Activate: Beast Mode

https://media.giphy.com/media/9zKB9ynAsiE3S/giphy.gif

There's an astronomical difference between simply *writing code* and being a great developer. Anyone can pick up the basics of **syntax** and formatting.

But what actually takes someone from being simply 'capable' to being 'great'?

The ability to learn.

The secret sauce to a developer's excellence can be found in the ability to harness an effective learning process, thus, moving past being a shell of static, safe mediocrity.

Learning through more systematic resources like books, courses, and tutorials can help you unfold your coding journey and pick up the basics *no problem*.

However, just like making a complex cooking recipe, becoming a better **developer** means that you need to diversify the techniques that you use and make sure to alternate between various learning resources.

With that said, below are our tidbits of wisdom to propel you forwards to a brighter coding future.

Be Teachable

Too many **computer science** graduates exude the “I know everything” bravado. Let's get it straight: you don’t. And that’s okay. Your curiosity and self-doubt can sometimes program you into the mood for breaking new grounds.

Don’t Spread Yourself too Thin

Don’t get overwhelmed with the media headlines that declare a new movement with every new function or technology.

You won’t see most new frameworks or features catching fire, hence, there’s no real need to learn all of them all at once. We recommend focusing your energy on the following three pillars:

- Living and breathing the basics of the **tech stack** you use;
- Being familiar with the latest version of your go-to stack;
- Being well versed in technology that is secured by market titans.

https://media.giphy.com/media/7NUR5Qvr93ed8T2dsG/giphy.gif

In case you have a hard time understanding the fundamentals, create a list of go-to topics that is easy to maneuver and access.

CodeMonk, for example, will help your bright mind get the hang of the **programming ABCs**.

Your Work is Never Done

We're fully aware that your main objective is to write high-quality code that meets the specs.

However, making the code work is actually just your starting point. Granted, that's what makes a huge difference between an average performance and an outstanding performance. However, now that you acknowledge that your code works, make it even better.

It's useful to accept the fact that you'll have to iterate your creation. Only then will you be able to distill even the smallest pieces of the problem and brew up a robust solution.

Read Lots of Code, Write Lots of Code, Repeat

In your daily work, learning to read code will turn out to be more important than writing it. That's because a big part of your tasks still boils down to debugging and cleaning up other people's coding attempts or performing code reviews.

https://media.giphy.com/media/ZVik7pBtu9dNS/giphy.gif

Tech reading skills can narrow the learning curve and help your literacy of the programming languages.

Whether it's writing large programs, supportable code, or **software architecture**, you'll perform more confidently with some code-reading skills under your belt.

Ergo, once you've found a great project (preferably, an open-source one), sink your teeth into it.

You should be thinking about things like: What type of architecture does this project leverage? Why did they opt for a particular pattern?

Keep Calm, and Carry on Coding

Your code will never seem perfect - *just deal with it*. There will always be some parts of it that just don't seem good enough. That being said, that's another differentiator between great and mediocre programmers – the great ones are always moving forward.

So, to all the coders out there: Keep calm, and carry on coding. You'll pick up your greatness down the road. Just embarking on the coding journey is extremely admirable!

https://media.giphy.com/media/l0IsIbqihlTrv41UY/giphy.gif

Join us today to thank our newsletter sponsor, HackerEarth. HackerEarth matches great developers with world-class companies so that organizations can meet their tech-hire needs efficiently. HackerEarth includes candidate accumulation processes such as hosting contests and coding practice sessions so that

candidates are at their beast mode and ready to wow your company. To find the right tech talent for your company, visit the HackerEarth platform.

***

Got a tech story to share with our readers? Everything you've ever wanted to know about how to get published on Hacker Noon - **get it here.**