn to implement arrays in a programming solution (otherwise scalar), manipulate 1D array indexing, slicing, iteration (using loops, if), array creation (np.array, np.linspace, np.arange, np.ones, np.zeros, arrayname.fill, etc.), plot, and library functions accepting arrays (e.g., np.min, np.max, np.mean, np.average, etc.),

For the following problems (1-3) in (A) use programming structures, e.g., loops and if statements. Afterwards in (B) use library functions.

**Ex-1.** Write python code to input the array  $X = [10, 9, 8, 7, 6, 5, 4, 3, 2, 1]$  and compute the natural log of each element to construct the array  $X_{\log}$ .

- (A) Use loops
- (B) Use a numpy function

Don't forget to output your resulting array.

**Ex-2.** Write code to compute the average and the largest element of a 1D array. Test your program with the following set of values:  $A = [3, 4, 2, 6, 4, 3, 7, 8, 5, 4, 5, 6, 7]$ ;

- (A) Use loops (can't use mean(), average() or sort() functions)
- (B) Mean(), average() and all library functions allowed

Don't forget to output your resulting array.

**Ex-3.** Flip left to right the 1D array  $ZAP = [1, 2, 3, \dots, 10]$ ; and store the results in the new array PAZ whose elements would be  $PAZ = [10, 9, 7, \dots, 1]$ .

- (A) Use loops (and functions, such as append, or concatenate, hstack, etc.)
- (B) Use slicing

Solution: appendArray.py

**Ex-4.** Complete the code to filter multiples of 13 in the random array “azar” created with the following lines of code:

```
import numpy as np
np.set_printoptions(precision=2)
N=100 # for 22 players

# 1D array of float numbers within [1.0,100.0] range:
azar= np.random.uniform(1.0, 100.0, size=(N,))      # function is “uniform”
print("random float array azar in the range [1.0, 100.0]= \n", azar,"\n")
```

Don’t forget to store and output your resulting array with multiples of 13.

**Ex-5.** Write code to replace the element valued 7 by 14 for the array constructed by the statement:

```
X=numpy.array([y for y in range(11)]),
```

such as the new array X is:

```
[ 0 1 2 3 4 5 6 14 8 9 10]
```

**Ex-6.** Run the following code [keep the code indentation]. Problem description is at the end of L#11 presentation. Be aware the plot is shown in a separate window, named Fig XX.

```
# -*- coding: utf-8 -*-
"""
FILE: plotCoronaV.py, Created on Sun Apr 5 21:58:01 2020, @author: Marco
Estimate the coronavirus cases in Puerto Rico by exponential growth.
"""

import numpy as np
import matplotlib.pyplot as plt

# time vs infected cases
b=1.2
xo=1.0; x=xo; t=0.0
tt=np.array([t])      # Construc an array of t
xx=np.array([x])      # Construct an array of x
print('days cases')  # Table title {you can do better}
while x<=3.195e6:
    x=xo*b**t
    print('%.0f %.9f'%(t,x))
    xx=np.append(xx,x)    # Stores x values
    t+=1.0
    tt=np.append(tt,t)    # Stores t values

print("\nAll infected by day",int(t-1),"th")
```

```
# Plot starts here
fig = plt.figure()
ax = plt.axes()
ax.plot(tt,xx)
```

SOME SOLUTIONS:

EX#3 Solution

Flipping:

```
import numpy as np
```

```
zap=np.array([10,9,8,7,6,5,4,3,2,1], float)
paz=np.array([])
```

```
print("Append function")
for i in range(np.size(zap)-1,-1,-1):
    paz=np.append(paz,zap[i])
print(paz)
```

```
print("Hstack function:")
paz=np.array([])
for i in range(np.size(zap)-1,-1,-1):
    paz=np.hstack((paz,zap[i]))
print(paz)
```

```
print("Concatenate two arrays")
paz=np.array([])
for i in range(np.size(zap)-1,-1,-1):
    arr=np.array([zap[i]])
    paz=np.concatenate((paz,arr),axis=0)
print(paz)
```

```
print("Slicing")
paz=zap[-1::-1]
print(paz)
```

```
print("Reverse a List")
paz=list(zap)
paz.reverse() # reserve is python built in
print(paz)
```

