

Geschrieben im September 2025

CSS-Gestaltung von wichtigen Webseiten-Bereichen

Mit den vier HTML-Codes:

- `<header>`Kopf-Bereich`</header>`
- `<nav>`Link-Bereich`</nav>`
- `<main>`Inhalts-Bereich`</main>`
- `<footer>`Fuß-Bereich`</footer>`

können Webmaster Webseiten, mit Hilfe von Editoren (<https://phcode.dev>), innerhalb der Body-Bereiche, übersichtlicher als früher strukturieren und für Suchmaschinen optimal lesbar machen.

Beim Codieren von Webseiten werden die oben genannten vier Webseiten-Bereiche (`header`, `nav`, `main` und `footer`) in die jeweiligen Body-Bereiche von Webseiten geschrieben. Nur Inhalte in Body-Bereichen, sind in fertigen Webseiten sichtbar, natürlich ohne HTML-Codes.

```
<body>
  <header>Kopf-Bereich</header>
  <nav>Link-Bereich</nav>
  <main>Inhalts-Bereich</main>
  <footer>Fuß-Bereich</footer>
</body>
```

Alle HTML-Bereiche oder Elemente (Überschriften, Tabellen, Listen usw.) kann man mit CSS-Codes (in dazu gehörigen CSS-Dateien) individuell direkt gestalten. Umständliche CSS-Gestaltungs-Codes, in den einzelnen HTML-Codes von Webseiten, werden mit Hilfe von CSS-Dateien überflüssig, was eine enorme Arbeits-Erleichterung für Webmaster ist. Verbindungs-Links für CSS-Dateien werden in Head-Bereiche von Webseiten platziert:

Verbindungs-Links zu einer CSS-Datei:

```
<link href="css/style.css" rel="stylesheet">
```

css ist der Ordner, in dem die CSS-Datei untergebracht ist.

Body-Bereich

So habe ich in der CSS-Gestaltungs-Datei meiner Website (`style.css`) die Body-Bereiche meiner Webseiten codiert:

```
body {
max-width: 65%;
margin: 0 auto;
font-size: 17px;
```

Maximale Breite
Mittige Ausrichtung
Schrift-Größe

```
font-family: system-ui, sans-serif;
line-height: 1.5rem;
background-color: white;
}
```

Schrift-Familie
Zeilen-Höhe oder Abstand
Hintergrund-Farbe

Die Hintergrundfarbe mit weiß anzugeben ist eigentlich unnötig, weil das automatisch geschieht, wenn ich nichts angebe. Da ich aber alle Hintergründe meiner Webseiten (html) Dodgerblau gefärbt habe, muss ich die Hintergrund-Farbe der body-Bereiche sehr wohl mit weiß codieren, weil sonst alles Dodgerblau eingefärbt würde.

system-ui: ist die Schriftart des jeweiligen Betriebs-Systems (Windows, Apple, Linux)
sans-serif: glatte Schrift ohne Verzierungen (Arial, Calibri oder Verdana zum Beispiel)
serif: Schrift mit Verzierungen (Times New Roman)
monospace: Schrift mit gleich breiten Buchstaben (Courier New)

```
html {
background-color: dodgerblue;
}
```

Farbe des gesamten Hintergrunds

Header-Bereich

In Header-Bereiche schreibe ich die h1-Haupt-Überschrift der jeweiligen Webseite und eine untergeordnete h2-Erklärungs-Überschrift. Dazu kann man noch ein Erkennungs-Logo machen, wenn man das will.

Die Header-Bereiche meiner Webseiten habe ich in der CSS-Datei (**style.css**) so codiert:

```
header {
padding: 0.3rem;
background-color: #ffffaa;
}
```

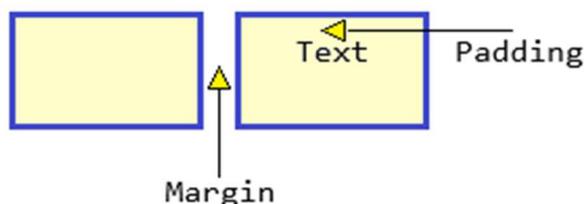
Innerer Abstand zum Rand
Hintergrund-Farbe (hellgelb)

Die H1 und H2-Überschriften in den Header-Bereichen meiner Webseiten habe ich in der CSS-Datei (**style.css**) so codiert:

```
header h1, h2 {
margin-top: 0.3rem;
margin-bottom: 0.3rem;
margin-left: 1rem;
}
```

Äußerer Abstand zum Rand

top: oben
bottom: unten
left: links
right: rechts



Nav-Bereich

In Nav-Bereiche von Webseiten wird die grundsätzliche Verlinkung (Navigation), von Webseite zu Webseite, eingebaut.

Die Nav-Bereiche in meinen Webseiten habe ich in der CSS-Datei (`style.css`) so codiert:

```
nav {  
  display: flex;           Inhalte flexibel gestalten  
  justify-content: center; Inhalte mittig ausrichten  
  flex-direction: row;    Von links nach rechts Anordnung  
  flex-wrap: wrap;        Zeilen-Umbruch  
  background-color: royalblue; Hintergrund-Farbe: Dunkelblau  
}
```

*Display-Flex gehört zur relativ neuen **CSS-Flexbox-Technik**. Damit kann man HTML-Boxen, Bereiche oder Elemente, die man früher nur mit Hilfe von Tricks nebeneinander platzieren konnte, ganz einfach dazu bringen, sich nebeneinander- und bei Bedarf (Kleinere Displays) auch untereinander, anzuordnen.*

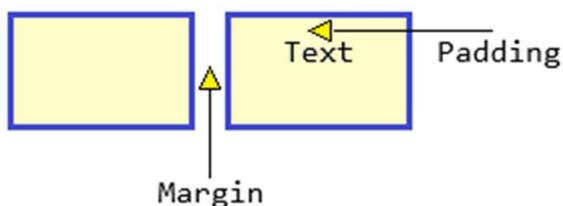
Gestaltung der Links im Nav-Bereich

```
nav a {  
  color: white;           a = Code-Buchstabe für Link  
                          Linkfarbe: Weiß  
  font-size: 1.4rem;     Schrift-Größe  
  margin: 1rem;          Abstand von Link zu Link und zum Rand  
}
```

Main-Bereich

In die Main-Bereiche von Webseiten, **nicht mit Body-Bereichen verwechseln**, kommen die wirklichen Webseiten-Inhalte.

```
main {  
  padding-left: 1rem;     Innerer Abstand zum Rand: links  
  padding-right: 1rem;   Innerer Abstand zum Rand: rechts  
}
```



Footer-Bereich

In die Footer-Bereiche von Webseiten kommen:

E-Mail-Adresse, Kontakt-Seite, Datenschutz-Seite, Impressum usw. Grundsätzlich habe ich die Footer-Bereiche wie die Nav-Bereiche gestaltet.

```
footer {  
display: flex;  
justify-content: center;  
flex-direction: row;  
flex-wrap: wrap;  
background-color: royalblue;  
margin-bottom: 1rem;  
}
```

Inhalte flexibel gestalten
Inhalte mittig ausrichten
Von links nach rechts Anordnung
Zeilen-Umbruch
Hintergrund-Farbe: Dunkelblau
Äußerer Abstand nach unten

```
footer a {  
color: #ffffff;  
font-size: 1.2rem;  
margin: 1rem;  
}
```

a = Code-Buchstabe für Link
#ffffff hat die gleiche Funktion wie white (weiß)
Äußerer Abstand in alle Richtungen

Listen

```
ul {  
list-style: none;  
padding-left: 0;  
}
```

Ohne Vorzeichen

Alternativen für: none

list-style-type:

disc (gefüllter Kreis), **square** (Quadrat), **circle** (leerer Kreis)

Linien

```
hr {  
border: 0;  
height: 2px;  
background-color: deepskyblue;  
}
```

Ohne Rahmen
Dicke der Linie
Linien-Farbe

Überschriften

```
h3 {  
color: blue;  
font-size: 1.3rem;
```

Schrift-Farbe
Schrift-Größe

```
background-color: lavender;  
padding: 3px;  
font-weight: 500;  
}
```

Hintergrund-Farbe
Hintergrund-Höhe
Stärke der Schrift

Grafiken responsiv machen

Für Darstellungen in Mobil-Telefonen

```
img {  
vertical-align: middle;  
max-width: 100%;  
height: auto;  
}
```

Vertikale Ausrichtung: Mitte
Display-Anpassung
Bild-Höhe: automatisch

Tabellen-Darstellung

```
table {  
border-collapse: collapse;  
}
```

separate: Doppelte Rahmen-Linie
collapse: Normale Rahmen-Linie

```
td {  
padding: 5px;  
border: 1px dashed #0000ff;  
}
```

Tabellen-Inhalts-Zelle
Innerer Abstand: Schrift zum Rand
Rahmen: Stärke, Darstellung, Farbe

```
th {  
padding: 5px;  
border: 1px dashed #0000ff;  
background-color: #ffffcc;  
}
```

Tabellen-Titel-Zelle
Innerer Abstand: Schrift zum Rand
Rahmen: Stärke, Darstellung, Farbe
Hintergrundfarbe: hellgelb

Außer **dashed** (*gestrichelte Linie*) geht auch **solid** (*durchgehende Linie*) und **dotted** (*Punkt-Linie*). Es gibt noch mehr Gestaltungs-Möglichkeiten für Linien, die ich aber nicht benutze.

CSS-Codes für Darstellungen von HTML-Elementen in Mobil-Telefonen

```
@media all and (max-width: 600px)
{nav, footer {flex-direction: column;}}
```

 Tabellen-artiges Aussehen

```
@media all and (max-width: 600px)
{header, footer, nav, h3, main h2, figcaption
{text-align: center;}}
```

 Mittige Ausrichtung

```
@media all and (max-width: 600px)
{body {max-width: 100%;}}
```

 Maximale Breite: 100 Prozent

```
@media all and (max-width: 600px)
{dd a, li a {line-height: 3rem;}}
```

 Zeilen-Höhe