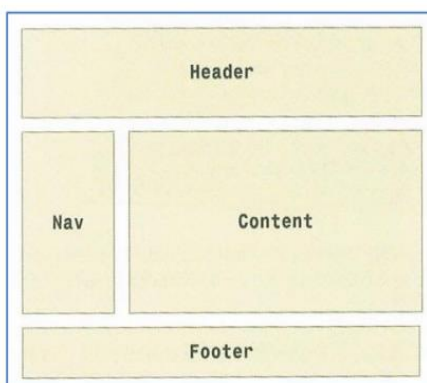


LAYOUT MIT DIV-BOXEN IN HTML5 UND CSS

Verlinkung der CSS-Datei im Headbereich der HTML-Seite

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

Grundgerüst einer allgemeinen HTML-Seite



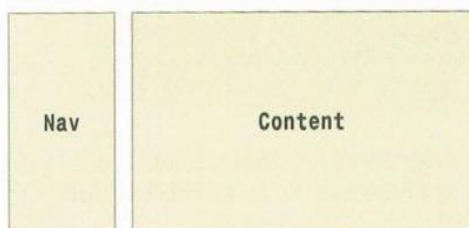
```
<body>
  <header>
    <div id="header">Kopfleiste</div>
  </header>

  <nav>
    <div id="navigation">Navigation</div>
  </nav>

  <main>
    <div id="content">Content</div>
  </main>

  <footer>
    <div id="footer">Footer</div>
  </footer>
</body>
```

Grundgerüst einer zweispaltigen HTML-Seite



```
<body>
  <nav>
    <div id="navigation">Navigation</div>
  </nav>

  <main>
    <div id="content">Content</div>
  </main>
</body>
```

Layoutpositionierung einer zweispaltigen HTML-Seite in der CSS-Datei (style.css)

```
body {
    font-family: Arial, Tahoma, Times New Roman;
    background-color: coral;
    margin: 0;
    font-size: 0.8em;
}

* {
    box-sizing: border-box;
    /*Padding and border ändert die Größe der DivBox nicht! */
}

#navigation {
    float: left;
    width: 20%;
    min-width: 80px;
    padding-left: 20px;
}

#content {
    float: left;
    width: 80%;
    height: 90%;
    padding-left: 15px;
}

img {
    width: 70%;
    height: auto;
}

/* Responsive layout: Navigation und Content untereinander darstellen*/
@media screen and (max-width: 600px) {
    #navigation, #content {
        width: 100%;
        height: auto;
    }
}

a:link { color: blue; text-decoration: none }
a:visited { color: blue; text-decoration: none }
a:active { color: blue; text-decoration: none }
a:hover { color: blue; text-decoration: underline }
```

Das Box-Modell

Als »Box« wird der gesamte Bereich bezeichnet, den ein Element einnimmt – einschließlich Rahmen und Abstände, d.h.

Boxbreite = `width + padding + border + margin`

Boxhöhe = `height + padding + border + margin`



margin

margin, der äußere Rand, bildet den Abstand des Elements gegen den umgebenden und benachbarten bzw. folgenden Block. *margin* wird insbesondere beim Positionieren von HTML-Blöcken eingesetzt, um die Layout-Elemente einer Webseite auf Abstand voneinander zu halten.

padding

padding ist ein »Innenabstand« zum Rand und hält die Inhalte einer Box vom Rahmen der Box fern. CSS *padding* zeigt die gleiche Hintergrundfarbe wie das Element. Zwischen *margin* und *padding* liegt *border* – der Rahmen.

CSS-Befehl: Absolute Boxgröße einschließlich Margin und Padding

```
* {
  box-sizing: border-box;
  /*Padding and border ändert die Größe der DivBox nicht! */
}
```

Das `*` ist eine Wildcard und steht eigentlich für ALL

CSS Syntax

`float: none|left|right|initial|inherit;`

Die *float*-Eigenschaft ist ein mächtiges Werkzeug zur Seitengestaltung. Mit Hilfe dieser Eigenschaft können Inhaltsbereiche nebeneinander gestellt werden; so entstehen z.B. die beliebten »Spaltenlayouts«.

Mögliche Werte:

left

Das Element wird möglichst weit links platziert und die Inhalte der nachfolgenden Elemente fließen rechts daran vorbei.

right

Das Element wird möglichst weit rechts platziert und die Inhalte der nachfolgenden Elemente fließen links daran vorbei.

none

Das Element wird nicht umflossen

Die float-Eigenschaft kann jedem Element zugewiesen werden.

CSS Syntax

`display: none|block|inline-block|inline|flex|grid|table|table-row|table-cell;`

Jedes HTML-Element ist eine Box

Der Browser erzeugt für jedes HTML-Element eine Box.

CSS *display* legt den Typ der Box fest und weist ihr ein Verhalten zu. Das Verhalten von *display: block* ist: "Neue Zeile", das Verhalten einer Tabellenzelle (*display: table-cell*): "Text kann vertikal zentriert werden".

Selbst innerhalb von Textzeilen entstehen Boxen, wenn Wörter in einem HTML-Tag wie *em* oder *a* liegen. Wird einem HTML-Element wie *span* oder *em* ein *display: block* zugewiesen, entsteht ein Zeilenumbruch vor und nach dem Element.

[display: inline-block](#) bzw. [display: block](#)

Das sind inline-Elemente, die ähnlich wie Blockelemente formatiert werden können, z.B. können die Attribute *width*, *height*, *padding* und *margin* gestzt werden.

[inline-block](#) wandelt z.B. ein *label*-Tag in Formularen in ein Blockelement ohne Zeilenumbruch um. Mit [display: block](#) erhält man ein Blockelement mit Zeilenumbruch.

Mit diesen Elementen kann ein Formular in Spalten ohne aufwändige Tabellen aufgebaut werden.

Name	<input type="text"/>
Hausnummer	<input type="text"/>
PLZ	<input type="text"/>

`display: inline-block` auch ist eine Alternative zu *float*.

[display: flex](#)

Die Flexbox ist ein alternatives Modell für die Positionierung neben dem bekannten Block-Modell mit *position: absolute / relative* und *float: left / right*.

Das klassische Block-Modell für die Positionierung richtet Elemente entweder horizontal (*display:inline*) oder vertikal (*display:block*) aus. Äquivalent dazu richtet der Flex-Container die in ihm enthaltenen Elemente ebenfalls entweder horizontal (*display:flex; flex-direction:row*) oder vertikal (*display:flex; flex-direction:column*) aus. Der Flex-Container bestimmt als das Verhalten seiner Kinder.

Weitere Eigenschaft des Flex-Containers lassen die Flexboxen umbrechen, richten Flexboxen links, rechts, zentriert, an der Oberkante oder an der Unterkante aus. Der freie Platz zwischen den Boxen kann gleichmäßig verteilt werden.

Box 1	Box 2	Box 3
-------	-------	-------

Das Flex-Modell hat in den letzten Jahren einiges an Wandel mitgemacht, sowohl in Hinsicht auf die Spezifikation als auch in Hinsicht auf die Implementierung in den verschiedenen Browsern.

[display: table-row / table-cell](#)

Tabellen sind out und tabu? HTML-Tabellen als Layoutgerüst sind dem allgegenwärtigen *float* gewichen. Dabei hatten Tabellen und Tabellenzellen einige nützliche Eigenschaften, die mit Hilfe von *display: table*, *table-row* und *table-cell* wieder zum Vorschein kommen: