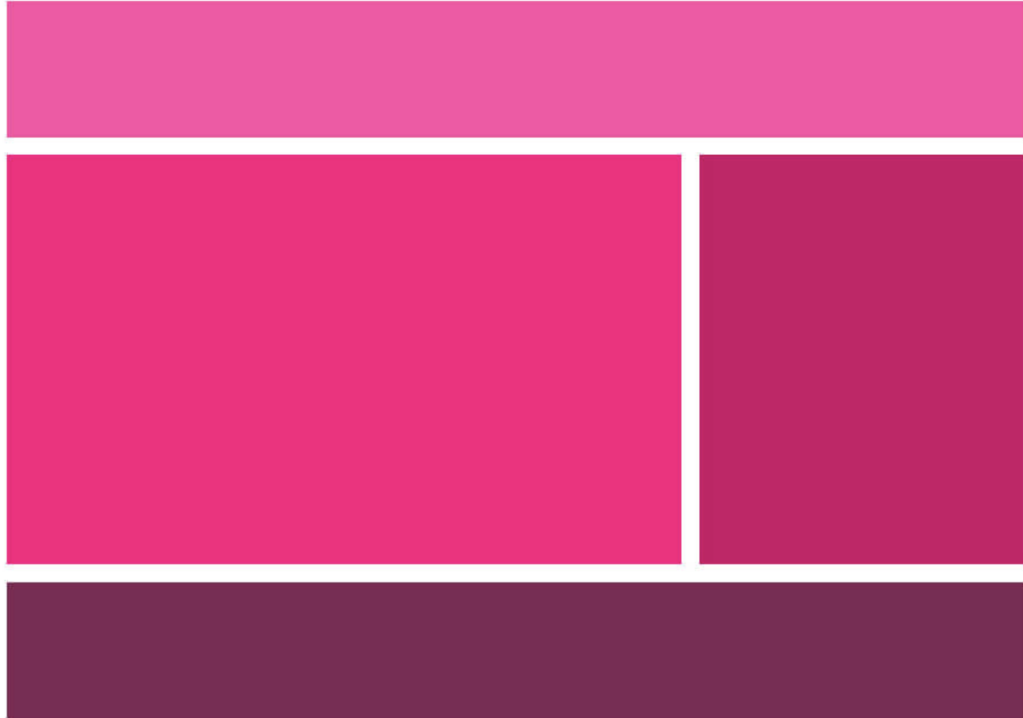
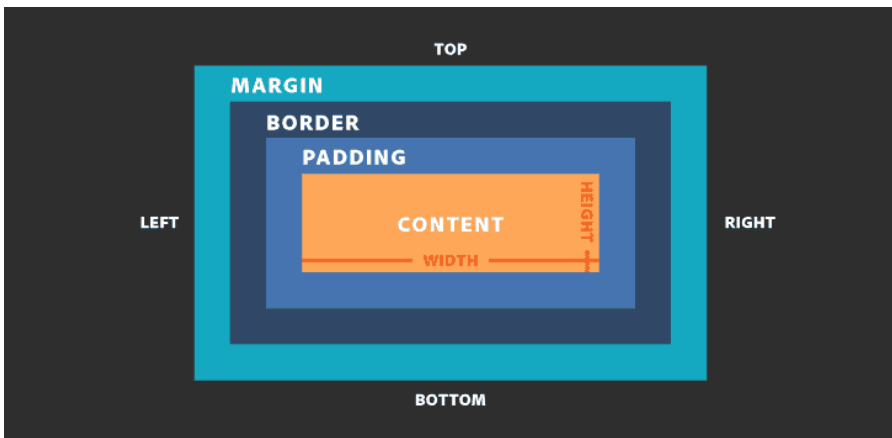


CSS BASICS - Box Layout Context



Box Model



```
header {  
  /* BOX-SIZING */
```

```
  box-sizing: content-box;  
  box-sizing: border-box;
```

Ist default value
totale box breite =
width + padding + border

Oft ist dieser value vorteilhaft
totale box breite =
width - (padding + border)

```
  /* CONTENT */  
  width: 200px;  
  height: 100px;
```

```
  /* PADDING */  
  padding-top: 10px;  
  padding-left: 10px;  
  padding-bottom: 10px;  
  padding-right: 10px;  
  padding: 10px 20px;  
  padding: 10px;
```

```
          /* top/bottom, left/right */  
  /* all sides */
```

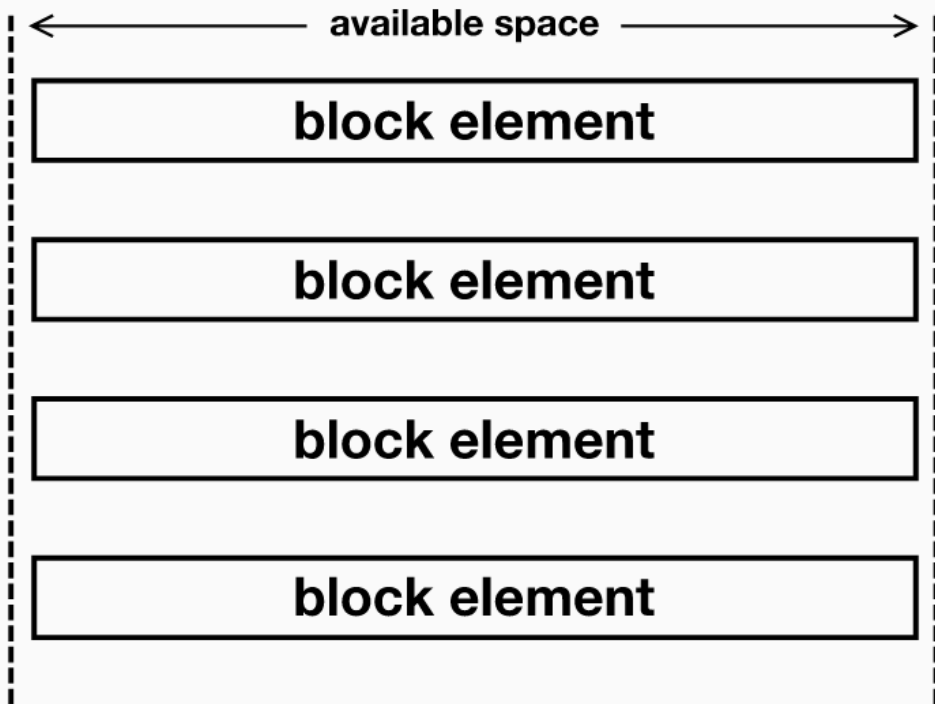
```
  /* BORDER */  
  border: 1px solid black; /* all sides */
```

```
  /* MARGIN */  
  margin: 10px;
```

```
          /* all sides
```

```
}
```

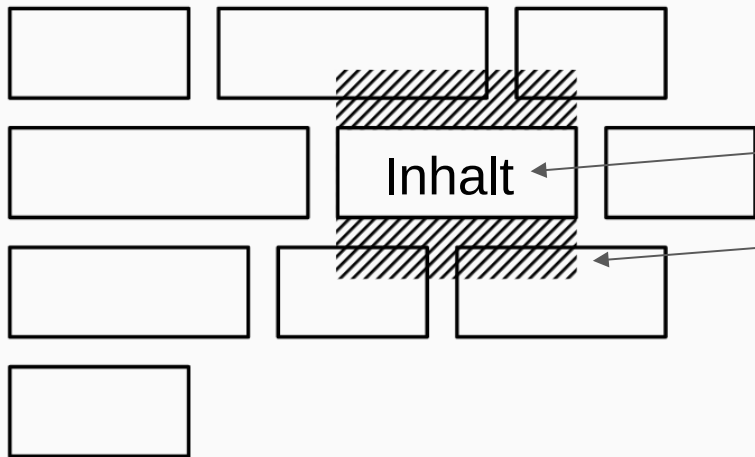
Block Element



```
header {  
    /* DISPLAY */  
    display: block;  
}
```

Inline Element

inline



```
header {  
  /* DISPLAY */  
  display: inline;
```

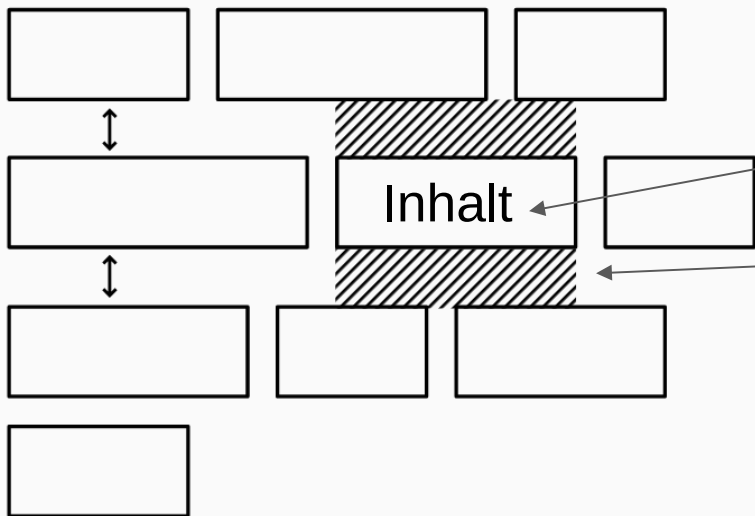
```
  /* CONTENT */  
  width: 100px;  
  height: 50px;
```

```
  /* MARGIN TOP/BOTTOM */  
  margin-top: 10px;  
  margin-bottom: 10px;
```

```
}
```

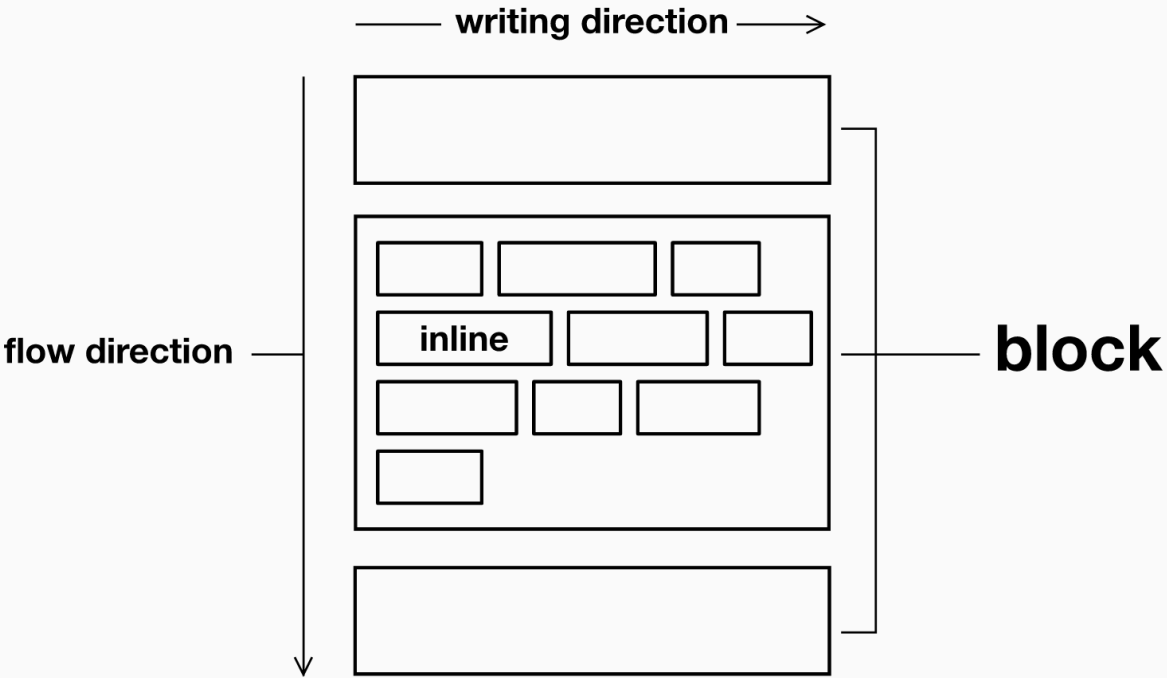
Inline Block Element

inline-block



```
header {  
  /* DISPLAY */  
  display: inline-block;  
  
  /* CONTENT */  
  width: 100px;  
  height: 50px;  
  
  /* MARGIN TOP/BOTTOM */  
  margin-top: 10px;  
  margin-bottom: 10px;  
}
```

Block Element vs Inline Element



Block Element vs Inline Element

BLOCK:



INLINE:



Block

- Beginnt immer mit einer **neuen Zeile**
- Der Parent bestimmt die Breite (Nimmt die ganz verfügbare Breite ein).
- `<main>`, `<nav>`, `<header>`, `<footer>`, `<aside>`, `<article>`, `<section>`, `<div>`, `<p>`, `<h1..h6>`

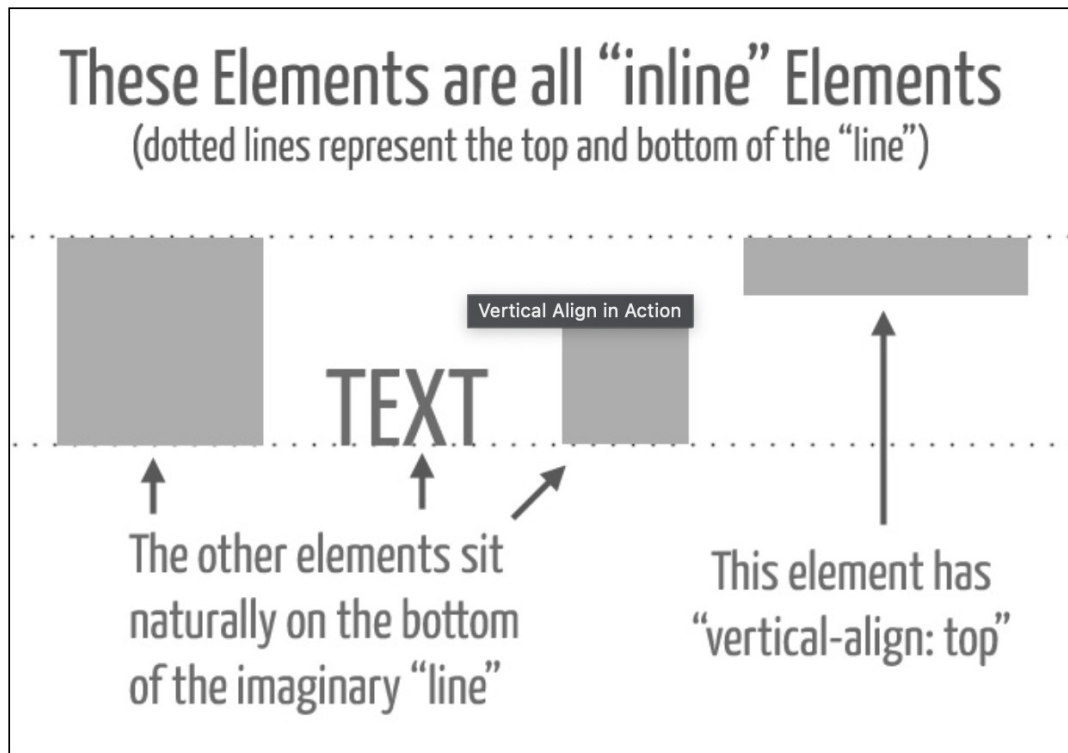
Inline

- Beginnt nicht mit einer neuen Zeile, sondern wird in eine **bestehende Zeile** eingebettet.
- Der Inhalt bestimmt die Breite (Nimmt nur was notwendig ist).
- `<a>`, ``, ``, `<label>`

Block Element vs Inline Element

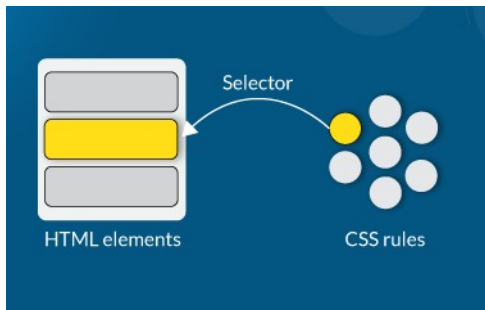
Behaviour	Inline	Inline-Block	Block
Respects left/right padding	YES	YES	YES
Respects top/bottom padding	YES	YES	YES
Respects left/right margin	YES	YES	YES
Respects top/bottom margin	NO	YES	YES
Respects width/height	NO	YES	YES
Default width expands to parent element (not width of its contents)	NO	NO	YES
Forces a line break	NO	NO	YES

The Line Box



```
img {  
  /* VERTICAL ALIGN */  
  vertical-align: top;  
  vertical-align: middle;  
  vertical-align: bottom;  
}
```

2.Common Selectors



```
<header id="main-header">
  <h1 class="light">..</h1>
</header>
<main>
  <article>
    <header>
      <h2 class="light">..</h2>
    </header>
  </article>
</main>
```

Tag-Selector

Auswahl **ALLER** Elemente über ihren HTML-Tag.

```
header { .. }
```

Id-Selector

Auswahl **EINES** Elements über sein HTML-ID Attribut.

```
#main-header { .. }
```

Class-Selector

Auswahl **ALLER** Elements über ihre HTML-Class Attribut.

```
.light { .. }
```

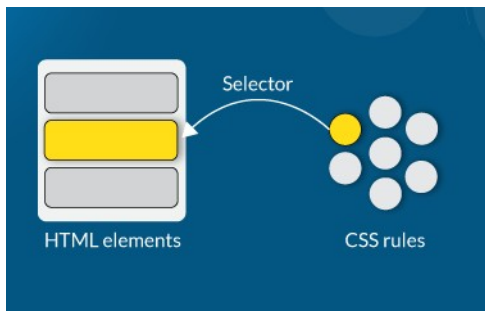
Nachkommens-Selector

Auswahl **ALLER** Elemente die untergeordnete Elemente des Elements sind.

```
main header { .. }
```

Deep Dive Selectors: https://developer.mozilla.org/de/docs/Web/CSS/CSS_Selectors

Common Selectors



```
<nav>
  <button>Menü</button>
  <ul>
    <li><a href="#"></a></li>
    <li><a href="#"></a></li>
  </ul>
</nav>
```

Pseudo-Selector :hover

Wird nur aktive, wenn der User die Mouse über das Element bewegt. Achtung: Geht nicht in Mobile.

```
nav li:hover {
  background-color: steelblue;
}
```

Pseudo-Element ::before / ::after

Fügt ein Text oder Bild vor / nach einem Element hinzu.

```
nav button::before {
  // Fügt das Wort Test vor/nach dem Button ein
  content: "Test"
  // Fügt ein SVG-Image vor/nach dem Button ein
  content: url("data:image/svg+xml, ...");
}
```

Units in CSS

px

Absolute pixel value

%

A percentage of the parent element.
100% is the width of the parent element

em

Relative to the font size of the element

vh

Relative to 1% of the viewport's height

rem

Relative to the font size of
the root element

vw

Relative to 1% of the viewport's width

Deep Dive Units: <https://smazee.com/blog/css-units-px-em-rem-vh-vw-vmin-vmax>

Common Properties

Box Model

box-sizing: content-box | border-box

width: 50% | 100px

height: 75% | 100px

padding: 10px // alle seiten 10px

padding: 10px 5px // oben/unten 10px und links/rechts 5px

padding-left/right/top/bottom: 10px // nur eine Seite 10px

border: 1px solid black

margin: 10px // alle seiten 10px

margin: 10px 5px // oben/unten 10px und links/rechts 5px

margin-left/right/top/bottom: 10px // nur eine Seite 10px

content-box:

Ist der default value

totale box breite =
width + padding + border

border-box:

Oft ist dieser value vorteilhaft

totale box breite =
width - (padding + border)

Common Properties

Font / Text

font-family: 'Sans Source Pro', sans-serif

font-style: normal | italic

font-weight: normal | bold | 100 - 900

font-size: 1.5rem

line-height: 1.5rem

color: blue | #333

text-decoration: none | underline

text-transform: lowercase | uppercase

Color

color: blue | #333

background-color: steelblue | #16416B;

Box Shadow

box-shadow: 2px 3px 2px silver;

Box Shadow Generator:

<https://www.cssmatic.com/box-shadow>

Display / Position

display: block | inline | inline-**block** | flex | none

position: static | relative | absolute | sticky | fixed

// Gehen nur, wenn nicht position: static

top: 10px

bottom: 10px;

left: 5px;

right: 5px

Visibility

// Ist das Element sichtbar?

visibility: visible | hidden

Overflow

// Was passiert mit Inhalt, der größer als Parent ist?

overflow: visible | hidden | scroll

Common Properties

Alignment

// Ich Inline-Child, richtige mich/andere in der Line-Box aus.

// Für derartige Alignments wird Flexbox empfohlen (kommt später)

vertical-align: top | middle | bottom

// Ich Parent-Block, richtige den Text meiner Inline-Childs aus.

text-align: left | center | right

// Ich Child-Block, zentriere mich horiz. im Parent-Block

margin: 0 auto;

Transform

// auf 80% verkleinern

transform: scale(80%)

// Verschiebe Element 10px nach rechts und 5 px nach unten

transform: translate(10px, 5px)

Common Properties

Entfernen eines Elements

// Variante 1:

Entfernt das Element aus dem normalen Fluss so dass andere Elemente es ausfüllen können.

display: none

// Variante 2:

// Lässt das Element im normalen Fluss, so dass es weiterhin Platz einnimmt.

visibility: hidden

// Variante 3:

// Keine Höhe und versteckt den Inhalt

max-height: 0

overflow: hidden;

Common Properties

Transition

// Animiere alle css properties dieses Elements

// um 0.5s mit der ease-transition function

transition: all 0.5s ease;

// Animiere das height css property dieses Elements

// um 0.5s mit der ease-transition function

transition: height 0.5s ease;

<https://css-tricks.com/almanac/properties/t/transition/>