



# Introducing Bootstrap 4

---

Jörg Krause

Apress®

# Introducing Bootstrap 4



Jörg Krause

Apress®

## ***Introducing Bootstrap 4***

Jörg Krause  
Berlin, Germany

ISBN-13 (pbk): 978-1-4842-2381-9  
DOI 10.1007/978-1-4842-2382-6

ISBN-13 (electronic): 978-1-4842-2382-6

Library of Congress Control Number: 2016961809

Copyright © 2016 by Jörg Krause

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director: Welmoed Spahr  
Acquisitions Editor: Louise Corrigan  
Development Editor: James Markham  
Technical Reviewer: Massimo Nardone  
Editorial Board: Steve Anglin, Pramila Balen, Laura Berendson, Aaron Black, Louise Corrigan,  
Jonathan Gennick, Todd Green, Celestin Suresh John, Nikhil Karkal, Robert Hutchinson,  
James Markham, Matthew Moodie, Natalie Pao, Gwenan Spearing  
Coordinating Editor: Nancy Chen  
Copy Editor: Judy Ann Levine  
Compositor: SPi Global  
Indexer: SPi Global  
Artist: SPi Global, image courtesy of Freepik.

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail [orders-ny@springer-sbm.com](mailto:orders-ny@springer-sbm.com), or visit [www.springer.com](http://www.springer.com). Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail [rights@apress.com](mailto:rights@apress.com), or visit [www.apress.com](http://www.apress.com).

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at [www.apress.com/bulk-sales](http://www.apress.com/bulk-sales).

Any source code or other supplementary materials referenced by the author in this text is available to readers at [www.apress.com](http://www.apress.com). For detailed information about how to locate your book's source code, go to [www.apress.com/source-code/](http://www.apress.com/source-code/).

Printed on acid-free paper

*This book is for everyone making their first steps into the world of software.  
Or for those wish to bring their knowledge up to date.*

*The future of software development may be the web, the cloud, or where ever.  
In any case, it is not on an isolated local system.*

*This book is part of a project, a set of textbooks, to help you  
fight the challenges of web development.*

*The subjects are not necessarily new, but do form a unique subject.*

# Contents at a Glance

<b>About the Author .....</b>	<b>xv</b>
<b>About the Technical Reviewer .....</b>	<b>xvii</b>
<b>Introduction .....</b>	<b>xix</b>
<b>■ Chapter 1: It Begins with CSS.....</b>	<b>1</b>
<b>■ Chapter 2: Introduction to Bootstrap.....</b>	<b>23</b>
<b>■ Chapter 3: Structure of the Page .....</b>	<b>33</b>
<b>■ Chapter 4: Typography.....</b>	<b>45</b>
<b>■ Chapter 5: Forms .....</b>	<b>69</b>
<b>■ Chapter 6: Additional Modules .....</b>	<b>105</b>
<b>■ Chapter 7: Components .....</b>	<b>117</b>
<b>■ Chapter 8: Active Components .....</b>	<b>171</b>
<b>Index.....</b>	<b>207</b>

# Contents

<b>About the Author .....</b>	<b>xv</b>
<b>About the Technical Reviewer .....</b>	<b>xvii</b>
<b>Introduction .....</b>	<b>xix</b>
<b>■ Chapter 1: It Begins with CSS.....</b>	<b>1</b>
Basics.....	1
Syntax .....	2
Selector .....	2
Elements (Tags) .....	2
IDs.....	3
Classes .....	3
Attributes .....	3
Logical Selection .....	4
More Selectors .....	4
The Box Model.....	7
Blocks of the Box.....	8
The Box Model in CSS3.....	9
Compliance with Media.....	9
Syntax.....	10
Parameter .....	12
The Viewport .....	17
Set Viewport .....	17
Viewport Settings .....	19

- Units ..... 19
  - Absolute Units..... 19
  - Relative Units..... 21
- **Chapter 2: Introduction to Bootstrap** ..... 23
  - What’s New in Bootstrap 4? ..... 23
    - Global Changes..... 23
    - Grid System ..... 23
    - Tables ..... 23
    - Forms ..... 24
    - Buttons ..... 24
    - Pulldown Menus ..... 24
    - Panels ..... 24
    - Others ..... 24
  - Installation..... 24
    - CDN..... 25
    - Repository for Local Installation ..... 25
    - Install using NPM ..... 26
  - Structure of the CSS *Files* ..... 27
  - Page Building ..... 28
  - Browser Support ..... 29
  - ARIA..... 29
    - HTML5—the *Role* Attribute ..... 29
  - Optimization ..... 31
- **Chapter 3: Structure of the Page** ..... 33
  - Introduction ..... 33
    - The HTML5 Doctype..... 33
    - Settings of the Viewing Area..... 33

The Grid System .....	34
Container .....	34
The Grid in Detail .....	35
<b>■ Chapter 4: Typography.....</b>	<b>45</b>
Headings .....	45
Text and Text Elements.....	46
Basic Font.....	46
Text Elements .....	47
Orientation .....	50
Transformations.....	52
Lists.....	52
Tables .....	55
Styles for Tables .....	55
Responsive Tables .....	60
Auxiliary Classes .....	65
Check Distances .....	66
<b>■ Chapter 5: Forms .....</b>	<b>69</b>
Structure of a Form .....	69
Simple Form Elements .....	69
Single-Line Forms .....	71
Form Elements with Blocks.....	74
Sizes .....	75
Dealing with Checkboxes and Radio Buttons .....	76
Additional Buttons .....	77
Horizontal Forms .....	80
Input Elements .....	82
Input Elements.....	82
Select Elements.....	85



Static Texts in Forms .....	85
Behavior of the Form Elements .....	87
Validation Information in Forms .....	89
Form Elements in the Grid .....	95
Adaptation of the Field Height .....	96
Help Text in Forms .....	97
<b>Buttons .....</b>	<b>98</b>
Semantic Buttons .....	99
Size and Appearance .....	100
States .....	103
<b>■ Chapter 6: Additional Modules .....</b>	<b>105</b>
<b>Symbols.....</b>	<b>105</b>
Alternatives.....	106
Use of Symbols.....	108
Use on Buttons .....	108
Messages .....	109
Common Symbols.....	110
<b>Responsive Images .....</b>	<b>110</b>
<b>Thumbnails.....</b>	<b>112</b>
Embedded Sources.....	112
<b>Colors and Backgrounds .....</b>	<b>112</b>
Text Color.....	113
Background Color .....	114
<b>Alignment of Elements in Flow.....</b>	<b>115</b>
Centering .....	115
Break: Clearfix .....	115
<b>Show and Hide Content.....</b>	<b>116</b>

<b>Chapter 7: Components</b> .....	<b>117</b>
Drop-Down Menus .....	117
Alignment of the Menu .....	119
Decorative Elements .....	119
Subheads in Menus .....	119
Divider Line.....	120
Deactivated Links .....	121
Toolbars.....	121
Vertical Alignment.....	123
General Options .....	124
Menu Button.....	124
Simple Button with Menu .....	124
Split-Button Menu .....	125
Sizes of Menu Buttons.....	126
Special Menu Options.....	127
Navigation .....	128
Tabs .....	128
Navigation Buttons (pills).....	129
Universal Settings.....	130
Navigation Bar .....	132
Breadcrumb Navigation .....	137
Page Scrolling.....	138
Tags .....	141
Identification Labels .....	142
Big Screen (jumbotron) .....	144
Page Headers .....	145
Messages .....	146
Progress Bar.....	148

<b>Media</b> .....	<b>152</b>
Orientation .....	152
Media Lists .....	153
<b>Common Lists</b> .....	<b>153</b>
Tags in Lists.....	154
Links in Lists.....	155
Buttons in Lists.....	155
<b>Cards</b> .....	<b>159</b>
Headings.....	160
Footers.....	161
Cards with Tables .....	161
Cards with Lists .....	162
Card with Pictures .....	163
Cards in the Grid.....	164
Deck of Cards .....	165
Card Groups.....	166
Cards in Columns.....	167
<b>■ Chapter 8: Active Components</b> .....	<b>171</b>
Setup and Activation .....	171
The Programming Interface.....	172
Conflict Prevention .....	172
Events .....	173
Transitions .....	173
Modal Dialogs.....	174
Features.....	174
Sizes .....	176
Dialog with Grid .....	177

General Information about Behavior .....	181
Options .....	182
Use.....	182
<b>Pull-Down Menu (Dropdown) .....</b>	<b>183</b>
Common Information about Behavior .....	183
Options .....	183
<b>Scroll Bar Supervisor (ScrollSpy) .....</b>	<b>185</b>
Use.....	185
Options .....	186
<b>Pinned Navigation (Affix) .....</b>	<b>186</b>
<b>Reversible Tabs (Tab).....</b>	<b>187</b>
Use.....	187
Options .....	189
<b>Tooltip.....</b>	<b>189</b>
Multiline Links .....	190
General Tips.....	191
Options .....	191
<b>Content Overlay (Popover).....</b>	<b>192</b>
Application.....	192
Options .....	194
<b>Message (Alert) .....</b>	<b>196</b>
Options .....	196
<b>Action Buttons (Button) .....</b>	<b>197</b>
Toggle Button.....	197
Options .....	198

<b>Content Insertion (Collapse)</b> .....	<b>198</b>
Application.....	198
Content Groups—The Accordion .....	199
Options .....	202
<b>Image Roundabout (Carousel)</b> .....	<b>202</b>
Application.....	202
Options .....	204
<b>Index</b> .....	<b>207</b>

# About the Author



**Jörg Krause** works as a trainer, consultant and software developer for major companies worldwide. Build on the experience of 25 years of work with web and many, many large and small projects. Jörg believes it is especially important to have solid foundations. Instead of always running to create the latest framework, many developers would be better advised to create and provide a robust foundation.

Jörg has written over 50 titles in the renowned and prestigious specialist publishers in German and English, including some bestsellers. Anyone who wants to learn this subject compact and fast, its right here. On his website [www.joergkrause.de](http://www.joergkrause.de) much more information can be found.

# About the Technical Reviewer



**Massimo Nardone** has more than twenty-two years experience in security, web/mobile development, cloud, and IT architecture. His true IT passions are in the security and android field.

He has been programming and teaching how to program with android, Perl, PHP, Java, VB, Python, C/C++, and MySQL for more than twenty years. He holds a Master of Science degree in Computing Science from the University of Salerno, Italy.

He has worked as a project manager, software engineer, research engineer, chief security architect, information security manager, PCI/SCADA auditor, and senior lead IT security/cloud/SCADA architect for many years.

His technical skills include: security, android, cloud, Java, MySQL, Drupal, Cobol, Perl, web, and mobile development; MongoDB, d3, Joomla, Couchbase, C/C++, WebGL, Python, Pro Rails, django CMS, Jekyll, Scratch, and many others.

He is currently working as the chief information security officer (CISO) for Cargotec Oyj.

He worked as visiting lecturer and supervisor for exercises at the Networking Laboratory of the Helsinki University of Technology (Aalto University). He holds four international patents (PKI, SIP, SAML, and Proxy areas).

Massimo has reviewed more than forty IT books for different publishing company, and he is the coauthor of *Pro Android Games* (Apress, 2015).

# Introduction

## Bootstrap 4 – the new CSS Framework

This book describes in a compact and clear form the CSS Framework Bootstrap 4. Bootstrap was originally designed, around 2010, on Twitter and was from the outset developed with the idea of “Mobile First” – first for mobile devices . It has since grown to become one of the best and most popular frameworks for the design of websites. Accompanied by a large number to bootstrap engaging design templates – so-called themes – it offers secure and robust style definitions for everyday tasks.

### Target Audience

This book is aimed at beginners and web developers who are new to the web world. Bootstrap serves mainly the front-end developer. It is a series of CSS instructions and these additional JavaScript (ECMAScript) libraries, apply where CSS alone is not enough.

Maybe you are also a web designer, who discovered Bootstrap as an excellent way to upgrade your web pages with dynamic elements. Here, you are dealing with texts, with forms, with the presentation of database content, i.e. everything that constitutes a dynamic web site. Then you will need to know one of this band, namely for the creation of a professional user interface, in particular a clear form.

In any case, I tried not to ask any prerequisites or conditions to the reader. You do not need to be a computer scientist, not in perfect command of language, don't need to know rocket science. No matter in what context they have encountered on Bootstrap, you will be able to read this text.



**Bootstrap** For all examples to understand, you need a working environment for creating web pages. This can be Ruby on Rails, PHP, ASP.NET or Node.js. I recommend node.js for an easy start. It's a nice way to get Bootstrap up and running.

---

If you have accidentally found this text and cannot do anything with the term “Bootstrap”, read it anyway. You will be learning one of the most modern techniques of web development and the future belongs to the circle of excellent developers who can be build good- looking and device-independent sites.



**What You Should Know** Readers of this series have hardly any requirements. Some HTML cannot harm and who already have seen a static HTML page (the source code, of course) is certainly good and pure. I assume that you have at least a current operating system, on which you will find an editor with which you can create web pages.

---

## Examples

You can find the sample project to this book on Github:

- <https://github.com/joergkrause/Bootstrap4-Book>

The project uses sample files contain the pure HTML. It works best with Visual Studio, but can largely be used independently.

## As You Can Read This Text

I will not dictate how you should read this text. In the first draft of the structure, I have tried several variations and found that there exists no ideal form. If I were to orientate myself to the various types of applications, the text would divided into several chapters that are not interrelated. The one or the other readers would then be annoyed that he put a lot of money for a book, where only a fifth is usable. This band solves the problem by reducing it to a very small issue is focused and no “blah-blah” for the inflation of the volume is.

Beginners should read the text as a narrative from the first to the last page. Those who are already somewhat familiar, the less interesting sections for him, he can safely skip sections. If references are required, I have included the corresponding appropriate cross- references.

## Realizations

The theme is set not technically easy to master, because scripts are often too extensive and it would be nice if you could support the best optical reading form. I have therefore included extra line breaks used to aid readability, but have no place in the editor of their development environment.

In general, each program code is set to a non-proportional font. In addition, scripts have line numbers: :

```
1 body {  
2     color: black;  
3 }
```

If you find you need to enter something in the prompt or in a dialog box, this part of the statement is in bold:

```
$ bower install bootstrap
```

The first character is the prompt and is not entered. I use in the book, the Linux prompt, the bash shell. The commands will work, without any exception, unchanged even on Windows, only then STOP C:> or something similar at the beginning of the line.

Expressions and command lines are sometimes peppered with all types of characters, and in almost all cases, it depends on each character. Often, I'll discuss the use of certain characters in precisely such an expression. Then the "important" characters with line breaks alone and also in this case, line numbers line numbers are used to reference the affected symbol in the text exactly (note the : character in line 2):

```
1  a.test {
2      :hover {
3          color: red
4      }
5  }
```

The font is non-proportional, so that the characters are countable and opening and closing parentheses are always among themselves.

## Symbols

To facilitate the orientation in the search for a solution, there is a whole range of symbols that are used in the text.



**Tip** This is a tip

---



**Information** This is an information

---



**Warning** This is a warning

---

# CHAPTER 1



# It Begins with CSS

Before you begin to deal with bootstrap, you should understand CSS. As an introduction to this book, there is a short refresher to CSS.

CSS (cascading style definitions)—is a layout and formatting language for creating and formatting markup languages such as HTML. Ideally, the HTML document contains only semantic information, and with CSS this is then formatted in design and typography.

HTML already brings some basic formatting, such as a larger font for headlines, which can be customized with CSS as well as the unformatted elements. Format tags in HTML formatting and their formatting attributes generally should not be used. They are obsolete in HTML 5. It is replaced by CSS.

With CSS, it is also possible to define output types separately for different media such as a monitor (screen), projection (projection), and printing (print).

## Basics

An HTML document consists of semantically meaningful awards for headings, paragraphs, lists, and so forth. The CSS statements must be placed in such a way that the browser can assign these elements.

Basically, there are three ways to store these messages and they are as follows:

- The `style`-attribute that any HTML elements knows.
- The element `<style></style>`, which summarizes several style rules.
- The `<link />`-element that references a file that contains multiple style definitions

The preferred way is to use a CSS file. This can be done in the browser cache and the content can be displayed with the appropriate tools. Then you can reduce the bandwidth savings (not because of the bandwidth, but because of the associated performance gain).

Local `<style>`-elements should be used only in exceptional cases, for example, to temporarily change any possible complex files. Such local style instructions have a higher priority. The `style`-attribute expands or modifies the styles once per single element. It has the highest priority over any conflicting rules from the local or imported styles.

The link to the CSS file takes place in the head section of the HTML document:

```
1 <link rel="stylesheet"
2     type="text/css"
3     href="styles/style.css">
```

Note that the path to the file must be specified relative to the HTML document.

## Syntax

The syntax of CSS is relatively simple. The basic structure consists of two building blocks:

1. Selector
2. Ruleset

The selector determines which element or elements the rules apply to.

```
1 Selector {
2     Ruleset
3 }
```

When styles are in `style`-attributes, they are only valid for that item, therefore the selector is eliminated.

The standard rule turn (ruleset) consists of rules. These are to be written in the following form:

Style: Parameter;

The semicolon at the end is required.

## Selector

The selector is the instrument that allows the elements on the page can be achieved selectively. The whole scheme of selectors is fairly comprehensive. At this point, first, the most important building blocks are shown.

## Elements (Tags)

With CSS, you can address individual items. The syntax for this is as follows:

```
TagName { Ruleset }
```

If you have all of the elements `<p>` want to achieve, it is enough to write the following:

```
p { }
```

## IDs

Often, a single tag is formatted. The tag must be identifiable, that is, it must have an ID that is accessible. The same HTML attribute contains a string that is achieved in the CSS properties. An ID, however, may only be applied to one element within a document and must therefore occur only once. The syntax for this as follows:

```
#id { }
```

An example of a button looks like this:

```
1 <style>
2 #send {
3   color: red;
4 }
5 </style>
6 <button id="send">Send</button>
```

## Classes

Frequently several elements are addressed. These are classes that are written in HTML in the class attribute. You can attribute this to multiple space-separated classes, by combining rulesets.

This saves extensive definitions. Bootstrap utilizes this possibility comprehensively and achieves this with a few rulesets and a large number of modification options. Unlike IDs, they can define many classes of elements, and therefore, they may occur more than once in a document.

The syntax can look like this:

```
.class { }
```

Another example of a button:

```
1 <style>
2 .btn {
3   color: red;
4 }
5 </style>
6 <button class="btn">Send</button>
```

## Attributes

Attributes can be referenced using the following syntax of HTML elements:

```
[name] { }
[name="value"] { }
```

An example of another button:

```

1 <style>
2 [data-item] {
3   color: blue;
4 }
5 </style>
6 <a href="link.html" data-item="22">Next</button>

```

If the value of the attribute (to the right of the = sign) is not specified, then only the existence of the attribute is considered sufficient to apply to the rules.

## Logical Selection

It often happens that rulesets are to be applied to multiple selectors. For this purpose, an OR logic is required, which is noted as a comma in CSS:

```
a, b { }
```

Between a and b there is no link, the rule is applied independently to both. The placeholders a and b in the example may be more complex selectors.

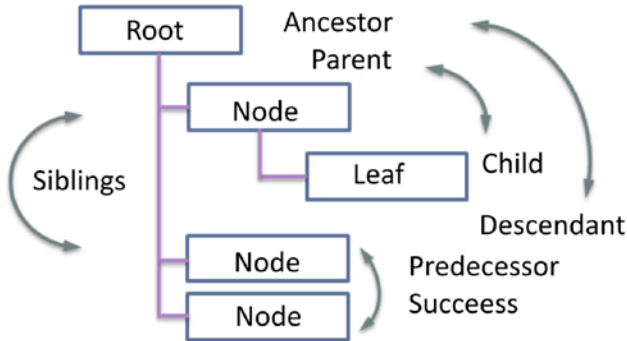
## More Selectors

In practice, these selectors are not enough. Table 1-1 provides a compact overview of all the other forms.

**Table 1-1.** Simple CSS Selectors

Symbol	Description
*	Universal / all
Tag	Element's name
.class	Class (Attribute class)
#id	ID (Attribute id)
[a]	Attribute presence
[a=v]	Attribute value
[a~v]	Attribute contains a value as a standalone word
[a =v]	Attribute contains no value
[a^=v]	Attribute starts with value
[a\$=v]	Attribute value ends with
[a*=v]	Attribute contains a value

Dealing with hierarchies is essential because HTML documents are hierarchies, which are often called trees. Figure 1-1 shows the relationship between the elements in the document tree.



**Figure 1-1.** Elements of the hierarchy of a HTML page

Table 1-2 shows the syntax for CSS.

**Table 1-2.** CSS Selectors for Hierarchies

Symbol	Description
$e > f$	Selection if $f$ is a child element of $e$
$e f$	Selection, if $f$ is a descendant of $e$
$e + f$	Selection, if $f$ is a successor of $e$
$e \sim f$	Selection when $e$ is sibling of $f$

In contrast to the possibilities of relationships missing selectors for ancestors, predecessors, and parents; you can do this by swapping the elements.

Pseudoselectors are those that have no comparable representation in HTML, but result from the position of elements or use. There are three types of pseudoselectors:

- Static positions (see Table 1-3 for a list of static selectors)
- Selection of areas (see Table 1-4 for a list of CSS selectors for areas)
- Dynamic behavior (see Table 1-5 for a list of dynamic CSS selectors)