

PHP and MySQL

A few more things...

Variables in PHP (review)

- Assign a variable using the assignment operator ("=")
 - Anything on the right gets inserted into anything on the left
 - Anything inserted replaces anything in the variable
 - In PHP, all variables start with \$
 - E.g. `$myVariable = "Hello";`
- Concatenation
 - Add to a variable using the concatenation operator (".=")
 - Same as the assignment operator, but instead of replacing, it appends
 - E.g.
`$myVariable = "Hello";`
`$myVariable .= " Professor!";`

Collect data from an HTML form

- Superglobals: `$_POST[]` or `$_GET[]`
- Pull data submitted via an HTML form
 - E.g. `$customerName = $_POST['name'];`

File 1: index.html

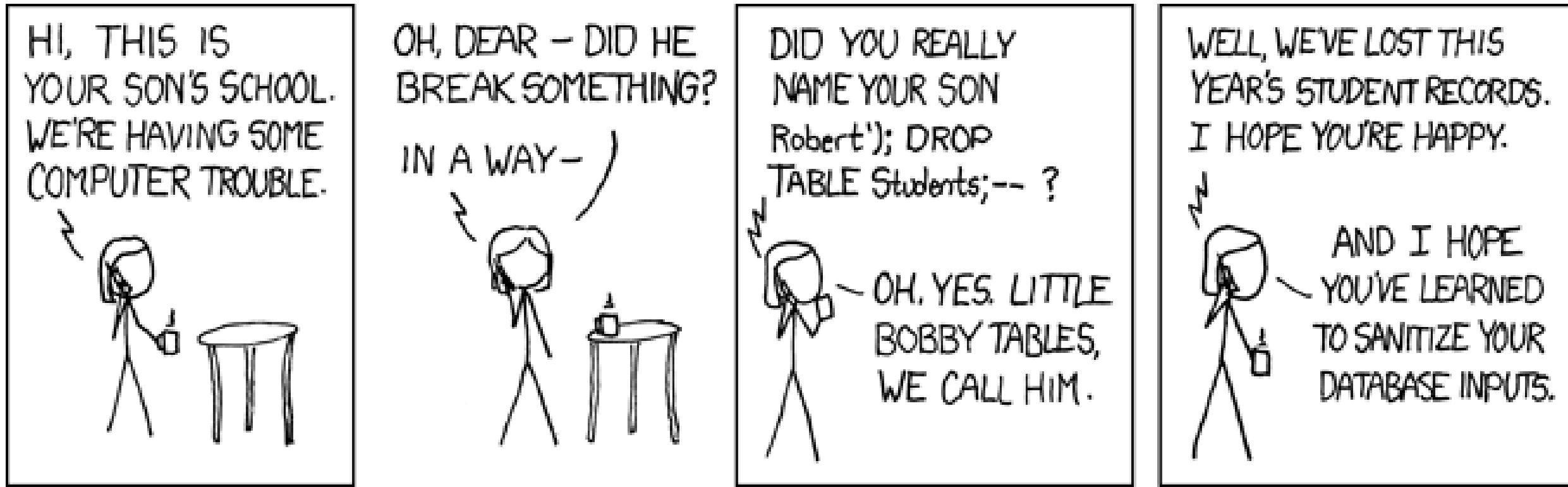
```
<form method="post" action="welcome.php">
  Name: <input type="text" name="name">
  E-mail: <input type="text" name="email">
  <input type="submit">
</form>
```

File 2: welcome.php

```
Welcome <?php echo $_POST["name"]; ?>
Your email address is: <?php echo $_POST["email"]; ?>
```

NEVER trust a user!

- What if your data comes from a form and a user types a quote into a data field?
- Can you see why this would be a problem?



Scrub incoming form data

- Nefarious users can trick servers to run code by entering it into a web form and submitting it
- Protect your web server by scrubbing any text inputs
- Use the PHP trim() and stripslashes() commands

```
$message = Trim(stripslashes($_POST['message']));
```

or...

```
$message = mysqli_real_escape_string($connection,  
$_POST['message']);
```