





---

Category:Guitars Category:Kontakt (audio editor) users I've been using WordPress for three years and in that time I have been through a good number of plugins and themes. This makes my experience with plugins a bit different from most people's, though. I find myself writing a lot about plugin development. If you haven't yet, take a look at the plugin writing tutorial and the development tutorial for even more information. If you're writing a plugin, you'll probably be asked how do you know if your plugin should use the core hooks, or if it's better to use a 3rd party framework. If you've read these tutorials, you should be ready for this tutorial which will introduce you to hooks. Why Use Hooks? I'll just get this out of the way: hooks are powerful. They let you do all sorts of wonderful things, but they also allow you to do some very weird and bad things. It seems to be a common misconception that plugins are just for a lot of extra stuff. However, the core of a plugin is often surprisingly small. Most plugins just wrap around core functions to make them a bit more useful to other plugins or to add some extra capabilities. The hooks you create will probably be used in a similar way. And if you ever find yourself saying "this is silly, I could just do this in the core function", take a moment to think about it. You might just be writing a lot of boilerplate, and that means a lot of waste. Back to hooks... The big thing to remember is that hooks are simply variables to store data in. This means that you can't change data directly, just the reference to the data. You can use the Global Hooks API to set and get the data, but you can't manipulate the data directly. Hooks are meant to let you get/set data that is not directly accessible to your plugin. I could write a silly plugin that adds some data that is not included in any of the core functions. If I want to get/set that data, I'm going to have to go through the core hooks, or make my own. Calls to the Global Hooks API will give you a direct reference to the data, but you don't get to 2d92ce491b