Multimedia im Netz

Online Multimedia

Winter semester 2015/16

Tutorial 03 – Major Subject

Today's Agenda

- Quick test
- Server side scripting: Stateful web apps
  - Cookies
  - Sessions
- Object-oriented programming with PHP
- Quiz
- Discussion of previous assignments

Quick Test

- We will distribute a 1-page test sheet in the tutorial
- Fill it out and hand it back
- Returned next week in the same tutorial slot
- Be prepared for the exam ;)

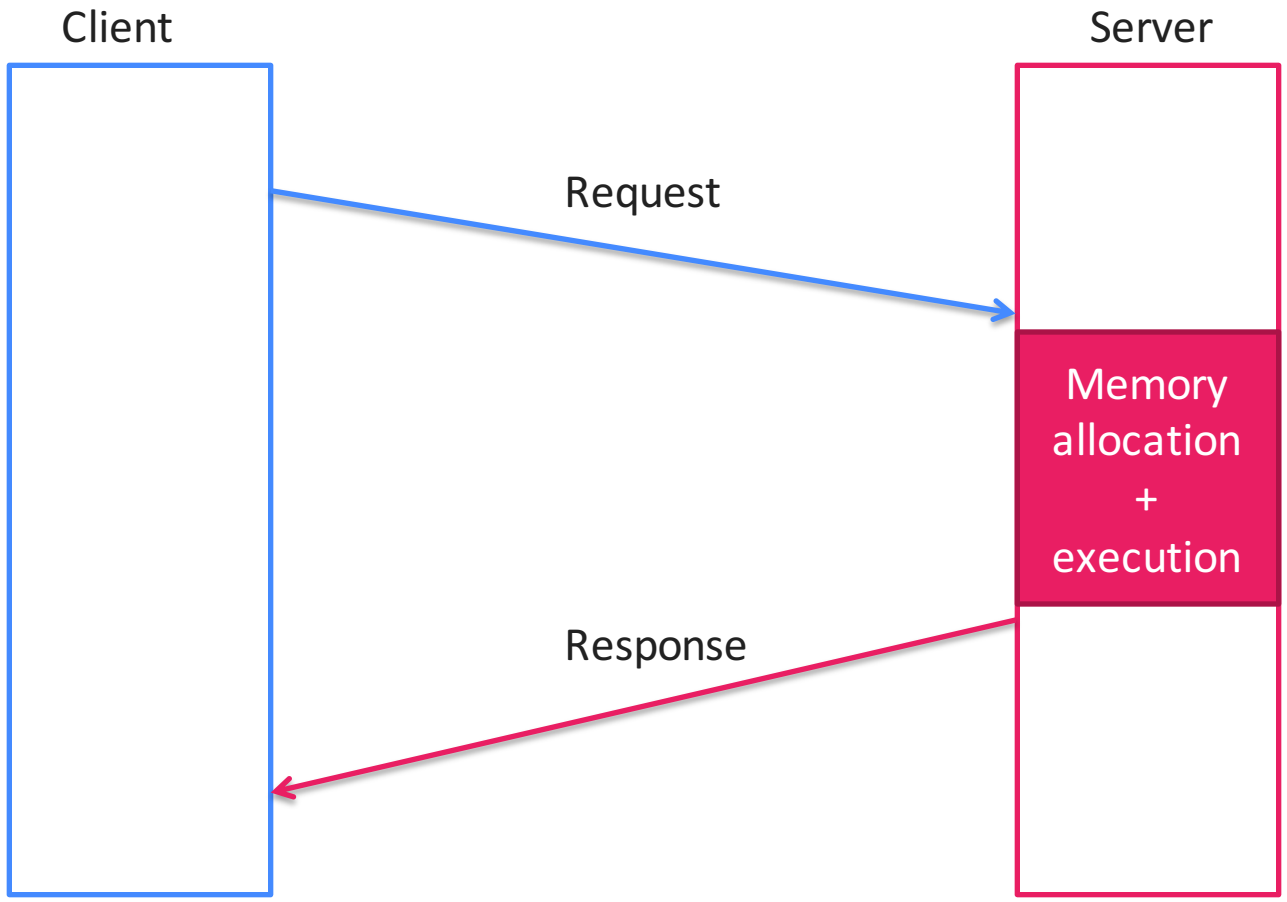
PHP and Statefulness: Sessions and Cookies

Break Out

- Visit a website where you have an account.
- Explore how the browser transmits cookies
- Which information is inside the cookies?
- Find out which websites stored cookies in your browser.

- Group discussion: What are the dangers of cookies, when are they harmless?

Problem: HTTP is stateless



Cookies

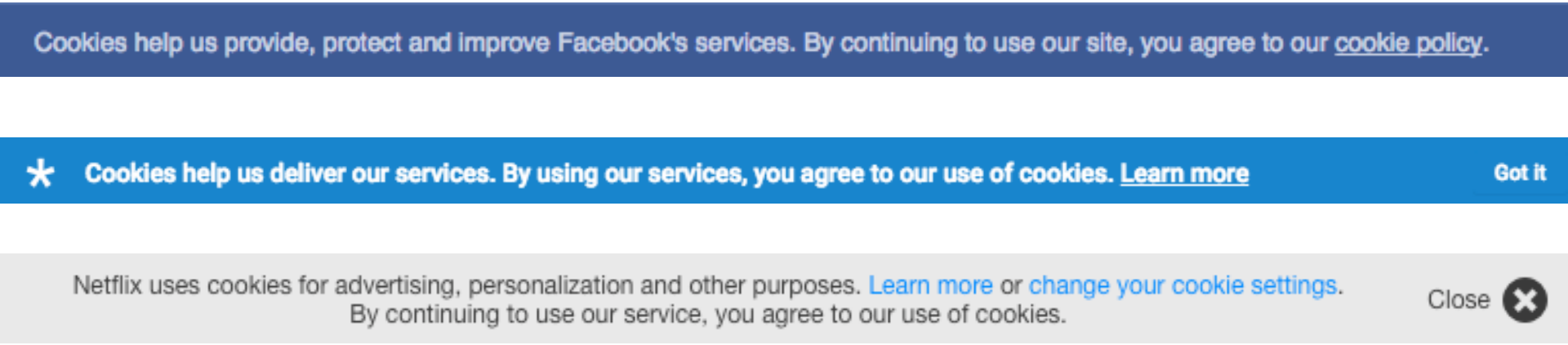
- Goals:
  - Persist information on the client side
  - Identify client
- HTTP cookies:
  - Stored in browser
  - Usually small, serialized data (text)
  - Sent with all request headers depending on current host URL
- Example usages:
  - Items in a shopping cart
  - Measure interaction (navigation on a site)
  - Authentication

Cookies are not…

- Necessarily evil:
  - Malware containers
  - Viruses
  - Spam
- A place to store large data
  - only small, serializable chunks
  - use the local storage API instead
- Dependent on server-side scripting
  - Also available with JavaScript

The Cookie Dilemma

- There is a “cookie law” that requires web site operators to inform the visitors about the use of cookies.
- Users do not necessarily read / understand / want this
- Almost all sites require cookies



Cookies in PHP

- Cookies belong to the HTTP header
  - Must be set before any output is generated
  - Before the <html> tag
  - Before any print / echo / var_dump statements!
- Create a cookie: setcookie(...)

```
setcookie("MMNCookie","Hello statefulness!");
```

- Read a cookie:

```
var_dump($_COOKIE['MMNCookie']);
```

  - reading is done on the **server**
  - so reading only works after the cookie is **sent back to the server**!
  - that is, after refreshing the page after cookie was set

Example: PHP cookies

```php
<?php
if(isset($_POST['name'])){
    setcookie('Name',$_POST['name']);
}
?>
<!DOCTYPE html>
<html>
[...]
<body>
<?php
if(isset($_COOKIE['Name'])){
    echo '<h1>Hello '.$_COOKIE['Name'].'</h1>';
}
?>

<form method="post">
    <label>Name: <input type="text" name="name"/></label>
    <input type="submit" />
</form>
</body></html>
```

Sessions

- Cookie disadvantage: Only stored on the remote client
- Sessions maintain “states” on the **server side**
- Store current state of variables as long as connected to the client
- On the client side, sessions are identified with a session ID cookie:
  - default cookie name in PHP: PHPSESSID
  - renaming possible with session_name()

Sessions with PHP

- Sessions need to be started before any output occurs (like cookies)
- Create session ID cookie:
  session_start()
- Delete the session ID cookie:
  session_destroy()
- Read / write session values:
  - superglobal `$_SESSION` array
  - immediately reset session like this `$_SESSION = array();`

Example: Counting visits

```
<?php session_start(); ?>
<!DOCTYPE html>
<html>
[...]
<body>

<?php

if(!isset($_SESSION['count'])){
    $_SESSION['count'] = 1;
}
else{
    $_SESSION['count']++;
}

echo '<p>Current count: '.$_SESSION['count'].'</p>';

?>
</body></html>
```

Example: Destroying Sessions

```php
<?php session_start(); ?>
<!DOCTYPE html>
<html>
[...]
<body>

<?php

if(isset($_POST['destroy'])) {
    session_destroy();
    $_SESSION = array();
}

if(!isset($_SESSION['count'])){
    $_SESSION['count'] = 1;
}
else{
    $_SESSION['count']++;
}

echo '<p>Current count: '.$_SESSION['count'].'</p>';

?>
<form method="post">
    <input type="submit" name="destroy" value="Reset"/>
</form>
</body>
</html>
```

Break Out: Hangman

- Create a “hangman” game with PHP.
- A Google search for “hangman” might inspire you.

- Use a hard-coded word first, then think about ways to randomize the word or have players compete against eachother

- Take 25 minutes time
- Present your solution to your peers

Object Oriented Programming in PHP

OOP Basics

- OOP paradigms and concepts (examples):
  - **classes** that turn into **objects** when instantiated
  - **inheritance**
  - **interfaces**
  - „Everything is an object“
- PHP class signature:

```php
<?php
class Lecture{
    //put members and methods here
}
?>
```

Define Member Variables

```php
<?php
class Lecture{
    var $title = "Online Multimedia";
    var $semester = "winter 2015/2016";
    var $professor = "Prof. Dr. Heinrich Hussmann";
    var $date = "Thursdays 10-13h";
}
?>
```

Using a Class

- Importing and instantiating a class;

```php
require_once("Lecture.php");
$mmn = new Lecture();
var_dump($mmn);
```

- Access to member values: arrow notation

```php
echo 'Title: '.$mmn->title;
echo 'Semester: '.$mmn->semester;
echo 'Professor: '.$mmn->professor;
```

Adding Methods to Classes

```php
<?php
class Lecture{
    var $title = "Online Multimedia";
    var $semester = "winter 2015/2016";
    var $professor = "Prof. Dr. Heinrich Hussmann";
    var $date = "Thursdays 10-13h";

    function setDate($date){
        $this->date = $date;
    }

    function getDate(){
        return $this->date;
    }
}
?>
```

Calling methods

```php
$mmn->setDate("Thursday morning");
echo $mmn->getDate();
```

Constructors

- PHP‘s constructors are methods with a special name: **__construct();**

```
function __construct($ttl, $sms, $prf,
$dt){
     $this->title= $ttl;
     $this->semester = $sms;
     $this->professor = $prf;
     $this->date = $dt;
}
```

- Use constructor:

```
$mmi = new Lecture("MMI",
    "Winter semester","Prof. Butz","Wendesdays");
```

Round-up Quiz

1. What does it mean to “serialize data”?
2. Why are cookies only accessible after a page refresh?
3. What is the difference between “cookies” and “sessions”?
4. Are cookies stored on the server or on the client?
5. What does session_destroy() actually do?
6. How do you define a member variable for a class?
7. How do you access a method with a given object?

Thanks!

What are your questions?

Let’s begin with the Assignment!

- Download the assignment sheet
- Start with task 1
- You can collaborate with your neighbor

- Turn in the assignment by November 9th, 12:00 noon via UniWorX