

Multimedia im Netz

Wintersemester 2012/13

Übung 02

PHP-Server im CIP-Pool

- php.cip.ifi.lmu.de
- Zugriff *nur* im Institutsnetz, also in CIP-Pools
- Zugriff von außerhalb z.B. über ssh-Tunnel
<http://www.rz.informatik.uni-muenchen.de/FAQ/Aussenzugriff.faq.html>
- oder: nx.cip.ifi.lmu.de mit NX-Client von
<http://www.nomachine.com/>
- Damit arbeiten, als ob man am CIP-Pool-Rechner sitzt 😊
<http://www.rz.ifi.lmu.de/Dienste/nx/>

Interaktive Web-Anwendungen

Formulare in HTML

- Normalerweise bekommt der Nutzer Informationen präsentiert
- Durch Formulare kann er Informationen eingeben: Element `<form></form>`
- Zahlreiche Formularelemente (einige werden im folgenden vorgestellt)

Formularelemente

radio

- Rot
- Grün
- Blau

checkbox

- Rot
- Grün
- Blau

textarea

Text eingeben

text

 Farbe eingeben

button

file

password

 Passwort

Formularbeispiel

```
<html>
<body>
<form>
  <p>Lieblingsfarbe angeben:</p>
  <p>
    <input type="radio" name="xy" value="red">Rot<br/>
    <input type="radio" name="xy" value="blue">Blau<br/>
    <input type="radio" name="xy" value="pink">Pink<br/>
  </p>
</form>
</body>
</html>
```

Formulare: Daten weitergeben

- Im Beispiel können Daten eingegeben werden
- Aber wie werden sie weitergegeben?
- action-Tag im <form>-Element gibt an, welches Dokument aufgerufen werden soll
- durch den Button <input type="submit"> werden die Daten abgeschickt

Formulare: Action

```
<html>
<body>
<form action="go.php" method="post">
  <p>Lieblingsfarbe angeben:</p>
  <p>
    <input type="radio" name="xy" value="red">Rot<br/>
    <input type="radio" name="xy" value="blue">Blau<br/>
    <input type="radio" name="xy" value="pink">Pink<br/>
  </p>
</form>
</body>
</html>
```


Formulare: Submit

```
<html>
<body>
<form action="go.php" method="post">
  <p>Lieblingsfarbe angeben:</p>
  <p>
    <input type="radio" name="xy" value="red">Rot<br/>
    <input type="radio" name="xy" value="blue">Blau<br/>
    <input type="radio" name="xy" value="pink">Pink<br/>
  </p>
  <input type="submit" value="Absenden">
</form>
</body>
</html>
```

Formulare: Daten verarbeiten

Code der Datei „go.php“, die die eingegebenen Daten verarbeitet:

```
<html>
<body>

<?php
    $radioSelection = $_POST[ 'xy' ];
    echo "<p>Deine Lieblingsfarbe ist " .
        $radioSelection . " :)" ;
?>

</body>
</html>
```

Speichern von Daten

- Daten können/müssen gespeichert werden
- Client-Seitig: nur wenige Daten ablegbar, z.B. Cookies
- Server-Seitig: riesige Datenmengen ablegbar, z.B. in Datenbanken
- In Sessions werden Variablen server-seitig gespeichert

Sessions

- Warum? Der Inhalt von Variablen geht beim Neuladen der Seite verloren.
- Über Session-ID (gespeichert in Cookie oder URL) werden die Daten auf dem Server gespeichert und von dort geholt.
- Session starten:

```
<?php session_start(); ?>
```

Wichtig: Muss ganz am Anfang des Codes stehen

- Variablen-Inhalte in Sessiondaten speichern

```
$_SESSION[ 'versuche' ] = 10;
```
- Variable kann später wieder ausgelesen werden

```
echo $_SESSION[ 'versuche' ];
```



Sessions: Beispiel (1)

```
<?php
    session_start();
    $_SESSION['number'] = 999;
?>

<html>
  <head><title>Zahl merken</title></head>
  <body>
    <h1>Zahl merken</h1>
    <form action="zahl.php">
      <p>Wir merken uns die Zahl 999 :)</p>
      <input type="submit" value="weiter"/>
    </form>
  </body>
</html>
```



Sessions: Beispiel (2)

```
<?php
    session_start();
?>

<html>
    <head><title>Zahl merken</title></head>
    <body>
        <h1>Zahl merken</h1>
        <?php
            echo "Die gemerkte Zahl ist die " .
                $_SESSION[ 'number' ] . " :)" ;
        ?>
    </body>
</html>
```

Danke! Fragen?