

STUDIERN. WISSEN. MACHEN.

CSS



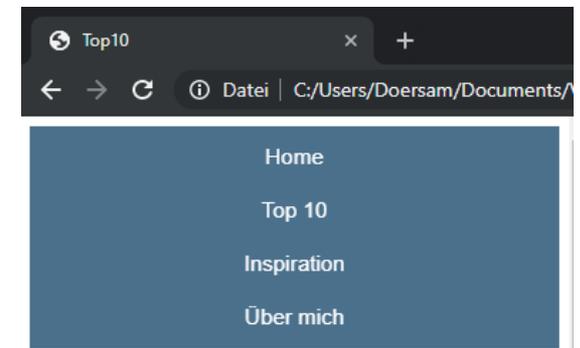
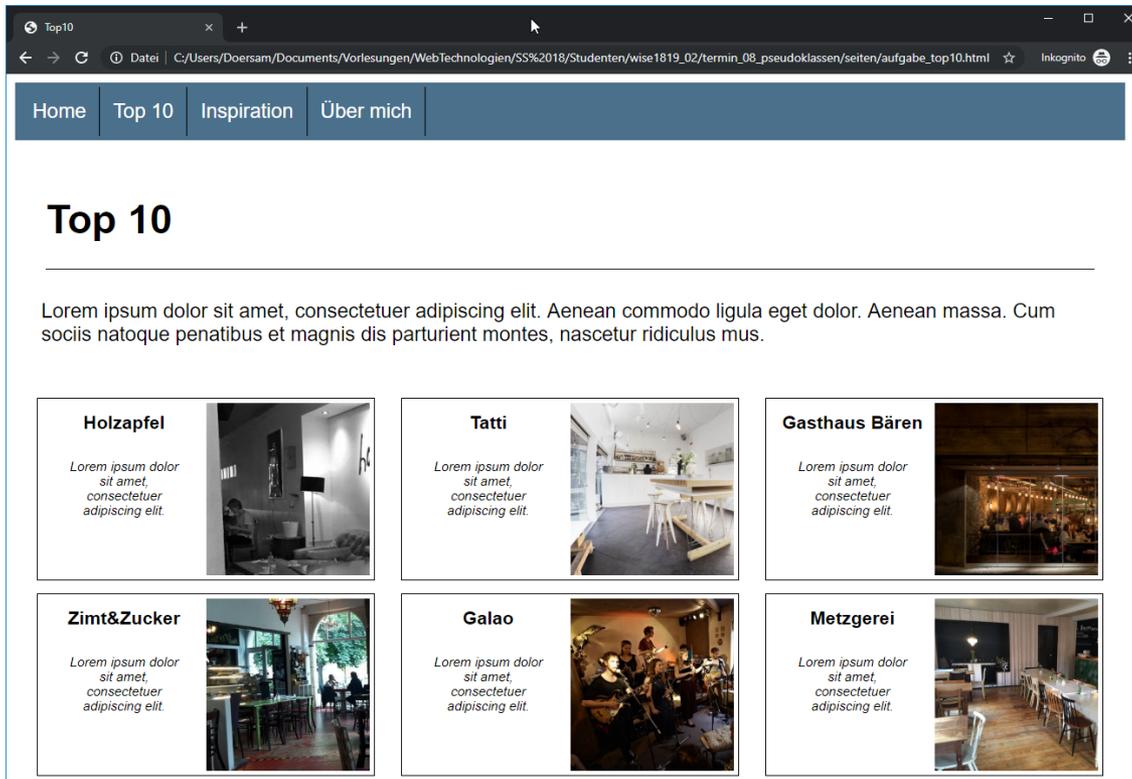
HTML



CSS

Positionierung mit Flexboxen

Responsives Design



Top 10

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.



Ziel: unterschiedliche Layouts für unterschiedliche Hardware / Bildschirmauflösung

Flexboxen

- › Um das responsive Design so einfach wie möglich gestalten zu können, wurden mit CSS3 so genannte *Flexboxen* eingeführt.
- › Vorteile von Flexboxen:
 - › Flexible Gestaltung der Boxenbreite und –höhe
 - › Flexible Anordnung von Boxen in Zeilen oder Spalten
 - › Flexible Ausrichtung von Boxen aneinander
- › Dabei definiert der Webseitenentwickler nur das gewünschte Verhalten – die Positionierung von Boxen berechnet der Browser automatisch.

Flexboxen

- › Flexboxen werden durch die Eigenschaft **display** für den *Container* definiert:

```
#container {  
    display: flex;  
}
```

- › Bei den enthaltenen Boxen (*Flex-Items*) müssen keine weiteren Informationen hinterlegt werden.
- › Der Browser berechnet standardmäßig die Breite und Höhe aller direkt enthaltener Boxen so, dass sie
 - › alle nebeneinander stehen und
 - › die gleiche Höhe haben.

Flexboxen - Beispiel

```
#container {
  display: flex;
}

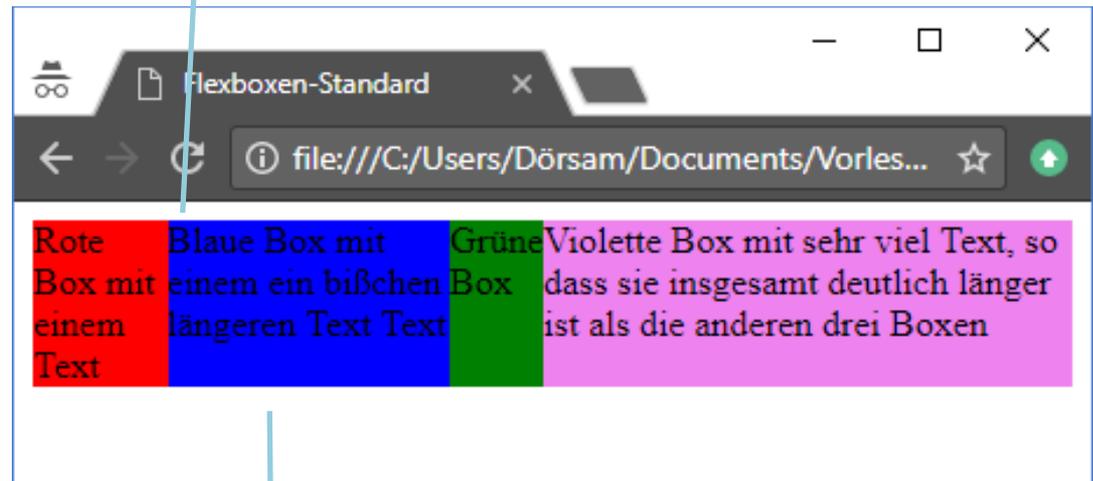
#fitem1 {
  background: red;
}

#fitem2 {
  background: blue;
}

#fitem3 {
  background: green;
}

#fitem4 {
  background: violet;
}
```

Alle Boxen stehen nebeneinander.



Alle Boxen sind gleich hoch.

Vertikale Flexboxen

- › Standardmäßig stehen alle Boxen nebeneinander.
- › Das wird durch die Eigenschaft **flex-direction** erreicht, die standardmäßig auf den Wert **row** gesetzt ist.
- › Sollen die Spalten untereinander angeordnet werden, so muss die Eigenschaft auf den Wert **column** gesetzt werden. Die Boxen sind dann automatisch so breit wie der übergeordnete Container.

```
#container {  
  display: flex;  
  flex-direction: column;  
}
```

- › Weitere mögliche Werte sind **row-reverse** und **column-reverse** (umgedrehte Reihenfolge der Boxen)

Vertikale Flexboxen - Beispiel

```
#container {
  display: flex;
  flex-direction: column;
}

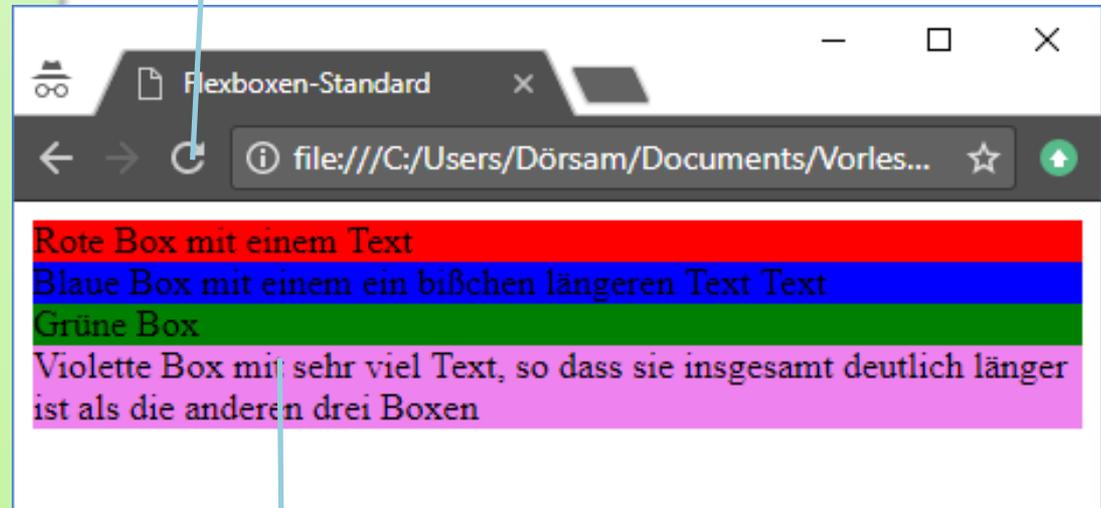
#fitem1 {
  background: red;
}

#fitem2 {
  background: blue;
}

#fitem3 {
  background: green;
}

#fitem4 {
  background: violet;
}
```

Alle Boxen stehen untereinander.



Alle Boxen sind gleich breit.

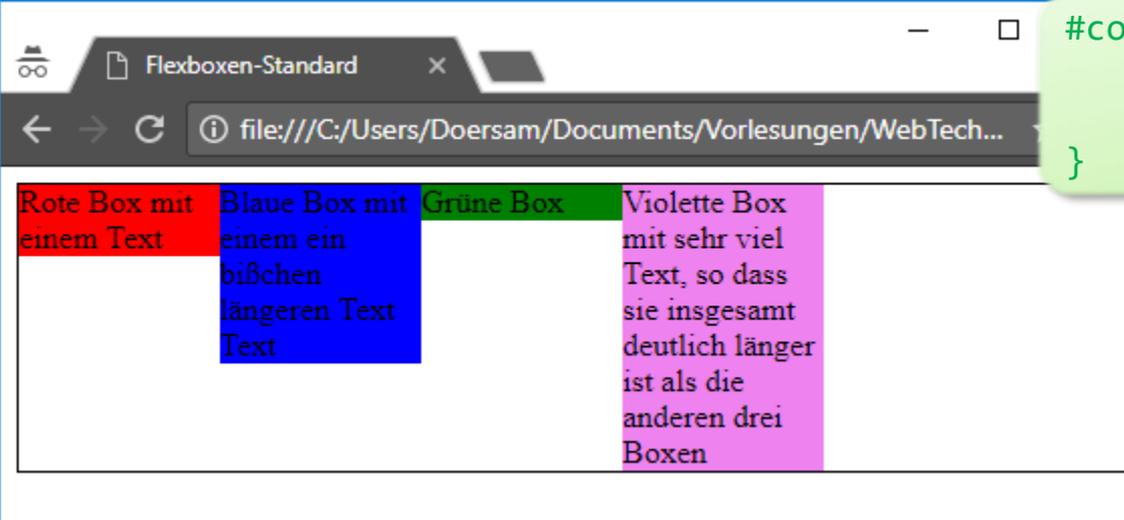
[Interaktives Beispiel](#)

Flexboxen – Aufgabe 1

Flex-Container: Ausrichtung der Flex-Items

- › **align-items**: Ausrichtung der Flex-Items vertikal an der horizontalen Achse oder horizontal an der vertikalen Achse.
- › Mögliche Werte und Bedeutung (bei **flex-direction: row;**):
 - › **flex-start**: Ausrichtung oben; Höhe richtet sich nach dem Inhalt der jeweiligen Box
 - › **flex-end**: Ausrichtung unten; Höhe richtet sich nach dem Inhalt der jeweiligen Box
 - › **center**: Ausrichtung mittig ; Höhe richtet sich nach dem Inhalt der jeweiligen Box
 - › **stretch**: Ausrichtung oben; gleiche Höhe für alle Boxen (Standard)
 - › ...

Flex-Container: Ausrichtung der Flex-Items



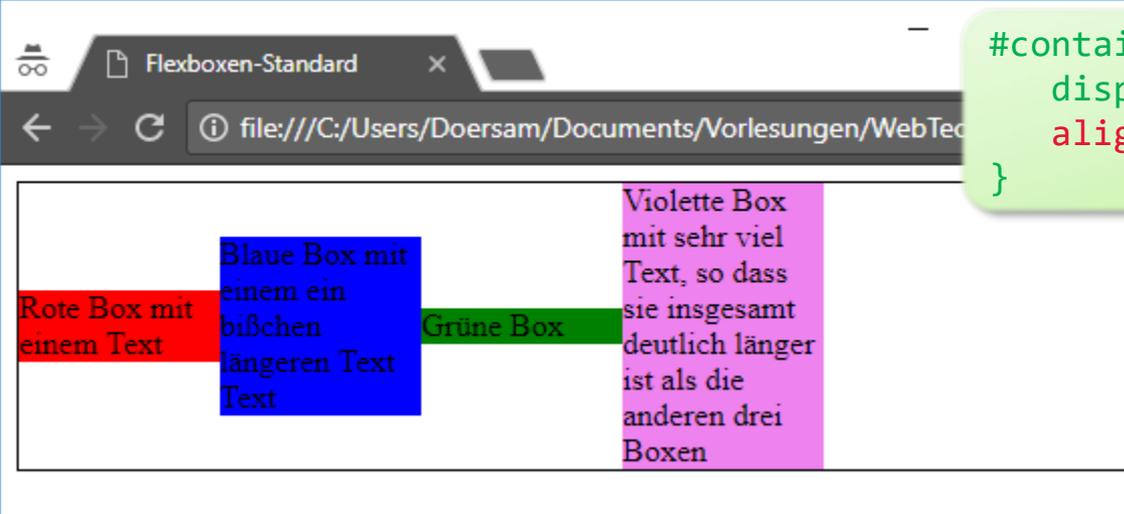
Rote Box mit einem Text

Blaue Box mit einem ein bißchen längeren Text

Grüne Box

Violette Box mit sehr viel Text, so dass sie insgesamt deutlich länger ist als die anderen drei Boxen

```
#container {
  display: flex;
  align-items: flex-start;
}
```



Rote Box mit einem Text

Blaue Box mit einem ein bißchen längeren Text

Grüne Box

Violette Box mit sehr viel Text, so dass sie insgesamt deutlich länger ist als die anderen drei Boxen

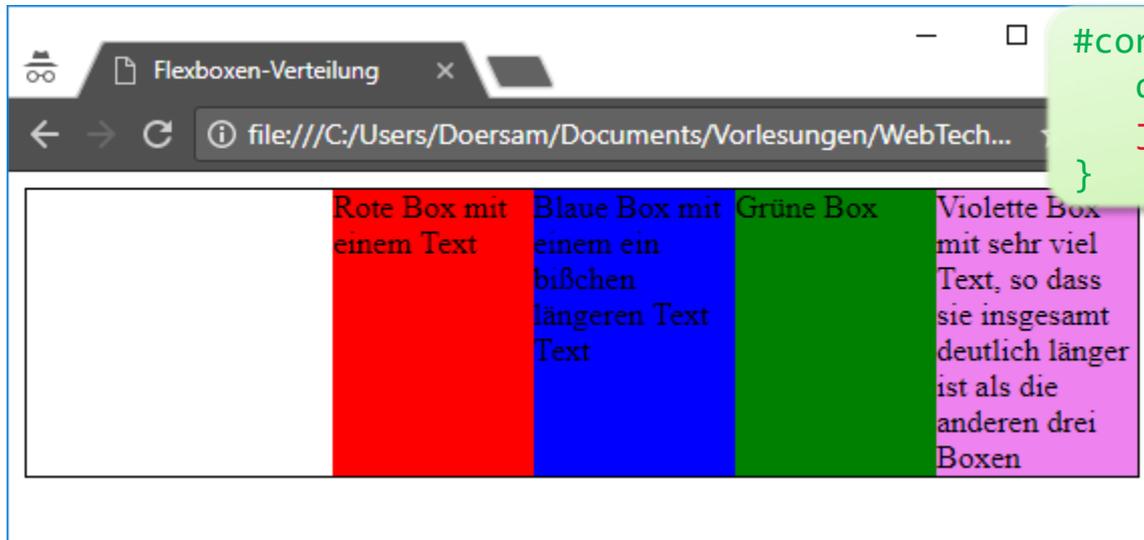
```
#container {
  display: flex;
  align-items: center;
}
```

Flexboxen – Aufgabe 2

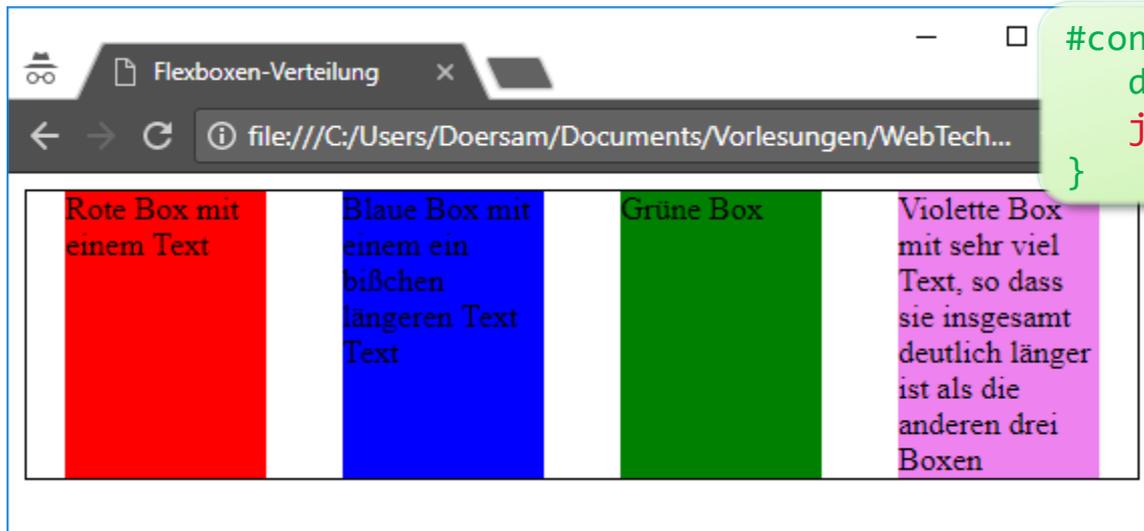
Flex-Container: Verteilung der Flex-Items

- › **justify-content**: bestimmt die Verteilung der Flex-Items, wenn sie so klein sind, dass sie nicht die gesamte Breite oder Höhe des Flex-Containers ausfüllen.
- › Mögliche Werte und Bedeutung (bei **flex-direction: row;**)
 - › **flex-start**: Ausrichtung links (Standardwert)
 - › **flex-end**: Ausrichtung rechts
 - › **center**: Ausrichtung mittig
 - › **space-between**: Zwischen den Flex-Items wird gleichmäßiger Zwischenraum gelassen
 - › **space-around**: Zwischen den Flex-Items sowie vor dem ersten bzw. nach dem letzten Flex-Item wird gleichmäßiger Zwischenraum gelassen

Flex-Container: Verteilung der Flex-Items



```
#container {
  display: flex;
  justify-content: flex-end;
}
```



```
#container {
  display: flex;
  justify-content: space-around;
}
```

[Interaktives Beispiel](#)

Flexboxen – Aufgabe 3

Flex-Container: Zeilenumbruch

- › **flex-wrap**: Automatischer Zeilenumbruch für Flex-Items, wenn sie insgesamt zu breit für den Container sind.
Mögliche Werte:
 - › **no-wrap**: kein Zeilenumbruch
 - › **wrap**: Zeilenumbruch
 - › **wrap-reverse**: Zeilenumbruch, wobei die *Reihenfolge der Zeilen* (nicht der einzelnen Flex-Items) umgedreht wird.

Flex-Container: Zeilenumbruch



```
#container {
  width: 300px;
  display: flex;
  flex-wrap: wrap;
}
```

[Interaktives Beispiel](#)

Flex-Container: Verteilung der Zeilen

- › **align-content**: Verteilung der Zeilen, wenn die Flex-Items in mehreren Zeilen stehen. Mögliche Werte:
 - › **flex-start**: alle Boxen werden so weit wie möglich oben angeordnet
 - › **flex-end**: alle Boxen werden so weit wie möglich unten angeordnet
 - › **center**: der obere und untere Abstand vom Container zu den Boxen sind ungefähr gleich
 - › **space-between**: zwischen den Boxenreihen wird ein gleichmäßiger Abstand gezeichnet
 - › **space-around**: zwischen sowie vor und nach den Boxenreihen wird ein gleichmäßiger Abstand gezeichnet
 - › **stretch**: die Boxenhöhen werden so definiert, dass keine Abstände zwischen den Boxen existieren

Flex-Container: Verteilung der Zeilen



```
#container {
  width: 300px;
  height: 300px;
  display: flex;
  flex-wrap: wrap;
  align-content: space-around;
}
```

[Interaktives Beispiel](#)

Flexboxen – Aufgabe 4

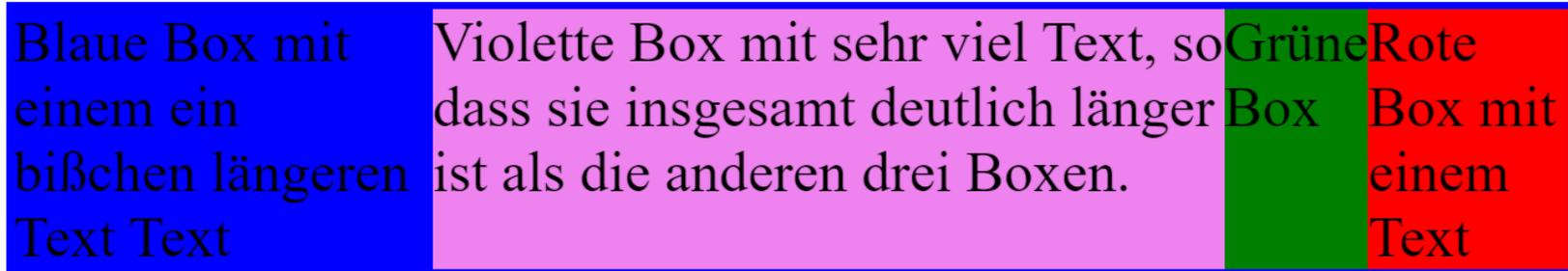
Flex-Items

- › Alle bisherigen Eigenschaften wurden in der *Flexbox* gesetzt, also im Container.
- › Diese Eigenschaften wirken sich auf die *Positionierung der Inhalte, der Flex-Items*, aus.
- › Die *Flex-Items* selbst können auch *zusätzliche Eigenschaften* haben.
- › Beispiele:
 - › **order**: Reihenfolge der Flex-Items innerhalb der Flexbox

Flex-Items: Reihenfolge

- › **order**: Reihenfolge der Flex-Items
- › Mögliche Werte: ganze Zahlen
- › Bedeutung:
 - › Zuerst werden alle Flex-Items in der Standard-Reihenfolge angeordnet, bei denen die **order**-Eigenschaft nicht gesetzt ist.
 - › Anschließend werden alle weiteren Flex-Items in der aufsteigenden Reihenfolge ihrer **order**-Werte positioniert.

Flex-Items: Reihenfolge



```

#container {
  display: flex;
}

#fitem1 {
  background: red;
  order: 2;
}

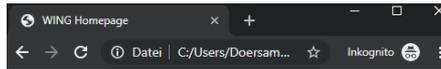
#fitem2 {
  background: blue;
}

#fitem3 {
  background: green;
  order: 1;
}

#fitem4 {
  background: violet;
}
  
```

[Interaktives Beispiel](#)

„Mobile First“-Vorgehen: Beispiel eines Ziel-Layouts




Wirtschafts- ingenieurwesen Medien

News vom Studiengang



Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.



Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

>Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod



Wirtschaftsingenieurwesen Medien

News vom Studiengang

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.



Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.



Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod

„Mobile First“-Vorgehen: Boxen für die Smartphone-Ansicht



„Mobile First“-Vorgehen: Boxen für die Desktop-Ansicht





Wirtschaftsingenieurwesen Medien

News vom Studiengang

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.





Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

„Mobile First“-Vorgehen: Resultierende Boxenstruktur in HTML

```

<body>
  <header>
    <div>
      <img></img>
    </div>
    <div>
      <p></p>
    </div>
  </header>
  ...

```

```

...
<main>
  <h1></h1>
  <div>
    <div>
      <img></img>
    </div>
    <div>
      <p></p>
      <p></p>
    </div>
  </div>
  <div>
    <div>
      <img></img>
    </div>
    <div>
      <p></p>
    </div>
  </div>
</main>
<footer></footer>
</body>

```

Flexboxen: Beispiel WING-Homepage: Schritt 1

Kopfbereich: Breite vom Logo- und Text-Boxen einschränken, nebeneinander stellen

```
header {  
    display: flex;  
}  
  
header .logo {  
    width: 30%;  
}  
  
header .name {  
    width: 70%;  
}
```

Flexboxen: Beispiel WING-Homepage: Schritt 2

Kopfbereich: Die Textbox zentrieren (vertikal und horizontal)

```
header .name {  
    display: flex;  
    align-items: center;  
    justify-content: center;  
}
```

Flexboxen: Beispiel WING-Homepage: Schritt 3

„News-Gruppen“ als
Flexboxen definieren:
Smartphone:
untereinander

Desktop:
nebeneinander,

Inhalte auf 48%
beschränken

```
.gruppe {
  display: flex;
  flex-direction: column;
  margin-bottom: 10px;
}
img {
  max-width: 100%;
}

@media screen and (min-width: 768px) {
  .gruppe {
    flex-direction: row;
    justify-content: space-between;
  }

  .gruppe .newsimg {
    width: 48%;
  }

  .gruppe .newstexte {
    width: 48%;
  }
}
```

Flexboxen: Beispiel WING-Homepage: Schritt 4

erste News-Box: Reihenfolge der Inhalte umdrehen

```
@media screen and (min-width: 768px) {  
  .ungerade {  
    flex-direction: row-reverse;  
  }  
}
```

Flexboxen: Beispiel WING-Homepage: Schritt 5

Footer formatieren

```
footer {  
    height: 30px;  
    padding-top: 6px;  
    margin-top: 10px;  
    border-top: 1px solid black;  
}
```