

Homework #26 Numpy.

Note:

Please, run your code before submitting. If you get an error, try to fix it before submitting your homework. If you get help from anyone, please, make sure that you actually understand the solution.

Tasks:

1. Using `np.array()`, manually create a numpy array that looks like this:

5	0
7	1

Hint: you can either pass nested list to `np.array`, or use `array.reshape(x,y)` method.

2. Using either `np.zeros()` or `np.ones()`, create a two-dimensional numpy array that has 3 rows and 4 columns. The data type of the array should be `np.int32`. To specify the number of rows and columns use shape parameter in functions `np.zeros()` and `np.ones()`.
3. Try adding a number (for example 9) to your array from task 2. Describe what output do you get.

- Using `numpy.random.randint` create a 5 by 5 array, with integer values randomly drawn between 4 and 8 (included). Print the value of the cell located in the second row and third column of the array.

Hint: the function `randint()` uses following syntax:

```
random.randint(low, high=None, size=(x,y), dtype=int)
```

Where `low` and `high` describe the interval from which the values are randomly drawn and `size` is a tuple with the shape parameters.

- Create a 5 by 5 array, with values randomly drawn between 0.1 and 0.2. This task might require some creativity. Hint: you can create an array, and divide its values.

Print all values located in the fourth row of the array. Can you print these values one at a time (that is, one per line)? Can you print these values all at once (the whole row)? Hint: Try the slicing techniques that we used when working with lists.

- Using `np.arange()`, create a 3 by 3 array that looks like this (notice that we start with 1, not 0!):

1	2	3
4	5	6
7	8	9

`np.arange()` uses the following syntax:

```
numpy.arange(start, stop, step, dtype=None)
```

Where start and stop parameters stand for the first and the last value in range, and step specifies the step between values. By default start is 0, and step is 1

Let's call this array X. What is `np.sum(X)`? What is `np.sum(X, 0)`? What is `np.sum(X, 1)`?

7. What is `np.mean(X)`? What is `np.mean(X, 0)`? What is **`np.mean(X, 1)`**?