# T12: Modules

* Assignment T12 should be completed in a team of two.
* To begin, go to “File” and Select “Make a Copy...”
* Move your copy of the document to the folder for your section.
* Rename the file to **T12: Modules - usernames** (replace usernames with your usernames). To do this, click the label in the top left corner of your browser.

## Learning Objectives

* Learn how to work with modules in Python
* Learn more about the relationship between Python files

## Modules

Modules provide us with access to extremely powerful Python libraries. Some of these libraries can be incredibly complex, such as [Numpy](http://www.numpy.org/) (the math library, but on serious steroids). Others, like the Turtles library, allow us to do “simple” things like draw on the screen. The whole reason we are using Anaconda, which we haven’t touched much (yet), is so we can gain access to these modules easily, when we need them. Even better, we can build our own modules to solve our own, specific problems.

Another value of modules is the ability to hide code that isn’t important to you at that very moment. For example, while we’re writing code, it’s nice to have the test suite right there to see and use. But once we’re done testing all of our individual functions, and we’re ready to write the final main() function that puts all the pieces (i.e., functions) together to solve the big picture problem, we want that test suite out-of-sight, out-of-mind. A useful way of preserving that code for later use (like when you need to make a modification to the code in five years) is to put it in it’s own file, and import it as a module.

Let’s go back to A6: It’s in your Genes. This was the largest code you’ve seen in the class, largely because of the sizeable test suite. Let’s move it out and clean up our code a little. Here are the two new files:

* [a6\_genes\_solved.py](https://drive.google.com/open?id=0B0J8Yj0B6KRScHNsd0N2ZXhyTk0)
* [a6\_genes\_test.py](https://drive.google.com/open?id=0B0J8Yj0B6KRScXpST0hXTzFUbVE)

Read through the code, noticing the differences related to the test suite.

|  |  |
| --- | --- |
| How did I get the functions defined in **a6\_genes\_solved.py** into the file **a6\_genes\_test.py**. In other words, why can the test suite call functions like is\_nucleotide()? | 1. |
| Does the a6\_genes\_solved file still work fine as a stand-alone program? Does the test suite still work in testing each function? | 2. |
| The test suite used to be called in the main() function. Where is it called now? | 3. |
| What does this code do?  **if** \_\_name\_\_ == **"\_\_main\_\_"**:  main()  HINT: Remove it and run the two files again. What changed? | 4. |
| At the moment, does the seem more confusing that the previous way things were done? Or less confusing? Describe why/why not here.  If you are currently finding it to be more confusing, discuss it with your partner, your tablemates across the way, or the instructor and TA’s. The goal here is to clarify, not further confuse. Don’t move on until you’re comfortable with the idea of code living in different files. | 5. |

Now, I want you to create a new module, similar to what we’ve done in above. I’ve provided you with a very poorly commented piece of code [t12\_dots.py](https://drive.google.com/file/d/0B0J8Yj0B6KRSQXR4RFVIQzBqUEE/view?usp=sharing). I would like you to:

1. Understand the code by running it a few times.
2. Write the docstrings and any other comments you think will help clarify the code.
3. Verify/Unverify the code is correct in solving the intended problem. You will do this by creating a test suite to test this code. You should do this as a separate module. You only need to build test cases for functions that return values.
4. Your test suite module must include:
   1. An import statement to include the main code
   2. A testit() function, which can be reused from previous assignments
   3. A test\_suite() function, which tests any fruitful functions
   4. A call to the test\_suite() at the main level of the test suite file
5. Your main code should not be altered, except for the docstrings and comments you add.

## Submission Instructions

1. (Submitter) Save your codes as **T12\_dots\_usernames.py and T12\_dots\_test\_usernames.py**. Replace usernames with your Berea usernames.   
   NOTE: Incorrect filenames will automatically reduce your grade by 1 point. Fortunately, the format is always the same no matter what the assignment.
2. Zip the two Python files together.
3. (Submitter) Upload the Python file to Moodle by the due date listed on the course website: <https://trello.com/b/w7bIrLoV/>.
4. (All Other Partners) Open up Wordpad. Create a new text document (.txt) and include all members names in it.
5. (All Other Partners) Save the document as **T12\_dots\_usernames.txt**. Replace usernames with your Berea usernames.
6. (All Other Partners) Upload the document to Moodle by the due date listed on the course website: <https://trello.com/b/w7bIrLoV/>.