

Delivering Dynamic Web Content with Apache, MySQL, PHP, and Perl

Paul Jones
Developer Services
Novell, Inc.
pjones@novell.com

Cory Aitchison
Developer Services
Novell, Inc.
caitchison@novell.com



Novell.

N

Who's in the audience

PHP?

Perl?

MySQL?

Apache?





Agenda

Setup

Discuss PHP

Write PHP

Discuss Perl

Write Perl

Resources



N

Setup - MySQL

Change root's password

```
mysqladmin -u root password secret
```

Configure phpMyAdmin

```
/srv/www/htdocs/phpMyAdmin/config.inc.php
```





Setup - phpMyAdmin

Create a database

Create a table

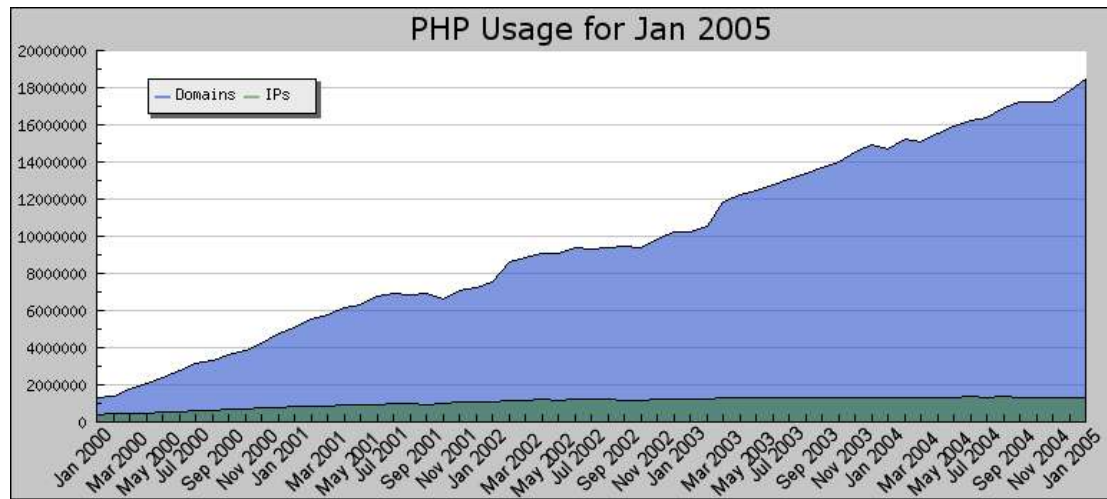
Create a user



Discuss PHP

What is PHP

PHP is a widely-used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML.



Source: Netcraft

N

PHP is easy to learn

All variables have a preceding \$

Weakly typed language

- `$var = 1;`
- `$var = "hello";`
- `$var = array(3, 4, 5);`
- `$var = array("name" => "value", 2 => "two");`
- `$var = new myClass;`
- `$var = null;`

N

Funny Variables

Variable Variables

- `$name = "Paul";`
- `$var = "name";`
- `echo $$var;`

References

- `$name = "Paul";`
- `$ref = &$name;`
- `$ref = "Cory";`
- `echo $name;`

N

Control Structures

if, else, elseif

while, do-while

for, foreach

switch

break, continue

N

Functions

Variable number of parameters

Parameters may have a default value

Any type of variable may be returned

Functions may be saved as a variable

```
function myfunction($arg1, $arg2 = 'Paul')
{
    echo "$arg1 $arg2\n";
}

$func = 'myfunction';
$func('Hello');
```

N

Two Scopes

PHP has only two scopes.

- Inside a function
- Outside a function (global)

Keyword `global` will give access to global variables

Example:

```
$outside = "hello";  
function inside() {  
    $inside = "paul";  
    echo $outside." ".$inside."\n";  
    global $outside;  
    echo $outside." ".$inside."\n";  
}
```

N

Classes, Objects, Exceptions

PHP5 has a new object model

PHP4 had a quasi object model

All general OO functionality is available in PHP5

- `$object->method();`
- `$object->property;`

N

Form Processing

`$_GET, $_POST, $_REQUEST`

```
<FORM method="POST">
  <input type="text" name="username">
  <select multiple name="values[]">
    <option value="1">One</option>
    <option value="2">Two</option>
    <option value="3">Three</option>
  </select>
</FORM>
```

```
$username = $_POST['username'];
$values = $_POST['values'];
```



PHP & MySQL

`mysql_connect`

`mysql_select_db`

`mysql_query`

`mysql_fetch_array`

`mysql_fetch_row`

`mysql_free_result`

`mysql_close`

N

Example

```
$link = mysql_connect('host', 'user', 'password');
mysql_select_db('database');

$query = 'SELECT * FROM my_table';
$result = mysql_query($query);

while ($row = mysql_fetch_array($result, MYSQL_ASSOC)) {
    foreach ($row as $name => $value) {
        echo "$name = $value\n";
    }
}

mysql_free_result($result);
mysql_close($link);
```

N

Write PHP



N

Objective

Write a PHP page that will take the name and score post elements and insert them into your database.





Hints

Connect to MySQL

Select your database

```
INSERT INTO table (col1, col2) VALUES (val1, val2)
```

Close your connection

Display something to the user stating it was done



N

Possible Solution

```
if(!isset($_POST['submit'])) {
    header("Location: moles.html");
}
$link = mysql_connect('localhost', 'mole', 'novell');
mysql_select_db('moles');

$query = "INSERT into highscores (name, score) VALUES
('".$_POST['name']."', '".$_POST['score']."')";
$result = mysql_query($query);

mysql_close($link);

header("Location: index.php");
```

N

Discuss Perl

Perl has only three built in data types: scalar, array, hash

scalars hold a single value

```
$var = "value";
```

arrays are ordered lists of scalars

```
@var = ("val1", 2, "val3");
```

hashes unordered set of key value pairs

```
%var = ("key1" => "val1", "key2" => 2);
```

Beware of your context

N

Control Structures

if, elsif, else

unless

while, until, do-while, do-until

for, foreach

switch

last, next, redo

N

Subroutines

Pass as many variables as you want

Pass by value

They are available in @_

You may return any type

```
sub mysub
{
    my $param1 = $_[0];
    return $param1+1;
}

print mysub(1);
```



Scope

my

variables are local to the current function

variables are local to the current file

local

variables are local to the current function or any called subroutine

N

Classes, Objects, Exceptions

They all sort of exist in perl5

They will be more robust in perl6 (parrot)

```
$object->method();
```

```
$object->property;
```

N

Perl & MySQL

```
use Mysql
```

```
Mysql->connect
```

```
Mysql->query
```

```
Mysql->fetchrow
```

```
Mysql->fetchhash
```

```
Mysql->fetchcol
```

N

Example

```
$db=Mysql->connect("host", "database", "user", "password");

$query = 'SELECT * FROM my_table';
$result = $db->query($query);

while (%row = $result->fetchhash) {
    my @headers = keys %row;
    foreach $header (@headers){
        print $header . " = " . $row[$header] . "\n";
    }
}

undef $result;
undef $db;
```

N

Write Perl



N

Objective

Write a Perl page that will retrieve a list of names and scores from the database and display them in a nice fashion.



N

Hints

Connect to MySQL

```
SELECT col1, col2 FROM table ORDER BY col1  
DESC LIMIT 10
```

Display the results

undef the connection and result

N

Possible Solution

```
use Mysql;
print "Content-type: text/html\n\n";
$db=Mysql->connect("localhost","moles","mole","novell");
$query = 'SELECT name,score FROM highscores ORDER BY score
DESC LIMIT 20';
$result = $db->query($query);

print "<table style='border: dashed 1px blue;'>";
while (($name, $score) = $result->fetchrow) {
    print "<tr><td>$name</td><td> $score</td></tr>";
}
print "</table>";

undef $result;
undef $db;
```

N

Developer Resources

Software, Sample Code, and Documentation

<http://developer.novell.com/>

<http://forge.novell.com/>

<http://www.php.net/>

<http://www.cpan.org/>

<http://www.mysql.com/>

<http://www.apache.com/>



Novell®

Unpublished Work of Novell, Inc. All Rights Reserved.

This work is an unpublished work and contains confidential, proprietary, and trade secret information of Novell, Inc. Access to this work is restricted to Novell employees who have a need to know to perform tasks within the scope of their assignments. No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of Novell, Inc. Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

General Disclaimer

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. Novell, Inc., makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc., reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.



Novell.