

Learning Module: Tech Competency

Due: Friday, January 16, 2015 at 11:59pm

Description: This Learning Module is intended to help you establish a baseline level of competency in the two main computing environments for MTH 325: the **Sage** scientific programming software and **SageMath Cloud**, a cloud-based platform for computing with Sage and other tools. You will also learn a small portion of **Markdown**, a language for doing simple text formatting.

SPECIAL OFFER: Students who submit this module before Wednesday, January 14, 2015 at 11:59pm and receive a Pass mark on it will receive one additional token for the semester. Remember, each token entitles you to a do-over on a Learning Module, a 24-hour extension, or a makeup of a Timed Module session.

Assessment process: Please remember that all Learning Modules are graded either **Pass** or **No Pass** based on whether your work satisfies the specifications discussed in class. These specifications are housed on the course Blackboard site. **It is your responsibility to review the specifications and check to make sure your work meets all of them before submitting your work. Please use the checklists provided on Blackboard as a guide.**

Submission: For this particular module, you are not going to submit a file, but instead you'll be creating a project in SageMath Cloud and a folder that will store your work. So there is nothing to submit by email this time, but your work will need to be in place by the deadline in order for it to be graded. Instructions for creating this folder and putting work into it will be given below.

Academic honesty notice: All specific implementations of those ideas and all specific work you do must be your own and not contain anyone else's work. Failure to abide by university academic honesty policies may result in failure not only of the module but of the entire course.

Module Activities

1. **SageMath Cloud.** SageMath Cloud (SMC) will be our primary platform for doing computational work in MTH 325. You will get a brief demo of SMC in class during the first week of the semester. For this module: Create a project in SMC that has the title:

`LastName #mth325 #secno`

where `LastName` is your LastName and `#secno` is your section number, either 01 or 02. The hashtags should be in the title of the project. For example, if Jane Brown is in section 02, her project would be titled `Brown #mth325 #02`. Once you have created the project, go into the **Settings** for the project and add me (email talbertr@gvsu.edu.) as a collaborator. Then, create a folder within that project called **Tech Competency Learning Module**.

2. **Sage.** Sage is a free, open-source computer algebra system and scientific computing platform. It is built on the Python programming language, and you can program with it using Python syntax and data structures. But it also extends Python with a vast set of libraries for performing mathematical computations, including (especially) symbolic computation. One works with Sage through documents called *notebooks*. In MTH 325, we will work with Sage notebooks both online using SageMath Cloud, and offline using local installations of Sage. Before going any further, do the following:

- In SageMath Cloud, create a new Sage notebook. This is just for scratch work, so you can call it whatever you want and put it wherever you want.
- Go to the website: <http://www-rohan.sdsu.edu/~mosulliv/Teaching/sdsu-sage-tutorial/index.html> and work through the following sections in the tutorial by typing along with the text in your Sage notebook:
 - **About this tutorial** (the entire section)
 - **Sage as a calculator**, just the subsection “Arithmetic and Functions” (stop when you get to “Solving Equations and Inequalities”)
 - **Programming in Sage**, the subsection **Sage Objects** starting with “Booleans” and continuing through the end. (You can also skip the first two code blocks under “Variables”.)
 - **Programming in Sage**, the subsection **Programming Tools** except for the final part “External Files and Sessions”.

For more help with some of the programming constructions in the last section, please consult the tutorial *Think Like a Computer Scientist* at <http://interactivepython.org/runestone/static/thinkcspy/toc.html#t-o-c>.

Once you’ve gained some experience working with Sage, create a new Sage notebook called

`LastName Sage Competency`

where `LastName` is your last name. *Make sure to put this file in your Tech Competency Learning Module folder that you made earlier.* In this file, do the following:

- Take your birthdate in day/month/year format and concatenate the day and month together. you now have two integers of 3–4 digits each. For example, my birth date is July 10, 1970 which is 07/10/1970 which becomes the two integers 710 and 1970. Call the day/month integer d and the year y . In Sage, calculate: $d + y - (dy)$; d^y ; $\lceil y/d \rceil$; the greatest common divisor of d and y ; the divisors of y ; and the natural logarithm of d .
- Use Sage to convert the integer d from the previous exercise to its binary representation. (Hint: Try typing the integer followed by a period, then try tab-completion to see if there’s a method that might be useful. Tab completion is discussed in the tutorial.)
- Write a **for** loop that iterates through the numbers 2, 3, 4, ..., 100 and prints off a list containing the prime factors of each of those numbers one at a time. (Hint: Look for a method that will produce the prime factors of an integer, discussed in the tutorial.)
- Write a **while** loop that does the same thing as the **for** loop in the previous exercise.
- Take one of the loops from the previous two exercises and modify it so that it prints off the *number* of prime factors for each of the integers 2, 3, 4, ..., 100.
- Write a function (using the **def** keyword) called **hypotenuse** that takes two numbers **a** and **b** and returns the value $\sqrt{a^2 + b^2}$. This function should take two lines of code: One line for the **def** statement and a second line to compute and return the value. If you’re lazy you can do it in three lines.
- Create a new function called **hypotenuse2** that takes two numbers **a** and **b** and returns the value $\sqrt{a^2 + b^2}$ if both a and b are not equal to 0, and returns -1 otherwise.
- Create a new function called **hypotenuse3** that takes two numbers **a** and **b** and returns the value 0 if $a = b$; returns -1 if $a \neq b$ but either a or b is negative; and returns $\sqrt{a^2 + b^2}$ otherwise.

3. **Basic Markdown.** Markdown is a simple, lightweight text formatting language that is somewhat recent but which has become an industry standard for writing formatted text documents, especially code documentation. A Markdown document is just a simple plain text file, saved with the extension `.md`, in which text is modified using simple notation to produce formatting. There are several tutorials on Markdown available online. Use this one for now:

<https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>

Read through this webpage and then do the following.

- (a) Write a Markdown file that reproduces the following:

This is basic markdown practice

Why MTH 325 is awesome

MTH 325 is awesome for *many* reasons:

- It has math in it
- It uses Sage
- It has **lots** of math in it

Two important websites for the class

1. [SageMath Cloud](http://cloud.sagemath.com)
2. [Sage Cell Server](http://sagecell.sagemath.org) which is like a place for throwaway Sage calculations.

Notes: The top line is an “H1” header; the two other headers are “H2”; the word “lots” is boldface; the two links are to <http://cloud.sagemath.com> and <http://sagecell.sagemath.org> respectively. You can type this Markdown document in any text editor you like; you can preview your results using the website <http://markdownlivepreview.com/>. Once you have gotten the Markdown file looking right, save it as `Markdown practice.md` and upload it to the Tech Competency Learning Module folder in SMC.

- (b) You can add Markdown into any Sage notebook by starting a new line with the command `%md` and then writing in Markdown starting on the following line. Once you hit shift-enter, the Markdown will render into formatted text. Go back to the Sage notebook that you created in the first set of exercises and add a Markdown cell at the top of the notebook that has your name in an H2 header, and then write a paragraph of text that includes italics and boldface. (To add a new cell at a particular location in a Sage notebook, click on the gray line separating two cells.)
4. **Reflection on your work.** Create a new Markdown file called `Reflection.md` and in that document, write at least 500 words to discuss:
- A brief description of the timeline for your work on this module, including the date you started, the date you finished, the approximate amount of time spent, and any other details.
 - One or two new things you learned in this module that were particularly interesting.
 - One or two things you found particularly challenging about this module.
 - Any questions, concerns, comments, etc. you have about this module, your work on it, or what you learned in it.

Put this file into your Tech Competency Learning Module folder when done. (You can either write the file offline and then upload it, or just do the entire thing in SageMath Cloud. However, you cannot preview Markdown files in SageMath Cloud.)