

5 – HTML in PHP einbetten

Infoblatt



ChatGPT erklärt: Warum HTML-Tags in PHP ausgeben?

Wenn man in PHP **echo** benutzt, um etwas auf der Webseite auszugeben, dann erscheint das ohne weitere HTML-Formatierung im Browser.

Wenn man jedoch HTML-Tags in den **echo**-Befehl einbindet, kann man den Text gestalten oder strukturieren – also z. B. Überschriften, Absätze oder Farben hinzufügen.

```
<?php
echo "Überschrift";

echo "<h1>Überschrift</h1>";
?>
```

Überschrift

Überschrift

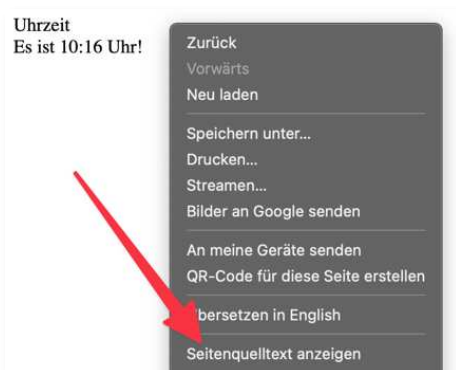
Man bettet also HTML in **echo** ein, damit man nicht nur Text, sondern richtiges Webseiten-Layout mit PHP erzeugen kann.

Außerdem kann man die HTML-Ausgabe mit CSS gezielt formatieren.

Bisher haben wir PHP meistens ohne HTML benutzt, etwa so:

```
<body>
<?php
echo "Uhrzeit";
echo "<br>";
echo "Es ist " . date('h:i') . " Uhr!";
?>
</body>
```

Wenn wir uns aber den Seiten Quelltext anzeigen lassen ...



... sehen wir, dass wir echt dreckigen Code generiert haben:

```
10 <body>
11   Uhrzeit<br>Es ist 10:16 Uhr!</body>
12
```

Es sieht im Browser besser aus, wenn wir alle Inhalte irgendwie durch HTML auszeichnen, z.B. durch **<p>**, **<h1>**, **<h2>**, **<div>** usw. Vor allem die Verwendung von CSS wird einfacher, weil man dann bspw. gezielt einzelne Tags formatieren kann (s.u.).



HTML-Tags echoen

Ab sofort versuchen wir, ordentlichen HTML-Code zu generieren. Wir schreiben also NICHT mehr

```
echo "Willkommen";  
echo "Schön, dass du da bist!";
```

sondern

```
echo "<h1>Willkommen</h1>";  
echo "<p>Schön, dass du da bist!</p>";
```

Wiederholung: HTML-Tags

Häufig benutzte HTML-Tags sind:

<code><h1>Überschrift</h1></code>	steht für »headline« = Überschrift. Möglich h1 bis h6
<code><p>Inhalt, Inhalt</p></code>	steht für »paragraph« = Absatz
<code>
</code>	steht für »break« = einfacher Zeilenumbruch
<code><table>, <thead>, <tr>, <td></code>	Tabellen
<code><div>Blockelement</div></code>	steht für »division« - ein Blockelement, das viele andere Elemente zusammenfassen kann
<code></code>	steht für »image« - Bild (siehe unten)



Wiederholung: Zeichenketten und Variablen verknüpfen

Mit dem Punkt verknüpfen wir Zeichenketten, Variablen etc. Also etwa

```
$ueberschrift = $_POST['tfUeberschrift'];  
echo "<h1>" . $ueberschrift . "</h1>";
```

Wir können Variablen oft auch direkt in den **echo**-String schreiben:

```
echo "<h1>$ueberschrift</h1>";
```

Die Ausgabe wäre in beiden Fällen die gleiche.

Tipp: Verwendung von CSS

Du hast zwei Möglichkeiten, CSS zu verwenden:

Externes Stylesheet

Das ist unser übliches Vorgehen – wir hängen ein Stylesheet an, wo wir die CSS-Regeln definieren.

```
<link rel="stylesheet" href="style.css">
</head>
<body>
<?php
    echo "<h1>Dein Passwort</h1>";
    echo "<p>$passwort</p>";
```

```
05-html-in-php > # style.css > ...
1  h1 {
2      color: green;
3  }
4  p {
5      font-size: 15px;
6      color: red;
7  }
```

inline-Definition

Wir können CSS-Regeln auch direkt als style-Attribut echoen. Allerdings müßen wir uns damit den HTML-Code voll, das hat zwei Nachteile:

1. Der Code wird unübersichtlich.
2. Die Stil-Definitionen sind nicht wiederverwendbar.

```
echo "<h1 style='color: green;'>Dein Passwort</h1>";
echo "<p style='color: red; font-size: 15px;'>$passwort</p>";
```

Pro-Tipp: Zeilenumbrüche und Tabstopps im Quellcode

Wenn Sie in den HTML-Code schauen (rechte Maustaste auf die Seite -> Seitenquelltext anzeigen), sehen Sie, dass keinerlei Zeilenumbrüche gemacht werden:

PHP-Code

```
echo "<h1>Überschrift</h1>";
echo "<p>Willkommen im Shopsystem!</p>";
```

Auf der erzeugten HTML-Seite wird ausgegeben:

```
<h1>Überschrift</h1><p>Willkommen im Shopsystem!</p>
```

Wenn Sie wollen, dass im Code, der beim Browser ankommt, Zeilenumbrüche gemacht werden, dann müssen Sie `\n` echoen:

PHP-Code

```
echo "<h1>Überschrift</h1>";
echo "<p>Willkommen im Shopsystem!</p>";
```

Auf der erzeugten HTML-Seite wird ausgegeben:

```
<h1>Überschrift</h1>
<p>Willkommen im Shopsystem!</p>
```

Profis können mit `\t` auch noch Einrückungen vornehmen.

