Multimedia im Netz
Online Multimedia
Winter semester 2016/17

Tutorial 01 – Major Subject
Today’s Agenda

• Syllabus
  – Dates and procedure
  – Assignments, sample solutions, codelabs
  – Slack!

• Requirements

• PHP Basics
  – xAMP Stack / CIP Pool / VirtualBox
  – Syntax

• Passing form data between browser and server
  – Forms
  – GET & POST

• Quiz
Syllabus
Dates

• **For major subject students** *(Master)*
  Medien-/Informatik, Mensch-Computer-Interaktion:

<table>
<thead>
<tr>
<th>Day</th>
<th>Time</th>
<th>Tutor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monday</td>
<td>16 – 18 h</td>
<td>Thomas Weber</td>
</tr>
<tr>
<td>Monday</td>
<td>18 – 20 h</td>
<td>Thomas Weber</td>
</tr>
<tr>
<td>Wednesday</td>
<td>14 – 16 h</td>
<td>Peter Juras</td>
</tr>
<tr>
<td>Wednesday</td>
<td>16 – 18 h</td>
<td>Peter Juras</td>
</tr>
</tbody>
</table>

• **For minor subject students** *(Bachelor)*
  Kunst und Multimedia, Pädagogik, Statistik, Lehramt:

<table>
<thead>
<tr>
<th>Day</th>
<th>Time</th>
<th>Tutor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Friday</td>
<td>14 – 16 h</td>
<td>Tobias Seitz</td>
</tr>
</tbody>
</table>
Tutorials – Why are we doing this?

• Application and immersion of lecture content
• Hands-on activities and discussion
• Opportunity to discuss and ask questions
• Preparation of the upcoming assignment
• Discussion of the solutions to exercises
• Preparation for work / job.
Procedure – Part 1

• Slides and assignment online prior to tutorial
• Due dates for assignments: one week. **Wednesday to Wednesday**
• News, updates, and important announcements on the official website: [http://www.medien.ifi.lmu.de/lehre/ws1617/mmn/](http://www.medien.ifi.lmu.de/lehre/ws1617/mmn/)
Procedure – Part 2

• Doing the assignments is completely voluntary.
• We recommend you do the assignments.
  – They’re fun and challenging.
  – They go beyond the lecture content.
  – They prepare you to pass the exam.
  – Statistics show that you get better grades if you do the assignments
• Assignments are turned in via UniWorX
  – Make sure to check the due date
  – You can’t hand in an assignment after the deadline.
  – Individual- or group submission
  – Make sure to do the right assignment:
    • Assignment 01 (HF) = Hauptfach, major subject students
    • Assignment 02 (NF) = Nebenfach, minor subject students
Sample Solutions & Material

• We do **not** provide sample solutions.
• Student-provided solutions from last year on GitHub.

• Materials:
  – Code from the tutorials also goes on GitHub:
    [https://github.com/MIMUC-MMN/tutorials-16-17](https://github.com/MIMUC-MMN/tutorials-16-17)
  – You are welcome to share **your own** solutions of this year here:
    [https://github.com/MIMUC-MMN/assignments-16-17](https://github.com/MIMUC-MMN/assignments-16-17)
Codelabs - Let’s do the assignments together

Date/Time: Wednesday 18-20h
Exam

• Final date, time, and location will be announced shortly. Probably: 16.02. or 17.02.2017
• Open Book: You can bring print-outs, books, notes (but no electronic devices)
• They exam includes tasks from both the lecture and tutorial!
## Tutorial Plan (subject to change)

<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
<th>Technology Focus</th>
</tr>
</thead>
<tbody>
<tr>
<td>24.10.</td>
<td><strong>Introduction and Repetition</strong></td>
<td>HTML, PHP</td>
</tr>
<tr>
<td>31.10.</td>
<td>Persistent Content (Sessions)</td>
<td>HTML, PHP</td>
</tr>
<tr>
<td>07.11.</td>
<td>Persistent Content (Databases)</td>
<td>PHP, MySQL</td>
</tr>
<tr>
<td>14.11.</td>
<td>Web Page Interactivity</td>
<td>jQuery</td>
</tr>
<tr>
<td>21.11.</td>
<td>Asynchronous Content</td>
<td>jQuery</td>
</tr>
<tr>
<td>28.11.</td>
<td>Going Vanilla</td>
<td>JavaScript</td>
</tr>
<tr>
<td>05.12.</td>
<td>Server Side Javascript, Static Content</td>
<td>NodeJS, Express</td>
</tr>
<tr>
<td>19.12.</td>
<td>Watermarking User Generated Content</td>
<td>NodeJS, fs, image-watermark</td>
</tr>
<tr>
<td>26.12.</td>
<td></td>
<td>Christmas Break</td>
</tr>
<tr>
<td>02.01.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>09.01.</td>
<td>Repetition, Web-Development: Tipps &amp; Tricks</td>
<td>Tools</td>
</tr>
<tr>
<td>16.01.</td>
<td>MetaData, e-Books</td>
<td>MPEG-7, ePub</td>
</tr>
<tr>
<td>23.01.</td>
<td>Audio Player on Web Sites</td>
<td>Polymer</td>
</tr>
<tr>
<td>30.01.</td>
<td>WebRTC Conferencing</td>
<td>Polymer</td>
</tr>
<tr>
<td>06.02.</td>
<td>Repetition</td>
<td>-</td>
</tr>
</tbody>
</table>
Mini Tests

• You can participate in four tests prior to the exam
• Test sheets are distributed and collected during the tutorials
• You can collect the corrected test sheets in the tutorials in the following week
• Notes:
  – There are no bonus points for passing the mini tests.
  – There won’t be a mock-exam (“Probeklausur”)
• Benefits
  – Make sure you’re up to speed with current material
  – Reduce work-load at the end of the semester
News, Readings, Q&A via Slack!

https://mimuc.slack.com/messages/mmn-ws1617
Requirements
Knowledge & Skills

• HTML5
  – skills are essential for this course
  – if you feel you need to get back in the game, please do some tutorials beforehand
  – Example Tutorials:
    https://www.codecademy.com/learn/web
    http://learn.shayhowe.com/html-css/building-your-first-web-page/

• Command Line / Terminal
  – Linux / macOS: bash commands
  – Windows: PowerShell / GitBash

• Git
  https://rogerdudler.github.io/git-guide/
Tools, tools, tools

https://mimuc.slack.com/files/tobi.seitz/F2PCNRW4T/Required_Toolkit

Be prepared!

Over the course of the semester you will need to install a couple of tools. We're trying to keep it simple to keep up with the tutorials and to complete the tasks. I've ordered them according to the order you might need them in:

1. Git. If you don't know git, [https://rogerdudler.github.io/git-guide/](https://rogerdudler.github.io/git-guide/)
2. NodeJS + npm. [https://nodejs.org/en/](https://nodejs.org/en/) - If you have npm running, install these (you can do that from the Git-Bash):
   a. Bower  `npm install -g bower`
   b. Express Generator  `npm install -g express`
   c. Polymer-CLI  `npm install -g polymer-cli`
   d. gulp  `npm install -g gulp`
   e. Browser-Sync  `npm install -g browser-sync`
3. Text Editor / Web IDE - choose one! -
PHP Basics
PHP: Hypertext Preprocessor

• Server-side scripting language dating from 1995
• Current stable versions 5.6.27 and 7.0.12
• Official website: http://php.net (logo source)

• To get you started: tutorials (just a few examples)
  – http://www.php-einfach.de/php-tutorial/php-tutorial.php (German)
  – http://www.w3schools.com/php/
Option 1: PHP at the CIP-Pool

- PHP usage is restricted: [http://www.rz.ifi.lmu.de/Merkblaetter/homepage.html](http://www.rz.ifi.lmu.de/Merkblaetter/homepage.html)
- To facilitate correction of your assignments, they need to work in the CIP pool:
  - PHP version 7.0.8
  - Put into directory `~/public_html/php`
  - Usage under your personal webspace (replace “login” with your CIP-account name):
  - You can only put PHP files in this directory. If you use images, you have to put them in `public_html` and other subdirectories.
Option 1: CIP Pool - File Location

PHP files go here!
Option 1: CIP Pool - Remote Access

- Server is only accessible in the WAN / CIP Pools
  - Via an ssh-tunnel
    http://www.rz.informatik.uni-muenchen.de/FAQ/Aussenzugriff.faq.html
  - VPN: https://www.lrz.de/services/netz/mobil/vpn/

- Nicely working solution: Remote Desktop Connection
  - Instruction here http://www.rz.ifi.lmu.de/Dienste/rdp.html/
  - Does not work with “Starter” Versions of Windows.
  - Mac App:
Option 2: B.Y.O.D.

- You can use your own machine and install a web server there (Apache)

- XAMPP: Convenient bundle

- e.g. LAMP: Linux, Apache, MySQL, PHP,

- Get it here for Windows, Mac, Linux: https://www.apachefriends.org/download.html

- See if it works: http://localhost
Option 2: Apache doesn’t start?

• Make sure to:
  – Check the port that is configured in httpd.conf / apache2.conf
  – Apache usually listens on port 80
  – Quit Skype (it sometimes listens on port 80/443)
  – On Unix-based systems this command shows you which ports are already taken:
    netstat -ntlp | grep LISTEN

• If you use JetBrains phpStorm, it has a built-in Webserver. You only need to tell it where the php binaries are.
Option 3: Virtual Box Image

• Download Virtual Box (also installed at CIP pool)
  https://www.virtualbox.org/wiki/Downloads

• Download Appliance Image
  http://www.medien.ifi.lmu.de/team/tobias.seitz/files/lehre/mmn/mmn-machine.ova

• Login: mmn       PW: virtualizeit!
• MySQL-user: root  PW: shm00t
Option 3: Virtual Box Image - Import
Option 3: Virtual Box Image - Troubleshooting
Option 3: Virtual Box Image - Root Folder

Open browser and go to http://localhost, an index will appear.
OK, let’s start...
Hello World!

Create the file `test.php` or use the one provided on GitHub:

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

On a CIP-pool machine:

1. Put it into `~\public_html\php`
2. Open a web browser and go to `http://php.cip.ifi.lmu.de/~CIP_KENNUNG/php/test.php`
3. It should say “My first PHP script!”
4. Collaborate with your neighbor if there are any problems.
The output...

My first PHP script!
Embedding PHP into HTML

```html
<!DOCTYPE html>
<html lang="de">
<head>
    <meta charset="UTF-8"/>
    <title>PHP embedded into HTML</title>
</head>
<body>
<h2>
    <?php echo "My Heading"; ?>
</h2>
<?php echo "<p>My paragraph</p>"; ?>
</body>
</html>
```
Syntax

- PHP can be **embedded** into HTML Documents.

```php
<?php ...

- **Variables** are prefixed with a $-sign:

  ```php
  $someVar = 5;
  ```

- **Printing text:**

  ```php
  echo "Sometext";
  echo "Even <b>HTML</b> can be printed";
  ```

- **Concatenation** is done with a dot!

  ```php
  echo "Variable content: " . $someVar;
  ```

- **Comments**

  ```php
  // This is a comment
  /* This is a comment
  that spans multiple lines! */
  # I can’t get enough of those comments!
  ```
Variables Inside Double-Quoted Strings

```php
<?php

$currentTime = date("d.m.Y, H:i:s", time());

echo "It's $currentTime";

?>
```

Types and Operators

- PHP is weakly / dynamically typed
- Data types: Boolean, Integer, Float, String, Array
- Arithmetic operators:
  +  -  *  /  %
- Bit-operators:
  &  |  ^  ~  <<  >>
- Comparison:
  ==  ===  !=  <>  <  >
- Increment and decrement operators:
  ++$a  $a++  --$a  $a--
- Logic operators:
  &&  ||  !  XOR
What output does this generate? (1)

```php
<?php

// 1:
echo 1 + "10 little pigs";

// 2:
$test = 2 . "10 little pigs";
echo $test;

// 3:
echo 3 , "10 little pigs";

?>
```
Control Statements

• If-else:
  ```
  if($a > $b){
      echo "a is greater than b";
  } else {
      echo "a is not greater than b";
  }
  ```

• Ternary operator (syntactic sugar)
  ```
  echo $a > $b ? "a greater than b" : "a less than b";
  ```
What output does this generate? (2)

```php
<?php
$intZero = 0;
$stringZero = "0";

if($intZero == $stringZero)
    echo "== Equal";
else    echo "== unequal";

if($intZero === $stringZero)
    echo "=== identical!";
else    echo "=== unidentical!";
?>
```
Arrays

• Arrays in PHP are actually “ordered maps”
  – Association between key and value
  – So an array is more or less: array, list, hash-table, dictionary, collection, stack, queue, etc. altogether

• Simplification: there are two types of arrays:
  – index-based (regular)
    ```php
    $indexedArray = array(1, 2, 3, 4);
    ```
  – associative (dictionary)
    ```php
    $associativeArray = array("apples"=>"1","bananas"=>4);
    ```

• The types of keys and values can be mixed! (integers, strings, arrays, objects etc.)
While Loops

- Example
  ```php
  $isHomerHungry = true;
  while($isHomerHungry){
      echo 'Homer is still hungry.&nbsp;';
      $isHomerHungry = (rand(0,10) != 10);
  }
  echo "<p>Homer is not hungry anymore</p>";
  ```

- Make sure to find a reliable break condition.
For / Foreach Loops

- `for($donut=1;$donut<=10;$donut++){`  
  `echo "Homer is eating donut $donut";`  
  `}`

- `Foreach:`
  `$donuts = array(  
    "sprinkled",  
    "maple",  
    "glazed");  
  
  foreach($donuts as $donut){`  
  `echo "Homer likes $donut donuts. ";`  
  `}`

- `break`: terminates the execution of the loop.
- `continue`: current loops is interrupted and the loop continues with the next iteration.
Break Out

• Write a small script that...
  – takes an array of arbitrary length
  – prints the doubled value

• If you have time:
  – consider what to do if the array contains strings.
  – Create <li> elements for all items
  – the values are stored in an associative array. How do you access them?

• Take 10 minutes time.
Functions

• Void function:
  ```php
  function someFunction($parameter1, $parameter2){
      // do something
  }
  ```

• With a return value:
  ```php
  function square($number){
      return $number * $number;
  }
  ```
  ```php
  echo square(4);
  ```
Interactive Webpages with PHP
PHP + Forms

• PHP can handle user input, but only after it was sent to the server, where the script is executed.
• Typical user input comes from HTML <form> elements
• There are many different input elements (see next slide)
<input type="..." />

**radio**
- Red
- Green
- Blue

**checkbox**
- Cream
- Milk
- Sugar

**text**
Your text: my text

**button**
Hardest button to button.

**file**
Choose File
No file chosen

**password**
Password:
Example Form: Favorite Color

<!DOCTYPE html>
<html>
<head lang="en">
  <meta charset="UTF-8">
  <title>Favorite Color!</title>
</head>
<body>
<p>Please pick your favorite color:</p>
<form>
  <label><input type="radio" name="color"/> Red</label>
  <label><input type="radio" name="color"/> Green</label>
  <label><input type="radio" name="color"/> Blue</label>
</form>
</body>
</html>
Passing Data between Browser and Server

• The example from the previous slide, allows the user to make a selection, that is, to **enter data**

• How do we pass it from the user’s browser to the server, where we can evaluate the data?
  – action="..." attribute tells where the data should go
  – method="... " attribute tells how it should be “wrapped”

• An `<input type="submit" />` sends the form
Extending the Form: Action, Method, Submit

```html
<form action="formExample.php" method="post">
  <label>
    <input type="radio" name="color" />
    Red
  </label>
  <label>
    <input type="radio" name="color" />
    Green
  </label>
  <label>
    <input type="radio" name="color" />
    Blue
  </label>
  <input type="submit" name="submit" value="Save" />
</form>
```
Extending the Form: Values

```html
<label>
  <input type="radio" name="color" value="red" />
  Red
</label>

<label>
  <input type="radio" name="color" value="green" />
  Green
</label>

<label>
  <input type="radio" name="color" value="blue" />
  Blue
</label>

<input type="submit" name="submit" value="Save" />
```
Extending the Form: Control output

```php
<?php
if(isset($_POST['color'])) {
    echo "<p>Your favorite color is " . $_POST['color'] . "</p>";
} else {
    ?><p>Please pick your favorite color:</p>
<form action="formExample_finish.php" method="post">
    <label>
        <input type="radio" name="color" value="red" />
        Red
    </label>
    <label>
        <input type="radio" name="color" value="green" />
        Green
    </label>
    <label>
        <input type="radio" name="color" value="blue" />
        Blue
    </label>
    <input type="submit" name="submit" value="Save">
</form><?php
} ?>
```
GET & POST

• GET
  – Query string is sent in the URL of the request:
    http://localhost/test.php?lecture=onlineMultimedia
  – Parameters are visible to the user!
  – Superglobal variable in PHP: $_GET (Associative Array!)

• POST
  – Query string is sent in the HTTP message body of the request
  – Superglobal variable in PHP: $_POST (Associative Array!)
Comparison

<table>
<thead>
<tr>
<th>GET Requests</th>
<th>POST Requests</th>
</tr>
</thead>
<tbody>
<tr>
<td>can be <strong>cached</strong></td>
<td>are <strong>never cached</strong></td>
</tr>
<tr>
<td>stay in the <strong>browser history</strong></td>
<td>do <strong>not</strong> show up in the browser history</td>
</tr>
<tr>
<td>can be <strong>bookmarked</strong></td>
<td><strong>cannot</strong> be bookmarked</td>
</tr>
<tr>
<td>have a <strong>fixed length</strong></td>
<td>do not have length restrictions</td>
</tr>
<tr>
<td>should be used to <strong>retrieve</strong> data</td>
<td>should be used to <strong>modify</strong> data</td>
</tr>
<tr>
<td>should <strong>not</strong> be used with sensitive data</td>
<td>are a little safer for sensitive data</td>
</tr>
</tbody>
</table>

[http://www.w3schools.com/tags/ref_httpmethods.asp](http://www.w3schools.com/tags/ref_httpmethods.asp)
Round-up Quiz

1. Is a PHP script evaluated in the browser or somewhere else?
2. What does the acronym LAMP stand for?
3. How do you concatenate strings in PHP?
4. What’s the difference between the == and === operator?
5. What’s going on here:
   ```php
   $grades = array('johnson'=>1.0);
   $grades['smith'] = 3.0;
   ```
6. Is GET or POST more suitable for transmitting passwords? Why?
7. Is the correct syntax `count($array)` or `$array->count()`?
Thanks!
What are your questions?
Appendix
Useful String Functions

• `strlen`: returns the length of a string
• `strstr`: finds the first occurrence of a substring
• `substr`: returns a substring
• `htmlspecialchars`: converts special characters to HTML codes
• `strip_tags`: removes all PHP and HTML tags from a string
• `explode`: splits a string and returns an array with the chunks
• `implode`: takes an array and concatenates the fields to a string
• `str_replace`: replaces all matches with a replacement string
Useful Array Functions

- **count**: returns the number of elements in the array
- **array_search**: searches an array and returns found index
- **in_array**: determines if a value exists in the array
- **shuffle**: shuffles an array
Link Collection

• https://secure.php.net/docs.php
• http://www.w3schools.com/php/php_intro.asp
• https://www.codecademy.com/courses/web-beginner-en-StaFQ
• Make atom even greater:
<?php

function square(&$number){
    $number = $number * $number;
}

$myNumber = 2;
echo "<p>My number is $myNumber</p>"; // 2

square($myNumber);

echo "<p>Now, my number is $myNumber</p>"; // 4
?>

https://secure.php.net/manual/de/language.references.pass.php