Liam's Lab Data Science Course Syllabus
Summer 2021, May 28 – Current

Lecture: Saturdays, 10 AM – 1/2 PM @ secret Lab Hideout
Office Hours: Tuesday nights, 8 PM – 9 PM
Homework: Weekly, due Thursday 12:00 AM

Course objective: data analyst (whatever that means…) skills.

Author's note:

The course is taught in a way that puts creativity first. Instead of just listing out the results of arithmetic operators in python, for example, the course will ask students to critically think about how to check if a number is even, then reveal the use of "%", "==", and if statements. This leads to a messy Table of Contents, since things are uncovered in a non-linear way, but the main use of a ToC is to keep track of everything we've done so far. This renders the course inaccessible on the face of it to new-comers, but I'm working on building a nice, Khan Academy-inspired online website!

I've also realized this "Introduction to Programming" part of the course covers what would be likely the first chapter of a Data Structures & Algorithms book (in fact, from the sources of inspiration I've looked at, it is!). So, why does it take so long at Liam's Lab to learn the basics? Let's get rid of this anxiety around learning as fast as possible. Like when learning the basics of another language, the first week – all the syllables and pronunciation – is the hardest to master (not even to a native extent, as that's impossible, just to really understand what's going on). I think it makes sense in some language classes to then just make the first week an introduction and continuously, throughout the years, reinforce better pronunciation.

In programming, though, I think we diverge a little bit from language, and end up a little bit more like math. In math, there's no way you can just brush off Algebra 2 and move onto Calculus, since Calculus uses Algebra. In the same vein, with programming you can't just brush off nested loops, then teach Bubble Sort, because Bubble Sort uses nested loops. I think a lot of people **do** kind of "keep going" even if it doesn't make sense and say they eventually "get it" over time, but I think that's a false narrative. I don't think just throwing someone into the deep end teaches them to swim. Most of the time, you will go back to the shallow end, re-learn the basics, then try again. There's nothing wrong with that and this is how I learned – I'm just curious to try out a different approach. Not to mention, this isn't how I learned math or how I learned language and I think I just ended up doing this in programming because my educational structure wasn't all there. Even now, after years and years of Mandarin or piano, any Master will tell you say simple things slowly, and the hard stuff will become easy. That's why we take a long time to go over the basics, the simple stuff. This way, we master our "form", how we relate to Python itself, and cultivate a creative mind, one that's comfortable with trying out new things since new things can actively be imagined.

Liam, Lab Member 00
July 5, 2021

Table of Contents

Recursion 07/03
- Factorials, chop by letter
- Fibonacci
- Drawbacks

Nested loops 07/03

Sequence operators with strings 07/03
- Largest palindromic product of two 3-digit numbers

Big-Oh Notation 07/03
- What's the time complexity of a nested for loop?
- Why are dictionaries O(1)?
- Two-Sum: $O(n^2)$ and O(1) solutions