

Control Statements

lukas.feiler@lukasfeiler.com

<http://teaching.lukasfeiler.com>

Control Statements

- if
- switch
- for
- while

```
if (boolscher Ausdruck) {  
    //if true  
}  
else {  
    //if false  
}
```

Boolsche Ausdrücke

True, False

Example	Name	Result
<code>\$a == \$b</code>	Equal	TRUE if \$a is equal to \$b.
<code>\$a === \$b</code>	Identical	TRUE if \$a is equal to \$b, and they are of the same type.
<code>\$a != \$b</code>	Not equal	TRUE if \$a is not equal to \$b.
<code>\$a !== \$b</code>	Not identical	TRUE if \$a is not equal to \$b, or they are not of the same type.
<code>\$a < \$b</code>	Less than	TRUE if \$a is strictly less than \$b.
<code>\$a > \$b</code>	Greater than	TRUE if \$a is strictly greater than \$b.
<code>\$a <= \$b</code>	Less than or equal to	TRUE if \$a is less than or equal to \$b.
<code>\$a >= \$b</code>	Greater than or equal to	TRUE if \$a is greater than or equal to \$b.

Beispiel: if.php

Boolsche Ausdrücke mittels Comparision Operators

Example	Name	Result
! \$a	Not	TRUE if \$a is not TRUE.
\$a && \$b	And	TRUE if both \$a and \$b are TRUE.
\$a \$b	Or	TRUE if either \$a or \$b is TRUE.

Beispiel: logicalOperators.php

Boolsche Ausdrücke mit logischen Operatoren verbinden

Spezifikation:

Ein File bestehend aus PHP und HTML; eine HTML-Form mit 2 Eingabefeldern für Username und Passwort soll per POST submitet werden können.

Wenn der Username 'user1' und das Passwort 'password1' ODER der Username 'user2' und das Passwort 'password2' lautet wird eine geheime Nachricht angezeigt.

Bugs finden und beheben:

ifBug.php


```
switch ($i) {  
    case 0:  
        //execute if $i == 0  
    case 1:  
        //execute if $i == 1  
    default:  
        //execute if $i != 0 && $i != 1  
}
```

z.B. switch.php

```
for (expr1; expr2; expr3) {  
    //for loop body  
}
```

expr1 ...wird nur einmal an Anfang ausgeführt

expr2 ...boolscher Ausdruck; Abbruch wenn false
(continue condition)

expr3 ...wird am Ende jedes Durchlaufs ausgeführt

z.B. for.php

Mit for statements über Arrays mit numerischen Keys
loopen

benötigte Funktion: count
z.B. forLoopOverArray.php

Spezifikation:

Berechnung der Summe aller Zahlen von 0 bis 100 mittels eines for loops.

Zusatzaufgabe für Fortgeschrittene:

Berechnung der Summe aller geraden Zahlen von 0 bis 100 mittels eines for loops.

Bugs finden und beheben:

forBug.php

```
while (boolscher Ausdruck) {  
    //while loop body  
}
```

While loop body wird ausgeführt solange der boolscher Ausdruck true ergibt.

z.B. while.php

Mit while statements über Arrays loopen

Funktioniert im Unterschied zu for loop immer!
z.B. whileLoopOverArray.php

Spezifikation:

Alle per POST an das PHP-File submitete Daten sollen ausgegeben werden. Zu diesem Zweck sollen >2 Textfelder verwendet werden.

Tip: `while(list($key, $value) = each($array))`

Bugs finden und beheben:

whileBug.php

Andere PHP-Files einbinden („includen“)

idR

```
require_once($filename);
```

z.B. require.php

`require($filename);`

Tritt ein Fehler auf wird die Ausführung beendet.

`include($filename);`

Tritt ein Fehler auf wird nur ein Warning ausgegeben.

`require_once($filename);`

wie require; ein file kann jedoch nicht mehrfach eingebunden werden.

z.B. `require_once.php`, `require_multiple.php`

`include_once($filename);`

wie include; ein file kann jedoch nicht mehrfach eingebunden werden

4 Möglichkeiten andere PHP-Files einzubinden

Spezifikation:

PHP-File #1 bindet PHP-File #2 ein. PHP-File #1 gibt Variablen, die in PHP-File #2 definiert werden aus.

Bugs finden und beheben & Sicherheitslöcher stopfen

login.php
accounts.inc