**You will be placing all your code into the scripts.js file.**

Let's represent a relationship between two animals in our collection. Imagine that our app has a 'friendslist' on an animal's profile which lists out all of the animal's friends. Let's walk through the process together.

**Create a Friendslist**

* Create an array for the list of friends' usernames.
* Create a variable called friends and assign it to the empty data structure.
* Using your animals array, add two usernames to friends.
  + ensure that you are just putting the username value, not the entire object.
  + be careful not to use a destructive method like pop() that will remove the whole value from the animals array.
* Inspect your friends data structure. What does it look like?

**Create a Relationships object**

Imagine now that we have more than one kind of relationship in our app, we have friends and then we have romantic matches. Let's create an object to organize these different relationships!

* Create a variable called relationships assign it to an empty object.
* Add your friends data structure to the relationships object.
* Inspect your object. What is it's "length"?
* Create a variable called matches and assign it to an empty array.
* Add the matches array to the relationships object. It should look like this:
  + relationships = { friends: ['duck', 'camel'], matches: []}
* Using the relationships object, add at at least one username to matches.
  + Hint: the matches array is now nested inside the relationships object!
* Inspect your object. Is the matches array now populated with some lucky animal?
* Loop through your animals collection, adding the relationships object to each animal object. Name the propertyrelationships.
  + Note: it is ok that these are all the same relationship object.