

Kommentare

JavaScript:

```
<script language="javascript">  
// hier auch möglicherweise ein Kommentar  
    mach was = tuwas()  
/*  
    hier könnte Ihre Kommentar stehen  
*/  
    mache Ausgabe in Dokument;  
</script>
```

PHP

```
<?php  
// Hallo ich bin ein Kommentar  
    mach was  
/* Autor: Bernd Bluemel  
    Datum der letzten Aenderung: 17.09.01  
    Zweck: Addition zweier Zahlen  
*/  
# Hier noch eine Möglichkeit zum auskommentieren  
?>
```

Kommentare (Gegenüberstellung)

JavaScript	PHP
// auskommentiert	// hier auch
/* und jetzt über mehrere Zeilen */	/* in PHP das gleiche auch über mehrere Zeilen */
	# Auch # so kann man # in PHP # kommentieren

Kommentare Tips

/* Fehlersuche in PHP

\$foo="Euro"; /* Währung */ ←

\$bar="iksy";

***/**

Kommentar endet bereits hier
Zeile 4 erzeugt Fehlermeldung,
weil /* nicht alleine stehen darf

/* Blockkommentar //

echo "Wir befinden uns jetzt hier";

//-----*/

Einfaches Aktivieren bzw. Deaktivieren
des Blockkommentars:
In der ersten Zeile * gegen / tauschen.

Variablen (Grundsätzliches)

sind programminterne Namen für Daten

-> Verweis auf einen Speicherbereich

können speichern:

1. Zahlen (Integer, Float usw.) (Punkt statt Komma als Dezimaltrenner)
2. Zeichenketten (Strings)
3. Kalenderdaten

Variablen in JavaScript und PHP

- sind (leider?) untypisiert

d.h.

Datentyp kann nicht zugewiesen werden

z.B.

\$zahlenVariable=string

vielmehr erfolgt eine automatische Typumwandlung
kann Fehlerquelle sein... ☹

Deklaration von Variablen

JavaScript	PHP
<code>var foo;</code>	<code>\$foo=12;</code>
<code>var SonstEinName;</code>	<code>\$bar="Mittwoch";</code>

In PHP können keine Variablen deklariert werden
Bei beiden automatische Typzuordnung (z.B. String oder Integer)

Übung:

[deklarationUndZuweisung.phps](#)

Übung:

Lassen Sie sich den Datentyp in PHP wie folgt ausgeben:

```
$foo=12;  
$foo=gettype($foo);  
echo $foo;
```

Regeln bei Vergabe von Variablennamen

JavaScript

`var foo`

beginnt mit Buchstaben

PHP

`$foo="bar";`

nach \$ muss Buchstabe folgen

- nach erstem Buchstaben Buchstaben, Ziffern und wieder Buchstaben möglich,
-> Heutelst12345Mittag
- Variablennamen dürfen ausser _ keine Sonderzeichen und auch keine dt. Umlaute (ä ü ö ß) enthalten
- beliebig lang:
DiesIstEinGanzLangerVariablenNameMitZahlen1234546567677
- Gross- und Kleinschreibung beachten! sumMe != Summe
- Reservierte Worte sind KEINE Variablennamen
z.b. \$mail

Datentypen in JavaScript und PHP

- **Ganzzahlvariablen** (Int) -1; 1000; 5; -1000
- **Reelle oder Fließkommazahlen** (Floats) -1.2, -100.001 Erinnerung: Punkt statt Komma!!
- **Wahrheitswerte**

true oder false

```
$ok = mail("cm@fb6.info", "Training", "Hi, Donnerstag ist Vorlesung", "From: bb@buero.de");  
if (!$ok)  
{  
    echo "Mail wurde nicht versendet";  
}  
else  
{  
    echo "Glück gehabt, Mail wurde versendet!";  
}
```

- **Strings (Zeichenketten)**
"Dies ist ein String. Er kann \ \$auch Zahlen wie 123 enthalten"
Länge von 0 bis 2 Milliarden Zeichen
- **Ressource(PHP), Arrays und Objekte (dazu mehr im 6. Semester)**

Gültigkeitsbereich von Variablen

Variablen sind nur in dem Bereich gültig, in dem sie definiert wurden

```
$tag="Donnerstag";
```

```
function testEins ()
```

```
{
```

```
    echo "Heute ist $tag";
```

```
}
```

```
testEins();
```

funktioniert NICHT



```
function testZwei()
```

```
{
```

```
    global $tag;
```

```
    echo "Heute ist $tag";
```

```
}
```

```
testZwei();
```

funktioniert



Bei Funktionen oder Klassen
darauf achten!

Referenzierung bei Variablen

```
$ersterSummand=13;
```

```
$ergebnis=&$ersterSummand;
```

```
$ersterSummand=25;
```



```
echo $ergebnis;
```

Angezeigt wird:

25

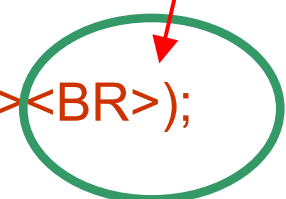
Auswirkungen von Schreibfehlern

```
<script language = "JavaScript">
    eingabe = prompt ("Erste Eingabe", "");
    ersteEingabe = parseInt (eingabe);
    eingabe = prompt ("Zweite Eingabe", "");
    zweiteEingabe = parseInt (eingabe);
    ergebnis = ersteEingabe + zweiteEingabe;
    document.write("<B> Die Summe ist: " + ergebnis + "</B><BR>");
    ergebnis = ersteEingabe * zweiteEingabe;
    //Schreibfehler ergebnis statt ergebnis
    document.write("<B> Das Produkt ist: " + ergebnis + "</B><BR>");
    // die Summe wird zum zweiten Mal ausgegeben
    // das Programm ist fehlerhaft
</script>
```

Erkennen von Schreibfehlern

```
<script language = "JavaScript">
    eingabe = prompt ("Erste Eingabe", "");
    ersteEingabe = parseInt (eingabe);
    eingabe = prompt ("Zweite Eingabe", "");
    zweiteEingabe = parseInt (eingabe);
    ergebnis = ersteEingabe + zweiteEingabe;
    document.write("<B> Die Summe ist: " + ergebnis + "</B><BR>");
    ergebnis = ersteEingabe * zweiteEingabe;
    //Schreibfehler ergebnis statt ergebnis
    document.write("<B> Das Produkt ist: " + ergebnis + "</B><BR>");
    // die Summe wird zum zweiten Mal ausgegeben
    // das Programm ist fehlerhaft
</script>
```

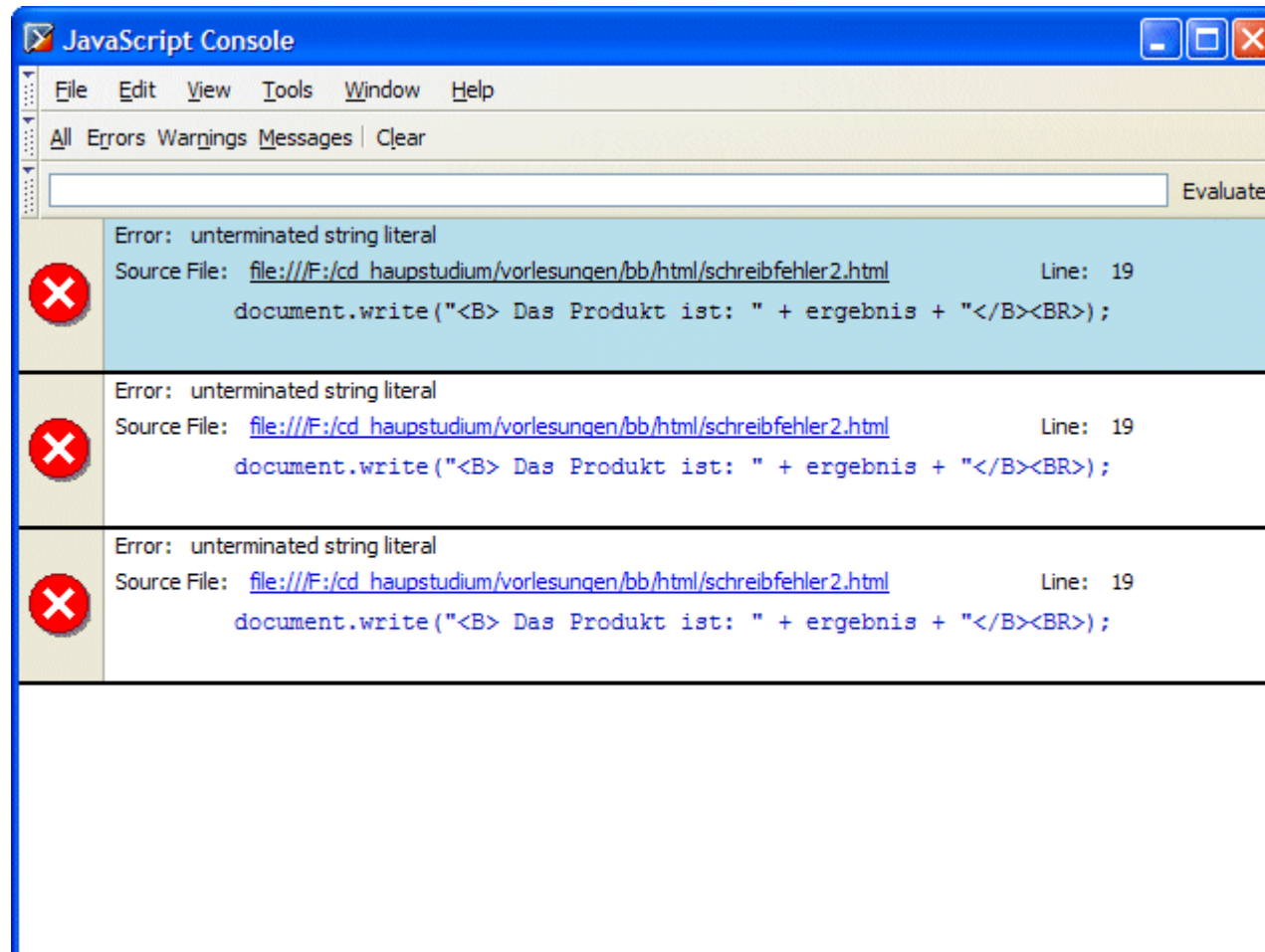
Dieser Fehler
wird angezeigt



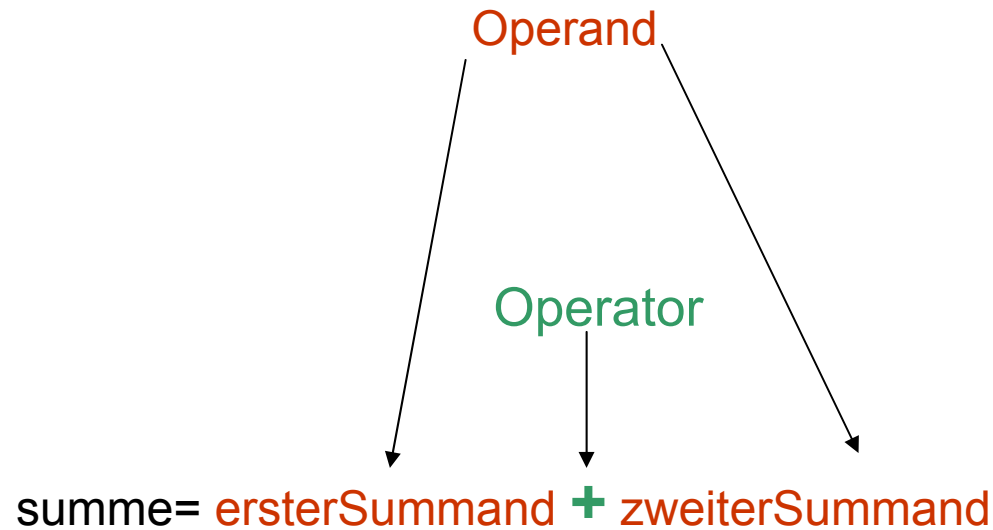
Screenshot Fehleranzeige

JavaScript-Console / Mozilla:

Tools->Web-Development->JavaScript Console



Ausdrücke



Operand:

- Inhalt einer Variable
- Konstante
- Funktionsaufruf

Operatoren – Übersicht

- Zuweisungsoperatoren: =
- Arithmetische Operatoren: + - / % *
- Bitoperatoren: & [und] ~ [Negation] | [oder] entsprechen boolescher Algebra
- Vergleichsoperatoren
- Logische Operatoren
- Tenäre Operator

Operatoren – Zuweisungsoperator

- ist einfach das Gleichheitszeichen =
summe = ersterSummand + zweiterSummand
- Der Ausdruck rechts -> ersterSummand + zweiterSummand <- wird berechnet und der Variablen -> summe <- auf der linken Seite zugewiesen
- Rechts muss immer ein Ausdruck stehen (wie schon erwähnt: Ausdruck kann auch Inhalt einer Variable sein)
- = überschreibt, d.h:
\$summe="foo";
\$summe = \$ersterSummand + \$zweiterSummand;
echo \$summe;

Operatoren – Arithmetische Operatoren

- () * / % + -
- Punkt vor Strich
- Klammern können verwendet werden, werden von innen nach aussen berechnet

Operatoren – Arithmetische Operatoren - Übung

```
<script language = "JavaScript">
    var i = 2;
    var j = 3;
    var l;
    var r= 2;
    var s=3;
    var t;
    l=i+j; // 5;
    document.write("l1: " + l + "<br>");
    l=i-j; // -1
    document.write("l2: " + l + "<br>");
    l=j%i; // 1 (Modulo-Bildung)
    document.write("l3: " + l + "<br>");
    l=i*j; // 6
    document.write("l4: " + l + "<br>");
    t=s/r; // 1.5 Division
    document.write("t: " + t + "<br>");
    l=r+i*j; // 8
    document.write("l4: " + l + "<br>");
    l=(r+i)*j; // 12
    document.write("l5: " + l + "<br>");
</script>
```

Operatoren – Arithmetische Operatoren – Abkürzende Notation

```
i += j    // i = i + j
i -= j    // i = i - j
i *= j    // i = i * j
i /= j    // i = i / j
i \% = j  // i = i \% j
i++       // i = i + 1
i--       // i = i - 1
```

Vorsicht bei Schleifen:

```
++i != i++
```

weil:

```
$a=0;
```

```
$b=0;
```

```
echo "a=". $a++ .$a++ .$a++ .$a++ ."<br>";
```

```
echo "b=". ++$b .++$b .++$b .++$b ."<br>";
```

```
a=0123
```

```
b=1234
```

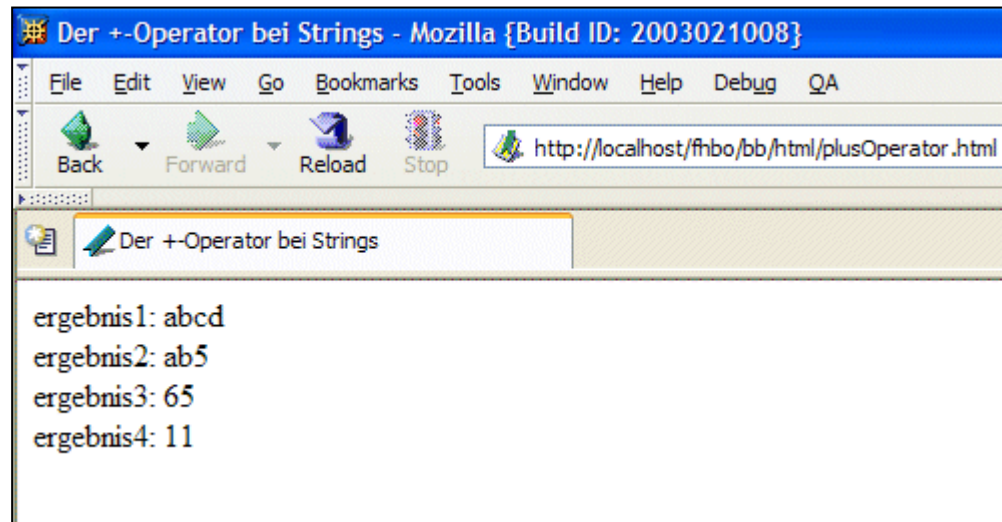
Beachten ☺

Bei \$a wird erst die
Variable ausgegeben,
dann erhöht,
Bei \$b umgekehrt

Operatoren – Plus Operator bei JavaScript

```
<script language = "JavaScript">
    var s1="ab";
    var s2="cd";
    var ergebnis;
    var i=5;
    var j="6";
    ergebnis=s1+s2;
    document.write("ergebnis1: " + ergebnis + "<br>");
    ergebnis=s1+i;
    document.write("ergebnis2: " + ergebnis + "<br>");
    ergebnis=j+i;
    document.write("ergebnis3: " + ergebnis + "<br>");
    j=parseInt(j);
    ergebnis=j+i;
    document.write("ergebnis4: " + ergebnis + "<br>");
</script>
```

Operatoren – Plus Operator bei JavaScript



Operatoren – Plus und Punkt bei PHP - Übung

```
<?php
    $s1="ab";
    $s2="cd";
    $i=5;
    $j="6";
    $ergebnis=$s1+$s2;
    echo "ergebnis1: $ergebnis <br>";

    echo "ergebnis2: $ergebnis <br>";
    $ergebnis=$s1+$i;
    echo "ergebnis3: $ergebnis <br>";

    echo "ergebnis4: $ergebnis <br>";

    echo "ergebnis5: $ergebnis <br>";
    $ergebnis=$j+$i;
    echo "ergebnis6: $ergebnis <br>";
```

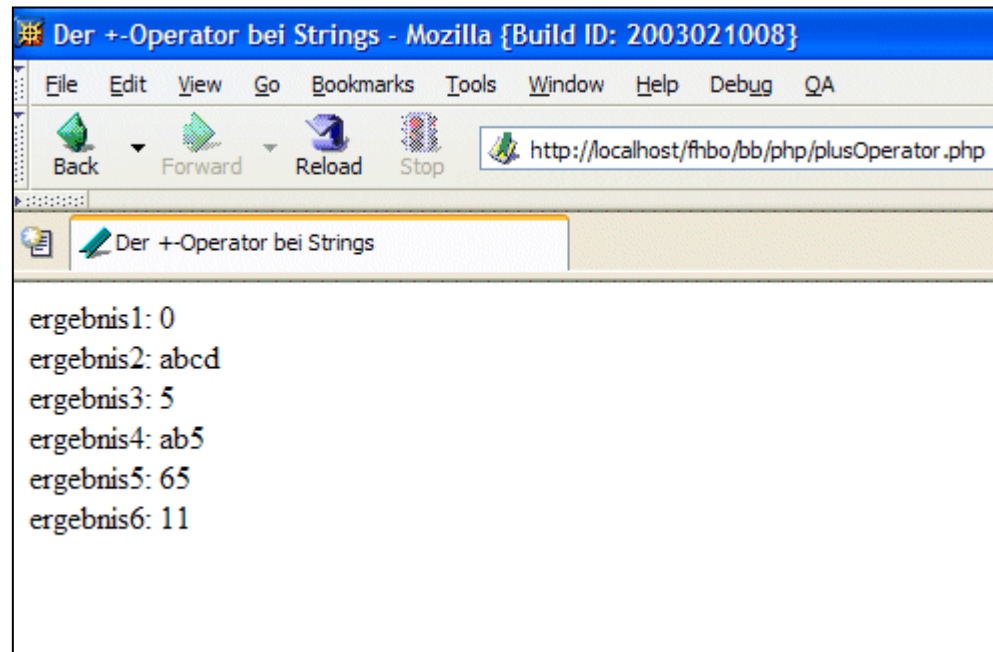
\$ergebnis=\$s1.\$s2;

\$ergebnis=\$s1.\$i;

\$ergebnis=\$j.\$i;

?>

Operatoren – Plus und Punkt bei PHP - Ergebnis



Vergleichsoperatoren in JavaScript und PHP

werden in in Kontrollstrukturen und Schleifen eingesetzt

```
for ($i=0; $i<=20; $i++)
```

```
if($foo=="HalloWelt")  
{ machWas }
```

==	x == y	x ist gleich y
!=	x != y	x ist ungleich y
>	x > y	x ist größer als y
<	x < y	x ist kleiner als y
>=	x >= y	x ist größer gleich y
<=	x <= y	x ist kleiner gleich y

Beachten:

= bedeutet Variablenzuweisung

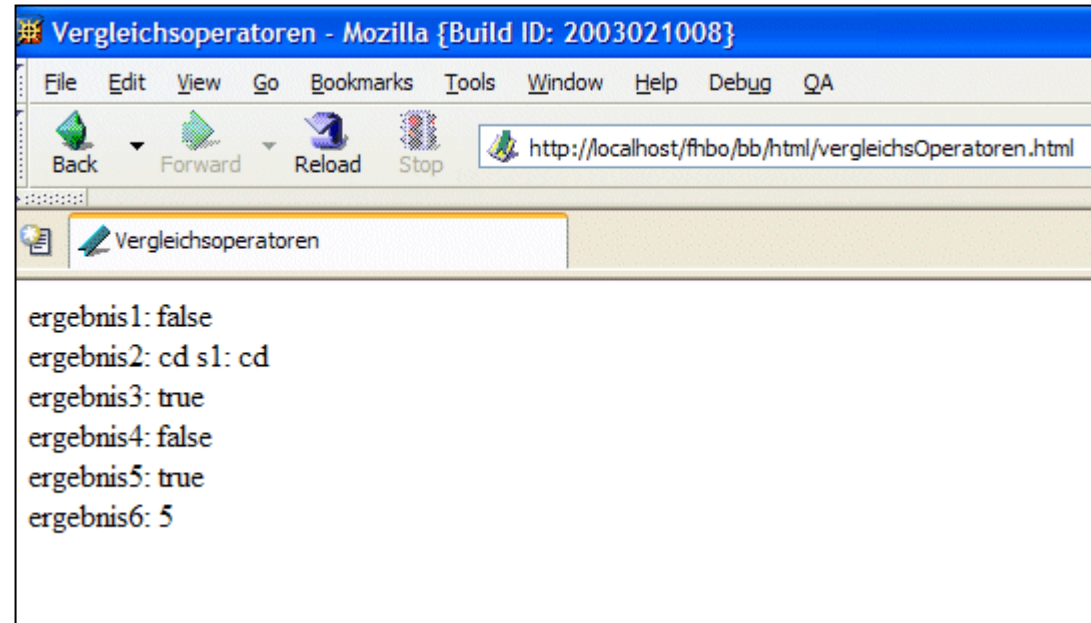
== bedeutet Vergleich !!!

VERWECHSLUNGSGEFAHR !!!!!

Vergleichsoperatoren in JavaScript - Übung

```
<script language =" JavaScript">
    var s1="ab";
    var s2="cd";
    var i=5;
    var j="5";
    ergebnis=(s1==s2);
    document.write("ergebnis1: " + ergebnis + "<br>");
    ergebnis=(s1=s2);
    document.write("ergebnis2: " + ergebnis + " s1: " + s1 + "<br>");
    ergebnis=(s1!=i);
    document.write("ergebnis3: " + ergebnis + "<br>");
    ergebnis=(s1>i);
    document.write("ergebnis4: " + ergebnis + "<br>");
    ergebnis=(j==i);
    document.write("ergebnis5: " + ergebnis + "<br>");
    ergebnis=(j=i);
    document.write("ergebnis6: " + ergebnis + "<br>");
</script>
```

Vergleichsoperatoren in JavaScript - Ergebnis



Logische Operatoren UND

Vergleiche verketteten oder negieren

Logisches Und -> &&

gibt true wenn beide Operatoren true ergeben, ansonsten false

```
var essen;  
beilage="Pommes";  
mahlzeit="CurryWurst";  
if(beilage ist gleich "Pommes" && mahlzeit ist gleich "CurryWurst")  
{  
    alert ("Mittagspause");  
}  
else  
{  
    alert ("Die Nummer vom Pizza-Service ist: 123456");  
}
```

```
$zahl=12;  
if($zahl>10 && $zahl < 20)  
{  
    echo "Zahl gefunden";  
}
```

Logische Operatoren ODER

Logische Oder Verknüpfung

Ergibt true wenn einer seiner beiden Operanden true ergibt, ansonsten false

```
$zahl=25;
```

```
if(($zahl==25) || ($zahl ==145))
```

Logische Operatoren NEGATION

wird nur auf einen Operator angewendet
ergibt true wenn sein Operand false ist und umgekehrt

```
$zahl=25;  
if($zahl != 20)  
{  
    echo "Die Zahl ist nicht 20";  
}  
else  
{  
    echo "Die Zahl ist 20";  
}
```

Logische Operatoren - Übung

```
<script language =" JavaScript">
    var i=5;
    var j=6;
    var k=6;
    var l=5;
    ergebnis = (i==j) && (j==k) // Wert von ergebnis: false, da i nicht j
    document.write("ergebnis1: " + ergebnis + "<br>");
    ergebnis = (i==j) || (j==k) // Wert von ergebnis: true, da j gleich k
    document.write("ergebnis2: " + ergebnis + "<br>");
    ergebnis = (i==l) && (j==k) // Wert von ergebnis: true
    document.write("ergebnis3: " + ergebnis + "<br>");
    ergebnis = (i==l) || (j==k) // Wert von ergebnis: true
    document.write("ergebnis4: " + ergebnis + "<br>");
    ergebnis = !(i==j) // Wert von ergebnis: true
    document.write("ergebnis5: " + ergebnis + "<br>");
    ergebnis = !(i!=j) // Wert von ergebnis: false
    document.write("ergebnis6: " + ergebnis + "<br>");
</script>
```

Ternäre Operator

```
?:  
<script language="JavaScript">  
    var i=1;  
    var j=2;  
    var k;  
    (i==j)? k=i : k=6;  
    document.write ("k hat den Wert: " + k);  
</script>
```

Konstanten

- Feste Werte innerhalb eines Programms
- Kann jederzeit und überall im Script definiert werden
- Unterscheiden sich von Variablen, dass man Wert während Programmablaufs nicht ändern kann
- Verwendung: Feste Werte, Fehlermeldungen usw.

Gleiche Werte

```
<script language = "JavaScript">
    var einkaufspreis;
    var nettoVerkaufspreis;
    var bruttoVerkaufspreis;
    einkaufspreis=prompt ("Bitte geben Sie den Einkaufspreis ein!","");
    nettoVerkaufspreis=1.16*einkaufspreis;
    bruttoVerkaufspreis=1.16*nettoVerkaufspreis;
    document.write("Netto-Verkaufspreis: " + nettoVerkaufspreis + "<br>" +
        "Brutto-Verkaufspreis: " + bruttoVerkaufspreis);
</script>
```


Konstanten in PHP

```
define ("konstantenName", "Wert");
```

```
define ("NO_CONNECTION_TO_DB_SERVER","Keine Verbindung zum Server!");  
define ("NO_DB_INTRANET", "DB Intranet konnte nicht ausgewaehlt werden!");  
define ("NO_MATRIKEL", "Die Matrikelnummer existiert nicht!");
```

[Formular zur Konstantendemonstration](#)

```
<?php  
define("MEHRWERSTEUERSATZ","1.16");  
define("GEWINNSPANNE","1.16");  
$nettoVerkaufspreis=GEWINNSPANNE*$einkaufspreis;  
$bruttoVerkaufspreis=MEHRWERSTEUERSATZ*$nettoVerkaufspreis;  
echo "Der Bruttoverkaufspreis betr&auml;gt: $bruttoVerkaufspreis!";  
?>
```