

Introduction to Python

How to Study

1. Get an overview of what we're doing this week: the **flashcards**, **slides** and optional **videos**.
 - It's probably easiest to flip through and read the **slides** for the week.
 - Complete 30 minutes of **flashcards**.
 - If desired, you may also use the **content videos** (linked at top of page) or **inclass videos**.
 - The purpose is to get a sense of what we're doing this week and why.
2. Work with individual features in isolation: your **scratch file** and the **non-lab** exercises.
 - Visit **pythonreference.com** and click on a **Session** link. These are that week's features.
 - Your goal is to observe the **type** and **value** that is needed for the **argument(s)** and **return value** of each feature. Examples must be *coded and run*, not just mentally acknowledged.
 - You may use a test **.py** file to to play around with each feature.¹
 - Or, go through the non-lab exercises, as these also review each feature. Make sure to have the exercise solutions available - don't ever stress a solution - look at it, then code it again.
3. Work with the features in combination: the **_lab.py** exercises.²
 - After trying out each feature at least once, attempt the **_lab.py** inclass exercises.
 - These are more like puzzles, asking you to apply multiple features to solve a problem.
 - Again, keep the solutions at hand -- no need to stress any solution. Look at it, then try to code it again.
4. Complete the homework: the **homework** and **homework discussion** documents.
 - The homework problems fashion the features we've learned into a practical project.
 - The homework discussion should be consulted if you feel stuck or not sure where to begin.
5. Always pay attention to **exceptions**, and the **type and value of your variables**.
 - When you get an exception, ask *why did Python have to stop on this line?*
 - When things go wrong but the exception isn't helpful, ask *are the types and syntax correct?*

¹ Use the feature the way it's shown in the example, then "mess around" with it a bit. See what happens when you try a different value, or different type of object. How does Python react? Remember you are also developing a working relationship with the Python interpreter, this means that you're making mental note of (though never expected to memorize) how Python responds to different things you might try (for example, use **len()** to get the length of a string, then try to get the length of an int -- see what happens).

² The **_lab.py** exercises are designed to allow you to put the week's features into action. They ask you to solve a problem by using one or more of the features discussed up to that point. This is the start of your being able to construct a practical program -- saying "I want to get from here to there -- what features do I use, and in what order?"