

# Bootstrap 5 - Framework – Theorie



<https://www.youtube.com/watch?v=c9B4TPnak1A>

## Inhalt:

Ersten 7 Seiten sind Theorie, die Übung startet erst Seite 8.

- Das Framework einbinden
- `<row>` erstellen
- `<col>` Spalten erstellen
- `<offset>` nutzen

Bootstrap ist ein Open-Source-Produkt, welches von Mark Otto und Jacob Thornton für Twitter entwickelt wurde. Ziel war es, eine standardisierte Oberflächenentwicklung für Websites zu schaffen, um so den Wartungsaufwand vieler Seiten zu minimieren.

Bootstrap ist ein kostenloses und Frontend-Web-Framework, das darauf abzielt, Webseiten und -anwendungen zu erstellen. Es konzentriert sich auf die Front-End-Entwicklung und stellt HTML- und CSS-basierte Designvorlagen bereit, die Typografie, Formulare, Schaltflächen und Navigation abdecken. Optional können auch JavaScript-Erweiterungen integriert werden, um Nutzerinteraktionen wie Dialogboxen, Bilder-Slideshows und Tabs zu ermöglichen.

Bootstrap basiert auf HTML, CSS und LESS. Diese enthalten Gesamtvorlagen für Formen, Buttons, Grid (Raster)–Systemen, Tabellen und andere Elemente der Oberflächengestaltung. LESS ist eine Stylesheet-Sprache, die dafür zuständig ist, das Schreiben von CSS zu erleichtern. LESS ist plattformübergreifend, es läuft also auf allen Betriebssystemen und hat das Ziel, Variablen, Mixins und Verschachtelungen zu verwenden, um Code-Wiederholung und Tippfehler zu vermeiden. Zusätzlich gibt es zu den CSS-Stilen noch optionale JavaScript-Erweiterungen, welche auf jQuery basieren.

Frameworks beinhalten vordefinierte Elemente, die das Erstellen und Programmieren von Webseiten erleichtern sollen.

Mit Hilfe von Bibliotheken werden bereits häufig verwendete Abläufe, Programmteile und Programme zusammengefasst. Diese können anschließend aufgerufen und zur Programmierung verwendet werden. Bei solchen Bibliotheken, auch Libraries genannt, ist generell darauf zu achten, um welche Library es sich handelt.

Eines der größten Vorteile von Bootstrap ist das integrierte Responsive-Design. Dieses ermöglicht es, den Inhalt einer Website so zu optimieren, dass er auch auf Geräten wie Tablets oder Handys angepasst wird.

Die Elemente der Website werden im Gegensatz zur Desktop-Ansicht jetzt untereinander angezeigt, um auch mit mobilen Geräten den Überblick zu bewahren. Des Weiteren kann die Navigationsleiste in der mobilen Ansicht nun über einen Button, welcher mit drei Querstrichen dargestellt wird, aufgerufen werden. Um bestimmte Elemente in der mobilen bzw. in der Desktop-Ansicht ein- oder auszublenden, kann auf vorgefertigte Klassen zurückgegriffen werden, die mit der Pixelanzahl der Bildschirme arbeiten.

Des Weiteren enthalten Frameworks CSS-Stile (Cascadian Style Sheet), die ebenfalls in die Anwendung eingebaut werden können. Die Bausteine von CSS-Stilen stellen Layouts und User-Interface-Elemente zur Verfügung. Dies ist besonders sinnvoll, zumal man sich so Programmzeilen erspart, den man sonst mit viel Zeitaufwand selber schreiben müsste. Zudem verringert sich auch die Fehlerquote, die sich bei Eigenarbeit ergeben könnte. So werden die jeweiligen Verlinkungen zum CSS-File in das HTML-Gesamtgerüst eingebaut und durch das Einbinden der Dateien hat man den Zugriff auf die zur Verfügung gestellten Klassen.

Bootstrap baut auf dem Mobile-First-Prinzip auf, welches die Anpassung der Inhalte auf mobilen Geräten in den Vordergrund stellt. Das Open-Source Programm arbeitet mit einem 12 Spalten 960px Rastersystem (Grid-System) - somit passen sich die Spalten in Abhängigkeit der Breite des Gerätes an und ermöglicht ein flexibles Responsive Webdesign.

## Start: 2 Arten der Einbindung

<https://getbootstrap.com/>

- 1)CDN-System und Zugriff per Internet
- 2)Download und Einbinden in den lokalen Ordner.

### 1)CDN-System

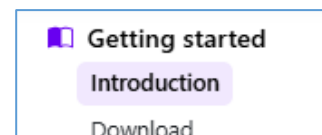
Was ist ein CDN (Content Delivery Network)?

Ein Content Delivery Network (CDN) ist ein Netz von Servern in verschiedenen Rechenzentren auf der ganzen Welt. Es stellt eine Möglichkeit dar, um auf Webanwendung weltweit zugreifen bzw. entwickeln zu können. Wenn man auf ein solches Netzwerk zurückgreift, verbindet man sich mit einer Vielzahl weiterer Server. Rufen Benutzer nun einen bestimmten Inhalt Ihrer Website auf, ermittelt ein Request-Routing-System, welcher Replica-Server im Content Delivery Network sich am besten für die **Auslieferung der Daten** eignet.

#### Info: Platzierung <script> für JavaScript:

Die Position der <script>-Elemente hat einen Einfluss darauf, wie schnell der Ladevorgang einer Website erscheint. Die <script>-Dateien z.B. „bootstrap.min.js“, sollten nicht im <head>-Bereich untergebracht werden, da das Herunterladen einer JavaScript-Datei andere Downloads anhält und sogar mit dem Seitenaufbau wartet, bis die Datei heruntergeladen ist. Daher sollte man erst am Ende der Seite, knapp vor dem schließenden <body>-Tag diese Datei unterbringen. Somit kann sie den Aufbau der Seite nicht beeinträchtigen.

Außerdem ist beim Ausführen des Skripts das DOM bereits geladen.



Klicke per „docs“ zum Punkt 2:

2. **Include Bootstrap's CSS and JS.** Place the <link> tag in the <head> for our CSS, and the <script> tag for our JavaScript bundle (including Popper for positioning dropdowns, popovers, and tooltips) before the closing </body>. Learn more about our [CDN links](#).

```
<!doctype html>
<html lang="en">
```

Kopiere aus dem Skript genau 2 Zeilen:

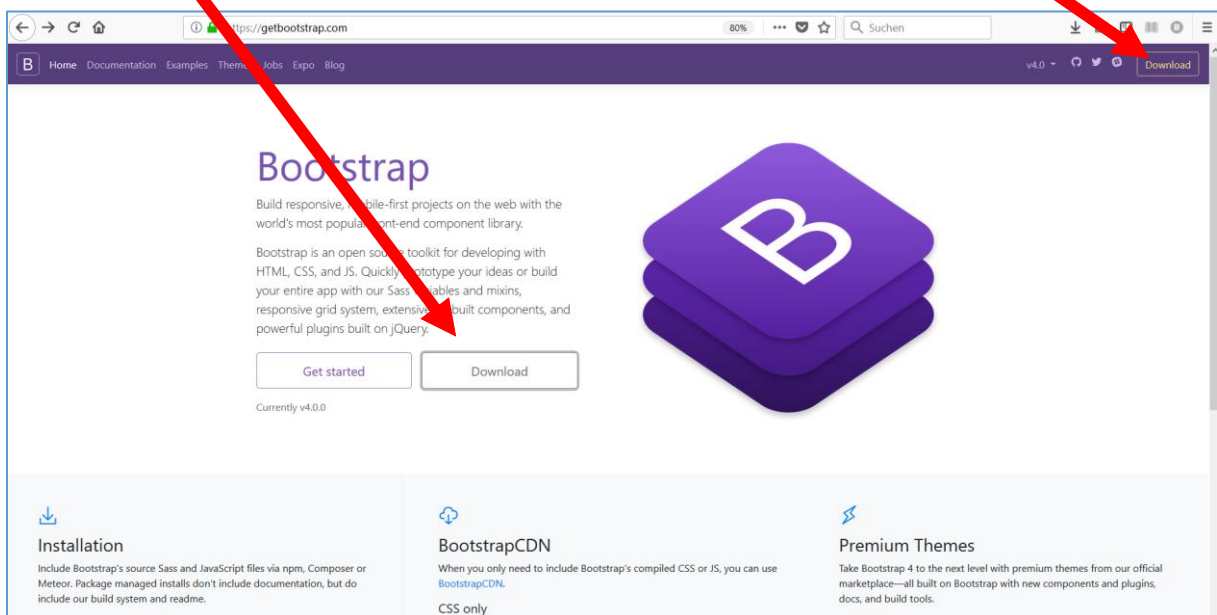
```
<title>Bootstrap demo</title>  
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dist/css/bootstrap.min.css" re
```

```
<h1>Hello, world!</h1>  
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dist/js/bootstrap.bundle.min.  
</body>
```

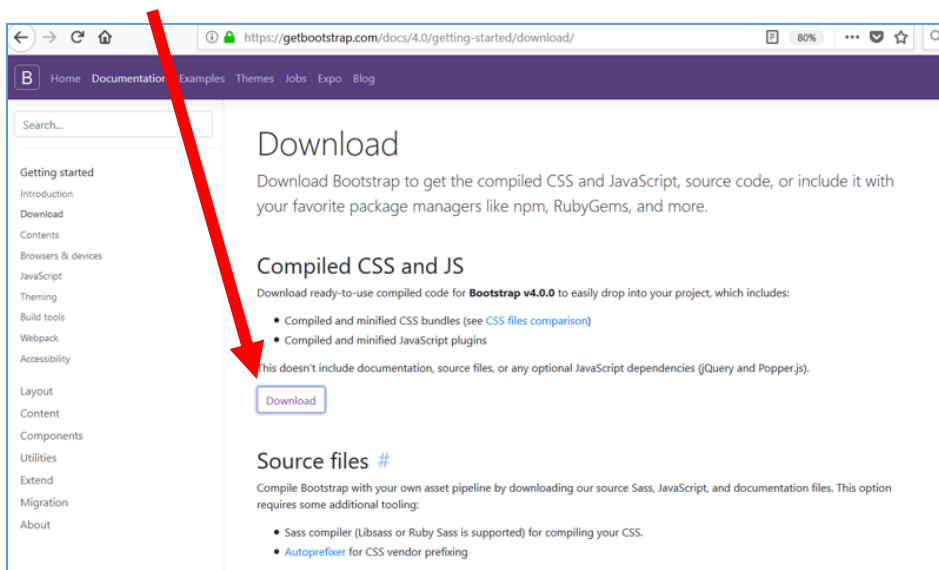
## 2) Download

Klick auf „Download“.

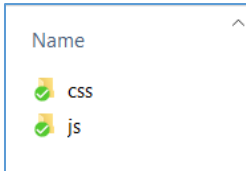
Nicht hier



Dann:



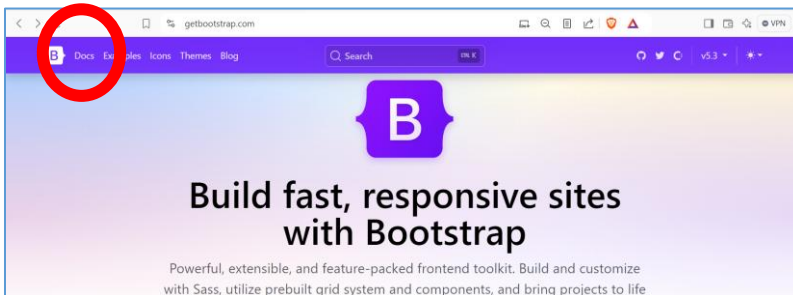
Entpacke und kopiere in den Ordner, in dem auch die Website entstehen soll.



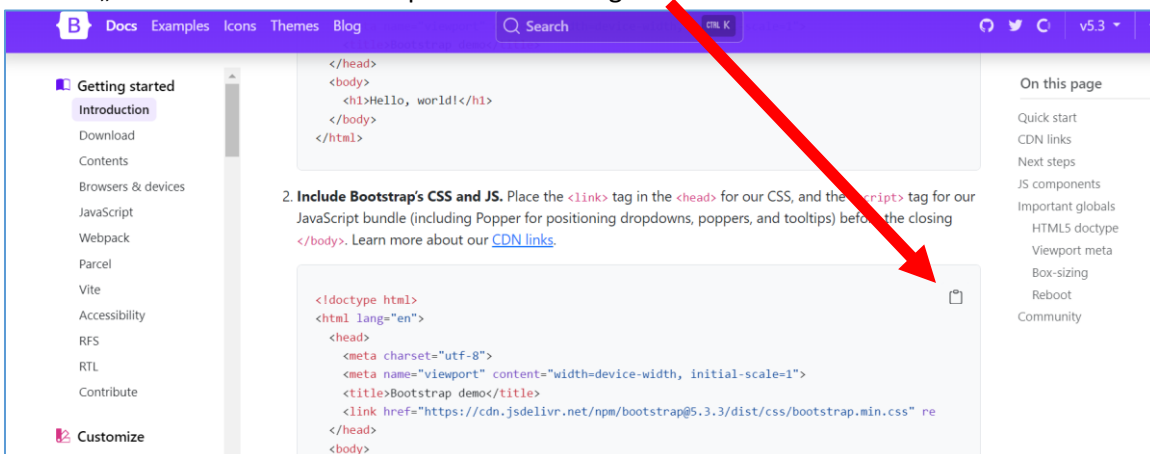
Erstelle eine Datei „index.html“ und befülle diese mit dem vorgegebenen Basic Template von der Website [www.getbootstrap.com](http://www.getbootstrap.com) durch „Kopieren und Einfügen“: Navigation unter „Starter template“ die einfache Datei kopieren.

Ziel: man erspart sich das Tippen, da hier bereits alle Links zu den passenden Ordner (css, js) definiert sind.

Klicke auf „Docs“:



Dann in „Quick starts“ Punkt 2. Kopiere rechts den ganzen Inhalt.



Füge den Inhalt in deine „index.html“ ein.

```
index.html
1 <!doctype html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <title>Bootstrap demo</title>
7 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" re
8 </head>
9 <body>
10 <h1>Hello, world!</h1>
11 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.
12 </body>
13 </html>
```

## Theorie:

Es gibt statt vier nun **5 Viewport-Breiten**

- xs extra small (schmäler als 576px)
- sm small (breiter als 576px)
- md medium (breiter als 768px)
- lg large (breiter als 992px)
- xl extra large (breiter als 1200px)

Breakpoint	Class infix	Dimensions
X-Small	<i>None</i>	0–576px
Small	<i>sm</i>	≥576px
Medium	<i>md</i>	≥768px
Large	<i>lg</i>	≥992px
Extra large	<i>xl</i>	≥1200px
Extra extra large	<i>xxl</i>	≥1400px

Dabei wird die kleinste Viewport-Größe nicht mit dem Begriff „col-xs“ angesprochen, sondern nur mit „col-1“ bis „col-12“. Die anderen Klassennamen behalten ihren Namen:

col-sm bis col-xl

## Das Raster im Detail - GRID-System nutzen (Spalten)

Bootstrap arbeitet mit einem **eigenen Spaltensystem**.

Das Raster wird durch zwölf (12) gleichbreite Spalten gebildet. Vordefinierte Klassen können benutzt werden, um Elemente ab einer bestimmten Spalte und über eine Anzahl Spalten hinweg zu platzieren.

Innerhalb der Spalten lassen sich Reihen festlegen, um eine wechselnde Nutzung der Spaltenbreiten zu erlauben. Die Vorgehensweise sollte sich an einigen wenigen Regeln orientieren:

- **Reihen (.row)** müssen in einem Container sein (*.container* (feste Breite) oder *.container-fluid* (volle Breite)).
- Reihen werden benutzt, um mehrere Elemente horizontal nebeneinander zu platzieren
- **Spalten (.col-x)** werden in Reihen platziert. Nur Spaltenelemente sind unmittelbare Kinderelemente von Reihen.
- Spaltenabstände (Lücke zwischen Spalten) werden mit *padding*-Regeln definiert. Diese Abstände werden links von der ersten Spalte und rechts der letzten Spalte mit negativen Abständen (*margin*) ausgeglichen. Dadurch ist Inhalt außerhalb des Rasters linksbündig gleich ausgerichtet.

## Der Aufbau der Spaltenklassen ist einfach:

- Die Einleitung beginnt mit *col-*
- Der mittlere Teil bestimmt die Zuständigkeit für Bildschirmbreiten (*xs, sm, md, lg, xl*) – die Breakpoints
- Die Zahl am Ende bestimmt die Anzahl der Spalten (1 bis 12)

Wenn man die Klasse `.col-md-4` nutzt, kann man also 3 solchermaßen dekorierte Elemente nebeneinander anordnen ( $3 * 4 = 12$ ).

Die Breitendefinition gilt, bis sie überschrieben wird. Wenn man `.col-md-2` nutzt, `.col-lg-2` aber nicht, gilt "md" auch für sehr große Geräte.

Aus den Breiten ergeben sich die Dimensionen der 12 Spalten.

Die Grenze von 12 Spalten ist nicht absolut und Inhalte werden nie abgeschnitten. Was nicht in das Raster passt, wird einfach auf die nächste virtuelle Zeile verschoben. Elemente, die in einer Spaltendefinition stehen, werden als Ganzes verschoben.

## Grid system

Info: <https://getbootstrap.com/docs/4.0/layout/grid/>

	xs	sm	md	lg	xl	xxl
	<576px	≥576px	≥768px	≥992px	≥1200px	≥1400px
<b>Container max-width</b>	None (auto)	540px	720px	960px	1140px	1320px
<b>Class prefix</b>	<code>.col-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>	<code>.col-xl-</code>	<code>.col-xxl-</code>
<b># of columns</b>	12					
<b>Gutter width</b>	1.5rem (.75rem on left and right)					

### Wo steht der „Viewpoint“?

In den Meta-Tags findet man diese Anweisung, dass auf kleinere Oberflächen Rücksicht genommen wird:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
1 <!doctype html>
2 <html lang="en">
3 <head>
4 <!-- Required meta tags -->
5 <meta charset="utf-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
7
8 <!-- Bootstrap CSS -->
```

**Wichtige Strategie:** immer auf 12 Spalten hinarbeiten z.B.  $4+8=12$ , d.h. 2 Spalten, wobei bei der „md“ Darstellung die rechte doppelt so breit ist wie die linke.

### Flexbox:

Der CSS-Layout-Standard „Flexbox“ erlaubt flexible Boxen zu definieren und anzuordnen.

- Für die **horizontale** Zentrierung ist Klasse „**justify-content-center**“ zuständig.
- Die **vertikale** Anordnung lässt sich über „**align-items-**“ umsetzen. „Align-items-center“ etwa zentriert die Elemente in der Vertikalen. Die Klassennamen heißen wie die intern verwendeten Flexbox-Eigenschaften.

## **<div> für die Gestaltung**

Bootstrap nutzt die DIV-Elemente, um eine Basis für das Raster zu finden. DIV kann man auch als „Container“ bezeichnen und das kann man mehrfach auf der Seite benutzen. In einem Container können beliebig viele Reihen (rows) verwendet werden. In jeder Reihe werden einzelne Spalten (cols) und deren Breite beliebig festgelegt.

```
<div class="row">  
  ...  
</div>
```

**Info:** Container dieser Art liefern ein festes, zentriertes und responsives Raster. Das heißt, die Breite passt sich in Sprüngen dem Gerät an und bleibt dann innerhalb eines Bereiches stabil. Die Sprungbereiche werden auch als “Breakpoints” bezeichnet – Umbruchpunkte.

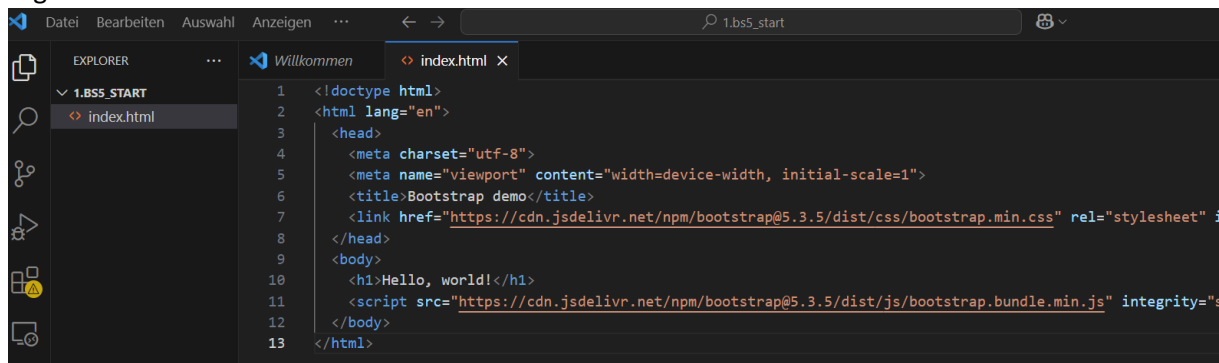
## Übung: index.html

- Erstelle einen Ordner mit dem Namen „1.bs5\_start“ im Windows-Explorer.
- Öffne diesen mit VisualStudioCode per „Ordner öffnen“

Ordner öffnen...

- Lege eine neue Datei „index.html“ an.
- Hole dir den Startcode von [getBootstrap.com](https://getbootstrap.com) und füge ihn ein. Somit werden wir mit den Links per CDN arbeiten und nicht die CSS- und JS-Datei downloaden. Nachteil ist aber, dass wir laufend den Traffic erhöhen mit jedem Arbeiten am Code.

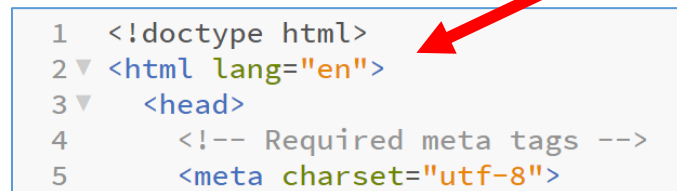
Ergebnis:



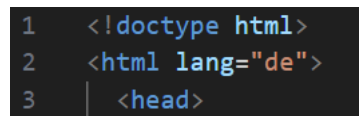
```
1 <!doctype html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6     <title>Bootstrap demo</title>
7     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dist/css/bootstrap.min.css" rel="stylesheet">
8   </head>
9   <body>
10    <h1>Hello, world!</h1>
11    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dist/js/bootstrap.bundle.min.js" integrity="
12  </body>
13 </html>
```

Sprache:

- Ändere auf die Sprache „deutsch“: `<html lang="de">` statt „en“
- Ändere den Titel auf „Hak Mistelbach“



```
1 <!doctype html>
2 <html lang="en">
3   <head>
4     <!-- Required meta tags -->
5     <meta charset="utf-8">
```



```
1 <!doctype html>
2 <html lang="de">
3   <head>
```

**Spalten (column):**

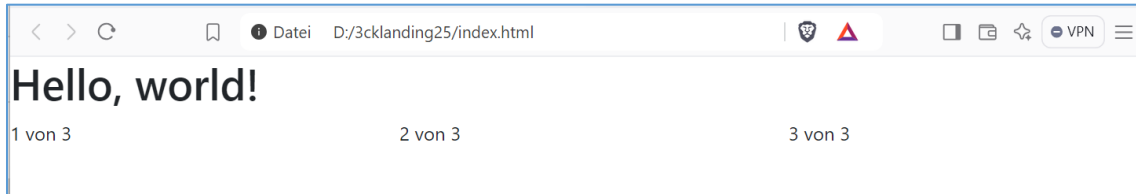
<https://getbootstrap.com/docs/5.3/layout/columns/>

Ein weiterer Vorteil von Flexbox bei Bootstrap ist es, dass man nicht mehr bei allen Spalten explizit die Breite angeben muss. Wenn man Rasterklassen ohne Zahl schreibt, nimmt jedes Element gleich viel Platz ein. Im folgenden Beispiel erhält also jede der Spalten ein Drittel des Platzes:



```
9 <body>
10 <h1>Hello, world!</h1>
11
12 <div class="row">
13   <div class="col">1 von 3</div>
14   <div class="col">2 von 3</div>
15   <div class="col">3 von 3</div>
16 </div>
17
```

Ergebnis:



Tipp: mit VSCode lässt sich diese Schreibweise sehr einfach abkürzen, nämlich mit folgender Kurzschreibweise:

```
.col
```

.col und die Entertaste erzeugt folgende Zeile:

```
<div class="col"></div>
```

Will man 3 Spalten erzeugen kann man diesen Vorgang mit 3 multiplizieren und dann Enter drücken:

```
11 <h1>Hello, world!</h1>
12
13 .col*3
14
```

Ergebnis:

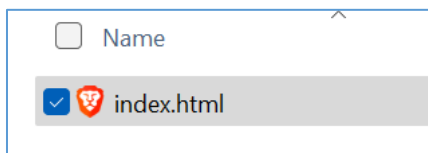
```
11 <h1>Hello, world!</h1>
12
13 <div class="col"></div>
14 <div class="col"></div>
15 <div class="col"></div>
16
```

### Änderung:

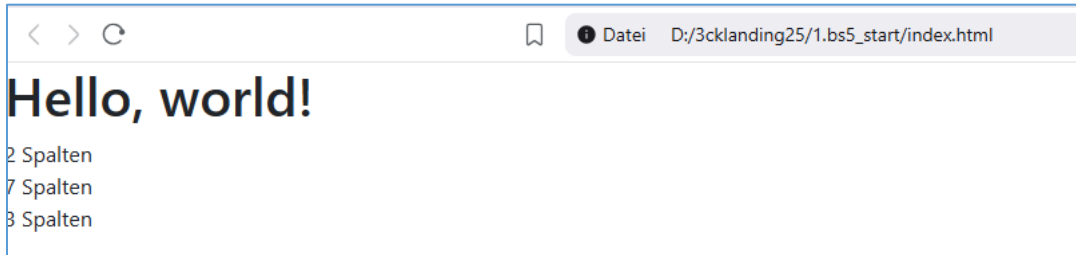
Will man die Spalten selbst einteilen, dann folgt dem „col“ mit Bindestrich die entsprechende Zahl.

```
12 <div class="row">
13   <div class="col-2">2 Spalten</div>
14   <div class="col-7">7 Spalten</div>
15   <div class="col-3">3 Spalten</div>
16   <!-- in Summe 12 Spalten -->
17 </div>
```

Öffne die Datei im Browser durch Klick auf die Datei im Windows Explorer:



Ergebnis:

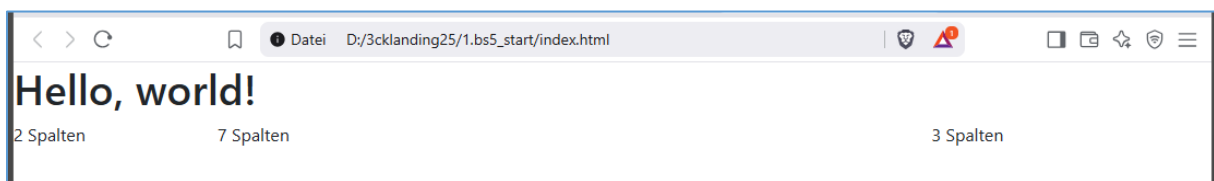


Die Spalten sind nicht in einer Reihe, das holen wir jetzt nach:

Erstelle über der ersten <col> eine neue DIV mit der Klasse „row“, also Reihe. Das schließende DIV muss dann noch ausgeschnitten und nach der letzten <col> eingefügt werden, damit es auf alle <col> wirken kann:

```
11 <h1>Hello, world!</h1>
12
13 <div class="row"></div>
14 <div class="col-2">2 Spalten</div>
15 <div class="col-7">7 Spalten</div>
16 <div class="col-3">3 Spalten</div>
17 </div>
```

Speichern und ansehen:



### Spalten pro <row> = 12

Die Summe soll 12 betragen. Hier kein Problem, da  $2 + 7 + 3 = 12$

**Info: eine oder mehrere Spalten rechts, die nicht in Anspruch genommen werden, verfallen einfach. Ist somit auch kein Problem, wenn es eine <row> dazu gibt.**

### Text einfügen:

([www.blindtextgenerator.de](http://www.blindtextgenerator.de) – lorem ipsum)



Kopiere die ersten Zeilen und füge sie in alle Spalten ein:

```

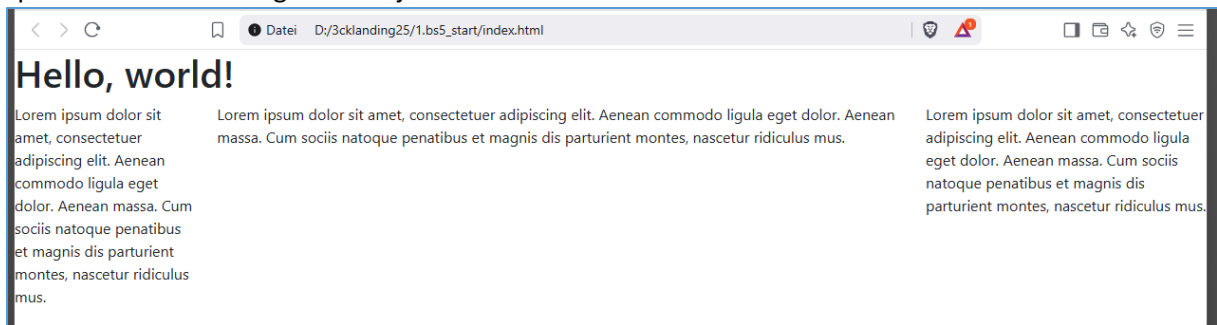
13 <div class="row">
14 <div class="col-2">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean d
15 <div class="col-7">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean d
16 <div class="col-3">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean d
17 </div>

```

Text auch zum Kopieren:

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. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem.

Speichern und dann Ergebnis bis jetzt:



## Verbesserung: Abstand links und rechts

Damit der Text und die Spalten nicht am Bildschirmrand kleben, kann ein einfaches <div> mit der passenden Klasse „container“ dies automatisch lösen:

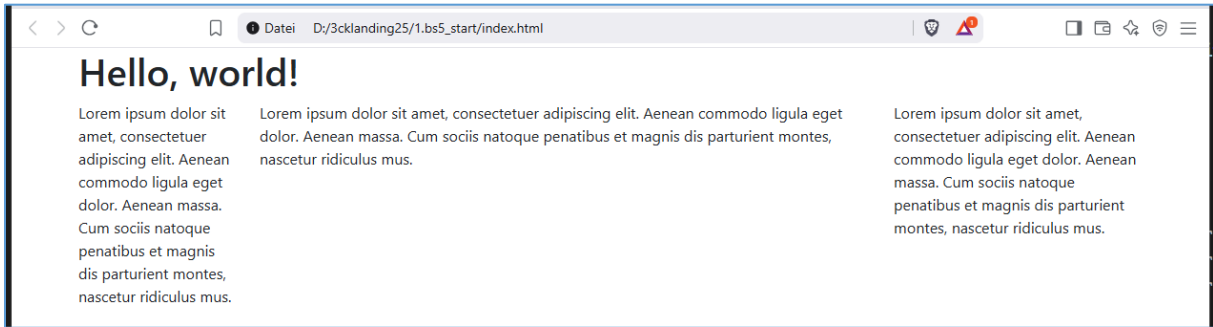
Diese DIV soll alles bis jetzt Vorhandene im <body> miteinschließen:

```

9 <body>
10 <div class="container">
11 <h1>Hello, world!</h1>
12
13 <div class="row">
14 <div class="col-2">Lorem ipsum dolor sit amet, consectetur adipisc
15 <div class="col-7">Lorem ipsum dolor sit amet, consectetur adipisc
16 <div class="col-3">Lorem ipsum dolor sit amet, consectetur adipisc
17 </div>
18
19 </div>
20 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dist/js/b
21 </body>

```

Ergebnis:



## Übung:

Erstelle unter der ersten <row>, also nach dem </row> in ca. Zeile 18 eine

- neue <row> mit
- NUR 2 <col> mit je 4 Spalten entstehen.
- Der Rest wird „verfallen“.

Überschrift <h2> in jedem der beiden <col>: „Wochentag“ und „Mittagsmenü“

Nutze die Schnellschreibweise:

```
18  
19 .row  
20  
21  
22
```

```
18  
19 <div class="row">  
20   .col-4*2  
21 </div>
```

```
16 <div class="col-3">Lorem ipsum dolor sit amet, consectetur  
17 </div>  
18  
19 <div class="row">  
20   <div class="col-4">  
21     <h2>Wochentag</h2>  
22   </div>  
23   <div class="col-4">  
24     <h2>Mittagsmenü</h2>  
25   </div>  
26 </div>  
27  
28 </div>  
29 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/di
```

Als Abstand zu den vorherigen Lorem ipsum Text setzte zwischen dem Ende der ersten <row> und der neuen <row> einige <br> hinein:

```
16 <div class="col-3">Lorem ipsum do  
17 </div>  
18 <br><br><br>  
19 <div class="row">  
20   <div class="col-4">  
21     <h2>Wochentag</h2>
```

## Spalten nicht „belegen“ - „offset“ schalten

Manchmal soll ein Spaltenbereich nicht angezeigt werden und somit die Spalten weiter rechts beginnen.

<https://getbootstrap.com/docs/5.3/layout/columns/#offsetting-columns>

Dafür wird einer Spaltenbreite, die dem „offset“ folgt, ein „offset“ mit der gewünschten Spaltenbreite hinten nachgeschrieben:

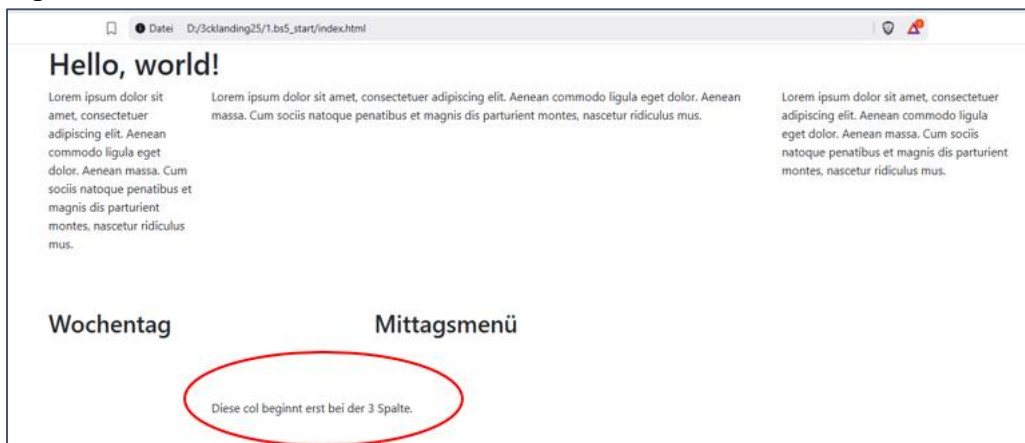
```
<div class="col-4 offset-2">Diese col beginnt erst bei der 3 Spalte.</div>
```

### Übung:

Erstelle unter der letzten </row> eine neue mit folgendem Inhalt:

```
24         <h2>Mittagsmenü</h2>
25     </div>
26 </div>
27 <br><br><br>
28 ✨ <div class="row">
29     <div class="col-4 offset-2">Diese col beginnt erst bei der 3 Spalte.</div>
30 </div>
31
32 </div>
33 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dist/js/bootstrap.bundle
```

### Ergebnis:



## Fußzeile anlegen – footer

Erstelle eine Fußzeile und verwende den „semantischen“ HTML5-Begriff `<footer>`.

Davor sollen 3 `<br>` für Abstand sorgen und darunter eine horizontale Linie ein egute Abtrennung bringen:

**Ziel:** ganz rechts soll in einer Spaltenbreite von 3 ein „Copyright digBiz Klasse“ stehen. Links davon soll frei sein (offset-9).

```
30     </div>
31     <br><br><br>
32     <hr>
33     <footer>
34     ... <div class="col-3 offset-9">Copyright digBiz Klasse</div>
35
36     </footer>
37
38 </div>
39 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dist/js/bootst
40 </body>
```

Ergebnis:



Tipp: „&copy;“ ergibt das Symbol

```
<div class="col-3 offset-9">&copy; digBiz Klasse</div>
```

## Übung:

Ändere den Footer so, dass links in den ersten 2 Spalten steht „Impressum“ und das Copyright digBiz Klasse dort bleibt, wo es ist.

Dazwischen muss also ein „offset“ verwendet werden.

ABER: es muss in einer Zeile stehen! Daher muss auch eine <row> eingefügt werden und die Zahl vom<offset> reduziert werden, da ja vorne 2 Spalten dazukommen.

Ziel:

Wochentag	Mittagsmenü
Diese col beginnt erst bei der 3 Spalte.	
Impressum	© digBiz Klasse

```
32 <hr>
33 <footer>
34   <div class="row">
35     <div class="col-2">Impressum</div>
36     <div class="col-3 offset-7">&copy; digBiz Klasse</div>
37   </div>
38 </footer>
39
40 </div>
41 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dis
```