

Thanks for taking a moment to listen to the product from read or listen.com, a subset of developers hangout where we take a moment to read tech related articles so you can listen to them while you're working, driving or exercising.

All right, enjoy.

Okay, so prompt engineering.

Honestly, when I first heard that term, I was like, is that like when people started calling garbage collectors memory management technicians?

I see what you mean.

Like, come on, just another buzzword.

Right.

But then I got into this material about using it with PHP.

And wow, this is not just hype.

No, not at all.

It's really fascinating stuff, actually, once you start digging in.

Like the document we're looking at for this deep dive, it was a real eye opener for me.

It's like we as PHP developers suddenly have this whole new language for talking to AI.

Yeah, that's a good way to put it.

It's really about understanding the nuances of how these LLMs interpret our instructions.

And it can do some seriously cool stuff.

It can.

And the best part is it doesn't have to be this super complicated thing either.

Like that example they give in the document format title, marketing assistant prompt.

So straightforward.

Right.

But from that, the system can figure out what to do.

That just blows me away.

It really highlights the power of LLMs to kind of like infer actions, even from just a little bit of information.

And speaking of structure, this whole system prompt thing really grabbed my attention.

Oh, yeah, definitely.

It's like these behind the scenes.

It's like, oh, my God, this is a really cool thing.

And then you have to think about how you're going to do it.

And then you have to think about how you're going to do it.

And then you have to think about how you're going to do it.

And then you have to think about how you're going to do it.

And then you have to think about how you're going to do it.

And then you have to think about how you're going to do it.

And then you have to think about how you're going to do it.

And then you have to think about how you're going to do it.

And then you have to think about how you're going to do it.

And then you have to think about how you're going to do it.

And then you have to think about how you're going to do it.

And then you have to think about how you're going to do it.

And then you have to think about how you're going to do it.

So as PHP developers, I mean, is this something we need to be thinking about, these differences?

It's definitely worth considering, kind of like when you're choosing a PHP framework, right?

Oh, yeah.

Each one comes with its own quirks, its own pros and cons.

So we've chosen our model.

Now we need to know how to talk to it, which is where I guess first shot and multi-shot prompting comes in.

Exactly.

And let's be honest, that first shot prompting, it's a bit of a gamble sometimes, like trying to win the lottery with a single ticket.

Yeah.

Been there, done that.

No lottery wins for me.

Right.

Multi-shot prompting, though that's where things get interesting.

Because we're giving the AI examples this time.

You got it.

You're showing it what you want, not just hoping it gets it right on the first try.

Like that cool example they use in the document with the SQL queries.

Exactly.

Instead of just asking what tasks are due this week, you'd give it a few different ways to structure that query.

And that's how we train the AI on the fly.

That just seems incredibly powerful for someone like me.

It is.

You're basically giving it a crash course in understanding your intent.

This is all making a lot more sense now.

It's like we're giving our AI a toolbox or something.

With tools to do even more complex things.

Right.

Think about things like querying databases or even making API calls.

Now you're talking my language.

That's what I'm talking about.

As a PHP developer.

I mean, that's got to get your wheels turning.

That's right.

Think about it like this.

You could write a prompt that actually triggers a database query, gets you the exact data you need, and then uses that to generate a whole report.

All right there in my PHP app.

Exactly.

No more manually writing out those crazy SQL queries.

Imagine what that would do for building those data-driven apps that we're always working on.

Yeah.

Talk about a game changer.

And this document even talks about tool chaining, which, I mean, it just sounds insanely powerful, doesn't it?

Tool chaining.

So we can like combine these tools to do even more.

Exactly.

Imagine this.

You've got one tool that can summarize these super long documents.

And then you've got another tool that's great at translating language.

Okay.

So you could say something like, "Hey, AI, summarize this French document, but give it to me in English."

All without having to do any manual translation.

That's wild.

So it's like we're putting together these teams of AI agents, each with their own thing that they're good at.

And then they just go to town on these tasks right there in our apps.

That's a great way to think about it.

And they give an example of this using that multi-shot prompting that we were talking about earlier.

Not with the SQL query for tasks, right?

That's one.

So imagine you take that, but you go one step further.

We're going to turn that over to you.

Okay.

And then we're going to turn that whole thing into a tool, a task creation tool.

Okay.

So instead of just asking the AI to find tasks, we can actually tell it to go ahead and create new tasks in the database.

We're giving it the power to take action.

Wow.

So we've gone from just retrieving information to actually making things happen in our apps based on these prompts.

You got it.

And you know what?

There's a whole deep dive later on that's entirely about tools.

But I think for now, let's shift gears a little bit and talk about something else.

Okay.

So let's talk about placeholders.

Placeholders.

Is that what I think it is?

It is.

Like those Mad Libs we used to do.

Remember those?

Oh, yeah, for sure.

Fill in the blanks.

That was fun.

Exactly.

It's like bringing that same kind of fun to our prompts.

Instead of having these static instructions every time, we can use placeholders, things like next week or now, to inject some real-time information.

Oh, okay.

So our prompts become dynamic.

They can change based on what's going on.

Exactly.

And think about how useful that would be for PHP applications.

Yeah.

Especially when you're working with stuff like dates and times.

Right, right.

Or if you wanted to include some user-specific data.

They actually say in the document that this could completely revolutionize user interfaces, make them so much more intuitive.

Oh.

I like where this is going.

Think about it.

Instead of having your users navigate through these complicated forms, they could just say, "Hey, show me all the orders I placed this month."

Okay.

Yeah.

That would be amazing.

No more trying to figure out what all those fields mean or ... Yeah. ... they're hunting for the right option in a drop-down menu.

Just tell the application what you want in plain English.

That's the dream.

Oh.

And this is all possible with these placeholders.

Exactly.

And to make things even easier, they even give you some PHP code, a class called `templatizer.php`.

Did you say PHP code?

Tell me more.

I knew you'd like that.

This class is actually pretty handy.

It handles all that replacement stuff for you.

So it takes those placeholders and swaps them out with the real values when the prompt runs.

Exactly.

So if you had a placeholder like `username`, this `templatizer` thing would just replace that with the name of whoever's logged in at that moment.

That's cool.

So it's like this templating engine, but specifically for AI prompts.

I like it.

It's a small detail, but it makes a big difference.

Being able to have that dynamic data in your prompts opens up all sorts of possibilities.

For sure.

We've covered a ton of ground here.

System prompts, all these different prompting techniques, these tools, and now these placeholders.

There's a lot more to this whole process.

It's a whole prompt engineering thing than I realized.

It's a whole new world.

And like they keep saying in the document, it's really about treating it as this iterative process.

So you keep experimenting and refining your prompts until you get them just right.

Exactly.

And speaking of which, that real world example, they walk you through the one about generating tags for a document that was super helpful, wasn't it?

Yeah, it really brought all this together.

They showed how to structure a good prompt, how important clean instructions are, you know, all of that.

I loved the part where they actually asked the LM itself for suggestions.

I know, right?

That was great.

How can I make this prompt better?

That's so important.

We don't always have to have all the answers ourselves.

Sometimes the AI can steer us in the right direction.

It's like we're working together almost.

The AI becomes this collaborator.

Exactly.

Which leads me to a question for you as the PHP expert here.

How do you think all of this ability to talk to databases like we've been seeing with those multi-shot prompting examples?

How's that going to change the way we design interfaces for our PHP projects?

It's pretty wild when you actually stop and think about it.

We're talking about moving away from all those dropdowns and radio buttons and just letting users talk to their apps naturally.

It's huge.

Right.

Like imagine a customer support system where the user can just type in their question, any question, and the AI not only understands what they're asking, but can actually pull up the information they need from the database to give them a real answer.

No more trying to figure out which keywords are the right ones.

No more trying to figure out which keywords are the right ones.

Exactly.

Think about how much that would improve the user experience.

And for us, Ph.D. developers, we're always building those kinds of interactive applications.

This could be revolutionary.

Yeah, absolutely.

And it's not just about user interfaces either, is it?

This whole natural language thing could completely change how we approach everything from reporting to data analysis.

Absolutely.

Instead of having to write these complicated database queries every time you need a specific report, you could just say, "Hey, you know what?"

I'm going to do this.

I'm going to do this.

I'm going to do this."

"Hey, you know what?

I'm going to do this.

I'm going to do this."

"Hey, you know what?

I'm going to do this."

It's exciting and a little bit intimidating at the same time, right?

Makes you wonder what the future holds for PHP development.

Definitely.

But for now, I think the best thing we can do is just dive in and start experimenting.

Totally agree.

Who knows what amazing things we'll be able to build when we learn how to speak the language of AI.

Exactly.

So to all you PHP devs out there, get out there, build some cool stuff, and most importantly, keep those prompts sharp.

And we'll catch you in the next deep dive. dive.