

Forum:[Hack Forums](#)

Thread: [Ruby on Rails or PHP? Future Career Advice Needed](#)

madcrazydrumma - March 25, 2015, 3:24 pm

Hey everyone!

The title pretty much suggests this... so here goes ?

I plan to be a web developer after doing my Computer Science degree in 3 years, but I am wondering what key skills I should learn to help boost my CV and online portfolio.

I already know quite a lot of PHP but I have never touched upon Ruby on Rails. I have however done quite a bit of Ruby itself as a standalone language.

The big question... What would be more useful to a hiring company in the near future: PHP or Ruby?

As well as this, what other skills would you recommend me learning for the next 3 years that would be of more use to me and my CV?

Thanks ?

Exemplification - March 25, 2015, 6:27 pm

You'll probably find jobs with older more stable businesses with PHP while RoR is more popular by startup. From my observations that is what it seems like anyway.

Neither are going anywhere in the next several years and both are skills that are sought after. Changing your focus just because it might look good on a CV is just going to reduce what you learn in depth about a language and having that in depth knowledge may be the deciding factor on a job.

Tasemu - March 25, 2015, 9:39 pm

I find it usually goes ruby for startups, php midsize then ruby again for monolithic sized platforms. I think either are fine to go for though I am slightly biased towards Ruby because I prefer it significantly to develop in.

madcrazydrumma - March 26, 2015, 1:54 am

Are there any major differences that would make Ruby a more preferred language to use?

Haxalot - March 26, 2015, 4:10 am

No one can answer this question because each company has their own technology preferences. And there have been endless debates of what languages should be used for server-side application development - most of the time it simply comes down to personal preference.

You said that you would like to be a web developer after graduating in three years time. So why not familiarise yourself with both during that time? If you're already familiar with PHP, then why not continue with that for now (and perhaps look into a couple of frameworks like Laravel), and then take a look into Ruby and its Rails framework for a different approach to software development. It will be far more beneficial to you to look into both (as well as other web development

languages/frameworks too!).

madcrazydrumma - March 26, 2015, 6:23 am

(Mar 26, 2015 04:10 AM)Haxalot Wrote: [\[CITE\]](#) No one can answer this question because each company has their own technology preferences. And there have been endless debates of what languages should be used for server-side application development - most of the time it simply comes down to personal preference.

You said that you would like to be a web developer after graduating in three years time. So why not familiarise yourself with both during that time? If you're already familiar with PHP, then why not continue with that for now (and perhaps look into a couple of frameworks like Laravel), and then take a look into Ruby and its Rails framework for a different approach to software development. It will be far more beneficial to you to look into both (as well as other web development languages/frameworks too!).

Do companies nowadays tend to use web development frameworks such as Laravel? If so what frameworks would recommend I look into?

I'm currently using the SASS (SCSS) framework for css and JQuery/AJAX for JavaScript

~Dec - March 26, 2015, 6:52 am

I'll sum this up for you.

If you want to enjoy your work, and push code quickly, learn Ruby.

If you want to hate your work, but have more job availability by being in high demand, learn PHP.

If you want to enjoy your work, be in high demand, be on the bleeding edge and have fun while learning, learn Node.js

If you want to be an amazing programmer, learn all three.

Of the frameworks that companies tend to use nowadays in all systems, based on experience I've had working as a professional:

PHP - Laravel

Ruby - Ruby on Rails, Sinatra

Node.js - Express, Hapi, Restify, Sails.js

There's also a lot of traction in regards to Meteor, and while "node-like" it isn't exactly Node.

On the frontend, there's lots of demand for people who know Backbone.js, AngularJS and Ember.js - however, Facebook's React.js is garnering incredible amounts of interest at an incredibly fast pace.

madcrazydrumma - March 26, 2015, 6:58 am

(Mar 26, 2015 06:52 AM)~Dec Wrote: [\[CITE\]](#) I'll sum this up for you.

If you want to enjoy your work, and push code quickly, learn Ruby.

If you want to hate your work, but have more job availability by being in high demand, learn PHP.

If you want to enjoy your work, be in high demand, be on the bleeding edge and have fun while learning, learn Node.js

If you want to be an amazing programmer, learn all three.

Of the frameworks that companies tend to use nowadays in all systems, based on experience I've had working as a professional:

PHP - Laravel

Ruby - Ruby on Rails, Sinatra
Node.js - Express, Hapi, Restify, Sails.js

There's also a lot of traction in regards to Meteor, and while "node-like" it isn't exactly Node.

On the frontend, there's lots of demand for people who know Backbone.js, AngularJS and Ember.js - however, Facebook's React.js is garnering incredible amounts of interest at an incredibly fast pace. That's the best answer I've got! Thank you ?

I'll workout a plan to fit in with university so I can further develop my skills, I like the sound of Node.js so I'll look into it

Haxalot - March 26, 2015, 8:04 am

(Mar 26, 2015 06:52 AM)~Dec Wrote: [\[CITE\]](#) If you want to hate your work, but have more job availability by being in high demand, learn PHP.

High demand, yes. Hate your work, what? I'm familiar with a number of languages, and PHP is by far my most favourite. This point simply comes down to personal preference, so please don't spread falsities like this around.

(Mar 26, 2015 06:52 AM)~Dec Wrote: [\[CITE\]](#) If you want to be an amazing programmer, learn all three.

Agreed.

(Mar 26, 2015 06:52 AM)~Dec Wrote: [\[CITE\]](#) PHP - Laravel ZF2 and Symphony are also quite popular candidates.

Warren Buffett - March 26, 2015, 8:40 am

Definitely start with PHP. RoR is slower than PHP but still as extensive. I would definitely start with PHP first.

~Dec - March 26, 2015, 10:49 am

(Mar 26, 2015 08:04 AM)Haxalot Wrote: [\[CITE\]](#) High demand, yes. Hate your work, what? I'm familiar with a number of languages, and PHP is by far my most favourite. This point simply comes down to personal preference, so please don't spread falsities like this around.

Sorry if I'm a little disdained when I speak of languages I find just a little "unimpressive", it's just that I got my certification from Zend before Laravel even came out - back then the inconsistencies in namespaces were a huge pain in the ass, the on-by-default security-impeding "easter eggs" were cause of much headache and the language just wasn't very intuitive in general, I felt embarrassed whenever I wrote code in PHP... even though I was following best practice it was just a mess to work with. And a lot of people shared that opinion at the time. So what I was saying wasn't exactly a "falsity" - but I don't discourage people from learning it.

If PHP is your favourite language though, then I find you weird. That's kind of like being the person who's favourite vegetable is brussel sprouts. Just my opinion though. Don't take it to heart.

Haxalot - March 26, 2015, 12:03 pm

(Mar 26, 2015 10:49 AM)~Dec Wrote: [\[CITE\]](#) If PHP is your favourite language though, then I find you weird. That's kind of like being the person who's favourite vegetable is brussel sprouts. Just my opinion though. Don't take it to heart.

Maybe I am a little weird :) I'm just so much more fluent in PHP than in other languages I use, which makes building PHP-based applications such an easy thing to do. This, to me, makes it an enjoyable experience because I don't face the same difficulties I do with other languages, like debugging issues, knowledge limitations on language features or tools, etc.

And don't worry, I don't like brussel sprouts...

Ezchan - March 26, 2015, 3:09 pm

If you plan to join a startup or new company after school ruby is more widely used, but php is also very prominent.

Crypto™ - March 26, 2015, 5:14 pm

(Mar 26, 2015 08:04 AM)Haxalot Wrote: [\[CITE\]](#) High demand, yes. Hate your work, what? I'm familiar with a number of languages, and PHP is by far my most favourite. This point simply comes down to personal preference, so please don't spread falsities like this around.

As an ex-PHP user, I wish I had never learned it, not because it was bad per sé, but because I learned practices that most developers would probably laugh at me for. I throw this article around a lot, but [PHP: A Fractal of Bad Design](#) is well worth a read regardless of your stance on PHP. Same with [PHP Sadnesss](#). Sure, it might be a language you like personally, but it's syntactically, it's terrible. Laravel is one of the few things that makes it bearable for me.

(Mar 26, 2015 06:52 AM)~Dec Wrote: [\[CITE\]](#) I'll sum this up for you.

If you want to enjoy your work, and push code quickly, learn Ruby.

If you want to hate your work, but have more job availability by being in high demand, learn PHP.

If you want to enjoy your work, be in high demand, be on the bleeding edge and have fun while learning, learn Node.js

If you want to be an amazing programmer, learn all three.

Of the frameworks that companies tend to use nowadays in all systems, based on experience I've had working as a professional:

PHP - Laravel

Ruby - Ruby on Rails, Sinatra

Node.js - Express, Hapi, Restify, Sails.js

There's also a lot of traction in regards to Meteor, and while "node-like" it isn't exactly Node.

On the frontend, there's lots of demand for people who know Backbone.js, AngularJS and Ember.js - however, Facebook's React.js is garnering incredible amounts of interest at an incredibly fast pace. This is so true. I'm a huge fan of Node.js myself, and haven't used anything else for a while, though Ruby has been on my list to learn for a while. You really should experience all three, but if you go into PHP, know that you will learn bad practices. (Practices that I'm still trying to unlearn)

Tasemu - March 26, 2015, 7:39 pm

I have to agree with Dec on pretty much every point he has made. I used to like php a lot too before I found the other languages. Now I never look back.

Haxalot - March 27, 2015, 4:31 pm

(Mar 26, 2015 05:14 PM)Crypto™ Wrote: [\[CITE\]](#) As an ex-PHP user, I wish I had never learned it, not because it was bad per sé, but because I learned practices that most developers would probably laugh at me for. I throw this article around a lot, but [PHP: A Fractal of Bad Design](#) is well worth a read regardless of your stance on PHP. Same with [PHP Sadnesss](#). Sure, it might be a language you like personally, but it's syntactically, it's terrible. Laravel is one of the few things that makes it bearable for me.

Sigh. I've read these resources - and others - that complain about PHP's imperfections. It's not exactly something new or surprising - every language has many (many) flaws.

Would you care to share some of the practices you learned in PHP that would make other developers laugh at you? And perhaps give some examples of its terrible syntax? It's just to satisfy my curiosity more than anything else really.

Crypto™ - March 28, 2015, 1:21 am

(Mar 27, 2015 04:31 PM)Haxalot Wrote: [\[CITE\]](#) Sigh. I've read these resources - and others - that complain about PHP's imperfections. It's not exactly something new or surprising - every language has many (many) flaws.

Would you care to share some of the practices you learned in PHP that would make other developers laugh at you? And perhaps give some examples of its terrible syntax? It's just to satisfy my curiosity more than anything else really.

- 1) Mixing business and display logic without templating. (PHP actually encourages this behaviour by default.)
- 2) Suppressing errors by silencing them, rather than fixing them (even though some stupid shit causes errors/warnings)
- 3) No concurrency, because no support for it. (This is both a bad practice, and a language flaw. Concurrency results in much faster code, since you can carry on in a separate thread while particularly weighty code executes. I wrote a script that had to parse a large amount of web pages. Without concurrency, it took almost two hours to go through 1000 entries. With concurrency, it took approximately fifteen minutes.)

PHP is particularly destructive to beginners in my opinion, since an experienced developer knows how to avoid these pitfalls, but novice developers are coaxed into a very linear approach, which, unless you learn about modularizing code, properly handling errors, sanitizing user input, and a laundry list of other tasks that simply aren't as prevalent in other languages. I'm not saying that PHP's the only offender, but in my experience, it's significantly more susceptible to these pitfalls.

I have a weird thing with Cron Jobs as well for executing PHP tasks. I feel like it's super hacky, and not a good way to deal with background work.

Haxalot - March 28, 2015, 6:39 am

(Mar 28, 2015 01:21 AM)Crypto™ Wrote: [\[CITE\]](#) 1) Mixing business and display logic without templating. (PHP actually encourages this behaviour by default.)

I realise PHP makes this easy to do by enabling you to inline it within other code, effectively also making it a native templating language. But to say that it encourages this behaviour isn't exactly true - and it certainly far from promotes it either.

(Mar 28, 2015 01:21 AM)Crypto™ Wrote: [\[CITE\]](#) 2) Suppressing errors by silencing them, rather than fixing them (even though some stupid shit causes errors/warnings)

PHP's different error level handling is really one of its strengths. It does not force a developer to run strict, standards compliant code. It simply lets the developer choose what error reporting level they wish to run, and instead recommends the code base run cleanly in the highest error reporting level possible (E_ALL). This is a strength because it acts as a guide to writing cleaner, future-proof code rather than acting as an enforcement (which isn't as programmer-friendly).

(Mar 28, 2015 01:21 AM)Crypto™ Wrote: [\[CITE\]](#) 3) No concurrency, because no support for it. (This is both a bad practice, and a language flaw. Concurrency results in much faster code, since you can carry on in a separate thread while particularly weighty code executes. I wrote a script that had to parse a large amount of web pages. Without concurrency, it took almost two hours to go through 1000 entries. With concurrency, it took approximately fifteen minutes.)

It's not baked into the core, but don't dismiss the [pthreads](#) extension.

Personally, I feel the predominance of crappy tutorials out there for PHP is more of the problem than the language itself - but maybe I'm just biased :)

Crypto™ - March 28, 2015, 2:22 pm

(Mar 28, 2015 06:39 AM)Haxalot Wrote: [\[CITE\]](#) I realise PHP makes this easy to do by enabling you to inline it within other code, effectively also making it a native templating language. But to say that it encourages this behaviour isn't exactly true - and it certainly far from promotes it either.

PHP's different error level handling is really one of its strengths. It does not force a developer to run strict, standards compliant code. It simply lets the developer choose what error reporting level they wish to run, and instead recommends the code base run cleanly in the highest error reporting level possible (E_ALL). This is a strength because it acts as a guide to writing cleaner, future-proof code rather than acting as an enforcement (which isn't as programmer-friendly).

It's not baked into the core, but don't dismiss the [pthreads](#) extension.

Personally, I feel the predominance of crappy tutorials out there for PHP is more of the problem than the language itself - but maybe I'm just biased :)

That's definitely possible. I know my knowledge of it was just glommed together from a bunch of mediocre tutorials, so it's entirely possible. I'm always going to prefer Node.js to PHP though :P

JavaProphet - March 28, 2015, 7:12 pm

I would highly recommend you check out www.avuna.org

It's java, but faster and less resource intensive than PHP & RoR put together doubled.

Haxalot - March 29, 2015, 2:42 am

(Mar 28, 2015 07:12 PM)JavaProphet Wrote: [\[CITE\]](#) I would highly recommend you check out www.avuna.org

It's java, but faster and less resource intensive than PHP & RoR put together doubled.

That's quite a statement - you wouldn't happen to have any figures on performance comparisons between the three with real world apps?

Crypto™ - March 29, 2015, 10:05 am

I got curious today about the number of available jobs for PHP vs Ruby on Rails, so I did some research:

Indeed.com

Query: title:PHP - 205 Results

Query: title:Ruby on Rails - 276 Results

Query: title:Rails - 299 Results

Careers.StackOverflow.com

Query: PHP & type: Permanent - 368 Results

Query: Ruby on Rails & type: Permanent - 407 Results

Query: Rails & type: Permanent - 237 Results

TheMuse.com

Query: PHP - 123 Results

Query: Ruby on Rails - 2022 Results

Query: Rails - 53 Results

Lever.co

I used Google to quickly check their listings:

Query: "PHP" site:*.lever.co - 488 Results

Query: "Ruby on Rails" site:*.lever.co - 123 Results

Query: "Rails" site:*.lever.co - 220 Results

Greenhouse.io

Once again, Google:

Query: "PHP" site:boards.greenhouse.io - 542 Results

Query: "Ruby on Rails" site:boards.greenhouse.io - 192 Results

Query: "Rails" site:boards.greenhouse.io - 277 Results

Angel.co

Query: Job Type: Full-time & Role: Backend Developer & PHP - 223 Companies

Query: Job Type: Full-time & Role: Backend Developer & Ruby on Rails - 204 Companies

Query: Job Type: Full-time & Role: Backend Developer & Rails - 205 Companies

Query: Job Type: Full-time & PHP - 842 Companies

Query: Job Type: Full-time & Ruby on Rails - 977 Companies

Query: Job Type: Full-time & Rails - 1001 Companies

This is in no way comprehensive, but should provide some interesting stats.

mallard - March 29, 2015, 10:07 am

Ruby is for stupid gay hipsters that sit in Starbucks on the MacBooks while they work on their shitty startup with an .io domain of a word that normally ends in er and they take out the e

PHP is for manly men.

~Dec - March 29, 2015, 10:49 am

(Mar 29, 2015 10:07 AM)mallard Wrote: [\[CITE\]](#) Ruby is for stupid gay hipsters that sit in Starbucks on the MacBooks while they work on their shitty startup with an .io domain of a word that normally ends in er and they take out the e

PHP is for manly men.

Github

Groupon

Heroku

Hulu

Twitter

CodePen

Stripe

37Signals

Basecamp

HipChat

CampFire

YellowPages

All these companies were started by the "Hipsters" you speak of.

mallard - March 29, 2015, 10:50 am

(Mar 29, 2015 10:49 AM)~Dec Wrote: [\[CITE\]](#) Github

Groupon

Heroku

Hulu

Twitter

CodePen

Stripe

37Signals

Basecamp

HipChat

CampFire

YellowPages

All these companies were started by the "Hipsters" you speak of.

damn hipsters

i was a hipster before it was cool

Strider64 - March 29, 2015, 1:23 pm

I'm a dinosaur when it comes to programming, although I really haven't become serious about programming since the last few years. I started out with COBOL that had to deal with punch cards, if anyone knows about punch cards you know the anguish. ?

Anyway PHP -vs- Ruby on Rails is kind of silly, any programming language can be poorly coded. Though I do agree being a PHP developer that PHP lends itself to a lot of bad code floating around the net. An to throw in the mix there is a lot of obsolete code that uses mysql that for some strange reason beginners seem to be drawn to. A lot of them will change over to mysqli or PDO once they are notified they are using obsolete code, but you always have the few knuckleheads "I'll change over once I learn PHP"! Why learn something that is going to be obsolete? I never understand this way of thinking. Anyways this is not to bash PHP or Ruby, but just to give my opinion that both are just a tool in a developers toolbox. ?