Classes Exercises

The “Person” Class

- Modeling a person is a classic exercise for people who are trying to learn how to write classes. We are all familiar with characteristics and behaviors of people, so it is a good exercise to try.
  - Define a *Person(object)* class.
  - In the *__init__(self)* function, define several attributes of a person. Good attributes to consider are name, age, place of birth, and anything else you like to know about the people in your life.
  - Write one method. This could be as simple as *introduce_yourself(self)*. This method would print out a statement such as: "Hello, my name is Eric."
  - Create a Person, set the attribute values appropriately, and print out information about the person.
  - Call your method on the person you created. Make sure your method executed properly; if the method does not print anything out directly, print something before and after calling the method to make sure it did what it was supposed to.

The “Sample” Class

- Modeling a sample is another classic exercise.
  - Define a *Sample(object)* class.
  - In the *__init__(self)* function, define several attributes of a sample. Some good attributes to consider are geographic coordinates (x,y), year), sample collector, or any other aspect of a sample you care to include in your class.
  - Write one method. This could be something such as *describe_sample(self)*. This method could print a series of statements that describe the sample, using the information that is stored in the attributes. Try to be creative.
  - Create a Sample object, and use your method.
  - Create several Sample objects with different values for the attributes. Use your method on several of your Sample.
  - Use the file class_glossary.pdf and try to associate the following words with pieces of your code related to the sample class: *Class, Attribute, Method, Instance, Instantation.*
  - Create a *Child class* that inherits from the class *Sample*, it could be a *SkinSample*, a *SoilSample* or a *MoonSample* for example. Also make an *instance* of that class.
  - Make sure that your code then contains method that *overrides* a parent method.

Python in style exercise

Rewrite the script presented in the file *pythoninstyle_exerciseA.py* (folder day_03/code).

Try to avoid redundancy. Focus on structure, functions, names of variable and on commenting your code.

Once you have written a clean code that returns the same thing than yesterday, try adding one option to your script that could be of use to the biologist that is looking for a pattern in the DNA sequence.

Plotting Exercise

There are many more possibilities using *pyplot* and *pandas*. If you have time left, combine the knowledge you got these last three days to read some of your own data into python using *pandas* and make it into a python plot. If you want some inspiration, look at the pyplot library which contain dozens of plots and their associated code.

http://matplotlib.org/gallery.html

Define some functions

Define some functions

Write the following functions for lists of integer:
- *coun_odd* to get the number of odd numbers in the given list.
- *get_even* to get the list of all even numbers in the given list.
- *arith_mean* which returns the arithmetic mean of values given in the lis.
- *median* that return the median of values given in the list.

Use the above functions to display the number of odd numbers, the list of even numbers,
the arithmetic mean, and the median of the numerical list entered by the user.
Bonus 1: Can you solve the problem with numpy functions?
Bonus 2: stop one of these functions with a warning if there are other types than integers in your input list.

Optional Reading Exercise

In 2015,Paul Ford wrote a brilliant narrative piece about understanding the way code works and the world the coders evolve in. It made the buzz all around the world of programmers but also in the world of science writing. It is visual, interactive, funny and approach a lot of the concepts that are hard to understand in a very easy way so have a look at it:

https://www.bloomberg.com/graphics/2015-paul-ford-what-is-code/

Optional exercise

Consider spending some time to learn more about the details of *pandas* doing the remaining tutorials.