PHP and MySQL
Session 1: What is PHP?

Izzy Johnston

izzycjohnston@gmail.com

@izzy_johnston

Downloads

- Practice Files
  - http://.izzycjohnston.com/gdi/PHP_Practice_Files.zip
- PCs
  - http://www.wampserver.com/en/download.php
- Macs
  - http://www.mamp.info/en/downloads/index.htm

- Text Editors
  - PCs—Notepad++, jEdit, Aptana
  - Macs—Text Wrangler, jEdit, Aptana

Class Structure

- Session 1: What are PHP and MySQL?
- Session 2: Getting started with MySQL
- Session 3: Database Manipulation
- Session 4: Advanced PHP & PHP in the "real world"

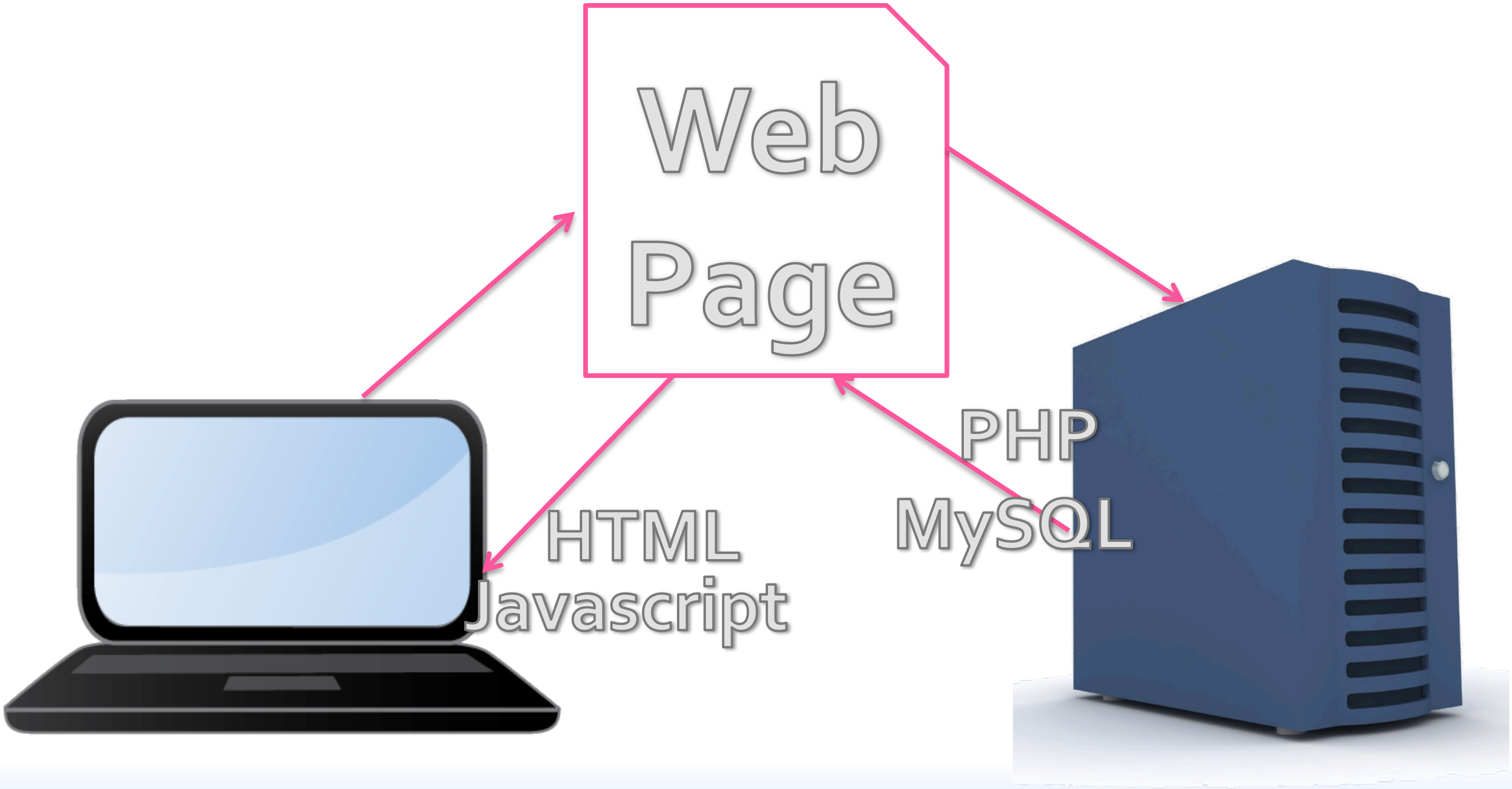
PHP—the nerdy definition

- **P**HP: **H**ypertext **P**rocessor
- Server-side language
- Supports database languages like MySQL

PHP—the English definition

- Dynamic content
- Database integration
- Do more with HTML
- Why code thousands of pages if you could make just one?

Client-side vs. Server-side Languages



Web Page

HTML
Javascript

PHP
MySQL

Client-side vs. Server-side languages, cont.

- Actions that a web browser can interpret
- Ability to see the source code
- Can see without being live on the web
- Code every page

- Actions to complicated for a web browser
- Source code translated into HTML or Javascript
- Need a web-wide or local server
- Code one page and change content dynamically

Variables

- Variable=Chunk of content
  - Numbers
  - Words
  - Characters
- Format
  - $name

Mathematic Operations

- Assign Value

```
$a=2;
$b=3;
```

- Basic Mathematic Equations

```
$c=$a+$b;
$d=$c-($a+$b);
```

- Reassign Value

```
$d=$c;
```

Concatenation—
the best word in the English Language

- Assign Value
  - $a="Hello";
  - $b="my name is";
- Concatenate
  - $c=$a . $b;
- Don't forget the spaces!
  - $a="Hello";
  - $a="Hello ";

If/Else Statements

```
if (some parameter){
    do this;
}
else {
  do that;
}
```

If/Else Statements, cont.

```
$a=2;
$b=3;
if ($a>$b){
    echo $a;
}
else {
  echo $b;
}
```

If/Else Statements, cont.

```
$a=5;
$b=3;
if ($a>$b){
    echo $a;
}
else {
  echo $b;
}
```

While Loops

```
$a=0;
while ($a<10){
   echo $a;
   $a=$a+1;
}
```

Mathematical Operators

Symbol	Meaning	Example
+	Addition	$x + 2 = 4
-	Subtraction	$x - 2= 0
*	Multiplication	$x * 2 = 4
/	Division	$x / 2 = 1
%	Modulus (Remainder)	5 % $x = 1
++	Increment	$x ++ = 3
--	Decrement	$x -- = 1

*For all examples, $x=2

Comparison Operators

Symbol	Meaning	Example
==	Equals?	$x == $y FALSE
!=	Does not equal?	$x != $y TRUE
>	Is greater than?	$x > $y FALSE
<	Is less than?	$x < $y TRUE
&&	AND	if ($x<5 && $y >5)
||	OR	if ($x<5 || $y >5)
!	NOT	if !($x==$y)

*For all examples, $x=2 and $y=3

Functions--look for the ()

```
function functionname (parameters) {
...code to be executed
}

function multiply ($a, $b){
    $c=$a*$b
    echo $c;
}
```

Functions, cont.

```
multiply ($a, $b){
   $c=$a*$b;
   echo $c;
}

multiply (3, 5);
multiply (4, 2);
```

Let's develop it!—HTML Forms

form.html

```
<html>
 <head>....</head>
 <body>
    <form action="result.php" method="get">
      <label>Enter Degrees in Fahrenheit</label>
      <input name="degrees" type="text" />
      <input  type="submit" value="Get Degrees in Celsius"/>

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

FTP or Local Server

- If you have MAMP up and running, place the two practice files in the htdocs directory
- If you have WAMP up and running, place the two practice files in the www directory
- If you have neither, upload the files to my server:
  - Server: izzycjohnston.com
  - Username: gdi_june
  - Password: php2011

Let's develop it—Calling from forms

result.php

```
<?php
 $f = $_GET['degrees'];
 $f = htmlspecialchars($f, ENT_QUOTES,
   'UTF-8');
?>
```

Let's develop it!—Print data

```
<?php
 $f = $_GET['degrees'];
 $f = htmlspecialchars($f, ENT_QUOTES,
   'UTF-8');
 echo "You entered " . $f . " degrees
   Fahrenheit.";
?>
```

Let's develop it!—Manipulate Data

```
<?php
 $f = $_GET['degrees'];
 $f = htmlspecialchars($f, ENT_QUOTES, 'UTF-8');
 echo "You entered " . $f . " degrees Fahrenheit.";

$c= ($f-32)*5/9;
echo "That is " . $c. " Celsius.";
?>
```

Let's develop it—Using a function

```
<?php
 $f = $_GET['degrees'];
 $f = htmlspecialchars($f, ENT_QUOTES, 'UTF-8');
 echo "You entered " . $f . " degrees Fahrenheit.";

$c= ($f-32)*5/9;
echo "That is " . $c. " Celsius.";

round( $c, 2);
?>
```

Forms and PHP.

- Use action="filename.php" to tell the form where to send info
- Use method="get" to show data in URL
- Name each part of the form that takes data
  - i.e. input type="text" name="degrees"
- Call the name by the method in the file from action, and assign it to a variable
  - i.e. $variable = $_GET['degrees'];

Homework—

- Using the form and results pages we created today as a starting point:
  - Create a form to do a conversion formula (Either inches to cm or ounces to pounds) and round to the nearest 100th –2 numbers after the decimal.
  - Create a form to do a string concatenation of your first and last name.
- Read about MySQL at W3Schools

Next week—
Introduction to MySQL

- What is MySQL?
- Relational database structure
- Create tables in a database
- Use PHP to insert and show data

Questions?