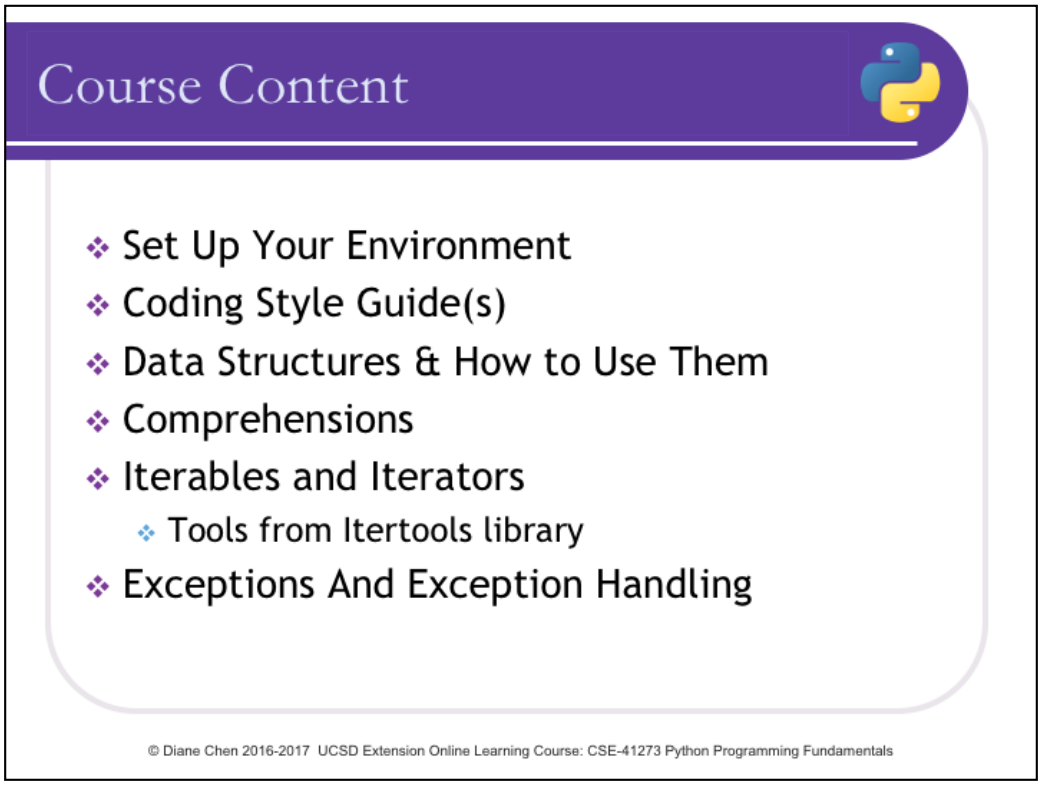
Python Programming Fundamentals

- Expand your knowledge of Python in this class!
- Become a Pythonista!

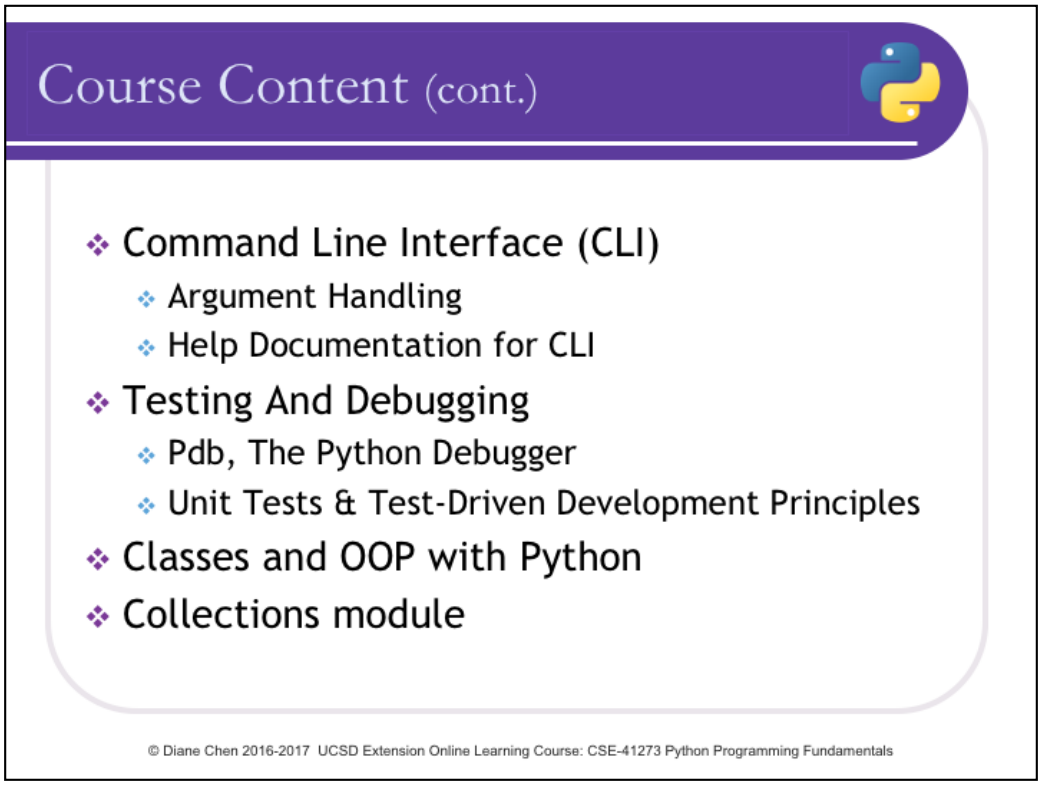
Hello everyone! My name is Diane Chen and I'll be your instructor for the Python Programming Fundamentals class. In this class, you will expand your knowledge of the Python programming language. I hope you will learn to love Python as much as I do, and that after this class, you will continue to learn more about Python and become a Pythonista!



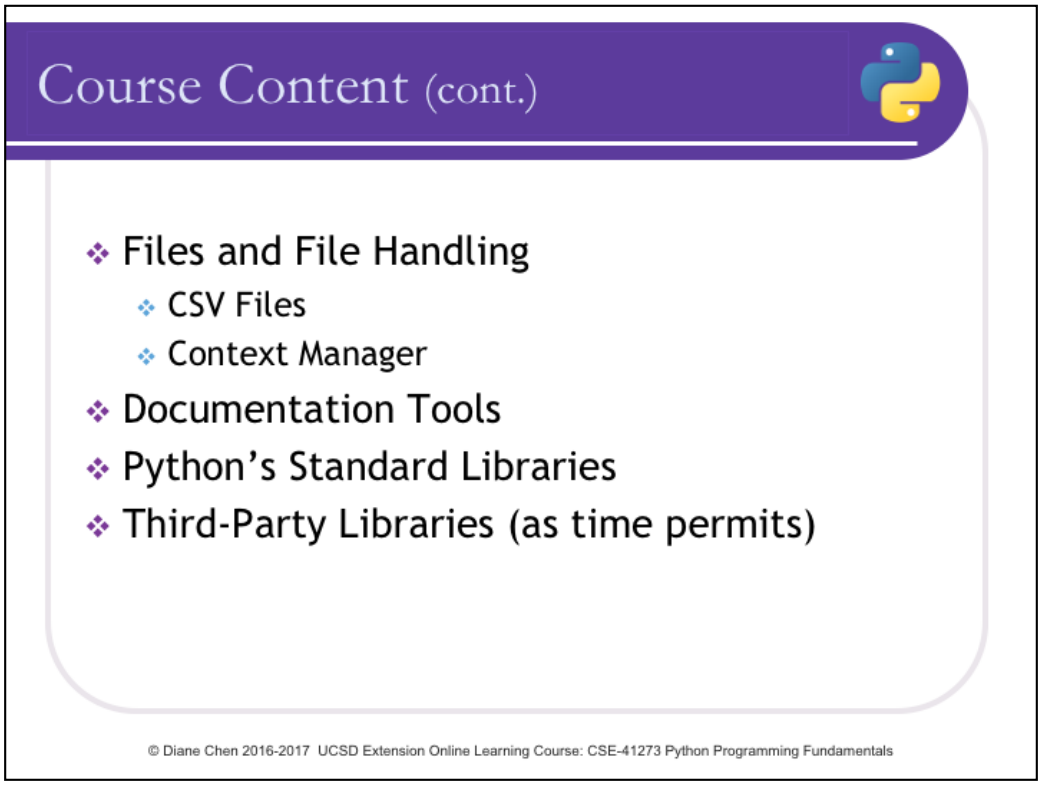
Let's talk about some of the things this course will cover – not necessarily in this order. I'm going to talk about setting up your environment. This involves installing Python and setting up what is known as a virtual environment. Some people say virtual-e-n-v and some people say virtual-env when they are referring to the virtual environment in Python. When you are working with Python, you usually have multiple projects that you are working on, and you will want each project to have its own virtual environment. You'll only need one for this class, but if you had several projects, some projects might need different libraries installed or even different versions of Python. You do this with virtual environments, so that changes to one project do not affect other projects.

We'll talk about coding style and documentation. These are the guidelines to follow to produce code that is readable. Python has many great data structures for you to use; I want to make sure you understand them all and how to make the best use of them.

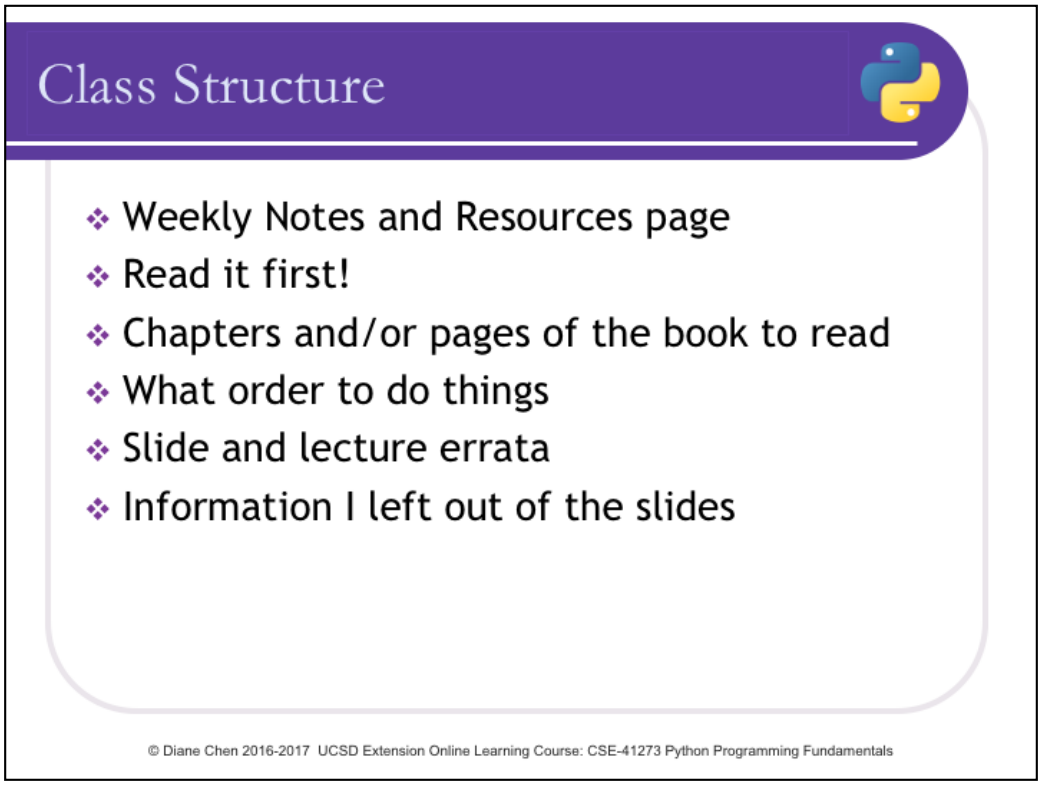
Comprehensions are very important in Python and an exceptional tool to use. You may already know list comprehensions; we will talk about other kinds of comprehensions. Iterables and iterators are another very important concept in Python. We will cover what they are and how to use them. We will also discuss some of the special tools available in the Itertools library, one of the many standard libraries in Python. We will talk about exceptions and exception handling. Exceptions are when errors occur in Python, both unexpected and also expected; we will cover how to deal with them.



We'll be talking about the command line interface, often referred to as just CLI, surprisingly enough. The command line interface is what happens when we are running a Python program from the command line - I'll show you how to handle arguments passed on the command line. We'll also cover how to include help documentation for a script that runs on the command line. Testing and debugging is another important subject. Pdb is the Python debugger; I'll talk about how to use it. More importantly, we'll cover making unit tests and the concepts of test-driven development. Classes and object-oriented programming in Python is not as important as, for example in Java, but are concepts you should understand, and we will cover them. I'll also cover some of the more useful special data structures found in the Collections module.



Files and file handling comprise another topic we will cover, including a standard library for reading CSV files that makes life easier when dealing with them. We'll also cover using a context manager with our file handling. The context manager ensures that the file is always closed, even if an error occurs while processing the file. On some systems, not closing a file could leave it locked so that it can't be opened again without hassle. Documentation tools are important to know. If you write a python package that you want to submit to the repository so others can use it, you want to have good documentation. If people can't figure out how to use the package, they aren't going to use it. Python has many standard libraries that you should be aware of. We will cover a number of the more important and useful ones. There are an amazing number of third-party libraries for Python - pretty much anything you need to do, there is a library that will help you. I will cover how to install third-party libraries, and some of the more interesting ones as time permits.

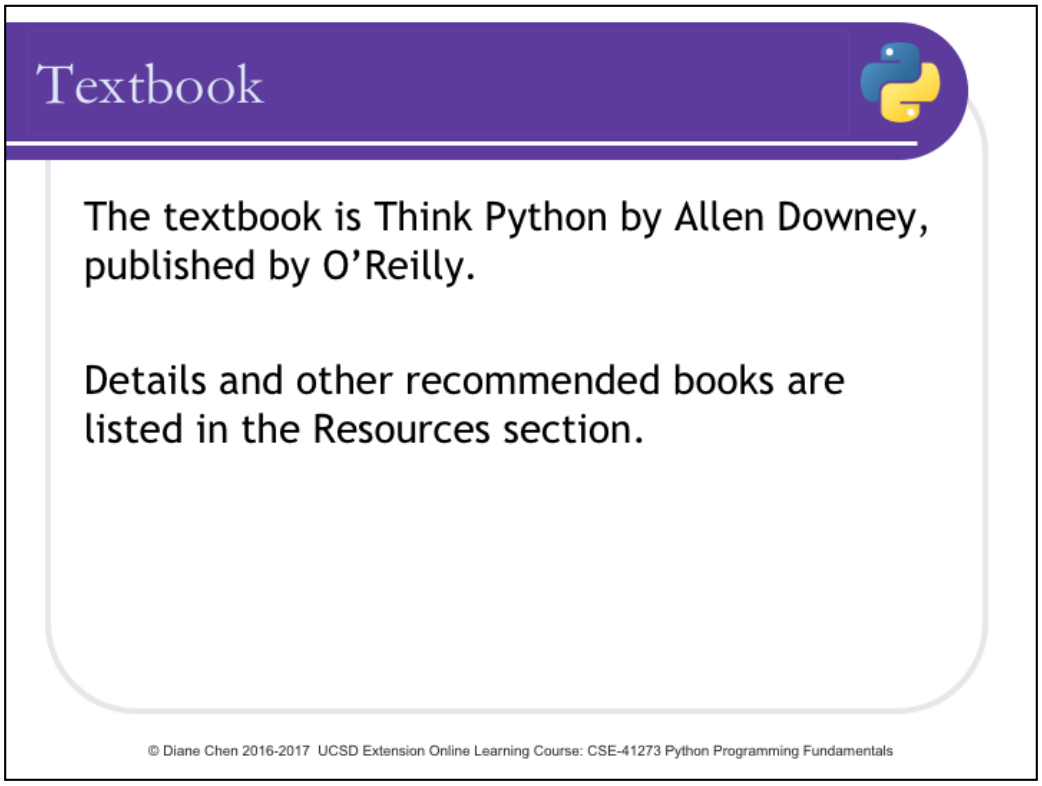


I have tried to structure the class so that it is easy to follow. Each week's Lessons section will have a page called Notes and Resources.

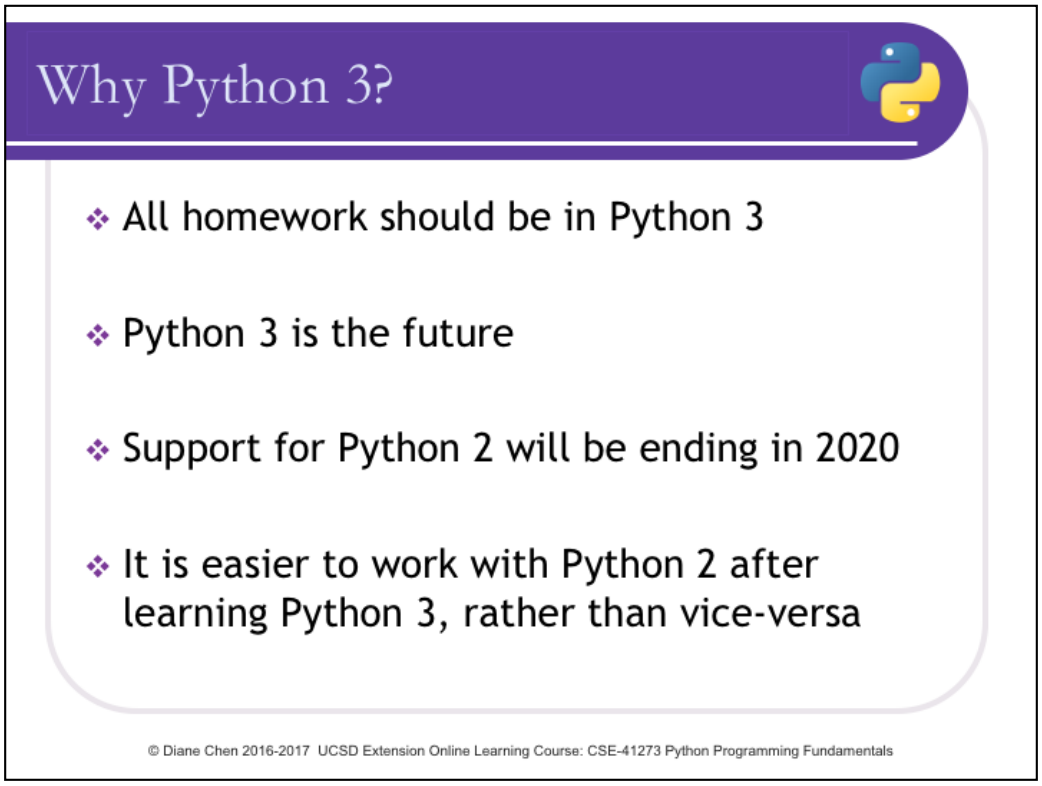
Be sure to read it first thing as it will be your guide for the week. I will have the chapters and/or pages of the textbook to study for the week. If there is homework or a quiz, it will be listed with the subject matter.

Information about what order to do things in will be there too, if relevant. Sometimes I may want you to read some things before watching the lecture videos, other times not. If there are errors in the slides or lecture that I don't have time to fix before the class starts, I'll put that in there too.

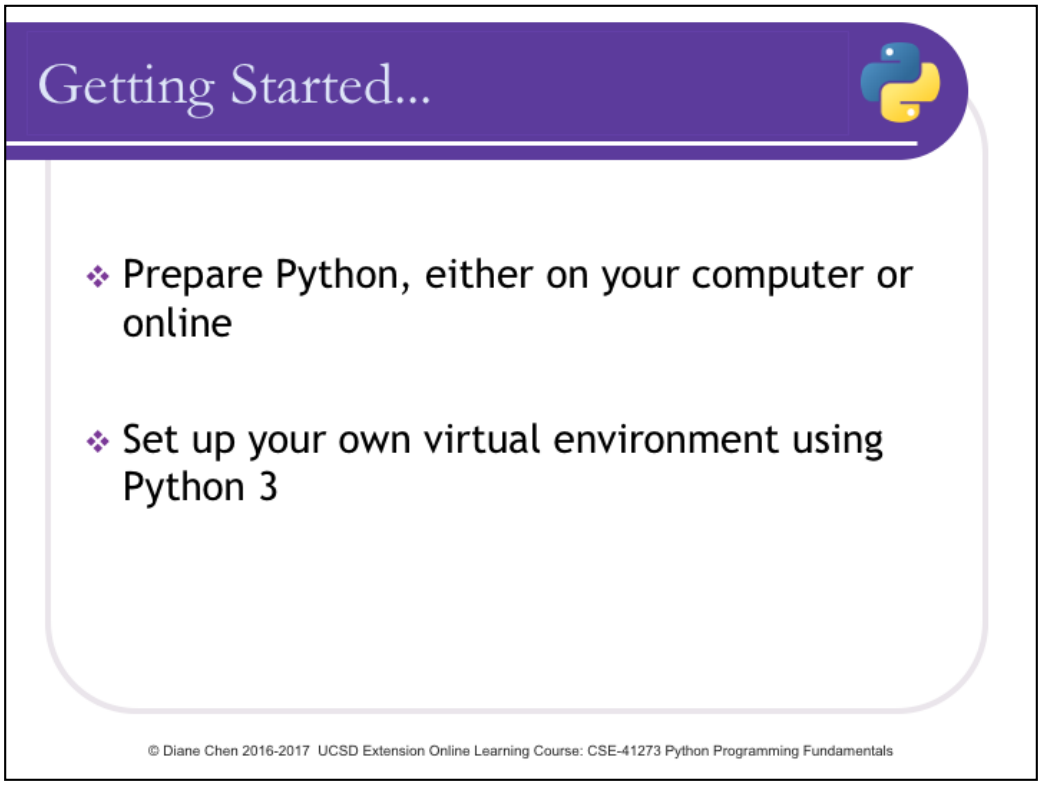
And, since I'm still fairly new to teaching the class and I keep changing things around, I will probably find at the last minute that I left out this or that piece of information from the lecture – anything like that will also be in this file. I will also have additional resources for you to look at to help your understanding of the week's subject matter. These would be things like links to blog posts, other helpful websites or videos. There's lots of good information on the internet, the problem is finding the good stuff in the midst of all the other stuff!



A word about the textbook. It wasn't easy to decide on a textbook for this class. The textbook I chose is called Think Python by Allen Downey. Although it is a very basic Python book for new programmers, I like it a lot for its clear explanations. I especially like his approach to solving problems. I didn't want to require more than one book and there really isn't any other book I like that contains enough of what I feel is important for this class to make me want to require it. I'm not following the book exactly in order and we will go into many subjects not included in the book, and some of the books' subjects in more detail, but you will find it very useful. You can access the book on the Internet or, better yet, get one for yourself. Details and other recommended books are listed in the Notes and Resources section for week 1. I do encourage you to look at other resources.



A few words about why we are using Python 3. You may already be aware that there are two primary versions of Python, Python 2 and Python 3. Python 2 is legacy software, and a lot of software still exists that uses it. However, we will be using Python 3 for this class. Some of the difference are obvious, and some are more subtle. I will try to remember to talk about them as they come up in the subject matter. Python 3 is the future. Cool new things are added at each new point release that make a programmer's life easier. Official support for Python 2 will be ending in 2020, which is not that far off! I also believe that it is easier to learn Python 3 and move to Python 2, should you have to, than it is to learn in Python 2 and move to Python 3. Once you set up your virtual environment, and always remember to use it, you will be working in Python 3, and I expect all the homework to be written in Python 3.



To start with, first set up Python on your computer.

In the notes for week 1, I have 2 documents, one about installing Python, and one about creating the virtual environment.

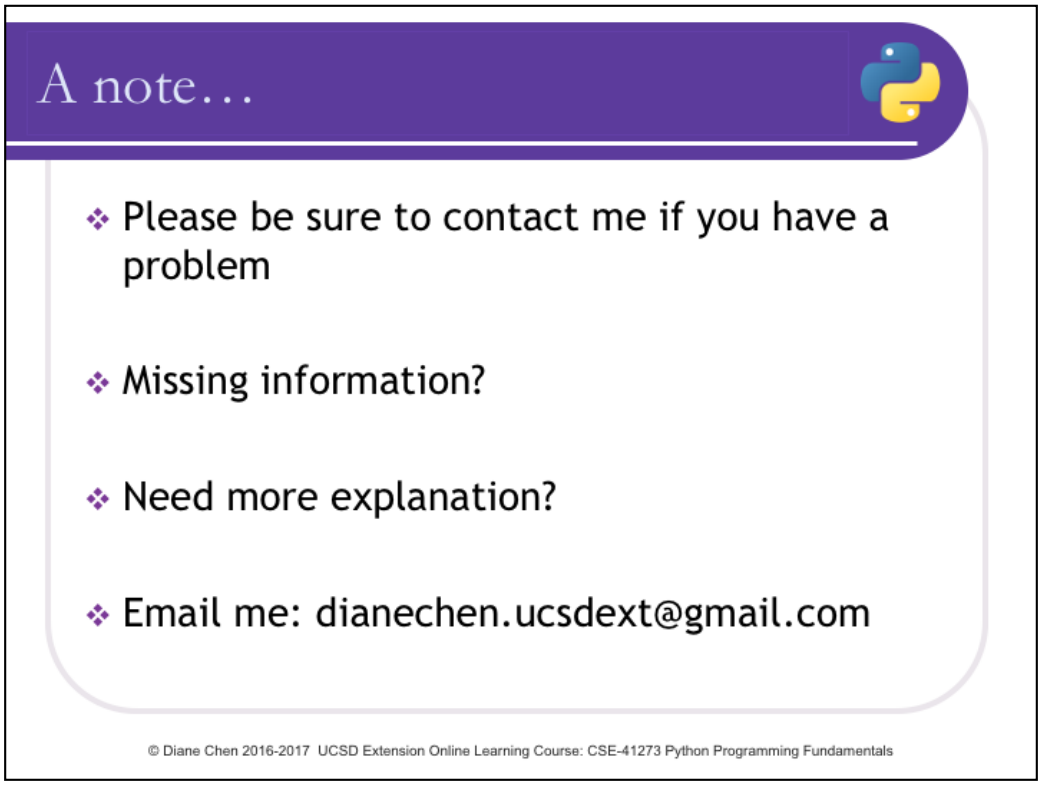
If you already have Python, you may need to upgrade; I want everyone to be working in at least Python 3.5, if not 3.6. Note that it is possible to have multiple versions installed together, so if your computer already has Python 2, it is OK to also install Python 3, and you can also have different point release versions. For example, my computer is a Mac, and it came with Python 2. I also have Python 3.4, 3.5 and 3.6 installed together.

If you don't want to install Python or upgrade your Python version, or if you want to use a Chromebook, you can use an online service. In the installation notes, I have instructions on setting up a free account on PythonAnywhere.com. As we are not doing graphics programs in this class, you will be able to do all of your work at PythonAnywhere.com if you prefer.

The instructions for setting up your virtual environment include setting it up on your computer or on PythonAnywhere.com.

Be sure to get all this set up done as soon as possible so you will be able to begin working right away. If you have problems with installation or the virtual environment, please post in the Discussion board or email me.

Don't forget to always use the virtual environment whenever you are doing anything for the class.



As a side note, if at any time during the class, you have a problem with the class materials, or questions about what I've covered, please contact me. Is there information I said I would provide and you can't find it? Have I assumed you know something that makes no sense to you? Do you just need more explanation on something? Please tell me. You can post on the discussion board or email me at dianechen.ucsdext@gmail.com to let me know. I can't fix a problem I don't know about. I will have some office hours posted where I will monitor the email and discussion boards closely and get back to you quickly. At other times, I will monitor the boards and my email regularly and make every effort to get back to you within 24 hours - usually it is much faster. With that said, it's onwards and upwards towards new heights of learning with Python!