

## 10 Mehrspaltigkeit mit flexbox (display: flex)

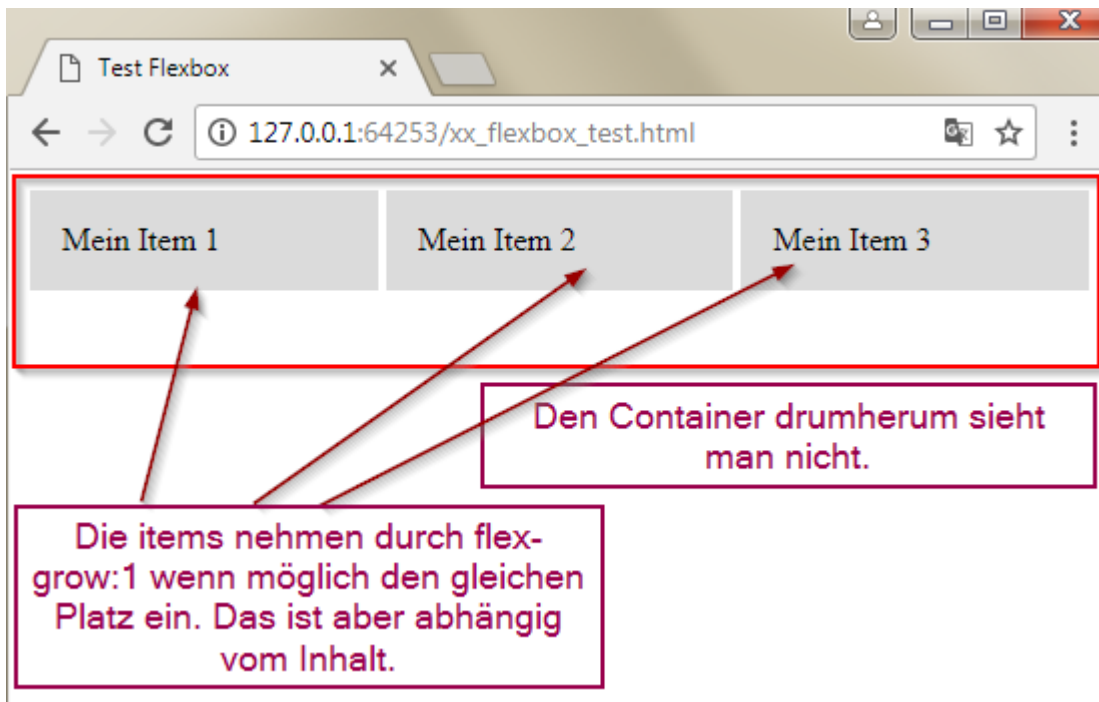
Zum Einsatz der CSS Technik flexbox benötigt man immer einen umschließenden Container und die darin enthaltenen Elemente nennt man dann items.

Da man häufig keinen umfließenden Container hat, kann man dazu auch den eigenschaftslosen Container **div** verwenden.

### 10.1 Kurzeinführung display: flex

Betrachten Sie folgendes Beispiel:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Test Flexbox</title>
    <style>
      #wrapper {
        display: flex;
      }
      .item {
        background-color: #dbdbdb;
        border: 2px solid white;
        padding: 1em;
        flex-grow: 1;
      }
    </style>
  </head>
  <body>
    <div id="wrapper">
      <div class="item">Mein Item 1</div>
      <div class="item">Mein Item 2</div>
      <div class="item">Mein Item 3</div>
    </div>
  </body>
</html>
```



## 10.2 Navigation mit flexbox gestalten

- Wenden Sie die Technik auf die Liste der Navigation an, damit diese Elemente nebeneinander angezeigt werden und bei Platzmangel untereinander dargestellt werden.
- Was könnte der Flex-Container sein?

- 
- Was sind die items?

- 
- In der Lösung können Sie auch flex-wrap:wrap einsetzen, was bewirkt diese Anweisung? Wozu wäre das sinnvoll?
-

### 10.3 Flexbox – ausführlich

Mit dem Modul der Flexboxen oder Flexbox Layout (Flexible Box) ermöglicht es Bereiche (Items), deren Breite unbekannt ist, effizient zu verteilen und zu positionieren. Die Idee besteht darin, dass den Items die Möglichkeit gegeben wird, ihre Größe zu verändern, je nachdem wie viel Platz vorhanden ist.

#### 10.3.1 Literatur

<https://www.w3.org/TR/css-flexbox-1/>

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

<https://blog.kulturbanause.de/2013/07/einfuehrung-in-das-flexbox-modell-von-css/>

<http://flexboxfroggy.com/> – spielerisches Lernen

<https://www.smashingmagazine.com/2015/08/flexible-future-for-web-design-with-flexbox/>

#### 10.3.2 Zum Vergleich: **Spaltenlayout mit float**

Betrachten wir folgendes Beispiel:

```
<DOCTYPE html>
<html lang="de">
  <head>
    <meta charset="utf-8" />
    <title>Floatende Container</title>
    <style>
      * {
        box-sizing: border-box;
      }
      body {
        font-family: Calibri, Verdana, Arial, sans-serif;
      }
      .left, main, aside {
        padding: 5px;
        float: left;
        background-color: #d4dcef;
      }
      div.left {
        width: 20%;
      }
      main {
        width: 60%;
      }
      aside {
        width: 20%;
      }
    </style>
  </head>

  <body>
    <div id="wrapper">
      <div class="left">
        <h2>Linke Spalte</h2>
        <p>
          Lorem ...
        </p>
      </div>
```

```
<main>
  <h1>Hauptinhalte</h1>
</main>
<aside>Seitenanmerkungen</aside>
</div>
</body>
</html>
```

→ Welche Eigenschaften von float fallen Ihnen sofort auf?

---

---

---

→ Sie möchten drei gleich große Spalten haben, die den Anzeigebereich ausfüllen, wie lösen Sie dieses Problem?

---

---

---

→ Sie möchten nicht nur drei, sondern neun div-Container haben, jeweils drei in einer Reihe. Wie können Sie dies mit float lösen?

---

---

---

### Fazit:

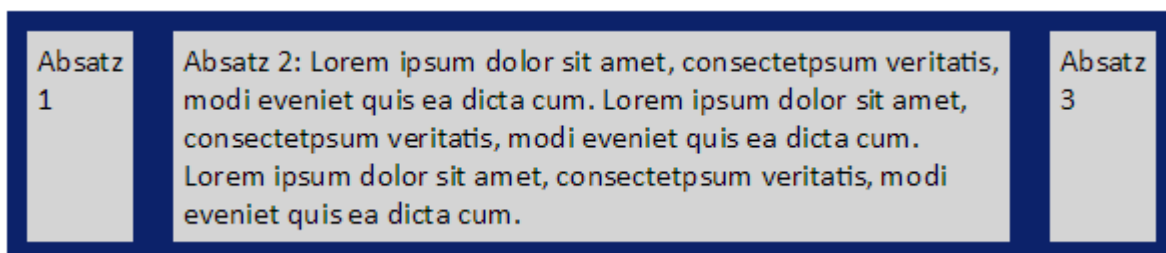
Der Aufbau einer Webseite mit floatenden Bereichen war und ist immer ein Problem, da floatende Elemente keine Höhe haben und es damit Schwierigkeiten mit einheitlichen Höhen gibt. Zudem muss darauf geachtet werden, dass die Eigenschaft „float“ mit dem Befehl clear: all oder clear: left auch wieder beendet wird, damit Elemente nicht aus Versehen in vorhandene Lücken rutschen.

## 10.3.3 Beispiel 1 - 01\_index\_flex1.html

```

<!DOCTYPE html>
<html lang="de">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Flexbox Beispiel 1</title>
  <style>
    body {
      font-family: Calibri, Verdana, Arial, sans-serif;
    }
    #wrapper {
      background-color: #0c226a;
      display: flex;
    }
    #wrapper > div {
      padding: 5px;
      background-color: #d4d4d4;
      margin: 10px;
    }
  </style>
</head>
<body>
  <div id="wrapper">
    <div>Absatz 1</div>
    <div>Absatz 2: ... ipsum dolor sit amet, consectetursum veritatis, modi
eveniet quis ea dicta cum. </div>
    <div>Absatz 3</div>
  </div>
</body>
</html>

```



Wie Sie erkennen, werden alle nebeneinander angezeigt **und** auf die gleiche Höhe gebracht!!

Die Breite ist abhängig von den Inhalten.

Dem übergeordneten Container kann man auch die Richtung mitgeben. row ist der default-Wert.

```

section {
  flex-direction: row | row-reverse | column | column-reverse;
}

```

### 10.3.4 Veränderung der Reihenfolge der Kindelemente mit **order**

Man kann den Kindern auch über **order** eine andere Reihenfolge mitgeben.

Testen Sie folgendes aus:

```
#wrapper > div:nth-child(1) {  
    order:2;  
}  
#wrapper > div:nth-child(2) {  
    order:1;  
}  
#wrapper > div:nth-child(3) {  
    order:3;  
}
```

Die Reihenfolge wird entsprechend geändert.

→ Wozu könnte eine Veränderung der Reihenfolge eingesetzt werden?

---

---

---

---

Das Flexbox-Modell bietet neue Möglichkeiten, die mit der alten float-Technik nur mühevoll und durch umständliche Tricks erledigt werden konnten. Dies ist besonders für das responsive Webdesign interessant.

Die Reihenfolgen können mühelos angepasst werden, ebenso die Breiten und die Höhen.

Das Flexbox-Modell arbeitet jeweils mit einem übergeordneten Container, dem die Eigenschaft `display: flex` mitgegeben werden muss. Anschließend können viele Eigenschaften sowohl für den Container als auch für die Kindelemente (items) angegeben werden. Eine gute vollständige Übersicht findet sich hier:

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

### 10.3.5 Flex-grow - 02\_index\_flex2.html

Wir wollen erst einmal ein einfaches Design mit Flexbox entwerfen und nutzen dazu unterschiedliche Eigenschaften des übergeordneten Containers und der items.

Unser Layout soll im breiten Zustand drei nebeneinanderstehende Bereiche haben, die unterschiedlich breit sind. Der mittlere Absatz soll einen größeren Platz einnehmen.

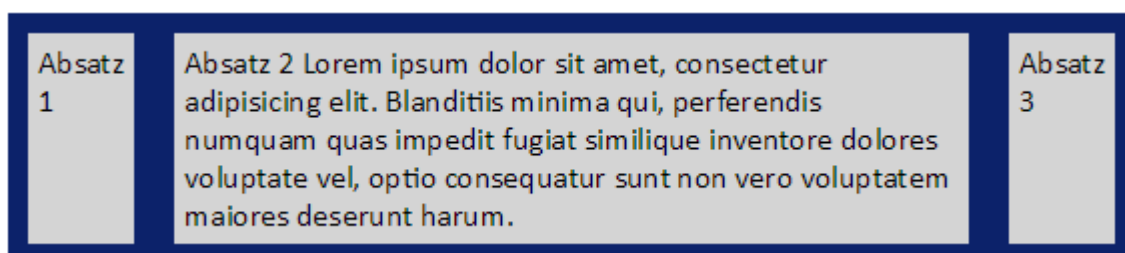
Testen Sie folgendes einfaches Layout:

```
#wrapper {  
    background-color: #0c226a;  
    display: flex;  
}  
#wrapper > div {  
    padding: 5px;  
    background-color: #d4d4d4;  
    margin: 10px;  
}  
#wrapper > div:nth-child(1) {  
    flex-grow:1;  
}  
#wrapper > div:nth-child(2) {  
    flex-grow:2;  
}  
#wrapper > div:nth-child(3) {  
    flex-grow:1;  
}  
  
</style>  
</head>  
<body>  
    <div id="wrapper">  
        <div>Absatz 1</div>  
        <div>Absatz 2 </div>  
        <div>Absatz 3</div>  
    </div>  
</body>  
</html>
```



Sieht so weit klasse aus, aber nicht, wenn im Absatz zwei noch mehr Inhalt eingegeben wird.

Dann nimmt der zweite Absatz mehr Platz ein, dem Verhältnis von 1:2:1 entspricht das jetzt nicht!



### 10.3.5.1 flex-grow

In dem ersten Beispiel benutzen wir die Eigenschaft flex-grow des Kindelementes (items). Dessen Standardwert ist 0. Wie sieht die Spezifikation von flex-grow aus?

This <number> component sets flex-grow longhand and specifies the flex grow factor, which **determines how much the flex item will grow relative to the rest of the flex items** in the flex container when positive free space is distributed. When omitted, it is set to 1." (Quelle: <https://www.w3.org/TR/css-flexbox-1/#flex-property>)

Oder:

„This defines the ability for a flex item to grow if necessary. It accepts a unitless value that serves as a proportion. It dictates what amount of the available space inside the flex container the item should take up.

If all items have flex-grow set to 1, the remaining space in the container will be distributed equally to all children. If one of the children has a value of 2, the remaining space would take up twice as much space as the others (or it will try to, at least).“ (Quelle: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>)

Da wir es aber mit flexiblen Elementen zu tun haben, deren Platz gleichmäßig über die verfügbare Breite ausgedehnt wird, nimmt in diesem Fall der mittlere Bereich mehr Platz ein, da die anderen beiden Absätze keinen weiteren Platz benötigen.

**Wenn kein Inhalt in den Boxen ist, verhalten sie sich genau nach den Werten von flex-grow.**

**Allein scheint dieses Mittel wo noch nicht zu funktionieren.**

### 10.3.5.2 flex-shrink

Mit flex-shrink wird bestimmt, ob ein Element kleiner werden darf und in welchem Verhältnis. Der Standardwert ist 1. Flex-shrink hat allerdings ohne eine grundsätzliche Angabe der Breite keine Auswirkungen, da nicht klar ist, von welchem Basiswert etwas nicht bzw. in welchem Verhältnis es kleiner werden darf.

### 10.3.5.3 flex-basis

Damit wir die Eigenschaft flex-shrink testen können, benutzen wir die zusätzliche Angabe flex-basis.

Wir gehen davon aus, dass der linke Bereich eine feste Mindestbreite von 150px haben soll und nicht schmaler werden darf (weil dann z. B. eine Navigation nicht mehr sichtbar ist.)

Dazu empfehlen sich die Angaben.

flex-grow: 1

flex-shrink:0

flex-basis:150px

→ Testen Sie!

### 10.3.5.4 Übung

Linke Spalte: mindestens 150px, darf nicht kleiner werden, sonst 1 Teil von vier

Mittlere Spalte: mindestens 300px, darf kleiner werden, wenn Platz ist 2 Teile von vier

Rechte Spalte: mindestens 150px, darf kleiner werden, sonst 1 Teil von vier.

|   |   |                        |
|---|---|------------------------|
| <b>Firma</b> <ul style="list-style-type: none"> <li><a href="#">Link</a></li> <li><a href="#">Link</a></li> <li><a href="#">Link</a></li> <li><a href="#">Link</a></li> </ul> | <b>Hauptinhalt</b> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Voluptates in, rerum illum quo eaque dolore perferendis consectetur aperiam atque neque, laborum quod explicabo. Ratione cupiditate voluptas dolor rem, magni eaque!</p> | <b>Werbung oder so</b> |
|---|---|------------------------|



### 10.3.5.5 Lösung

```
<DOCTYPE html>
<html lang="de">
  <head>
    <meta charset="utf-8">
    <title>Flexbox Layout</title>
    <style type="text/css">
      body {
        font-family: Calibri, Verdana, Arial, sans-serif;
      }
      #wrapper {
        background-color: #0c226a;
        display: flex;
      }
      #wrapper > div {
        padding: 5px;
        background-color: #d4d4d4;
        margin: 10px;
      }
      #wrapper > div:nth-child(1) {
        flex-grow:1;
        flex-shrink:0;
        flex-basis: 150px;
      }
      #wrapper > div:nth-child(2) {
        flex-grow:2;
        flex-shrink:1;
        flex-basis: 300px;
      }
      #wrapper > div:nth-child(3) {
        flex-grow:1;
        flex-shrink:1;
        flex-basis: 150px;
      }
    </style>
  </head>
  <body>
    <div id="wrapper">
      <div>
        <h1>Firma</h1>
        <nav>
          <ul>
            <li><a href="#">Link</a></li>
            <li><a href="#">Link</a></li>
            <li><a href="#">Link</a></li>
            <li><a href="#">Link</a></li>
          </ul>
        </nav>
      </div>
      <div><h2>Hauptinhalt</h2><p>Lorem ipsum ...</p></div>
      <div><h2>Werbung oder so</h2></div>
    </div> ...
```

### 10.3.5.7 Abkürzung der drei Eigenschaften mit flex

flex ist die Kurzform für die Eigenschaften

flex-grow: Welche anteilige Breite nimmt das Element ein, wenn der Container breit wird (das Element wächst, falls nötig)?

flex-shrink: Welche anteilige Breite nimmt das Element ein, wenn die Breite schmaler wird?

flex-basis: Gibt die default-Breite an, bevor weiterer Platz vergeben wird.

| Angaben für flex   |                 |
|--|-----------------|
| Default  | flex: 1 0 auto; |
| /* 0 0 auto */   | flex: none;     |
| /* One value, unitless number: flex-grow */              | flex: 2;        |
| /* Two values: flex-grow   flex-basis */                 | flex: 1 30px;   |
| /* Two values: flex-grow   flex-shrink */                | flex: 2 2;      |
| /* Three values: flex-grow   flex-shrink   flex-basis */ | flex: 2 2 10%;  |

→ Benutzen Sie die Kurschreibweise für das eigene Beispiel.

### 10.3.6 Umbrüche mit wrap erzeugen

```
.container {
  flex-wrap: nowrap | wrap | wrap-reverse;
}
```

→ Testen Sie wrap für den #wrapper aus!!

Normalerweise benutzt man flex-flow: row wrap; als Kurzschreibweise.

## 10.5 Einfaches Spaltenlayout mit flex

### 10.5.1 03\_index\_flex3\_website.html

```

<title>Flexbox einfache Website</title>
<style>
  * {
    box-sizing: border-box; /* Alternatives Box-Model Entscheidend, damit
padding und border nicht auf die Breite addiert werden */
  }
  body {
    font-family: Calibri, Verdana, Arial, sans-serif;
    margin: 0;
    padding: 0;
  }
  #wrapper {
    display: flex;
  }
  nav, main, aside {
    padding: 5px;
    background-color: #d4d4d4;
    border: 1px solid white;
  }
  nav {
    flex: 0 0 25%; /* flex-grow: 0; flex-shrink:0 ; flex-basis: 25% */
  }
  main {
    flex: 0 0 50%;
  }
  aside {
    flex: 0 0 25%;
  }
</style>
</head>
<body>
  <div id="wrapper">
    <nav>Hier steht die Navigation</nav>
    <main><h2>Hier steht der Hauptinhalt</h2><p>Lorem, ipsum dolor sit amet
consectetur adipisicing elit. Soluta, veniam deserunt molestias quidem, voluptate
mollitia, sapiente quae blanditiis sunt amet quisquam quam in id quibusdam deleniti.
Facere aliquid tenetur distinctio.</p> </main>
    <aside>Seitenspalte</aside>
  </div>
</body>

```

### 10.5.2 Weitere Beispiel dazu

04\_index\_flex4\_header\_footer // Mit header und footer

05\_index\_flex5\_responsive // mit Media Query

06\_index\_flex6\_kaestchen // Anordnung vieler gleich hoher und breiter Elemente

## 10.6 Zusammenfassung Flexbox

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

| Eigenschaften des übergeordneten Containers – flex container |              |  |
|--|--------------|--|
| Eigenschaft  | Beschreibung | Mögliche Werte, Zeichnungen<br>Beispiele |
| display:flex   |              |  |
| flex-direction   |              |  |
| flex-wrap  |              |  |
| flex-flow  |              |  |
| justify-content  |              |  |
| align-items  |              |  |
| align-content  |              |  |

| <b>Eigenschaften der enthaltenden Container – flex items</b> |                     |   |
|--|---------------------|---|
| <b>Eigenschaft</b>   | <b>Beschreibung</b> | <b>Mögliche Werte, Zeichnungen, Beispiele</b> |
| order  |                     |   |
| flex-grow  |                     |   |
| flex-shrink  |                     |   |
| flex-basis   |                     |   |
| flex   |                     |   |
| align-self   |                     |   |

## 10.7 Übung 1 zu Flexbox

→ Erstellen Sie eine Site, die die verschiedenen Ansichten wie unten hat.

| Reihenfolge im HTML-Quellcode  | Checkliste  |
|--|---|
| <ul style="list-style-type: none"> <li>header: „Einfache Site“</li> <li>Hauptinhalt</li> <li>Anmerkungen</li> <li>Werbung</li> <li>Footer</li> </ul> | <ul style="list-style-type: none"> <li>Reihenfolge einhalten</li> <li>flexbox verwenden</li> <li>die Container passen sich an die Breite an – mit Prozentwerten arbeiten</li> <li>Mobile first</li> </ul> |

schmal

|   |
|---|
| Einfache Site   |
| <p>Hauptinhalt, hier stehen die wichtigsten Sachen</p> <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Qui laborum magnam libero possimus mollitia omnis, nam repellat incidunt voluptatum rem earum accusamus magni corporis. A cumque dolore delectus quas optio. Lorem ipsum dolor sit amet, consectetur adipisicing elit. Molestias officia ipsam fugit ipsum laborum dolor sunt fuga nihil iste, recusandae debitis ex, est sapiente, quod ut autem neque veritatis suscipit. Lorem ipsum dolor sit amet, consectetur adipisicing elit. Dolorem repellendus voluptate quos accusamus voluptatibus delectus modi excepturi placeat eius consectetur debitis neque iste, vero eum nesciunt labore quidem temporibus architecto!</p> |
| Werbung bringt Geld   |
| Anmerkungen eher unwichtig  |
| Fußzeile  |

mittel

|  |
|--|
| Einfache Site  |
| <div>Anmerkungen eher unwichtig</div> <div> <p>Hauptinhalt, hier stehen die wichtigsten Sachen</p> <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Qui laborum magnam libero possimus mollitia omnis, nam repellat incidunt voluptatum rem earum accusamus magni corporis. A cumque dolore delectus quas optio. Lorem ipsum dolor sit amet, consectetur adipisicing elit. Molestias officia ipsam fugit ipsum laborum dolor sunt fuga nihil iste, recusandae debitis ex, est sapiente, quod ut autem neque veritatis suscipit. Lorem ipsum dolor sit amet, consectetur adipisicing elit. Dolorem repellendus voluptate quos accusamus voluptatibus delectus modi excepturi placeat eius consectetur debitis neque iste, vero eum nesciunt labore quidem temporibus architecto!</p> </div> |
| Werbung bringt Geld  |
| Fußzeile   |

breit

|   |
|---|
| Einfache Site   |
| <div>Anmerkungen eher unwichtig</div> <div> <p>Hauptinhalt, hier stehen die wichtigsten Sachen</p> <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Qui laborum magnam libero possimus mollitia omnis, nam repellat incidunt voluptatum rem earum accusamus magni corporis. A cumque dolore delectus quas optio. Lorem ipsum dolor sit amet, consectetur adipisicing elit. Molestias officia ipsam fugit ipsum laborum dolor sunt fuga nihil iste, recusandae debitis ex, est sapiente, quod ut autem neque veritatis suscipit. Lorem ipsum dolor sit amet, consectetur adipisicing elit. Dolorem repellendus voluptate quos accusamus voluptatibus delectus modi excepturi placeat eius consectetur debitis neque iste, vero eum nesciunt labore quidem temporibus architecto!</p> </div> <div>Werbung bringt Geld</div> |
| Fußzeile  |

## 10.8 Übung 2 zu Flexbox

Aussehen breit

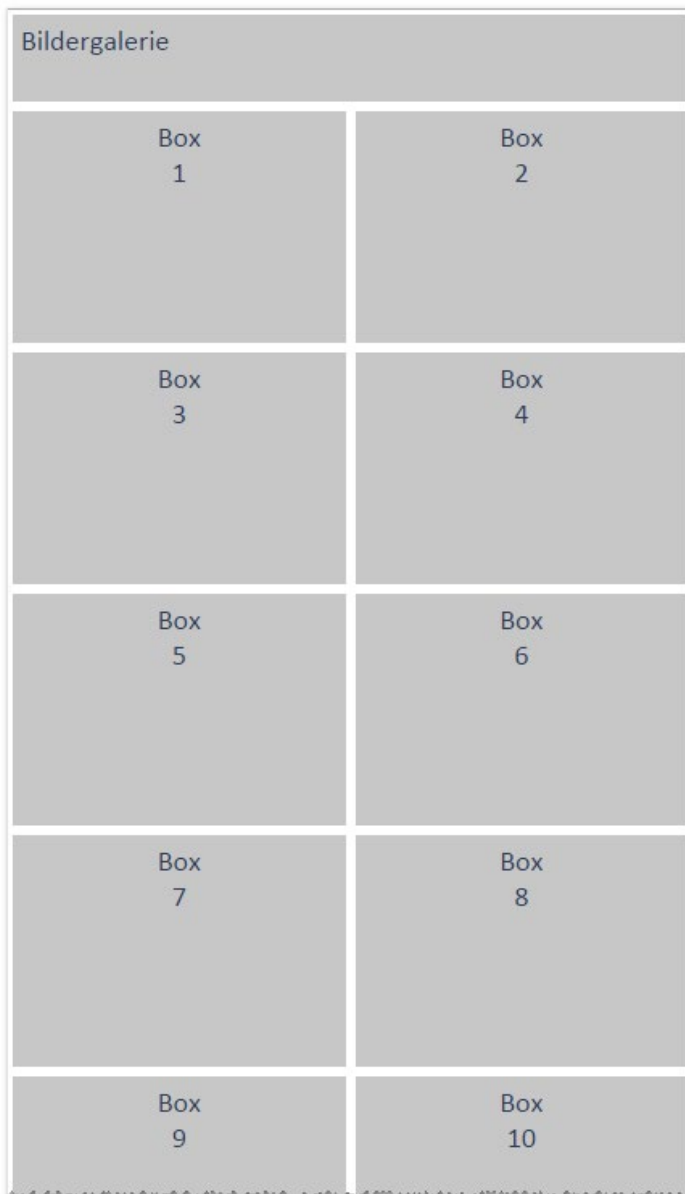
| Bildergalerie |          |          |           |           |           |
|---------------|----------|----------|-----------|-----------|-----------|
| Box<br>1      | Box<br>2 | Box<br>3 | Box<br>4  | Box<br>5  | Box<br>6  |
| Box<br>7      | Box<br>8 | Box<br>9 | Box<br>10 | Box<br>11 | Box<br>12 |
| Fußzeile      |          |          |           |           |           |

mittel

| Bildergalerie |          |          |           |           |
|---------------|----------|----------|-----------|-----------|
| Box<br>1      | Box<br>2 | Box<br>3 | Box<br>4  | Box<br>5  |
| Box<br>6      | Box<br>7 | Box<br>8 | Box<br>9  | Box<br>10 |
| Box<br>11     |          |          | Box<br>12 |           |
| Fußzeile      |          |          |           |           |

die Boxen rutschen nach und nach untereinander

schmal



### Checkliste

- ✓ nur mit flexbox arbeiten – media-Queries sind nicht nötig



## 10.9 Übung 3 zu Flexbox

### Aussehen breit

|                              |          |       |        |        |        |
|------------------------------|----------|-------|--------|--------|--------|
| logo, suche etc              |          |       |        |        |        |
| hier käme jetzt die navi hin | Box groß |       |        |        | Box A  |
|                              |          |       |        |        | Box B  |
|                              |          |       |        |        | Box C  |
| Box 1                        | Box 2    | Box 3 | Box 4  | Box 5  | Box 6  |
| Box 7                        | Box 8    | Box 9 | Box 10 | Box 11 | Box 12 |
| Fußzeile                     |          |       |        |        |        |

### Aussehen mittel

|                              |       |       |        |        |        |
|------------------------------|-------|-------|--------|--------|--------|
| logo, suche etc              |       |       |        |        |        |
| hier käme jetzt die navi hin |       |       |        |        |        |
| Box groß                     |       |       |        | Box A  |        |
|                              |       |       |        | Box B  |        |
|                              |       |       |        | Box C  |        |
| Box 1                        | Box 2 | Box 3 | Box 4  | Box 5  | Box 6  |
| Box 7                        | Box 8 | Box 9 | Box 10 | Box 11 | Box 12 |
| Fußzeile                     |       |       |        |        |        |

Aussehen schmaler

Die Anzahl der dargestellten Boxen 1 bis 12 passt sich der Breite an.

ganz schmal

|                              |           |
|------------------------------|-----------|
| logo, suche etc              |           |
| hier käme jetzt die navi hin |           |
| Box<br>groß                  |           |
| Box<br>A                     |           |
| Box<br>B                     |           |
| Box<br>C                     |           |
| Box<br>1                     | Box<br>2  |
| Box<br>3                     | Box<br>4  |
| Box<br>5                     | Box<br>6  |
| Box<br>7                     | Box<br>8  |
| Box<br>9                     | Box<br>10 |
| Box<br>11                    | Box<br>12 |
| Fußzeile                     |           |