

# What should I learn to increase my skills? [closed]

Asked 16 years, 4 months ago   Modified 6 years, 4 months ago

Viewed 2k times



9



**Closed.** This question is [opinion-based](#). It is not currently accepting answers.



**Want to improve this question?** Update the question so it can be answered with facts and citations by [editing this post](#).

Closed 10 years ago.

[Improve this question](#)

My path to a 'fulltime'- developer stated as a analyst using VBA with Excel, Access, and then onto C#. I went to college part time once I discovered I had a passion for coding not business.

I do about most of my coding in C#, but being an ASP.NET developer I also write in HTML, JavaScript, SQL etc. . . the usual suspects.

I like to keep moving forward find the edge that will get me to the next level, the next job, and of course more

money. Most importantly I just want to learning something new and challenge me.

I have spent time recently learning LINQ, but was wondering what should I learn next? Something on the .NET Framework or a new language technology?

c#

.net

Share

Improve this question

Follow

edited Aug 18, 2018 at 4:44



Chris

4,381 ● 3 ● 27 ● 57

asked Aug 22, 2008 at 20:25



David Basarab

73.2k ● 43 ● 130 ● 157

9 Answers

Sorted by:

Highest score (default)



6



If you want to be one of the best you need to specialise. If you become very good in many skills then you may never become truly excellent in one. I know because I have taken this route myself and have found it difficult to get employment at times. After all, who wants someone who is capable at many languages when there is someone who excels at the specific thing they need. If a company develops in C# then who would want someone who is OK at C# but also is good at C, Visual Basic, Perl and Cobol,

when all they really want is the best possible C# developer for the money they can afford.

After all, you will only ever be employed for one, maybe two of your skills. There are very few jobs for people who are good in 10 or 15 skills.

If you are looking to a new skill then maybe check out the job boards and find which skills are particularly in need, but be aware that what is the flavour of the month this year may not even be on the scene next year, which will make all of that effort to learn the skill futile and wasted.

What I would say is:

1. do one thing, and do it well. This may include supporting skills (C#, ASP.Net, SQL, LINQ etc).
2. If you want to choose something else, then choose something complementary.
3. Possibly most importantly, choose something you will enjoy. Maybe Ruby on Rails is the current flavour of the month, but if you don't enjoy doing it, then don't do it. Really, it's not worth it. You will never wish, on your death bed, that you had worked more in something you didn't enjoy.

Another direction you could look at is maybe not for a particular development skill, but look for something else, maybe soft skills like people management, better business understanding or even look to something like literary skills to help improve your communications skills.

All of these will help to allow you to do what you want to do more, and cut down on the stuff you really don't enjoy, thus helping to make your job more enjoyable.

Apologies for the waffling here. Hope you are still awake :)

Share Improve this answer

Follow

edited Aug 1, 2013 at 16:51



Daniel Daranas

22.6k ● 9 ● 65 ● 121

answered Aug 22, 2008 at 21:13



Xetius

46.6k ● 25 ● 91 ● 124


---

3 Although I agree that you should focus on a particular language, I still think that if you are versed in lower level languages such as C and C++ it will make you a better programmer in higher level languages such as C# or Java...so I still think that you should at least try to learn other languages. Other than that, not every language is suitable for every particular job, so the more the know, the more you can apply for specific tasks...which in turn will make you more productive, and ofcourse employers like that.

– [Andreas Grech](#) Aug 12, 2009 at 6:44

---

1 I agree that specialization is important but I don't entirely agree with you. I worked in the data capture group at an analytics company and our group had SDK's for all the mobile platforms (there's 3 languages and 4 runtime environments alone) as well as projects in C#, C++, Java and JavaScript. Any developer with truly good fundamentals should be able to become proficient in any language/runtime fairly quickly. I still think you should specialize but a deep understanding of software is more valuable than know .NET

well for example. I would still encourage people to learn many languages. – [evanmcdonnal](#) Aug 1, 2013 at 17:04 

---



Yeah, the more I get into software, I start to see myself focusing less on the language and more on the design..

2



Yeah there are framework bits we need to get our head around but most of the time ( *most* not all ) you can look those up as-and-when you need them..



But a good design head? That takes years of experience to start getting it working right..

And that is what the companies really pay for.. "Build it and they will come" and all that...

Share Improve this answer

answered Aug 22, 2008 at 20:30

Follow



[Rob Cooper](#)

28.9k ● 26 ● 105 ● 142



1



As you continue to gain more experience in ASP.Net, C#, etc - it's always good to go check out the competition and see if it sparks ideas on how you can do things better in what you're doing. Taking a look at something like Rails or Django might change how you look at designing or building your apps.



Share Improve this answer

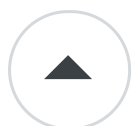
answered Aug 22, 2008 at 20:34

Follow



RyanBrady

6,943 ● 4 ● 28 ● 32



1



If you're now proficient with the languages and technologies you use, then start spending more time focusing on the design, solution architecture, and systems integration. The "bigger picture" that will set you apart from your contemporaries.



Check out some Martin Fowler books like "Patterns of Enterprise Application Architecture", or Eric Evans' "Domain-Driven Design".

Share Improve this answer

answered Aug 22, 2008 at 20:37

Follow



Ian Nelson

58.6k ● 20 ● 78 ● 104



1

Maybe learn more about Usability (best practices, testing, etc.) if you haven't already done so.



Steve Krug's "Don't Make Me Think" is a good book to start with. [Jakob Nielsen](#) always has interesting stuff as well.



Share Improve this answer

answered Aug 22, 2008 at 20:38

Follow



[upheaval](#)

33 ● 2 ● 6

---



1



The more languages you know, the more marketable you are. Look and see what the more popular (market for, not fan base) languages are, then add on some cutting edge tech that is not in much use yet, rounded out by general programming skill.

With your skill set I would recommend (as far as languages):

- Java as a starting point
- For .Net add in the .Net MVC (you have LINQ or that would be here also)

Language agnostic skills:

- Design Patterns (includes the MVC)
- Domain Driven Design
- Test Driven Design

Share Improve this answer

Follow

answered Aug 22, 2008 at 20:53



[Carlton Jenke](#)

3,015 ● 3 ● 27 ● 30



1



Here would be my suggestions:

1) Design Patterns - These are really neat as well as being very useful in some situations.

2) AJAX - Assuming you haven't already done some of this, it is an interesting part of Web Development from my





view.



3) Determine which parts of the chain do you enjoy the most: Front-end work(HTML, CSS, Javascript), middleware(C# for business logic parts), or back-end(MS-SQL with stored procedures, indexes, triggers, and all that stuff). If it is all of it then try to stay where the team doing web development is small as otherwise you may be asked to choose.

4) Algorithm design and analysis - Do you know various sorting algorithms? Do you know various techniques to create an algorithm, e.g. greedy, recursion, divide and conquer, dynamic programming, using custom data types like heap in heapsort etc. This can be new and cool.

5) Determine if there is a part of the development process you favor: Analyst, designer, programmer, tester, debugger? All can have varying degrees of being near the code, IMO.

Share Improve this answer

answered Oct 24, 2008 at 18:30

Follow



JB King

11.9k ● 4 ● 40 ● 49



@ Michael DSL=[Domain Specific Language](#)

0

As for what you should learn, that depends on what you're interested in.





Are you looking to challenge yourself while staying in the same medium (web-centric applications)? I would suggest learning about Apache and the LAMP (Linux, Apache, MySQL, PHP) architecture and challenge yourself to build a web application that you could readily build with ASP .NET using it.

Want to learn something completely different? Try [Prolog](#) or [LISP](#) and see what you can do with those.

Maybe you'd like to get into embedded software? Learn C to start.

You have a wide variety of ways to improve your skills, and each one has career paths attached to them. (Well, maybe not Prolog, but it's fun!)

Share Improve this answer

answered Aug 22, 2008 at 20:51

Follow



Marc Reside

2,956 ● 1 ● 24 ● 26



-1



Why don't you swap stacks and look at the LAMP stack?

Or how about a functional language like haskell?

Or write a DSL?

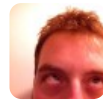
Or an app for your phone?



Share Improve this answer

answered Aug 22, 2008 at 20:31

Follow



[Johnno Nolan](#)

29.6k ● 19 ● 113 ● 160

---

What is the point of swapping the stack? Whichever that is?

– [Andrei Rînea](#) Oct 8, 2008 at 9:20

---