

In einer If Verzweigung muss nicht immer nur eine einzige Bedingung geprüft werden. PHP bietet hierfür logische Operatoren, die z. B. zwei Bedingungen mit dem UND Operator prüfen und nur dann ein TRUE an die If Verzweigung absetzt, wenn beide Bedingungen TRUE sind.



Liste der logischen Operatoren

Name	Beispiel	TRUE
Und and &&	<code>(\$a == 1 and \$b == "de")</code> — alternative Schreibweise — <code>(\$a == 1 && \$b == "de")</code>	Nur wenn \$a den Wert 1 hat und \$b den den String "de"
Oder or 	<code>(\$a == 1 or \$a == 5)</code> — alternative Schreibweise — <code>(\$a == 1 \$b == 5)</code>	Wenn \$a den Wert 1 oder 5 hat (eine Bedingung reicht aus).
Entweder Oder xor	<code>(\$a == 1 xor \$b == 2)</code>	Sobald \$a den Wert 1 oder \$b den Wert 2 hat, aber nicht wenn beide wahr sind.



Betrachte das Codebeispiel und kreuze jene If Verzweigungen an, die ausgeführt werden!

```
<?php
$band = "Beatles";
$songNr = 3;
$vorhanden = true;

 if($band == "Beatles" and $songNr == 3) {
    echo '<p>1: MP3-Song: Hey Jude wird gespielt</p>';
}

 if($songNr == 6 or $band == "Beatles") {
    echo '<p>2: MP3-Song: Hey Jude wird gespielt</p>';
}

 if($band == "Beatles" && $songNr == 3 && $vorhanden=false) {
    echo '<p>3: MP3-Song: Hey Jude wird gespielt</p>';
}

 if($band == "Beatles" && $songNr == 5 or $vorhanden=true) {
    echo '<p>4: MP3-Song: Hey Jude wird gespielt</p>';
}

 if($band == "Beatles" xor $vorhanden=true) {
    echo '<p>5: MP3-Song: Hey Jude wird gespielt</p>';
}

?>
```



Eine bedingte Gruppierung mittels Klammern ist ebenfalls möglich:

```
if(($a == "de" and $Nr == 3) or ($a == "at" and $Nr == 4))
```