Multimedia im Netz
Online Multimedia
Winter semester 2015/16

Tutorial 03 – Minor Subject

Today's Agenda

- Quick test
- Server side scripting: Sessions with PHP
- Breakout task
- 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 ;)

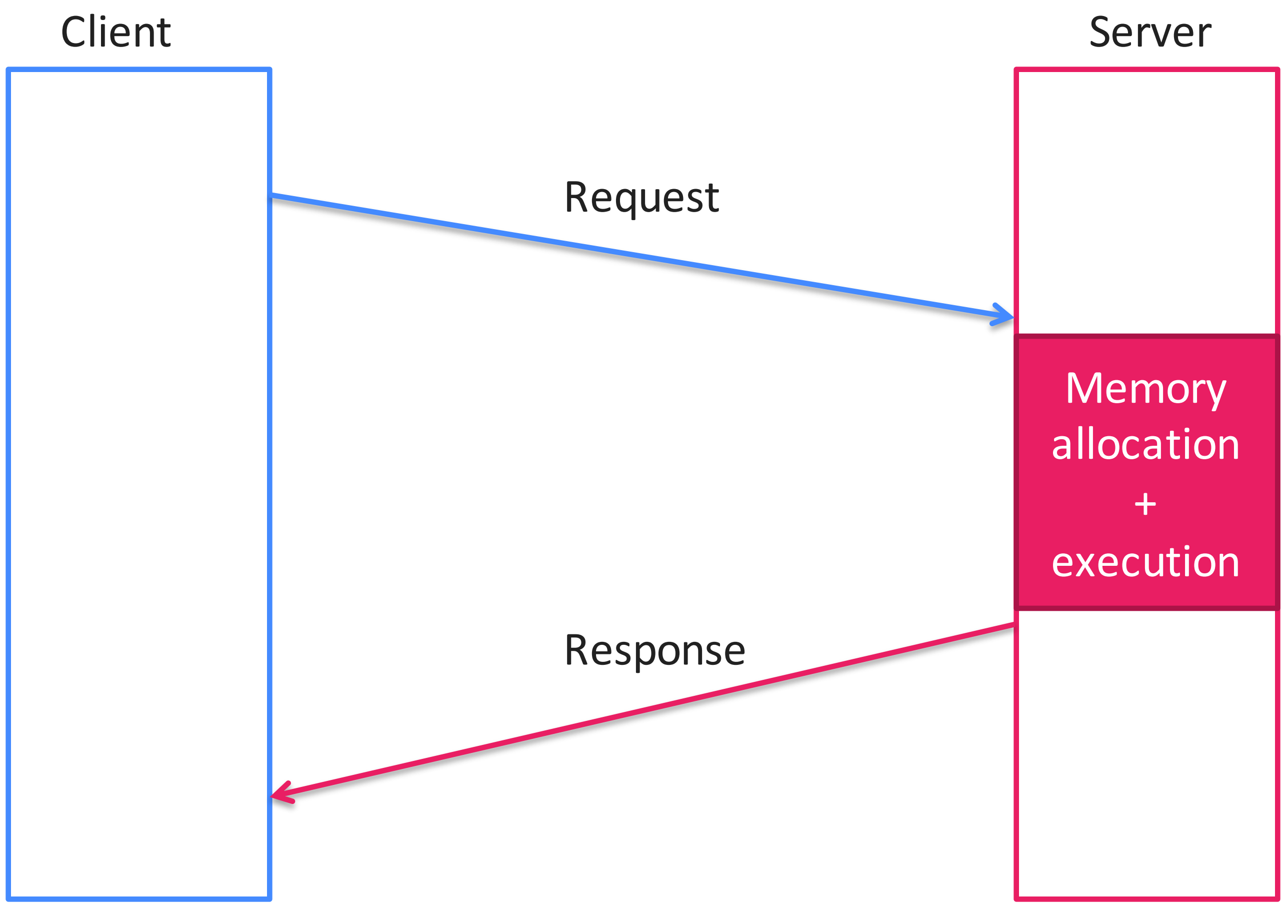
PHP Sessions

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 text-based data
  - Sent with all requests depending on current host URL
- Example usages:
  - Items in a shopping cart
  - Measure interaction (navigation on a site)
  - Authentication

Cookies are not...

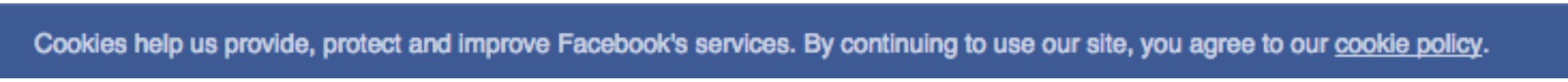
- Necessarily evil:
  - Malware containers
  - Viruses
  - Spam

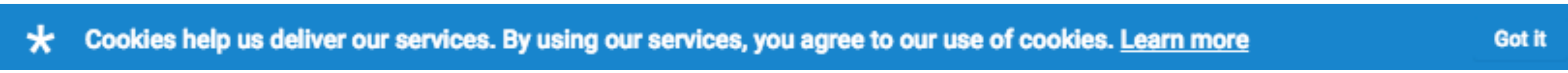
- A place to store large data
  - only small, serializable chunks
  - use the local storage API instead for larger data

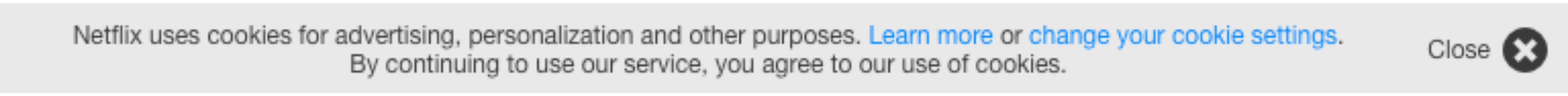
- Dependent on server-side scripting
  - Also available with JavaScript

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 help us provide, protect and improve Facebook's services. By continuing to use our site, you agree to our cookie policy.


Cookies help us deliver our services. By using our services, you agree to our use of cookies. Learn more


Netflix uses cookies for advertising, personalization and other purposes. Learn more or change your cookie settings.
By continuing to use our service, you agree to our use of cookies.

Sessions

- Sessions maintain “states” on the **server side**
- Sessions 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**
- 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 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>
```

Session Functions - Overview

- **session_start**
  Start or resume a session
- **session_destroy**
  Destroy all data from a session, including session ID and cookies (only after page refresh!)
- **session_unset**
  Free all session variables, but maintains the session ID
- **session_name**
  Get or set the session name

Taken from: https://secure.php.net/manual/en/book.session.php
last access on 02/11/2015

Example: Resetting Sessions

```
<?php session_start(); ?>
<!DOCTYPE html>
<html><head lang="en">
 <meta charset="UTF-8"><title>Session Reset</title>
</head>
<body>

<?php
echo session_name() . '<br />';
echo session_id() . '<br />';

$_SESSION['answer'] = 'yes';

session_unset();

echo session_id() . '<br />';
echo $_SESSION['answer'].'<br />'; // ?

$_SESSION['answer'] = 'yes';
echo $_SESSION['answer'].'<br />'; // ?
?>
</body></html>
```

Break Out: Parking lot counter

- Imagine you are the gatekeeper at a parking lot.
- The parking lot holds exactly 15 spots.
- You need to keep track of the occupied slots yourself, so you use a web app that has a +1 and -1 button.
- The page shows the current count.
- If the count reaches the maximum number, the +1 button is greyed out (inactive). The same is true, if no cars are at the parking lot.
- Use PHP-Sessions to maintain the current count.

- Take approx. 30 minutes time
- Present your solution to your peers

Round-up Quiz

1. Why are sessions necessary?
2. Can you initialize a session only at the beginning of a script?
3. Are (session-)cookies stored on the server or on the client?
4. What does `session_destroy()` actually do?
5. What is the difference between `session_destroy()` and `session_unset()`?
6. Why do you need to refresh the page to see the effects of `session_destroy()`?

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 11th, 12:00 noon via UniWorX