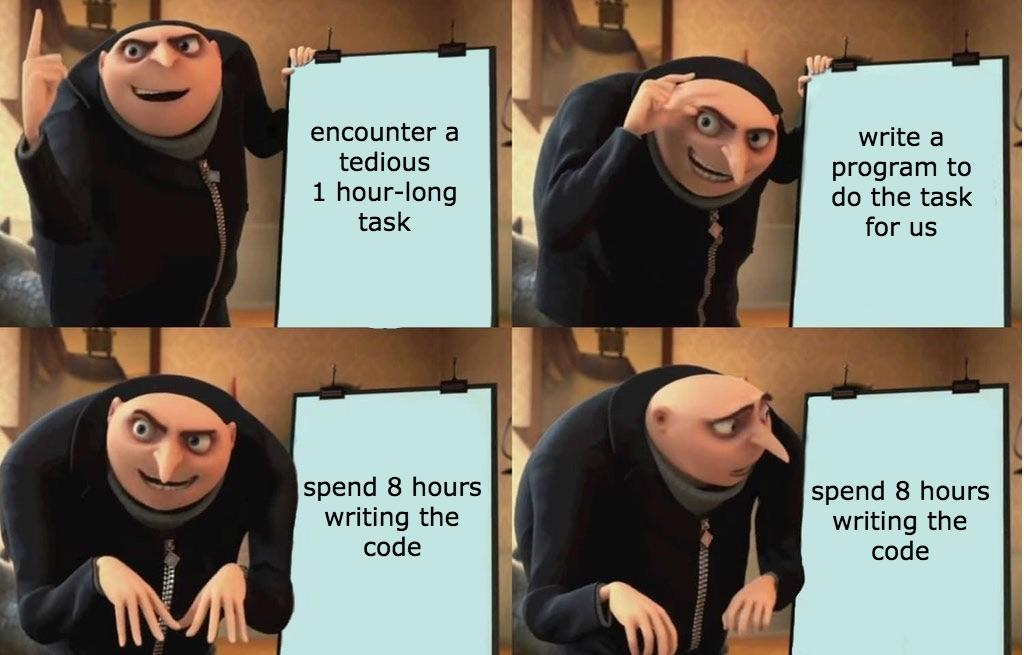
**RTC Python Workshop 2020: Link To Resources**

**Thank you:** Thank you for attending the workshop and for filling the feedback form! Don’t hesitate to reach out if you would like further advice on how/where to begin your programming journey! The learning curve can be steep but there are a lot of resources out there to help and the experience is really rewarding! Best, ~Anisha

**My tips and recommendations:**

[](https://www.reddit.com/r/ProgrammerHumor/comments/83vrwb/gru_automatizes_tasks/)

^ This is how I learned how to code

* **Do it.** My main recommendation is to start doing it. Don’t wait until you think you’ll know enough. I definitely know enough and I taught this workshop.
* **Teach.** While creating and teaching workshops, I often learn things I didn’t know before. So I think teaching others in your lab or other people who want to learn can be useful for you and for them.
* **Google.** The answer is almost always out there. Copy-paste your errors after making them a bit more generic, search using keywords like “mean column pandas dataframe python”
* **On courses.** Taking courses can be useful but don’t do just that. More than anything else, programming requires hands-on experience
* **It’s hard.** It is hard and frustrating to learn how to code. Expect this when you are learning and don’t be discouraged if this is something you really want to learn (but do take breaks from it).
* **Here is a secret.** Once you learn one programming language, learning others can often be much easier because most languages use the same basic principles. You’ll have to google a bunch to learn how to do a specific function in the new language but you’ll have the nuts and bolts.

Other resources for beginners:

* [Python Data Science Handbook by Jake VanderPlas](https://jakevdp.github.io/PythonDataScienceHandbook/)
* [Google’s Python Class](https://developers.google.com/edu/python)
* [Youtube video](https://www.youtube.com/watch?v=rfscVS0vtbw) (It’s 5 hours long. I haven’t watched the whole thing but it looks good!)
* [Tutorial by Python](https://docs.python.org/3/tutorial/)
* [General tips](https://realpython.com/python-beginner-tips/) if you are new to programming

Ok, but I don’t want to use jupyter notebooks forever…

* [How to install Python on your computer](https://www.codecademy.com/articles/install-python-data-analysis)
* Highly recommend using [Spyder](https://www.spyder-ide.org/)--it makes it so easy to see what’s going on and makes Python interactive

Other good practices while programming:

* [GitHub](https://guides.github.com/activities/hello-world/)
* [On installing Python](https://www.codecademy.com/articles/install-python-data-analysis) (see section on miniconda and anaconda)

From the workshop:

* [Debugging](https://jakevdp.github.io/PythonDataScienceHandbook/01.06-errors-and-debugging.html)
* Seaborn Links

<https://seaborn.pydata.org/>

<https://seaborn.pydata.org/tutorial.html>

<https://seaborn.pydata.org/examples/index.html>

* Seaborn is based on matplotlib: <https://matplotlib.org/gallery.html>
* Pandas:

<https://pandas.pydata.org/pandas-docs/stable/getting_started/10min.html#min>

<https://pandas.pydata.org/pandas-docs/stable/getting_started/tutorials.html>

[Psychopy](https://www.psychopy.org/) (used to create experiments with Python--this is what [Pavlovia](https://pavlovia.org/) hosts):

<https://www.socsci.ru.nl/wilberth/nocms/psychopy/print.php>

<http://www.djmannion.net/psych_programming/vision/intro/intro.html>