

Mit einem einfachen Punkt `.` kann man zwei Strings zu einem verbinden. Wenn ein Text über ein Eingabefeld übertragen wird, ist manchmal notwendig, den String zu verändern. Oft muss man kritische Zeichen (z. B. `<` `>`) wandeln oder Eingaben optimieren.

PHP



`str_replace("Suche", "Ersatz", "String")`

[**suche**] Sucht nach einem Textteil in einem String.

[**Ersatz**] Ersetzt den gefunden Teil.

[**String**] In welchem String gesucht und ersetzt wird.

Die Methode ersetzt jedes Vorkommen im String!

```
<?php
    $meinString = "Autogesetz: Mein Auto ist besser als dein Auto!";
    $meineSuche = "Auto";
    $meinErsatz = "Motorrad";

    $meinString = str_replace($meineSuche, $meinErsatz, $meinString);
    echo $meinString;
?>
```



Weitere Funktionen zum Ersetzen sind:

`str_ireplace()` ignoriert Groß- und Kleinschreibung, sonst gleich wie `str_replace()`

`substr_replace()` ersetzt einen String ab einem bestimmten Startpunkt mit einer bestimmten Länge.

PHP

`strtolower($string)` ← Wandelt einen String in Kleinbuchstaben

`strtoupper($string)` ← Wandelt einen String in Großbuchstaben

`ucfirst($string)` ← das erste Zeichen als Großbuchstabe.

```
<?php
    $eingabe = "schOPPENHAUER";
    $eingabe = strtolower($eingabe);
    echo "Nachname: " . strtoupper($eingabe);
    echo "<br>Sehr geehrter Herr " . ucfirst($eingabe);
?>
```

PHP



`trim($string)`

Löscht Leerzeichen und Whitespace am Anfang und am Ende eines Strings. Unter Whitespace versteht man auch Zeilenumbrüche (`\n``\r`)

```
<?php
    $eingabe = "  Datenverarbeitung \n\r";
    $eingabe = trim($eingabe);
    echo "(" . $eingabe . ")<br>";
?>
```