

HTML und PHP

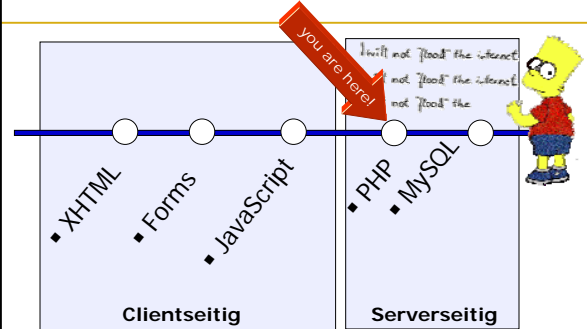
Praktikum Internet SoSe 2006

prInt

Lars Braubach



Blackboard



prInt

Lars Braubach



Gliederung

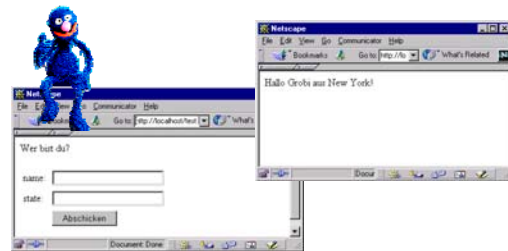
- **Nochmal das CGI-Konzept**
- PHP – Grundlagen
- Sprachelemente
- Tipps

prInt

Lars Braubach



Die "Was passiert dann?"-Maschine

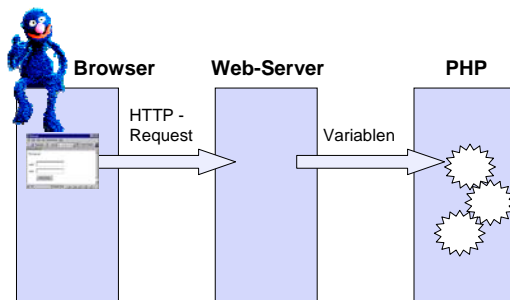


prInt

Lars Braubach



Die "Was passiert dann?"-Maschine



prInt

Lars Braubach



test.php

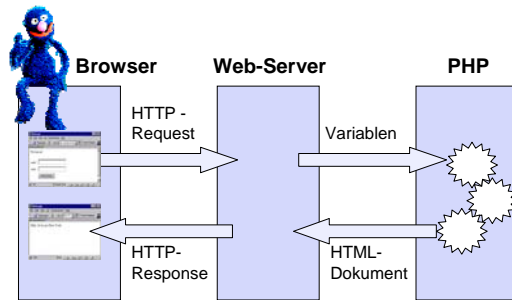
```
<? if !isset($_POST['name']){ ?>
<form action="test.php" method="post">
  name: <input type="text" name="name"/><br/>
  state: <input type="text" name="state"/><br/>
  <input type="submit" value="Abschicken"/>
</form>
<? else:
    echo ("Hallo ".$_POST['name']." aus"
        .$_POST['state']."!");
  }
?>
```

prInt

Lars Braubach



Die "Was passiert dann?"-Maschine



prInt

Lars Braubach



Gliederung

- Nochmal das CGI-Konzept
- **PHP – Grundlagen**
- Sprachelemente
- Tipps

prInt

Lars Braubach



Was ist PHP ?

- "PHP: Hypertext Preprocessor"
- Geschichte:
 - 1995 von Rasmus Lerdorf entwickelt – Personal Home Page Tools -> PHP/FI
 - 1998 Neuentwicklung PHP3 von Andi Gutmans und Zeev Suraski -> Zend Technologies
- Serverseitige Skriptsprache (aktuell: Version 5.1.2)
 - Besitzt inzwischen sogar objektorientierte Ansätze
- CGI-Programm oder Webserver-Modul
- Viele praktische Features
 - Datenbank-Schnittstellen
 - Sessions
 - Cookies
 - XML ...

prInt

Lars Braubach



Mischen mit XHTML

- `<?php print("<?xml ... >"); ?>`
- `<html xmlns=" [...]">`
- `<body>`
- `<?php`
- `print("Hello ");`
- `print("world!
");`
- `?>`
- Hello XHTML.
- `</body>`
- `</html>`

ACHTUNG:

Da die Zeichenfolge "<?" in den PHP-Modus wechselt, muss der XML-Prolog über eine print-Anweisung in das XHTML-Dokument geschrieben werden!

prInt

Lars Braubach



Gliederung

- Nochmal das CGI-Konzept
- PHP – Grundlagen
- **Sprachelemente**
- Tipps

prInt

Lars Braubach



Variablen

- `$varname ;`
 - integer 1
 - double 1.34545
 - string "Hello World!"
 - array \$array["prInt"]="toll";
 - object
- Automatisches Typcasting

```
$a=1;
$b="2";
$c=$a+$b; //c==3
$d=$a.$b; //d=="12"
$e=(int)"2 \n";
```

prInt

Lars Braubach



Strings (1)

- Ausgabe:
`echo($string)` oder `print($string)`
- Einfache Anführungsstriche (') -> keine Auswertung von Escape Sequenzen und Variablen
 - Beispielausdrücke in Strings:
`$a=42;`
`echo("Die Antwort ist $a\n");`
`echo('Die Antwort ist $a\n');`
`echo("\n\n" ist eine\nnewline!);`
ergibt:
Die Antwort ist 42
Die Antwort ist \$a\n\n ist eine
newline!

prInt

Lars Braubach



Strings (2)

- Die heredoc Notation für Strings
- Syntax:
 - Start des Strings <<<[bezeichner]
 - Ende des Strings [bezeichner] GANZ am Anfang einer neuen Zeile
- Beispiel:
 - `$str = <<<EOD`
Beispiel eines Strings über mehrere Script-Zeilen durch Gebrauch der heredoc-Syntax.
`EOD;`
`echo($str);`
- Anwendungsfall: Auskommentieren von Codeblöcken

prInt

Lars Braubach



Strings (3)

- `sprintf()` formatiert Strings:
`$isodate = sprintf("%04d-%02d-%02d", $year,$month, $day); //z.B.: 1900-01-02`
- Ist String \$str eine Zahl?
`$str=" 5\n";`
`$str=trim($str);`
`if($str==($str+0)){print("Zahl!");}`
`//intval($str);`
- Wie lang ist ein String?
`$str="Dieser String";`
`print($str." ist so lang: ".strlen($str);`

prInt

Lars Braubach



Arrays

- Mischung aus nummerierter und assoziativer Form
- Syntax: `array([Schlüssel =>] Wert , ...)`
 - Ein Schlüssel kann entweder ein integer oder ein String sein
 - Der Wert kann einen beliebigen Typ besitzen
 - Zum Löschen von Elementen `unset()` benutzen
- Erzeugung mit mehreren Elementen:
 - `$array1 = array(2, 3, 5, 7, 11);`
 - `$array2 = array("one"=>1, "two"=>2);`
 - `$array3 = array(5=>"five", "six", "seven");`
- Zuweisung von Elementen:
 - `$array1[0] = "13";`
 - `$array2["one"] = 1;`


prInt

Lars Braubach



Operatoren

Standard

+ - * /  =
`$a=5+6;`

Erweiterte

+= -= *= /= .= ++ --
`$a+=5; // $a=$a+5;`

Boolsche

== === != !== < > <> <= >= !
`if($a==5){exit();}`

prInt

Lars Braubach



Kontrollstrukturen

```
if(Bedingung){Anweisung;}  
elseif(Bedingung){Anweisung;}  
else{Anweisung;}
```

```
while(Bedingung){Anweisung;}
```

```
do{Anweisung;} while(Bedingung)
```

```
for(Exp1; Bedingung; Exp2){Anweisung;}
```

```
foreach($array as $value) Anweisung  
foreach($array as $key => $value) Anweisung
```

prInt

Lars Braubach



Funktionen, Namensraum (1)

```
$a=1;
$b=my_prod(6,7); // $b==0
$b=my_prod2(6,7); // $b==42

function my_prod($c,$d){
    return $c*$d*$a;
}
function my_prod2($c,$d){
    global $a; // $a verfügbar machen
    return $c*$d*$a;
}
```

print

Lars Braubach



Funktionen, Namensraum (2)

```
$a=1;
$b=my_prod3(6,7); // $b==42

function my_prod3($c,$d){
    return $c*$d*$GLOBALS["a"];
}
```



print

Lars Braubach



Funktions-Bibliotheken

```
Beispiel: Aufgabe5.php
<?
    require("common.php");
    commonHeader("Aufgabe 5");
    h1("Aufgabe 5: PHP");
?>
Heute sollt Ihr....
<? commonFooter(); ?>
```

print

Lars Braubach



Gliederung

- Nochmal das CGI-Konzept
- PHP – Grundlagen
- Sprachelemente
- **Tipps**

print

Lars Braubach



CGI

form.html

```
<form action="action.php" method="post">
    Wer bist du?
    <input type="text" name="eingabe"/>
    <input type="submit"/>
</form>
```

action.php

```
<h2>Übertragene Daten:</h2>
<?
    echo("Hallo " . $_POST["eingabe"] . "!<br/>\n");
    echo("IP: " . getenv("REMOTE_ADDR") . "<br/>\n");
?>
```

print

Lars Braubach



test.php

```
<? if !isset($_POST['name']){ ?>

<form action="test.php" method="post">
    name: <input type="text" name="name"/><br/>
    state: <input type="text" name="state"/><br/>
    <input type="submit" value="Abschicken"/>
</form>

<? else:
    echo ("Hallo " . $_POST['name'] . " aus"
        . $_POST['state'] . "!");
}
?>
```

print

Lars Braubach



Multiple Select

- Auswerten von Multiple Select-Feldern:

```
<select multiple name="test[]">
  <option value="eins">eins</option>
  <option value="zwei">zwei</option>
  <option value="drei">drei</option>
</select>

foreach($test as $key => $val)
{
  print "$key => $val<br/>";
}
```

print

Lars Braubach



Eure Aufgabe

- Kopiert die Lösung vom letzten Mal in ein neues Verzeichnis
- Einheitliches Layout aller Seiten mit `require()`;
- Überprüfung der FORM-Daten mit PHP
- Preview der FORM-Daten

- Bonusaufgaben: Preview mit Änderungsmöglichkeit, Besuchercounter, Dateiupload

- Viel Spaß :-)



print

Lars Braubach



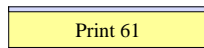
Grafische Ausgaben

- Voraussetzung: Der Server muss die GD-Library bereitstellen

- Erzeugung:
 - `$image = ImageCreateFromJPEG(imagename);`
 - `$image = ImageCreate(xsize, ysize);`

- Bearbeitung:
 - `ImageRectangle(image, x1, y1, x2, y2, color);`
 - `ImageText(image, font, x, y, text, color);`

- Ausgabe:
 - `ImagePNG($image);`
 - `ImageDestroy($image);`



print

Lars Braubach

