Second Generation Technologies

…moving on from CGI

Key Concepts For Today

- Web development stacks
- Building a web application using non-CGI technologies
- Security Risks

Alternatives to CGI

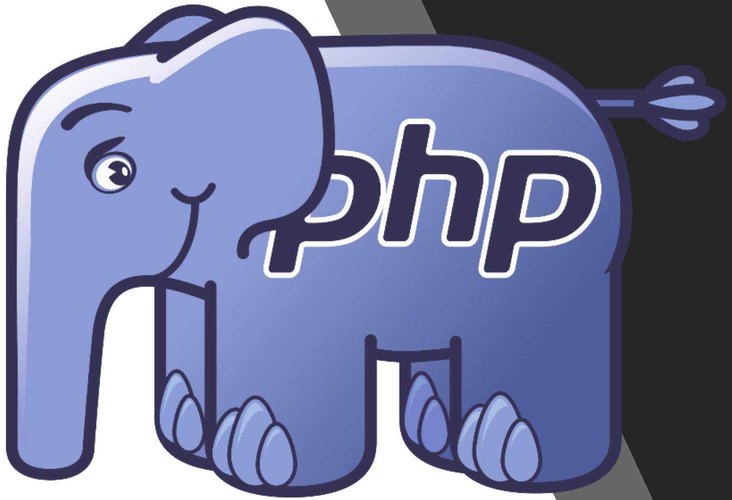
- Some are server side and some client side
  - Client side the code is sent to the user and processed locally
  - Server side the code is hidden from the user and processed on the server
- JavaScript
  - “The programming language of the Web”
  - HTML, CSS and JS are thought of as the three core technologies of the web
  - Usually used client side but also server side with node.js
- Active Server Pages
  - Second generation server-side technology from Microsoft.
  - Look for ‘.asp’ extensions
  - Was popular because it’s easy (especially when dealing with databases) and is from Microsoft
  - Now technologically obsolete

Alternatives to CGI

- ASP.net
  - Successor to ASP with a number of advantages
  - Often just called .NET but really .NET is a more general term
  - Heavily used by businesses which is why you need to have met it
- PHP
  - Open source alternative to ASP… we will get to this later today
  - Doesn’t require a specific web server (unlike ASP)
  - Look for ‘.php’ extensions

Some Web Development Stacks

- A web development stack usually contains an operating system, web server, database server and programming language
- There are a huge variety available…
- LAMP
  - Linux, Apache, MySQL or MariaDB, Perl or PHP or Python
- MAMP and WAMP – as above with Windows and Mac
- WISA
  - Windows Sever, Internet Information Services, SQL Server, ASP.NET
- MEAN
  - MongoDB, Express.js, AngularJS, Node.js
- MERN
  - MongoDB, Express.js, React.js, Node.js



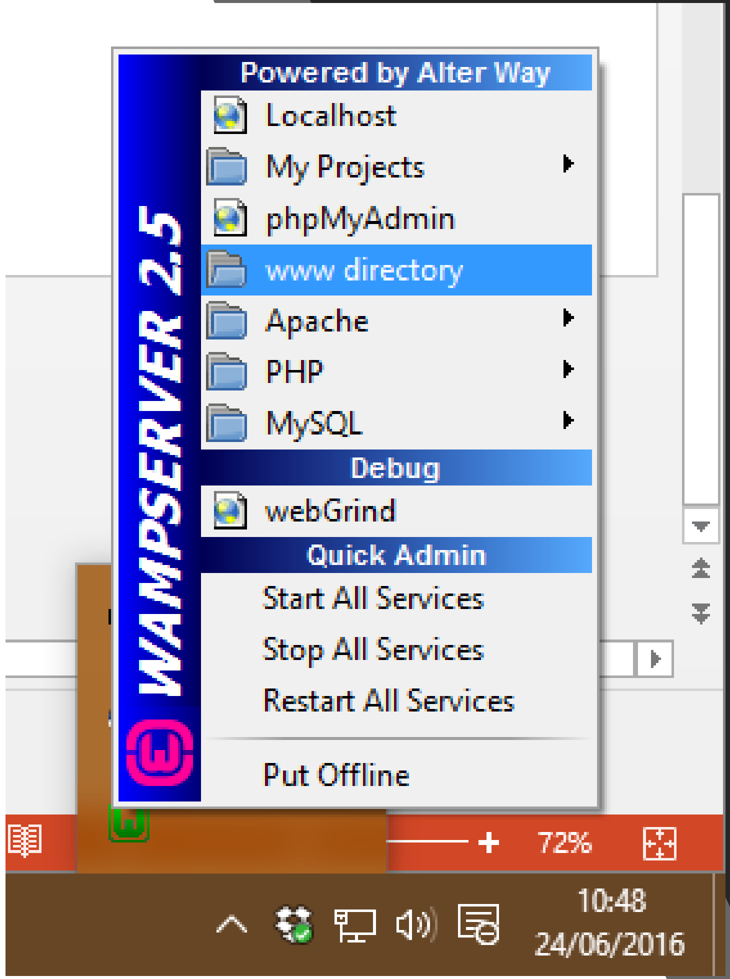
PHP

- Created in 1994 by Rasmus Lerdorf
  - Started life as a set of CGI scripts to track views of his **P**ersonal **H**ome **P**age
  - Over time functionality was added, it grew and was made public and open source
  - It is now one of the top ten web development languages
- Interpreted language
  - PHP parsing engine on the server parses PHP files and performs any PHP tasks
- Not as sophisticated as ASP.NET but much cheaper and easier to set up
  - …by the way ASP.NET is now open source but setting up a windows server usually costs more – but with ASP.NET Core it's possible to host on non-windows servers
- Unlike with ASP, PHP requires you learn and use it's own scripting language

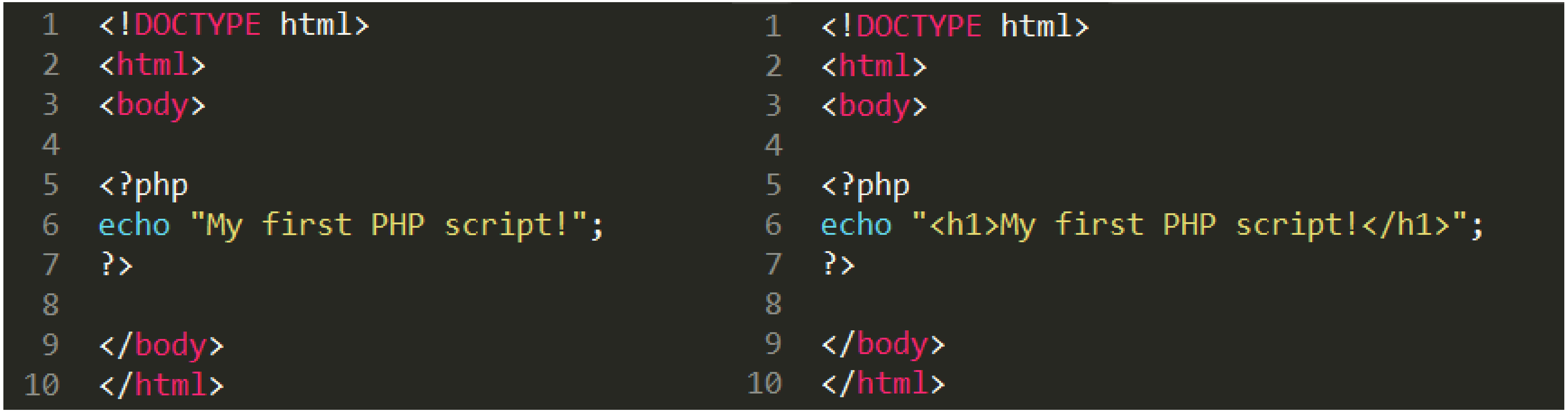
A warning…

- In the following examples I will be mixing html and php in the same files
- THIS IS BAD PRACTICE – Very much the old way of doing things
- We should always separate logic from the view for reasons we will go into next lecture

Local PHP development and testing



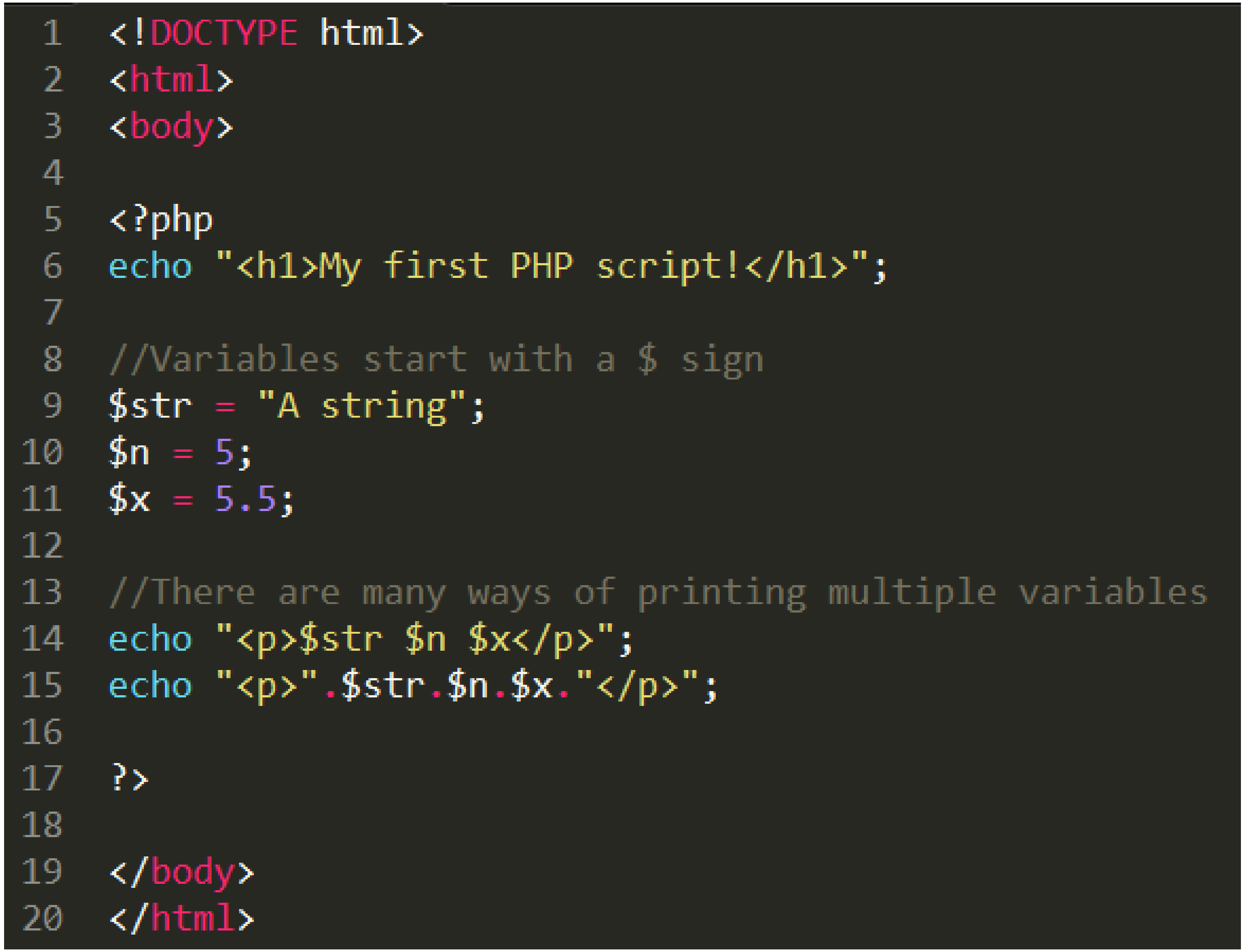
- We can’t just edit text files and view them in a browser we need a web server
- I use WampServer http://www.wampserver.com/en/ but XAMPP is multiplatform https://www.apachefriends.org/index.html
- Start server, left click on www directory to find your ‘public_html’ directory
- Then left click and click localhost to launch the site
- I usually create a new folder for each project I work on, these can then be viewed using ‘My Projects’



```
<!DOCTYPE html>
<html>
<body>

<?php
echo "My first PHP script!";
?>

</body>
</html>
```

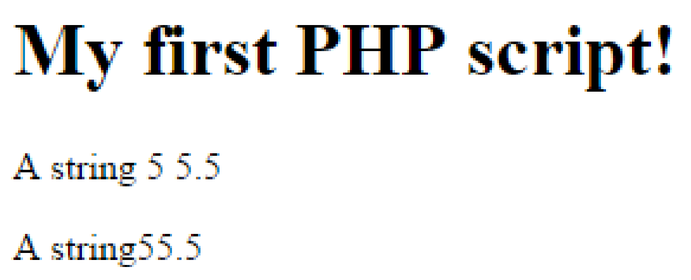
```
<!DOCTYPE html>
<html>
<body>

<?php
echo "<h1>My first PHP script!</h1>";
?>

</body>
</html>
```

- PHP is entered in between <?php ?> tags
- echo outputs variables
- If we put html tags in the output these will be interpreted by our browser



```
<!DOCTYPE html>
<html>
<body>

<?php
echo "<h1>My first PHP script!</h1>";

//Variables start with a $ sign
$str = "A string";
$n = 5;
$x = 5.5;

//There are many ways of printing multiple variables
echo "<p>$str $n $x</p>";
echo "<p>".$str.$n.$x."</p>";

?>

</body>
</html>
```

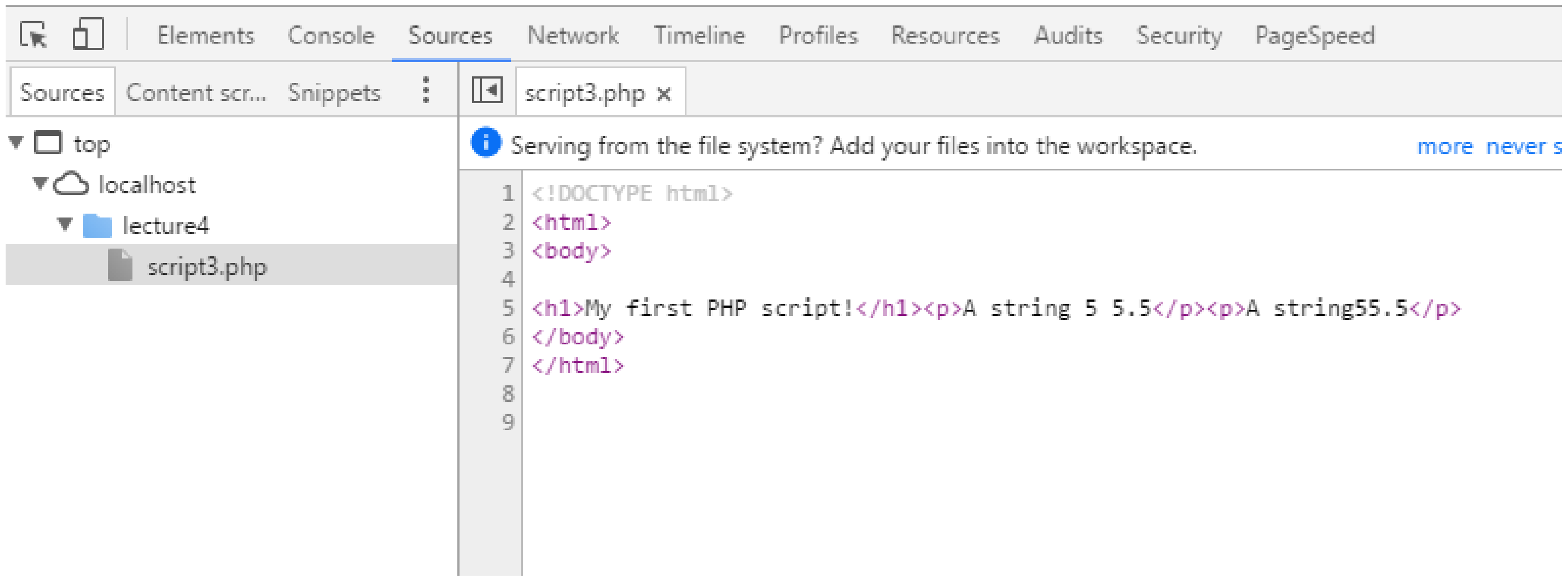
- Variables do not need declaring
- Variables must start with $
- This is to simplify parsing



My first PHP script!

A string 5 5.5

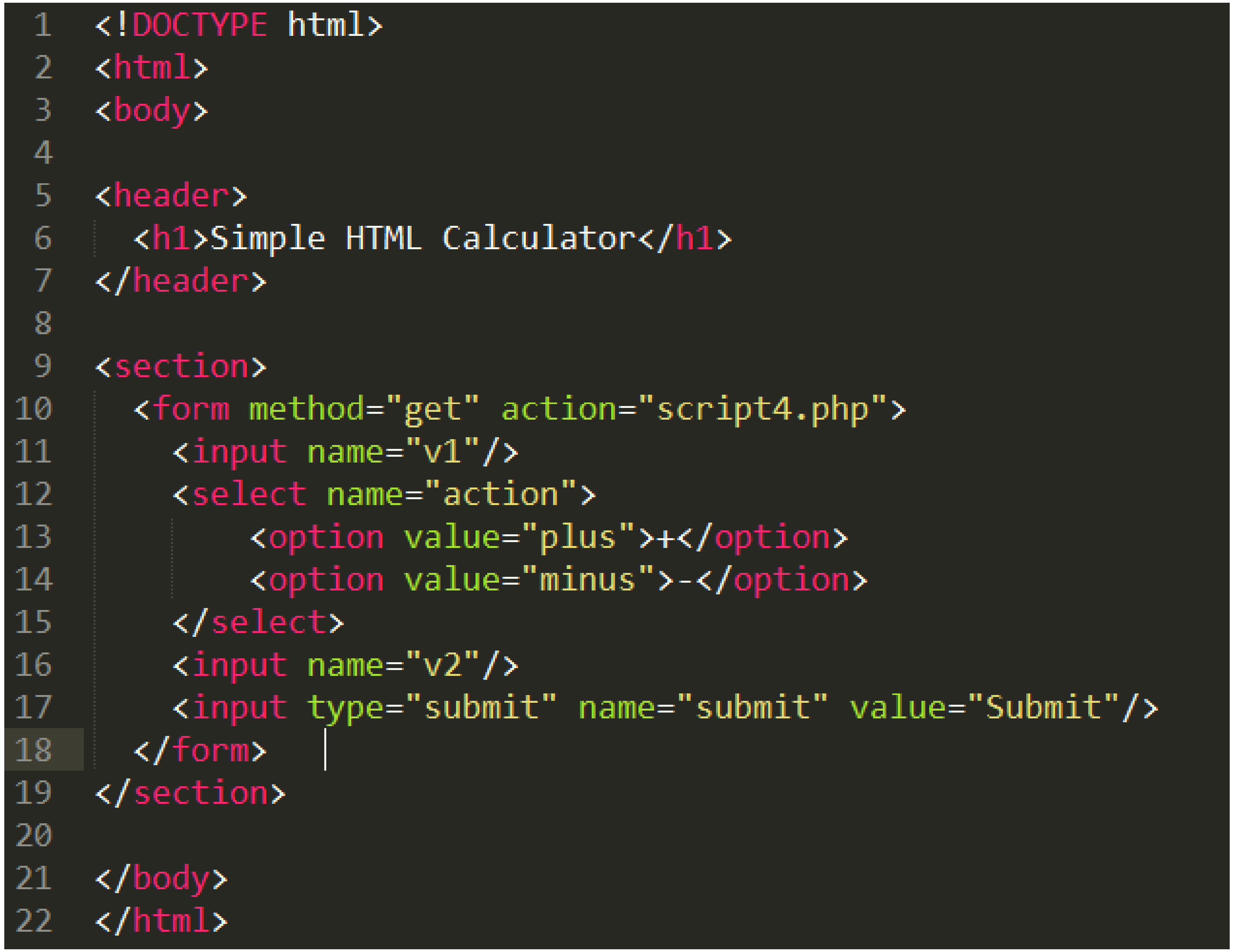
A string55.5



Serving from the file system? Add your files into the workspace.

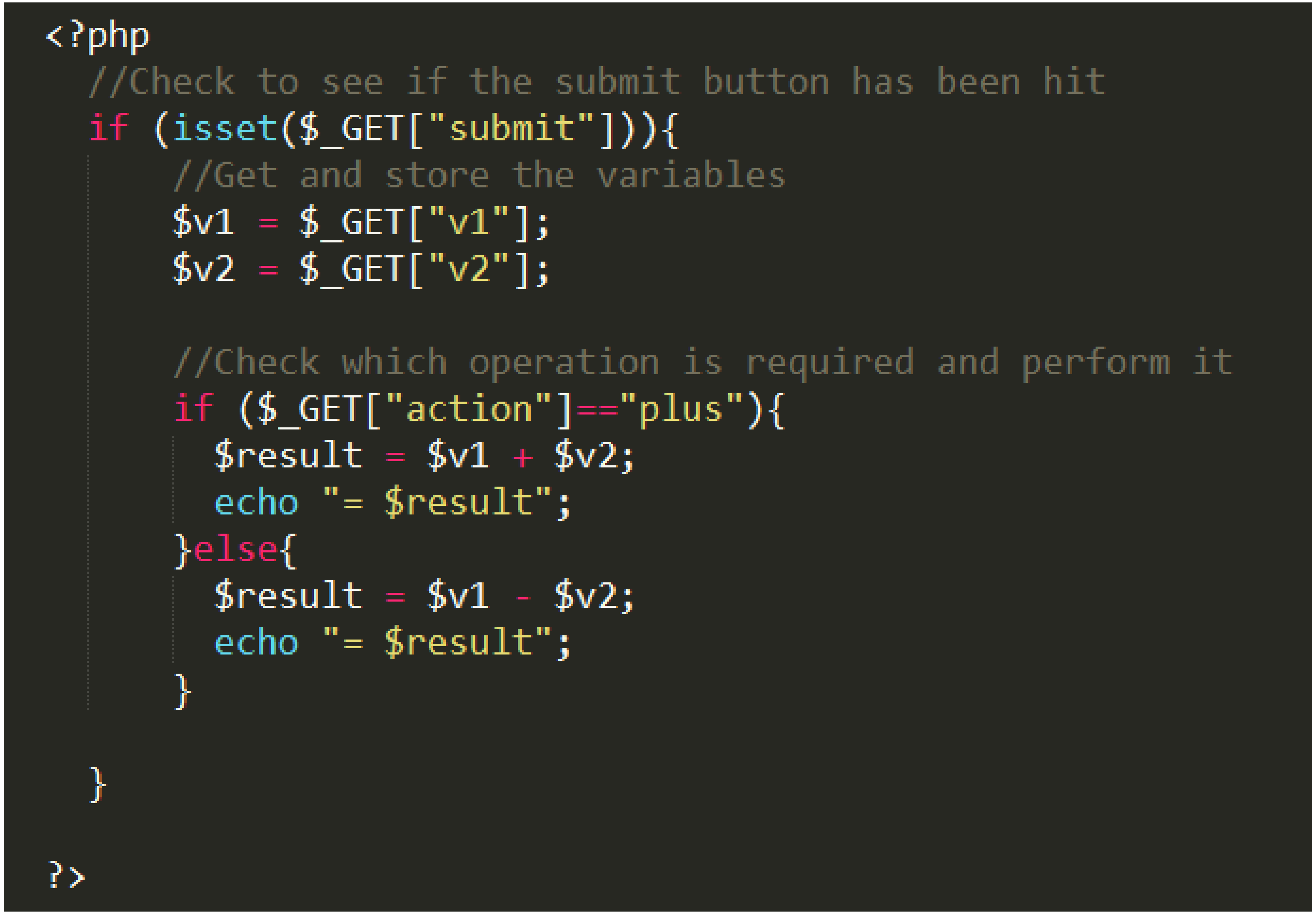
```
<!DOCTYPE html>
<html>
<body>

<h1>My first PHP script!</h1><p>A string 5 5.5</p><p>A string55.5</p>
</body>
</html>


```

- Note there is no php source in the html the browser receives…



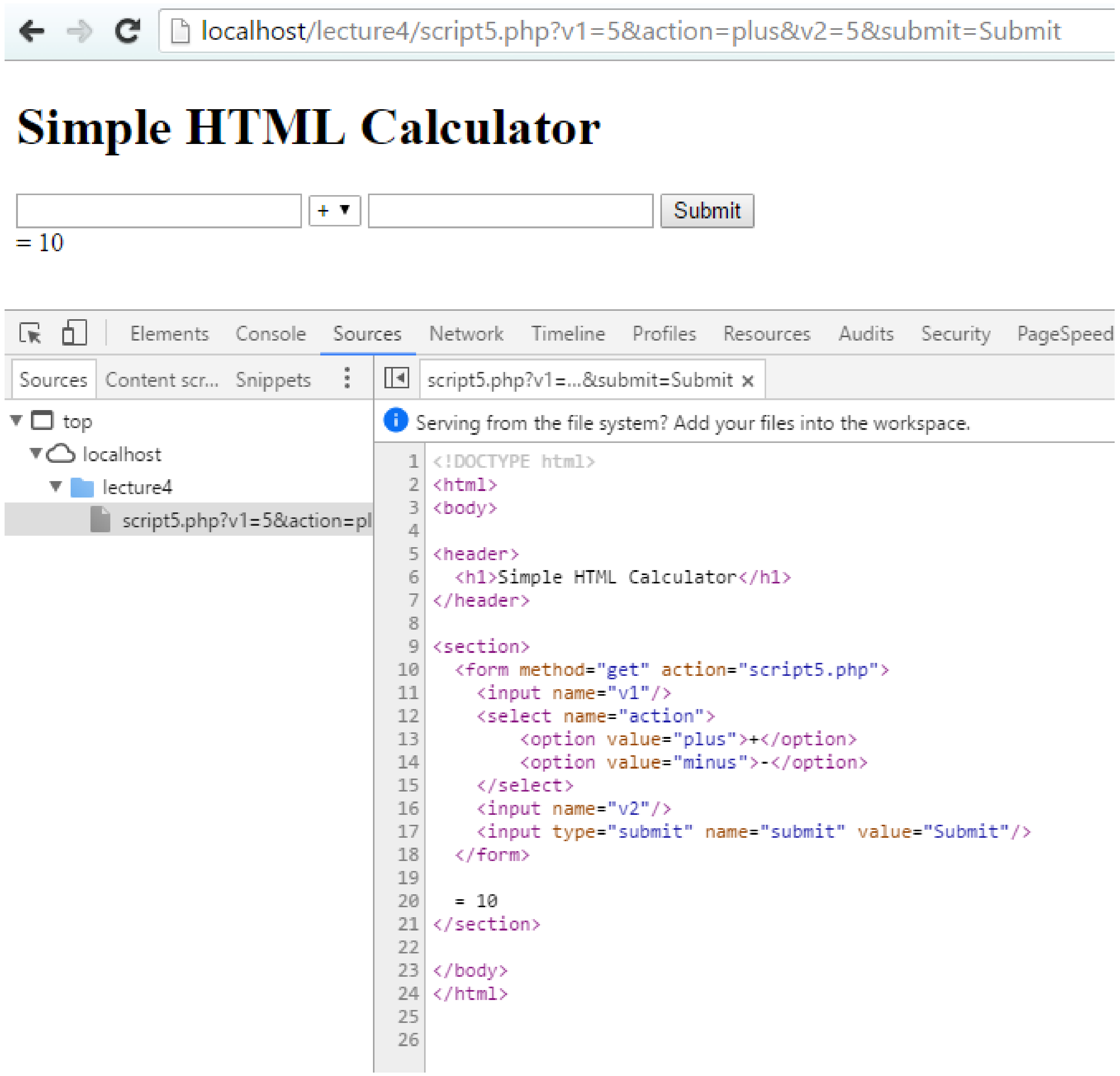
```
<!DOCTYPE html>
<html>
<body>

<header>
  <h1>Simple HTML Calculator</h1>
</header>

<section>
  <form method="get" action="script4.php">
    <input name="v1"/>
    <select name="action">
        <option value="plus">+</option>
        <option value="minus">-</option>
    </select>
    <input name="v2"/>
    <input type="submit" name="submit" value="Submit"/>
  </form>
</section>

</body>
</html>
```

- Building a simple calculator with php
- The html form works as before
- We don’t need the method and action since these are defaults, but I want to make sure you see what is going on



```php
<?php
  //Check to see if the submit button has been hit
  if (isset($_GET["submit"])){
      //Get and store the variables
      $v1 = $_GET["v1"];
      $v2 = $_GET["v2"];

      //Check which operation is required and perform it
      if ($_GET["action"]=="plus"){
        $result = $v1 + $v2;
        echo "= $result";
      }else{
        $result = $v1 - $v2;
        echo "= $result";
      }

  }

?>
```

- We need to check if the button has been hit because web applications have no concept of state



localhost/lecture4/script5.php?v1=5&action=plus&v2=5&submit=Submit

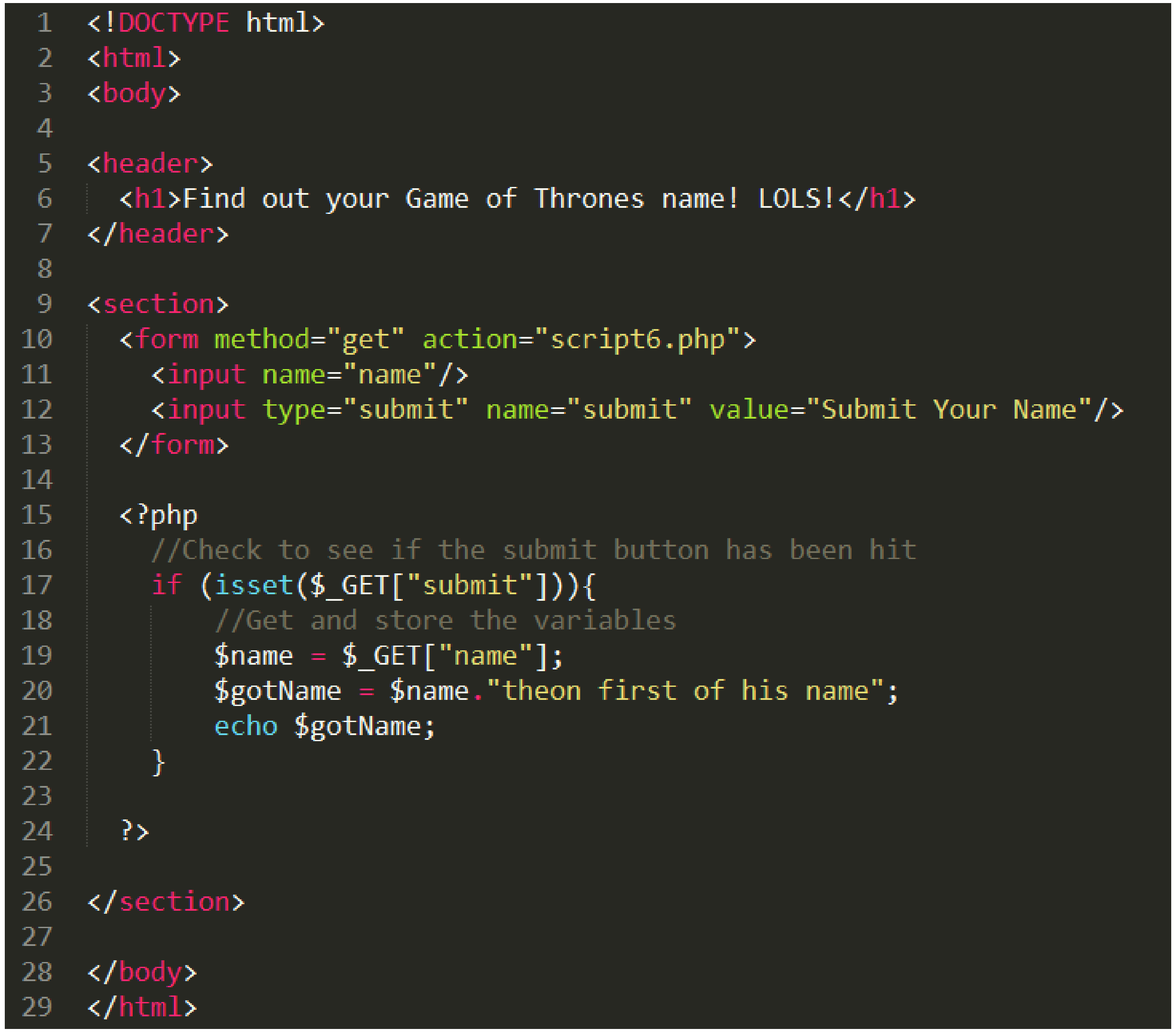
Simple HTML Calculator

= 10

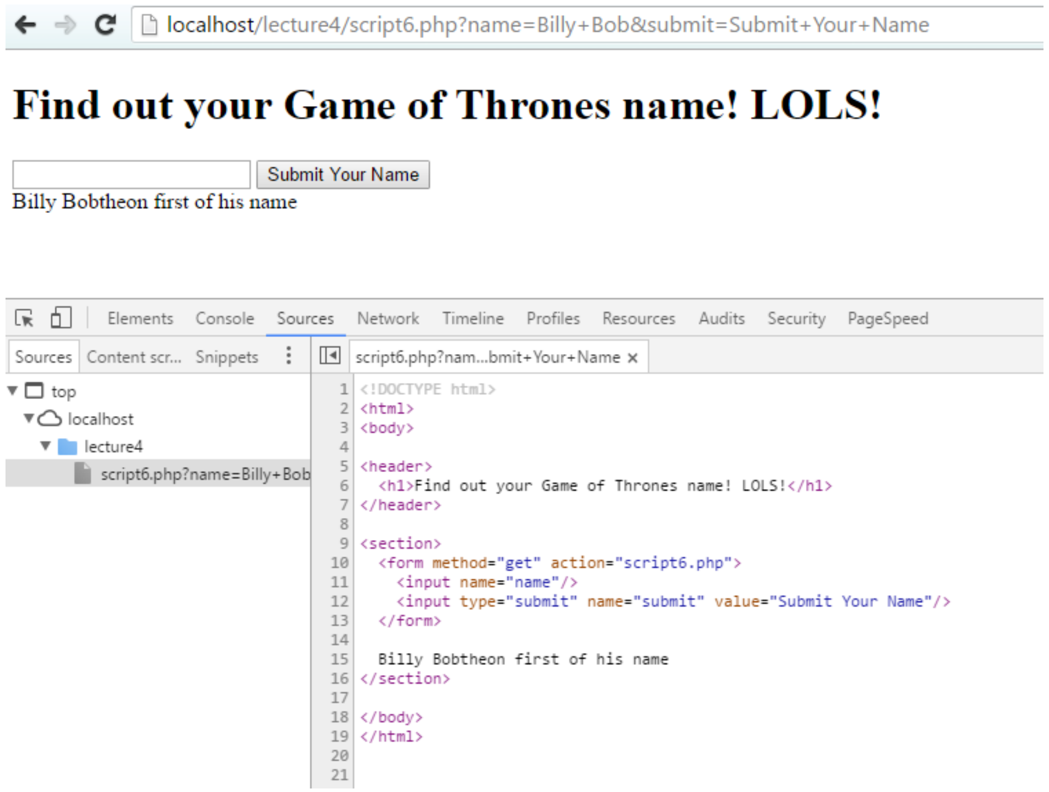
Serving from the file system? Add your files into the workspace.

```
<!DOCTYPE html>
<html>
<body>

<header>
  <h1>Simple HTML Calculator</h1>
</header>

<section>
  <form method="get" action="script5.php">
    <input name="v1"/>
    <select name="action">
        <option value="plus">+</option>
        <option value="minus">-</option>
    </select>
    <input name="v2"/>
    <input type="submit" name="submit" value="Submit"/>
  </form>

  = 10
</section>

</body>
</html>
```

- Notice how we have to refresh the whole page – AJAX is something we will come to later which addresses this limitation



```php
<!DOCTYPE html>
<html>
<body>

<header>
  <h1>Find out your Game of Thrones name! LOLS!</h1>
</header>

<section>
  <form method="get" action="script6.php">
    <input name="name"/>
    <input type="submit" name="submit" value="Submit Your Name"/>
  </form>

  <?php
    //Check to see if the submit button has been hit
    if (isset($_GET["submit"])){
        //Get and store the variables
        $name = $_GET["name"];
        $gotName = $name."theon first of his name";
        echo $gotName;
    }

  ?>

</section>

</body>
</html>
```



localhost/lecture4/script6.php?name=Billy+Bob&submit=Submit+Your+Name

Find out your Game of Thrones name! LOLS!

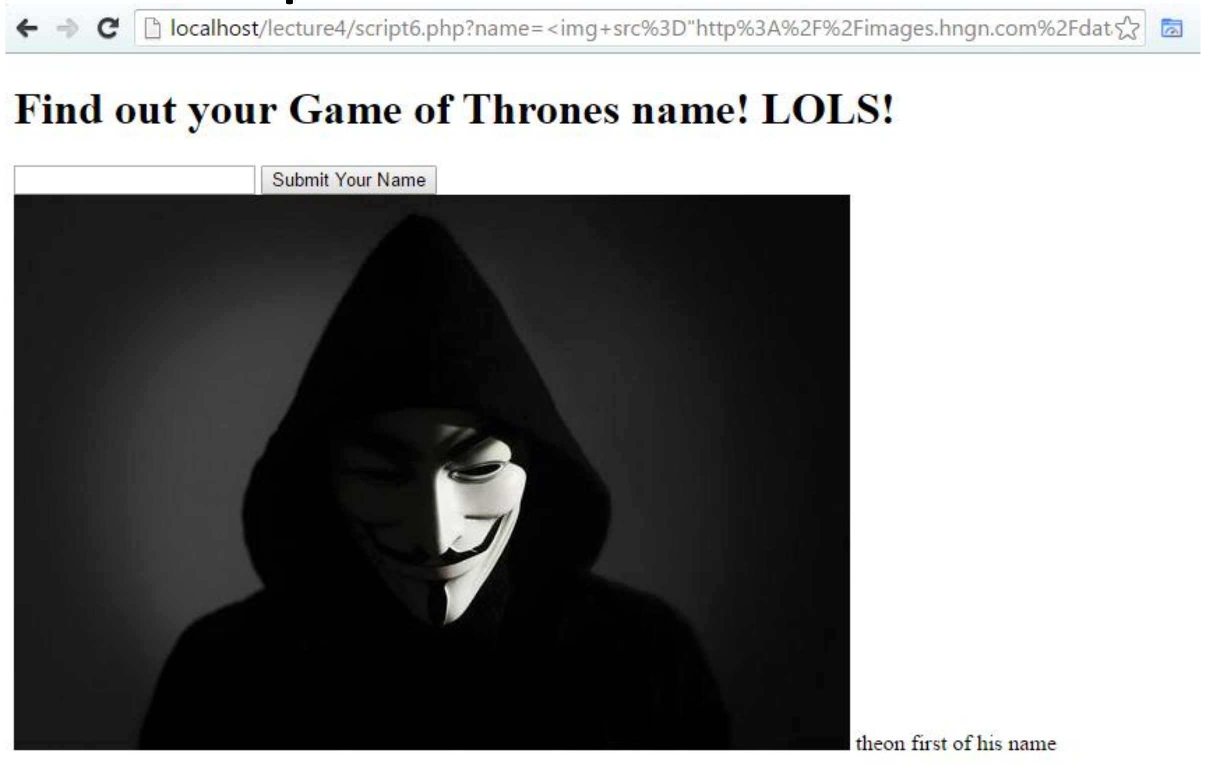
[ ] Submit Your Name

Billy Bobtheon first of his name

Elements Console Sources Network Timeline Profiles Resources Audits Security PageSpeed
Sources Content scr... Snippets
top
localhost
lecture4
script6.php?name=Billy+Bob
script6.php?nam...bmit+Your+Name

```html
<!DOCTYPE html>
<html>
<body>

<header>
  <h1>Find out your Game of Thrones name! LOLS!</h1>
</header>

<section>
  <form method="get" action="script6.php">
    <input name="name"/>
    <input type="submit" name="submit" value="Submit Your Name"/>
  </form>

  Billy Bobtheon first of his name
</section>

</body>
</html>


```

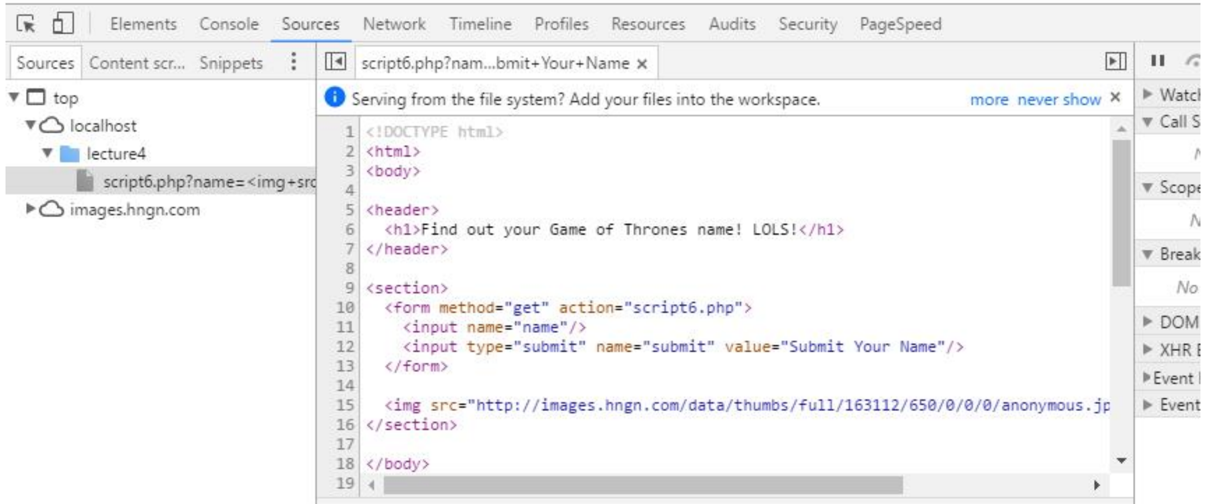
The importance of validation



localhost/lecture4/script6.php?name=<img+src%3D"http%3A%2F%2Fimages.hngn.com%2Fdat

Find out your Game of Thrones name! LOLS!

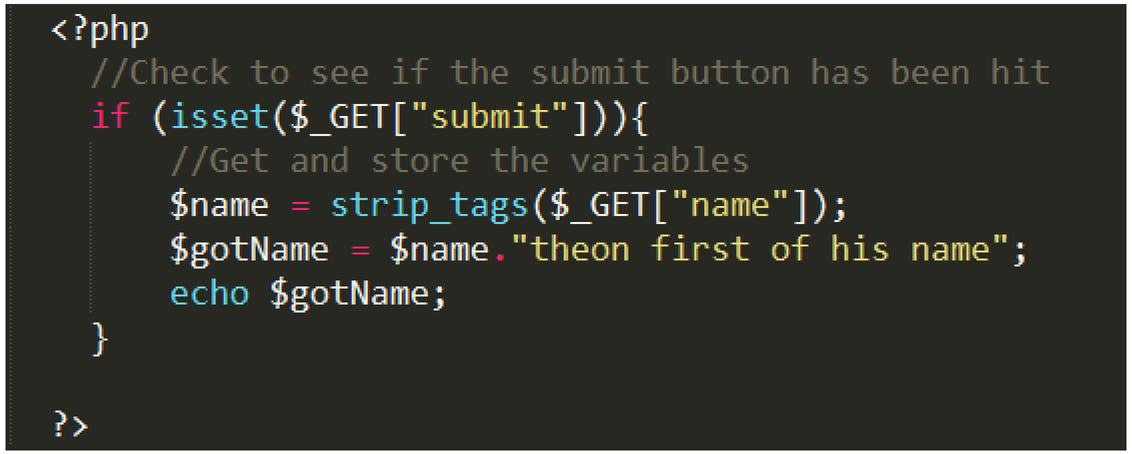
theon first of his name



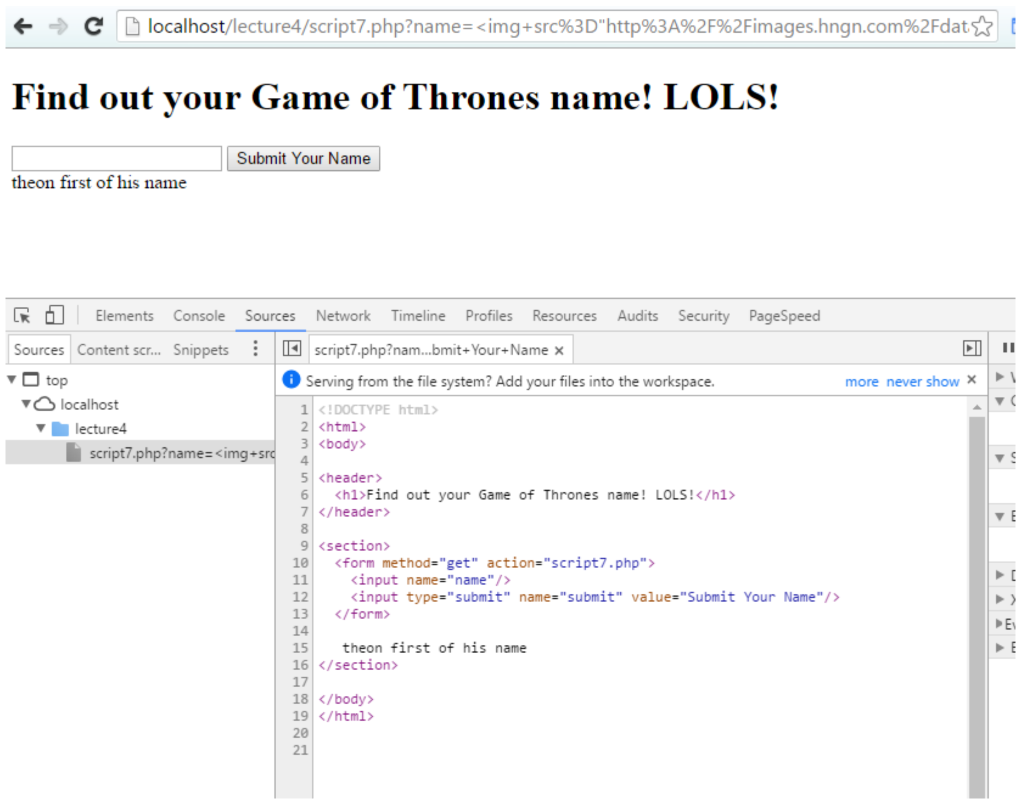
Serving from the file system? Add your files into the workspace. more never show

```
<!DOCTYPE html>
<html>
<body>

<header>
  <h1>Find out your Game of Thrones name! LOLS!</h1>
</header>

<section>
  <form method="get" action="script6.php">
    <input name="name"/>
    <input type="submit" name="submit" value="Submit Your Name"/>
  </form>

  <img src="http://images.hngn.com/data/thumbs/full/163112/650/0/0/0/anonymous.jp
</section>

</body>
```

- Working with strings is risky
- This example is essentially all client side… but what if this string was a username on a high score board which was stored and displayed to all visitors to your site?



```php
<?php
  //Check to see if the submit button has been hit
  if (isset($_GET["submit"])){
      //Get and store the variables
      $name = strip_tags($_GET["name"]);
      $gotName = $name."theon first of his name";
      echo $gotName;
  }

?>
```



localhost/lecture4/script7.php?name=<img+src%3D"http%3A%2F%2Fimages.hngn.com%2Fdat

Find out your Game of Thrones name! LOLS!

Submit Your Name

theon first of his name

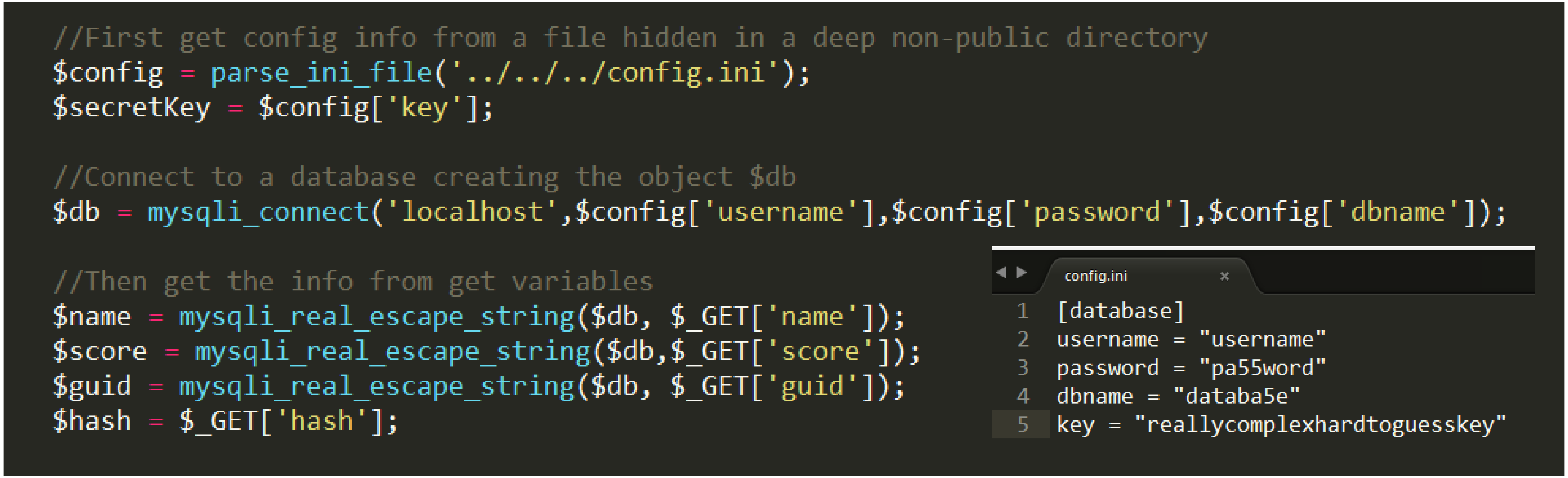
Serving from the file system? Add your files into the workspace. more never show

```html
<!DOCTYPE html>
<html>
<body>

<header>
  <h1>Find out your Game of Thrones name! LOLS!</h1>
</header>

<section>
  <form method="get" action="script7.php">
    <input name="name"/>
    <input type="submit" name="submit" value="Submit Your Name"/>
  </form>

   theon first of his name
</section>

</body>
</html>
```

- There are lots of functions and information to help you protect against these kinds of attacks (cross-site scripting)
- Your job is to always be thinking about it
- Think about security at the design stage…
- http://www.php.net/manual/en/security.php

Example – Submitting a score to a web database from a video game in Unity3D

- In Unity3D the function WWW(“url”) can be used to make an HTTP request and process the response
- In the game source code we can build a url with GET parameters to submit the score
- We need to stop users uploading any score they wish
- My, very basic, implementation in php is split over the next few slides
- Not perfect 100% secure implementation, but it is something I have used for real and it illustrates some principles

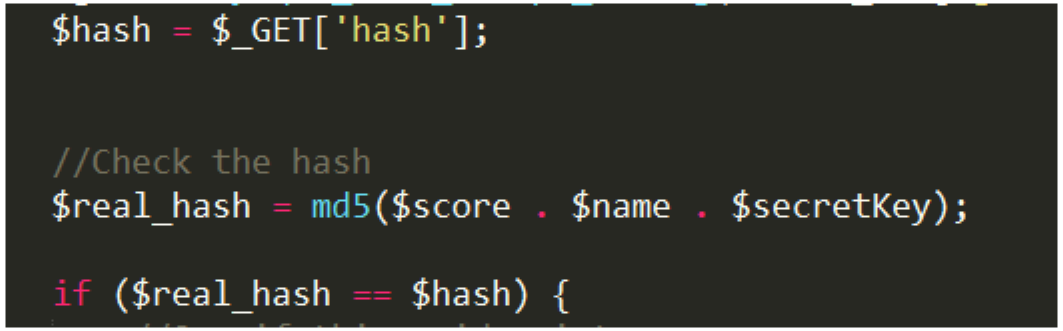


```php
//First get config info from a file hidden in a deep non-public directory
$config = parse_ini_file('../../../config.ini');
$secretKey = $config['key'];

//Connect to a database creating the object $db
$db = mysqli_connect('localhost',$config['username'],$config['password'],$config['dbname']);

//Then get the info from get variables
$name = mysqli_real_escape_string($db, $_GET['name']);
$score = mysqli_real_escape_string($db,$_GET['score']);
$guid = mysqli_real_escape_string($db, $_GET['guid']);
$hash = $_GET['hash'];
```

config.ini

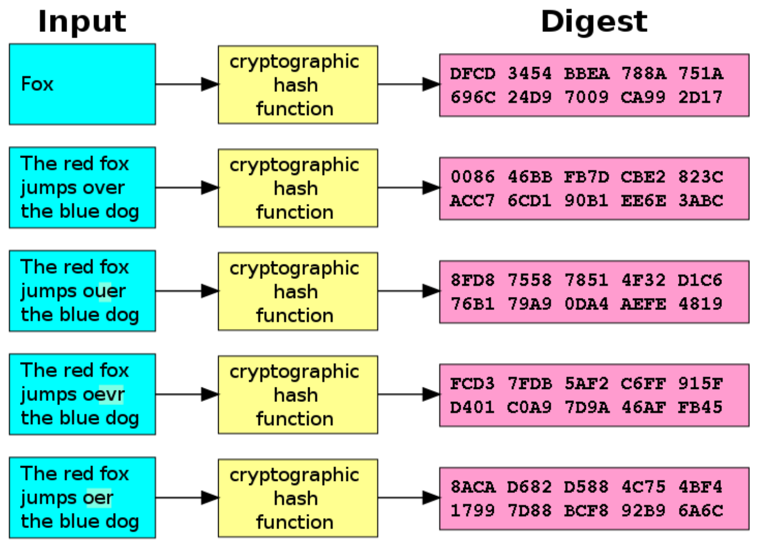
```ini
[database]
username = "username"
password = "pa55word"
dbname = "databa5e"
key = "reallycomplexhardtoguesskey"
```

- Config.ini has all our database information – arguably this would be better as another .php file just in case we put it in a public directory
- mysqli_connect connects us to the database
- mysqli_real_escape_string adds the escape character \ before potentially dangerous characters which could be used for an SQL injection… (http://www.w3schools.com/sql/sql_injection.asp)

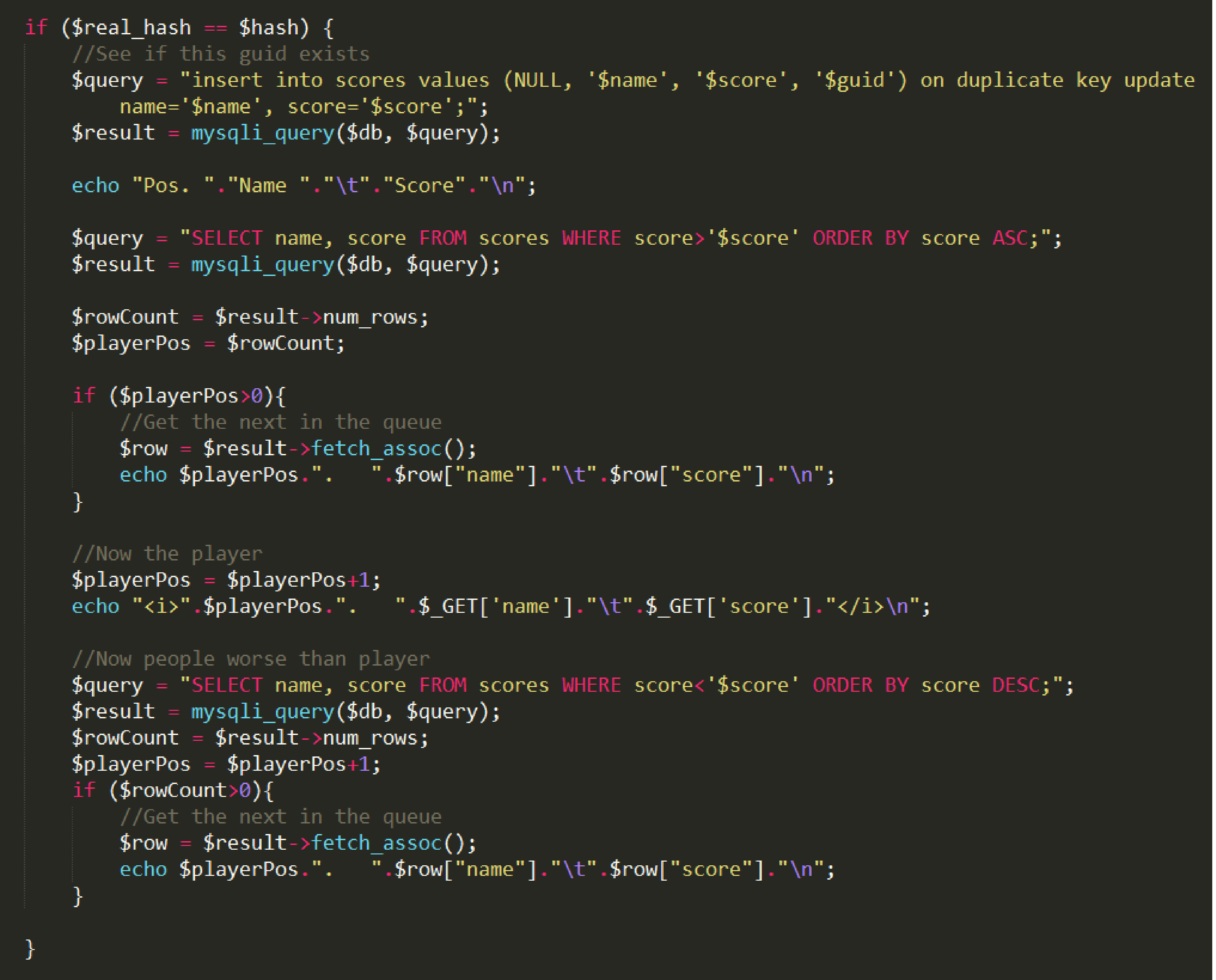


```
$hash = $_GET['hash'];


//Check the hash
$real_hash = md5($score . $name . $secretKey);

if ($real_hash == $hash) {
```

- In the game source code a string is made by joining the score, user name and a secret key
- That string is hashed and the hash is uploaded with the score and name
- On the server the hash is generated using the secret key stored there – if they don’t match we know not to allow data entry

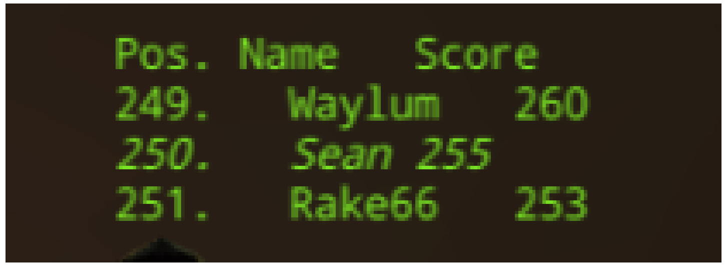

Input
Digest
Fox
cryptographic hash function
DFCD 3454 BBEA 788A 751A 696C 24D9 7009 CA99 2D17
The red fox jumps over the blue dog
cryptographic hash function
0086 46BB FB7D CBE2 823C ACC7 6CD1 90B1 EE6E 3ABC
The red fox jumps ouer the blue dog
cryptographic hash function
8FD8 7558 7851 4F32 D1C6 76B1 79A9 0DA4 AEFE 4819
The red fox jumps oevr the blue dog
cryptographic hash function
FCD3 7FDB 5AF2 C6FF 915F D401 C0A9 7D9A 46AF FB45
The red fox jumps oer the blue dog
cryptographic hash function
8ACA D682 D588 4C75 4BF4 1799 7D88 BCF8 92B9 6A6C

- Hash functions
  - One way
  - Quick to compute
  - Small change to input causes big changes to hash
  - Infeasible to generate the input from the output
- The MD5 hash function used here actually has vulnerabilities
- …but it is still challenging to break, in my application the reward for breaking security is your name at the top of list, which only the person in 2nd place sees – not worth the effort



```php
if ($real_hash == $hash) {
    //See if this guid exists
    $query = "insert into scores values (NULL, '$name', '$score', '$guid') on duplicate key update
        name='$name', score='$score';";
    $result = mysqli_query($db, $query);

    echo "Pos. "."Name "."\t"."Score"."\n";

    $query = "SELECT name, score FROM scores WHERE score>'$score' ORDER BY score ASC;";
    $result = mysqli_query($db, $query);

    $rowCount = $result->num_rows;
    $playerPos = $rowCount;

    if ($playerPos>0){
        //Get the next in the queue
        $row = $result->fetch_assoc();
        echo $playerPos.".    ".$row["name"]."\t".$row["score"]."\n";
    }

    //Now the player
    $playerPos = $playerPos+1;
    echo "<i>".$playerPos.".    ".$_GET['name']."\t".$_GET['score']."</i>\n";

    //Now people worse than player
    $query = "SELECT name, score FROM scores WHERE score<'$score' ORDER BY score DESC;";
    $result = mysqli_query($db, $query);
    $rowCount = $result->num_rows;
    $playerPos = $playerPos+1;
    if ($rowCount>0){
        //Get the next in the queue
        $row = $result->fetch_assoc();
        echo $playerPos.".    ".$row["name"]."\t".$row["score"]."\n";
    }

}
```

- The rest of the code updates that user's score and outputs a crude table with the player above and below him/her
- Unity3D then reads that string from the WWW function and prints it in the game UI
- This is an example of a web application not using a browser



```
Pos. Name    Score
249.   Waylum   260
250.   Sean 255
251.   Rake66   253
```

A quick aside on debugging

Bugs and You

From next lecture I expect you to be spending time outside of lectures coding and trying things

Stuff will break, and break a lot. You need to be comfortable with this as a developer.

Things to Remember when Debugging

- All bugs happen for a logical reason - you made a mistake or didn’t think of an edge case.
- If your code does something different than my code, or a tutorial you are following, then **you** did something different or are using a different version of the language/framework/both.
- You are going to be working with frameworks which have lots of interconnecting interacting pieces. Make sure you understand these… often the symptoms of a bug can be a long way from the cure. (Share Nintendo example)
- Understand what tools you have to debug and always use them!!
- Learn to like them, they make you learn things, and often show you errors in your code which may have made it into production!