

PHP and HTML forms

cs380

Presentations

- Unix information security — Steve
 - https://docs.google.com/presentation/d/1oLVJBdRubKid4YpvbFHcFHU-kp9UPRPvaPQUOeNVRhA/edit#slide=id.g98addb75cc_o_6
- Sql Insert — Saumyaa
 - <https://docs.google.com/presentation/d/1YKdM7NXATl8PecSnG9ojuqHQVDV4NpTBKFrIUJMQUXo/edit#slide=id.p>
- SQL Delete — Lipi
 - https://docs.google.com/presentation/d/1jIMZo5RHcJ5cGUX-4RL5YGoLQstkuFOuUqcnOfhR6Es/edit#slide=id.g9aoeece853_o_10
- SQL Update — Rosie
- SQL Create — Zainab

Super Global variables

- `$_SERVER`

- where the script is running
- and lots of other stuff

- `getallheaders()`

- a subset of `$_SERVER`
- visible in browser developer console

- `$_COOKIE`

- the cookies set for this page

- `$_GET`

- `$_POST`

- submissions from forms

- `$_REQUEST`

```
<html><body>
<?PHP
print_r($_SERVER);
?>
</body></html>
```

Server.php

```
<html>
<head>
<style>
table, tr, td, th { border:5px solid #dd0000; border-collapse:
</style>
</head>
<body>
<table>
<?PHP
foreach (getallheaders() as $key=>$val) {
    print("<tr><td>$key</td><td>$val</td></tr>");
}
?>
</table>
</body>
</html>
```

headers.php

PHP from the command line

- invoke
- UNIX>php something.php
- UNIX>php -f sometings.php

- php args.php now=then
then=name name=geoff

args.php

```
<?php
if(defined('STDIN') ) {
    echo "Running from CLI\n";
    $_GET=array();
}

if ($argc>1) {
    for ($i=1; $i<$argc; $i++) {
        echo $argv[$i] . "\n";
        $av = explode("=", $argv[$i]);
        $_GET[$av[0]]=$av[1];
    }
}

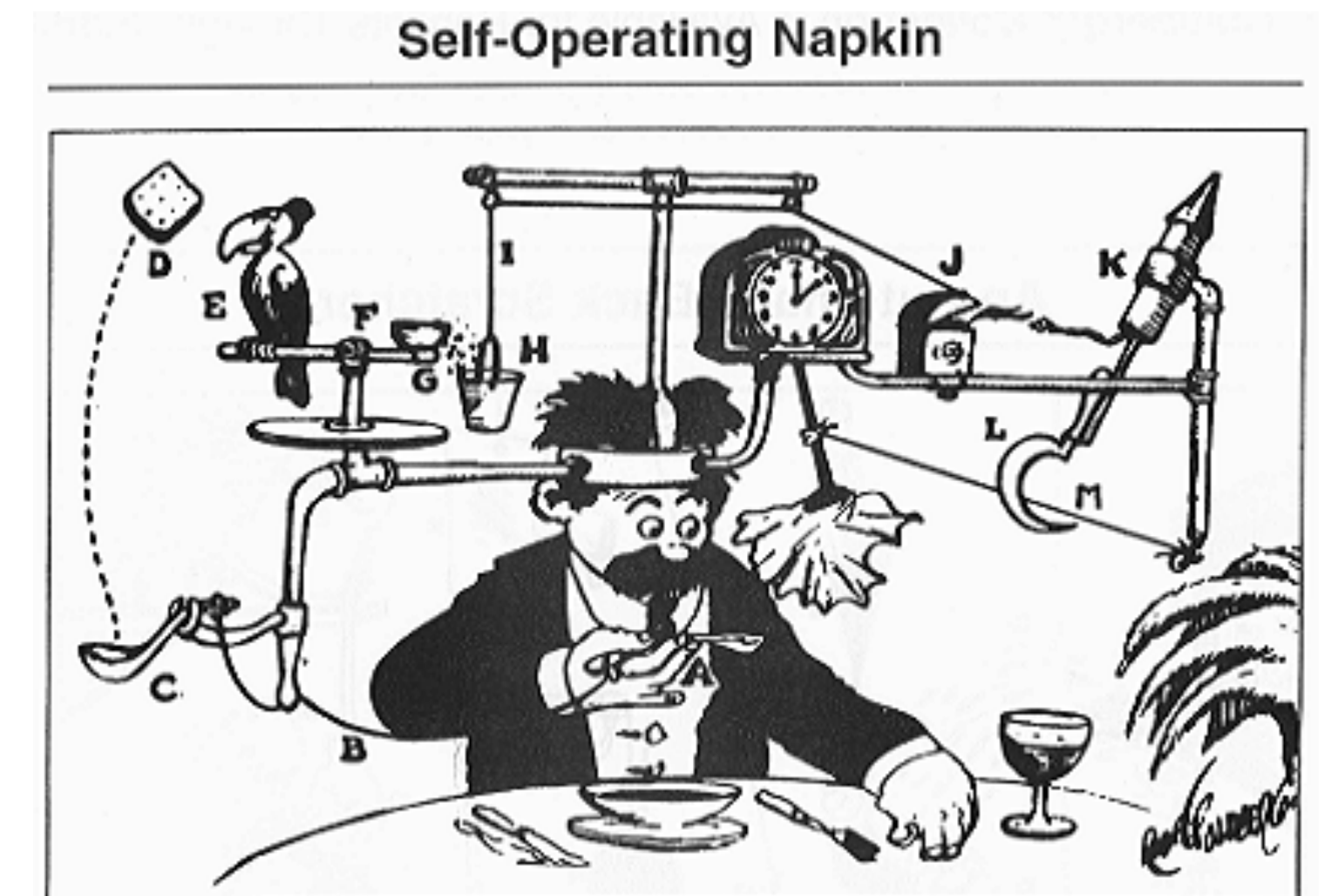
print_r($_GET);
```

Running PHP

- You must run PHP on machines onto which PHP has been installed
- To view php programs through a browser the web server must be configured for PHP
- Macs have php installed. You can configure the internal web server for PHP if you are so motivated
- comet.cs.brynmawr.edu is configured with PHP
 - Put files with .php extension into you public_html directory
 - <http://comet.cs.brynmawr.edu/~gtowell/380/Lec4/hw1.php>
 - If you miss “comet” what happens will depend on your browser and the server. On my mac, the file download to my computer

PHP Practice

- Write PHP to create an HTML page with a table of the first 50 fibonacci numbers
- Write PHP to create a table that prints all prime numbers less than 1000.
 - Use the “sieve of Eratosthenes” algorithm
 - Use at least 1 function and 1 global array for this task.
 - https://en.wikipedia.org/wiki/Sieve_of_Eratosthenes
 - once you have written a basic seive, rewrite it using “map” rather than array in which initial map only stores potentials that are not multiple of 2 or 3 (or 5 or 7).
 - How much larger a prime can you compute in the space used by 1000?
- Write a “Rube Goldberg”-ish hello world program in PHP.
Use arrays, functions, loops, ...
 - alternately phrased, obfuscated PHP code
 - in the end it should just print “hello world”



PHP review: My Rube Goldberg

```
<html>
<body>
<?php
function itoc($int) {
    return chr(ord('a')+$int);
} ?>
<?php
$silly = array(array(7, 4, 11, 11, 14), array(22, 14, 17, 11, 3));
$colors = array("red", "green", "aqua", "blue" );
?>
<?php foreach ($silly as $wa) {
    foreach ($wa as $wci) {
        $ww=$wci % count($colors);
        echo("<span style=\"color:$colors[$ww]\">" . itoc($wci) . "</span>");
    }
    echo(" ");
}
?>
</body>
</html>
```

Topics

- Get / Post
- PHP — single vs double quotes

Responsive PHP

Using PHP to dynamically create an essentially static page is pretty boring

```
<html>
<?php
if (strpos(getallheaders()["User-Agent"], "Chrome")) {
    echo "<body style=\"background-color:green\">";
} else {
    echo "<body style=\"background-color:yellow\">";
}
?>
</body>
</html>
```

Then at least the page changes as you change browsers.

Far more important, use and receive User Input

User Input

HTML Forms

- Simple forms allow asking users questions or for information, or even for files
- Forms have
 - attributes:
 - action: specify where the information is to be sent
 - This can also be a JavaScript function
 - method: how the info should be sent
 - either get or post
 - Elements
 - each element should have name and type attributes.
 - a button type that causes the action
 - names are passed with data

```
<html>
<head>
<style>
.button { display: block; border: none;
background-color: #4CAF50; color: white;
padding: 14px 28px; font-size: 16px;
cursor: pointer; text-align: center;
border: 1px solid transparent; border-radius: 4px;
border-color: black; margin-top: 10px; }
.button:hover { background-color: #ddd; color: black;
}
</style>
</head>
<body>
<center>
<h1> Forms!!!</h1>
<span style="color:f">Hello</span>
<form action="rcvr.php" method="post">
  <input type="hidden" id="ttime" name="ttime" value="3487">
  <table>
<tr><td>Give me some text</td><td><input type="text"
name="textdata">
</td></tr>
<tr><td colspan=2 align="center">
  <input class="button" type="submit" value="Submit time"
name="submit">
</td></tr>
</table></form>
</body></html>
```

Receiving Input

on the server side

- \$_POST, \$_GET
 - recall \$_SERVER,
 - \$_... are all special variables
- Really boring example, just prints the submission

```
<?php
$rr = rand(1,100);
?>
<!DOCTYPE html> <html> <body>
<?php
echo "<table border=\"1\">" ;
foreach ($_POST as $name => $value) {
    echo "<tr><td>" . $name . "</td><td>" . $value .
    "</td></tr>" ;
}
echo "</table>";

?>
<a href="form1.html">Back to form</a>
</body> </html>
```

Superglobal Vars

mentioned some of these in last class

- Available in all scopes
- Do not need “global” keyword
- `$_REQUEST = $_GET+$_POST+$_COOKIE`
 - use just get or post when you want to enforce method

`$GLOBALS`

list of global vars

`$_SERVER`

`$_GET`

`$_POST`

`$_FILES`

uploaded files

`$_COOKIE`

`$_SESSION`

`$_REQUEST`

`$_ENV`

like unix `printenv`, but often empty

\$_SESSION

- persist information about a “session”
 - Session is usually
 - browser specific
 - but shared across tabs in browser
 - machine specific
 - of fairly long duration (but you can define it to be any length)
 - Implemented in 2 parts:
 - a cookie (which PHP automatically puts) on client
 - a file on server

Session save is automatic

This MUST be first

```
<?php
session_start();
?>
<!DOCTYPE html><html><body>
<?php
$count = 1;
if (isset($_SESSION) &&
array_key_exists("count", $_SESSION)) {
    $count = $_SESSION["count"] + 1;
}
else {
    echo("No session<br>");
}
$_SESSION["count"]=$count;
print "<table border=\"1\">" ;
foreach ($_SESSION as $name => $value) {
    echo "<tr><td>" . $name . "</td><td>" .
$value . "</td></tr>" ;
}
echo "</table>";
?>
</body></html>
```

echo and print
do not actually require
parens

Forms and Sessions

- html same except for action
 - now “session.php”
- PHP
 - copy all POST into session
 - Note dynamic creation of keys
 - Print everything in session
- Stuff accumulates
 - This can get expensive

```
<?php
session_start();
?>
<!DOCTYPE html><html><body>
<?php
// Set session variables
$count = 1;
if (isset($_SESSION) &&
array_key_exists("count", $_SESSION)) {
    $count = $_SESSION["count"] + 1;
}
$_SESSION["count"] = $count;
foreach ($_POST as $name => $value) {
    $_SESSION[$name . "_" . $count] = $value;
}
echo "<table border=\"1\">" ;
foreach ($_SESSION as $name => $value) {
    echo "<tr><td>" . $name . "</td><td>" .
$value . "</td></tr>" ;
}
echo "</table>";
?>
<a href="form2.html">Back to form</a>
</body></html>
```

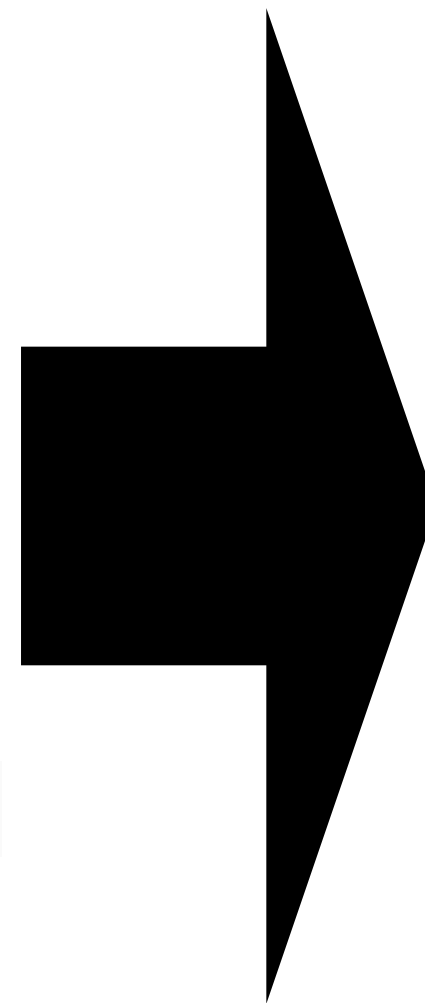
PHP

Event-driven and preparation/presentation

- Is PHP event-driven?
- Separating preparation from presentation?
 - use php “include” command
 - put all of your functions in a separate file (myfunctions.php)
 - or at least at top of the page
 - `<?php include(“myfunctions.php”) ?>`
 - try to separate html and php
 - often an unachievable goal

Separation

```
<?php
session_start();
?>
<!DOCTYPE html><html><body>
<?php
// Set session variables
$count = 1;
if (isset($_SESSION) &&
array_key_exists("count", $_SESSION)) {
    $count = $_SESSION["count"] + 1;
}
$_SESSION["count"] = $count;
foreach ($_POST as $name => $value) {
    $_SESSION[$name . "_" . $count] = $value;
}
echo "<table border=\"1\">" ;
foreach ($_SESSION as $name => $value) {
    echo "<tr><td>" . $name . "</td><td>" .
$value . "</td></tr>" ;
}
echo "</table>";
?>
<a href="formers.html">Back to form</a>
</body></html>
```



```
<?php
session_start();
function updateSession($parray) {
    $count = 1;
    if (isset($_SESSION) &&
array_key_exists("count", $_SESSION)) {
        $count = $_SESSION["count"] + 1;
    }
    $_SESSION["count"] = $count;
    foreach ($parray as $name => $value) {
        $_SESSION[$name . "_" . $count] = $value;
    }
}
?>
<!DOCTYPE html><html><body>
<?php updateSession($_POST); ?>
<table border=1>
<?php
foreach ($_SESSION as $name => $value) {
    echo "<tr><td>" . $name . "</td><td>" .
$value . "</td></tr>" ;
}
?>
</table>
<a href="formers.html">Back to form</a>
</body></html>
```

Maybe a
function here
also

Absolute path is safest in includes

Using Include

```
<?php
session_start();
include '/home/fields43/public_html/380/myfun.php';
?>
<!DOCTYPE html><html><body>
<?php updateSession($_POST); ?>
<table border=1>
<?php
foreach ($_SESSION as $name => $value) {
    echo "<tr><td>" . $name . "</td><td>" .
$value . "</td></tr>" ;
}
?>
</table>
<a href="formers.html">Back to form</a>
</body></html>
```

Just because include is within php block, does not imply that include is php. So you need <?php ... ?>

```
<?php
function updateSession($parray) {
    $count = 1;
    if (isset($_SESSION) &&
array_key_exists("count", $_SESSION)) {
        $count = $_SESSION["count"] + 1;
    }
    $_SESSION["count"] = $count;
    foreach ($parray as $name => $value) {
        $_SESSION[$name . "_" . $count] = $value;
    }
}
?>
```

SQL and PHP

- mysqli gives an interface that is almost identical to mysql UI.
 - Especially for queries
- Steps:
 - login and connect to database
 - do a query
 - check if there are results

```
$conn = new mysqli("localhost", "db_user",  
"", "shakespeare");
```

```
$result = $conn->query($q);
```

```
if ($result->num_rows > 0) {  
::  
}
```

The shakespeare database

```
use shakespeare;  
Database changed  
MariaDB [shakespeare]> show tables;
```

Tables_in_shakespeare
plays
slines

2 rows in set (0.000 sec)

```
MariaDB [shakespeare]> describe plays;
```

Field	Type	Null	Key	Default	Extra
playid	char(5)	NO	PRI	NULL	
pname	varchar(64)	YES		NULL	

2 rows in set (0.001 sec)

```
MariaDB [shakespeare]> describe slines;
```

Field	Type	Null	Key	Default	Extra
pid	char(5)	NO	PRI	NULL	
speaker	varchar(64)	NO		NULL	
lineinplay	int(11)	NO	PRI	NULL	
theline	varchar(2048)	NO		NULL	

4 rows in set (0.001 sec)

One random line

```
<?php
$conn = mysqli_connect("localhost", "db_user", "", "shakespeare");
// Check connection
if (mysqli_connect_error()) {
    die("Connection failed: " . $conn->connect_error);
}
$q = "with pp as (select playid,pname from plays as pl order by rand() limit 1) select pp.pname, speaker,
lineinplay, theline from slines as sl inner join pp on pp.playid=sl.pid order by rand() limit 1;";
echo $q . "\n";
$result = mysqli_query($conn, $q);
if ($result) {
    while ($row = mysqli_fetch_assoc($result)) {
        foreach( $row as $kk=>$vv ) {
            echo " " . $vv;
        }
    }
    echo "\n";
}
}
?>
```