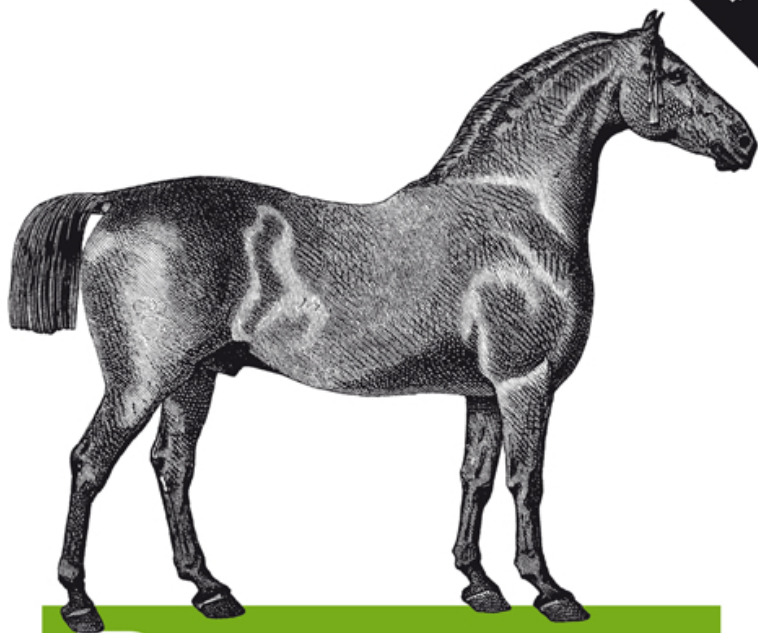


O'REILLY®

Behandelt
Bootstrap 4



Bootstrap

kurz & gut

O'REILLYS TASCHENBIBLIOTHEK

Jörg Krause

Papier
plus⁺
PDF.

Zu diesem Buch – sowie zu vielen weiteren O'Reilly-Büchern –
können Sie auch das entsprechende E-Book im PDF-Format
herunterladen. Werden Sie dazu einfach Mitglied bei oreilly.plus⁺:

www.oreilly.plus

Bootstrap

kurz & gut

Jörg Krause

O'REILLY®

Jörg Krause

Lektorat: Ariane Hesse

Fachgutachter: Jörgen W. Lang, Jens Meiert

Korrektorat: Sibylle Feldmann, www.richtiger-text.de

Herstellung: Stefanie Weidner

Umschlaggestaltung: Michael Oréal, www.oreal.de

Satz: III-satz, www.drei-satz.de

Druck und Bindung: Media-Print Informationstechnologie, www.mediaprint-druckerei.de

Bibliografische Information Der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN:

Print 978-3-96009-087-8

PDF 978-3-96010-231-1

ePub 978-3-96010-232-8

mobi 978-3-96010-233-5

Dieses Buch erscheint in Kooperation mit O'Reilly Media, Inc. unter dem Imprint »O'REILLY«. O'REILLY ist ein Markenzeichen und eine eingetragene Marke von O'Reilly Media, Inc. und wird mit Einwilligung des Eigentümers verwendet.

1. Auflage

Copyright © 2018 dpunkt.verlag GmbH

Wieblinger Weg 17

69123 Heidelberg

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Die Informationen in diesem Buch wurden mit größter Sorgfalt erarbeitet. Dennoch können Fehler nicht vollständig ausgeschlossen werden. Verlag, Autoren und Übersetzer übernehmen keine juristische Verantwortung oder irgendeine Haftung für eventuell verbliebene Fehler und deren Folgen.

5 4 3 2 1 0

Inhalt

Vorwort	IX
1 Einführung in Bootstrap	1
Grundlegendes zu Bootstrap 4	1
Installation	2
Struktur der CSS-Dateien	4
Seitenaufbau	6
Browserunterstützung	7
ARIA	8
Optimierung	10
Hilfsklassen	12
Reboot	13
2 Struktur der Seite	15
Einführung	15
Das Rastersystem	16
Das Flex-Raster	28
3 Typografie	35
Überschriften	36
Text und Textelemente	37
Listen	43
Tabellen	46
Hilfsklassen	57

4	Formulare	61
	Struktur eines Formulars	61
	Eingabeelemente	74
	Schaltflächen	94
5	Weitere Bausteine	101
	Symbole	101
	Responsive Bilder	105
	Eingebettete Quellen	107
	Farben und Hintergründe	107
	Ausrichtung von Elementen im Dokumentenfluss	110
	Inhalte anzeigen und verstecken	111
6	Komponenten	113
	Aufklappmenüs (dropdown)	113
	Werkzeugleisten (toolbar)	118
	Schaltfläche mit Menü (button group)	121
	Navigation (nav, navbar)	125
	Pfadnavigation (breadcrumb)	136
	Seitenweises Blättern (pagination)	137
	Kennzeichnungen (tags)	140
	Jumbotron (jumbotron)	143
	Seitenüberschriften (page header)	144
	Meldungen (alert)	144
	Fortschrittsbalken (progress)	146
	Medien (media)	151
	Allgemeine Listen (list group)	152
	Karten (cards)	157
7	Aktive Komponenten	171
	Einrichtung und Aktivierung	171
	Die Programmierschnittstelle	172
	Übergänge (transition)	174
	Modale Dialoge (modals)	174
	Aufklappmenü (dropdown)	184

Scrollbar-Überwachung (scrollspy)	186
Umschaltbare Tabs (tab)	189
Tooltipps (tooltip)	191
Popover (popover)	194
Meldungen (alert)	199
Interaktive Schaltflächen (button)	200
Inhaltseinblendung (collapse)	201
Bilderkarussell (carousel)	206
Index	209
Über den Autor	213

Vorwort

Dieses Buch beschreibt kompakt und übersichtlich das CSS-Framework Bootstrap 4. Bootstrap entstand ca. 2010 ursprünglich bei Twitter und war von vornherein mit dem Gedanken »mobile first« – zuerst für mobile Geräte – entwickelt worden. Es hat sich seitdem zu einem der besten und bekanntesten Frameworks für die Gestaltung von Websites entwickelt. Begleitet von einer Vielzahl auf Bootstrap aufsetzenden Gestaltungsvorlagen – sogenannten Themes –, bietet es sichere und robuste Stildefinitionen für Alltagsaufgaben.



Bootstrap-Beispiele

Um alle Beispiele nachvollziehen zu können, benötigen Sie eine lauffähige Umgebung zum Erstellen von Webseiten. Nähere Informationen dazu bietet das umfassende Werk für Einsteiger in die Welt der Webentwicklung, »Jetzt werde ich Web-Entwickler!«, vom selben Autor.

Schreibweisen

Skripte sind häufig recht umfangreich. Um sie optisch besser lesbar zu gestalten, habe ich oft zusätzliche Zeilenumbrüche eingebaut, die im Editor Ihrer Entwicklungsumgebung aber nichts zu suchen haben.

Generell wird Programmcode mit einer nicht proportionalen Code-schrift gesetzt.

```
body {  
    color: black;  
}
```

Wenn Sie etwas auf der Kommandozeile oder in einem Dialogfenster eingeben müssen, ist dieser Teil der Anweisung fett gesetzt:

```
$ npm install bootstrap --save
```

Das erste Zeichen ist der Prompt, der nicht mit eingegeben wird. Ich verwende im Buch den Linux-Prompt der Bash-Shell. Die Kommandos funktionieren ausnahmslos unverändert auch in Windows, nur steht dann **C:>** oder etwas Ähnliches am Anfang der Zeile.

Ausdrücke und Befehlszeilen sind häufig mit allen Arten von Zeichen gespickt, und in fast allen Fällen kommt es auf jedes Zeichen an. Oft werde ich die Verwendung bestimmter Zeichen in einem solchen Ausdruck genau erläutern. Wichtige Zeichen werden durch zusätzliche Zeilenumbrüche hervorgehoben.

Im folgenden Beispiel möchte ich etwa das **:**-Zeichen hervorheben und füge deshalb einen Zeilenumbruch nach der ersten öffnenden geschweiften Klammer hinzu, der nicht zwingend ist.

```
a.test {  
  :hover {  
    color: red  
  }  
}
```

Beispiele

Sie finden das Beispielprojekt zu diesem Buch auf GitHub:

- <https://github.com/joergkrause/bootstrap4-kurz-und-gut>

Das Projekt nutzt Musterdateien, die pures HTML enthalten. Zur Nutzung ist es empfehlenswert, die Inhalte mit einem Browser abzurufen. Dazu muss im Stammordner ein Webserver gestartet werden.

Beispiele optimal nutzen

Gehen Sie folgendermaßen vor (für Anfänger ohne Vorkenntnisse):

1. Installieren Sie NodeJs (<https://nodejs.org>).
2. Installieren Sie einen Git-Client passend zum Betriebssystem.

3. Öffnen Sie jetzt eine Konsole (Windows) oder ein Terminal (Linux).
4. Klonen Sie das Git-Verzeichnis (siehe oben) in einen Ordner.
5. Navigieren Sie in diesem Verzeichnis in den neuen Ordner, in dem sich der Klon befindet.
6. Geben Sie `npm install` ein und warten Sie die Installation ab.
7. Geben Sie `npm start` ein. Der Webserver startet auf Port 3000.
8. Öffnen Sie einen Browser und navigieren Sie zu *http://localhost:3000*.
9. Folgen Sie den Anweisungen auf der Website, die nun erscheint.



Dieses Element kennzeichnet einen Hinweis oder Tipp.



Dieses Element weist auf eine Gefahr oder ein mögliches Problem hin.

Einführung in Bootstrap

In diesem Kapitel finden Sie einige grundlegende Informationen zu Bootstrap 4 und zu seiner Installation.

Grundlegendes zu Bootstrap 4

Ein erster Überblick über die wichtigsten Eigenschaften von Bootstrap dient als Grundlage für alle folgenden Kapitel.

Das Einheitensystem

Das Einheitensystem nutzt die Einheit `rem` (CSS) bzw. `em` (Media Queries). Der globale Font, der als Ausgangspunkt dient, wurde von 14 auf 16 Pixel hochgesetzt. Diese Änderung wurde vor allem mit Blick auf die höher auflösenden Displays mobiler Geräte vorgenommen. Die Einheiten `rem` bzw. `em` nutzen die Breite des Buchstaben `M` als Basis der Berechnung und definieren ein relatives Größenverhältnis der Elemente, was gerade auch für responsives Webdesign wichtig ist. Es bietet Vorteile beim Entwickeln eines gefälligen Schriftbilds.

Das Rastersystem

Es gibt folgende Rastervarianten für die verschiedenen Displaygrößen vom Smartphone bis zum Displaybildschirm: Extra Small `xs`, Small `sm`, Medium `md`, Large `lg` und Extra Large `xl`. Das letzte, `xl`, dient der Unterstützung besonders hochauflösender Displays (3.000 x 2.000 Pixel und mehr).

Panels

Die in früheren Versionen benutzten Anzeigeformen Wells, Panels und Thumbnails sind entfallen und werden durch eine neue Komponente mit dem Namen *Card* ersetzt.

Installation

Bootstrap 4 kann über die eigene Website (<http://getbootstrap.com>) und npm (Node Package Manager) heruntergeladen werden. Darüber hinaus können Sie Bootstrap aus den SASS-Quellen auch selbst erstellen und dazu den Quellcode direkt von GitHub (<https://github.com/twbs/bootstrap>) beziehen. Bootstrap nutzt außerdem für die Rohdateien der Cascading Style Sheets die Sprache SASS, ein Präprozessor, der in CSS übersetzt. Die primäre Sprache in Bootstrap 4 zur Erstellung der Stile ist SASS.

CDN

Bootstrap ist via MaxCDN verfügbar. Ein CDN (*Content Delivery Network*) erlaubt einer Website, häufig benötigte öffentliche Dateien von weltweit verteilten Servern abzurufen. Wenn ein Nutzer aus den USA Ihre in Deutschland gehostete Website aufruft, wird das CDN dafür sorgen, dass die Bootstrap-Dateien von einem Server in den USA abgerufen werden. Dies entlastet Ihren Server, die Leitungen des Providers, das Internet insgesamt, und der Benutzer erlebt einen schnelleren Ladevorgang. Im Grunde gewinnen dabei alle. Wenn Sie im Intranet programmieren, sind CDNs dagegen eher ungünstig. Erwarten Sie nur lokale Nutzer in Deutschland, bringt ein CDN keinen Vorteil.

Die Bootstrap-Dateien werden bei der Nutzung des CDN folgendermaßen eingebunden:

```
<!-- Das neueste kompilierte und minimierte CSS -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/
bootstrap/4.0.0/css/bootstrap.min.css">
```

```
<!-- Das neueste kompilierte und minimierte JavaScript -->  
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/  
js/bootstrap.min.js"></script>
```



Bootstrap 4 Final

Dieses Buch wurde für die Version 4.1.0 überarbeitet, wie sie im April 2018 verfügbar war. Die Versionsangaben im Buch lauten dagegen *4.0.0*. Wenn Sie Version 4.1.0 benutzen, ersetzen Sie alle Versionsangaben entsprechend. Benutzen Sie eine spätere Version, passen Sie die Angaben an. Während der Lebensdauer dieses Buchs wird es neue Versionen geben, sodass »4.0.0« hier als Platzhalter aufgefasst werden kann.

Repository für lokale Installation

Um eine lokale Kopie zu erhalten, nutzen Sie *npm* (<https://www.npmjs.com/>). Wenn Sie in Ruby on Rails entwickeln, sollten Sie sich »Bootstrap for SASS« (<https://github.com/twbs/bootstrap-rubygem>) anschauen.

Mit npm installieren

Mithilfe von *npm* wird Bootstrap folgendermaßen installiert:

```
$ npm install bootstrap@latest --save
```

Wenn Sie *npm* benutzen, werden Sie möglicherweise *NodeJs* einsetzen. Binden Sie Bootstrap dann folgendermaßen ein:

```
require('bootstrap')
```

Dieser Befehl lädt Bootstraps jQuery-Plug-ins in das jQuery-Objekt. Das Modul *bootstrap* selbst exportiert nichts. Sie können die jQuery-Plug-ins individuell laden, indem Sie **/js/*.js*-Dateien im obersten Verzeichnis des Pakets laden.

Bootstraps *package.json* enthält einige zusätzliche Metadaten unter den folgenden Abschnitten:

- *sass*: Pfad zu Bootstraps SASS-Quelldateien
- *style*: Pfad zu Bootstraps nicht minimiertem CSS, das mit den Standardeinstellungen vorkompiliert wurde (ohne Anpassungsmöglichkeit)



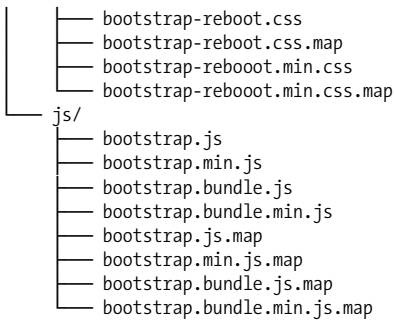
Bootstrap wird in Version 4 mit SASS (<http://sass-lang.com/>) entwickelt. Schauen Sie beim Herunterladen auf die aktuellen Informationen, um sicherzustellen, dass Sie den richtigen Präprozessor einsetzen. SASS ist auf allen Plattformen verfügbar. Bootstrap verwendet *Autoprefixer*, um mit den Vendor-Präfixen in CSS zu arbeiten. Falls Sie Bootstrap von der SASS-Quelle kompilieren möchten und nicht das mitgelieferte Gruntfile verwenden, muss Autoprefixer selbst in den Build-Vorgang integriert werden. Werden vorkompilierte Bootstrap-Dateien oder das Gruntfile verwendet, ist *Autoprefixer* bereits eingebunden. Die Vorgehensweise gilt für Gulp und andere Task Runner äquivalent, wenn diese eingesetzt werden sollen.

Bootstrap kann in zwei Arten von Paketen heruntergeladen werden – in der kompilierten Version und zusätzlich in minimierten Varianten. Bootstrap benötigt jQuery als Grundlage der Komponenten.

Struktur der CSS-Dateien

Die vollständige Liste der Dateien einer Umgebung, die Bootstrap verwendet, sieht folgendermaßen aus:

```
bootstrap/dist
├── css/
│   ├── bootstrap.css
│   ├── bootstrap.css.map
│   ├── bootstrap.min.css
│   ├── bootstrap.min.css.map
│   ├── bootstrap-grid.css
│   ├── bootstrap-grid.css.map
│   ├── bootstrap-grid.min.css
│   └── bootstrap-grid.min.css.map
```



Zu Bootstrap gibt es zwei abhängige Bibliotheken, die zur Nutzung aller Funktionen erforderlich sind:

- jQuery
- Popper

Davon wird freilich immer nur ein Teil genutzt. Die Darstellung gibt lediglich die Dateien wieder, die bei der Installation via npm bereitgestellt werden.

Es gibt die vier großen Bereiche Layout, Content, Components und Utilities in Bootstrap, die in verschiedenen Dateien definiert sind:

- *bootstrap.css*: Layout, Content, Components, Utilities
- *bootstrap-grid.css*: nur Grid des Teils Layout, nur Flex-Unterstützung in Utilities
- *bootstrap-reboot.css*: nur Reboot in Content (siehe Abschnitt *Reboot* am Ende dieses Kapitels)

Vorkompilierte Dateien sind die einfachste Methode, Bootstrap zu nutzen. Die *min*-Versionen sind zusätzlich minimiert (verdichtet). Die *map*-Dateien dienen dazu, die minimierten Versionen im Browser debuggen zu können – sie verbinden die vollständige Version mit der minimierten.

Wenn man die Tooltips und Popover nutzen will, wird die Zusatzbibliothek *Popper* benötigt. Information zu Popper gibt unter <https://popper.js.org/>.



Minimizr

Wenn Sie einen Minimizr (manchmal auch Minifier/Uglifier genannt) in Ihrem Projekt einsetzen, übergeben Sie die nicht minimierten Dateien. Einige Minimizr zerstören den Code, wenn er bereits mit einem anderen Minimizr verdichtet wurde.

Fonts

Neben CSS und JavaScript enthält die Distribution keine Fonts, die die Icons liefern. Icon Fonts sind eine besonders kompakte und einfache Methode, um einfarbige Symbole in Websites einzubinden.

Empfehlenswert ist die Bibliothek Font-Awesome.

Font-Awesome gehört nicht zu Bootstrap 4. Es wird aber oft ergänzend für Symbole benutzt. Bis Bootstrap 3 wurden Icons (Glyphicons) mit ausgeliefert, wegen der großen Vielfalt an derartigen Schriftbibliotheken wurde die Wahl jetzt dem Entwickler überlassen. Font-Awesome ist eine der wichtigsten und umfassendsten derartigen Fonts. Näheres hierzu finden Sie in Kapitel 5 im Abschnitt *Symbole*.

Seitenaufbau

Ist alles vorbereitet, können Sie die erste Seite erstellen. Diese Seite sollte das Basislayout der gesamten Applikation liefern. Eine erste Version könnte folgendermaßen aussehen:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1, shrink-to-fit=no">
    <meta http-equiv="x-ua-compatible" content="ie=edge">

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/
bootstrap/4.0.0/css/bootstrap.min.css" crossorigin="anonymous">
  </head>
```

```

<body>
  <h1>Hallo Bootstrap 4</h1>

  <script src="https://ajax.googleapis.com/ajax/libs/jquery/
2.1.4/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/
js/bootstrap.min.js" crossorigin="anonymous"></script>
</body>
</html>

```

Die drei ersten Metatags müssen am Anfang des `<head>`-Blocks stehen. Die in früheren Versionen üblichen Kompatibilitätseinträge für ältere IE-Versionen gibt es nicht mehr. Browser älter als IE10 werden nicht unterstützt. Beim Betriebssystem ist die Unterstützung für iOS6 oder älter nicht mehr gegeben.



Wenn Sie diese Seite lokal via *file:///* betrachten, funktionieren einige Teile nicht wie erwartet. Versuchen Sie, alle Entwicklungsschritte immer mit einem lokalen Webserver auszuführen (Visual Studio mit IIS Express, IIS, NodeJs mit einem HTTP-Serverpaket oder ein lokaler Apache oder NGinx sind perfekt dafür geeignet).

Beachten Sie, dass Bootstrap selbst nach jQuery geladen werden muss, wenn die JavaScript-Komponenten benutzt werden.



Laden Sie jQuery nur dann von einem CDN, wenn die Website später öffentlich sein soll. Für das Intranet ist eine lokale Kopie besser geeignet.

Browserunterstützung

Auch bei Bootstrap ist die Browserunterstützung ein Thema. Es wurde viel Aufwand getrieben, um möglichst viele Browser und Plattformen zu unterstützen. Aktuell sieht dies folgendermaßen aus:

Tabelle 1-1: Browserunterstützung

	Chrome	Firefox	Internet Explorer	Edge	Safari
Android	OK	OK	X	OK	X
iOS	OK	OK	X	OK	OK
macOS	OK	OK	X	X	OK
Windows	OK	OK	OK	OK	X
Windows Mobile	X	X	X	OK	X

Chromium und Chrome für Linux, Firefox für Linux und Internet Explorer 7 sollten funktionieren, werden aber nicht offiziell unterstützt. Die breiteste Abdeckung von Browsern haben Sie derzeit auf macOS und auf Windows. Opera wird auf macOS und Windows unterstützt. Andere Browser nutzen meist eine der Render-Engines der oben gezeigten Stammversionen und verhalten sich dann entsprechend.

ARIA

Die Unterstützung von barrierefreien Anwendungen (*Accessible Rich Internet Applications Suite* = ARIA (<http://www.w3.org/TR/wai-aria/>)) ist eigentlich ein HTML-Thema. Einige Beispiele im Text sind darauf bereits ausgerichtet und tragen diese Attribute. Es folgt eine Übersicht darüber, wie das funktioniert. Bootstrap unterstützt ARIA implizit und umfassend.

HTML5 – das role-Attribut

Die role-Attribute werden in die relevanten HTML-Tags gesetzt. Sie verbessern die semantische Auszeichnung und helfen damit Screenreadern und anderen für den barrierefreien Zugang eingesetzten Geräten bei der korrekten Ausgabe:

- **banner:** das Element ist ein Banner
- **complementary:** das Element ergänzt einen Bereich, steht meist anstatt eines `<aside>`

- content: regulärer Inhalt
- info: Zusatzinformationen
- form: Formular
- main: Hauptbereich
- navigation: Navigationsbereich
- search: Suchformular

Die folgenden role-Attribute beschreiben die Struktur der Seite:

- article: Artikel (Text)
- columnheader: Spaltenkopf
- definition: Definition
- directory: Verzeichnis
- document: Dokument
- group: Gruppe
- heading: Kopfbereich
- img: Bilder
- list: Liste
- listitem: Listenelement
- math: mathematische Formel
- note: Notiz oder Ergänzung
- presentation: Präsentation, Anzeigeunterstützung
- region: Bereich
- row: Zeile
- rowheader: Kopf einer Zeile (links der Zeile)
- separator: Trennlinie in Menüs oder Listen
- toolbar: Werkzeugleiste

Ein Anwendungsbeispiel:

```
<hr role="separator" />
```

In Tags wie <nav> oder <aside> ist das role-Attribut redundant. Folgendes ist also nicht erforderlich:

```
<nav role="navigation">
<aside role="complementary">
```



ARIA kein Thema?

Neben dem barrierefreien Zugang erleichtern die semantischen Attribute auch die Wartung und Pflege des Quelltexts. Es ist allemal einfacher, mit `role="banner"` zu arbeiten als mit Dutzenden von verschachtelten `<div>`-Elementen.

Optimierung

Korrekt erstellte Bootstrap-Seiten können um einiges umfangreicher ausfallen als klassische HTML-Seiten. Die Stabilität der Seiten hat ihren Preis. Sie sollten daher sorgfältig abwägen, wie Elemente erstellt werden. Ein typisches Beispiel sind lange Listen mit vielen Optionen. Hier kommen neben den Listenelementen auch diverse Schaltflächen oder Menüs zum Einsatz. Vor allem bei serverseitig generiertem Code erscheint das unkritisch und verursacht wenig Aufwand. Hier ein Beispiel:

```
<ul class="list-group">
  <li class="list-group-item">Erstes Element
  <div class="btn-group">
    <button type="button"
      class="btn btn-default dropdown-toggle"
      data-toggle="dropdown" aria-haspopup="true"
      aria-expanded="false">
      Aktion <span class="caret"></span>
    </button>
    <ul class="dropdown-menu">
      <li><a href="#">Löschen</a></li>
      <li><a href="#">Verschieben</a></li>
      <li><a href="#">Umbenennen</a></li>
      <li role="separator" class="divider"></li>
      <li><a href="#">Herunterladen</a></li>
    </ul>
  </div>
</li>
<li>... weitere Elemente</li>
<li>... weitere Elemente</li>
</ul>
```

Diese Liste benötigt ca. 530 Zeichen, in UTF-16 ist das mehr als 1 KByte (bei UTF-8 werden nur die Zeichen außerhalb von ASCII mit mehreren Zeichen belegt). Wenn Sie 40 Elemente auf der Seite anzeigen, was aufgrund des Drop-downs optisch kein Problem ist, sind das 40 KByte HTML bei einer Nutzlast von grob geschätzt 2 KByte (40-mal Text pro Eintrag mit 50 Byte). Hier lohnt es sich, JavaScript einzusetzen.

Das folgende Skript definiert einmal eine Vorlage mit dem Code, und das JavaScript im folgenden Listing fügt ihn dann zur Laufzeit an jedes Listenelement an. Zur Steuerung werden HTML5-Attribute benutzt.

```
<ul class="list-group" data-list-target>
  <li class="list-group-item">Erstes Element</li>
  <li>...weitere Elemente</li>
  <li>...weitere Elemente</li>
</ul>
<div class="btn-group" data-list-template>
  <button type="button"
    class="btn btn-default dropdown-toggle"
    data-toggle="dropdown" aria-haspopup="true"
    aria-expanded="false">
    Aktion <span class="caret"></span>
  </button>
  <ul class="dropdown-menu">
    <li><a href="#">Löschen</a></li>
    <li><a href="#">Verschieben</a></li>
    <li><a href="#">Umbenennen</a></li>
    <li role="separator" class="divider"></li>
    <li><a href="#">Herunterladen</a></li>
  </ul>
</div>
```

Nun wird mittels JavaScript der durch `data-list-template` adressierte Code ausgelesen und dann geklont an die durch `data-list-target` erreichbare Liste angehängt. Benutzt wird hier jQuery:

```
// Ausführung, wenn das Dokument geladen wurde.
$(function(){
  // Template laden, klonen und verstecken.
  var template = $('[data-list-template]').hide().clone();
  // Listenelemente suchen, kopierte Vorlage anhängen, anzeigen.
```