

# PHP and MySQL Tutorial

Michael Coblenz

1

Topeka, KS - High school sophomore Brett Tyson was suspended today after teachers learned he may be using PHP.

"A teacher overheard him say that he was using PHP, and as part of our Zero-Tolerance policy against drug use, he was immediately suspended. No questions asked," said Principal Clyde Thurlow. "We're not quite sure what PHP is, but we suspect it may be a derivative of PCP, or maybe a new designer drug like GHB."

Parents are frightened by the discovery of this new menace in their children's school, and are demanding the school do something. "We heard that he found out about PHP at school on the internet. There may even be a PHP web ring operating on school grounds," said irate parent Carol Blessing. "School is supposed to be teaching our kids how to read and write. Not about dangerous drugs like PHP."

In response to parental demands the school has reconfigured its internet WatchDog software to block access to all internet sites mentioning PHP. Officials say this should prevent any other students from falling prey like Brett Tyson did. They have also stepped up locker searches and brought in drug sniffing dogs.

Interviews with students suggested that PHP use is wide spread around the school, but is particularly concentrated in the geeky nerd population. When contacted by BBspot.com, Brett Tyson said, "I don't know what the hell is going on dude, but this suspension gives me more time for fraggin'. Yee haw!"

PHP is a hypertext preprocessor, which sounds very dangerous. It is believed that many users started by using Perl and moved on to the more powerful PHP. For more information on how to recognize if your child may be using PHP please visit <http://www.php.net>.

2

## Introductions

- [ Me
- [ You

3

## The Metadata

- [ 2 parts, each 3 hours
- [ PHP first, MySQL second
- [ End at lunch today
- [ Continue tomorrow

4

## The Rules

- [ Think before you code!
  - Don't just copy code from here
- [ No wimps!
  - This class is hardcore
- [ Don't get lost!
  - Questions are required, not optional

5

## Why are we here?

- [ Want sites that change - *dynamic* sites.
- [ Need more power!
- [ Powerful sites need powerful tools
- [ The tool makes the developer



6

# Our Example

- [ An online store!

# Static vs. Dynamic Sites

- [ Static: stays the same
  - Might be edited periodically
- [ Dynamic: changes on demand
  - search, change preferences, interact, post, edit

# What is PHP?

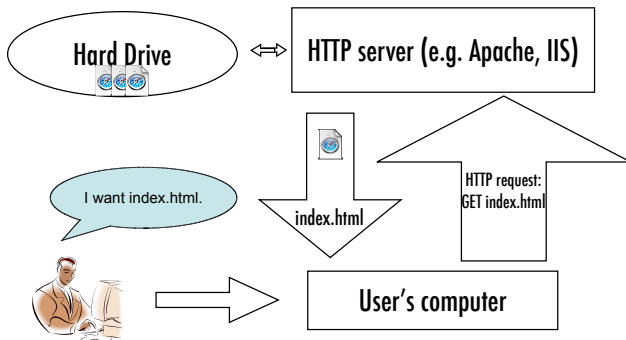
- [ Software package
- [ [www.php.net](http://www.php.net)
- [ Works with web server software
- [ Similar to: ASP, ColdFusion, Java Server Pages (JSP)



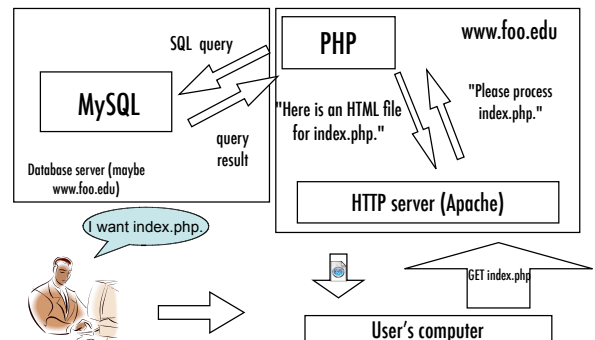
# Why a database?

- [ Need to store lots of data!
- [ Fast access to data
- [ Data must persist
- [ Data is structured

# How Static Pages Work



# How Dynamic Pages Work



# The Plan

PHP	today
MySQL	today/tomorrow
Integration	tomorrow
Project: online store!	tomorrow

13

# Down to business!

— [ Let's learn PHP!

14

# Really: What is PHP?

- [ PHP: Hypertext Preprocessor
- [ Program to help server
  1. Run script (a program!)
    - Generate HTML
  2. Send HTML to HTTP server
  3. HTTP server sends HTML to user

15

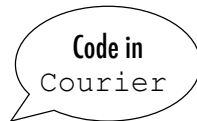
# PHP Language Intro

- [ Think: "I'm generating an HTML file."
- [ Two categories:
  - PHP code
  - HTML
- [ HTML gets sent to user directly
- [ PHP code executed, output sent to user

16

# Which witch is which?

- [ Delimiters:
  - `<?php`
  - `?>`
- [ Between delimiters: **MUST** be PHP code
- [ Not between delimiters: **MUST** be HTML



17

# Statements

- [ PHP code consists of statements
  - [ Semicolon-terminated
    - Exception: last statement
- ```
<?php
echo ("Hello, world!");
doStuff ();
echo ("Goodbye, world!");
?>
```

18

## Delimiter Examples

| Input                                      | Output                   |
|--------------------------------------------|--------------------------|
| <code>&lt;?php printPi (); ?&gt;</code>    | 3.14159265359...         |
| <code>printPi ();</code>                   | <code>printPi ();</code> |
| <code>&lt;?php echo (40 + 2); ?&gt;</code> | 42                       |
| <code>40 + 2</code>                        | <code>40 + 2</code>      |

19

## Hello, world!

— [ A working example

— [ Put this in `hello.php` in Sites folder

```
<HTML><HEAD>
<TITLE>Hello!</TITLE></HEAD>
<BODY>
<?php echo ("Hello, world!");?>
</BODY></HTML>
```

— [ Look at `<server>/~josti#/hello.php`

20

## Spot the syntax errors

```
< ?php echo ("hi") ?>
```

```
<?php echo (hi) ?>
```

```
<?php echo ("hi")
```

```
    echo ("hi") ?>
```

```
<?php echo ("hello!"); ?>
```

use instead:

```
<?php echo ("hello!"); ?>
```

21

## Variables

— [ Name preceded by `$`

— [ No declarations, very weak typing

— [ Must begin with letter or underscore

— [ Names contain only letters, numbers, underscores

— [ `$name = "George";`

— [ **WARNING:** undefined variables have NULL value; can use without error!

22

## Comments

— [ Comment your code!

— [ Code is for people, not machines.

PHP does this; `//` ignores this

`/*` PHP ignores

all of this `*/`

23

## If statements

```
if ($grade > 64) {
    echo ("You passed!");
}
else {
    echo ("What a loser!");
}
```

Curly braces only required for more than one statement

24

## More tests

```
if ($grade >= 64 && !($grade > 75))
    echo "lucky punk!";

if (calculateMeaningOfLife() == 42)
    echo "Phew!";
```

25

## Handy Boolean operators

Not	!	Greater than	>
And	&&	Less than	<
Or		Greater than or equal to	>=
		Less than or equal to	<=
		Not equal	<>

26

## Spot the errors

Not syntax errors...

or "What does this code do?"

```
— [ if (name == "Dave") echo "D";
— [ if ($name = "Dave") echo "D";
— [ if ($foo < 6); echo "less!";
— [ if (1 < 4)
    if (5 > 42) echo "one";
    else echo "two";
```

27

## Loops

```
for ($i = 0; $i < $max; $i++)
{
    echo ("iteration " . $i);
}

Notice the ". " - it's the string concatenation operator.
while (isMoreData()) {
    printData();
}
```

28

## Exercise

- [ Create a web page that displays your name 500 times.
  - use a for loop.
- [ After that works, modify it to also print a line number on each line.

29

## Functions to Know & Tell

- [ die("You lose! No data for you!");
- [ mail(\$toAddress, \$subject, \$message, \$headers);
- [ isset(\$variable);
- [ But how do I read the documentation?
  - We'll see later.

30

## Arrays

- [ Actually ordered maps
  - keys (strings or ints) to values (stuff in array)
- [ `$arr = array (5 => 1);`
- [ `echo $arr[5]; // 1`
- [ `$arr["x"] = 42; remember this for MySQL`
- [ `echo $arr["x"]; // 42`

31

## Tables refresher (HTML)

```
<TABLE BORDER="1">
<TR>
  <TD>100</TD>
  <TD>200</TD>
  <TD>300</TD>
</TR>
<TR>
  <TD>400</TD>
  <TD>500</TD>
  <TD>600</TD>
</TR>
</TABLE>
```

100	200	300
400	500	600

32

## Generating tables

```
<TABLE>
<?php
$rows = 10; $cols = 3;
for ($i = 0; $i < $rows; $i++) {
  echo "<TR>";
  for ($j=0; $j < $cols; $j++) {
    echo "<TD>row $i,col $j</TD>";
  }
  echo "</TR>";
}
?>
</TABLE>
```

33

## Questions?

- [ Next up: Form processing

34

## Forms

- [ Two parts
  - form (HTML)
  - form processor (PHP)

35

## The Form

```
<FORM action="processor.php"
method="post">
<INPUT type="text" name="username"
size="20"/>
<INPUT type="password"
name="password" size="20"/>
<INPUT type="submit" value="click
here!"/>
</FORM>
```

36

```
<FORM action="processor.php" method="post">
<INPUT type="text" name="username" size="20"/>
<INPUT type="password" name="password" size="20"/>
<INPUT type="submit" value="click here!"/>
</FORM>
```

- [ Method: how browser sends data
  - GET
  - POST
- [ Action: where browser sends data
- [ INPUT: a field for data entry

37

## The form processor

- [ Input fields are variables (PHP magic!)
  - use `$_POST["variable"]` if sent with *post*.
  - `$_GET["variable"]` for *get*.
  - Changed in PHP 4.2 for security.
- [ Do whatever is appropriate with them.
  - Query a database, perhaps?
- [ Can be the same file as the form

38

## A simple login system

```
<?php
if ($_POST["username"] == "teacher" &&
    $_POST["password"] == "goldfish") {
    echo ("Here are the grades.");
}
else {
    echo ("You lose!");
}
?>
```

39

## Exercise

- [ Make a form for users to enter their name and favorite color.
- [ If their favorite color is the same as yours, print something special.

40

## One page forms

```
<?php
if (isset($_POST["username"])):
    echo ("Hello, $_POST[username]");
else:
?>
<FORM ...>....</FORM>
<?php endif; ?>
```

Notice the colons!  
Avoids ugly echos.

41

## PHP include/require

- [ Avoid code duplication
- [ `include ("header.php");`
- [ `require ("login.php");`
- [ Same, but `require` fails if its arguments doesn't exist.

42

## Handling login

- [ Problem: each page wants \$username, \$password.
- [ Solutions?
  - Yes, you.

43

## Cookies

Or: angering ignorant users in one easy step!

- [ Save a little data on user's machine
- [ Like, say, the username, once they authenticated.
- [ `setcookie ("username", $username);`
- [ Be more careful if you care about security.

44

## How can login work?

- [ `require ("login.php");`
- [ in `login.php`: if username not set, ask and die.
- [ That's it!
- [ Not very secure. There are better ways.

45

## On to MySQL!

46

## What is SQL?

- [ Structured Query Language
  - Not English, Spanish, French,...
- [ Query: request
  - "Please add Joe to the user list."
  - "Who posted today?"

47

## What is MySQL?

- [ Particular implementation of SQL
- [ Others:
  - SQLite (included in PHP 5)
  - PostgreSQL
  - MS SQL Server

48

## MySQL includes...

- [ mysqld
  - Daemon - runs all the time
  - Server process
  - Manages files that contain actual database
- [ mysql
  - Client - runs when you need it
  - Talks to mysqld

49

## Relational Databases

- [ Not your mother's databases!
- [ Database == spreadsheet?

50

## A good database?

Mug	Color	Price
Apple	White	\$10
Microsoft	Black	\$2
Sun	Green	\$5
DEC	green	\$100

51

## A Better Database

ID	Color
0	White
1	Black
2	Green

**Colors**

ID	Mug	Price	Color
0	Apple	10	0
1	Microsoft	2	1
2	Sun	5	2
3	DEC	100	2

**Mugs**

52

## Relational Databases

- [ Avoid Duplication
- [ Assign a UNIQUE ID to each row
- [ Refer to data in other tables by ID
- [ Unambiguous!

53

## Buzzwords

- [ Table: Spreadsheet
- [ Database: Collection of tables
- [ Row or record: Entry in table
- [ Column: Field for entering data
- [ Schema: A particular database setup

54

## Types

- Think "kinds"
- Types of tools: hammer, screwdriver
- bang (hammer)? OK.
- turn (screwdriver)? OK.
- turn (hammer)? No!

55

## Some data types

- INT: -2147483648 to 2147483647
- FLOAT, DOUBLE
- DATE, TIMESTAMP
- VARCHAR(20): up to 20 characters
- TEXT: up to 65,535 characters
- BLOB: like TEXT, but sorting is case insensitive

56

## Your turn!

- Design a database schema for a store for your mug collection.
- Hint:
  - mugs table
  - users table (better record who bought what!)
- What data do you need?
- Groups of three

57

## Using MySQL

- Use mysql client to talk to server
- ssh (with PuTTY). See blackboard.
- Login: josti#. Password: international
- `mysql -u josti# -p`
- Now: only type MySQL commands.

58

## MySQL commands

- `use josti#;`
- End commands with semicolons
- Case InSenSiTiVe
- `show tables;`

59

- ```
CREATE TABLE mugs (id INT NOT NULL AUTO_INCREMENT, PRIMARY KEY (ID), description VARCHAR(40), price float);
```
- Pattern: create table <table\_name> (column definition, column definition, ...)
- Column definition: <column\_name> <type>

60

## Changing table schemata

- [ `ALTER TABLE mugs ADD quantity int`

61

## Indices and Keys

- [ **Index: pre-sort by this, so searching is MUCH faster**
- [ **Primary key: unique identifier**
- [ **Constraints are your friend!**
- [ **NOT NULL, DEFAULT**

62

## Adding Data

- [ `INSERT INTO mugs (description, price) values ('DEC', 100);`
- [ **or: INSERT INTO mugs SET description='DEC', price=100;**
- [ **Add a few records of your own!**

63

## Getting data

- [ **You SELECT the data you want FROM the table it's in.**
- [ `SELECT description, price FROM mugs`
- [ `SELECT description FROM mugs ORDER BY description ASCENDING`
- [ `SELECT description FROM mugs WHERE price < 10`

64

## A brief syntax interlude

- [ "In the beginning there was the word. But by the time the second was added to it, there was trouble. For with it came syntax..." —John Simon
- [ "Colorless green ideas sleep furiously." - Noam Chomsky
- [ Let's look at the manual.

65

- [ `INSERT [LOW_PRIORITY | DELAYED] [IGNORE] [INTO] tbl_name [(col_name, ...)] VALUES ({expr | DEFAULT}, ...), (...), ... [ ON DUPLICATE KEY UPDATE col_name=expr, ... ]`
- [ [stuff]: **stuff is optional**
- [ {a | b}: **pick a or b, not both. Must have one.**

66

```
ALTER [IGNORE] TABLE tbl_name
    alter_specification [, alter_specification] ...
```

```
alter_specification:
    ADD [COLUMN] column_definition [FIRST | AFTER col_name ]
    | ADD [COLUMN] (column_definition,...)
    | ADD INDEX [index_name] [index_type] (index_col_name,...)
    | ADD [CONSTRAINT [symbol]] PRIMARY KEY [index_type]
      (index_col_name,...)
    | ADD [CONSTRAINT [symbol]] UNIQUE [index_name] [index_type]
      (index_col_name,...)
    | ADD [FULLTEXT|SPATIAL] [index_name] (index_col_name,...)
    | ADD [CONSTRAINT [symbol]] FOREIGN KEY [index_name]
      (index_col_name,...) [reference_definition]
    | ALTER [COLUMN] col_name {SET DEFAULT literal | DROP DEFAULT}
    | CHANGE [COLUMN] old_col_name column_definition [FIRST|AFTER
      col_name]
    | MODIFY [COLUMN] column_definition [FIRST | AFTER col_name]
    | DROP [COLUMN] col_name
    | DROP PRIMARY KEY
    | DROP INDEX index_name
    | DROP FOREIGN KEY fk_symbol
    | DISABLE KEYS
    | ENABLE KEYS ... (it keeps going!)
```

67

## More SQL

- [ UPDATE mugs SET price=15  
WHERE id = 1
- [ UPDATE mugs SET  
description = 'hello, world'  
*Oops! Don't do this!*
- [ DELETE FROM mugs WHERE id  
= 42
- [ Try it yourself

68

## Multiple tables

- [ And you thought it was complicated enough!
- [ mugs table: description, id, color  
— but color is an int
- [ colors table: name, id
- [ Want list of mugs and their colors

69

```
mugs: description, id, color
colors: name, id
```

- [ Problem:
  - Combine data from two tables
- [ Attempted Solution:
  - SELECT mugs.description AS name, colors.name AS color  
FROM mugs, colors
- [ Does this work?

70

## Wrong answer!

```
+-----+-----+
| desc | color |
+-----+-----+
DEC	Blue
Sun	Blue
DEC	White
Sun	White
+-----+-----+
```

A subset is  
shown here  
for  
simplicity....

- [ What happened?

71

## Analysis

```
+-----+-----+
| desc | color |
+-----+-----+
DEC	Blue
Sun	Blue
DEC	White
Sun	White
+-----+-----+
```

- [ MySQL followed instructions.
- [ SELECT mugs.description  
AS name, colors.name AS  
color FROM mugs, colors
- [ Got all combinations of mugs and colors!

72

## The Fix

```
previously: SELECT
mugs.description AS name,
colors.name AS color FROM
mugs, colors
```

- [ What didn't we specify?
- [ SELECT mugs.description AS name,  
colors.name AS color  
FROM mugs, colors  
WHERE mugs.color = colors.id

73

## SQL functions

- [ SELECT 1 + 2
- [ SELECT MAX (price) FROM  
mugs
  - MAX is an "aggregate" function
  - also COUNT
- [ SELECT name FROM users  
WHERE age < 18 OR age > 65
- [ SELECT name FROM users  
WHERE name LIKE 'Mi%e%'

74

## User accounts

- [ Have a users table.
- [ Store passwords.
  - No! Use sha1!

75

## MySQL interface

- [ New! Need PHP 5, MySQL 4.1
- [ Object-oriented
- [ The plan:
  1. Close anything you open
  2. Open connection
  3. Do query
  4. Use results

76

## Using results

```
<TABLE>
<?php
$mysqli = new mysqli ("localhost", "josti#",
"inter", "josti#");
if (mysqli_connect_errno())
    echo "connect error!";
$result = $mysqli->query("SELECT * FROM mugs");
while ($row = $result->fetch_assoc()) {
    echo "<TR><TD>$row[description]</TD> <TD>
    $row[price]</TD></TR>";
}
?>
</TABLE>
```

77

## Secret Forms

- [ Scenario: How do you buy a mug?
- [ Want "buy" link on entry...
  - but link, not form
- [ Need to hide data in link
- [ GET to the rescue!
  - (never seen GET in a cape and tights?)

78

## Secret forms: the trick!

- [ **Need to encode post ID in link**
- [ `http://.../showMugs.php?mug=42`
- [ `...?mug=42&sortBy=date&order=0`
- [ **Use `urlencode($uglyStuff)` ;**
  - e.g. "Thing with spaces & stuff"
  - hence `...?foo%20%37bar%20+baz`

79

## The project: a store!

1. Display all mugs for sale.
2. Add a "buy" link next to each mug.
  1. When clicked: go to a "complete purchase" page
3. Add an "inventory" field to the mugs table so you can track how many are in stock.
4. It should subtract one from the inventory every time someone buys one.
5. Starter code is provided.

80