

# "PHP"

---

Wez Furlong,  
Director of Engineering,  
Message Systems, Inc.

- \* I'm a "Core Developer" of PHP
- \* Mostly known for the Streams and Database layers, also implemented COM and Objective-C bridges
- \* I design and oversee development of very high performance email server software

# Why am I here?

- \* PHP is widely used as a web server glue language
- \* Wouldn't it be great if the web devs could transfer their language knowledge to the client side--Silverlight?

So, PHP?

# So, PHP?

- \* I have to confess to having no practical experience with implementing PHP on the CLR or DLR



# So, PHP?

- \* I have to confess to having no practical experience with implementing PHP on the CLR or DLR
- \* Although I did look at implementing on Parrot a few years ago

# So, PHP?

- \* I have to confess to having no practical experience with implementing PHP on the CLR or DLR
- \* Although I did look at implementing on Parrot a few years ago
- \* I can certainly set up a couple of things for Tomas (he has demos!)

# Ugly Duckling

- \* PHP was designed to solve the specific problem of making it easy for Rasmus to make his home page
- \* Not a good example of neat language design



```
<html><head><title>hello</title></head>  
<body>
```

```
<?php  
if (isset($_GET['name'])) {  
    echo "Hello $name";  
} else {  
?>
```

Please enter your name:

```
<form><input type=text name=name>  
<input type=submit></form>  
<?php } ?>
```

```
</body></html>
```

# Types

- \* Dynamically typed
- \* null, string, integer, double, array, object, resource, boolean
- \* classes, interfaces, functions are not first class

# Scoping

- \* PHP has two scopes: global and local
- \* Need to explicitly reference the global scope if you're inside a function
- \* Can compile locals directly to backend variables unless...
- \* local scope is referenced indirectly... `eval()` is one example

# namespacing: TBD

- \* No shipped version of PHP has namespaces
- \* Still debating packaging features, such as import and symbol aliasing
- \* But at least we've settled on :: as the namespace separator (!)

# OO Model

- \* PHP 4: Confirmed as a drunken hack
- \* PHP 5: Richer features
- \* OO Model is single inheritance with interfaces
- \* Classes are immutable



# Dynamic OO

- \* In addition to things like PPP:
- \* Magical methods can intercept attempts to read, write or call undefined properties or methods
- \* Writes to undefined properties define those properties on an instance
- \* Classes are immutable: cannot add methods at runtime

# Interfaces

- \* Ugly duckling version: essentially a subset of classes
- \* PHP validates implementation requirements when the interface is fully bound

# Conditional Declarations

- \* There are no compile time conditionals
- \* Functions and types may not be fully bound until the script is running
- \* Functions and types may not be re-defined once defined (runtime error)

```
if ($_GET['moon'] == 'full') {  
    class A {}  
    function notalwayshere() {}  
} else {  
    class B {}  
}  
if (some_condition()) {  
    class D extends A {}  
} else {  
    class D extends B {}  
}
```

# Challenges

- \* Dynamic language features get in the way of “good” optimizations (bah!)
- \* Large number of C based extensions that link to external libraries--unsafe



# Solved already?

- \* Phalanger is an implementation of PHP targeting the CLR
- \* Developed independently from the PHP community
- \* We're finally co-operating and looking at taking advantage of the DLR