PHP and Java have many syntactical differences. This is not surprising as Java is a compiled general-purpose programming language, and PHP is a server-side scripting language mainly used for web development.

With PHP, keywords, functions, and classes are not case-sensitive. You can have mixed-casing in PHP scripts and they will be interpreted and evaluate fine. In Java, case sensitivity matters because the compiler is case-sensitive. Creating variable names with different casing will cause Java to treat them as different variables.

In Java, strings are enclosed in double quotes. Strings in PHP can be expressed in four ways including single and double quotes (Ajtai, 2010). Single quotes in Java are only used when considering the literal char type.

PHP scripts are contained within <? php ?> tags, and PHP files often contain HTML. The PHP is executed on the server and the HTML is sent to the browser. In Java, all the code belongs to classes.

With weakly-typed PHP and the lack of case-sensitivity, I can see bugs and bad coding styles creeping into Java code if a programmer was not fluent and able to switch context easily among both languages. On the other hand, I can see PHP as offering more freedom when attempting to prototype a project quickly because of the lack of need to explicitly declare variable types and worry about case-sensitive issues.

Atjai, P. (2010, August 10th). *What is the difference between single-quoted and double-quoted strings in PHP?* Retrieved from <https://stackoverflow.com/questions/3446216/what-is-the-difference-between-single-quoted-and-double-quoted-strings-in-php>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

I'll echo what Ashley and Niki mentioned and add some thoughts of my own. The code is more than jarring, I think it is ugly. It does seem similar to HTML, but even HTML has better symmetry and flow on a given page from what I have seen so far.

Even as a less-than-favorite programming language, PHP powers many of the largest sites on the web (Atwood, 2008). Facebook, Wikipedia, Yahoo!, Tumblr, Flickr, and Wordpress were built with PHP. Jeff Atwood goes on to mention in his blog Coding Horror, "sufficiently talented coders can write great applications in terrible languages". The takeaway is that choice of language is secondary. PHP may be ugly, but it works well for the web. Do not get "married" to any one technology, and always choose the right tool for the job.

Atwood, J. (2008, May 20th). PHP Sucks, But It Doesn't Matter. Retrieved from <https://blog.codinghorror.com/php-sucks-but-it-doesnt-matter/>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Jeannie's focus on associative arrays in PHP is a nice surprise, I would have overlooked the convenience and freedom that associative arrays bring to programming in PHP. I welcome the benefits, but just as others have hinted… I think mixed arrays could create difficulties when debugging.

Another issue that has not been mentioned so far is that PHP applications may have a mix of code within. PHP, HTML, JavaScript, and SQL could all be wrapped into a web application. It may be challenging to trace execution while moving from one language to another when a bug appears.

Finally, there is an argument that PHP Web apps do not always have “a continuous linear flow of execution”, but rather a series of HTTP requests and responses with each transaction independent of the preceding and following actions (Grehan, 2010). I see all of these complexities “hiding” errors from us as we learn to program in PHP.

Grehan, R. (2010, February 3rd). Debugging PHP Web apps is hard to do. Retrieved from <https://www.infoworld.com/article/2628045/development-tools/debugging-php-web-apps-is-hard-to-do.html>