

Bootstrap Framework – Theorie



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 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 Programzeilen 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.

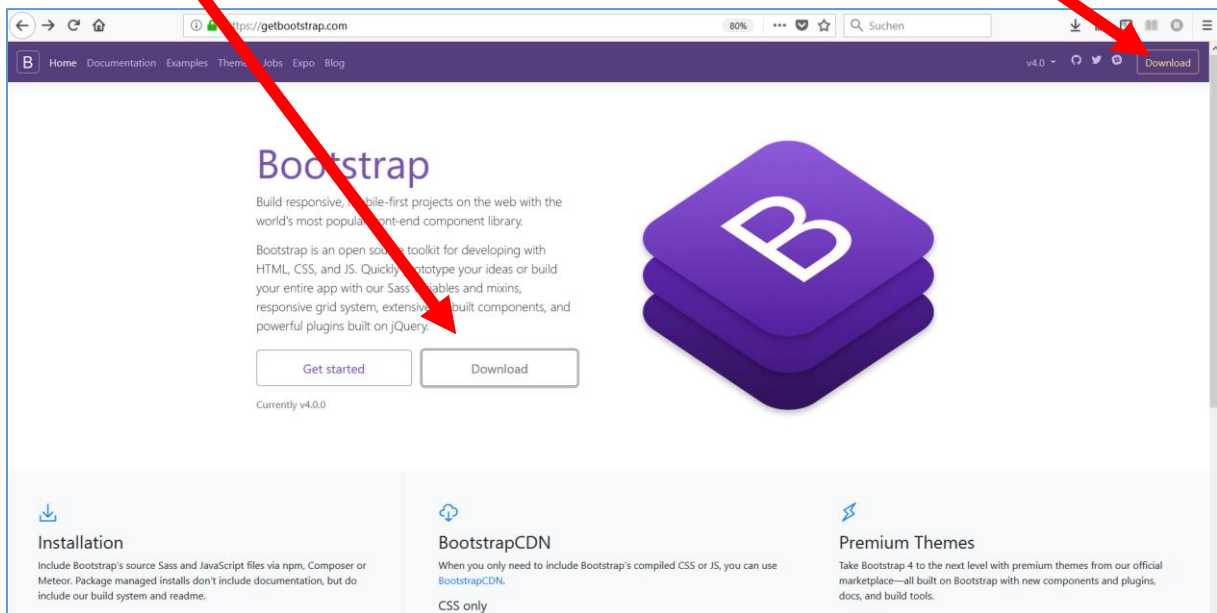
Bootstrap baut auf dem Mobile-First-Prinzip auf, welches die Anpassung der Inhalte auf mobilen Geräten in den Vordergrund stellt. Das Open-Source Programm arbeitet mit einem 12 Spalten 960px Rastersystem (Grid-System) - somit passen sich die Spalten in Abhängigkeit der Breite des Gerätes an und ermöglicht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

Start:

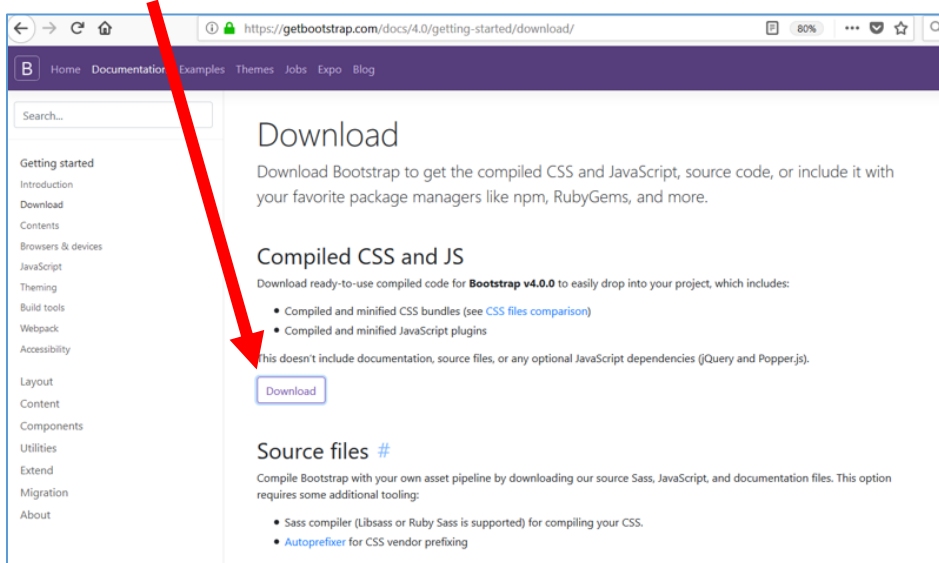
<https://getbootstrap.com/>

Klick auf „Download“.

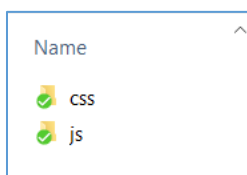
Nicht hier



Dann:



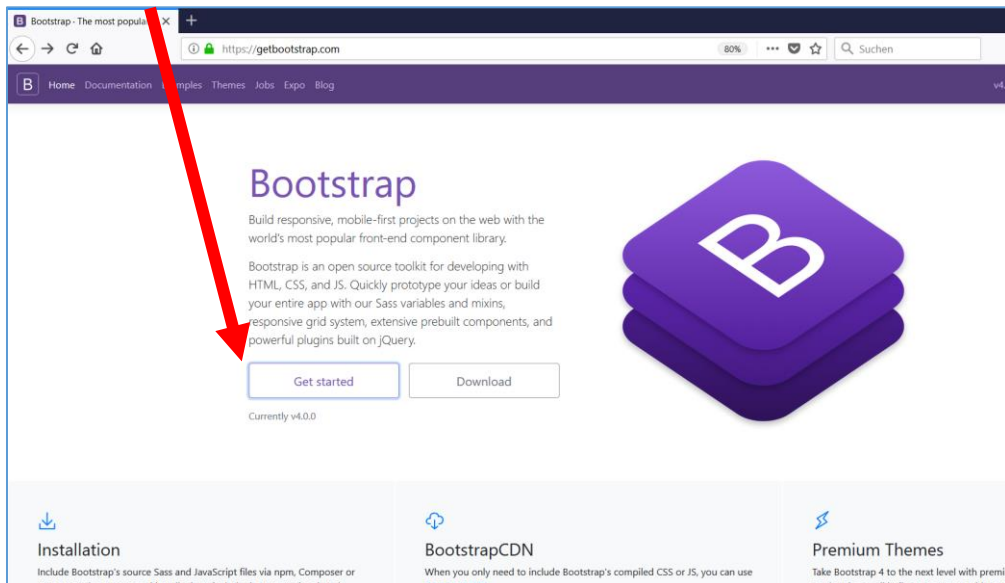
Entpacke und kopiere in den Ordner, in dem auch die Website entstehen soll.



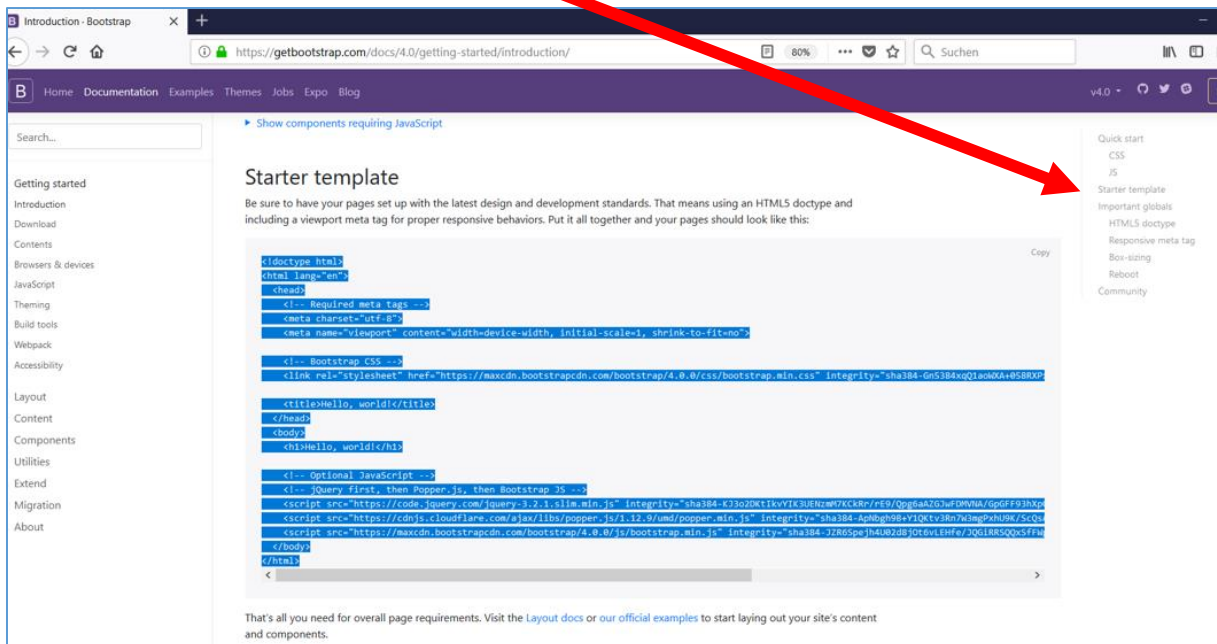
Erstelle eine Datei „index.html“ und befülle diese mit dem vorgegebenen Basic Template von der Website www.getbootstrap.com (oder www.holdirbootstrap.de) durch „Kopieren und Einfügen“: Navigation unter „Starter template“ die einfache Datei kopieren.

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

Klicke auf „Get started“:



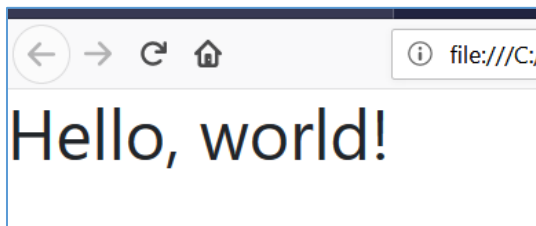
Kopiere die Datei in die „index.html“. Nutze rechts oben den Button „Copy“ und in der rechten Navigation den Link „Starter template“



Erstelle eine „index.html“ und füge den oben kopierten Code ein und speichere es.

```
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 -->
9 <link rel="stylesheet"
  href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
  integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJLSAwiGgFAW/dAiS6JXm"
  crossorigin="anonymous">
10
11 <title>Hello, world!</title>
12 </head>
13 <body>
14 <h1>Hello, world!</h1>
15
16 <!-- Optional JavaScript -->
17 <!-- jQuery first, then Popper.js, then Bootstrap JS -->
18 <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-
  KJ3o2DKtIkVYIK3UENzmM7KCKRr/rE9/Qpg6AAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous">
  </script>
19 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
  integrity="sha384-ApNbgh9B+Y1QKt3Rn7W3mgPxxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
  crossorigin="anonymous"></script>
20 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
  integrity="sha384-JZR6Spejh4U02d8j0t6vLEHfe/JQGiRRSQQxSfFFWpi1MquVdAjyUar5+76PVCmYL">
  </script>
```

Das Ergebnis sieht folgendermaßen aus:



Theorie:

Für die Version Bootstrap 4 wurde der Code deutlich überarbeitet.

Viele der Komponenten von Bootstrap benötigen zusätzlich die JavaScript-Bibliotheken jQuery und Popper.js.

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)

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

Info: jQuery UI

Auch wenn jQuery die Grundlage der Bootstrap-Komponenten ist, steht jQuery UI in Konkurrenz zu Bootstrap und sollte auf keinen Fall gleichzeitig eingesetzt werden.

Das Raster im Detail

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-xx-n)** 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 (*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/>

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
Max container width	None (auto)	540px	720px	960px	1140px
Class prefix	<code>.col-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>	<code>.col-xl-</code>
# of columns	12				
Gutter width	30px (15px on each side of a column)				
Nestable	Yes				
Column ordering	Yes				

Wo steht der „Viewpoint“?

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

```
<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:

Bootstrap 3 hat sein Raster im Wesentlichen mit „float“ realisiert. Ganz anders der neue CSS-Layout-Standard „Flexbox“. Er erlaubt flexible Boxen zu definieren und anzuordnen.

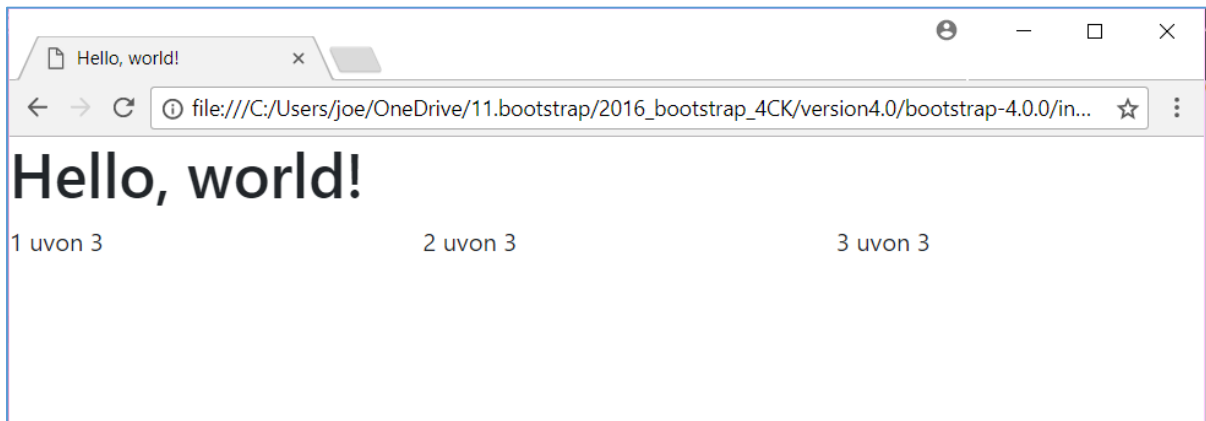
- Für die horizontale Zentrierung ist Klasse „justify-content-center“ zuständig.
- Die vertikale Anordnung lässt sich über „align-items-“ umsetzen. „Align-items-center“ etwa zentriert die Elemente in der Vertikalen. Die Klassennamen heißen übrigens wie die intern verwendeten Flexbox-Eigenschaften.

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:

```
21
22 ▾   <div class="row">
23       <div class="col">1 u von 3</div>
24       <div class="col">2 u von 3</div>
25       <div class="col">3 u von 3</div>
26   </div>|
27 </body>
```

```
<div class="row">
  <div class="col">1 u von 3</div>
  <div class="col">2 u von 3</div>
  <div class="col">3 u von 3</div>
</div>
```

Ergebnis:

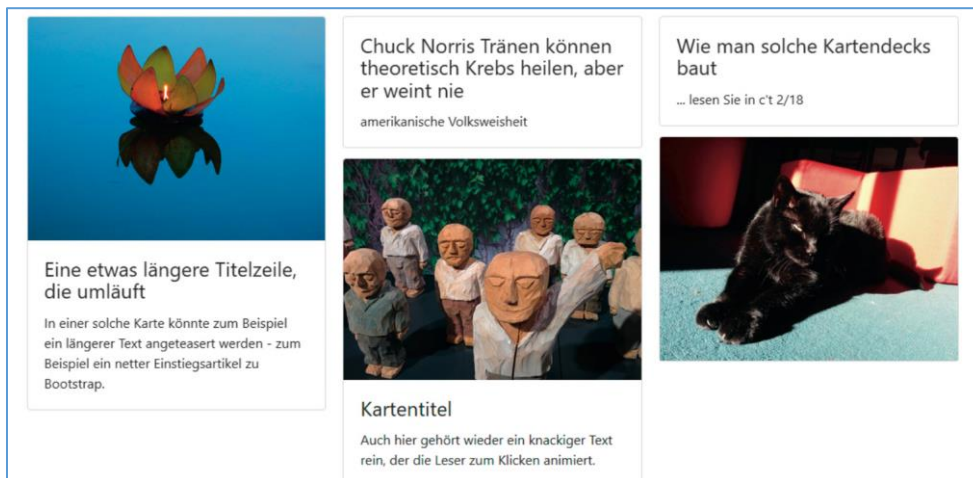


Macht man im Beispiel noch ein weiteres Element mit `<div class="col">` ergänzt, teilen sich 4 Spalten den Platz, von denen jedes ein Viertel des Platzes belegt.

Damit lässt sich auch sehr einfach die Reihenfolge festlegen bzw. ändern. Ein Element mit der Klasse „1“ steht vor einem Element mit „2“.

Cards:

Karten sind ein häufig genutztes Stilmittel moderner Web- und Mobil-Layouts, zum Beispiel für Übersichten.



Es wurde die Komponente „cards“ eingeführt. Cards (Karten) können Bilder, Texte, Listen oder Buttons enthalten, selbst Navigationsleisten innerhalb von Karten sind möglich. Den Kopf- und Fußbereich kann man optisch absetzen, die Karten selbst auch einfärben. Wenn man mehrere Karten nebeneinander platziert, bekommen diese – Flexbox sei Dank – alle die gleiche Höhe.

Wie in Bootstrap üblich, bestimmen vorgegebene Klassen die Gestaltung. Das umfassende Element erhält die Klasse „card“, ein oben angeordnetes Bild „card-img-top“. Der eigentliche Inhalt steht innerhalb von „card-body“ und kann einen Titel „card-title“ und einen Text „card-text“ enthalten:

```
29 <div class="card">
30   
31   <div class="card-body">
32     <h4 class="card-title">Kartentitel</h4>
33     <p class="card-text">Text für die Karte</p>
34   </div>
35 </div>
```

```
<div class="card">
  
  <div class="card-body">
    <h4 class="card-title">Kartentitel</h4>
    <p class="card-text">Text für die Karte</p>
  </div>
</div>
```

Weitere Möglichkeiten mit „cards“:

Karten lassen sich in Form von „card-group, card-deck und card-column“ gruppieren.

- card-group erzeugt aus mehreren Karten ein einzelnes, zusammenhängendes Element, bei denen die Karten jeweils die gleiche Höhe und Breite zugewiesen bekommen

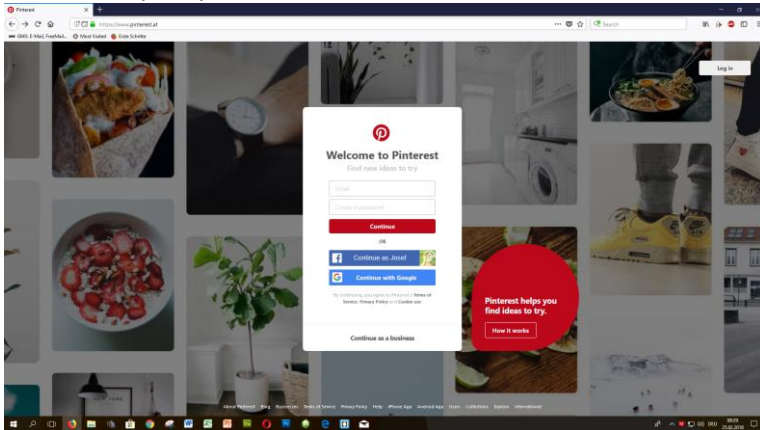
- card-deck hat denselben Effekt wie Gruppen, allerdings mit dem Unterschied, dass die Karten nicht zusammenhängen.
- card-column wiederum organisiert Karten optisch ähnlich wie die Inhalte bei www.pinterest.com, allerdings von oben nach unten und von links nach rechts.

```

<div class="card-deck">
<div class="card">
...
</div>
<div class="card">
...
</div>
<div class="card">
...
</div>
</div>

```

Beispiel: pinterest.com



Um Inhaltselemente mit einer optionalen Unterzeile zu erstellen, gibt es in HTML5 das `<figure>`- und das `<figcaption>`-Element. Bootstrap unterstützt auch die Klassen `figure`, `figure-img` und `figure-caption`.

Unterschied zu Bootstrap 3:

Gestrichen wurden:

- Icon-font-Glyphicons – ersetzt durch Font Awesome
- jQuery-Plugin-Affix – die horizontale Leiste scrollt zuerst mit, bleibt aber dann, bevor man sich aus dem sichtbaren Bereich bewegt, oben in einer fixen Position stehen. Bei Bootstrap 4 wird stattdessen empfohlen, den CSS-Stil „position: sticky“ zu verwenden. Allerdings funktioniert dies im IE nicht und in Edge erst ab Version 16. Dafür benötigt man also noch ein Polyfill, das diese Fähigkeiten bei Edge nachrüstet.
- Panels (gerahmte Boxen) – ersetzt durch Cards-Komponente
- Wells (runde Boxen mit Hintergrundfarbe) – ersetzt durch Cards-Komponente
- Thumbnails (Bilder mit zusätzlichen Inhalten) – ersetzt durch Cards-Komponente

- Man muss nicht mehr doppelt schreiben:
 - Früher: `<div class="col col-md-5">`
 - Jetzt: `<div class="col-md-5">`
-

Images

- Renamed `.img-responsive` to `.img-fluid`.
- Renamed `.img-rounded` to `.rounded`
- Renamed `.img-circle` to `.rounded-circle`

- Offset
 - Früher: `col-md-offset-4`
 - Jetzt: `offset-md-4`