This is how you’re going to learn to code- you’re in for a brutal treat but it’ll pay off :3

I didn’t know how functions worked when I first started

btw➜ @lachinemearning

Okay lets get one thing straight, AI is NOT suddenly replacing software engineers overnight. People who tell you that there is no point in going into computer science because the AI warlords are developing quicker than an embryo everyday, are exaggerating the capabilities of AI lol

AI is a mere tool. Software developers who had to write redundant code by hand and spend hours trying to clean code because they’re under so much tech debt from their startup days- now can give that laborious work to AI so they can start spending their time being pulled into useless daily standups.

I’m joking…well half….the point is, learning how to code is still necessary.

Imagine you ask AI: “write me a doom game copy but in asm x86 instead of C” and you have not touched a line of x86 assembly in your life, what do you think is going to happen when AI gives you code that is broken?

You will have 0 idea where it's going wrong, then you’ll make the rookie mistake of sending it back to AI for it to hallucinate more. It’s the blind leading the blind.

When you know how to code, you A. know how to prompt AI to speed up the work for you, B. you know if what AI is giving you is complete bogus, and C. if AI falls, if you run out of credits, you won’t be hindered. Companies still value capable programmers.

Someone who knows how to code and use AI to their advantage **>** someone who rejects AI completely/doesn't know how to code and uses AI blindly.

And actually, a lot of the time, the mistakes are so simple that ping ponging back and forth with an LLM is **MORE** work than looking at the code and realising you instantiated your classes in the wrong order. But the LLM is soooooo task driven it doesn't realise there's additional context to how you want your task even carried out.

Picking a programming language to learn

Before you even choose a language, you need to figure out **why** you want to get into programming. **Computer science** is such a **huge** field, that different **sectors** use **different** languages.

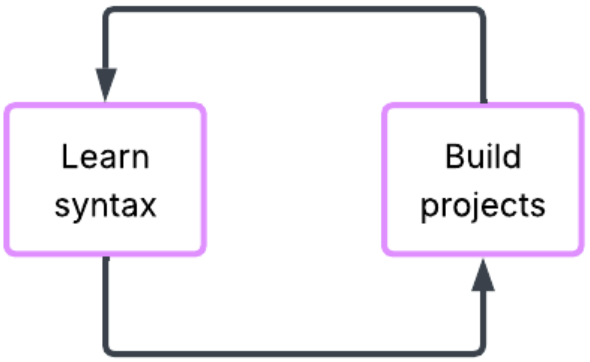
There is *no* **bad** programming language, there are just **optimal** programming languages *fit* for certain tasks.

And *working* in **machine learning** and **data science**, you won't use C to build neural networks and make LLM api calls- that just isn't realistic to the industry. You will most likely be using **python** due to its sheer **library support** and quick **prototyping** speed.

Don't let someone ego you into saying rust is the best language btw. Those are the same people who probably don't even know what langgraph is, so don't listen to them.

The same way we wouldn't build low level software in python, you wouldn't build high level machine learning pipelines in C. All languages are suited to a specific purpose ok?

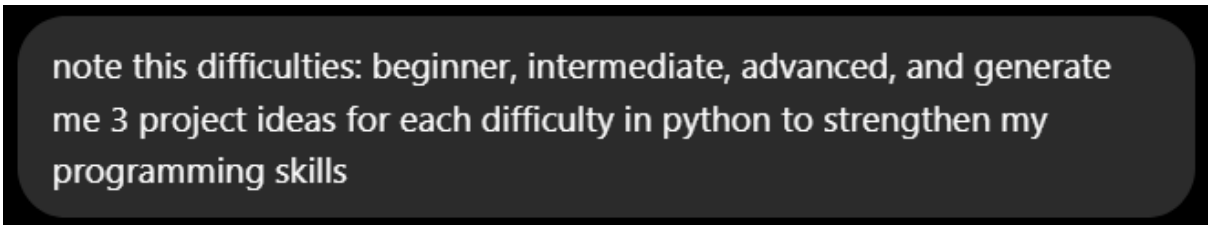
The programmer's learning loop



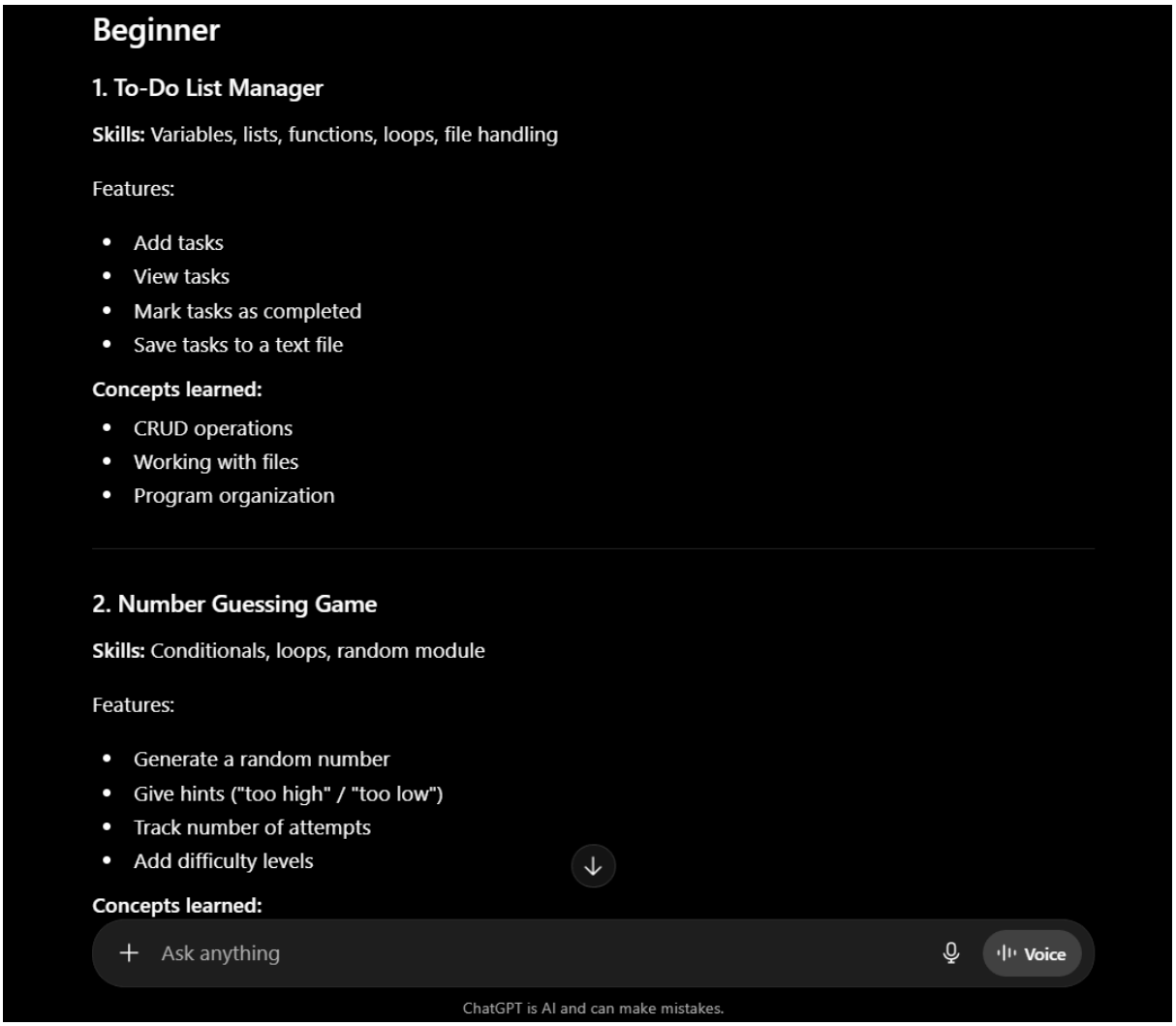
This loop is all you need…I'm serious. There is no rocket science. You learn how to actually use the language from a **free website** (w3schools is my favourite) and then actually **build** with it to get used to it.

Iris, how do I know what projects to build? Well my friends, I love telling people that you need to **ask AI**! Genuinely AI is like a 24/7 tutor and this is a great time to ask it exactly that!

Here is a chatGPT example prompt for learning python:



And look at the response:



It not only gave me **3** projects each, but also gave me **concepts** that I will **LEARN** from that project, and the **key features** I need to implement. This is what I mean when I say AI is an amazing **tool**!!

Now of course you can build your own projects and delve straight into the depths of hell with an ambitious project; but this course is meant to be free and for people who are just starting out. So if you're a complete beginner, take it easy :)

How to learn syntax

Syntax by definition, are just the **rules** of programming. How you define an object class in one language may **differ** in another. So even if you know 5 programming languages, you'll still need to learn the rules of a **brand new** language like anyone else.

The only thing that would differ is your **speed** at picking up a language; The **more** programming languages you know, the **quicker** you generally pick up new languages.

This is because.. Lets say I know **python** and **c++..**

Javascript will come *easier* to me because I'm already familiar with the *scripting* nature of python.

And if I'm learning **rust**, the idea of **smart pointers** will be a bit more *intuitive* compared to someone who doesn't even know what a pointer is, and what makes a smart pointer 'smart' (*all hail C++ 11*)

Learning a programming language isn't just about learning how to use the language, it's learning about:

- **What is the purpose of the language** *(Typescript for webapps and Rust for high speed engines and operating systems)*
- **Why the language is built the way it is** *(Typescript runs on a runtime engine and rust is compiled into native machine binary)*
- **Speed and Performance difference** *(Rust's compilation speed is slower compared to Typescript)*

Learning syntax is as simple as going to your **site** of choice like **w3schools** or **codecademy**, and *doing* their **exercises**. Alternatively you can just play around with the language in your **integrated development environment** (**IDE**) or **text editor** of choice.

The idea is you should be able to do things like *declare functions, create objects, concatenate strings, use different abstract data types*, all of this without a tutorial until you can build projects.

If you're going to **commit** to a language, I recommend learning how it works **under the hood** too. This isn't mandatory, but a lot of the concepts will be very cool to see in real time: things like

- understanding memory stack with functions
- why global variables can cause memory leaks
- “resource acquisition is initialisation”

It will indirectly make you a **better programmer** because you will be aware of where you are writing heavily **inefficient** code.

This is why computer science courses are built the way they are built: the theory knowledge prior to jumping into building things, is incredibly helpful.

How to build projects

Build the **9** projects from the AI prompt given to you from start to finish. Try **not** to use AI for any help. **ONLY** because I don’t trust you enough to actually learn and not resort to shortcuts, if you go to AI for any sort of help.

I’ll be honest, *the industry vibe codes...* Ever since LLMs came out, companies have become greedy for more and more output. But, don’t let that dishearten you from learning to code. Remember what I said before, **AI is only as smart and good as you make it.** So I need you to build the habit of building without AI first so that you aren’t stuck relying on the AI to do the thinking for you.

You need to use that trick I outlined earlier where you ask your LLM of choice for project ideas and build right away. Genuinely when you keep building, your strength in a language increases. Here are my **top** resources:

- **Geeksforgeeks** (has a tutorial and explanation for absolutely every concept imaginable)
- **W3schools** (coding syntax with exercises)
- **Stack overflow** (if you have a question, someone’s probably asked it and had it answered very brutally)
- **Codecademy** (very holds-your-hand type tutorial)

KEY INFORMATION!!! PLS you will need this as a reminder..

- Imposter syndrome is very normal, you temporarily relieve it by quenching your thirst and greed for knowledge
- This is YOUR coding journey, don't compare yourself to other people
- Any trivial question, don't be scared to ask AI, it's a 24/7 tutor
- However, do **NOT** fall in the trap of asking it questions constantly until you end up just copying and pasting code with little to no understanding
- Joining communities is helpful, like my discord server- when you surround yourself with ambitious people, you become the better version of yourself

Okai bai, get building **meow**! =^..^=S