**ASP.NET vs PHP – complete comparisons**

**What is PHP?**

PHP was originally created by Rasmus Lerdorf in 1995. PHP is used to be a short for Personal Home Page but now it’s known as Hypertext Preprocessor. PHP is a server-side scripting language for creating dynamic Web pages. PHP code is executed on server side and the result is send to your internet browser. PHP code can be hosted on every OS/Web server; it can be Windows server with IIS or Linux with Apache. I would like to point out that there are PHP frameworks, for example: Zend, CodeIgniter, CakePHP etc. Definitely, you wonder what frameworks are, and which are the benefits of using them.

Simple demo of hello world in PHP

|  |
| --- |
| **<?php** Echo "Hello, World!"; **?>** |

When you request this page, server process it and return to you as HTML.

|  |
| --- |
| Hello, World! |

**What is ASP.NET?**

As I wrote for PHP above, for ASP.NET is similarly. ASP.NET is web application framework which is made by Microsoft to allow programmers to build dynamic Web sites or web services. It was first released in January 2002 with version 1.0 of the .NET Framework, and the latest updated version 4.5. ASP.NET support more programming languages like C#, VB.NET, JScripts etc. ASP.NET frameworks support two models of developing, Web-Forms and MVC model.

Simple demo of ASP.NET hello world (WebForms model)

|  |
| --- |
| Page**:** **<**asp**:**Label ID**=**"Label1" runat**=**"server"**></**asp**:**Label**>**  Code behind**:** Label1.Text **=** "Hello World"**;** |

**What is framework?**

I put this question because ASP.NET is Web-framework which supports many languages. It means that if we compare ASP.NET and PHP, it would be like this:

* ASP.NET framework with C# language vs. PHP language with Zend framework.
* or C# vs. PHP

“Framework” owns finished classes or classes which are already wrote and you could use them immediately. For example, how would you do certain staffs with framework or without framework. If we want to place validators on certain form, where the name and the last name would be obliged:

* Without framework, we have to write code for client and server side, which will verify that validation, by ourselves. (1 hour)
* With framework, we could make the same, by using the finished validators , or just setting them. (5 min).

**Example for basic caching in PHP, PHP with Zend framework and ASP.NET**

* In PHP, you have to create a class by yourself, which will have to generate your page as HTML, from time to time.
* PHP with Zend framework: You’ll call the cache class, and easily, in few lines of code, you could create a basic output cache. Example:

|  |
| --- |
| **<?php** *// include the class require\_once('Zend/Cache.php'); // front end options $frontendOptions = array( 'lifetime' =--> 60*  );    *// backend options*  $backendOptions = array(  'cache\_dir' =&gt; 'cache' *// Directory where to put the cache files*  );    *// make object*  $cache = Zend\_Cache::factory('Output',  'File',  $frontendOptions,  $backendOptions);    *// make an id*  $cacheID='user1';    *// everything before this is not cached*  if (!($cache-&gt;start($cacheID))) {  *// begin cache*  *// end cache*  $cache-&gt;end();  }  *// everything after this is not cached*  **?>** |

* For ASP.NET, just add OUTPUT cache, or it would be like this:

|  |
| --- |
| OutputCache Duration**=**"100" VaryByParam**=**"none" |

**Creating a Grid which reads database information.**

* PHP without framework: To explain this, I would need few pages. It will be necessary to make a class for communication with a base, a class for confrontation and screening the information from the base, javascript for paging and sorting. (2-3h)
* PHP with framework: still too much coding…
* ASP.NET Web forms: I won’t speak about this, the video will show you.
* ASP.NET MVC: Less coding than PHP or PHP with framework but much more time vs. ASP.NET Web Forms.

Keep reading this text, because if you stop now, you’ll think that ASP.NET Web forms is absolute winner. No, there are too many faults related to it.  
Frameworks help the developers to write code in lesser time than usual. There is always the option of writing your own framework for your application. However, when you have something like Cake PHP doing a lot of nifty things for you or a Zend that provides great functionality, you can select one depending on your requirements.

**Future-proof**

None of both platforms will disappear soon. Those are the platforms leaders for creating web sites. In this moment, Microsoft forces the MVC model.

**Basic pros and cons**

|  | **ASP.NET** | **PHP** | **Comments** |
| --- | --- | --- | --- |
| Latest Stable Version | 4.5 | 5.3.8 | November 2011 |
| License | MS EULA | PHP License v3.01 |  |
| Price | ASP.NET – free  OS – Windows – not free  Webserver- IIS – included in OS | PHP – free  OS- Linux free  Webserver- Apache free | PHP can work on any webserver and any OS. ASP.NET only works on Windows/IIS. There is Apache module to be able to make work ASP.NET on Linux :) . |
| Hosting Price | Shared hosting – same  VPS/Dedicated – little expensive | Shared hosting – same  VPS/Dedicated – little cheaper | Check “Cost” title for more info, why i says that shared hosting is same price. |
| Promoted by | Microsoft | PHP Community |  |
| Support by owner | Yes | No |  |
| Security Fixes | Auto Update – part of OS | Need to update separately when available |  |
| Languages | C#  VB.NET  JScripts  a lot more | PHP |  |
| Database | any | any | Prefered database for ASP.NET is MS-SQL, for PHP is MySQL |
| Development IDE | Visual Studio  MonoDevelop | any ide | You can write php code even in Notepad, however prefered ide is ZendStudio |
| Easy to learn | yes | yes | From my expirience, ASP.NET Webforms with visual studio is very easy to learn. |
| Content Management Systems | yes | yes | PHP is absolute winner here. |
| Backward Compatibility | Old code will work on new framework versions without change | Old code might need tweaks to make it work on latest versions | There can be some problems with converting ASP.NET versions, however those fixes are few and quick compared to PHP. |
| Security | depend of architecture | depend of architecture | There is no “AUTO” button to make your website secured. Its all about developer expirience.. |
| Performance | Faster | Slower | Check “Performance and Speed” title |

**Basic SEO overview**

When we talk about SEO optimization, my first opinion is that PHP is better. My experience is that HTML output, in the Web forms model for creating web sites, has few lines of code more, which is not necessary for the look of the site. First, I would point out those HIDDEN values such as:

* \_\_VIEWSTATE
* \_\_EVENTARGUMENT
* \_\_EVENTVALIDATION
* \_\_EVENTTARGET
* ToolkitScriptManager1\_HiddenField

You must wonder how those HIDDEN values affect on the SEO optimization. I think that Google is extremely smart, so when Google crawl your page, he don’t take account of that kind of values, for example certain text which is about your post. But, on the other hand, you have excess text, and if you consider that one type is one byte, than, according to the size of those HIDDEN values, you may have an expendable text of few KB. Also, you should know that all search engines like Google, when crawl your page, they limit the size of the HTML code, and I think it is 100KB.  
For example:

* ASP.NET Webforms model – 100KB text, 5KB hidden values, when google crawl your page, he will index 95KB of your text. 1.1 sec download time
* PHP – 100KB text – when Google index your page, he will crawl 100% of your text. (same page, 1 sec download time)

As you saw, two same pages, made in PHP and ASP.NET Web forms, have different size; accordingly, the PHP page would be faster, when your Internet browser download. But, those are small differences and nobody could note them. Here, we can also point out that one of the measures of Google, for better SEO, is the speed of your web site.

Ok, there isn’t a big difference?  
Well, it’s not exactly like that, till now I spoke about the excess of code in the hidden fields, and now I’ll explain you all about the excess code of ID’s on HTML tags. When you create a master page in ASP.NET, and you create a Default page which inherits by the master, than, if you put ASP.NET link control with ID=”myID”, this link would be rendered, for example:

* id=” ctl00\_myID” , and those are the additional 6 bytes of ID.

Also, SEO URL in Web forms, by default, look like this:

* www.mysite.com/Default.aspx?id=2?catid=32

It’s much better for you to have SEO URL’s, for example:

* www.mysite.com/new/article/My-Article-Title

Video of ASP.NET GridView HTML Source

If you made this with PHP, the html source should look similar like on this image.  
[](http://perfectwebtutorials.com/wp-content/uploads/2011/11/phpgrid.jpg)

I would like to say that the version ASP.NET 3.5 sp1 has the option, ClientIDMode=static, on which you can control the rendering of the ID. Also, you can move all the hidden values to the footer, or, to keep the viewstate on a server side. Also, you can change all the links as you want. But, I wouldn’t like to go into details, because this kind of optimization sometimes could be problematic. If those things are important to you, then I recommend you ASP.NET MVC model.  
I’ll not speak about Web Forms vs. MVC, but the biggest benefit of using the MVC model is that the look of the HTML is under your total control. Here, we don’t have ID fields or Id extensions or excess code. So, if ASP.NET Web Forms is worst than PHP for SEO optimization, then ASP.NET MVC model for SEO optimization is absolutely same with PHP.

**Scalability and Ease of Maintenance**

Scalability and ease of maintenance have nothing to do with whether you select PHP or ASP.net platform. Web Application scalability and ease of maintenance primarily depend on:

* Programmers’ experience
* Using the best programming practices
* Using a solid programming framework
* Following programming guidelines and standards

**Compiled vs interpreted languages**

This is also an interesting debate, but, I couldn’t draw a precise conclusion about when we use “compiled code” or which code is faster. But, since Facebook has changed the PHP code in C++, they made “compiled” and, also, spoke out that now they are accelerated, or, they use less CPU resources surely, I can say that the “compiled” is faster than the “interpreted” code.

But, on the other hand, if you want to change something in the “complied code”, he has to be compiled again, from the beginning; while in the “interpreted” you can change your code very easily.   
First of all, at the very base of the argument it has to be realized that the two languages are very different. Asp.Net is an optimized and compiled language, meaning code you enter is reduced to a set of machine-specific instructions before being saved as an executable file. Even if you do not explicitly compile your code before you deploy it, it compiles the first time it is run, then after that it runs as compiled code.

Php on the other hand is an interpreted language, meaning it is saved as the code you write and run directly from that code. It is widely accepted and proved many times over that Compiled programs generally run faster than interpreted ones because interpreted programs must be reduced to machine instructions at runtime. Here is a quote from Wikipedia that shows just how much faster they can be:

“A program translated by a compiler tends to be much faster than an interpreter executing the same program: even a 10:1 ratio is not uncommon. The mixed solution’s efficiency is typically somewhere in between.”   
Conclusion: If you have a web site and you change it all the time, or you don’t have too many visitors executing some function, then, the interpreted code is better. On the other hand, if you have too many visitors executing too many functions, and by the way you don’t make often changes to your code, then, the complied code is faster. Check my title about PHP vs ASP.NET performance!

**Cost**

When you read the title, you must imagine that ASP.Net is more expensive than PHP. Let us find out whether it is correct. I’ll pick out three examples:

* Creating a small-medium website, and hosting in certain firm. MOST POPULAR
* Creating a small-medium website, and hosting on an own server.
* Creating a big project, and hosting on an own server.
* I wont compare this two languages for enterprise solutions for several reasons..

Needful programs & tools for creating a dynamic website:  
1. IDE  
2. Database  
3. OS  
4. Web server  
5. Platform

| **small/shared hosting** | **IDE** | **OS/Web server** | **Database** |
| --- | --- | --- | --- |
| Free PHP | Free, NetBeans | Same price | Same price |
| Free ASP.NET | Free, VS Express | Same price | Same price |

I must insinuate that in the past, it was more expensive to host on ASP.NET web site, shared hosting, but nowadays, that make no difference. I’ll take one of the biggest hosting companies for shared hosting, GoDaddy, as an example, where the Linux hosting and the Windows hosting cost equal.

| **small/dedicate server** | **IDE** | **OS/Web server** | **Database** |
| --- | --- | --- | --- |
| Free PHP | Free, NetBeans | Free Linux/Apache | Free MySQL |
| Free ASP.NET | Free VS Express | Paid Windows server ~400$ | Free SQL Express |

| **small/dedicate server** | **IDE** | **OS/Web server** | **Database** |
| --- | --- | --- | --- |
| Free PHP | Paid Zend Studio ~300$ | Zend Server ~1700$ | Free MySQL |
| Free ASP.NET | Paid VS Pro ~ 700$ | Paid Windows server ~500$ | Paid SQL server ~2000$ |

**Support and Resources**

I must say that ASP.NET is promoted by Microsoft, while PHP is promoted by PHP Community. I must mention that, also Microsoft provides you ASP.NET support, BUT that doesn’t mean that ASP.NET has more support(tutorials, how to make ,plugins, etc). I talk about this by my own experience, because I have worked with both, PHP and ASP.NET, and on the other hand the numbers show the same. Just enter on Google:

PHP: 25,270,000,000 results  
ASP.NET: 221,000,000 results  
PHP Tutorial: 223,000,000 results  
ASP.NET Tutorial: 29,800,000 results  
PHP Gallery: 124,000,000 results  
ASP.NET gallery: 25,700,000 results  
PHP cache: 385,000,000 results  
ASP.NET cache: 20,300,000 results  
PHP how to: 1,410,000,000 results  
ASP.NET how to: 200,000,000 results

**Time to learn**

Many people say that PHP is far easier to learn then ASP.NET, but, I quarrel with this fact. I have 3 years work experience with PHP, and 6 years with ASP.NET; so, I can say that in this moment, I’m able to make a better site in ASP.Net then in PHP. I only had a problem at the beginning, till I figure out how ASP.NET works, because it is much more different than PHP. But, on the other hand, ASP.NET MVC is very similar to PHP, indeed most of PHP frameworks support the MVC model.

MVC is short of  
Model – database classes  
Controller – manipulating with business logical and database classes before send information to VIEW.  
View – html output structure

Basic example”:  
Model – I have a base with products which have a name and a price.  
Controller – I take all the products from a base which price is higher than 100 dollars.  
View – It performs the selected products.

**Time to develop and CMS**

This is a relative question, but I’ll try to answer it. First, I would like to point out that it’s quite easier to find an open source PHP CMS than ASP.NET open source CMS, and, on the other hand, I think that it’s not possible to find such powerful CMS for ASP.NET as that which is for PHP. For example: Word press, which is PHP CMS for creating blogs vs BlogEngine.NET CMS which is for ASP.NET.

So, if we want to make a blog in which we would have categories, posts, gallery, registration with user privileges (to post comments), creating custom pages, then, the absolute winner is PHP CMS. Why?

Compared to Word Press, BlogEngine.NET is relatively immature and has a much smaller community. That’s not to say Blog Engine isn’t good on its own merits, but it’s definitely more niche where Word Press is mainstream. That’s not to say only PHP with CMS for creating blogs is better than ASP.NET for creating blogs, but, it means that creating CMS with PHP, regardless of the fact if it’s intended for blogs, portals or shopping cart, is faster and have a bigger support, tutorials or “how to” posts.

Google search results for “blogengine.net tutorial” = about 839,000 results

WordPress tutorial: About 89,100,000 results

On the other hand, as for example we need to create a Custom web site which couldn’t be created with CMS. Then, this type of web site could be created much faster, if we use ASP.NET.

Note: When I heard about the Microsoft course for the MVC model for web site creation, by using ASP.NET MVC and ADO.NET, I realized that we could make a custom web site in few days. Starting with database scheme, then, business logical and html structure.  
It could be faster if we use ASP.NET Web Forms, because it has “Fastest development time (RAD)”. For complex charts/reports/grids, you’d have to roll your own with PHP or ASP.NET MVC (or use a helpers which someone else has created). But with Web Forms you have a wealth of server controls (like Telerik).

ASP.NET MVC, model doesn’t have ASP.NET controls, but there are some control templates such as Telerik MVC controls, which can accelerate your time for creating some web application. Check http://demos.telerik.com/aspnet-mvc/grid

**Editors and Tools**

In this case the absolute winner is PHP. I seriously think that I haven’t heard about an IDE for ASP.NET except Visual Studio and Mono, I can say that is easier to find ONLINE ASP.NET IDE than desktop (coderun.com) :) . And, if other IDEs exist, I bet that the number isn’t higher than 5. Also, it’s important to mention that those editors work only on Windows, while for PHP, we have a big range of editors: commercial, free and cross –platform.

I’ll list some PHP editors:  
Cross-platform free editors : Eclipse, jEdit, Netbeans IDE  
Cross-platform commercial editors : UltraEdit , Zend Studio ,TextMate, phpDesigner

The adventige of Visual Studio is that they have tons of professional add-ons, both commercial and free.  
Add-on gallery can be found on Nuget website. http://nuget.org/

**Platform Independent**

I’ll not explain too much about this topic, because PHP is absolute winner. PHP can work on Windows IIS and on Linux Apache, while ASP.NET can work only on Windows IIS server. But if you plan to buy shared hosting from a famous company, as I said, there is no difference in the price.

Note: The price for VPS or Dedicate is different, but, for a shared hosting is same.

Note: ASP.NET can work on Linux Apache server, but what about MS-SQL? :)

**Which popular sites run on which platforms?**

The following table lists the top popular sites, the platform and languages they use.

Note: Please don’t confuse C with C# (pronounced C Sharp) — they are completely different programming languages. ASP.net is mostly programmed in C# (C Sharp) or Visual Basic and not C.

Site Up Since Server Platform Programming  
Language

* **Google.com** – Linux C, Java, C++, PHP & MySQL, Python for search
* **Facebook.com** – Linux, PHP, MySQL and C++
* **YouTube.com** – Linux, C, Java and MySQL
* **Yahoo.com** – Linux, C++, C, Java, PHP & MySQL
* **MSN.com**– Windows, ASP.net
* **Live.com** Windows, ASP.net
* **stackoverflow.com** Windows, ASP.net
* **MySpace.com** Windows, ASP.net (one of few giants who converted their core technology, from ColdFusion to ASP.NET. I dont remember where was that post, but i remember that MySpace gain about 50% of server resources reduction with ASP.NET)
* **Wikipedia**Linux, PHP & MySQL
* **Amazon.com** – Linux, Solaris, C++, Java, J2EE
* **WordPress.com** – Linux, PHP & MySQL

**StackOver Flow software**

* Windows Server 2008 R2 x64: Operating System
* SQL Server 2008 R2 running Microsoft Windows Server 2008 Enterprise Edition x64: Database
* Ubuntu Server
* CentOS
* IIS 7.0: Web Server
* HAProxy: for load balancing
* Redis: used as the distributed caching layer.
* CruiseControl.NET: for builds and automated deployment
* Lucene.NET: for search
* Bacula: for backups
* Nagios: (with n2rrd and drraw plugins) for monitoring
* Splunk: for logs
* SQL Monitor: from Red Gate – for SQL Server monitoring
* Bind: for DNS
* Pingdom: an external monitor and alert service.

**StackOver Flow hardware**

* 10 windows server 2008 r2 (3 za stackoverflow)
* 2 sql server 2008 r2

Note: The Stackoverflow group has around 20 servers, 12 of them use Microsoft licenses. I must mention that there are 4 big web sites which host on those 20 servers: http://stackexchange.com/, http://stackoverflow.com/ , http://superuser.com/ , http://serverfault.com/, that’s why I mentioned above that 3 from 10 windows servers are intended for stack3 overflow, because they use 2sql server for all the pages. I’ll take one in, order to calculate them.

So, if you pick up licenses for hosting a site like StackOverFlow, you’ll need around:

3x Win Standard 600$ + 1x SQL standard 2000$ = 3800$.  
I’ll add all the other licenses for software which use them.  
Visual Studio 2010 Team Suite ~3000$  
14 total developers, or lets say 4 for stack over flow.  
4 x VSP ~600$ = 1800$

Total = ~8600$  
I think that this is not a big amount if you have a site like Stack over flow. Stack over flow has around 16 000000 unique users per month. So, you can see whether those 8300 dollars are too much, if they place Google ad sense. :) )))))))

NOTE: This calculation can be 50,000 $ if you calculate SQL Enterprise Edition (~20000 $) (if they use Cluster SQL, because they have two SQL servers)…..

**Flickr**

* PHP
* MySQL
* Shards
* Memcached for a caching layer.
* Squid in reverse-proxy for html and images.
* Linux (RedHat)
* Smarty for templating
* Perl
* PEAR for XML and Email parsing
* ImageMagick, for image processing
* Java, for the node service
* Apache
* SystemImager for deployment
* Ganglia for distributed system monitoring
* Subcon stores essential system configuration files in a subversion repository for easy deployment to machines in a cluster.
* Cvsup for distributing and updating collections of files across a network.

**Performance and Speed**

Maybe, this is the most interesting comparison between PHP and ASP.NET. Generally, you can find on Internet that PHP is faster than ASP.NET or the opposite. I’ll test both, and I’ll give to you the source code of both scripts if you want to test them by yourself.

I made two functions: IfElse and Loops.  
  
[Download source test code.](http://perfectwebtutorials.com/wp-content/uploads/bench.rar)

Configuration:  
TEST1:  
OS: Windows 7 Ultimate  
CPU: Intel P7570  
RAM: 3GB  
ASP.NET C# WebForms 4.0  
IIS 7.5 – default settings  
Apache 2.2.21 – default settings  
PHP 5.3.8 – default settings

TEST2 (just for php)  
VPS 1 core Xeon E5520  
2GB ram  
Centos  
Apache 2.2.16  
Php : 5.2.14  
  
Results:  
  
TEST1  
PHP IfElse: 6.438997 secounds  
ASP.NET ifelse: 0.1020059 secounds  
PHP loops: 5 secounds and 213732 milliseconds  
ASP.NET loops: 0.0680039

TEST2(even with xeon)  
PHP IfElse: 2.367657  
PHP loops: 2.717072

ASP.NET is much more faster then PHP, numbers proof that!

If you think about Facebook, please understand that:  
1. It uses a re-written PHP and  
2. Transformer+compiler called HipHop which transforms PHP code to highly optimized C++ code and then compiles with gcc. We are talking here about the PHP we get for application developers from [php.net](http://php.net/).