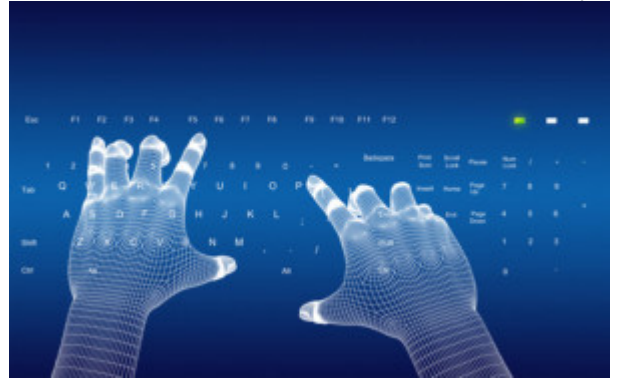


Recently I came upon this question on Quora,



What is the best way to learn programming? Every single effort that I make just goes in vain

Here's what I wrote for an answer:

Here's a really brief answer as I am really busy with programming myself.

I believe coding is more about algorithms, rather than syntax.

To approach a problem, you first need to have a solid algorithm in hand.

Once you have that you can write it in any language by little Googling.

But how to create this algorithm.

I have seen many people failing at this most important aspect of programming.

Here's my advice.

Always try to write an algorithm by observing how your Human mind approaches it.

Of course, your mind has evolved and solves a lot of problems like sorting numbers trivially, but try to think of the time, when you first learnt of sorting items based on their size.

How you looked at the size of each item, and every time you saw an item bigger than the ones you had seen before, you would make a note of that in mind.

Than you would repeat the process.

So it all comes down to breaking down how your human mind does a task, and then just writing an algorithm based on that.

Besides, not all programming is tough. In my humble opinion I believe, making an app like Instagram is an easier task, than writing applications like Mathematica, Quantum Espresso, etc.

These applications are heavily reliant on writing the best/fastest/cheapest algorithms, to do the computations in minimum time using the least resources. (Even supercomputers can take a few minutes to solve some of the mathematical/scientific problems)

This makes your programming skills better.

On the other hand one can afford to cut some slack if they are building a game as they can just say that the game requires a minimum of 16 gigs ram and a quad core processor. They don't (need to) care if they can do the same task in a better way.

So to sum up, I would recommend that you practise to build your own algorithms for a lot of available mathematical

problems. That would sharpen your mind as a programmer.

If you could tell me what level you're on in programming then maybe I could recommend you some problems.

Good Luck coding!

EDIT:

I would like to add some interesting facts that you may not know. You don't have to worry about multiplying /adding numbers in the programming languages. But ever wondered, how you would come up with an algorithm which does this extreme trivial task?

You see multiplication is just addition(done a number of times). So one can write the code for multiplication using the addition feature.

Then Division is just subtraction performed a number of times. (Also did you notice, how we broke down the problem of division and multiplication to the way we did it when we first learnt that in elementary school. Now that your mind can do it in a number of ways.)

Now the next problem is how to perform addition?

The answer is all in the circuitry. YES! I mean I don't know if it is still the way to do it or not, but this is how I have connected the dots.

You see I had a course on Digital Systems, where we built a circuit to perform additions and subtraction. We had to take care of all the carry overs/borrows etc.

Then I took a course on Microprocessor programming. Now interestingly the microprocessor 8085 had a function for addition built-in. And in my opinion, they must have done it the way I told(having an addition and subtractor digital circuit).

So you see how we have managed to create such complex and advance systems using just simple and basic circuits.

Another interesting thing to know is the NAND(Logic Gate) circuit.

Interestingly it can be used to create any logic circuit.

So how do you create this NAND circuit(gate)?

You do it by creating an AND gate using diodes, and then apply a NOT made using a transistor.

So here's how it all started:

1. Texas instruments revolutionized the digital industry by providing the transistors and other semi conductor elements.
2. These were used to build a simple NAND gate.
3. NAND gates can be used to build any other logic gate, thereby realizing an ADDER-SUBTRACTOR Circuit.
4. Microprocessors, can thus perform addition/subtraction, and with a little coding, multiplication/division.
5. And everything just BOOMED ever since.

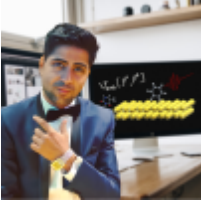
The above is just my own inference of how computer world(mainly mathematical) came up.

Of course I maybe wrong, but the above sequence, once you know it can be used to build your own microprocessor.

Of course you would need a lot more knowledge on memories, registers, counters, etc

Question link:

<https://www.quora.com/What-is-the-best-way-to-learn-programming-Every-single-effort-that-I-make-just-goes-in-vain>



Manas Sharma

I'm a physicist specializing in computational material science with a PhD in Physics from Friedrich-Schiller University Jena, Germany. I write efficient codes for simulating light-matter interactions at atomic scales. I like to develop Physics, DFT, and Machine Learning related apps and software from time to time. Can code in most of the popular languages. I like to share my knowledge in Physics and applications using this Blog and a YouTube channel.

manas.bragitoff.com/







Share this:

Click to share on Facebook (Opens in new window)

Click to share on Twitter (Opens in new window)

Click to share on WhatsApp (Opens in new window)

Click to share on Pinterest (Opens in new window)

Click to share on Reddit (Opens in new window)

Click to share on LinkedIn (Opens in new window)

Click to email a link to a friend (Opens in new window)

Click to print (Opens in new window)

Click to share on Tumblr (Opens in new window)

Click to share on Pocket (Opens in new window)

Click to share on Telegram (Opens in new window)

[wpdon id="7041" align="center"]