

AGENDA

Curriculum WDD

4100_07	REPETITION / SPRACHEN / STATISCHE WEBSITE / SYNTAX / GRUNDGERÜST / TAGS AUFGABE / QUELLCODE und DOM / WEB INSPECTOR / SEMANTIK
	CSS REGEL / CSS EINBINDEN / CSS RESET / CSS MEDIA TYPES / CSS SELEKTOREN + SPEZIFITÄT / MASSEINHEITEN
4100_08	FARBCODES / INLINE- UND BLOCKELEMENTE / FLOAT
	POSITIONIERUNG / BOX-MODEL / FLEX / EINMITTEN
4100_09	NAVIGATION / FARBVERLÄUFE / FONTS / TEXT FORMATING / ICON FONT
	BILDER / HINTERGRUND / CSS SPRITES / FAVICON
4100_10	HAUSAUFGABEN / EINMITTEN / TABELLEN / IFRAME / FORMULAR
	VIDEO/AUDIO / RETINA / BARRIEREFREIHEIT
4100_11	HAUSAUFGABEN / REKAPITULIEREN CSS SELEKTOREN / SUBNAVIGATION / ZWISCHENTEST
	REKAPITULIEREN BARRIEREFREIHEIT / HELPER TOOLS / WORKFLOW / SCREENDSIGN ÜBERSETZEN / WISSENSQUIZ

POSITIONIERUNG

CSS EIGENSCHAFT POSITION

Wonach wird ein Element positioniert?

position: static;

- **default-Wert!**
- im normalen HTML-Fluss
- CSS top, right, bottom, left reagieren nicht, nur margin!

position: relative;

- Verschiebung, gemessen an der Normalposition oder Anfangsposition des Elements selbst.
- CSS top, right, bottom, left werden aktiviert.
- Aktiviert z-Achse des Elements.

position: absolute;

- Positionierung, gemessen am Rand des nächsthöheren Vorfahrenelements, also umspannende Element, das die Position relative oder absolute aufweist.
- Das Element wird aus dem HTML-Tag-Fluss ausgelöst.
- Letzter Fallback der Ausrichtung (sozusagen das letzte Eltern-Element) wäre: <body>
- CSS top, left, right, bottom werden aktiviert.
- Aktiviert z-Achse des Elements.

position: fixed;

- Positionierung, gemessen am "Viewport", d.h. am Browserfenster.
- Bleibt beim Scrollen stehen.
- Das Element wird aus dem HTML-Tag-Fluss ausgelöst.
- CSS top, left, right, bottom werden aktiviert.
- Aktiviert z-Achse des Elements.

„nicht positioniert“

„positioniert“

CSS POSITION STATIC

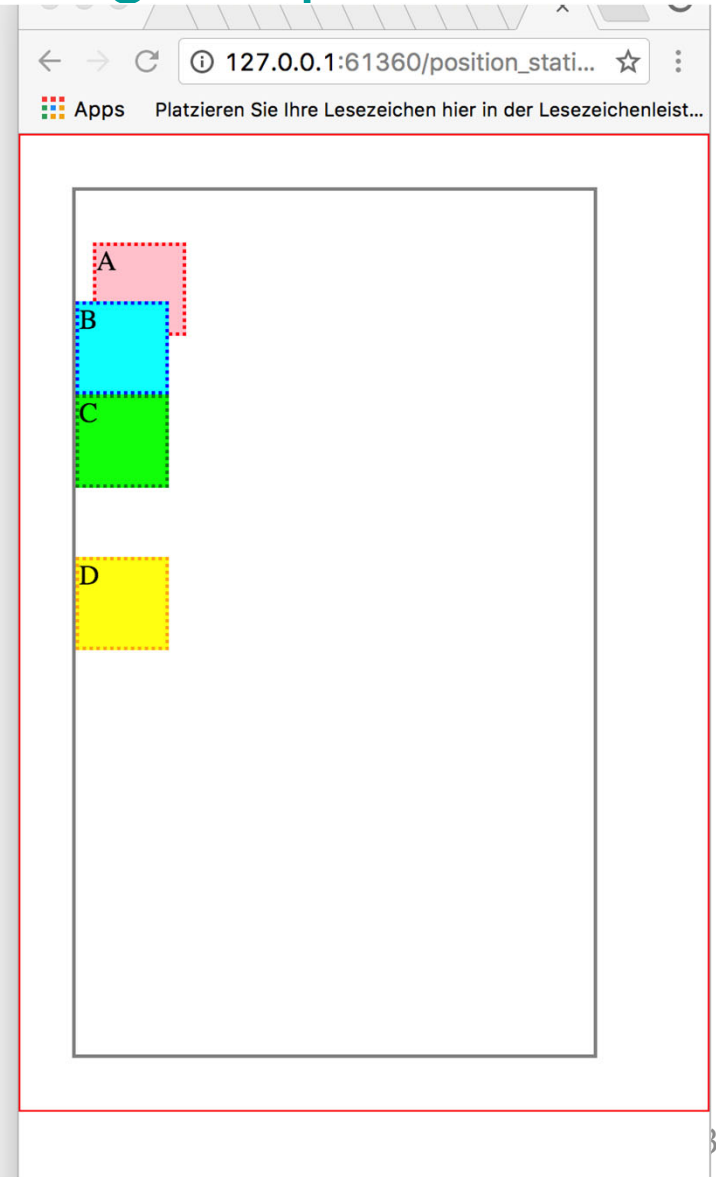
Erlaubte CSS Eigenschaften top/margin-top/z-index

Zum Spicken

position: static;

- **default**
- **margin +/-**
 - z.B. margin-top: 10px; => Element verschiebt sich und nachfolgende Elemente nach unten.

```
1 <!DOCTYPE html>
2 <html lang="de">
3 <head>
4   <meta charset="utf-8">
5   <title>Position Static => margin ok</title>
6   <style>
7     * { margin: 0; padding: 0; }
8   }
9   body { border: 1px solid red; }
10  div#wrapper { border: 2px solid grey; width: 300px; height: 500px; margin: 30px; }
11
12  div#wrapper > div { width: 50px; height: 50px; }
13
14  div#wrapper div#A { border: 2px dotted red; background: pink; margin-top: 30px; margin-left: 10px; }
15
16  div#wrapper div#B { border: 2px dotted blue; background: cyan; margin-top: -20px; }
17
18  div#wrapper div#C { border: 2px dotted green; background: lime; margin-top: 0px; }
19
20  div#wrapper div#D { border: 2px dotted orange; background: yellow; margin-top: 40px; }
21
22  </style>
23 </head>
24 <body>
25   <div id="wrapper">
26     <div id="A">A</div>
27     <div id="B">B</div>
28     <div id="C">C</div>
29     <div id="D">D</div>
30   </div>
31 </body>
32 </html>
```



CSS POSITION ABSOLUTE

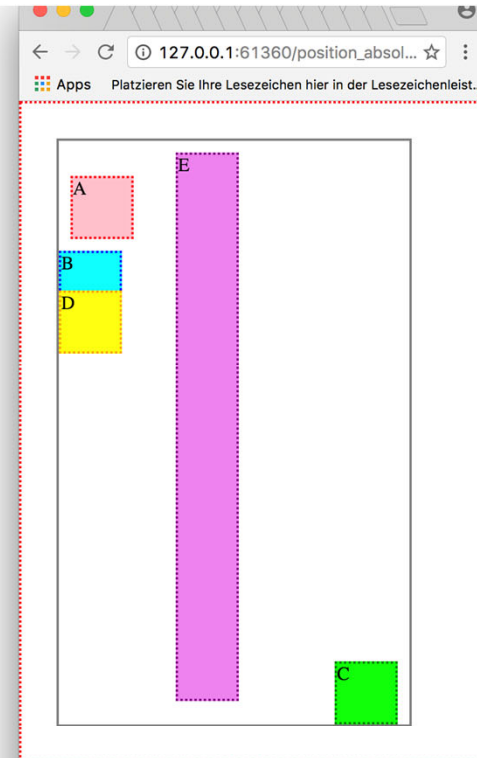
Zum Spicken

Erlaubte CSS Eigenschaften top/margin-top/z-index

position: absolute;

- **margin +/-**
 - z.B. margin-top: 10px; => Element verschiebt sich und nachfolgende Elemente nach unten.
- **top | bottom | left | right +/-**
 - z.B. top: 10px; => Element verschiebt nur sich nach unten.
- **z-index +/-**
 - z.B. ich kann damit die Elemente anordnen auf der z-Achse. Default ist: auto.
- **height/width: auto; und top/bottom/left/right**
 - ich kann das Element zum Rand des nächstpositionierten Elternelements aufziehen.

```
1 <!DOCTYPE html>
2 <html lang="de">
3 <head>
4   <meta charset="utf-8">
5   <title>Position Absolute mit den Elementen C und E </title>
6   <style>
7     * {
8       margin: 0; padding: 0;
9     }
10    body {
11      border: 2px red dotted;
12    }
13    div#wrapper {
14      border: 2px solid grey;
15      width: 300px;
16      height: 500px;
17      margin: 30px;
18      position: relative; /* Dieses Elternelement ist NICHT static */
19    }
20    div#wrapper > div {
21      width: 50px;
22      height: 50px;
23    }
24    div#wrapper div#A { ... }
25    div#wrapper div#B { ... }
26    div#wrapper div#C {
27      border: 2px dotted green; /* BEISPIEL MIT ABSOLUTE - am Elternrand kleben */
28      background: lime;
29      position: absolute; /* Distanzen in dieser Regel richten sich nach dem nächsthöheren
30                           Element, welches relative, absolute, fixed, und falls jenes
31                           Elternelement nicht vorhanden ist, dann body-Tag. Hier ist es
32                           der wrapper, der positioniert wurde. */
33    }
34    div#wrapper div#D {
35      margin-left: 0px;
36      margin-top: 0px;
37      margin-bottom: 0px;
38      right: 10px;
39      bottom: 0px;
40      z-index: 1000; /* z-Achse möglich! */
41    }
42    div#wrapper div#E {
43      border: 2px dotted purple; /* BEISPIEL MIT ABSOLUTE - aufgezogen */
44      position: absolute; /* Mit absolute und top/bottom und height:auto kann ich
45                           ein DIV auch strecken zum Rand des Elternelements! */
46      background: violet;
47      margin-top: 0px;
48      margin-bottom: 0px;
49      top: 10px;
50      bottom: 20px;
51      left: 100px;
52      height: auto;
53    }
54  </style>
55 </head>
56 <body>
57   <div id="wrapper">
58     <div id="A">A</div>
59     <div id="B">B</div>
60     <div id="C">C</div>
61     <div id="D">D</div>
62     <div id="E">E</div>
63   </div>
64 </body>
65 </html>
```



CSS POSITION RELATIVE

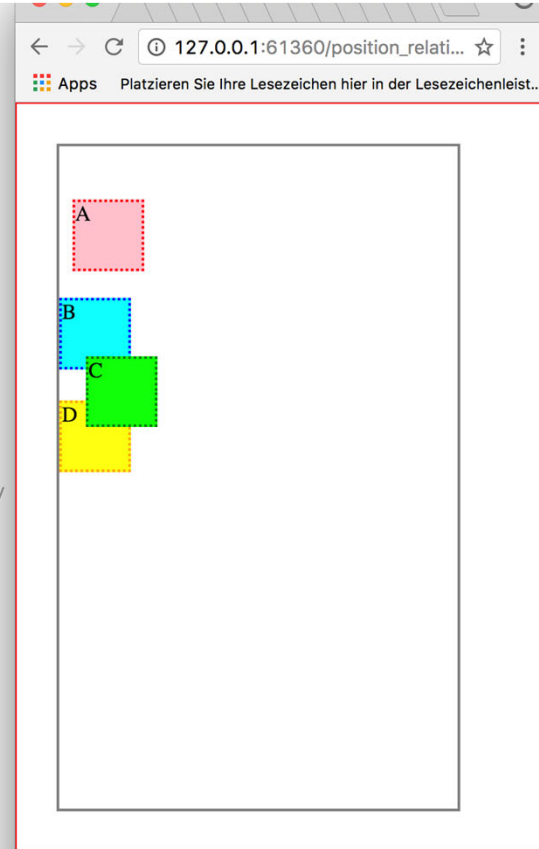
Zum Spicken

Erlaubte CSS Eigenschaften top/margin-top/z-index

position: relative;

- margin +/-
 - z.B. margin-top: 10px; => Element verschiebt sich und nachfolgende Elemente nach unten.
- top | bottom | left | right +/-
 - z.B. top: 10px; => Element verschiebt nur sich nach unten.
- z-index +/-
 - z.B. ich kann damit die Elemente anordnen auf der z-Achse. Default ist: auto.

```
1 <!DOCTYPE html>
2 <html lang="de">
3 <head>
4   <meta charset="utf-8">
5   <title>Position Relative</title>
6   <style>
7     * {
8       margin: 0; padding: 0;
9     }
10    body {
11      border: 1px red solid;
12    }
13    div#wrapper {
14      border: 2px solid grey;
15      width: 300px;
16      height: 500px;
17      margin: 30px;
18    }
19    div#wrapper > div {
20      width: 50px;
21      height: 50px;
22      position: relative; /* RELATIVE für alle DIV-Kinder */
23    }
24    div#wrapper div#A {
25      border: 2px dotted red;
26      background: pink;
27      margin-left: 10px; /* margin => verschiebt sich */
28      margin-top: 30px; /* und nachfolgende Elemente */
29      margin-bottom: 0px;
30      top: 10px; /* top/bottom/left/right => Verschiebung */
31      bottom: 0px; /* des Elements selber. Nachfolgende
32                  Elemente bleiben unbeeindruckt. */
33    }
34    div#wrapper div#B {
35      border: 2px dotted blue;
36      background: cyan;
37      margin-top: 10px;
38      margin-bottom: 0px;
39      top: 20px;
40      bottom: 0px;
41    }
42    div#wrapper div#C {
43      border: 2px dotted green;
44      background: lime;
45      margin-left: 20px;
46      top: 10px;
47      bottom: 0px;
48      z-index: 1000; /* z-Achse möglich! */
49    }
50    div#wrapper div#D {
51      border: 2px dotted orange;
52      background: yellow;
53      margin-top: -20px;
54      top: 10px;
55    }
56  </style>
57 </head>
58 <body>
59   <div id="wrapper">
60     <div id="A">A</div>
61     <div id="B">B</div>
62     <div id="C">C</div>
63     <div id="D">D</div>
64   </div>
65 </body>
66 </html>
```



CSS POSITION FIXED

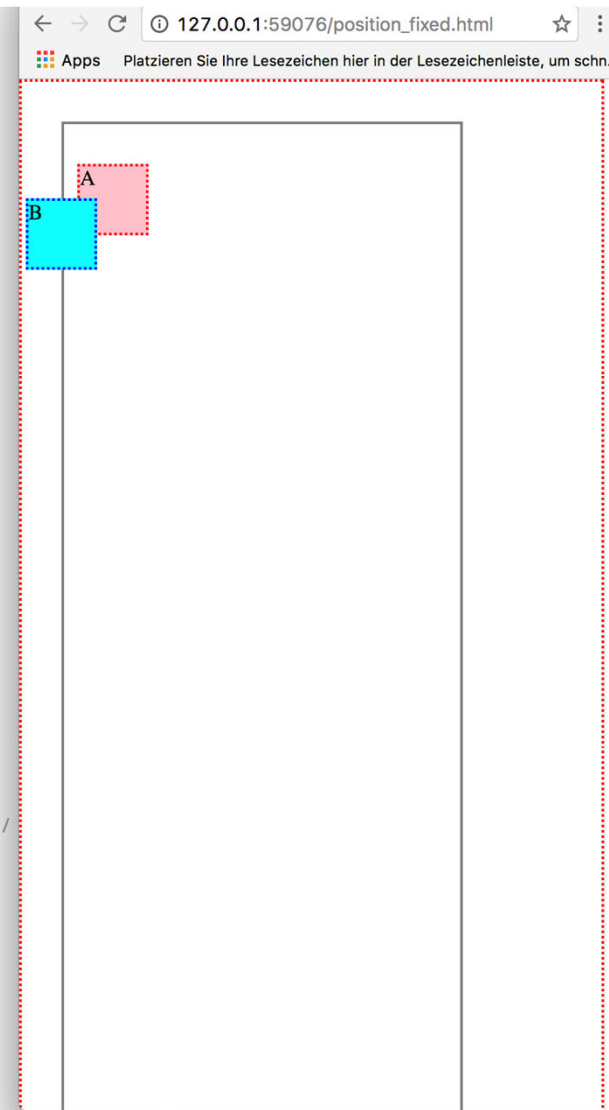
Zum Spicken

Erlaubte CSS Eigenschaften top/margin-top/z-index

position: fixed;

- **margin +/-**
 - z.B. margin-top: 10px; => Element verschiebt sich und nachfolgende Elemente nach unten.
- **top | bottom | left | right +/-**
 - z.B. top: 10px; => Element verschiebt nur sich nach unten.
- **z-index +/-**
 - z.B. kann ich damit die Elemente anordnen auf der z-Achse.
Default ist: auto.
- **height/width: auto; und top/bottom/left/right**
 - z.B. kann ich damit das Element zum Rand des nächstpositionierten Elternelements aufziehen.

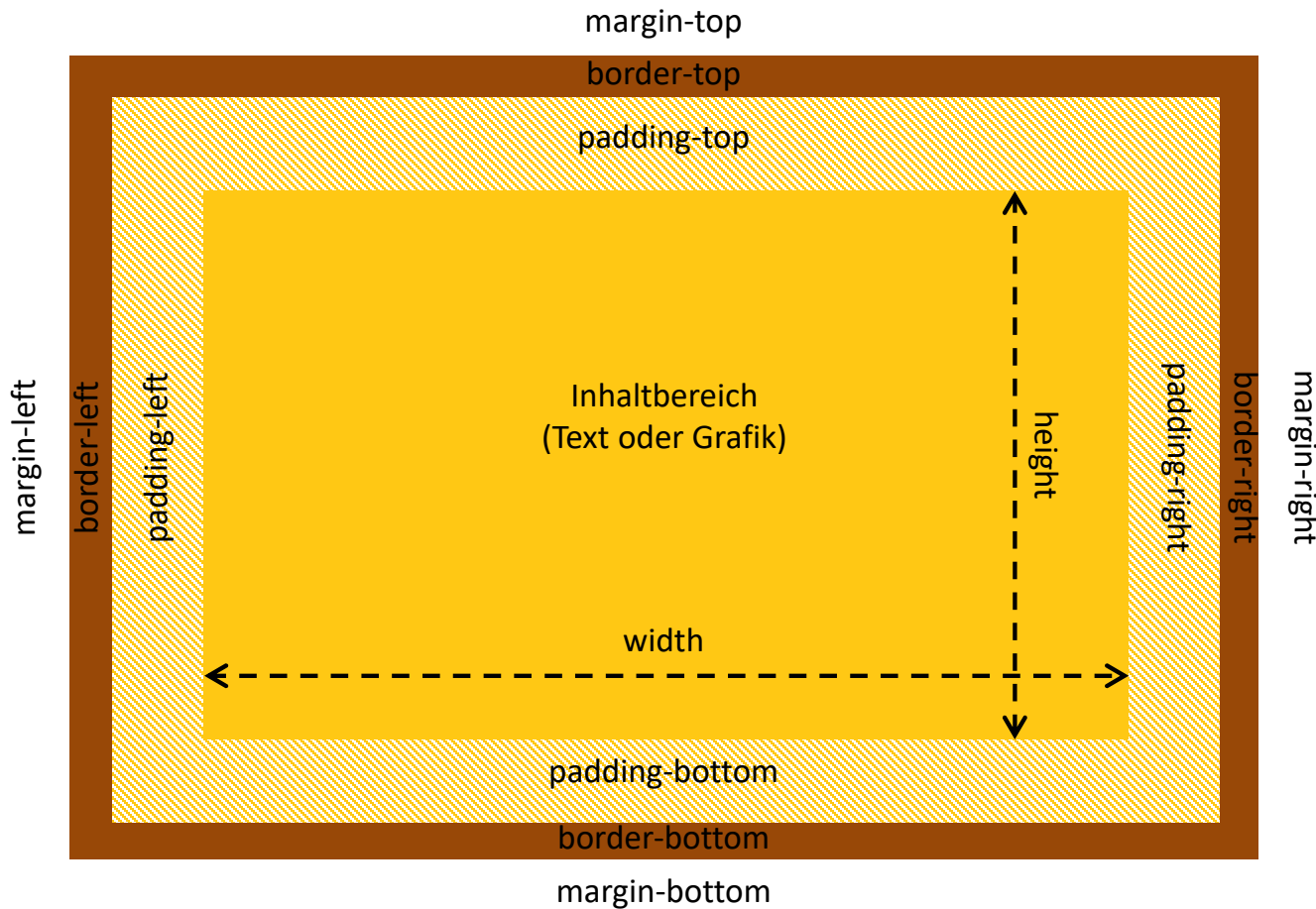
```
1 <!DOCTYPE html>
2 <html lang="de">
3 <head>
4   <meta charset="utf-8">
5   <title>Position FIXED - ab dem Viewport gemessen - siehe DIV B</title>
6   <style>
7     * {
8       margin: 0; padding: 0;
9     }
10    body {
11      border: 2px red dotted;
12    }
13    div#wrapper {
14      border: 2px solid grey;
15      width: 300px;
16      height: 450px;
17      margin: 30px;
18    }
19    div#wrapper > div {
20      width: 50px;
21      height: 50px;
22    }
23    div#wrapper div#A {
24      border: 2px dotted red;
25      background: pink;
26      margin-left: 10px;
27      margin-top: 30px;
28      margin-bottom: 0px;
29      top: 10px;
30      bottom: 0px;
31      /* z-Anordnung: DIV A wird dann über DIV B gelegt: */
32      /* position: relative;
33       z-index: 100000; */
34    }
35    div#wrapper div#B {
36      border: 2px dotted blue;
37      background: cyan;
38      margin-top: 10px;
39      margin-bottom: 0px;
40      /* Scroll nach unten im Browser für den Effekt */
41      position: fixed; /* fixed + top/left/bottom/right */
42      top: 80px;
43      left: 5px;
44      z-index: auto; /* default, könnt auch Nummer vergeben */
45      /* auch hier könnt ich die Höhe wieder max. ausdehnen: */
46      /*bottom: 20px;
47      height: auto; */
48    }
49  }
50 </style>
51 </head>
52 <body>
53   <div id="wrapper">
54     <div id="A">A</div>
55     <div id="B">B</div>
56   </div>
57 </body>
58 </html>
```



BOX-MODEL

BOX-SIZING

CONTENT-BOX VS. BORDER-BOX

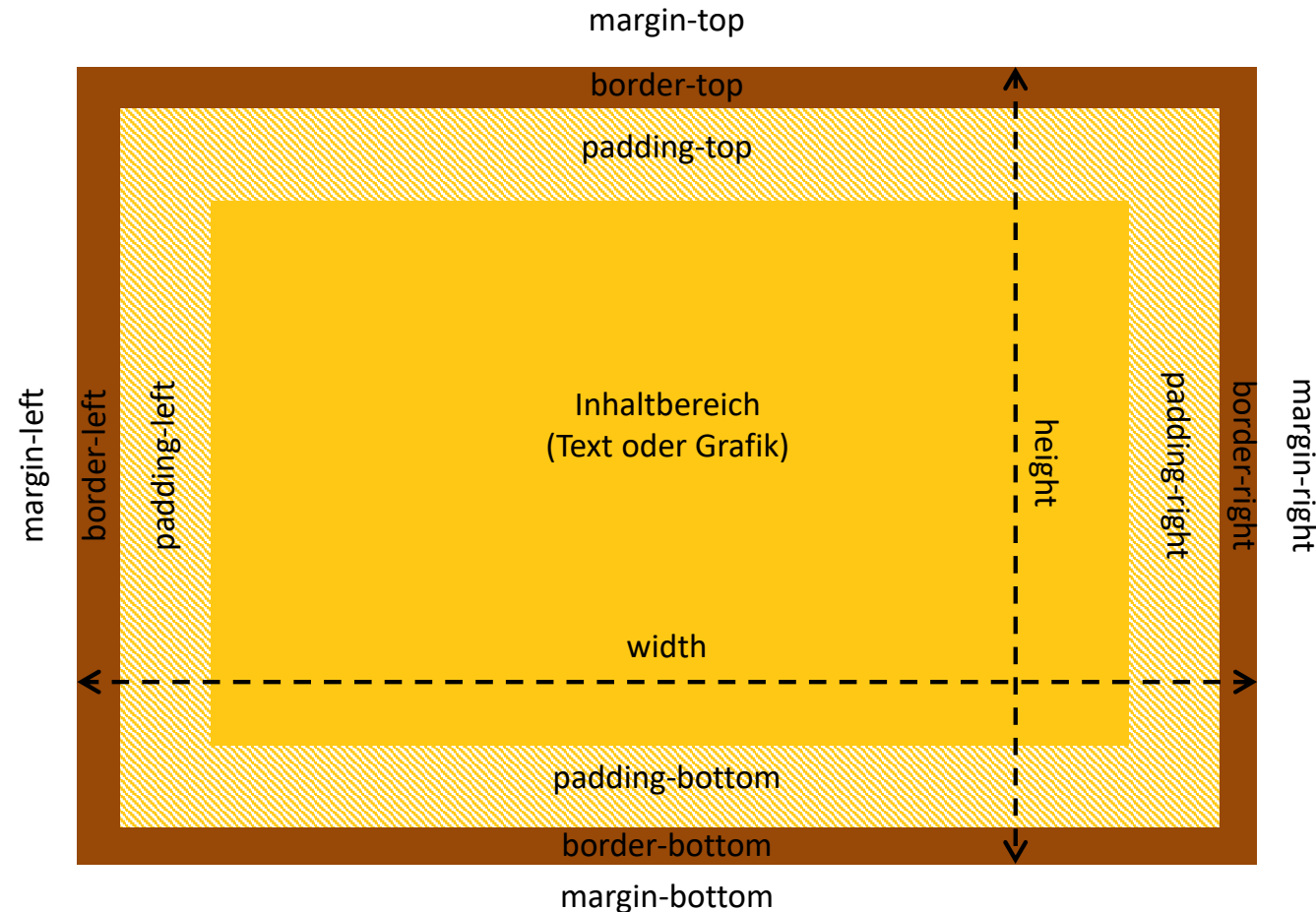


Eigenschaften

- im CSS:
box-sizing: content-box;
- **Standard-Wert (default)**
- **wird nicht vererbt**

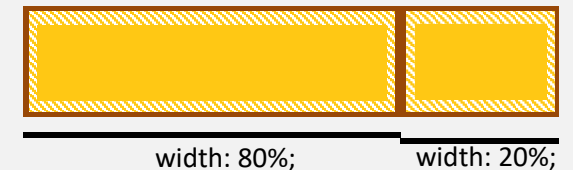
BOX-SIZING

CONTENT-BOX VS. BORDER-BOX



Eigenschaften

- im CSS:
box-sizing: border-box;
- wird nicht vererbt
- mehrere Boxen mit dieser CSS-Eigenschaft lassen sich in Höhe und Breite fehlerfrei mit **%-Angaben** bestimmen:



- Der Internet Explorer hatte mit diesem Ansatz eine Vorreiterrolle und wollte dies ohne spezielle CSS-Anweisung als Standard setzen.

BOX-SIZING: BORDER-BOX; GO FOR IT!

Fallback für aus vorgefertigten Elemente/Features mit content-box, die von externen Frameworks stammen

Einfache Notation im CSS

* = diese Notation bezieht sich auf alle HTML-Elemente – rechenintensiv!

```
* {  
  box-sizing: border-box;  
}
```

Bessere, flexiblere Notation

```
body {  
  box-sizing: border-box;  
  /* box-sizing wird eigentlich nicht vererbt, aber : */  
}  
  
*, *:before, *:after {  
  box-sizing: inherit;  
}
```

Erkläre warum!

Die flexiblere Notation erlaubt es, für ein spezifisches Element xyz sowie dessen Nachfahren das Box-Model z.B. auf content-box umzustellen. Die Geschwister-Elemente von xyz sind davon nicht betroffen.

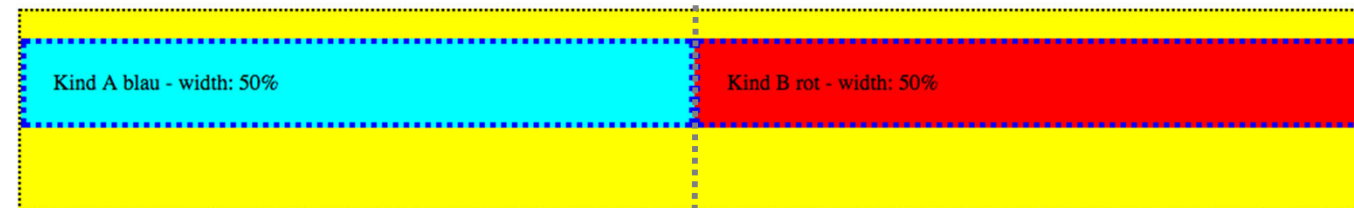
ÜBUNG BOX-SIZING / VERERBUNG

Siehe auch Demo-Datei.

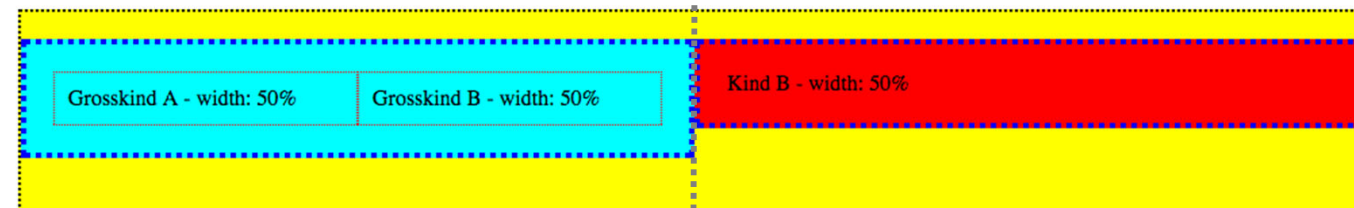
Welches Box-Model
wurde angewandt?

fifty-fifty...

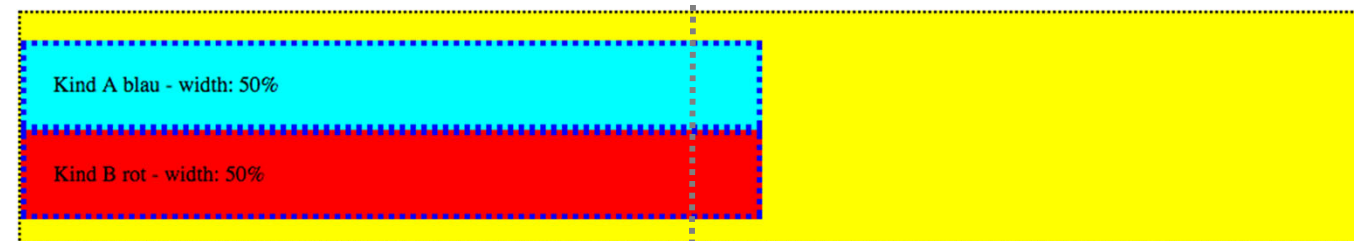
Beispiel 1: - die Mama hat Zwillinge mit Padding+Border bekommen:



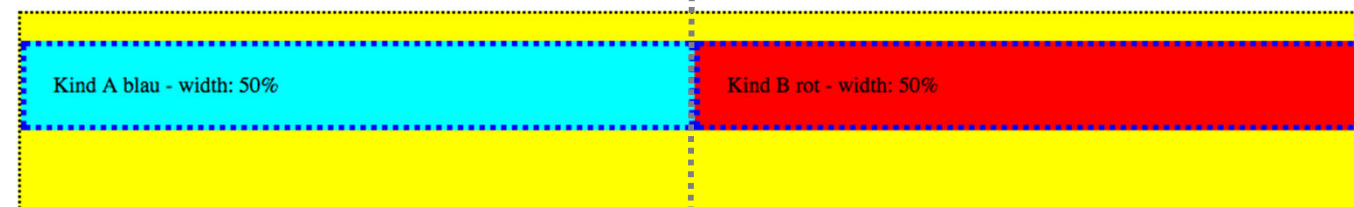
Beispiel 2: - die Grosskinder haben auch padding+border



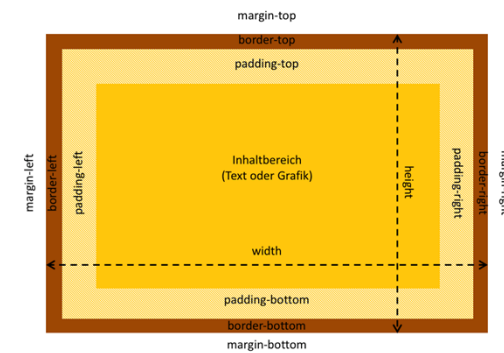
Beispiel 3: - wir heben mit der CSS-Klasse .contentBox das Border-Box-Modell auf.
Box-sizing: content-box wird auch auf die Kinder vererbt.
Wir sehen: mit width:50% haben die Zwillinge horizontal nebeneinander nicht mehr Platz.



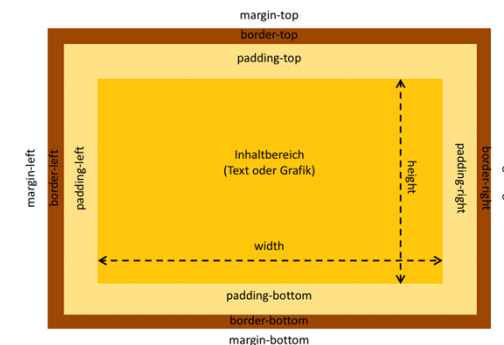
Beispiel 4: - ich funktioniere nach der "alten" Vererbung (border-box):



Border-box (1)



Content-box (2)



FLEX

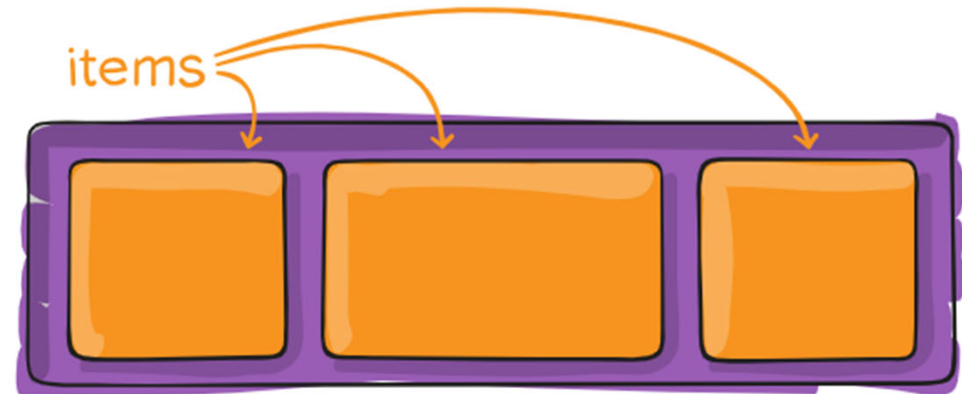
Easy way of creating layouts

container



**Properties for the Parent
(flex container)**

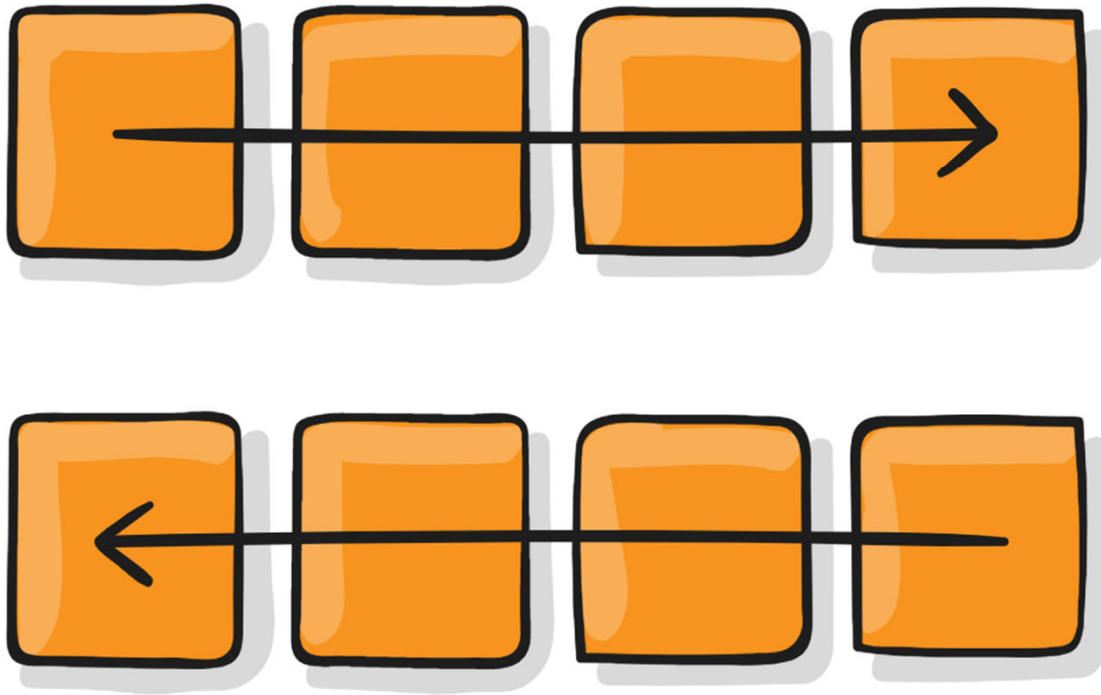
items



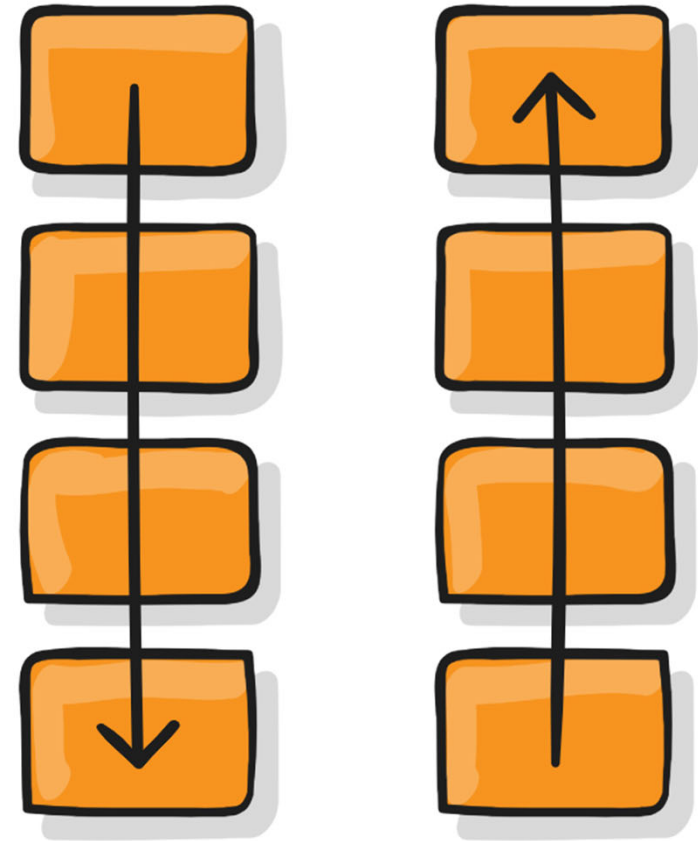
**Properties for the Children
(flex items)**

FLEX

flex-direction



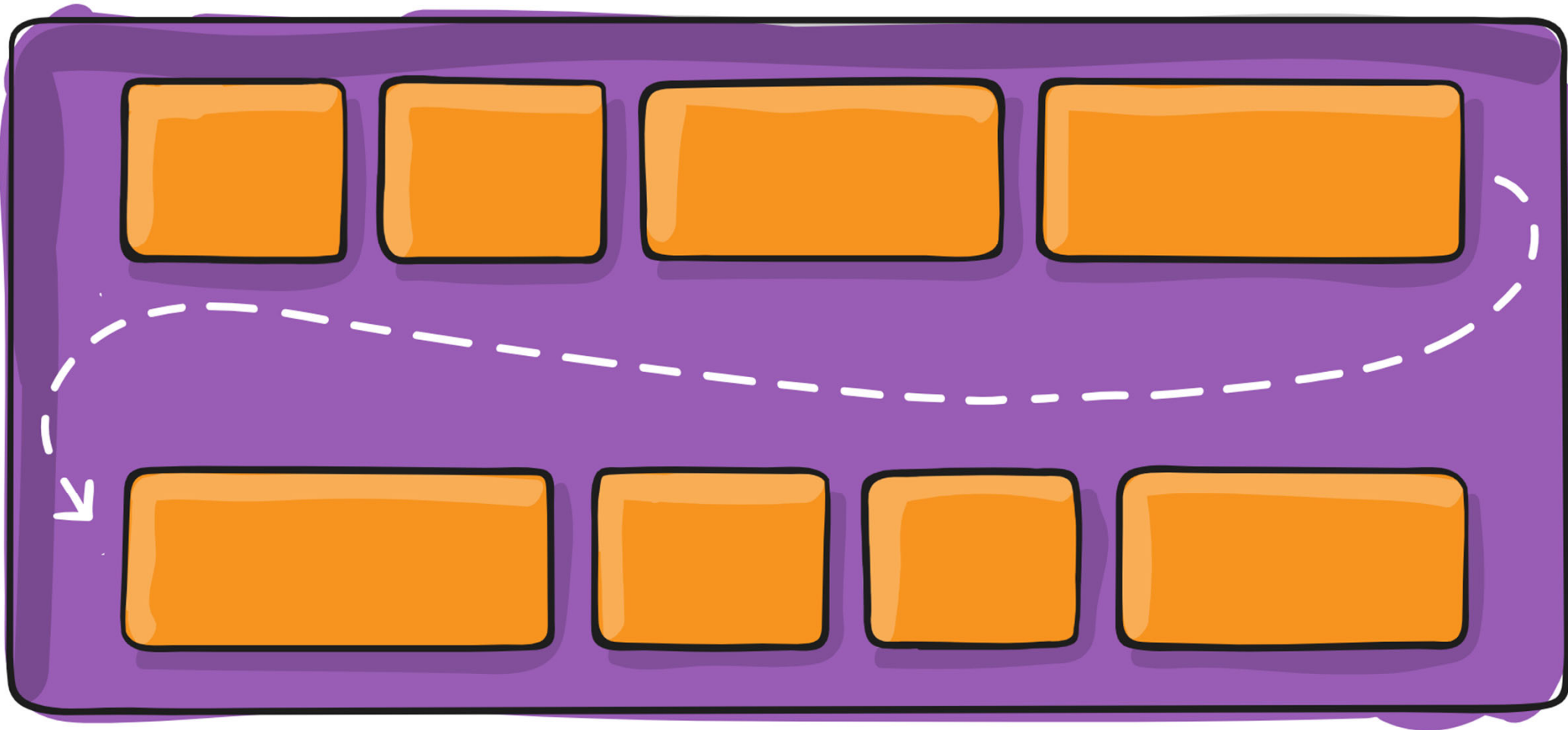
row (default)



column

FLEX

flex-wrap



FLEX

justify-content

flex-start



flex-end



center



space-between



space-around



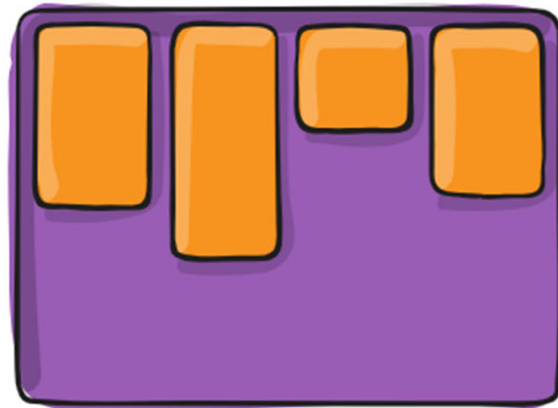
space-evenly



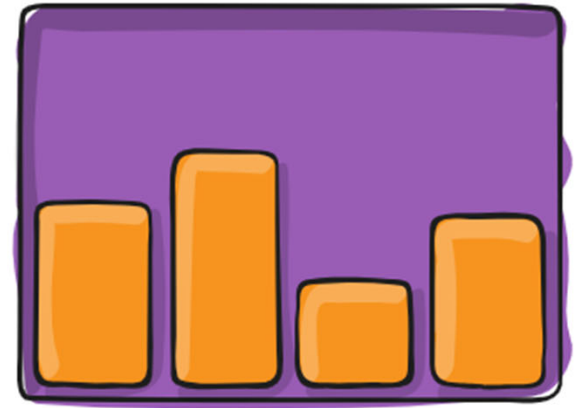
FLEX

align-items

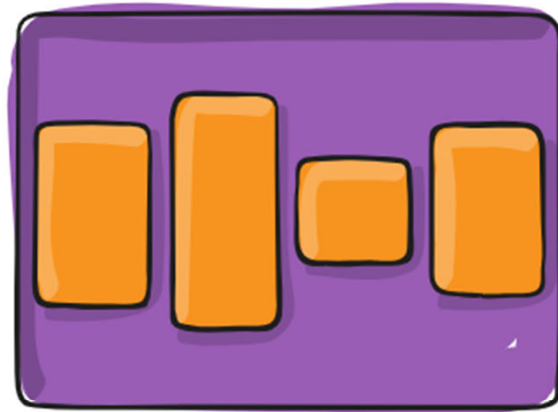
flex-start



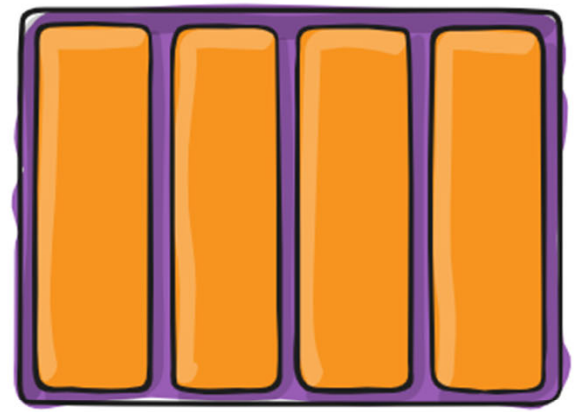
flex-end



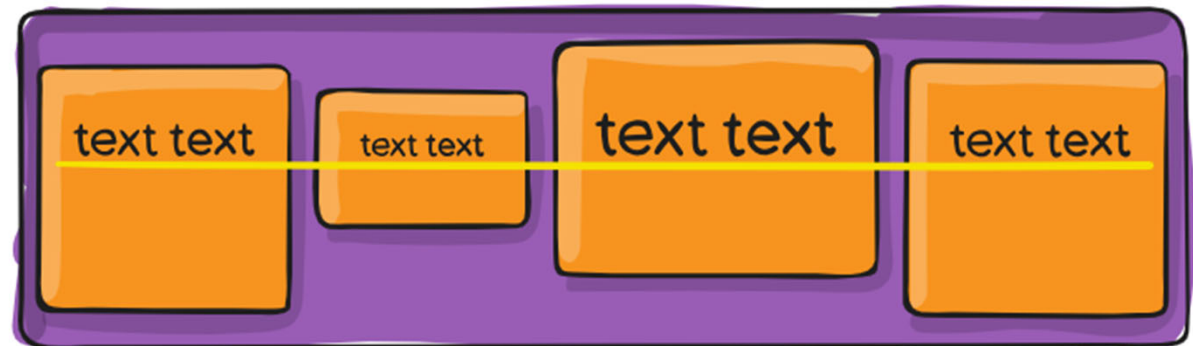
center



stretch



baseline



EINMITTEN

EINMITTEN VON ELEMENTEN

Wir können horizontal und/oder vertikal einmitten.
Teste vor allem ab IE9!

- Es gibt moderne und altmodische Lösungen.
Sei dir bewusst, dass das moderne CSS transform: translate (x-Achse, y-Achse) zur Zeit noch viele Internet Explorer-Anhänger ausschliesst!
- Wie teste ich für ältere Versionen des Internet Explorers?
 - Du hast einen PC:
In neueren Internet-Explorer kannst du im "Web Inspector" ältere Versionen vom IE emulieren und somit testen.
 - Du hast einen Mac:
ein Notbehelf zum Testen ist: <http://browsershots.org/>
- Siehe Ordner einmitten auf dem Server für Beispiele

EINMITTEN VON ELEMENTEN

Zusammenfassung der CSS-Einmittungstechniken

- Zentriert (aber auch nur horizontal oder vertikal möglich) ab IE 11:
 - das Kind-Element hat:
position: relative; top: 50%; left: 50%;
und Korrektur: transform: translate(-50%, -50%);
- Horizontal – für alle Browser:
 - für INLINE Kind-Elemente:
text-align: center; im Eltern-Element
 - für BLOCK Kind-Elemente:
width: __px; margin: 0 auto; im Kind-Element
- Vertikal – für alle Browser:
 - egal ob INLINE oder BLOCK Kind-Element:
display: table-cell; vertical-align: middle; im Eltern-Element