

Thoughts about programming, productivity and more...

By
Hassan Kanj

(2nd edition) August 2021

This booklet is a work in progress, a modified version and more posts may be added in the future, so make sure to visit <https://www.hassankanj.com/publications> in order to get the latest version.

Copyright © 2020 - 2021 Hassan Kanj.



My name is Hassan Kanj, I am an independent programmer (Freelancer), a maker, and I host a podcast and teach programming related topics in Arabic on my YouTube channel (coder voice - صوت المبرمج), and how to design hardware products, in addition to 3D printing topics on my second YouTube channel (KanjWorkshop).

I enjoy working and experimenting with different aspects of the programming stack, whether it is hardware, software or web.

In case you want to know more about my skills or if you would like to hire me, feel free to check my website for more details: <https://www.hassankanj.com>.

And here's my LinkedIn profile as well: <https://www.linkedin.com/in/hassankanj>

Preface

This booklet is a collection of posts I wrote and published online, it is mainly targeted towards programmers, though some of these posts may be useful to people from other professions as well.

Table of contents

Preface	4
Do you feel tired/overwhelmed when learning a new technology? here is a strategy you can follow	7
Some of the most important tools for programmers are pen and paper	8
Why you should keep going despite the difficult circumstances in your country	9
If you are a freelance full stack web developer, these are some tips that may help you	10
What to do when programming is no longer fun for you	11
What do you know about programming? Relatively nothing	13
It's really awesome how much you can learn from side projects	15
When working on a programming side project, don't aim to make it "perfect"	16
To be more productive, automate	17
When estimating a task, consider energy optimization and not just time optimization	18
A tip to be more productive and efficient is to benefit from existing tools	20
تعريف بلغة البرمجة بايثون	21
A good way to become a better programmer is to read and understand code written by others	24
أهمية صناعة المحتوى وكيف ممكن يفيدكم هل الشئ	25
An Open Source Project From Lebanon (initiative)	26
Here's how to be efficient and satisfied if you like to work on different projects	27
Embedded systems programming: Part 1	29
Embedded systems programming: Part 2	31
As a programmer, aim to be confident yet humble	35
How I trained a team of junior programmers using videos and WhatsApp	38
For senior developers: It is ok if you don't know	41
Add daily habits to your life	43

How to increase your fees or salary	45
I decided to live stream while working and here's why	47
I am considering teaching programming for a fee, are you interested?	49

Do you feel tired/overwhelmed when learning a new technology? here is a strategy you can follow

Published in: July 2020

Do you feel tired/overwhelmed when learning a new technology and/or do you feel like there's too much to learn? well, here's a strategy you can follow:

A good way to learn a new programming language or framework, is to build side project(s) using this language, and especially if it is a project that you are interested to build.

You can start by reading/watching tutorials about the new language (to understand the basics and at least some of the best practices [these are helpful so you can learn from more experienced experts]) then you can jump directly into your side project and learn as you go, and these are some points why this is helpful:

1- You are building a project you are excited about (even if it is a simple project), so this by itself will motivate you to learn the language while enjoying the process.

2- You won't feel overwhelmed by all the 'theoretical' tutorials and the endless pages of documentation, instead you will learn the basics first, then while working on your project you will start googling the part you don't know how to code. so in this way, you will learn by doing, and you start fixing your coding problems when needed.

3- From the start, you will feel the progress, because you are actually building a project, instead of spending months just reading books/articles because you want to feel 'ready'.

Some of the most important tools for programmers are pen and paper

Published in: July 2020

Some of the most important tools for programmers are pen and paper..

Start designing your app from there, and don't just jump directly into coding.. no need to draw fancy diagrams or detailed flowcharts.. it can be as simple as drawing some stickmen, arrows and rectangles that are connected together to provide at least a general overview on how your system is connected.. and once you have the big picture.. coding should be easier and less buggy.

And of course I am talking about relatively small apps/side projects, much bigger projects may need a more formal approach, and you can use specialized apps as well. But my point is even a single paper that contains a basic hand drawn sketch can have a dramatic effect on the whole programming experience. So don't underestimate this part.

Why you should keep going despite the difficult circumstances in your country

Published in: July 2020

We are currently living in very difficult circumstances in Lebanon, and it can feel very challenging and overwhelming to think about improving our skills or work on new programming projects, and it really is.. and this is totally understandable.

But thinking about it, what other options do we really have?!

The situation is really stressful and there's so much uncertainty, but I will give you 3 simple reasons why it is still a good idea to prioritize your programming journey:

1- The days are passing anyway, and even if we are feeling down because of everything around us, let's try to allocate even if just 30 minutes per day to work on some projects, or read about a new programming language or even to improve our linkedIn profile and update our skills there

2- in this difficult situation, and especially if you are unemployed, this is exactly what you need to do to increase your chances to get a job.

3- It is actually a great idea to work on side projects or to improve your programming skills in these circumstances, because this can provide you with a small rest from all the negativity/stress caused by the situation, even if it was for a short period of time.

If you are a freelance full stack web developer, these are some tips that may help you

Published in: July 2020

If you are a full stack web developer, and you deal directly with businesses, these are some tips that may help you, these are based on my own observations and on my experience as a freelancer and a full stack web developer:

1- if you are dealing directly with the client, take some sales courses online or simply check YouTube to learn more about sales and how to deal with clients.

2- improve your listening/communication skills and try to understand the client's business and problems, this is an important skill in order to build the right product and to suggest the correct solution for the problem.

3- if you want to focus on the full-stack then pay attention to your front-end skills, mainly CSS and similar skills, since clients care so much about the UI, and even if you have a great back-end code they probably won't care at all if UI is bad, so focus on this part and don't ignore it.

4- keep improving your skills in general, and focus on a specific stack and master it and keep learning and improving.

What to do when programming is no longer fun for you

Published in: July 2020

There will be a time when programming may no longer be as much fun and exciting as it used to be for you, I remember when I was younger how much exciting it was and it still is for certain projects, these are my thoughts why this happens and how to deal with it:

- When you started programming it was exciting for you, especially that you were building fun projects, no specific deadlines, no stress, no testing, no clients requirements and so on. But real life is different, programming will become just a tool, a part in a bigger process that aims to fulfill the business requirements of a client.
- You may be working in a company where your sole responsibility is to focus on a small part of the process, and repeating that day in and day out which may not be really fulfilling..

Now, that being said, this may not be the case, and you may be really happy in your workplace and with the work you are doing there, but if this is not you and you are feeling a bit frustrated or feeling like 'this wasn't what I was expecting', these are some notes/suggestions you should know:

- First of all, this is normal, and it is work after all, when something you love becomes work, the game becomes different the responsibilities as well, you will be playing by the business rules, and the market rules, and what your employer wants, in this case, just consider it as a job, a way to get money.
- Keep working on side projects or read about the technologies you are excited about, at work you may be programming an accounting system that you aren't excited about, but if you are into game development, or building open source tools, just work

on that for fun in your free time, it will feel so much different than your 'programming at work'.

- Another thing you can do is actually taking a break from programming and focus on other unrelated hobbies and socializing, this can help you focus on other aspects of your life, which can make you more excited about programming again.
- It may not be you, it can be simply be your workplace, the culture there, the apps they are building, the process and the technology they are using, etc.. so by switching to a different company you may feel more energized and more interested in coding again, but think well about this option before deciding, especially when the economy is tough.
- And finally, if all the above didn't work, and programming for you is much more meaningful, and it is important for you to code projects that you are excited about, then you will have to quit your job and pursue your vision, but let me be clear about this, if you want to do that, you should really be aware of the consequences, and it doesn't mean it will be 'fun' all the time anyway, on the contrary, this can be really stressful, and you still need to adhere to the market rules, but at least you will have a saying in what to build, and also keep in mind that this won't be easy, and you may suffer financially because of that (at least for a while).. but still, all that being said, this is one of the options you have that can be really rewarding, understand the risks of this, and proceed accordingly.

What do you know about programming?

Relatively nothing

Published in: July 2020

What do you know about programming?

No matter how much experience you have, the answer is always 'relatively' nothing.. relatively to all the information out there, and relatively to all the new frameworks and technologies that are constantly appearing..

There's front-end, back-end, network programming, embedded systems, OS programming, and much more, and each one has its own branches.

but the answer to how much you don't know is useless, and it can even be depressing and frustrating if you are a bit insecure about your skills.

if you are stuck in this mindset, these are some notes you should consider:

- It is a fact that no matter what you learn, there's still so much more to know, but still, your reference should always be how much you learned compared to last month for example, you had no experience in NodeJS, and now you know how to build a basic website using NodeJS and ExpressJS, well that's some good progress! I mean who cares if you have no idea what's Django is! that's irrelevant for now.
- You only need to know/master a specific field to be productive and be able to earn money, everything else is a plus, after all, it is about providing value.
- What you really need to know, is how to find what you don't know in the fastest way possible in order to solve your programming

problems, that's a crucial skill, because you will shift your thinking from being "someone who doesn't know" to "someone who is 5 minutes away from knowing and understanding any topic".

That being said, I still highly encourage you to explore topics that are related to programming even if they aren't related directly to your main job description, but do that not because you are anxious that you don't know, but because you are curious to know more.. and there's huge difference between the two mindsets.

It's really awesome how much you can learn from side projects

Published in: July 2020

It's really awesome how much you can learn from side projects, I have worked on a number of side projects over the years, and I have learned many things that are useful to me as a programmer.

If you can allocate a small portion of your time for working on such projects, I highly recommend it.

It may feel like a break from your work as well, and by the way, you can build a tool or an app that can actually help you to be more productive and efficient in case you want to add additional benefits to this experience.

When working on a programming side project, don't aim to make it “perfect”

Published in: August 2020

When working on a programming side project, don't aim to make it “perfect”, just make sure it is good enough and well architected, but don't get obsessed with “perfection”, and each day make sure to add some improvements here and there, and fix some bugs if needed.. and after multiple iterations and consistent mini-improvements, you will end up with a code that you should be proud of.

but keep in mind that this also applies the other way around, if each day you introduced minor bugs and hacks and workarounds to make the code work, after a while you will end up with a lousy codebase and a lot of technical debt.

To be more productive, automate

Published in: September 2020

To be more productive, always check your daily repetitive tasks, and if you felt that you repeat them extensively, then it would be a good idea to automate them, you can do that by writing a small program or a simple bash script that can handle these tasks fully or partially.

When estimating a task, consider energy optimization and not just time optimization

Published in: September 2020

When we estimate a programming task, we usually estimate the time needed to finish the task, but we don't usually take into consideration how much energy a task may need based on the current way we do it.

This is a different perspective to look at estimations, but let me explain why this is crucial especially for long term projects, and how to optimize for energy rather than just for time.

but before I continue, I want to mention that this applies to jobs that always require new solutions, and where you always face new programming challenges, if your job requires the same coding snippets, your brain might be already optimized to handle these tasks almost subconsciously with minimum energy/stress.

That being said, have you ever noticed that sometimes you work 1 hour on a specific task and you feel extremely exhausted, tired, and stressed, while sometimes you work on another task for so many hours, and you feel more relaxed and enjoying the process? (assuming here both tasks are as challenging, and you are in the same initial mood when you started both tasks,..).

I believe this is related to the way we approach a specific task/problem, and specifically related to the personal processes we use, and I know this topic is really complicated and it depends on different factors, but in this post specifically, I will just focus on the processes used to finish the task.

Based on my own experience, I noticed that the process I use to work on a specific programming task has a direct effect on my

energy level and how much work I can accomplish afterward, and to give a very basic example, let's assume that to solve a programming problem you search Google for a solution (since all your company does that), and then after 1 hour of reading you found the solution, and you implemented it, but during this time you felt exhausted and frustrated, since reading isn't your most comfortable skill, while you are more of a visual person, so for the same task, you went to YouTube, watched some videos, and after 1.5 hours you solved the same problem, yes it was still a bit frustrating, and usually there's always stress at work, but in general, you enjoyed the process more, and you solved the issue while feeling more energetic than in the 1st case, it took you 30 minutes more (since watching videos might be more time consuming in this case), but your energy level was higher, which on the longer term can help you work more, and be even more productive for the company or while working on your own projects.

And this also applies to the programming tools you decide to use, and how you approach solving a problem (learning from someone VS learning on your own, etc..), so what I am trying to say, if you enjoy programming, but you notice a frustration pattern while working on something new or solving new problems or learning new stuff, maybe try to experiment with other methods and processes to accomplish the task, even if these methods were a bit slower compared to previous ones, but I think on the longer run it will be worth it and it will have a good impact on both your career progress, and also on your mental health.

A tip to be more productive and efficient is to benefit from existing tools

Published in: September 2020

A tip to be more productive and efficient is to benefit from existing tools, for example code builders, generators, CLI tools to handle specific tasks, etc...

Don't build everything from scratch even if you can, because the question isn't always "can I build this myself?" but rather "should I?"

Here's my recommendation, learn and have the knowledge and experience to build it from scratch, but then never do it. It is like being a black belt martial artist, the best martial artists avoid fights at all costs, but are more than ready when it is absolutely necessary, being efficient means managing your resources, and that includes time and energy.

There may be some people on the other side of this, who build everything using tools and builders to the extent that if something minor went wrong or needs some modifications, they get stuck and don't know how to fix it! This is bad! And that's why you need to know how to build it yourself, but to never build it from scratch, be like a black belt martial artist who only fights when necessary.

تعريف بلغة البرمجة بايثون

نُشرَ في: تشرين الأول ٢٠٢٠

ملاحظة: النص بشكل أساسي مكتوب باللهجة اللبنانية العامية وليس بالعربية الفصحى مع استخدام بعض التعبيرات التقنية باللغة الانكليزية

مرحبا جميعاً، هيدي البوست رح تكون بالعربي (مع بعض المصطلحات التقنية بالانكليزي) وبدّي إحكي فيها عن لغة البرمجة بايثون، تعريف عنها، فلسفتها، بنيتها اللغوية وبأياً مجالات فينا نستخدمها.

لغة بايثون هي interpreted language مش compiled، وهي general purpose language يعني لغة متعددة الاستخدامات، فينا نستعملها بمجالات مختلفة ان كان بالويب او تتبني desktop applications او بالعربي تطبيقات سطح المكتب، بالاضافة لغير استخدامات.

طوّرت بايثون بمعهد الرياضيات والمعلوماتية الهولندي CWI بمدينة أمستردام من قبل جويدو فان روسم بأواخر ثمانينات القرن الماضي وأول إصدار لإلها كان بال ١٩٩١، بايثون ٢ صدرت بل ٢٠٠٠ وهلق صرنا بالبايثون ٣ يللي أول اصدار لإلها كان بال ٢٠٠٨ وأكد بعدو عم ينشغل على تطويرها.

ع فكرة، بالرغم من إنو بايثون هي كلمة انكليزية تمثل عائلة من الثعابين أو الأفاعي الغير سامّة وبالرغم إنو اللوغو تبع بايثون حالياً هو عبارة عن ثعبانين حد بعض ولكن بالحقيقه و بالأساس اسم بايثون اختاروا غويدو تعبيراً عن إعجابه بفرقة كوميدية بريطانية اسمها مونتي بايثون.

[فلسفتها]

بايثون لغة قوية وبنفس الوقت عندا فلسفة بتركز على البساطة، الجمال، والوضوح. فلسفة اللغة بتتلخص بوثيقة اسمها "the zen of python" ياللي بتشمل مفاهيم مثل:

Beautiful is better than ugly
الجميل أفضل من القبيح.

Explicit is better than implicit

الصريح أفضل من الضمني.

Simple is better than complex
البسيط أفضل من المُركَّب.

Complex is better than complicated
المُركَّب أفضل من المعقد.

Readability counts
قابلية القراءة مهمة.

[syntax - بنية اللغة]

بايثون تستخدم الازاحة بالفراغات أو ال whitespace indentation لتحديد ال code blocks أو كتل الكود.

[Python libraries - مكتبات بايثون]

كثير عالم بتعتبر مكتبة بايثون القياسية أو ال standard library من أهم نقاط قوتها في قلبها أدوات بتناسب كثير من المهام, مثلا فيها modules او وحدات, لبناء واجهات مستخدم رسومية أو Graphical user interfaces ووحدات للاتصال بقواعد البيانات العلائقية أو ال relational databases والعديد من ال modules الثانية..

و غير ال standard library, في عنا كمان ل PyPi لي هياي إختصار ل Python Package Index, و يُعتبر المستودع الرسمي لحزم بايثون التابعة لجهات خارجيه (3rd party), و حالياً موجود في فوق ال ٢٦٠,٠٠٠ مشروع, بميزات و وظائف متنوعة من ضمنها:

- الذكاء الاصطناعي
- تحليل البيانات
- معالجة النصوص
- بالاضافة ل منصات برمجية لتطبيقات الانترنت
- وغيرها...

[أمثلة عن بعض مكتبات بايثون]

الذكاء الاصطناعي: TensorFlow, Scikit-Learn, Keras
برمجة المواقع: Django, Flask
الرسوم البيانية: Matplotlib
للتعامل مع قواعد البيانات: SQLAlchemy

وأکید هیدی مجرد عینة بسیطة من المکتبات و فی غیرها کثیر

المصادر

- <https://ar.wikipedia.org/wiki/>

[https://ar.wikipedia.org/wiki/%D8%A8%D8%A7%D9%8A%D8%AB%D9%88%D9%86_\(%D9%84%D8%BA%D8%A9_%D8%A8%D8%B1%D9%85%D8%AC%D8%A9\)](https://ar.wikipedia.org/wiki/%D8%A8%D8%A7%D9%8A%D8%AB%D9%88%D9%86_(%D9%84%D8%BA%D8%A9_%D8%A8%D8%B1%D9%85%D8%AC%D8%A9))

- [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))

A good way to become a better programmer is to read and understand code written by others

Published in: September 2020

A good way to become a better programmer is to read and understand code written by others, and here's why:

1- By reading code written by others, you expand your toolset of problems solving, since you become exposed to different writing styles, and a multitude of ways to solve a problem that you were totally unaware of.

2- You will also become better and faster at it, which is very beneficial when you want to do code review at work, and for productivity purposes.

3- it can motivate you to contribute to open source projects, especially if you were checking a project you really liked and you felt that you can actually contribute and improve that specific project.

And to start doing all that, you can simply go to GitHub and check projects there, and make sure to pick a language you like, also, it is ok if you didn't understand the whole code, even if you get just a minor benefit from what you read (like the styling for example, or a simple trick), it will still be worth it.

and finally, while you become familiar with reading code, don't forget to write code as well, and share it, so we can read it!

أهمية صناعة المحتوى وكيف ممكن يفيدكم هل الشي

نُشرَ في: تشرين الثاني ٢٠٢٠

عُدلَ في: أيار ٢٠٢١

ملاحظة: النص بشكل أساسي مكتوب باللهجة اللبنانية العامية وليس بالعربية الفصحى

مرحبا جميعاً،

بهيدا البوست بدّي خبركم عن أهمية صناعة المحتوى وكيف ممكن يفيدكم هل الشي. أنا صرلي فترة بِصنع محتوى خصّو بالبرمجة وبشاركو عالانترنت، بالأغلب كفيديوهات على يوتيوب بس كمان بكتب بوستات نصية ببعض الأوقات. وبآخر كم شهر زاد تركيزي على فيديوهات البرمجة وبلش ناس أكثر تحكيني أو تتواصل معي ت نشوف كيف فينا نتعاون أو تستشيرني بموضوع معين وزادت شبكة معارفي، وبنفس الوقت كنت عم قدّم محتوى بيساعد المبرمجين وبيحسنّ لهم حياتن المهنية. ومع إنّو بعدني بعتبر حالي بأول الطريق بيللي حابب أعملو بهل مجال بس لهلق النتيجة ايجابية وهون حابب إنصحكم تجربو إنتو كمان تصنعوا محتوى وتشاركوه، ما ضروري فيديو، في يكون نص أو انفوغرافيك أو مقال بسيط عن تجربتكم باستعمال لغة برمجية معينة.

و لو قد ما كانت خبرتكم بسيطة، دايماً في شي فيكن تقدموه وتفيدو غيركم في، لأن ما حدا عندو معرفة بكل المواضيع ودايماً فينا نتعلّم من بعض، وبنفس الوقت ممكن هيدا الشي يأمّنلكم فرص بطرق ما متوقعينا.

وهلق بعصر "السوشال ميديا" صار فينا نكون منتجين لمحتوى جيد بدل ما نكون مجرد مستهلكين للمحتوى وكل يللي بيتطلبو الموضوع منكم هوّي بعض الوقت وحتى لو نشرتو كل شهر بس مقال أو تجربة شخصية بمجالكم، هيدا يعني بالسنة صار عندكم ١٢ مقال أكثر من السنة يللي قبلها.

An Open Source Project From Lebanon (initiative)

Published in: February 2021



One of the ideas I had in mind for a while now, is that we need to work more as Lebanese developers on open source projects and to become more known in the international open source community and at the same time build some tools that can help us and other programmers (and not just build open source for the sake of open source), so I had an idea for a project, which is actually a badge that any open source repository from Lebanon can add to their README file.

This is just the beginning of an idea, that we can in the future build concepts, tutorials and documentation around it, like guides to help programmers from our country on how to build and contribute to open source projects, etc..

If we communicated with other Lebanese developers who already have open source projects, and suggest they add this badge to their own repo, this will be a good start, and then we can proceed from there.

For more details and to download the badge, check the GitHub repository here: <https://github.com/HassanKanj/opensource-project-from-lebanon>

Here's how to be efficient and satisfied if you like to work on different projects

Published in: February 2021

Some people can happily/easily focus on a single project at the same time and feel satisfied, while others are more naturally inclined to work on different projects and explore new stuff (this usually happens with those who are high in the “openness” trait according to the big-five personality traits model [Google it if interested]), and though this may seem very interesting, it can still have the drawback of not being efficient if this trait wasn't controlled or even restricted and balanced (and this can happen with creative people such as artists, and programmers with entrepreneurial tendencies,...)

Let's assume here that by a “project”, I mean work related project or a side project, or even the process of learning a new programming framework, etc..

So if you are someone who works on different projects at the same time, and you struggle with prioritizing or being very efficient, a good strategy is to balance things out and prioritize (in term of urgency, importance, etc..), but also to balance things out in terms of time allocation.

For example, you can allocate specific days to focus solely on a single project, while in other days you can work on multiple projects simultaneously, and fine tune this process as you see suitable, in this way you won't feel overwhelmed by your multiple interests and projects or even learning experiences, since you will dedicate certain days for extreme focus on a project, while you can still satisfy your need for experimenting and learning new stuff, by allocating other days just for that.

What I said may apply more for freelancers, or those who don't have full-time job, but for those who have a full-time job, they can still apply this concept after their work or in the weekends.

And on a side note, if you have the budget, you can always hire someone with a different personality trait that can complement yours, but maybe I will leave that to another post to discuss it in more details.

Embedded systems programming: Part 1

Published in: February 2021

Building websites, mobile apps and desktop applications is interesting, and can be challenging as well, but still, all these programs mainly run inside your typical hardware devices, such as laptops, PCs and mobile devices, and you have to interact with them using a screen and a keyboard.

But what if you want more than that, what if you want to build your own gaming console, your own coffee machine, or a custom mini-car that can move around inside your house, or even some non conventional crazy hardware that doesn't even exist yet! and all that using a code you wrote with some features that goes beyond the screen/keyboard interactivity, and then let that code runs freely in the physical world, instead of it being "trapped" inside your PC.

Yes, doing that is possible, more accessible than ever, and many programmers work specifically in these fields, where the code meets the hardware in the lowest levels, and saving every bit of memory is extremely important, and where you talk in Bytes instead of Gigabytes, and where so much magic happens.

In that world, the 32KB fully functional programs tell some scary stories at night about the Chrome monster that consumes many Gigabytes of memory to run, while they can happily run for years using few Kilobytes of memory, on a single battery..

Welcome to the world of Embedded Systems Programming.

Many of us are currently immersed in the world of high level programming with abundance of disk space and memory, that we may mistakenly assume that all the written code ends up as a mobile app, a website or a desktop application, however, there are currently so much code running around you which is hidden in plain sight, running silently and very efficiently, a code that's available

inside your AC, TV, remote control, music player, your smart watch, your car, and in so many more devices all around you.

Building and programming hardware products is one of my passions that I have first-hand experience with, and I would like to take you in a journey to explore that world in a series of posts that I am planning to write, and this is just the first part of it, so buckle up and prepare yourself because we are going downward to the low level of the computing world, and stayed tuned for part 2.

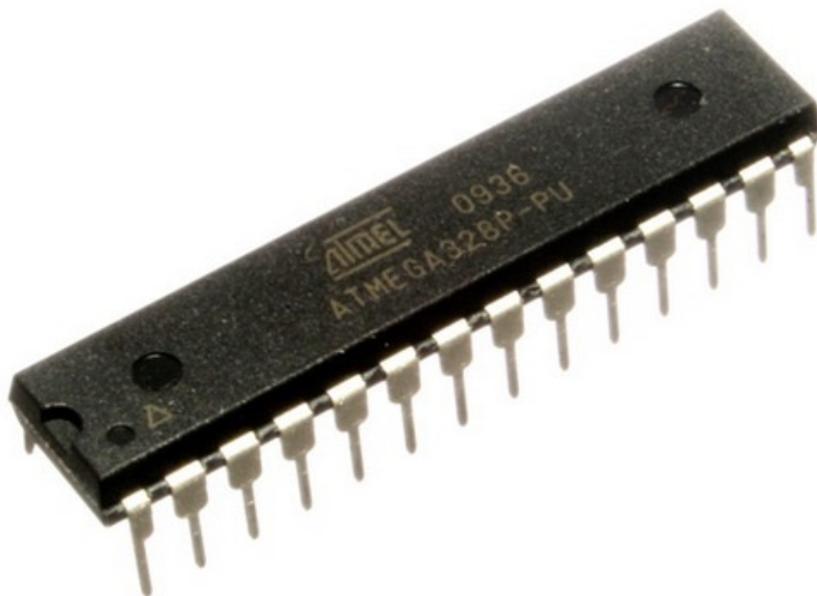
Embedded systems programming: Part 2

Published in: February 2021

Edited in: May 2021

This is the 2nd part of the embedded systems programming series, and in case you haven't read the part 1 post yet, make sure to check it first.

As programmers, most of you are accustomed to running your code on a standard bulky computer, but you may be wondering, if you want to build and program your own hardware gadget, let's say a small robot car that moves around (and maybe even produce some nostalgic 8-bit beeping sound while bumping into your furniture, because why not!), then how exactly you program the car, and where exactly you place the code.



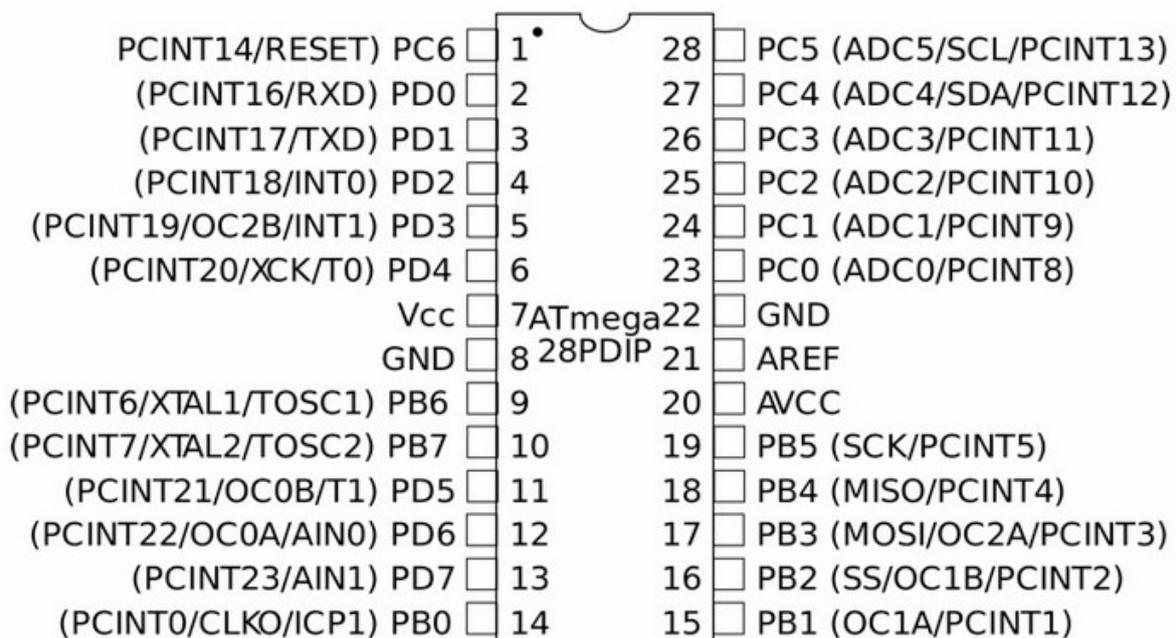
ATmega328P in 28-pin narrow dual in-line package (DIP-28N): <https://en.wikipedia.org/wiki/ATmega328#/media/File:ATMEGA328P-PU.jpg> (License: CC BY-SA 2.0).

In this series, I will answer such questions, and for this post specifically, I will introduce you to the Microcontrollers.

Just as we have big computers with processor(s), memory and I/O peripherals, we also have some really small computers that also have such components, and all these components are integrated inside a single chip that's called a Microcontroller.

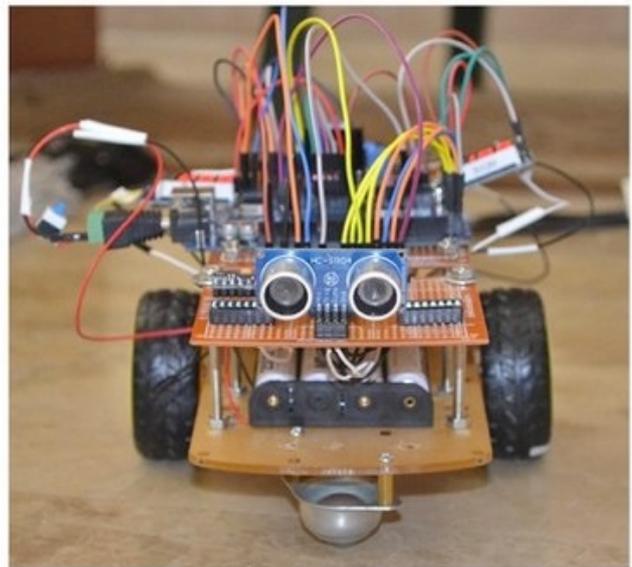
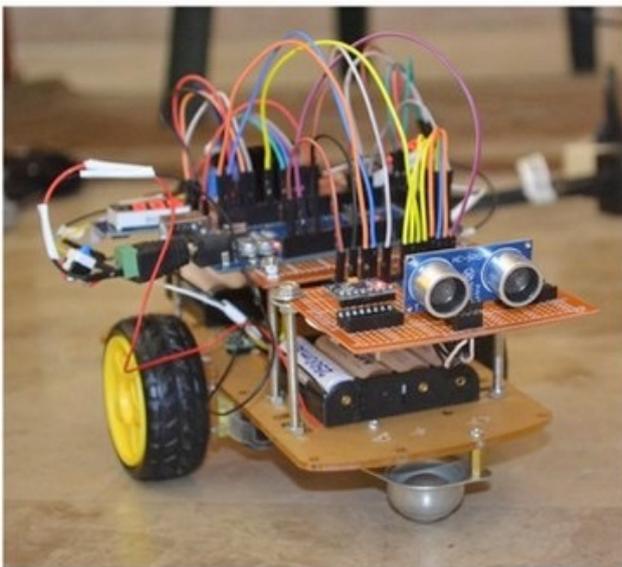
There is a wide range of Microcontrollers with different specs, and based on your intended usage and/or budget you can pick one that suits your needs, and they are relatively really cheap, you can get one for less than a dollar, or for a few dollars.

You may be wondering, how a Microcontroller interacts with its external environment, well, it does that through its pins, different pins can have different functionalities, for example they can act as input/output pins, or power pins, and certain pins can be responsible for specific communication protocols (for example SPI, I2C or CAN,...).



The Microcontroller receives input or send output as electrical signals that can be a digital high “one” or a digital low “zero”, though some pins can have analog functionalities where the voltage is more dynamic than a digital one or a digital zero.

Now, let’s get back to our robot car example, let’s say we want our car to detect obstacles and automatically avoid them while moving around, what we can do is to attach an ultrasonic sensor to the car which can detect the distance between the car and the obstacle in front of the car, and then attach the sensor pins to the Microcontroller input pins (now we have an input device that sends distance data for the Microcontroller to process), and then we attach motors to the Microcontroller as well (through an intermediary hardware which is the “motor driver”, but I won’t discuss that here), and the motors will be the “output” device, so in summary, our car will get distance data from the input device which is the sensor, and based on the code that we wrote, it will take decisions accordingly and send output signals to the motors to go left or right for example and avoid the obstacle, and we can also attach a buzzer, which is a small device that do the beeping sound, and beep more intensely the closer we get to an obstacle.



RC car project that I built and designed in the past.

We have so many devices and sensors that we can attach to microcontrollers (whether directly or indirectly) in order to receive data or to control the external world, such as temperature sensors, GPS modules, SIM cards modules, photoelectric (light) sensors, extremely precise motors, extremely powerful motors, LCD screens, and so many more, now you can start to imagine the wide range of projects that can be done by mixing these items to produce your own custom device, and don't forget that the main part is that we can program the Microcontroller with our own code so they can do all type of crazy stuff.

And that's it for part 2, and stay tuned for part 3.

As a programmer, aim to be confident yet humble

Published in: March 2021

A good combination you should aim for in your programming journey (and actually in life in general) is to be confident yet humble, having one without the other can be detrimental, toxic, and bad for you and others.

let me explain...

Confidence comes from learning, practicing, facing problems and solving them, being punched in the face then getting up and fighting back (literally and metaphorically).

Misery, ignorance and challenges are some predecessors to true confidence as much as they are the predecessors to hopelessness, and I don't believe that standing in front of a mirror and saying 10 times "I am confident" can achieve true confidence (though this can be helpful, especially in certain cases but don't count on that solely), and of course confidence is contingent on the topic, so you may be confident in certain areas while not that confident in others.

Being humble while being confident is essential, and I feel that to keep improving and also to be helpful to the community you should have both, because the moment you stopped being humble you may start thinking that you know everything or that you don't need to continue learning, and if you aren't that humble you won't be approachable by others such as other programmers online or colleagues at work, because they won't like you or they might be too scared to talk to you.

In my case for example, I try to keep myself in check, especially when my skills improve while finishing a project or when learning something new, because unintentionally you will feel more confident in your skills, which is great, but you just need to keep that in check,

and for me for example, I genuinely know that there are so much technical stuff that I have no idea about, and even though I've being a programmer for a while, I am 100% sure that there are so many stuff that junior programmers know that I have no idea about, and these ideas and realizations keep me grounded.

Now let's address over-confidence/borderline arrogance and over-humbleness (not sure if that's a word though!), and why they are really toxic to others.

In our society, you will meet people who pretend to know about EVERYTHING!! and have an opinion about EVERYTHING!!, and I would like to emphasize on the word "pretend", because I would love to have a genuine conversation with someone who has knowledge about different topics, and have spent time exploring different areas and formed their opinions accordingly, but I despise those who talk about everything with such "false confidence" while they literally have no idea what they are talking about, and not just that but also they won't listen when you try to mention some contradictory facts to what they mentioned.

For such people, pretending is so much easier than actually doing the work and genuinely understand the topics, for them "showing confidence" is more of a manipulative tactic (even overcompensation), especially that confidence is attractive, and their perfect victims will be mainly insecure people, but even worse, if their prey were skilled people who are insecure, having an imposter syndrome or even "overly-humble".

Overly-humble people that I am focusing on here, are those who are objectively very skilled in a certain topic but they are subjectively insecure about their knowledge or lack the skills to speak up, and this is bad because they are unintentionally contributing to the popularity of those who pretend they know, I believe that these people owe it to themselves to understand their true value and how skilled they actually are, which can then be projected as true confidence, which then can be projected positively in their work environment and on the whole community.

What I mentioned above was a bit general, now let's wrap it up and link it back to programming and how we can benefit from the above to create a better environment for all of us, and these are my suggestions:

First of all, Even if you are a junior programmer or an intern, state your opinion/approach on how to solve a specific problem, especially if you think it might be better than the suggested approach, and even if you don't have a solution but you think the suggested solution has issues, address them or demand explanation, programmers who are truly confident yet humble, will answer you and help you understand and even admit that they might be wrong, but in case the programmer made fun of you or dismissed what you said, then they probably belong to the more insecure/falsely confident type, and this shouldn't deter you from asking, if anything, it should help you spot the real deal and the fake ones, and in case you were witnessing such a scenario and you have an answer to the programmer who asked, step in and help, these small acts and encouragement can have a positive ripple effect across the whole community, especially in a domain where everything is evolving so fast, and many programmers have imposter syndrome or insecurity regarding their skills, and a small encouragement can go a long way.

And finally, you should have an honest conversation with yourself, and state/be confident about the skills that you are good at, which will help you understand your true self worth as a programmer and your importance to the employer, which can reflect positively on yourself, your colleagues and your salary.

How I trained a team of junior programmers using videos and WhatsApp

Published in: April 2021

I would like to share with you, how I am currently training (for the first time) a team of junior programmers remotely using custom videos and a WhatsApp group, I hope you find this experience useful to you in one way or another.

Let's start from the beginning, as an independent programmer, I decided that I need to hire a programmer (or maybe more) to help me with specific tasks when I get new projects.

so I modified my personal website and added this sentence:

"Are you a programmer who wants to work with me? Apply here.",

and when they go to the application page, there's a form they have to fill with some basic stuff such as their name, CV, skills, etc.. and a descriptive text as well, I will share with you part of that text below in order to get a general idea about what I was looking for:

Details about the job

Are you a junior programmer who is interested to help me to accomplish certain tasks?

The type of the work is task based, that means I assign a task (or more) to you that may literally just take few hours or less to finish, some tasks may take few days though.

Regarding the tasks themselves, they may range between different technologies and languages such as: NodeJS, php (Laravel, Wordpress, Drupal), HTML, CSS, React Native, E-commerce, hardware programming and even research regarding a specific topic, but let me be clear regarding some points:

You don't have to work on all of these, you may simply be interested to work on HTML tasks only, or to help me doing some research related to an e-commerce plugin, etc.. (you can specify that in the form at the end of this page)

You don't have to be knowledgeable about any of the above, as long as you have good work ethics and you are interested to learn, I will guide you through the process.

For now, this is not a part-time or a full-time position, it is just something you can do on the side in case you have some free time and you want to earn some extra money.

Expectations

here's what to expect if you decided to join:

I like to work in an organized way, I may provide you with videos and/or documentation that I personally made to help you with your task(s) and I will guide you as well.

Working with me will hopefully be a smooth process, since I am a programmer like you, so we can communicate better, and I can explain the tasks in a way that makes sense to you.

And I shared a link in a specific WhatsApp group (I didn't want to share it everywhere, at least as a start), and I get many applications, and here's the cool part, instead of interviewing them one by one, and going back and forth, I made a video and I explained to them what I am looking for and I went into some details, and I sent them a link to the video (which was an unlisted YouTube video), and also I mentioned to them that if they weren't chosen to work with me, they can still join the training program.

and most (if not all of them) were interested to join the training program even if they weren't picked, so I invited them to a private WhatsApp group and I started working on video tutorials that I published online and asked them to watch them and there's a specific deadline or else they will unfortunately be removed from the group, and then later I started asking them to send me the programming files that they worked on based on the tutorial, now, some of them didn't fully commit and they were no longer part of the group but a good part were committing and were benefiting and following the tutorials, and I was also flexible regarding the deadline, so if someone wasn't able to finish the "assignment" in time for one reason or another, I just need to know in a private message before the deadline and not after.

Now, in case you are in a similar situation, I highly encourage you to experiment with making good quality videos and share it with the team, it is a very efficient and scalable way to teach and train, and

make sure that the videos are really organized and clear, and provide your team with a chat group where they can ask questions and receive the needed support.

After a while of sending videos, we had a Zoom call where I get to know them more and we bonded as a team, it was a nice experience, it felt really good how these programmers are from different areas in Lebanon with different technical backgrounds but we were all meeting in a single place to work together, and all that started by simply creating an online form, then by creating some videos and by communicating through chat.

and yes this will take time and effort, but I feel that benefiting from the technology, setting some clear rules, and communicating efficiently can help you train teams, build projects in a fraction of the time compared to using other methods that aren't as efficient.

Now, the training is still in progress, we finished a good part so far, and I had another Zoom meeting with the team to discuss the next step (which is handling actual projects).

And as the team are learning, I was learning as well, since it was the first time for me that I do something like that, and finally, I hope you learned a thing or two from this story that is still unfolding in realtime.

For senior developers: It is ok if you don't know

Published in: April 2021

This message is for every developer, and especially for those who are in senior and leadership positions.

Please use more the expression: "I don't know" or "I don't know, but I can check then get back to you", when you don't know something, we need to normalize this type of healthy behavior, and yes, I somehow understand that you may feel insecure not knowing something especially if the question was asked from a junior developer, or if you are in a place where you are the center of attention and there's an assumption that you are the expert in a certain domain.

What's the worst that can happen? people won't "respect" you? people will think you are not good enough? well, first of all and paradoxically, people who will do that are usually the most ignorant, and people with more knowledge will totally understand that there are stuff you don't know even if you are an expert, and for those who would "make fun" of you, well this scenario is usually just in your head, and probably a very low minority of toxic people might do that, and if it happens, you can assert yourself and deal with them accordingly and if that's how they think maybe it is better to keep some distance from them.

That being said, if you notice that you say too much the expression "I don't know" whether as an answer or to yourself, then this can be a wake up call to work more on improving your skills, and to spend hours daily sharpening your knowledge so you can feel more confident in what you know.

And if you sometimes act like that, please don't feel defensive or annoyed from what I said, that's far from my aim, maybe we grew up in a culture that encouraged such a behavior (consciously or

subconsciously), and such a behavior isn't just related to programming, it usually expands to other domains and to a whole culture, but we can always start with ourselves and propagate and spread such concepts to our inner circles at least.

and I would like to end this post with two points:

The first one is by publicly admitting that I personally almost know nothing relatively to the huge pile of knowledge out there, even in the technologies that I worked on for so many years, and that's how I genuinely feel, and of course that doesn't mean you shouldn't be confident in what you know and ask for a good salary and compensation for your skills (just thought to mention that so you won't mix these 2 concepts)

and the final point is that, even if you become knowledgeable and a top expert, that still doesn't give you the excuse to act superior to others, yes you can be proud and confident with what you achieved, but others may know stuff that you have no idea about, and may own traits that you don't have, also their circumstances and limited beliefs may led them not to know much, in this case you can actually help them, guide them, and that may not necessarily change the world, but we don't need to all have the dream to change the world, sometimes making someone's afternoon brighter by sharing a piece of knowledge that they didn't know in the morning is still a great goal to aim for.

Add daily habits to your life

Published in: April 2021

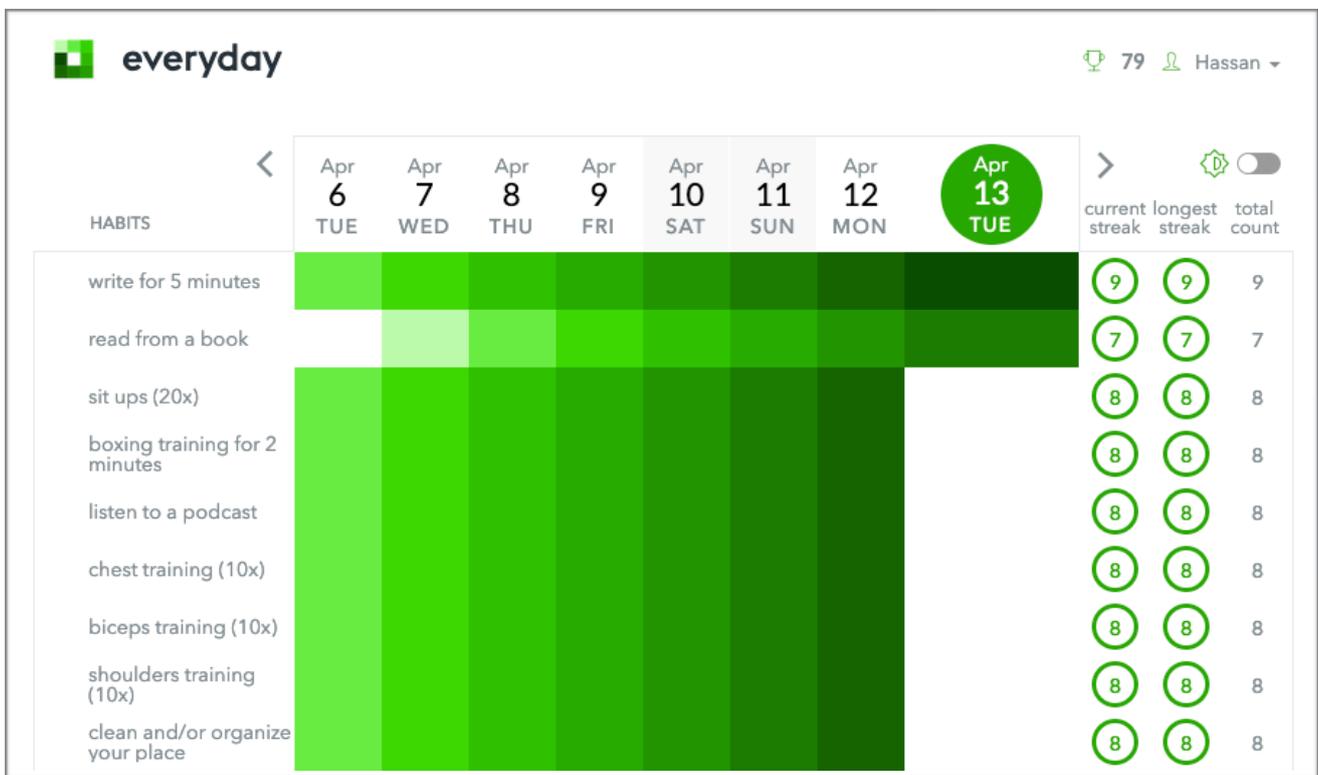
In the last few days, I started to focus seriously on having some persistent and daily habits, here's my experience so far, in addition to some tips that might be helpful to you as well, and this can be applicable to your programming tasks/habits (e.g allocating time for learning new technologies, working on side projects,..) or they can be more general as well.

Probably many of us tried to include certain habits into our daily routine, which might work for a while before we just stopped doing them for one reason or another.

There are many reasons why that's the case, but I would like here to focus more on some tips that I learned which will make habits stick on the long run, but before discussing that, I would like to emphasize that habits can be very simple behaviors and they don't necessarily have to be complicated tasks.

what you have to do is to make the habit/task so easy (at the beginning at least) that you won't mind performing it daily, let's say for example, you want to write daily, if you decided to write 30 minutes per day on a daily basis, after a week or more you may stop, so how about you literally set the duration to 5 minutes? can you do that? well, the answer is probably yes! and by the way, that's exactly what I did, one of the habits I set is to literally "write for 5 minutes", that's it!! and you can totally write for a longer duration, and that's what usually happens when I start writing, but the bare minimum would be to finish the allocated 5 minutes at least.

I am currently using an app called "everyday" (<https://everyday.app/>) to track my habits, it is a free app up to 3 habits, and you have to pay to get access to unlimited habits.



I felt it is worth it to invest a small amount in such an app (at least to pay for a month and try), since the return on investment for sticking to habits is totally worth it for me, whether regarding my health and financially as well.

There are some free alternatives (and actually I used to rely on a printed paper or Google sheets to track my progress in the past and you can use any method that works for you), but this app looked really good and easy to use with some cool features.

I have attached a screenshot from the "everyday" app, and I am enjoying the process so far, and two final tips I would like to mention, the first one is to try mixing more than one habit together (if possible) for example: listening to a podcast while working out, and the final tip is if you want to read few pages from a book, A good strategy is to leave the book near your bed so you can read early in the morning or before you go to sleep.

How to increase your fees or salary

Published in: April 2021

You may be wondering how much money you should get paid? and how much your skills are worth?

To have some good answers to these questions, you need to understand the real value of what you are providing, many people link money/salary/.. to time and hard work, and indeed there are a strong correlation, but in reality it boils down to the value provided.

When you approach "the money problem", you need to truly understand the value of what you are providing, and keep in mind that this can be really subjective, for example, have you ever went crazy while wondering why some people spend half their salary on a product or set of products that for you seems worthless?!! well, because for them these products/services/... are providing some real value that they are willing to gladly spend such amounts.

The value is then defined by the market and not by you (partially at least), being proficient in a very old programming language may not get you any jobs in a specific market, while it may get you a huge amount of money in a different market, especially if there is scarcity in such skills, an example of that is when a bank is using a legacy code that needs to be maintained and which depends on a very old programming language.

that being said, you can still increase your fees above the market range if you convinced your client of the value you are providing, and why you are charging more, and another approach you can follow to increase your fees is to learn some complementary or beneficial skills that your peers aren't focusing much on, skills that you know they provide additional value for your clients, for example if you are a programmer, and you learned about design or improved your soft skills or even your business skills so you can provide some profitable solutions for your clients, this can increase the

value of what you provide, and of course you don't have to do any of that, and focusing on programming might be more than enough, but going the extra mile might be totally worth it if that's something you are interested in.

I decided to live stream while working and here's why

Published in: April 2021

Edited in: May 2021

This post is for those who are working remotely and/or as freelancers, and need an additional motivation to feel more productive and more accountable, what I am going to discuss next is an idea that's going to help me be more productive and hopefully help you as well.

When you work as a freelancer, sometimes it can get a bit tricky to get things done, on the other hand if you work in a company and especially with a team, the environment can be more encouraging, and to motivate myself to be more productive and accomplish more I experimented with different stuff like setting daily habits, timers, etc.. and of course having a main vision regarding my career is a main driver as well.

one of the method I experimented with as well is using an app called "focusmate", which used to be a free app (and now it is paid with a limited free plan), the app's idea is to connect you with someone to work together silently, and you both have to open the webcam, and there are some strict rules, like you briefly mention what you want to work on to each other, then no talking during the whole session, and at the end you tell each other what you finished so far and that's it! when you read this, you may feel that this is a bit weird, but after trying it, it was very beneficial and it worked for me, you feel like you are working with someone, and it feels like work time and you are being held accountable.

As I mentioned the app used to be free for unlimited sessions, now it is free for only 3 sessions per week (according to their FAQ page), so I later thought that maybe I can setup a zoom meeting where others can join and set a webcam then we work together, though

this is a nice idea, I didn't want to spend time managing that, and making sure everyone stick to the rules and just work/study without talking or chatting, etc.. and also I needed to find a way to share the zoom link, etc.. so I didn't proceed with that.

But recently, I found a channel where the owner streams "study with me" videos! and I really loved the idea! he simply stream himself studying for long hours, and anyone can join him and it will feel like you are studying with someone.

And I decided to experiment with this idea, but instead of studying, I will be working and maybe other stuff as well such as recording video tutorials, reading books, etc.. and I will simply start a live stream and you can work/study with me silently, you can simply open the video and place it in a small section of your screen so it can provide the impression as if you are now in the "working" mode, and from my side, it will feel more serious for me to get things done since I am streaming what I am doing.

I also recommend that if you are "working with me" to mention that in the comment section of the live stream video and mention what you are working on (if you want to), and if others did the same, it will provide the impression that we are all working together, and it will be a nice encouragement as well.

It is an experiment, it may or may not work for you, but if you want to join me and check the live videos, here's a Facebook page I made where I am going to share the live streams: <https://www.youtube.com/channel/UCP-BpTn1jxEhOa1McYCB0jQ>

Also here's a video I made (in Arabic) talking about the idea: https://www.youtube.com/watch?v=X7_lsdqQPDA

I am considering teaching programming for a fee, are you interested?

Published in: May 2021

Edited in: May 2021

As some of you already know, I create and publish YouTube videos in Arabic about programming related topics (NodeJS, WordPress, Docker, React native, programming for the complete beginners,...) [YouTube channel: <https://www.youtube.com/codervoice>], in addition to programming webinars,... The content is available online for free and anyone can watch them.

Providing content and teaching programming topics is something I really enjoy, but still for now it is not my main source of income, but recently I started to think seriously to focus more on that and to provide paid services related to teaching, providing private tutoring sessions, etc..)

My plan is to keep the free content available on my channel but provide additional paid services and these are some suggestions I have in mind:

- Make intensive/condensed short courses (few days or so) over Zoom, where I explain certain topics or frameworks with live coding and practical examples (e.g WordPress, Laravel, NodeJS, React native, etc..) and you can ask questions and interact with me (and this material might be available for free on my channel as well, but some programmers may prefer live interactions where they can ask me questions and learn in a certain pace).
- Provide private tutoring sessions (one on one or with a small group) for beginners or even for senior programmers (in case they are looking for someone to explain the material in a short time instead of them going online and checking so many resources and waste their time [in case time is crucial to them])

- I am also thinking to write some PDF booklets that I can provide as additional materials for those attending the paid sessions (for example: summary of what we talked about during the sessions, code snippets and screenshots, guides, etc..)

- I may also provide some additional documents that are helpful for programmers (for example, if I provide a short course about freelancing, I can provide a proposal template, practical tips/ examples on how to interact with your clients,...)

- and if this worked out, I can work to keep adding new courses by learning and experimenting with new topics and then create and prepare courses and small programming projects related to that, and once I feel I understood the material well, I can provide it as a short course for you in an organized/interactive way over multiple Zoom sessions.

so these are some thoughts I have in mind, and I believe by making this a source of income for me, I can focus on it more and it will help me create even more content regarding different programming topics and enrich the Arabic programming content with high quality materials (paid and free as well).

That being said, I am interested to know your feedback regarding all that, and especially those who are willing to pay for such services (for example: would you be interested to pay for that? what topics interest you? what's the added value you are looking for which will motivate you to pay for such stuff instead of just Googling them yourself? etc..)

also I understand that the economical situation is difficult in our country, and I am definitely planning to provide special rates for those living in Lebanon which can be different from international rate, and even I am willing to take special cases into consideration and decrease the rates accordingly (for example a university graduate who doesn't have any income yet but very passionate to

learn in order to get a job, or for example teaching a single programmer VS teaching a small group of programmers, etc..)

And finally, I want you to know that it is extremely important to me that you benefit from the services I am offering, and that you don't feel like you wasted your money, I want the impression to be that the materials and the interactions were totally worth the amount paid or even that you got much more value compared to what you paid which led you to get a high paying job, so that's definitely one of my most important goals that I want to work on and it is totally part of my vision.

This is just the seed of an idea, and it is the first step, and I have a bigger vision regarding all that, but it all depends on how this goes and if there is enough interest, so I am moving with this step by step, but I am definitely looking forward for your feedback.