

Homework Assignment 1

Uploaded your solutions as ipynb Jupyter notebook file to <http://bCourses.berkeley.edu>.

(1) Here is a useful operator *= (6pts):

Enter the following commands:

```
a = 1
b = a
a *= 2
print(a)
print(b)
```

Explain with your own word what does the operator does! Look up the online documentation of Python operators, select four more two-symbol operators, provide an example calculation, and explain what your four operators do.

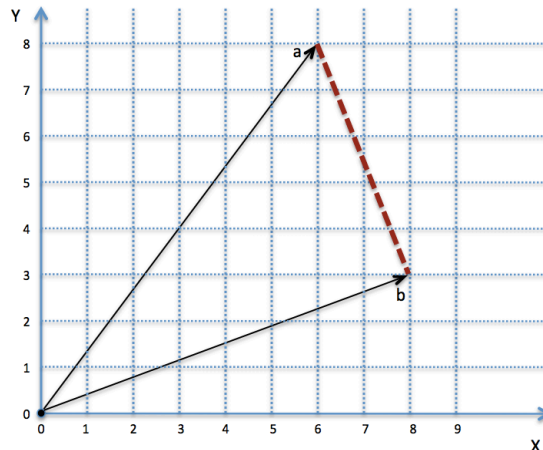
(2) Let us apply the *= operator of a 1D Numpy array (2pts):

After loading Numpy as in the lab assignment enter the following commands:

```
a = np.array( [ 1, 2, 3, 3, 5, 5 ] )
b = a
a *=2
print(a)
print(b)
```

One is inclined to say that something went wrong here. Explain why the behavior of the arrays is different from the behavior of the numbers above. Then correct these commands above so that the behavior of 'a' and 'b' is the same as above.

(3) Vectors in 2D (6pts):



Vectors are essentially lists of numbers but they can also have a geometric meaning if one assigns the first number to x and the second number to the y coordinate.

- (i) Please assign the $(x=6, y=8)$ vector to a 1D Numpy array a . Print the length of this vector using the Numpy function `np.linalg.norm()`.
- (ii) Assign vector b according to the diagram above. Using your variables a and b , compute the distance between points a and b with just one line of code.

(4) 2D plots (6pts):

Make a XY plot for the functions

$$f(x) = 3 \cos(x)$$

$$f(x) = 2 \cos(2x)$$

$$f(x) = \cos(3x)$$

The x values should be in degrees (not radians) and vary between 0 and 360. Please use different lines styles for each function. Label every curve as well as your axes. Use at least 100 points per curve.

(5) More 2D plots (8pts):

Make one XY plot for $0 \leq x \leq 1$ containing the following 6 functions

$$f(x) = x$$

$$g(x) = r x (1-x)$$

with $r = \{0.5, 1.5, 2.5, 3.5, \text{ and } 4\}$. No loop of r values is needed. Instead use six different plot commands. Label the axes, choose very different line styles for every curve, and add a legend that specify r in each case. Explain for what values of r do the curves $f(x)$ and $g(x)$ intersect?

(6) Even More 2D plots (4pts):

Produce and submit a second XY graph that shows $f(x)$ versus x as well as the function $h(x)$ for all five r values:

$$g(x) = r x (1-x)$$

$$h(x) = g(g(x)) = r g(x) (1-g(x))$$

(7) Plotting the world's CO₂ record (6pts):

In the lab, we practiced plotting the world's temperature data. Now we want to repeat these steps with the world's CO₂ data that is stored in columns 1 and 2 in a file `ice_core_CO2_data.txt`. Download this file and generate a plot CO₂ contents versus time. As in the lab, add a shaded region that illustrates the deviations from the averaged CO₂ contents.

When you are done, click "Kernel->Restart & Run All", check that all still worked, and only then submit your notebook file to bCourses.

(8) Loop and if statements (12pts):

Please cut and paste the following Python statements. In every case, give an explanation what happens in every case and why only certain 'i' values are printed.

```
for i in range(1,20):
    print(i)
```

```
for i in range(1,20,5):
    print(i)
```

```
for i in range(1,-20,-1):
    print(i)
```

```
for i in range(1,10,2):
    if (i < 5):
        print(i)
```

```
for i in range(1,20,2):
    print("All i values:",i,i%5)
    if (i % 5 == 0):
        print("*** Inside if:",i, i/5, i%5 )
```

```
for i in range(20):
    print(i)
    if (i%2==0 and i<20):
        print('***',i, i%5, i*i,i**2 )
```

(9) Enumerate (4pts): Please add the missing print statement

```
# For all labs and homeworks, please load python's numerical (numpy) and graphics (matplotlib) libraries
import numpy as np
import matplotlib.pyplot as plt
%config InlineBackend.figure_format = 'retina'
```

```
a = np.linspace(0,2,21)
```

```
# Let's practise the command 'enumerate' by executing the following lines:
for x,y in enumerate(a):
    print(x,y)
```

```
# Without defining new variables, add a print(...) command below so that the output
# is the same as that of print(x,y) above
for i in range(a.shape[0]):
    ...
```