# Exercise: Lists

### Summary

Demonstrates some of the basic operations on lists in Python.

### Instructions

Create a new script called `oath.py` that does each of the following steps. Although it's not required, you may want to include some additional print calls to describe what's being printed in places (see `demo.py` for examples).

1. Create a list called `oath` set to first few words of the Athenian Oath in lower case: "we will ever strive for the ideals". To be clear, `oath` should be a list of strings for the individual words, not a single string.
2. Print `oath`.
3. Use the `len()` to count the number of words in `oath` and then print a message that says `The length of the list is:` followed by a line with the length.
4. Use the `join()` call to put the elements from 3 to the end of `oath` together to create a new variable called `strive_str`. Remember that the first element of a list is 0 so element 3 is actually the fourth item.
5. Print a message that says `The passage is:` followed immediately by the contents of `strive_str` on the same line. You'll need to use the concatenation operator (+) to attach `strive_str` to the end of `The passage is:`.
6. Use the `append()` call to add one word, "and", to `oath` and print the updated version of the list.
7. Create a string `more_oath_str` equal to "sacred things of the city" and then use `split()` to turn that into a list called `more_oath`. Then print that out.
8. Use the `extend()` call to add `more_oath` to the end of `oath` and then print out the updated version of `oath`.
9. Build `oath_str` by joining `oath` with spaces and then print it out.
10. Use the `sorted()` call to create a new list called `sorted_oath` that is sorted in alphabetical order. Then print `sorted_oath`.
11. Create a list called `yoda` that consists of one word: `["code"]`. Then extend it with the last three words of `sorted_oath`.
12. Finally, join `yoda` with spaces and print the result out. If all has gone well it should look like a statement about the course.

If you aren't sure how to do some of these things, have a look at `demo.py` for some examples.

### Submitting

Once you're happy with everything and have committed all of the changes to your local repository, please push the changes to GitHub. At that point, you're done: you have submitted your answer.

### Tips

- You'll probably find you use `split()` and `join()` a lot: they're very useful for dealing with text.
- A handy way to get rid of extra spaces in input text (say two or more spaces when you'd really like to have exactly one) is to split the string and then join the words back together.