

[00:00:00] Speaker A: Okay. So, you know, you guys send in some really fascinating stuff and we try to cover as much as we can.

[00:00:04] Speaker B: Yeah.

[00:00:04] Speaker A: And somebody recently sent this over.

It's a guide from August Infotech Home.

[00:00:10] Speaker B: Okay.

[00:00:11] Speaker A: It's dated February 14, 2024.

[00:00:13] Speaker B: Right, right.

[00:00:14] Speaker A: And it's all about upgrading PHP for web development.

[00:00:17] Speaker B: Right.

[00:00:18] Speaker A: And we thought, you know, given that so much of the web runs on php, this would be, you know, really important area to kind of dive into.

[00:00:26] Speaker B: Absolutely. And I think this one is particularly interesting because it's focused on upgrading from PHP 7 to PHP 8.1. And that's not just, you know, chasing the latest shiny number, it's understanding why this specific upgrade is so crucial for your web presence in terms of security, in terms of performance, in terms of just, you know, future proofing everything.

[00:00:43] Speaker A: Yeah. And I think the, the guide reminds us right off the bat, you know, that PHP 7.4 actually reached its end of life back in November of 2022. So that really establishes PHT 8. And specifically, specifically 8.1 in this case is the current standard.

[00:00:58] Speaker B: Right.

[00:00:58] Speaker A: And with VHP 8.2 already out there, I mean, this is moving quickly.

[00:01:02] Speaker B: Yes, it is. Things are changing.

[00:01:04] Speaker A: I think for this deep dive, what we want to do is really go through this guide, pull out the most important insights, the actionable steps, so that you listening can really grasp what are the advantages, what are the potential hiccups and what's the actual, you know, process of upgrading to php 8.1 look like?

[00:01:25] Speaker B: Sounds good to me. Okay, let's do it.

[00:01:27] Speaker A: So let's jump right in. The guide starts off, you know, really making it clear that PHP8 is not just some minor update. I call it a substantial transformation.

[00:01:37] Speaker B: Yeah, they're not messing around.

[00:01:39] Speaker A: Yeah, they're not kidding around. It's not like you're just doing a point release here.

[00:01:42] Speaker B: No, it's a big deal.

[00:01:43] Speaker A: This is a major shift.

[00:01:44] Speaker B: Right.

[00:01:45] Speaker A: So they list out some of the new features and improvements and there's a lot of them.

[00:01:50] Speaker B: There are a lot.

[00:01:51] Speaker A: We've got things like named arguments, attributes. Yeah. Which sound kind of like secret labels for your code.

[00:01:56] Speaker B: Right. Like little tags you can add to give it more meaning.

[00:01:59] Speaker A: Union types, constructor, property promotion.

[00:02:02] Speaker B: Yeah, and these are all really about like behind the scenes stuff that empowers developers to write code that's cleaner, that's more expressive and ultimately easier to maintain.

[00:02:11] Speaker A: Right.

[00:02:11] Speaker B: For example, like named arguments, they're like labels for your function input. So instead of just having a bunch of values in a row, you can actually say, okay, this value is for this parameter, this value is for that parameter. And it just makes your code way easier to read and less likely to break if like the order of those inputs ever changes. Oh, okay. Or like attribute, like built in metadata. So like a framework might use an attribute to mark a specific function as an API endpoint, for example, which can make routing configuration much cleaner.

[00:02:43] Speaker A: Okay, that makes a little more sense, I think.

[00:02:45] Speaker B: Yeah, it's a little abstract, but it's powerful stuff.

[00:02:48] Speaker A: Yeah. And then there's the match expression, the null safe operator, which sounds like a safety net for when things might go wrong in your code, right?

[00:02:55] Speaker B: It is. It's like a little insurance policy for your, for your variables.

[00:02:59] Speaker A: And then this, this super fast thing called JIT compilation.

[00:03:03] Speaker B: Ah, yes, the jit or just in time compilation. This is a real game changer for performance.

So basically it compiles parts of your PHP code into machine code while the application is running. Yeah, which can lead to some serious spill.

I mean, the guide even mentions tests showing around a 25% PHP8 speed boost.

[00:03:21] Speaker A: 25%? I mean that's just huge for anybody listening, that's not just a marginal improvement. That's noticeable faster loading times for your visitors and that can have a huge impact on things like user experience, bounce rates, even search engine rankings.

[00:03:36] Speaker B: Yeah, Google loves fast websites.

[00:03:38] Speaker A: So I think the point here is moving to PHP8 isn't just about, you know, bragging rights that you've got the latest version number. It really translates to tangible benefits. Tangible benefits. Faster, more secure, more efficient websites for you and for your users.

[00:03:53] Speaker B: For everybody. Okay, so, so that brings us to the big question. Yeah, the why.

[00:03:57] Speaker A: Why upgrade to PHP8?

[00:03:59] Speaker B: Right. And the guide really frames this not as like a suggestion, but as a necessity.

[00:04:04] Speaker A: Okay, so let's talk about why. Why is this upgrade so crucial right now?

[00:04:09] Speaker B: Okay, well, the first and probably the most pressing reason is long term support. PHP7. It's no longer supported by the PHP community, which means no more security updates. Oh, so PHP8 on the other hand, is actively supported. It gets all the security patches, all the important updates. Think of it like this. Running an unsupported version of PHP is like leaving the front door to your website wide open.

[00:04:32] Speaker A: It's kind of a stark analogy, but it makes sense. You're just leaving yourself vulnerable to anything that comes along.

[00:04:37] Speaker B: You're basically saying, come on in, hackers, I'm ready for you. Yeah, it's like running outdated antivirus software on your computer.

[00:04:44] Speaker A: Right. You're just asking for trouble.

[00:04:46] Speaker B: Exactly. So that leads us right into the second point. Improved security. Now PHP8 has a much more robust security posture overall. It's got better data protection features, better encryption support. You know, in today's world, where security breaches are happening all the time, this is non negotiable.

[00:05:03] Speaker A: Yeah, I mean, nobody wants their website to get hacked or their user's data compromised because they're running some ancient version of php.

[00:05:10] Speaker B: Oh, you don't want to be that guy.

[00:05:13] Speaker A: So what other key reasons are there?

[00:05:14] Speaker B: Okay, well, better error handling is another big win. PHP8 gives you much better tools for identifying and fixing bugs in your code. Okay, so for developers, this can be a huge time saver. It's like having a much clearer roadmap when something goes wrong.

[00:05:29] Speaker A: Yeah. Anybody who spent hours, you know, staring at a screen trying to figure out why something's not working, we'll really appreciate that.

[00:05:36] Speaker B: Yeah, it's a lifesaver. So then, of course, we come back to performance.

PHP8 is just plain faster. It's more efficient, which means faster loading times for your website visitors, which, as we said before, Google loves.

[00:05:48] Speaker A: Yeah, and happier users.

[00:05:49] Speaker B: And happier users. Everybody wins.

[00:05:51] Speaker A: So I think we've established a pretty strong case for upgrading.

[00:05:54] Speaker B: Definitely.

[00:05:54] Speaker A: Bill, let's flip the script.

[00:05:55] Speaker B: Okay.

[00:05:56] Speaker A: What happens if you don't? What are the risks of sticking with php 7.4?

[00:06:02] Speaker B: Well, the guide lays out some pretty serious potential downsides.

[00:06:05] Speaker A: Okay, so first and foremost, security vulnerabilities. You're not getting the security updates.

[00:06:10] Speaker B: Exactly. It's a ticking time bomb. Every day that you stay on an unsupported version, you're increasing your risk.

[00:06:16] Speaker A: Right.

[00:06:17] Speaker B: And then there's the broader issue of just lack of support in general. If something breaks, you're on your own.

[00:06:22] Speaker A: Right. You can't exactly call up PHP support and say, hey, I'm running this, you know, five year old version and it's broken.

[00:06:31] Speaker B: Yeah, good luck with that.

[00:06:32] Speaker A: You're not going to get very far.

[00:06:33] Speaker B: No, they're going to laugh at you.

[00:06:35] Speaker A: And then on top of that, you're missing out on those performance improvements we talked about.

[00:06:39] Speaker B: Right. Why settle for a slow website when you can have a fast one?

[00:06:42] Speaker A: Yeah, exactly.

[00:06:43] Speaker B: So then there's compatibility issues. As new features come out in php, they might not work with your old code.

[00:06:49] Speaker A: Right.

[00:06:50] Speaker B: So things could start breaking in unexpected ways.

[00:06:53] Speaker A: And you know, the guide also mentions that you're just missing out on innovation.

You're kind of stuck in the past while the rest of the web development world is moving forward.

[00:07:02] Speaker B: You're like that guy. Still Using Internet Explorer. Yeah, you don't want to be that guy.

[00:07:07] Speaker A: So the message is pretty clear. Staying on PHP 7.4 has some real risks.

[00:07:12] Speaker B: Definitely.

[00:07:13] Speaker A: Okay, so now that we've sort of scared everybody into thinking they need to.

[00:07:16] Speaker B: Upgrade, hopefully not too scared.

[00:07:17] Speaker A: Hopefully not too scared, but.

But motivated. Right. Let's talk about how you actually do it. The guide gives a general overview of the steps involved.

[00:07:27] Speaker B: It does, and it lays them out pretty logically.

So first things first, you gotta check compatibility before you do anything drastic.

[00:07:34] Speaker A: Okay.

[00:07:35] Speaker B: Use a tool like the PHP compatibility checker. This will tell you if your code is ready for PHP8.

[00:07:42] Speaker A: So it's kind of like a pre flight check. Make sure everything's good to go.

[00:07:45] Speaker B: Exactly. You don't want to just jump out of the plane without a parachute.

[00:07:48] Speaker A: Good analogy.

[00:07:48] Speaker B: So next up, and this is crucial, always, always back up your data before making any major changes.

[00:07:54] Speaker A: Always.

[00:07:55] Speaker B: This is your safety net. If anything goes wrong, you can always revert back to the old version.

[00:07:58] Speaker A: Yeah, better to have it and not need it than need it and not have it.

[00:08:02] Speaker B: Exactly. So once you're backed up, you can actually update PHP itself.

This usually involves upgrading the PHP installation on your server, often through your operating system's package manager.

[00:08:14] Speaker A: Okay, so that's more of a server level thing.

[00:08:16] Speaker B: Right. You're basically telling your server, hey, I want to use this new

version of PHP now.

[00:08:19] Speaker A: Okay, so what about the website files themselves?

[00:08:22] Speaker B: Ah, that's where UPD dependencies comes in. You got to make sure all the extra pieces of software that your website relies on, libraries, plugins, extensions, all that stuff are compatible with PHP8. So you might need to update some of those components.

[00:08:35] Speaker A: Right, because most websites aren't just the core PHP code, there's usually a bunch of other stuff going on.

[00:08:41] Speaker B: Yeah, that's a whole ecosystem.

[00:08:42] Speaker A: Okay, so once you've updated PHP and all its dependencies, what's next?

[00:08:46] Speaker B: Testing and debugging. This is where you really put your website through its paces in the PHP environment. Make sure everything works as expected.

[00:08:54] Speaker A: Right, and this is where that improved error reporting in PHP8 comes in handy.

[00:08:58] Speaker B: Exactly. It's like having a really good debugger looking over your shoulder, pointing out any potential problems. So then once you've confirmed that everything is working smoothly in your testing environment, you can finally deploy the updated code to your live production environment.

[00:09:12] Speaker A: Okay, so that's the big picture view. Yeah, but the guide also has some more granular, like, firsthand insights into how to make this process as smooth as possible.

[00:09:22] Speaker B: Yeah, they get into the Nitty gritty, which I appreciate.

[00:09:25] Speaker A: So what are some of the key practical tips they share?

[00:09:27] Speaker B: Okay, well, they suggest starting by creating a complete backup of your existing project with the database and everything running on the older PHP version.

[00:09:36] Speaker A: Right.

[00:09:37] Speaker B: Store that somewhere safe, separately.

[00:09:39] Speaker A: Yeah.

[00:09:40] Speaker B: That's your baseline, your point of comparison. If anything goes wrong.

[00:09:43] Speaker A: Yeah. You can always go back and see how it used to work.

[00:09:45] Speaker B: Exactly. So then they say, clone your project, set up a separate database for it.

[00:09:49] Speaker A: Okay.

[00:09:50] Speaker B: And configure this new environment to use the new php.

[00:09:54] Speaker A: So you're basically creating a safe sandbox to play around in.

[00:09:57] Speaker B: Exactly. You don't want to mess with your live site directly until you're sure everything is working. So once you have your sandbox set up, make sure to install the same PHP extensions that you were using in the old environment.

That helps ensure that all the core functionality is still there.

[00:10:11] Speaker A: Right.

[00:10:11] Speaker B: And then this is crucial. Enable PHP error reporting.

You want PHP to be super chatty about any issues it encounters.

[00:10:20] Speaker A: Let me know everything that's going on. Don't hide anything.

[00:10:22] Speaker B: No surprises.

[00:10:23] Speaker A: Okay.

[00:10:24] Speaker B: And then once you've got error reporting enabled, you're going to start seeing those warnings, those fatal errors, those syntax errors.

[00:10:32] Speaker A: Right.

[00:10:32] Speaker B: And this is where the real work begins. You got to fix all those issues.

[00:10:35] Speaker A: Yeah.

[00:10:36] Speaker B: Make sure your code is playing nice with PHP8.

[00:10:38] Speaker A: Right.

[00:10:39] Speaker B: So the guide also emphasizes the importance of checking all your third party libraries. Make sure they're compatible with PHP8, upgrade them if necessary, and if there are any that are no longer supported or compatible, and who you might need to find replacements.

[00:10:53] Speaker A: Okay.

And then don't forget about Cron jobs, those little automated scripts that run in the background. You gotta make sure they're compatible with PHP8 as well. Right. Because they're often forgotten about until they break.

[00:11:03] Speaker B: Exactly.

[00:11:03] Speaker A: Yeah.

[00:11:04] Speaker B: So, and finally, of course, testing, testing, testing. Test everything.

Make sure it all works perfectly in the PHP8 environment, compare it to your original setup, make sure nothing is broken, and ideally, that things are even better.

[00:11:18] Speaker A: It sounds like a pretty thorough and systematic approach.

[00:11:21] Speaker B: It is. It's not something to be taken lightly.

[00:11:24] Speaker A: Okay, so we've talked about the benefits, the risks, the process, but the guide also has a little bit of a warning about doing this yourself.

[00:11:31] Speaker B: Yeah, they're not trying to scare you off, but they do want you to be aware of the potential pitfalls.

[00:11:37] Speaker A: So what are some of the things that can go wrong if you try to do this yourself without enough experience?

[00:11:42] Speaker B: Okay, well, plugins are a big one. Plugins that worked perfectly fine on PHP7 might suddenly break on PHP8.

[00:11:48] Speaker A: Right.

[00:11:48] Speaker B: They might not be compatible anymore.

[00:11:50] Speaker A: Oh, that would be a huge headache.

[00:11:51] Speaker B: Yeah. And even worse, your whole website could go offline.

[00:11:54] Speaker A: Oh, no.

[00:11:54] Speaker B: Which means you're losing traffic, you're losing potential revenue and your users are getting a bad experience.

[00:12:00] Speaker A: Right.

[00:12:00] Speaker B: And then there's the risk of actually making your performance worse if you don't do the upgrade properly.

[00:12:05] Speaker A: Which would kind of defeat the whole purpose.

[00:12:07] Speaker B: Yeah, exactly.

[00:12:09] Speaker A: So the guide suggests that in a lot of cases it's actually safer and more cost effective to hire a professional.

[00:12:15] Speaker B: Yeah. If you're not comfortable doing this yourself, it's definitely worth considering getting help from someone who does this kind of thing for a living.

[00:12:22] Speaker A: Right. Because even though you might save some money up front, if you end up breaking something and having to call in a professional to fix it, it could end up costing you way more in the long run.

[00:12:32] Speaker B: Yeah. You don't want to go down that road.

[00:12:34] Speaker A: So I think that's a really good point to keep in mind. Definitely. So to kind of wrap things up, you know, we've gone through a lot of information here about upgrading from PHP 7.4 to PHP 8. We have, and it's clearly a big step. It is, but it's one that comes with a lot of

benefits.

[00:12:51] Speaker B: Absolutely.

[00:12:52] Speaker A: You know, long term support, enhanced security, better error handling, and of course the performance improvements. But it's important to remember that sticking with PHP 7.4 is not without its risks.

[00:13:04] Speaker B: Right.

[00:13:04] Speaker A: You know, you're leaving yourself open to security vulnerabilities, you're missing out on important updates and you're, you're basically falling behind.

[00:13:13] Speaker B: You're stuck in the slow lane.

[00:13:14] Speaker A: Yeah. And the, the guy that we've been talking about really lays out a nice step by step process.

[00:13:20] Speaker B: Oh, it's a great resource for making.

[00:13:22] Speaker A: This upgrade as smooth as possible.

[00:13:23] Speaker B: It is.

[00:13:24] Speaker A: But as we've also discussed, you know, doing this yourself if you don't have the expertise can lead to some, some serious problems.

[00:13:31] Speaker B: Definitely.

[00:13:32] Speaker A: So, you know, think carefully about whether you're comfortable doing this yourself or whether it's worth getting some professional help.

[00:13:38] Speaker B: It's a decision that could save you a lot of headaches down the road.

[00:13:40] Speaker A: Absolutely.

So, you know, for you listening in, what do you think is the most appealing aspect of PHP8 for your own websites?

[00:13:49] Speaker B: Yeah. What are you most excited about?

[00:13:51] Speaker A: Is it the security? Is it the performance? Is it the new features? I mean, there's a lot to unpack here.

[00:13:56] Speaker B: There is.

[00:13:56] Speaker A: So take some time, dig into those resources, think about your own needs and make the best decision for your web presence.

[00:14:03] Speaker B: And stay tuned for more deep dives.

[00:14:06] Speaker A: We'll be back with more fascinating stuff soon.

[00:14:08] Speaker B: Absolutely.

[00:14:08] Speaker A: Until then, happy coding.

[00:14:09] Speaker B: Happy coding.