Why 99% of People Fail to learn to Code

#Programming

Introduction:

Coding can be fun, exciting and fascinating to watch your own creation of develop interesting program on giving your own best satisfaction. However, there are certain difficulties that may cause a slowdown or hinder progress, even if your solution was designed to meet a certain standard or efficiency. Therefore, here are some tips that could guide an **programmer/learner** on avoid making these crucial mistakes when learning a new **programming language**, **framework**, **technology** etc.

Avoid Tutorial Hell at all costs

-- Tutorial is not an effective learning tool on enhance your understanding towards programming.
-- You watch, you learn, you feel like a coding genius.
-- And then, you try to actually apply what you've learned, and it's like your brain short-circuits.
-- In return, you get **zero input** from it!!

For exp:
Let's say you're learning how to build a machine learning model by using tensor flow & Python.
-- You stumble across a tutorial that walks you through building a simple to-do list app.
-- **You follow along**, everything makes sense, you're feeling pretty good about yourself.
-- But then, when it's time to actually code it yourself, **your brain turns into a bowl of mud** (unable to comprehend any information that you had just learned).

To escape the endless suffering of the tutorial's torturing:

1. **Actively apply what you're learning.**
2. **Maybe adding some new features to a specific project.**
3. **Create a system to remind you to actually do the tasks on your list.**

Important Notes

- *Practice makes perfect*
- *Most importantly, don't get stuck in Tutorial Hell -- break free and start applying what you have learned!!!*
- *Resist the urge of making mistakes!!!*

Look for tutorials that have you build something with them

-- Learning the **basic explanation of variables and loops** are crucial.
-- No offense, these important concepts are **not thrilling enough to push your learning agenda**...
-- What you need is a practical tutorial that can shows you the **real-world work** that forces you on doing it.

For exp:
You want build Ai Chatbot using Python & frameworks.
-- Rather spends an hour on explaining what a variable is, look for the one that **acquires the resources** you build from scratch. Addition, from start to finish, you have successfully have fully functional chatbot that built by you.
-- That's kind of tutorial that you need!!!!!!

Important Notes

- Find tutorials that don't just explain the basics, **but also building something from get-go**.
- **You learn by doing it** which can makes you more engaging and memorable than watching someone explain the concepts only.

Learn how to properly Google

-- This is the most prominent asset that you really need on master the deadly weapon in the world.
-- Googling isn't just about typing in a bunch of keywords and hoping for the best.
-- Rather, It's a skill that takes time and practice to master.
-- once you do, it will save you hours (if not days) of frustration and confusion.

Here are some examples that we could look from **good vs bad searching answers**:

- **Bad** : "My C# application doesn't connect to the database"
- **Good** : "connection refused error MySQL C#"
- **Bad** : "How can I add database to my spring app"
- **Good** : "Connect Node.js to MongoDB"
- **Bad** : "How can I create a navigation bar that changes when resizing the window"
- **Good** : "Responsive navbar react example"
- **Bad** : "How to fix Spring Boot errors?"
- **Good** : "ClassNotFound" exception spring"

Important Notes

- Always **be specific** on the problems that you need to solve with a **relevant keywords** to match your issues.

Conclusion

- By **avoiding tutorial hell**, **looking for project-based tutorials**, and **mastering the art of Googling**, you can save yourself a lot of time and frustration.

- Remember that programming is not just about memorizing syntax or reading a book, but it’s more about **building things** and **solving problems.**
- The **more you build**, the **better you will become**.
- Just remember to **take breaks** and **have fun with it** — programming can be frustrating at times, but it’s also incredibly rewarding when you see your creation come to life.
