

What skills should a programmer have nowadays? [closed]

Asked 16 years, 1 month ago Modified 11 years, 9 months ago

Viewed 16k times



10



Closed. This question is [off-topic](#). It is not currently accepting answers.

💡 **Want to improve this question?** [Update the question](#) so it's [on-topic](#) for Stack Overflow.

Closed 11 years ago.

[Improve this question](#)

I have been programming using PHP, Javascript and have been a designer for a few years in a company. I am thinking of leaving I am wondering what skills do I need to have nowadays to become successful.

windows

Share

Improve this question

Follow

edited Mar 27, 2013 at 14:32



LittleBobbyTables - Au Revoir

32.7k ● 25 ● 111 ● 115

asked Nov 17, 2008 at 3:33



Teej

12.9k ● 9 ● 75 ● 94

9 Answers

Sorted by:

Highest score (default)



7

You seem to be inquisitive and interested in learning. That's qualification #1; things change fast in this field, and only the life-long learners survive.



Next step would be to scour job boards in your area ([Monster](#), [Craigslist](#), [Joel On Software Jobs Board](#), etc.)



and keep track of the skills you see coming up a lot.



That's probably a fairly good representation of what's "hot", at least near you.



Then, you can check here on stack overflow by tags (e.g. [PHP](#) and [javascript](#)) and see if the kinds of questions that people are asking make basic sense to you (not that you need to know every answer, but if it sounds like Greek, you're probably not ready to get a job in that subject).

To more directly answer your question: there are web jobs out there for PHP and javascript, though a heavy dose of CSS and Ajax probably wouldn't hurt if you're going that direction. A design background will help if you want a general "webmaster" type job, but note that those are often NOT programming-oriented jobs, because most stuff you can do on an average website is typically better done with an off the shelf CMS (you don't usually need to reinvent the wheel unless the web site you're doing is

highly functional in some way). As other posters have said, C# / .NET is probably good if you want something more corporate.

The real question is: what kind of work do you like doing? What makes you happy? Find that and do it well, and you'll get a job. :)

Share Improve this answer

Follow

edited May 23, 2017 at 12:19



Community Bot

1 • 1

answered Nov 17, 2008 at 3:54



Ian Varley

9,437 • 5 • 31 • 34

-
- 3 Don't forget Dice. That's probably the most important job board out there. Actually, by doing dice search on a technology, you know how hot it is most of the time.
– Charles Graham Nov 17, 2008 at 4:34
-



6



It really depends on what you intend to do.

I believe the following skills are important for all types of "programmers" regardless of their chosen language.

1. Accurate Estimation. Giving your boss accurate estimates makes their job easier, which makes your life much easier.
2. Communication. Being able to communicate effectively to everyone you work with is a must.

3. Knowing when to ask for help. Like you wrote, you can't know everything. I would rather have a developer ask a colleague for help, than try to spend days coming up with a solution.
4. Design Patterns. Every developer needs to understand basic design patterns.
5. Know a good scripting language. Perl, Python, Ruby, Bash, Powershell...take your pick.

Share Improve this answer

edited Nov 18, 2008 at 4:46

Follow

answered Nov 17, 2008 at 3:49



Alan

46.8k ● 19 ● 117 ● 136



5



I think one of the best assets a developer can have is to know how to use Google. Stay with me here.

I've worked with people who just don't know how to search for pre-existing solutions. When faced with any software project, I've found that half of the problem is solved by simply knowing what to type into Google. I know of the existence of a lot of different technologies, how they're intended to be used, and general chatter about them that I've read on other people's blogs, but I've never actually used them and maybe never will. But when a problem that fits their description comes my way, I know

what magic words to type into Google and what to start learning next for that particular project.

The other half of the problem is deciding what to take away from those Google results. [CodeProject](#) is a fantastic site to explore, and sometimes you'll hit an article that addresses your problem head on. Sometimes, you'll be able to use the provided code directly in your project (I did this with a Windows Forms LED digit control that I got there), but most of the time, you'll just take away the ideas of the article and use it as a guide to apply to your project. Sometimes you'll look at an article and decide that the person doesn't know what they are talking about. *Being able to assess the quality of, use, adapt, or simply be inspired from existing blog articles, libraries, and snippets is an invaluable asset.* You don't have to understand all of it--you can't know everything, you have to draw the "magic line" somewhere--but it's important that you're able to tell what looks well-written and what looks slap-dash and error-prone.

So, to that end, my best advice is to just load up your RSS reader with feeds from people who are working in the technologies that you are interested in. You don't have to read every individual posting, but you'd be amazed at how much you can pick up by just zooming by them all and glancing at the headlines each day.

To me, things like the programming language aren't really that important. I started with C++, dabbled in Java, worked for two years in PHP, and have been living and

breathing C# for another two. You can learn a programming language in a week. But learning the frameworks and idioms commonly used with that language, however, can take years, so for any such framework you can be sure that there will be some jobs somewhere involved with it. After bouncing around for awhile, you'll find one that you like (mine is the .NET world), but you'll still know your way around the others. So pick your poison, but keep reading to stay aware of what's out there. A *well-indexed* programmer can easily become a *well-rounded* one because he'll know how to go about tackling new problems that come up outside of his comfort zone.

Hope some of that makes sense.

Share Improve this answer

answered Nov 18, 2008 at 5:09

Follow



Nicholas Piasecki

25.6k ● 5 ● 82 ● 92



4



A portfolio of your work, lots of pictures, very short descriptions. PHP isn't as popular in the corporate world, but it does exist. You might want to consider learning JSP or C# ASP.NET to keep your options open.



Different companies have different needs, webdesign shops will be focused heavily on design and how fast you can make a site (usually using a framework). Corporate apps will be concerned with things such as bug fixing, testing, support multiple languages, using 3rd party controls.

Share Improve this answer

Follow

answered Nov 17, 2008 at 3:39



TravisO

9,540 ● 4 ● 39 ● 45



1

All you need is a finished study and capability to learn new things easily. They can't expect you know everything they tell you to do. But if you can learn well, that's going to be fine.



That's what i have been told in university. I think it holds true.



Share Improve this answer

Follow

answered Nov 17, 2008 at 3:38



Johannes Schaub - litb

506k ● 131 ● 917 ● 1.2k



0

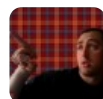
Learn test driven development and unit testing and have solid knowledge of a few javascript libraries.



Share Improve this answer

Follow

answered Nov 17, 2008 at 3:51



wusher

12.4k ● 22 ● 73 ● 96



-
- 2 I have 7 years experience and I have never had a job where this is required. It's hard enough to find jobs where this is even used. Trust me, I've had my eye on TDD for a while, and it's awesome, but not that many places use it unfortunately.
– [Charles Graham](#) Nov 17, 2008 at 4:32
-

Programming 40 years, and never in a shop that admitted using TDD (though some did, to a degree, by accident).
– [Hot Licks](#) Mar 27, 2013 at 14:37



0

If you already know PHP and Javascript, ensure your CSS and MySQL skills are up-to-scratch. The four make a nice package.



Share Improve this answer

answered Nov 17, 2008 at 3:58

Follow



[Dean Rather](#)

32.3k ● 15 ● 68 ● 72



0

In web development, at least (where I've mostly worked) there's often a division between front-end and back-end skills.



I'm speaking of coding here, so by "front-end" I mean things like CSS and JavaScript. There's often a fuzzy line here between design and coding, so people working on the front-end often have a bit of coding skill, a bit of design skill.



Back-end tends to be more java, or .net (most common for web, at least by the job ads I've seen) with a bit of python, ruby, php, and other. Likely the skills in the latter will more often land you in a smaller place. It's been my observation that the back-end skills are likely to pay a bit better. You might do well to look for somewhere where you'll do a bit of the front and a bit of the back and look to transition to doing more of the programming-type work.

Of course "success" is a very personal thing, so please disregard everything written above, and best of luck.

For some actual hard data, here's one site that ranks languages by popularity in job postings (there are others) <http://www.langpop.com/>

Share Improve this answer

answered Nov 17, 2008 at 4:14

Follow



Steve B.

57.2k ● 12 ● 97 ● 134



0



Basically, you need to understand that programming is not a lazy man's profession, even though it *is* far too sedentary. It requires *thought*, and it requires *practice*, and it requires *patience*. If you think you can learn a bunch of cookbook approaches and suddenly be a programmer, you're sadly mistaken.



(Though, unfortunately, many managers are convinced that there *are* cookbook solutions and base their hiring on what buzzwords you can recite. But you asked what skills

a programmer *should* have, not what he needs to know to be hired.)

Share Improve this answer

answered Mar 27, 2013 at 14:41

Follow



Hot Licks

47.7k ● 18 ● 94 ● 154
