Accessing Form Data with PHP

pluralsight
hardcore developer training

Lesson Objectives

In this lesson we will:

- Learn about HTML forms
- Access form data from a PHP script
- Perform input validation
- Write a "Contact Us" form for our library

HTML Forms

- An HTML form contains fields for users to input data

```
<form  action="contactus.php" method="get">
…
</form>
```

Submit the form to this page for processing

The method used to send the form data to the page:
GET: The data is encoded into the URL
POST: The data is in the body of the HTML request

Form Elements

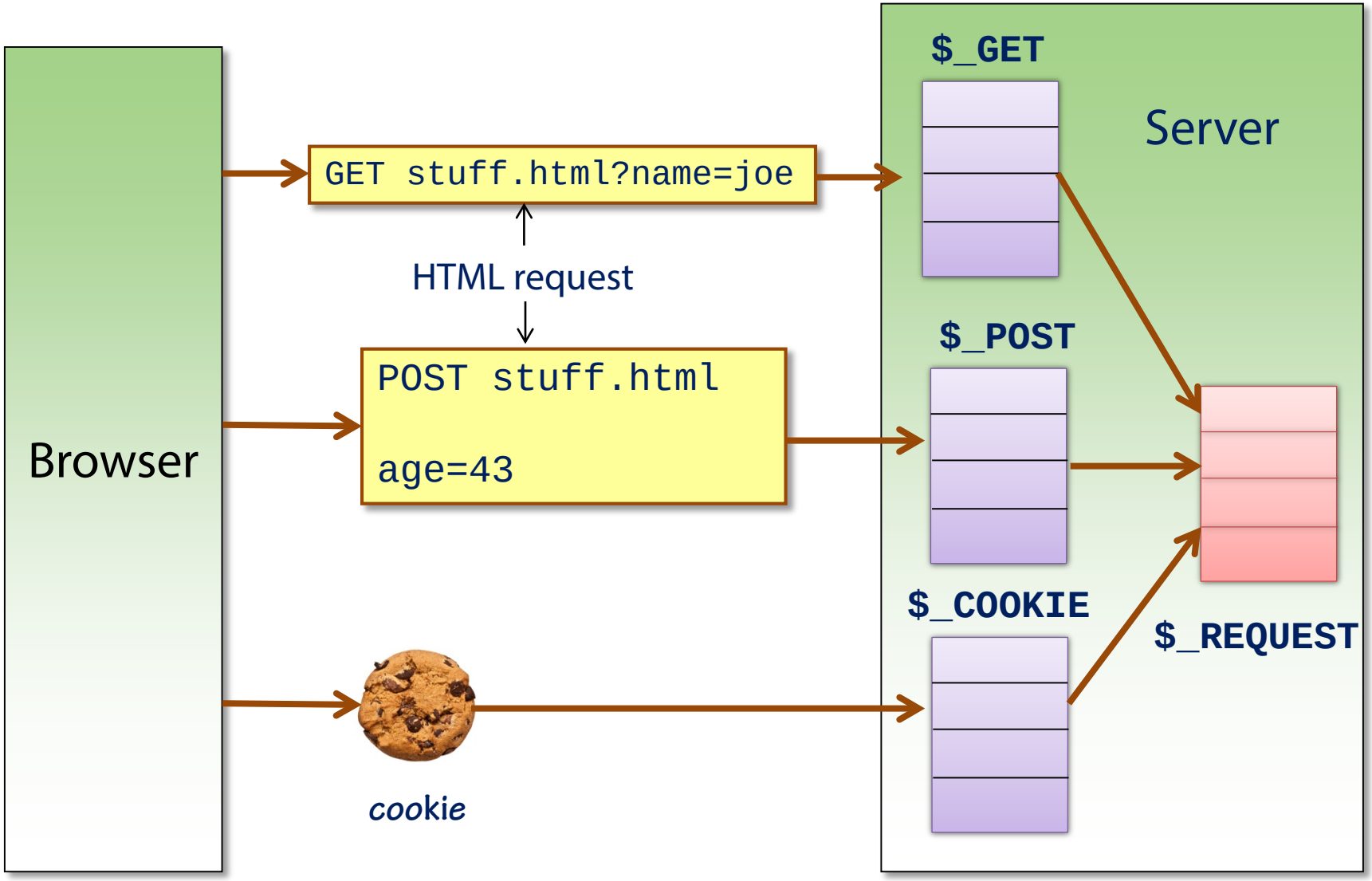
- **The `input` controls available within HTML forms include:**

Control	Description
`text`	**A single line text field**
`password`	**Text field with masked input**
`radio`	**Set of mutually exclusive selections**
`checkbox`	**Yes/no selection**
`submit`	**Button that posts the form data to a server**
`hidden`	**An invisible field that can be used to carry state information**

- **Other control types include:**
  - `textarea` A multi-line text input box
  - `select` A drop-down selection list

Demonstration: Creating a form

Passing Data to the Server



Retrieving Data from the Form

- In the form:

```
<form  action="contactus.php" method="post">
… <input type="text" name="customer">
  <input type="checkbox" name="ismember">
</form>
```

- In the script that the form is posted to:

```
<?php
  $customer = $_POST["customer"];
  $member=false;
  if (isset($_POST["ismember"]) $member=true;
?>
```

Demonstration: Accessing form data

Validating User Input

- **Good practice suggests validating all input that a user enters**
  - At minimum, verify that text boxes are not empty
  - Better still, verify input using PHP's string handling functions, or against a regular expression match
- **Example:**

```
<?php
  if ($_POST["customeremail"] == "") {
    echo "You did not enter an email address";
    exit;
  }
  if (! ereg("[a-z]+@[a-z]+\.[a-z]+",
             $_POST["customeremail"]) {
    echo "Email address is not valid";
    exit;
  }
?>
```

Demonstration: Validating User Input

Lesson Summary

- **HTML forms provide the user input for a web application**
  - Forms contain a variety of control types, each has a name attribute
- **Form data may be sent to the server using GET or POST methods**
  - From PHP, the data is retrieved from the $_GET and $_POST arrays
- **User input should be validated**
  - Simple check for non-blank field
  - Validation against regular expression match

Coming up in lesson 4 …