

F.2 Reguläre Ausdrücke in PHP

PHP enthält mehrere Methoden zur Anwendung von regulären Ausdrücken. Die Methoden beginnen mit dem Präfix `preg_`. Es ist unter anderem möglich,

- nach solchen Ausdrücken zu suchen (`preg_match`)
- Teile zu ersetzen (`preg_replace`)
- Zeichenketten aufzuteilen (`preg_split`)

Die Syntax von `preg_match` ist:

```
int preg_match (
    string $Suchmuster ,
    string $Zeichenkette
    [, array &$Treffer
    [, int $Flags
    [, int $Versatz  ]])
```

Die ersten beiden Argumente enthalten das Suchmuster und die zu durchsuchende Zeichenkette. Das Suchmuster wird in Begrenzungszeichen (üblicherweise `/`) eingebettet. Danach können Modifikatoren folgen, mit denen das Suchverhalten im Detail eingestellt wird. So kennzeichnet ein `i` eine Suche unabhängig von Groß- und Kleinschreibung. Die optionalen weiteren Argumente haben folgende Bedeutung:

- **\$Treffer**: falls angegeben, wird in dieses Feld das gefundene Muster eingetragen. Das erste Feldelement enthält das komplette Muster, eventuelle Teilmuster werden in die folgenden Elemente eingetragen.
- **\$Flags**: falls der Wert `PREG_OFFSET_CAPTURE` gesetzt ist, wird in dem Ergebnisfeld auch die jeweilige Position, an der das Muster gefunden wurde, eingetragen.
- **\$Versatz**: die Suche beginnt erst an der angegebenen Stelle in der Zeichenkette

Als Rückgabewert liefert `preg_match` 1 falls das Muster gefunden wurde, sonst 0. Im folgenden sind einige erste Beispiele: zusammengestellt. Einfache Suche nach `berg` in `Friedberg`:

```
$res = preg_match ("/berg/", "Friedberg" , $match );
echo $res . "<br>";
print_r($match);
```

Ergebnis:

```

1
Array ( [0] => berg )

$res = preg_match ("/..berg/", "Friedberg" , $match );
echo $res . "</br>";
print_r($match);

```

Ergebnis:

```

1
Array ( [0] => edberg )

```

Einsatz von Zeichentypen, Beispiel `\w` für *wort*-Zeichen:

```

$res = preg_match ("/b\\w\\wg/", "Friedberg" , $match );
echo $res . "</br>";
print_r($match);

```

Die doppelten `\`-Zeichen sind notwendig, da sonst PHP die Folge als Sonderzeichen interpretieren würde. Ergebnis:

```

1
Array ( [0] => berg )

$res = preg_match ("/..berg/", "Friedberg" ,
    $match, PREG_OFFSET_CAPTURE );
echo $res . "</br>";
print_r($match);

```

Ergebnis:

```

1
Array ( [0] => Array ( [0] => edberg [1] => 3 ) )

$res = preg_match (
    "/(\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})/",
    "111.23.56.223" ,
    $match, PREG_OFFSET_CAPTURE );
echo $res . "</br>";
print_r($match);

```

Ergebnis (umformatiert):

```

1
Array (
    [0] => Array ( [0] => 111.23.56.223 [1] => 0 )
    [1] => Array ( [0] => 111 [1] => 0 )
    [2] => Array ( [0] => 23 [1] => 4 )
    [3] => Array ( [0] => 56 [1] => 7 )
    [4] => Array ( [0] => 223 [1] => 10 ) )

```