

LEARNING PROGRAMMING IS A LONG RUN NOT A SPRINT



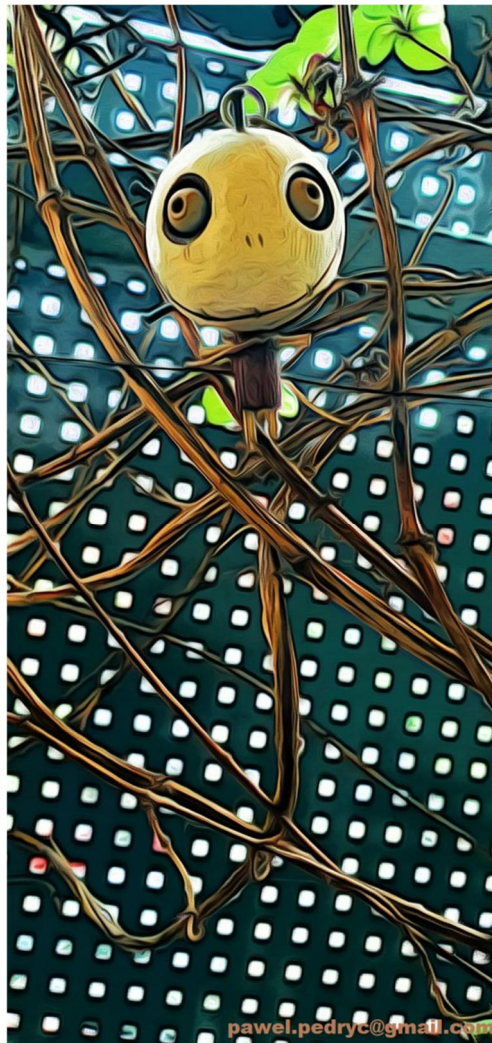
I've decided to share my experience as a programmer with the students here. I'm in programming for more than one year now. There are many things that I was using in the past for learning enhancement, and most of them I use today. Each of the students that take the Dataquest course have their own story, they all have a unique approach. Everyone is different and live in different circumstances, and it's possible that this essay won't provide good directions for everybody but there is a chance that it will help you.

The most important idea that I would like to convey here is what is in the title. IT IS very VERY important! Please read until the end of this paragraph and then take a 10-minute break. Just to think about it. If you are a beginner or maybe you studied programming for some time but feel burned out, this truth is crucial. It's impossible to learn programming in 1 week or 1 month. It's like running: if you know that you have to run 1000 km, you don't sprint. To succeed in the long run, you have to be smart about it. You shouldn't even think: how long does it take? You have to find a way to see the whole programming mission as something exciting. Without that, you will fail.

I decided to share my experience in a few short stories and examples.

Before I start, I would like to share a little bit of my background – it will make it easier to understand some of the examples given later. I finished my regular education many years ago. I work at a bank and, at the same time, in wholesale's gaming industry. The reason I started learning programming is very simple: I see it as a great opportunity for doing creative stuff and earning good money while doing it. The goal wasn't easy because working and studying at the same time can be overwhelming. But it doesn't have to be.

How to study when there is little time to do it?



If you want to learn anything, you need these few crucial things:

- Time,
- Energy,
- Motivation.

It would be good to learn at least 5-6 days per week. You probably think that it's impossible. In my opinion, in many cases it is more than possible. The issue lies elsewhere: people are not aware how much time they waste daily. To see that for yourself, open the settings on your smartphone and navigate to battery settings. On most smartphones, you can find some sort of chart or data set that shows how long you're using each app daily. I was playing games and reading stuff online for 3-4 h per day, almost every day! I realized that if I cut this time in half, I will have at least 2h for learning every day. That's a lot. That's 40h per month! 480 h per year!

We've established that there's definitely time available somewhere in your life. But how to actually find it?

In my case, the idea was: set up both jobs in such a way that would allow me to spend as little time as possible at work. Also, set things in such a way as to spend the smallest amount of energy while working.

How to do that without being fired? I followed an ancient Japanese idea: when you work 20% of the time, you are doing things in the most effective and profitable way. The rest of the time is mostly ineffective. Try to focus on the things you do during this 20% and do them as best as you can. I'm aware that this might not be possible at every job. If that's the case for you, maybe the first step is not to start programming but to get a job that will better correspond with your learning desires?

The second thing I did was to reorganize my work-time: I realized that I can do my 20% of work within a shorter span of time. Because most of my earnings depend on the number of sales deals I make, I realized that I can do all of my work in just a few hours (instead of pretending to work for 8 hours).

My actions created the right conditions for me to learn: I have more time now. Less money as well, sure, but until I can pay my rent and buy some food, that's fine. Why? Because I'd rather sacrifice 12-16 months to learn something that is interesting and gives me an opportunity for a better life, than earn twice as much as I earn now and keep doing something that doesn't bring me any real satisfaction.

How to find the energy to learn?



Suppose that you are not fully drained out of energy because of work. Even if you don't have to work hard at your job, a person's amount of energy is limited. Most likely, you will have to learn after work. That can be difficult. But is it really? What is so difficult about? From my experience,

it's the brain fatigue. It doesn't matter that my job is easy. Many different things have to be done each day: you have to buy groceries, take care of your family, run errands. And just like that, the whole day goes by and the only thing you want to do when you get home is watch a movie, and relax. How to deal with that? The answer is easy: you need to rest. In my case, I needed to change my whole daily (and nightly) schedule. I based my new schedule on one used by NASA astronauts. I wake up at 9 AM (eat some food and start my workday - I work remotely). After 2 PM I do the buy-groceries-drive-somewhere-run-an-errand part. I return home, eat, and go to sleep for 2h, most often between 7 PM and 9 PM. I try not to sleep longer, because if I do, I wake up feeling drowsy. Two hours is good nap time. After waking up I don't study right away. First, I dedicate 1-2h for something fun. It creates a positive flow and wakes me up. After that I can start learning, and finish as late as 3-4 AM when I go to sleep again. I have, in total, 7-8h of sleep per day: 6 hours from 3 AM to 9 AM plus the 2h evening nap.

Motivation is the third most important ingredient.

The most crucial parts of building your motivation are:

- Never learn.
- Never work.
- Take care of your mind and emotions.

The first two might seem unreasonable but it will all make sense, I promise.

If you already have time and space for learning, there's another thing you need: a proper perspective. You have to truly see programming as something interesting.

Let's get back to our main paradigms listed above.

To me, learning and working sound like pressure, stress and no fun at all. I don't like those concepts. Both of them are related to effort. Both of them drain your energy and weaken your motivation.

Instead of learning, do exciting things! Just realize that when something is exciting, time stops, it doesn't exist anymore - surely you had this experience! You can do exciting things without effort. And usually, you can do them longer than you do the studying, and much more effectively than you do the working. If you understand the depth of this idea, then you will NEVER learn and work again. Most likely you will achieve great success at your job too! It will be achieved without achieving - that's a great paradox and a secret to living a good life.

But what to do when for some reason you don't feel excited about discovering Python? Take a break and see what's going on in your mind.

I will point out only a few most common problems that may be a reason for not being excited and my personal solutions to them.

Problem 1.

You may be scared of one particular programming lesson.

Solution: Don't try to force yourself to learn it! Just set it aside and let yourself be with the fear.

Try to welcome it with openness and a smile. Beyond it lies a false belief that you need to succeed at all cost. Once you understand your fear, you will realize that you can handle the lesson with ease.

Problem 2.

You may stop feeling that learning programming is exciting because you started treating it like work or school.

Solution: As I explained before, that kind of mindset is wrong. But it's difficult to let go completely. It's rooted deeply into our social habits and the education system that often corrupts the idea of discovering things. Give yourself some time and contemplate the difference between working/learning and doing exciting things. Think what's more effective and which attitude will bring you joy instead of stress.

Problem 3.

You were too excited about your programming explorations and spent too much time learning without a break.

Solution: Take a break. Remember that it's not a sprint. If you are so excited that you can spend 6-8h programming, next time try to spend 1-2h less. If something is so exciting that it's impossible to let go, after finishing a project, give yourself a few days off. Never judge yourself for taking this break! You deserve it!

Problem 4.

You had a busy day at work or after work, had to do something with a great amount of effort, or an intellectual, an emotional engagement that made you feel wasted and burnt out.

Solution: forget about programming today. Remember that it's a long-distance run and you can't disregard your work, your social needs, or any other non-programming things completely. If there is a lot going on in your life, take it easy with programming.

Problem 5.

You don't understand one particular lesson and feel powerless.

Solution: Try to find a solution on the web and if you don't find any, ask the community on the Dataquest platform. This community is more open and friendly than others. In many cases I've found help there when I couldn't find it elsewhere, even on Stack Overflow.

Problem 6.

You had trouble understanding one particular programming lesson, found the solution on Stack Overflow, and you completely couldn't understand it.

Solution: This is a funny one. There are people out there that can create an inadequately complex answer for a very simple problem. It's like looking for an instruction on how to use matches and finding a 300-page book on the subject. If you encounter a long and complex answer for a specific problem that you know is a simple one - or you assume that it is - then look further and skip the long and bizarre answers. Sometimes a longer answer is better because it provides a more professional or a faster execution of a program but quite often it actually indicates someone's lack of understanding. Don't worry, in most cases you can find at least 5 answers. Look for the most elegant and easy one.

Experiencing Python may be fun but it may be a nightmare too. The key to success is following the learning method I have described here. I hope that it will help you turn programming into an exciting experience instead of a stressful obligation.