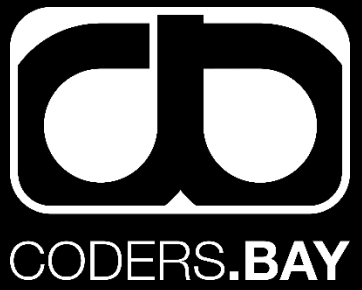


Seitenlayout in HTML & CSS
Schrödinger Kapitel 07

Inline-Elemente

- Als Inline-Elemente bezeichnet man HTML Elemente, die mitten im Text vorkommen können. Inline-Elemente verhalten sich genau wie Text, das heißt, sie werden umbrochen wie Fließtext, können mehrere Zeilen oder halbe Zeilen umfassen und so weiter.
- Inlineelemente erzeugen keine neue Zeilen
- können keine Werte für die Eigenschaften `width` und `height` annehmen
- Ausrichtung nur mit dem Innenabstand `padding` umsetzbar

- Beispiele
  - `a`
  - `abbr`
  - `b`
  - `cite`
  - `em`
  - `i`
  - `img`
  - `label`
  - `span`
  - `strong`
  - `sub`
  - `sup`

Block und Inline haben auch eine grundsätzliche Bedeutung für die Verschachtelung von Tags: Inline-Elemente dürfen nur Text und weitere Inline-Elemente enthalten, niemals Blockelemente. Blockelemente dürfen immer Inline-Elemente enthalten, manche auch weitere Blockelemente

BLOCKELEMENTE

CODERS.BAY

- Als Blockelemente werden HTML-Elemente bezeichnet, die nicht innerhalb des Textes vorkommen, aber Text umfassen können. Blockelemente dulden von Natur aus nichts anderes neben sich: Solange man es nicht umstellt, steht das vorherige Element über einem Blockelement, das folgende darunter. Außerdem sind Blockelemente immer rechteckig. Sie erzwingen einen Zeilenumbruch vor und hinter sich
- ohne weitere CSS-Eigenschaften erzwingen Blockelemente wie im folgenden HTML-Code-Ausschnitt `h1`, `h2`, `p`, `ul` und `li` innerhalb des Dokumentenflusses einer HTML-Datei einen Zeilenumbruch
- Das heißt, die Überschrift `h1` kann sich ohne entsprechende CSS-Eigenschaften niemals neben einer Überschrift `h2` befinden. Das Gleiche gilt auch für den Textabsatz und die darunter liegende Liste.
- Per CSS dem Blockelementen einen innenabstand `padding` und einen Außenabstand `margin` zuweisen.
- Au den Werten für Breite für den Inhaltsbereich, den Innenabständen und eventuellen Rahmeneigenschaften ergibt sich die Gesamtbreite eines Elements
- Der Außenabstand `margin` fließt nicht in die Berechnung dieses Wertes ein

```
...
<h1>Primary Headline</h1>
<h2>Subheadline</h2>
<p>Lorem ipsum dolor sit amet ...</p>
<ul>
  <li>Lorem ipsum...</li>
  <li>Sadispscing...</li>
  <li>Tempor...</li>
</ul>
...
```

Box-Model – stapelbare Boxen

CODERS.BAY

- `<div>` ist im Wesentlichen für Blockelemente das was `<span>` für Inline Elemente ist
  - hat keine eigene Bedeutung
  - ist ein generischer Container
  - besitzt keinen Abstand
  - wenn der Text zu lang ist, geht er einfach über den Rand

Inline-Block-Elemente

CODERS.BAY

- mit CSS erzeugbar
- mit der Eigenschaft `display` und dem Wert inline-block können Boxen erzeugt werden, die nach innen hin die Eigenschaft eines Blockelements besitzt und nach außen die eines Inlineelements
- kann horizontal und vertikal ausgerichtet werden

Weit hinten, hinter den Wortbergen, fern der Länder Vokalien und Konsonantien leben die Blindtexte...

Weit hinten, hinter den Wortbergen, fern der Länder Vokalien und Konsonantien leben die Blindtexte...

```
p { display: block; }
```

Weit hinten, hinter den Wortbergen, fern der Länder Vokalien und Konsonantien leben die Blindtexte...

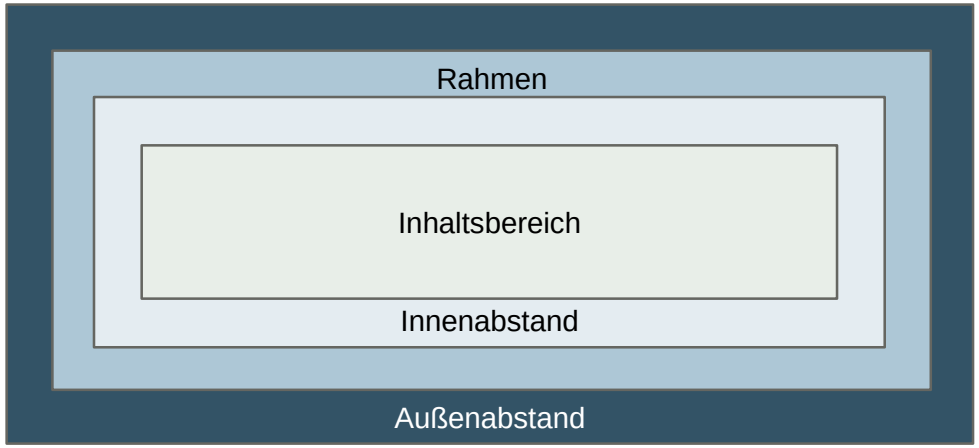
Weit hinten, hinter den Wortbergen, fern der Länder Vokalien und Konsonantien leben die Blindtexte...

```
p { display: inline-block; }
```

Box-Modell

- ein CSS-Layout ist grundsätzlich aus rechteckigen Boxen aufgebaut
- all diese Boxen basieren auf ein und dem selben Schema – dem Box-Modell
- der Außenabstand `margin` bleibt bei der Berechnung der Gesamtbreite des entsprechenden Elements außen vor

  - **Inhaltsbrereich**
    - Bereich für Texte, Bilder, Videos, etc...
  - **Innenabstand – "`padding`"**
    - befindet sich innerhalb des Elements und übernimmt Hintergrundfarben und –grafiken dieses Elements
  - **Rahmen – "`border`"**
    - umschließt den Inhaltsbereich eines Elements und dessen Innenabstand `padding`
  - **Außenabstand – "`margin`"**
    - befindet sich außerhalb eines Elements,
    - wird nicht zur Berechnung der Gesamtbreite herangezogen
    - kann keine Hintergrundfarbe und –grafik dieses Elements anzeigen
    - beschreibt somit einzig und allein den Abstand zu den umliegenden Elementen



- Wenn ein Inhaltsbereich eines Elements beispielsweise eine Breite `width` von 400 Pixel besitzt, der Innenabstand `padding` 20 Pixel und der Rahmen `border 5px` beträgt, resultiert daraus für dieses Element, sofern sich der Browser im Standardmodus befindet, eine Breite von 450px = (400 + 2 x (20 + 5))

Überblick

CODERS.BAY

Tag/Attribut	Attributwert	Bedeutung	Und wenn es eine Stapelzeug-Pappkiste wäre
`<div>`		Das bedeutungsloseste Blockelement, seit es HTML und CSS gibt	Die Kiste selbst. Alles andere sind Eigenschaften dieser Kiste
Attribute für Blockelemente:			

`padding-top`
`padding-right`
`padding-bottom`
`padding-left`
	Größenangabe in px, em oder Prozent	Der Abstand zwischen dem Inhalt des Elements und dem Rahmen	Die vielen bunten Styroporflocken zwischen Packgut und der Kiste oder Blasenfolie zum Ploppen lassen

`padding`		Kurzschreibweise für die vier Eigenschaften oben. Ein wert gilt für alle vier Seiten, mehrere Werte werden verteilt, nach dem Gesetz des Kompasses	Styroporflocken auf allen Seiten!
`border` und `border-*`	Dicke Rahmen in Pixeln, Farben und Linienart (gestrichelt, durchzogen, ...)	Dicke des Rahmens in Pixeln, Farbe und Beschaffenheit	Dicke, Farbe und Beschaffenheit der Kartonpappe

Überblick

CODERS.BAY

Tag/Attribut	Attributwert	Bedeutung	Und wenn es eine Stapelzeug-Pappkiste wäre

`margin-top`
`margin-right`
`margin-bottom`
`margin-left`
	Größenangabe in `px`, em oder Prozent	Der Mindestabstand zwischen dieser Box und der nächsten in dieser Richtung. So viel Platz bleibt mindestens frei	Der Sicherheitsabstand zwischen dieser Kiste und der nächsten, etwa Holzlatten zwischen zwei Kisten, damit sie sich nicht berühren, das ist eine Margin

`margin`		Wieder eine Kurzschreibweise, genau wie padding	Abstand rundherum, diese Kiste muss gefährlich sein
`width`	Größenangabe in px, em und Prozent	Breite des Inhaltsbereichs, wirklich nur des Inhaltes, alles andere, sogar das Padding, kommt noch dazu	Die Breite des Packgutes, ohne Styropor, ohne Karton, ohne Sicherheitsabstand
`height`	Größenangabe in px, em und Prozent	Höhe des Inhaltsbereichs, auch hier kommt alles Weitere noch dazu	Die Höhe des Packgutes, ohne Styropor, ohne Karton, ohne Sicherheitsabstand

Relativ und absolut

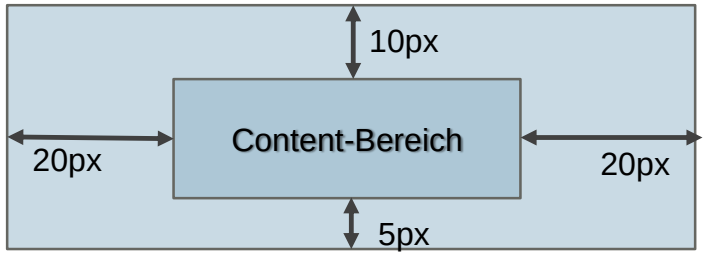
- Relative Angaben in Prozent
  - Prozentangaben bei Blockelementen beziehen sich immer auf das umgebende Element, das heißt auf den `<body>`, wenn es kein anderes umgebendes Blockelement gibt. `width: 50%` bedeutet also halb so breit wie das Elternelement.
  - Prozentangaben funktionieren nur zuverlässig, wenn für das umgebende Element eine Größe gesetzt ist
  - Prozentangaben für `padding` und `margin` beziehen sich auf die Größe des umgebenden Elements, nicht des Elements selbst.

- Relative Angaben in `em`
  - `em` sind und bleiben eine Einheit für Schriftgrößen, auch wenn man sie als Größenangaben für Boxen benutzt, haben sie einen Bezug zur Schriftgröße
  - Ein Element mit `height: 3em;` ist so hoch wie drei Zeilen Text von der Schriftgröße und Schriftart, die das Element selbst verwendet. Heißt aber nicht, dass auch drei Zeilen Text reinpassen. Zwischen den Textzeilen kommt noch der Zeilenabstand, der für die Größenangabe nicht berücksichtigt wird

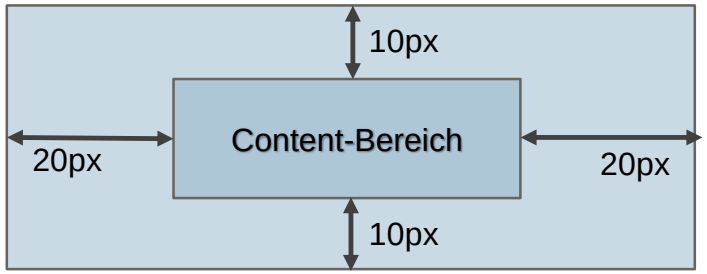
Ist keine Größe angegeben, wird ein Blockelement gerade so groß, dass sein Inhalt komplett hineinpasst

Das Gesetz des Kompasses

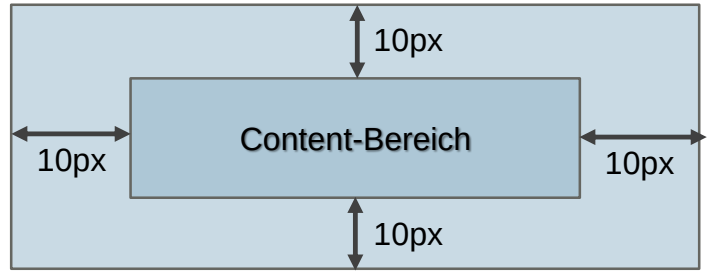
- von `margin` und `padding` werden die vier Werte auf die vier Himmelsrichtungen verteilt



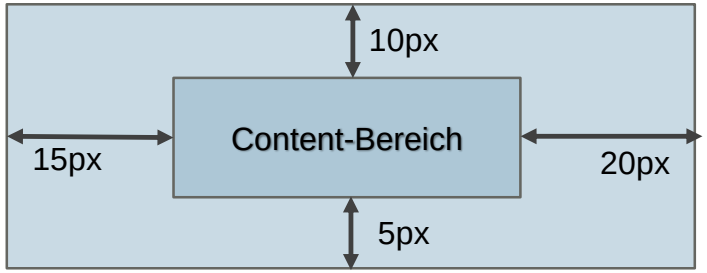

```
padding: 10px 20px 5px;
```




```
padding: 10px 20px;
```




```
padding: 10px;
```




```
padding: 10px 20px 5px 15px;
```

Abstände: margin Besonderheiten

CODERS.BAY

- Margin-Eigenschaften haben die Eigenheit, dass sie zusammenfallen (Collapsing Margins)
- Beispiel
  - wenn zwei Margins zusammentreffen wie `margin-bottom` und `margin-top` bei dem darunter liegenden Element, dann wird das größere Margin angewendet
- mit `margin: auto;` ist es möglich Blockelemente in einem anderen zu zentrieren
  `text-align: center;` wirkt sich nur auf Inline-Elemente aus

Overflow

- Standardeinstellung: `visible`
  - Inhalt läuft über den Rand des Elements hinaus und über das nächste Element, bis der Text zu Ende ist
- `overflow: hidden;`
  - Text läuft nicht mehr über den Rand hinaus sondern wird "versteckt"
- `overflow: scroll;`
  - ein horizontaler oder vertikaler Scrollbalken wird eingeblendet um den Text zu lesen
- `overflow: auto;`
  - zeigt den Scrollbalken dann, wenn er benötigt wird, sonst nicht
  - der Browser bietet bei überlaufenden Inhalt einen Scrollmechanismus an

CSS FÜR TEXT

CODERS.BAY

```
font-style: italic;
  /* underline, */
font-size: 12pt;
font-family: "Comic Sans", Arial, sans-serif;
font-weight: bold;
  /* normal, bold */
text-transform: uppercase;
  /* lowercase, capitalize, none */
font-variant: normal;
  /* small-caps */
text-decoration: underline;
  /* overline, line-through */
text-align: left;
  /* center, right, justify */
line-height: 0.7;
text-shadow: 0.2em 0.2em 5px gray;
  /* Rechtsverschiebung, Verschiebung nach unten, Unschärfe, Farbe für Schatte */A
```

CSS3

```
border-radius: 20px;
  /* Kompassregel */
border: 1px solid #000;
  /* Breite, Art, Farbe */
box-shadow: 10px 10px 5px #AAAACC;
  /* horizontale & vertikale Verschiebung, Unschärfe, Farbwert */
box-shadow: 10px 10px 5px #AAAACC inset;
  /* inset = Schatten nach innen */
```

CODERS.BAY

- `<footer>` enthält Informationen zum Inhalt der Seite, die schon genannten Copyright-Informationen, den Autor, einen Link zum Impressum und so weiter.
- Im `<header>` stehen einführende Informationen zum Seiteninhalt: eine kurze Beschreibung, was auf dieser Seite zu finden ist, ein Seitenlogo …
- Ein einzelner Eintrag in einem Blog oder einem Onlinemagazin kann mit `<article>` ausgezeichnet werden. Artikel können eigene `<footer>` und `<header>` haben. Und auch darin verschachtelte `<article>`, zum Beispiel für Kommentare zu einem Blogeintrag.
- Ein Artikel kann wiederum in `<section>`s unterteil werden. Sektionen sind Bereiche des Artikels, die eigenständig genug sind, dass sie eine eigene Überschrift haben können. Logischerweise können sie also auch wieder einen `<header>` und auch einen `<footer>` haben.
- In einem `<aside>`-Tag steht Inhalt, der zwar Bezug zur Seite (oder zum Artikel oder zur Sektion) hat, der aber entfernt werden könnte, ohne dass dem Hauptinhalt etwas fehlt.

- Beispiel:
  Eine Seite hat einen Themenbereich Blog, News und Kontakt. So sind die einzelnen Themenbereich einzelne `<section>`s. In der `<section>` Blog sind die einzelnen Blog Einträge `<article>`. In der `<section>` News sind die einzelnen Newsbeiträge `<article>`.

https://caniuse.com/

Z-INDEX

- festlegen, welches Element weiter oben liegt: Der höhere Wert gewinnt und liegt oben `z-index: 0;`

```
z-index: 0;
```

AUFGABE

CODERS.BAY

- **Aufgabe 1**
  - Schrödinger lernt HTML, CSS und JS
    Kapitel 7 durcharbeiten (Beispiele machen, ggf. in Aufgabe 6 einfließen lassen)
- **Aufgabe 2**
  Arbeite bei deiner Webseite zum Thema Reisen weiter
  Die Seiten sollen beinhalten:
  - Titel (steht dein Name)
  - Inline Elemente
  - Block Elemente
  - Block Elemente die sich wie Inline Elemente verhalten
  - Inline Elemente die sich wie Block Elemente verhalten
  - Inline-Block Elemente
  - Positioniere zwei Paragraphen nebeneinander
  - Füge Abstände hinzu
  - Benutze HTML5 Elemente für eine korrekte semantische Auszeichnung

  Benutze für das Styling nur noch dein externes Stylesheet.
  Kein Inlinestyling mehr!!

  Stylinganweisung die in **verschiedenen Kombinationen** verwendet werden müssen:
  - overflow
  - z-index
  - Arbeite die CSS Anweisungen von Folie 13 ein in verschiedenen Formen
  - Arbeite die CSS Anweisungen von Folie 14 ein in verschiedenen Formen
- **Aufgabe 3**
  - weiterführende Links lesen und in Aufgabe 6 umsetzen

ABGABEMODALITÄTEN

CODERS.BAY

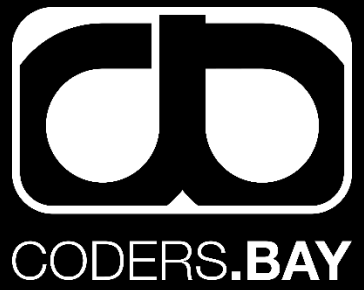
Abgabemodalität:

Erstelle einen Ordner mit deinem Namen

In diesem Ordner erstelle einen zweiten Ordner mit der Bezeichnung „Aufgabe 06“.
Die .css Datei wird in einem eigenen Ordner namens styling im abzugebenden Ordner gespeichert.
Die Bilder werden in einem eigenen Ordner namens img im abzugebenden Ordner gespeichert.

In dem Ordner „Aufgabe 06“ sind alle Datei, Ordner und alle Bilder die du benutzt, abgespeichert.
Die Aufgabe wird Freitags mit einem USB-Stick eingesammelt. Auf dem USB-Stick kopierst du den Ordner mit deinem Namen.

Überprüfe die Datei am Stick ob alle Verlinkungen ordnungsgemäß funktionieren!



Ende