

Datenbanksysteme SS 2007

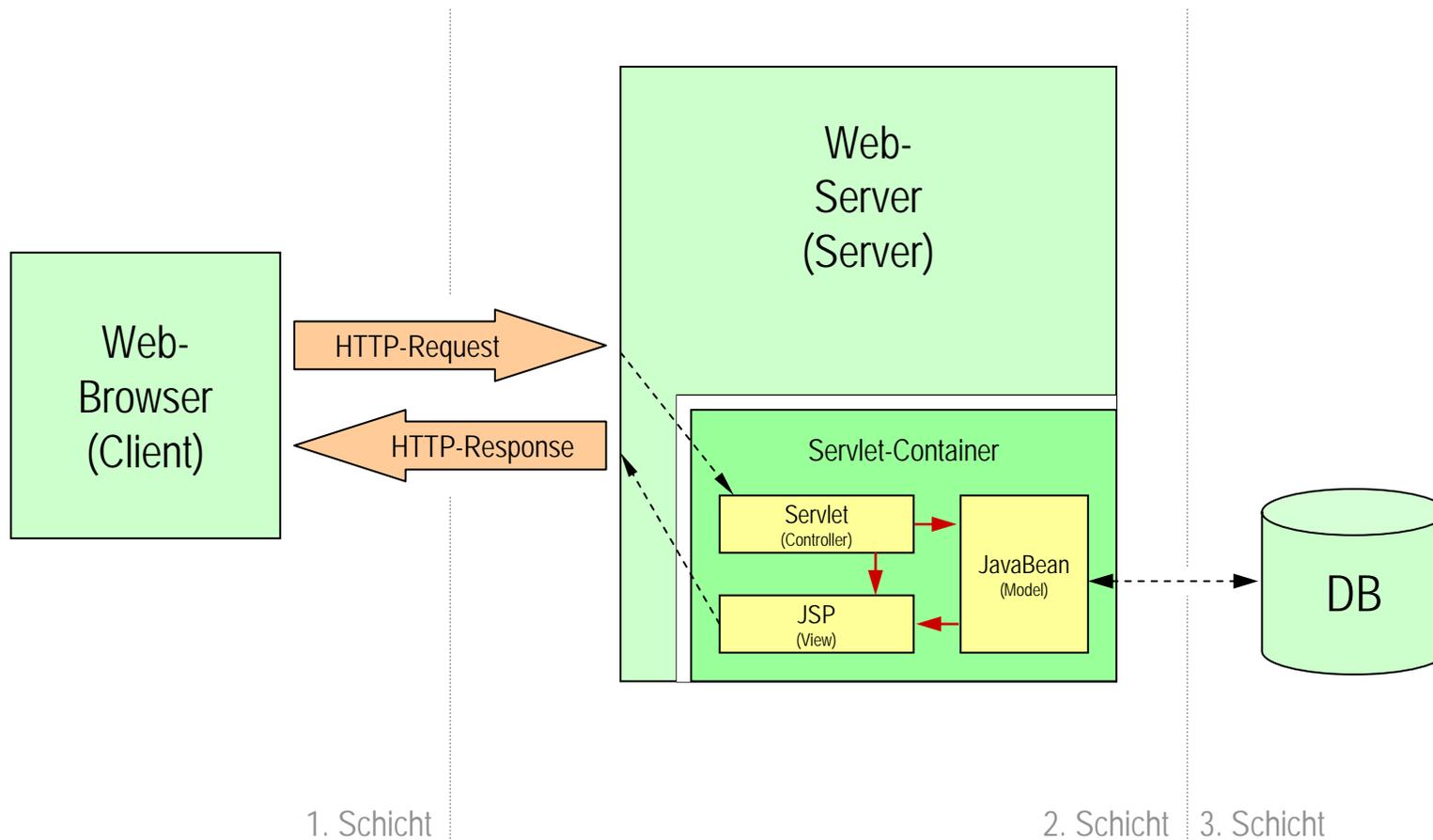
Frank Köster
(Oliver Vornberger)

Institut für Informatik
Universität Osnabrück

Kapitel 9c: Datenbankapplikationen

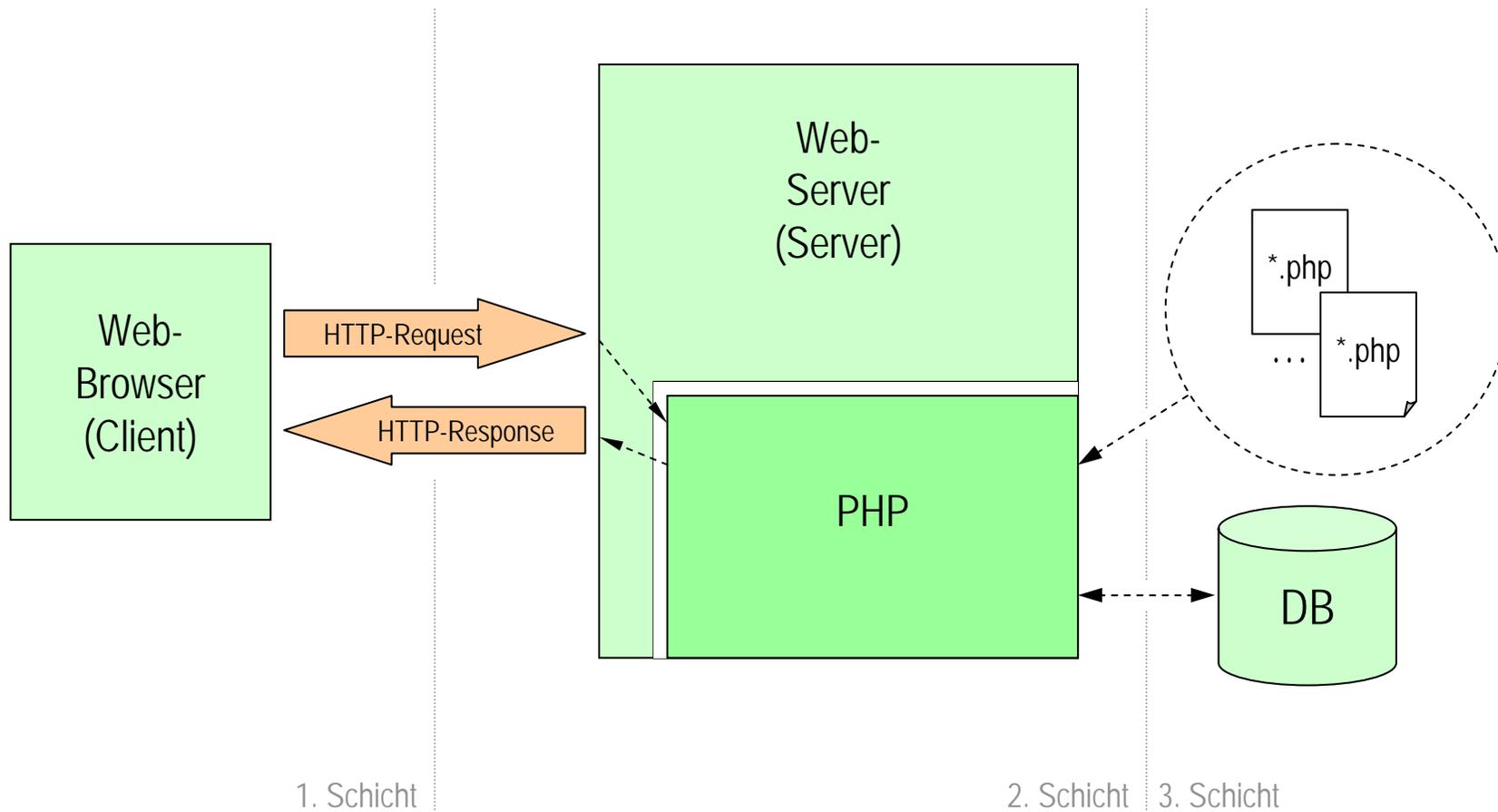
Architektur einer Web-Applikation mit ...

Servlets, JSPs und JavaBeans



Architektur einer Web-Applikation mit ...

PHP

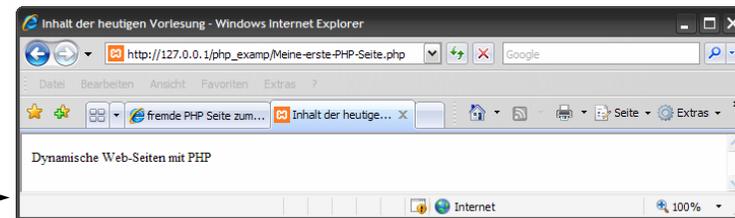


Das grundlegende Prinzip

*.php ⇒ HTML

Inhalt der Datei „Meine-erste-PHP-Seite.php“

```
<html>
<head>
  <title>Inhalt der heutigen Vorlesung</title>
</head>
<body>
  <?
  echo "Dynamische Web-Seiten mit PHP";
  ?>
</body>
</html>
```



Quelltext der vom Browser angezeigten Seite

```
<html>
  <head>
    <title>Inhalt der heutigen Vorlesung</title>
  </head>
  <body>
    Dynamische Web-Seiten mit PHP
  </body>
</html>
```



Programmieren mit PHP

PHP in HTML einbetten

- 4 Möglichkeiten
 - `<? ... ?>`
 - `<?php ... ?>`
 - `<script language="php"> ... </script>`
 - `<% ... %>`

"..." steht für beliebige PHP-Anweisungen.

„;“ markiert das Ende einer PHP-Anweisung / Sonderzeichen

- Es gibt in PHP eine Reihe von Sonderzeichen. Wichtigstes Sonderzeichen ist „\n“, wodurch eine neue Zeile eingeleitet wird.
 - ```
echo "Hallo\n";
echo "Hallo\n";
```
- Beachte unterschied zu
  - ```
echo "Hallo<BR>";  
echo "Hallo<BR>";
```



Programmieren mit PHP

Variablen

- Variablen sind zur Realisierung komplexer Applikationslogiken unerlässlich
– in PHP beginnen alle Variablennamen mit „\$“
 - ```
$text = "Dies ist ein Text
";
echo $text;
echo $text,$text,$text,"
";
```
  - ```
$o = " -OFT-";  
$s = " -SELTEN-";  
echo " ... $o$o$o$o$o$o$o !<br>";  
echo " ... $s !<br><br>";
```
 - ```
$x = 11;
$y = 12;
echo $x,"+", $y,"=", $x+$y;
```
- Der Variablentyp (ganze Integer, Float, String) wird je nach Verwendung von PHP bestimmt. Nutzer muss sich darum i.Allg. nicht kümmern.



# Programmieren mit PHP

---

## ARRAYs

- ARRAYs sind normale Variablen, die ihre Werte u.a. über die Funktion `array(...)` zugewiesen bekommen:

- `$Namen = array("Frauke", "Sabine", "Christine", "Silke", "Petra", "Marion", "...");`

```
echo "Name Nr. 0 ", $Namen[0], "
";
```

```
echo "Name Nr. 3 ", $Namen[3], "

";
```



# Programmieren mit PHP

---

## Vordefinierte Variablen

- Beispielsweise zur Übernahme von Eingaben aus HTML-Formularen – siehe hierzu die Formular-Spezifikation:
  - ```
<FORM ACTION="todo.php" METHOD=POST>  
  <INPUT NAME="paraname">  
  <INPUT TYPE="submit">  
</FORM>
```
- Beim Klick auf den Submit-Button wird die Datei `todo.php` aufgerufen, die bspw. den folgenden Inhalt hat:
 - ```
<?
 echo "Sie haben $paraname eingegeben."
?>
```



# Programmieren mit PHP

---

## Einfache Rechenoperationen

- PHP bietet die folgenden einfachen Rechenoperationen an

- **+** Addition, `$i+$j` (auch `$i++`, `++$i`)
- **-** Subtraktion, `$i-$j` (auch `$i--`, `--$i`)
- **\*** Multiplikation, `$i*$j`
- **/** Division, `$i/$j`
- **%** Reste-Bildung

z.B.: `$i%$j` ➔ `23%17` ergibt `6`  
(da 23 geteilt durch 17 gleich 1 Rest 6 ist).

- **.** Verknüpft Strings  

```
$l = "--l--a--n--g--";
$k = "--kurz--";
echo $l.$k; ➔ ergibt --l--a--n--g----kurz--
```



# Programmieren mit PHP

---

## Schleifen

- Beispiel für eine **while**-Schleife

```
▪ $t = "Dies ist ein Schleife!
\n";
 $i = 0;
 while ($i<10) {
 echo $t;
 $i++;
 }
```

- Es gibt auch

```
▪ do { ... } while ...
▪ for (...) { ... }
```



# Programmieren mit PHP

---

## Fallunterscheidungen (1/2)

- If - else
  - ```
if ($i<0) {  
    echo "$i ist kleiner als Null\n";  
}
```
 - ```
if ($i<0) {
 echo "$i ist kleiner als Null\n";
} else {
 echo "$i ist größer oder gleich Null\n";
}
```

Weiterhin:

|                                              |                                         |
|----------------------------------------------|-----------------------------------------|
| <code>\$i==10</code>                         | ist \$i gleich 10?                      |
| <code>\$i!=10</code>                         | ist \$i ungleich 10?                    |
| <code>\$i&gt;=10</code>                      | ist \$i größer oder gleich 10?          |
| <code>\$i&lt;=10</code>                      | ist \$i kleiner oder gleich 10?         |
| <code>(\$i==10) &amp;&amp; (\$j&gt;0)</code> | ist \$i gleich 10 und \$j größer als 0? |
| <code>(\$i==10)    (\$j==0)</code>           | ist \$i gleich 10 oder \$j gleich 0?    |



# Programmieren mit PHP

---

## Fallunterscheidungen (2/2)

- switch
  - ```
switch ($anzahl) {  
    case "eins":  
        echo "1";  
        break;  
    case "zwei":  
        echo „2“;  
        break;  
    case "drei":  
        echo "3";  
        break;  
    default:  
        echo "... der ganze Rest!";  
}
```

Datei hinzuladen

- `include("dateiname");`



DB-Zugriff (MySQL) mit PHP (1/3)

Verbindung zur DB herstellen

- Zunächst muss Verbindung von PHP zum DB-Server hergestellt werden:

```
$verbindung = @mysql_connect("localhost","uname","password");
```

- Prüfen, ob die Verbindung zur DB aufgebaut werden konnte:

```
if (!$verbindung) {  
    echo "Keine Verbindung zur DB möglich!\n";  
    exit;  
}
```



DB-Zugriff (MySQL) mit PHP (2/3)

Formulierung einer Query

- Konnte die DB-Verbindung aufgebaut werden, können SQL-Anweisungen auf der DB ausgeführt werden.

Zur Übersicht werden SQL-Abfragen an Variablen gebunden:

```
$query = "SELECT Name from Studenten";
```

- Diese Abfrage wird an den Datenbank-Server geschickt. Das Resultat fangen wir in der Variablen **\$erg** auf.

```
$erg = mysql_db_query("dbs07", $query, $verbindung);
```



DB-Zugriff (MySQL) mit PHP (3/3)

Verarbeitung des Resultats der Query

- Datensätze aus `$erg` müssen ausgelesen werden.
Bei `INSERT`, `UPDATE` oder `DELETE` bräuchte nur getestet werden, ob `$erg` „wahr“ oder „falsch“ ist – d.h. ob die Query funktioniert hat oder nicht.
- **SELECT**-Befehl erfordert hier:

- Pro Datensatz wurde die Spalte „Name“ selektiert. Diese lesen wir nun zeilenweise aus, wobei

```
list($Name) = mysql_fetch_row($erg);
```

immer nur eine Zeile liefert.

- Alle Zeilen sind mit Hilfe einer Schleife zugreifbar:

```
while (list($Name) = mysql_fetch_row($erg)) {  
    echo "$Name<BR>\n";  
}
```



Datenbanksysteme SS 2007

Ende von Kapitel 9c:
Datenbankapplikationen