NENG 685 Fall 2017
PF 5 Due Oct. 17, 2017

Name:

---

On Pre-flights:

- If you work with anyone else, document what you worked on together.
- If you are not using python, then substitute your language of choice when Python is specified.

Do not write in the table to the right.

| Problem | Points | Score |
|---|---|---|
| 1 | 10 | |
| 2 | 5 | |
| 3 | 5 | |
| 4 | 7 | |
| 5 | 4 | |
| 6 | 2 | |
| 7 | 12 | |
| Total: | 45 | |

1. (a) (2 points) How many columns and how many rows are created in the array with the command:
np.zeros((4,6))

   (b) (4 points) Name 4 ways to create an array with 2 columns and 3 rows. Bonus point if you can come up with a 5th way.

   (c) (3 points) Name 3 ways to automatically create an array/list containing the following numbers (without hard coding in the numbers):
[0, 2, 4, 6, 8, 10, 12]

   (d) (1 point) How do I convert the following array to an array with 4 rows and 2 columns:
tmp1 = np.array([0, 2, 4, 6, 8, 10, 12, 14 , 16])

**Solution:**

(a) 4 rows, 6 columns

(b) np.zeros((3, 2))
np.ones((3, 2))
np.empty((3, 2))
tmp = np.array([1, 2], [3, 4], [5, 6])

(c) np.arange(0, 14, 2)
np.linspace(0, 12, 6)
range(0, 14, 2)

(d) tmp1.shape = (4,2)

2. (a) (3 points) How does the dtype differ from standard python integers and floats in terms of memory management and precision?

   (b) (2 points) What is the default dtype in an array of mixed types?

**Solution:**

(a) You can control the precision.

(b) The least precise element.

3. (a) (3 points) How would you return a slice of the array *tmp1* from Question #1d that goes from 0 to 16 counting by 4s (in numerical value, not index) and save it to a variable named tmp2?

   (b) (2 points) Now, if I set

   tmp2[1]=0

   What does tmp1 look like given the method chosen for part a?

**Solution:** One solution: b = a[0::2] a = [0, 2, 0, 6, 10, 12, 14 , 16]

Another: b = np.array(a[0::2]) a = [0, 2, 4, 6, 10, 12, 14 , 16]

Another: import copy as cp b = cp.copy(a[0::2]) a = [0, 2, 4, 6, 10, 12, 14 , 16]

4. (a) (3 points) In your own words, what are structured arrays?

   (b) (4 points) Create a structured array to store 2 HW assignments, 3 preflights, and one project for 5 people. Write code to access the 2nd pre-flight for someone by their name.

**Solution:**

(a) In most real-world data analysis scenarios, it is useful to have a notion of a table that has named columns, where each column may have its own type.

(b) Scores = np.dtype([
('Bryan', [('homework', 'float', 2), ('preflight', 'int', 3),('project', 'f8', 1)]),
('Nick', [('homework', 'f8', 2), ('preflight', 'int', 3), ('project', 'f8', 1)]),
('Amy', [('homework', 'f8', 2), ('preflight', 'int', 3), ('project', 'f8', 1)]),
('Kevin', [('homework', 'f8', 2), ('preflight', 'int', 3), ('project', 'f8', 1)]),
('Robert', [('homework', 'f8', 2), ('preflight', 'int', 3), ('project', 'f8', 1)])])

Bryan's 2nd preflight can then be accessed by: print(grades['Bryan']['preflight'][0][1])

5. (4 points) Name two ways to add the following arrays together using *built-in* methods or functions (i.e. the function or method should exist and only require x and y as inputs):

   x = np.array([1, 2]) y = np.array([3, 4])

   What are some advantages/limitations of each approach used?

**Solution:**

x + y: Slower
add(x, y): More difficult to read

6. (2 points) In your own words what are Python ufuncs?

**Solution:**

NumPy has a notion of universal functions, or ufuncs, that provide an interface for transforming arrays. Roughly speaking, a ufunc is a special callable object that implements the reduce(), reduceat(), outer(), accumulate(), and at() methods, as well as a handful of attributes. The idea behind using ufuncs is to write data transformations as generically as possible.

7. (a) (3 points) What is one concept that you found difficult in the reading?
   (b) (3 points) What about the class structure works for you?
   (c) (3 points) What about the class structure **does not** work for you?
   (d) (3 points) What is something we should be doing in class but aren't?