

How to learn how to learn

 flaviocopes.com/how-to-learn

Notes and observations on how to learn things quicker and in a more effective way

Published Jan 27, 2019

Big big news: today I launched [my new **React Course!**](#)
Special launch price until January 31 - just 3 days left!

As developers and software engineers, we are doomed to a common destiny: we are required to learn for the rest of our lifetimes.

10-15 years ago you had to know Java and JSP and Servlets and Tomcat to be at the top of the industry. Maybe then you learned Objective-C and all about Cocoa Touch because mobile apps were the new trend and, let's be honest, really exciting! Now it's all about React and modern JavaScript - 3 years ago's JavaScript is not JavaScript any more - it's old style JavaScript. As a systems developer C++ then Go then Rust? What about Haskell and Elixir?

There's a major shift in trends and job markets every few years.

I like to learn. There's something in knowing new things that we can apply in the profession, and as a programmer everything you learn is a tool in your arsenal and an awesome help in your career.

In this post I want to discuss the things I do to learn faster and more effectively.

Beat procrastination

Procrastination is your enemy. It can take many forms, like watching Netflix or just browsing Reddit. It can also take the form of doing shallow work, small unimportant tasks that are not really worth doing. Except they make you feel productive while hiding you from the real deal: learning that new thing.

How do you beat procrastination? You train to do that. One way it so just start. For example one way writers force themselves to write is to focus on writing 200 words a day. Once you start writing and reach the end of those 200 words you are now probably in the mood to continue. You focus on the process, on the system, not the outcome. You can also write 200 lame words until you get in the "zone" and write 5000 words of Inc magazine level.

Manage distractions

We need to get any help we can to beat procrastination. One thing I do is to use a macOS app called SelfControl, and a Chrome Extension called Block Site. I block all social media that are not work related, plus all the news site I visit. I need to use the iPad or the phone to watch those.

Manage your time

Another way is by chunking time. The pomodoro technique is very helpful in this: it's just a way of working 25 minutes and take a 5 minute break. Adjust time as you prefer, but the main takeaway is that you set yourself a slot for work, and a slot for rest. I can't stand the fixed nature of this technique, however. I refuse any rigid structure around my work, but I do something similar. I just write when I'm inspired and if I'm not inspired, I just don't sit at the desk.

Divide and conquer

I make my living learning and then teaching things. For example next month I'll dive into GraphQL and make a course about it. There is a great benefit in doing this for me, as I learn new things all the time. And there's certainly no shortage of things to learn. There's also a great benefit for the person that takes one of my courses: I start where they start, so I don't fall in the trap that experts fall into, by assuming things and just creating some resource for the wrong audience. I like that.

This however comes with a challenge: how can I learn quickly enough? I work in chunks. I take a topic, take a couple courses, read as many books as I can on it, and try to come up with a table of contents of the whole argument. That is going to become the index of an ebook. Then I start writing about each of those chunks, independently. I have a list printed out, and every day I pick 1, 2 or more topics I am in the mood to learn and write about. Following the mood makes things easier. Also, if a day I don't want to do it, I just skip the day and get back tomorrow fresh and ready to roll.

Dividing a big topic into chunks is much more human and manageable. If I tackle 1 chunk at a day, at the end of the month there's quite a bit of chunks under my belt. Each one of them is part of the big picture, and all I have to do is to visualize this big picture. Each chunk is a blog post, and I get the big picture by creating an ebook.

I really encourage anyone to take my approach and try to immediately explain what they learn. Don't be afraid if you are a newcomer to the topic, there's always room for your perspective on the subject.

Learning from books

Another topic worth discussing is reading books. I have lots and lots and lots of books I *want* to read. My books app is full of great books I never read. I read many books, and many great

ones. Many times I get enlightened by a book, but after a month or so, I simply forget what I did read in there. How can you remember the things you learn by reading a book?

Here's my technique. Whether I use the Kindle or the iPad to read, or I got an actual book in my hands, I highlight all the main concepts and things I want to remember.

Underlying things I want to remember is not enough however.

After finishing the book I will re-read it by only reading the things I highlighted, and while I so I do a summary of the book. Privately, but I also started publishing those, like i did for the [Deep Work](#) book.

Not only I have affirmed those concepts in my mind, I did write them (a big, big help) and now I have a reference I can go back and see what I learned from the book.

Or maybe I just want to write what I learnt from a chapter or just a paragraph, and expand it with my opinion. That's great, too.

Apply what I learn

When it comes to learning a programming language or a framework, the best thing you can do is to create projects using it. Side projects are especially great for this because you don't need to fully grasp the enterprise-ready design pattern and you can just learn as you go. There's no need for a 100% scaling architecture just to learn something. That's paralyzing. Funny projects are also great. I like working on platforms that make the fun part first class, like [Glitch](#).

You can also just build games, or as many people do, launch mini products on Product Hunt. That is a great way to give you the adrenaline and push necessary to go the extra mile and refine something so that it's ready for people to see and use, rather than sitting on your hard disk completely unused and forgotten in 3 days.

Rest

Resting your mind is very important. It's definitely more *human* to spend 2 weeks 1 hour per day on something than 12 hours in a single day. Every night your mind has the opportunity to think about what you learnt. I find that's way better. Maybe you can alternate 1 hour of reading with 1 hour of podcast listening or hands-on project.

Also when you're stuck on something, don't spend 3 hours on it. Take a walk. Take a nap. Go outside and run. Do something else, then come back. I can't count the times I woke up with the solution ready, the day after.

Or the solution famously awaits you in the shower, or while you are in the middle of the woods walking your dog. It's amazing how many times ideas don't show up when you're working.

[🏠 Home](#)

☾ [Light|dark mode](#)

Free ebooks

🔥 Also check out [Modern Web Development](#), a journey towards learning Web Development and JavaScript

© 2019 Flavio Copes