[Home](http://www.webreference.com/) / [Authoring Resources and Information](http://www.webreference.com/authoring/index.html) / [languages](http://www.webreference.com/authoring/languages/index.html) / [xml](http://www.webreference.com/authoring/languages/xml/index.html) / [XML\_and\_PHP\_Simplified](http://www.webreference.com/authoring/languages/xml/XML_and_PHP_Simplified/index.html) / XML and PHP Simplified

**XML and PHP Simplified**

By Leidago Noabeb

[inShare](javascript:void(0);)

#[>](http://www.webreference.com/authoring/languages/xml/XML_and_PHP_Simplified/2.html)  
[[next]](http://www.webreference.com/authoring/languages/xml/XML_and_PHP_Simplified/2.html)

Often the Document Object Model is thought of as an abstract concept that is very difficult for beginners to deal with. With this series of articles, I will try to simplify and demystify the use and application of XML and the DOM. First, we will look at what XML is and then move on to what functions are available for use to manipulate and use XML.

**What is XML?**

XML is an acronym for eXtensible Markup Language; it was designed to make it easy to represent data in a structured way, almost like a database. Though it is not quite a database, it does allow for formatting and storing structured data persistently. Just because the word language appears in XML does not actually mean that XML is a language, it is better to interpret it as a specification that enables you to create your own markup language(s). It is a subset of Standard Generalized Markup Language, which is the mother of all markup languages; incidentally, it is also the parent of the more popular Hyper Text Markup Language or HTML. XML, as already stated, makes it easy to exchange information between different applications. For example, a program written in PHP will be able to process information created and stored in a XML document by another programming language such as PERL or ColdFusion. For those of you who are familiar with HTML, using XML documents should not be a problem at all, because it is similar to how documents are formatted in HTML. The main difference between the two is that the HTML specification has a fixed set of elements and attributes so you have to learn to use those if you want to write any sensible HTML document. That in itself is an advantage because it makes it easy for developers across the world to read and write HTML documents. For example if you want to make text bold in HTML you use the <b></b> or <strong></strong> tags. Any developer that is familiar with HTML will know what they are. XML on the other hand does not have any fixed elements or attributes. It allows you to create your own elements and attributes, this of course gives you the power to define your own language or to use someone else's definition. The flexibility that XML offers is what makes it so powerful not to mention useful. This is also why programs created by different applications can read and write to a XML document.

**HTML**- You can create a complete website knowing just this, however it'll look like those 90s websites with blue links. It's there to create basic structure for a website and it's impossible (as far as I know) to make any website/webpage without using HTML.  
  
**CSS**- This is what makes your websites beautiful and modern looking. It's just for all the visual effects and with new version simple animation too.  
  
**JavaScript**- It's the real programming language which runs in your browser. It gives your webpage power. Using JavaScript you can put calculations, logical conditions and all the programming functionality. HTML alone can't do that.  
  
**PHP**- It's also a programming language, but it doesn't run on your browser. It runs on the web server. When you request a webpage by putting the url in your browser or opening it's link, the web server accepts the request, take the HTML page, run the associated PHP script in it, convert the result obtained from PHP to HTML and then return it to your browser. In the browser all CSS and JavaScript runs. PHP is generally used for getting and putting data to/from the database (mostly MySQL). It runs on Linux based servers.  
  
**XML**- It's not related to web pages at all. It's just a way of storing data wrapped in XML files. The text data wrapping is used so that we don't have to use databases. You wouldn't want to send a complete database table when you just need the current weather information, would you? It's not just used to send data, sometimes it's used to represent User-interfaces too (in Android and iOS, while Windows uses XAML for that). The pages in your Android app are generally coded in XML, which the Android system (technically Dalvik VM or nowadays ART) reads and convert it to proper Java. So it's just for storing textual info which gets parsed and the parsing/reading mechanism uses it for their own purpose. The syntax looks like HTML tags.  
To read more about XML try the following answer   
     [What are the difference between HTML and XML?](https://www.quora.com/What-are-the-difference-between-HTML-and-XML/answer/Abhishek-Jain-25?share=1&srid=XIgg)  
  
***Bonus:-***  
**ASP**- It's another server side language just like PHP, but it runs on Windows based Servers. Currently it's obsolete and ASP .net is more used these days.  
  
**HTML5**- It's the next iteration of classic HTML (HTML 4.1). It's a complete suite of technologies, not just HTML. It gave power to HTML to become more than a basic structure provider. With HTML5 you can put videos and audios in your page without using Flash or silverlight (We all hate flash in websites, don't we?). It has support for high end graphics library WebGL which can give you similar experience as PC gaming based on DirectX and OpenGL. In layman term WebGL means support for high-end 3D graphics.  
  
**JSON**- JSON is just like XML. It's used for storing and sending text data. Its structure is harder to read than XML (due to a lot of curly braces), but it's generally smaller in size. For example, to send XML you need to write tags, schema and other characters too, while for JSON colons (:) and curly braces are enough. It reduces the bandwidth required.

Node.js is a server-side platform built on Google Chrome's JavaScript Engine (V8 Engine). Node.js was developed by Ryan Dahl in 2009 and its latest version is v0.10.36. The definition of Node.js as supplied by its [official documentation](https://nodejs.org/) is as follows −

Node.js is a platform built on [Chrome's JavaScript runtime](https://code.google.com/p/v8/) for easily building fast and scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.

Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

I will try and provide a slightly different perspective on things; if you are learning to code for the purpose of making money, this is how I would look at it;

Invest your time & effort in the tech that you believe will give the best return/yield.

There are plenty of people on here that will argue the case for each of the tech, perhaps these are good parameters to consider;

* Functional programming is on the rise and will most likely continue to trend and most likely will overtake object oriented coding in the future.
* Javascript is the only language that can be used on server, web front end, and mobile.
* Have a look at what programming languages are in demand (hint. Javascript isn’t the most in demand but its not exactly dying out, more than what can be said for Php) [The 9 Most In-Demand Programming Languages of 2017 - Coding Dojo Blog](http://www.codingdojo.com/blog/9-most-in-demand-programming-languages-of-2017/)

Now lets consider the money side of things; for this I use contracting rates here in the UK, PHP has huge numbers of dev’s out there, its the most popular language for web development for sure, but unfortunately due to its popularity and the sheer number of people that can do it, php dev’s get paid roughly half the amount of node.js (at least here in the uk). Have a look at job postings and compare for yourself.

Consider also the trends in software development; backend systems are becoming micro services, this is a huge strong point for node.js, web front end libs/frameworks are all done in Javascript, and mobile is either native or Javascript.

Php became popular at a time when building web applications required a lot of complexity and was an easy alternative to Java/.Net for most small projects (i remember a lot of people hated OO coding) but if I had to bet on the future I would put my money on Javascript/Node.js.

A lot of people will tell you Facebook was built on Php blah blah blah, then theres wordress blah blah blah, but always remember there is a very important “was” in there. I think if Facebook was built today it would most likely be on Node.js. This is not to say that Php will go away, no way! Php will still be around in the future, and so will Java, & Python etc… But Node.js is on the rise and will most likely continue to gain in popularity if trends/indicators are anything to go by.