

# **Multimedia im Netz**

## Wintersemester 2012/13

### Übung 01

# Organisatorisches

# Inhalte der Übungen

- Vertiefung der Vorlesungsinhalte
- Themen möglichst parallel zur Vorlesung
- Gelegenheit für Fragen
- Vorbereitung auf aktuelles Übungsblatt
- Besprechung von Lösungen der Blätter

# Ablauf der Übungen (I)

- Folien und neues Übungsblatt werden möglichst vor der Übung auf der Webseite veröffentlicht
- Bearbeitungszeitraum für ein Blatt jeweils eine Woche, manchmal zwei Wochen (genauere Informationen sind auf dem jeweiligen Übungsblatt)
- Aktuelle Meldungen immer auf der Website:  
<http://www.medien.ifi.lmu.de/lehre/ws1213/mmn/>

# Ablauf der Übungen (II)

- Die Abgabe der Übungsblätter ist **freiwillig!**
- Die Bearbeitung der Übungsblätter wird jedoch dringend empfohlen:
  - Vertiefung der Vorlesungsinhalte
  - Vorbereitung für die Klausur
- Die Abgabe erfolgt über UniWorX
  - Abgabefrist steht immer auf dem jeweiligen Übungsblatt
  - Nachträgliche Abgabe ist nicht möglich
  - Einzelabgabe

# Klausur

- Termin: voraussichtlich **20.02.2013** von **18–20 Uhr**
- Ort: Theresienstraße, Hörsaal C123
- Inhalte aus **Vorlesung und Übung** werden in der Klausur geprüft

# Fragen?

- Organisatorische Fragen, Feedback und Fragen zur Übung:
  - Alina Hang: [alina.hang@ifi.lmu.de](mailto:alina.hang@ifi.lmu.de)
  - Wichtige Informationen werden auf der Webseite angekündigt
- Fragen zur Übung, Korrekturen, etc.:
  - Die jeweiligen Tutoren helfen gerne weiter

# Einführung in PHP



# PHP - Allgemeines

- Serverseitige Skriptsprache seit 1995
- Aktuell in Version 5.4.8
- Offizielle Webseite: <http://php.net>
- Tutorials (nur kleine Auswahl)
  - <http://tut.php-quake.net/de/index.html>
  - <http://www.php-einfach.de/php-tutorial/php-tutorial.php>

# PHP im CIP-Pool

- Version 5.3.3
- Für die Korrektur müssen Lösungs-Abgaben in Form von PHP-Skripten im CIP-Pool laufen:
  - Ablegen unter public\_html/php
  - Aufrufen der Skripte unter <http://php.cip.ifi.lmu.de/~login/php/skript.php>
  - Nur innerhalb des LRZ-Netzes verfügbar

# Im CIP-Pool: Ein Test

- Mit Texteditor: Dokument **test.php** erstellen:

```
<?php
    echo "Hello World";
?>
```

- Unter public\_html/php ablegen
- <http://php.cip.ifi.lmu.de/~login/php/test.php> aufrufen
- Wird Hello World ausgegeben?

# Am eigenen Rechner

- Installation eines Webservers (Apache)
- XAMPP: Paket inkl. Apache, PHP, MySQL



- Erhältlich für Windows, Mac, Linux unter <http://www.apachefriends.org/de/xampp.html>
- Test: nach Apache-Start im Browser aufrufen: <http://localhost>

# Syntax

- PHP kann in HTML Dokumente **eingebettet** werden

```
<?php ... ?>
```

- **Variablen** werden mit \$ deklariert

```
$var = 5;
```

- **Textausgabe**

```
echo "Text der ausgegeben werden soll";
```

```
echo "Auch <b>HTML</b> kann so ausgegeben werden";
```

- **Konkatenation**

```
echo "Inhalt der Variablen: " . $var;
```

- **Kommentare**

```
// Das ist ein Kommentar
```

```
/* Das ist auch ein Kommentar  
mit mehreren Zeilen sogar! */
```

# Ein Beispiel

```
<html>
  <head>
    <title>PHP Beispiel</title>
  </head>
  <body>
    Dies ist die erste Zeile. <br/>

    <?php
      echo "Und dies ist die zweite Zeile.<br/>";
      $datum_uhrzeit=date("d.m.Y, H:i:s", time());
      echo "Heute ist der " . $datum_uhrzeit . " Uhr";
    ?>

    <br/>

  </body>
</html>
```

# Typen und Operatoren

- Schwache, Dynamische Typisierung
- Datentypen: Boolean, Integer, Float, String, Array
- Arithmetische Operatoren: +    -    \*    /    %
- Bit-Operatoren: &    |    ^    ~    <<    >>
- Vergleichs-Operatoren: ==    ===    !=    <>    <    >
- Inkrement-/Dekrementoperatoren: ++\$a    \$a++    --\$a  
\$a--
- Logische Operatoren: &&    ||    !    XOR

# Kontrollstrukturen (I)

- **if-else**

```
if ($a > $b) {  
    echo "a ist größer als b";  
} else {  
    echo "a ist NICHT größer als b";  
}
```

- **While**

```
$i = 1;  
while ($i <= 10) {  
    echo $i++;  
}
```



# Kontrollstrukturen (II)

- **Foreach**

```
foreach ($arr as $value) {  
    echo $value;  
}
```

- **break**: beendet die Ausführung der aktuellen Schleife
- **continue**: aktueller Schleifendurchlauf wird unterbrochen und es wird mit der Auswertung der nächsten Bedingung weitergemacht

# Funktionen

- **PHP Funktion**

```
function funktionsname(parameter1, parameter2,...) {  
    ...Anweisungen...  
}
```

- **Funktion mit Rückgabewert**

```
function quadrat ($zahl) {  
    return $zahl * $zahl;  
}
```

```
echo quadrat (4); // gibt '16' aus.
```

# Übungsblatt 1

- **Thema: Memory in PHP + Verständnisfragen**
- Bearbeitungszeit: 2 Wochen
- Abgabe: 07.11.2012 23:00 Uhr

Memory Spiel



Du hast das Spiel nach 22 Versuchen gelöst. [Neues Spiel](#)

- **Hinweis:** Übungsstunden am 07.11.2012 dienen als Programmierberatung / Hilfe vor der Abgabe
- Diese Woche für die Einarbeitung in PHP/HTML nutzen!
- Zweite Woche für die Lösung des Blatts.