

第 10 章

Bootstrapの カスタマイズ

これまで見てきたように、Bootstrap の特徴でもあるコンポーネントや小回りの利くユーティリティ、基本スタイルの Reboot などを利用すれば、デザインの苦手な人でも簡単に洗練された UI を作ることができます。しかし、提供されている機能では物足りず、スタイルをカスタマイズしたいケースも出てくるでしょう。本章では、Bootstrap をカスタマイズする方法として、Bootstrap のスタイルを上書きする方法と、Sass を使ってカスタムスタイルを追加する方法、CSS 変数を利用する方法を解説します。

10

SECTION

1

Bootstrap のオリジナルスタイルを上書きする

まず手軽にカスタマイズする方法として、Bootstrap のオリジナルのスタイルを上書きする方法を解説します。基本的には Bootstrap のスタイルを利用し、部分的に微調整したい場合にはこの方法が有効です。

10.1.1 カスタマイズ用 CSS を参照する

Bootstrap の公式サイトからダウンロードできるコンパイル済みの CSS ファイル bootstrap.css (bootstrap.min.css) は、そのままで 8,000 行を超えます。このファイルに直接手を加えるのは賢明ではありません。該当箇所を探しながら作業するのはとても面倒ですし、バージョンアップした際に、最新版に反映させるのも手間がかかります。別途カスタマイズ用の CSS ファイルを用意して上書きする方が、カスタマイズした箇所もわかりやすくて良いでしょう。

Bootstrap のオリジナル CSS を、独自のカスタマイズ用 CSS で上書きする場合は、Bootstrap の CSS を参照しているコードの後に、link 要素を追加し、カスタマイズ用 CSS (例では custom.css) を参照させます (リスト 10-1)。

▼リスト 10-1 bootstrap.css を別 CSS ファイルで上書きする

```
<link rel="stylesheet" href="css/bootstrap.min.css"><!-- オリジナルCSS -->
<link rel="stylesheet" href="css/custom.css"><!-- カスタマイズ用CSS -->
```

10.1.2 Bootstrap の CSS 設計の方針

カスタマイズ用 CSS を用意したら、スタイルを追加していくますが、このとき、Bootstrap で定義済みのクラス名はできるだけ使わない方が良いでしょう。オリジナルのスタイルを変更してしまうと、サイト全体に渡ってスタイルが変更されてしまい、また別の場所で初期スタイルのコンポーネントを使いたくても、そのままでは利用できなくなります。利用するには、上書きしたスタイルをさらに打ち消すスタイルを書く必要があり、コードも作業も複雑になります。新しいクラスセレクタを作り、HTML 側でクラスを追加する方がスマートです。もちろん、スタイルを再利用する予定がない場合は、同名のセレクタを使って手っ取り早くスタイルを上書きしても問題ありません。

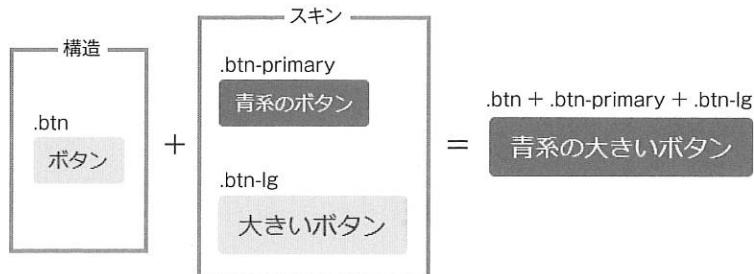
Bootstrap はクラス名の付け方が特徴的ですが、これはオブジェクト指向 CSS 「**OOCSS (Object-Oriented CSS)**」に基づいて設計されているためです。OOCSS とは、**構造**と**スキン**を分離してクラスを定義し、それらを組み合わせてスタイルを適用するという CSS 設計の概念の 1 つです。たとえば、Bootstrap のボタンコンポーネントは、btn、btn-{ 色 }、btn-{ サイズ } と 3 つのクラスが指定され、これにより、1 つのボタンが完成します。構造とは共通している部分で、ボタンの場合は「btn」です。これにスキンとなる色のクラス「btn-

「primary」やサイズのクラス「btn-lg」が加わって、「primary カラー（デフォルトで青系）の大きいボタン」が完成します（リスト 10-2、図 10-1）。

▼リスト 10-2 Bootstrap のボタンコンポーネントの例

```
<button type="button" class="btn btn-primary btn-lg">青系の大きいボタン</button>
```

▼図 10-1 Bootstrap のボタンコンポーネントの例



1つの要素に複数のクラスを指定することを**マルチクラス**と言いますが、この方法は個々の要素にそれぞれの役割を持つクラスが割り当てられているため、他のスタイル指定の影響を受けにくく、再利用性が高いのが特徴です。

新しいクラス名も、Bootstrap のルールに沿って作りたいところですが、これはかなり難易度が高くなるでしょう。そもそも Bootstrap は、あらゆるサイトで利用できるように汎用性の高いスタイルを定義済みクラスとしてまとめてあります。カスタマイズで追加したいスタイルは、その汎用性から外れるサイト固有のスタイルですから、Bootstrap の設計指針に当てはめるのは難しいと考えられます。では、どのようなクラス名を付けるのが良いのか、次項で考えてみましょう。

10.1.3 クラス名の付け方のポイント

Bootstrap のクラス名の付け方を参考にしながら、よりペタなクラス名の付け方を考えてみましょう。1つめのポイントは、**後でスタイルを変更しても影響のない名前にすること**です。Bootstrap では、ボタンの色を指定するクラスは、btn-primary や btn-success などのようになっていて、btn-blue や btn-red というように、直接的な色を表す名前にはなっていません。もし btn-blue という名前のクラスのスタイルを後で赤に変更した場合、整合性のないクラス名になってしまいます。新しく btn-red を作成するのが自然だと思いますが、そうするすべての HTML ファイルを書き換えることはできません。btn-primary であれば、CSS の変更のみで済みます。Bootstrap の定義済みクラスとかぶらないように、たとえば btn-normal、btn-default などを作っても良いかもしれません。あるいはカスタマイズで追加したクラスということがわかるように btn-custom-normal のように、独自の接頭辞を付けたクラス名にするのも良いでしょう。

次のポイントは、**場所に依存せず、再利用できる名前にすること**です。たとえば、header-text というクラス名を付けてしまった場合、これをフッターで再利用したいときに違和感が出てしまいます。main-text などであれ

ば、場所に関係なく、違和感なく再利用できます。

3つめのポイントは、**予測しやすい名前**にすることです。bootstrap のクラス名は、btn-lg のようにクラス名で予測しやすい名前になっています。ありがちな失敗ですが、btn-1、btn-2 のようなクラス名にしてしまうと、どんなスタイルが定義されているの見当がつきません。予測しやすい名前にするには、flex-row や align-self-start のように CSS のプロパティをクラス名に組み込むのも 1 つの方法です。

以上、クラス名を付けるときの 3 つのポイントを紹介しましたが、もっとも重要なのは、最初にどのような追加クラスが必要になるかをすべて洗い出しておくことです。組み立てながら都度クラスを定義していく方法では、手戻りが生じたり、後でメンテナンス性に影響が出たりすることになります。カスタマイズする前に、サイト全体を通してどのような追加が必要になるのかをすべて洗い出しておくことで、追加するクラス名や定義するスタイルに規則性を持たせることができます。

10.1.4 スタイルを上書きする際の注意点

Bootstrap オリジナルのスタイルには、@media ルールを使って、ブレイクポイントでスタイルが切り替わられているものがあります。カスタムスタイルを追加する場合は、この点に注意しましょう。

たとえば container クラスの場合は、リスト 10-3 のように宣言されており、デバイス幅によって container の最大幅が切り替わるようになっています。デバイス幅が 576px 以上のときは container の最大幅が 540px、デバイス幅が 768px 以上のときは container の最大幅が 720px……という具合です。

▼リスト 10-3 bootstrap.css の container クラスに定義されているスタイル

```
@media (min-width: 576px) {
  .container {
    max-width: 540px;
  }
}
@media (min-width: 768px) {
  .container {
    max-width: 720px;
  }
}
@media (min-width: 992px) {
  .container {
    max-width: 960px;
  }
}
@media (min-width: 1200px) {
  .container {
    max-width: 1140px;
  }
}
```

たとえば、コンテナの最大幅をもう少し大きくカスタマイズするために、custom-container クラスを作り、custom.css にスタイルを定義したとします（リスト 10-4）。

▼リスト 10-4 custom.css にカスタムクラスを設定した例

```
.custom-container {
    max-width: 1260px;
}
```

これを Bootstrap のオリジナル CSS の後から参照して上書きさせると、元々あった container クラスのブレイクポイントによる最大幅のスタイルの切り替えは無視され、デバイス幅に関係なく常に最大幅が 1260px になります（リスト 10-5）。

▼リスト 10-5 bootstrap.css を custom.css で上書きする

```
<link rel="stylesheet" href="css/bootstrap.min.css"><!-- オリジナルCSS -->
<link rel="stylesheet" href="css/custom.css"><!-- カスタマイズ用CSS -->
...
<div class="container custom-container"> . . . </div>
```

そのため、元々のスタイルを残しつつコンテナをカスタマイズで追加したい場合は、リスト 10-6 のように @media ルールでデバイス幅によってスタイルを切り替えます。

▼リスト 10-6 custom.css にカスタムクラスを追加した例

```
.custom-container {
    @media (min-width: 1400px) {
        max-width: 1260px;
    }
}
```

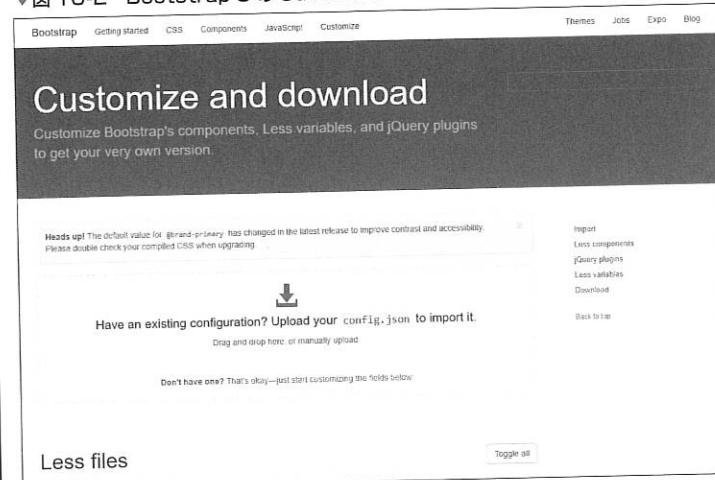
例では、カスタマイズ用に新しくクラスセレクタを作って上書きしていますが、同名セレクタを使って上書きする場合も同じです。意図せず、他のスタイルに影響してしまうことがあるので、カスタマイズする前にオリジナルのスタイルに @media ルールがないか確認してから追加するようにしましょう。

1
2
3
4
5
6
7
8
9
10

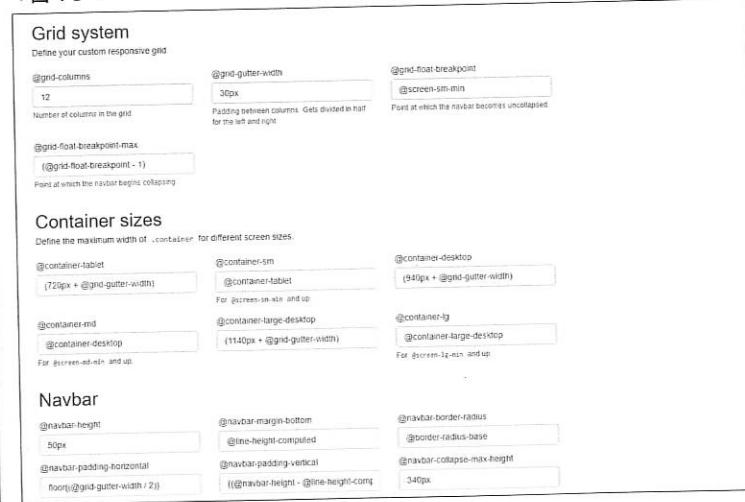
COLUMN Bootstrap 3までのもう1つのカスタマイズ方法

Bootstrap 3までは、Bootstrap の公式サイトに「Customize」ページがあり、使用するスタイルやコンポーネントにチェックを入れたり、LESS^{*1}の変数の値を変更したりしてカスタマイズしたものをダウンロードすることができます（図 10-2、図 10-3）が、Bootstrap 4では、利用できなくなりました。

▼図 10-2 Bootstrap 3 の Customize ページ



▼図 10-3 less の変数を変更できる



*1 LESS とは、Sassと同じようにCSSをプログラム的に実装できるCSSを拡張したメタ言語です。LessとSassは、基本的な部分でできることは同じですが、コンパイルを実行する言語が違い、LESSはJavaScript、SassはRubyで動作します。

10

2

Sass を使ってカスタマイズする

ここからは **Sass** を使って Bootstrap をカスタマイズする方法を紹介します。Sass とは、プログラムのように効率的に CSS を定義できるようにした言語で、変数や算術演算など、CSS にはない機能を利用することができます。Bootstrap 全体にかかるスタイルの初期設定は、Sass ファイルの `_variables.scss` に変数として定義されており、変数の値を変更することでカスタマイズ可能です。

10.2.1 Sass の利用環境を整える

本項では Sass が利用できるように環境を整えます。既に Sass を利用できる環境の人は、本項は飛ばしてください。Sass を導入する方法は色々ありますが、今回は、サーバーサイドの JavaScript 実行環境である **Node.js** で動く **node-sass** を使った Sass の導入方法を説明します。node-sass は、scss ファイルを css ファイルに高速にコンパイルすることができるライブラリです。

■ Node.js をインストールする

まずは、node-sass を利用するために、Node.js をインストールしましょう。Node.js の公式サイト (<https://nodejs.org/ja/>) にアクセスすると、推奨版と最新版がありますが、ここでは推奨版をダウンロードします（図 10-4）。

▼図 10-4 推奨版をダウンロード



ダウンロードが完了したらインストーラーをダブルクリックして実行しましょう。基本的にはインストーラの案内に従って進めていくだけでOKです。

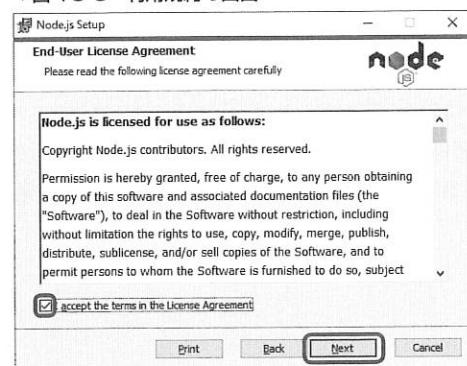
インストーラーを起動すると次の画面が表示されるので [Next] をクリックします（図 10-5）。

▼図 10-5 インストーラー起動画面



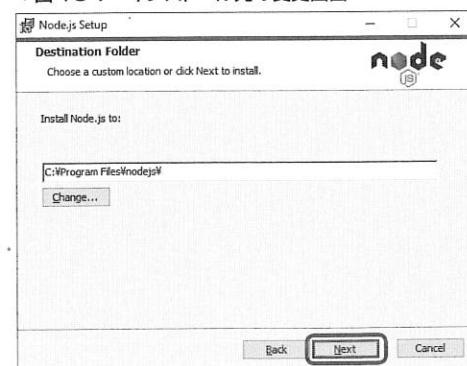
利用規約の画面で、ライセンス同意のチェックを入れて [Next] をクリックします（図 10-6）。

▼図 10-6 利用規約の画面



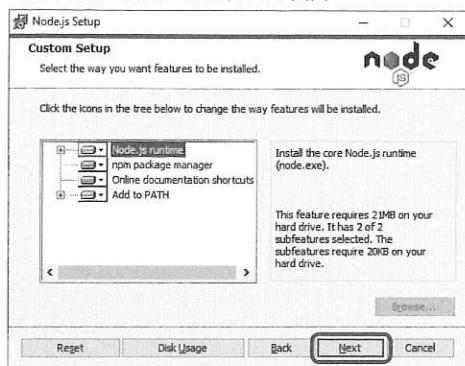
インストール先も特に変更の必要がなければ [Next] をクリックします（図 10-7）。

▼図 10-7 インストール先の変更画面



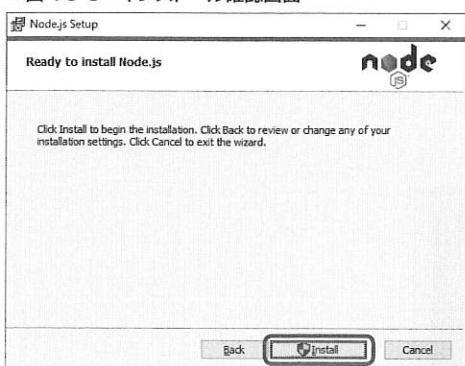
カスタムセットアップ画面も、特に変更する必要がなければそのまま、[Next] をクリックします（図 10-8）。

▼図 10-8 カスタムセットアップ画面



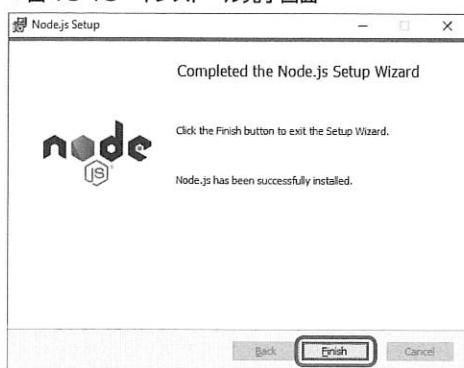
インストール確認画面では [Install] をクリックします。警告が出る場合がありますが、[はい] をクリックします（図 10-9）。

▼図 10-9 インストール確認画面



インストール完了後、[Finish] をクリックして画面を閉じます（図 10-10）。

▼図 10-10 インストール完了画面



インストールが完了したら、正常にインストールできたかを確認してみましょう（図 10-11）。タスクバーの検索ボックスに、**node** と入力し（①）「Node.js command prompt」を選択します（②）。

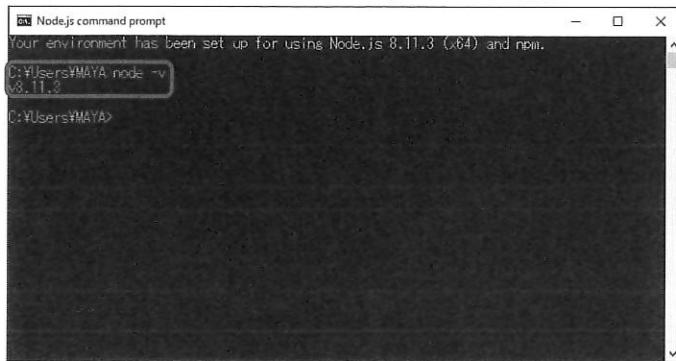
▼図 10-11 Node.js コマンドプロンプトの選択



次のコマンドを実行し、バージョン情報が表示されたらインストール成功です（図 10-12）。

```
node -v
```

▼図 10-12 インストールされている Node.js のバージョン情報



併せて、npm という Node.js モジュールの管理ツールもインストールされるため、同じようにバージョン情報が表示されるか確認してください（図 10-13）。

```
npm -v
```

1
2
3
4
5
6
7
8
9
10

▼図 10-13 インストールされている npm のバージョン情報

```
C:\Users\YAYA>node -v
v8.11.3
C:\Users\YAYA>npm -v
5.6.0
C:\Users\YAYA>
```

■ node-sass をインストールする

Node.js のインストールが完了したら、**node-sass** をインストールしましょう。コマンドプロンプトで次のように入力するとインストールが始まります。

```
npm install -g node-sass
```

完了したら、無事にインストール出来ているかを確認してみましょう。次のコマンドを入力し、node-sass のバージョン情報が表示されればインストール成功です（図 10-14）。

```
node-sass -v
```

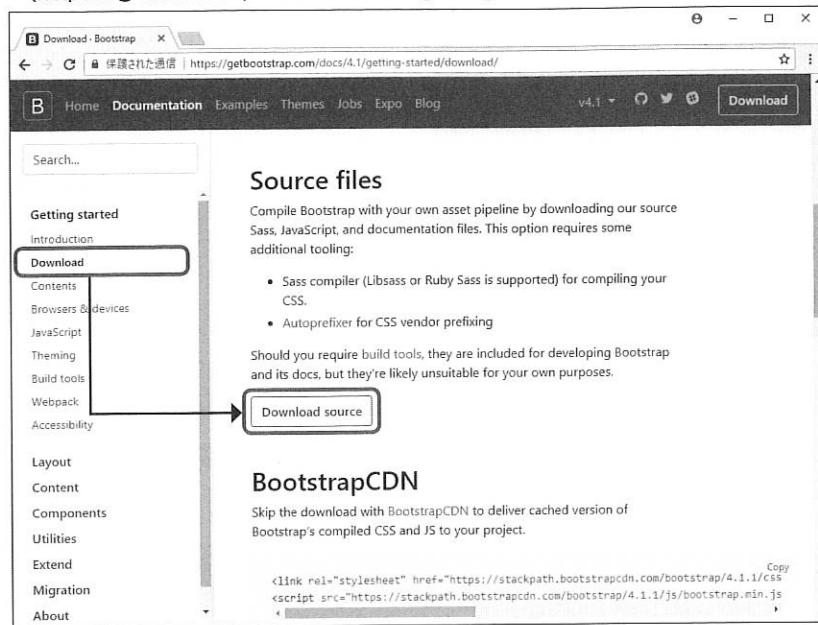
▼図 10-14 インストールされている node-sass のバージョン情報

```
C:\Users\YAYA>node-sass -v
node-sass 4.8.2  (Trapper) [JavaScript]
libsass 3.5.4  (Sass Compiler) [C/C++]
C:\Users\YAYA>
```

10.2.2 SCSS ファイルの準備

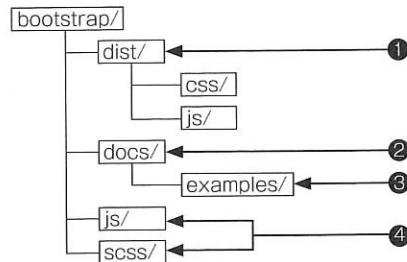
Sass の準備が整ったので、次は Bootstrap の Sass ファイルを用意しましょう。Bootstrap 公式サイトの Downloads ページの「Source files」から、[Download Source] をクリックし、ソースコード版の Bootstrap をダウンロードします（図 10-15）。

▼図 10-15 Bootstrap のソースファイルをダウンロードする
(<https://getbootstrap.com/docs/4.1/getting-started/download/>)



ダウンロードファイルを解凍すると、図 10-16 のフォルダーが展開されます。

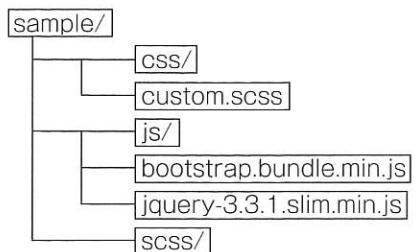
▼図 10-16 Bootstrap ソースファイルの構成



「dist」フォルダーには、コンパイル済みの CSS および JavaScript ファイルが入っています（①）。「docs」フォルダーにはドキュメント（解説）が入っています。その他にもライセンス情報や開発用のサポート情報などのファイルが入っています（②）。「docs」フォルダーの中の「examples」フォルダーには Bootstrap の使用例が入っています（③）。「js」フォルダーと「scss」フォルダーには、それぞれ JavaScript と CSS のソースコードが入っています（④）。

このうち必要なのは、CSS のソースコード式が入っている「scss」フォルダーです。これを作業フォルダー（本書では「sample」フォルダー）にコピーしましょう。これまでの sample フォルダーの中には、「css」や「js」フォルダーがありました。これらのフォルダーと同じ場所に「scss」フォルダーを配置します（図 10-17）。

▼図 10-17 作業フォルダーの構成



css フォルダーの中には、これまで bootstrap.css (または bootstrap.min.css) が格納されていましたが、これらは不要です。代わりにカスタマイズ用の Sass ファイル **custom.scss** を作成して保存します。

custom.scss には、scss フォルダー内の bootstrap.scss をインポートするように、リスト 10-7 のように記述します。この例は、Bootstrap を丸ごとインポートする設定です。

▼リスト 10-7 bootstrap.scss をインポートする (css/custom.scss)

```
// BootstrapのSassファイルをインポート
@import "../scss/bootstrap";
```

NOTE

部分的にインポートする

本書では Bootstrap のソースファイルの bootstrap.scss を丸ごとインポートしていますが、リスト 10-8 の例のように、必要な部分だけを選択して、部分的に読み込むこともできます。このとき、functions.scss、variables.scss、mixin.scss は必須です。また、各コンポーネント間には依存関係があるので注意してください。

▼リスト 10-8 部分的にインポートする

```
// 必須
@import "../scss/functions";
@import "../scss/variables";
@import "../scss/mixins";

// 任意
@import "../scss/reboot";
@import "../scss/type";
@import "../scss/images";
@import "../scss/code";
@import "../scss/grid";
```

では、custom.css をコンパイルしましょう。コマンドライン (Windows ならコマンドプロンプト、Mac ならターミナル) で css フォルダーに移動し、次の node-sass コマンドを実行します。

```
node-sass custom.scss custom.css
```

このコマンドで、custom.scss がコンパイルされ、custom.css に出力されます。custom.css を確認すると、膨大なスタイルが定義されているはずです。@import によって bootstrap.scss が読み込まれ、そのすべてがコンパイルされて書き出されているためです。これまで、HTML から参照させる CSS は、bootstrap.css (または bootstrap.min.css) でしたが、以降はこれを使用しません。コンパイルした custom.css を参照するよう指定します (リスト 10-9)。

▼リスト 10-9 custom.css を参照する

```
<link rel="stylesheet" href="css/custom.css">
```

10.2.3 背景色にグラデーションを使用できるようにする

手始めに、デフォルト設定では無効になっている**背景色のグラデーションを有効化**してみましょう。ソースファイルの scss/_variables.scss を開きます。Options の項目にある **\$enable-gradients** という変数を検索すると、値が false になっていることを確認できます (リスト 10-10)。

▼リスト 10-10 背景色のグラデーション設定 (sass/_variables.scss)

```
$enable-gradients: false !default;
```

この変数の値を変更することで、カスタマイズすることができます。このとき、_variables.scss を直接編集することもできますが、メンテナンス性を考えると、custom.scss にカスタマイズ内容を追記する方が良いでしょう。リスト 10-11 のように、Bootstrap の Sass ファイルをインポートする記述よりも前に、変数を上書きする記述を追加します。

▼リスト 10-11 背景色のグラデーションを有効化する (css/custom.scss)

```
// 変数の上書き
$enable-gradients: true;

// BootstrapのSassファイルをインポート
@import "../scss/bootstrap";
```

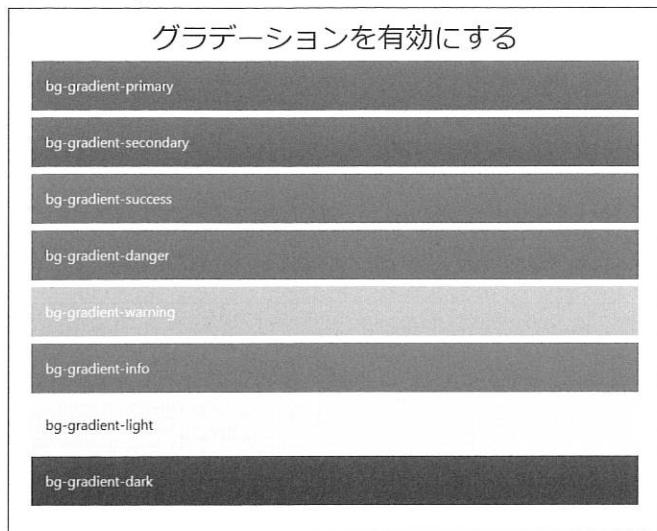
あとは前項と同じように、custom.scss をコンパイルして custom.css を作成します。次のサンプルで、背景のグラデーションが有効になっていたか確認できます (リスト 10-12、図 10-18)。

▼リスト 10-12 背景色のグラデーションを有効化した例 (gradient.html)

```
<div class="p-3 mb-2 bg-gradient-primary text-white">bg-gradient-primary</div>
<div class="p-3 mb-2 bg-gradient-secondary text-white">bg-gradient-primary</div>
<div class="p-3 mb-2 bg-gradient-success text-white">bg-gradient-primary</div>
<div class="p-3 mb-2 bg-gradient-danger text-white">bg-gradient-primary</div>
```

```
<div class="p-3 mb-2 bg-gradient-warning text-white">bg-gradient-primary</div>
<div class="p-3 mb-2 bg-gradient-info text-white">bg-gradient-primary</div>
<div class="p-3 mb-2 bg-gradient-light text-dark">bg-gradient-primary</div>
<div class="p-3 mb-2 bg-gradient-dark text-white">bg-gradient-primary</div>
```

▼図 10-18 背景色のグラデーションを有効化した例

1
2
3
4
5
6
7
8
9
10

10.2.4 Sass 変数を上書きする

前項では、_variables.scss に設定されている \$enable-gradients の値を custom.scss で変更して、背景のグラデーション化を有効にしましたが、同様の手順で変数の値を上書きし、CSS 設定を簡単に切り替えられます（表 10-1）。

▼表 10-1 カスタマイズ可能な主な Sass 変数

| 変数 | 値 | 説明 |
|-----------------------|------------------------|---|
| \$enable-rounded | true (デフォルト) または false | さまざまなコンポーネントで定義済みの border-radius スタイルを有効化 |
| \$enable-shadows | true または false (デフォルト) | さまざまなコンポーネントで定義済みの box-shadow スタイルを有効化 |
| \$enable-gradients | true または false (デフォルト) | さまざまなコンポーネントで background-image 経由で定義済みのグラデーションを有効化 |
| \$enable-transitions | true (デフォルト) または false | さまざまなコンポーネントで定義済みの transition を有効化 |
| \$enable-grid-classes | true (デフォルト) または false | グリッドシステムのための CSS クラスの生成を有効化 (例: container、row、col-md-1 クラスなど) |
| \$enable-caret | true (デフォルト) または false | dropdown-toggle の擬似要素キャレットを有効化 |
| \$enable-print-styles | true (デフォルト) または false | 印刷を最適化するスタイルを有効化 |

10.2.5 基本の配色を変更する

既述のように、Bootstrap 全体にかかるスタイルの初期設定は、_variables.scss にまとめられています。_variables.scss のどこで何が設定されているのかについては、ブロックごとにコメントが付けられており、配色については、_variables.scss の「//Color system」ではじまるコメントのブロックにまとめられています。最初に、白から黒のグレースケールが定義され、\$white、\$gray-100、\$black などの変数名でまとめられています（リスト 10-13）。

▼リスト 10-13 グレースケールの変数と色の設定 (_variables.scss)

```
$white:      #fff !default;
$gray-100:   #f8f9fa !default;
$gray-200:   #e9ecef !default;
$gray-300:   #dee2e6 !default;
$gray-400:   #ced4da !default;
$gray-500:   #adb5bd !default;
$gray-600:   #6c757d !default;
$gray-700:   #495057 !default;
$gray-800:   #343a40 !default;
$gray-900:   #212529 !default;
$black:      #000 !default;
```

primary、secondary などボタンやアラートにも使用されるカラーは、次のような変数でまとめられています。たとえば、\$blue と \$primary は紐づいているので、\$blue の値を変えると \$primary も変更されます（リスト 10-14）。

▼リスト 10-14 色とテーマカラーの設定 (_variables.scss)

```
$blue:      #007bff !default;
$indigo:   #6610f2 !default;
$purple:   #6f42c1 !default;
$pink:     #e83e8c !default;
$red:      #dc3545 !default;
$orange:   #fd7e14 !default;
$yellow:   #ffc107 !default;
$green:    #28a745 !default;
$teal:     #20c997 !default;
$cyan:    #17a2b8 !default;

$primary:   $blue !default;
$secondary:  $gray-600 !default;
$success:    $green !default;
$info:       $cyan !default;
$warning:   $yellow !default;
$danger:    $red !default;
$light:     $gray-100 !default;
$dark:      $gray-800 !default;
```

Bootstrap 4 では、Sass のマップというしくみでテーマカラーやグリッドのブレイクポイントなどのスタイルを生成しています。map とはキーと値のペアを定義できる連想配列のようなもので、Sass 変数と同様に、上書きして拡張できます。`_variables.scss` では、テーマカラーはリスト 10-15 のように定義されています。

▼リスト 10-15 色とテーマカラーの設定 (`_variables.scss`)

```
$theme-colors: () !default;
$theme-colors: map-merge(
  "primary": $primary,
  "secondary": $secondary,
  "success": $success,
  "info": $info,
  "warning": $warning,
  "danger": $danger,
  "light": $light,
  "dark": $dark
), $theme-colors);
```

`$theme-colors` マップに定義されている既存の色を変更するには、カスタマイズ用の Sass ファイル (`custom.scss`) にリスト 10-16 のような書式でコードを追加します。

▼リスト 10-16 既存のテーマカラーを変更する

```
$theme-colors: (
  "primary": #0074d9,
  "danger": #ff4136
);
```

また、`$theme-colors` に新しい色を追加するには、新しいキーと値を追加します（リスト 10-17）。

▼リスト 10-17 テーマカラーを新しく追加する

```
$theme-colors: (
  "custom-color": #900
);
```

`$theme-colors` やその他のマップから色を削除するには、`map-remove` を使用します（リスト 10-18）。

▼リスト 10-18 マップから色を削除する

```
$theme-colors: map-remove($theme-colors, "success", "info", "danger");
```

10.2.6 body の背景色、文字色、リンク色を変更する

body の背景色、文字色、リンク色は、`_variables.scss` の「// Body」ではじまるコメントのブロックにまとめられています（リスト 10-19）。

1
2
3
4
5
6
7
8
9
10

▼リスト 10-19 body の背景色と文字色の設定 (_variables.scss)

```
// Body
$body-bg: $white !default;
$body-color: $gray-900 !default;
```

body の背景色は \$body-bg、文字色は \$body-color という変数で設定されています。値には \$white や \$gray-900 という変数が入っていて、それぞれの変数に設定された内容に置き換えられます。これを変更するには、これまでのよう、カスタム用の SCSS ファイル (custom.scss) で変数を上書きしてコンパイルするだけです (リスト 10-20)。

▼リスト 10-20 body の背景色、文字色を変更する

```
// 変数の上書き
$body-bg: #eee;
$body-color: #000;

// BootstrapのSassファイルをインポート
@import "../scss/bootstrap";
```

またリンク色については、「// Links」ではじまるコメントのブロックにまとめられています (リスト 10-21)。

▼リスト 10-21 リンク色の設定 (_variables.scss)

```
// Links
$link-color: theme-color("primary") !default; ①
$link-decoration: none !default; ②
$link-hover-color: darken($link-color, 15%) !default; ③
$link-hover-decoration: underline !default; ④
```

リンク色は、\$link-color という変数です。値が目新しい書式ですが、これはテーマカラーの「primary」の設定が入るという意味です (①)。リンク時の下線、打ち消し線などの装飾は、\$link-decoration という変数で定義されており、デフォルトでは「なし」に設定されています (②)。

ホバー時の文字色は \$link-hover-color という変数で、darken(\$link-color, 15%) は、\$link-color で設定された色を 15% 暗くするという設定です (③)。ホバー時の装飾は \$link-hover-decoration という変数で、デフォルトでは「なし」に設定されています (④)。

以降も _variables.scss で何が設定されているか、カスタマイズ時によく利用される主な変数について、内容を確認していきましょう。

10.2.7 Spacing ユーティリティを変更する

- マージンやパディングなどの Spacing ユーティリティは、_variables.scss の「// Spacing」ではじまるコメントのブロックにまとめられています (リスト 10-22)。

▼リスト 10-22 スペーサーの設定 (_variables.scss)

```
// stylelint-disable
$spacer: 1rem !default;
$spacers: () !default;
$spacers: map-merge(
  0: 0,
  1: ($spacer * .25), ❶
  2: ($spacer * .5),
  3: $spacer,
  4: ($spacer * 1.5),
  5: ($spacer * 3)
), $spacers);
```

マージンやパディングの基本となる \$spacer の値には 1rem が設定されています (❶)。この \$spacer の値を変更することで、Spacing ユーティリティ (P.318 参照) の全スタイルを変更できます。たとえばデフォルトでは、\$spacer が 1rem に設定されています。数字の「1」は m-1 や p-1 クラスの「1」です。ここには、 $($spacer * .25)$ という式があるので、 $1rem * .25$ で、0.25rem 分のマージンやパディングが付きます (❷)。これを「`$spacer: 10px;`」に変更すると、m-1 や p-1 は「 $10px * .25$ 」で 2.5px 分のマージンやパディングが付き、m-3 や p-3 は 10px 分のマージンやパディングになります。また前項と同じように変更や追加も可能です (リスト 10-23)。

▼リスト 10-23 スペーサーの設定を変更する

```
$spacer: 10px; // 変更
$spacers: (
  1: ($spacer * .3), // 変更
  6: ($spacer * 3.5) // 追加
);
```

10.2.8 Sizing ユーティリティを変更する

同様に、w-25、h-50 クラスなどの Sizing ユーティリティはリスト 10-24 のような変数でまとめられています。

▼リスト 10-24 Sizing ユーティリティの設定 (_variables.scss)

```
$sizes: () !default;
$sizes: map-merge(
  25: 25%,
  50: 50%,
  75: 75%,
  100: 100%
), $sizes);
```

デフォルトでは、25%、50%、75%、100% が設定されていますが、こちらもマップを変更したり追加したりすることができます。リスト 10-25 の例では、width: 5%; のスタイルが定義された w-5 クラス、height: 5%;

のスタイルが定義された h-5 クラスが追加されます。

▼リスト 10-25 Sizing ユーティリティを追加する

```
$sizes: (
  5: 5% // 追加
);
```

10.2.9 ブレイクポイントを変更する

グリッドシステムのブレイクポイントは、_variables.scss の「// Grid breakpoints」ではじまるコメントのブロックにまとめられています（リスト 10-26）。

▼リスト 10-26 グリッドシステムのブレイクポイント設定 (_variables.scss)

```
$grid-breakpoints: (
  xs: 0,
  sm: 576px,
  md: 768px,
  lg: 992px,
  xl: 1200px
) !default;
```

これまで同様、\$grid-breakpoints のマップを変更することで、ブレイクポイントをカスタマイズすることができます（リスト 10-27）。

▼リスト 10-27 ブレイクポイントを変更する

```
$sizes: (
  sm: 420px // smのブレイクポイントを420pxに変更
);
```

10.2.10 コンテナを変更する

コンテナのブレイクポイントは、_variables.scss の「// Grid containers」ではじまるコメントのブロックにまとめられています（リスト 10-28）。

▼リスト 10-28 コンテナの最大幅の設定 (_variables.scss)

```
$container-max-widths: (
  sm: 540px,
  md: 720px,
  lg: 960px,
  xl: 1140px
) !default;
```

\$container-max-widths: のマップを変更することで、コンテナの最大幅をカスタマイズできます（リスト 10-29）。

▼リスト 10-29 コンテナの最大幅を変更する

```
$container-max-widths: (
  lg: 980px // lgの最大幅を980pxに変更
);
```

10.2.11 グリッドのカラム数やガター幅を変更する

グリッドのカラム数やガター幅は、_variables.scss の「// Grid columns」ではじまるコメントのブロックにまとめられています（リスト 10-30）。

▼リスト 10-30 グリッドのカラム数やガター幅の設定（_variables.scss）

```
$grid-columns: 12 !default;
$grid-gutter-width: 30px !default;
```

オリジナルのカラム数は 12 カラムを基本としていますが、\$grid-columns の値を変更することで 9 カラムや 16 カラムなど、独自のカラム数でグリッドシステムを構築できます。またオリジナルのカラム幅は、30px となっていますが、\$grid-gutter-width の値を変更することでカスタマイズできます（リスト 10-31）。

▼リスト 10-31 カラム数やガター幅を変更する

```
$grid-columns: 9;
$grid-gutter-width: 20px;
```

10.2.12 書式を変更する

フォントや、行の高さ、文字色などに関する設定は、_variables.scss の「// Font, line-height, and ……」ではじまるコメントのブロックにまとめられています。フォントファミリーの設定は、サンセリフ体のフォントは \$font-family-sans-serif、等幅フォントは \$font-family-monospace で種類別に設定可能です。また基本のフォントファミリーは、\$font-family-base で設定可能です（リスト 10-32）。

▼リスト 10-32 フォントの設定（_variables.scss）

```
$font-family-sans-serif: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, "Helvetica Neue", ↪
  Arial, sans-serif, "Apple Color Emoji", "Segoe UI Emoji", "Segoe UI Symbol" !default;
$font-family-monospace: SFMono-Regular, Menlo, Monaco, Consolas, "Liberation Mono", "Courier New", ↪
  monospace !default;
$font-family-base: $font-family-sans-serif !default;
```

フォントファミリーに続いて、フォントサイズ（大きさ）やウェイト（太さ）、見出しの書式なども変数でまとめられています。詳しくは _variables.scss を確認してください。

1
2
3
4
5
6
7
8
9
10

10

SECTION

3

CSS 変数を利用する

Bootstrap 4 のコンパイルされた CSS には、CSS 変数が組み込まれています。CSS 変数とは、Sass のように CSS で変数を扱えるようにしたものです。Bootstrap では、何度も繰り返し利用される色、ブレイクポイント、フォントファミリーが CSS 変数として定義されています。本章の冒頭で説明したオリジナルスタイルを上書きする方法でカスタマイズする際に、効率よくコードを書くことができます。

10.3.1 Bootstrap で定義されている CSS 変数

Bootstrap で定義されている CSS 変数はリスト 10-33 のとおりです。`:root` は、文書のルート要素を対象とした擬似セレクタです。CSS 変数はどのようなセレクタでも利用できますが、`:root` を利用することで、どこでも使えるようにしています。

▼リスト 10-33 Bootstrap 4 で利用可能な CSS 変数 (bootstrap.css)

```
:root {  
  --blue: #007bff;  
  --indigo: #6610f2;  
  --purple: #6f42c1;  
  --pink: #e83e8c;  
  --red: #dc3545;  
  --orange: #fd7e14;  
  --yellow: #ffc107;  
  --green: #28a745;  
  --teal: #20c997;  
  --cyan: #17a2b8;  
  --white: #fff;  
  --gray: #868e96;  
  --gray-dark: #343a40;  
  --primary: #007bff;  
  --secondary: #868e96;  
  --success: #28a745;  
  --info: #17a2b8;  
  --warning: #ffc107;  
  --danger: #dc3545;  
  --light: #f8f9fa;  
  --dark: #343a40;  
  --breakpoint-xs: 0;  
  --breakpoint-sm: 576px;
```

```
--breakpoint-md: 768px;  
--breakpoint-lg: 992px;  
--breakpoint-xl: 1200px;  
--font-family-sans-serif: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, "Helvetica Neue", Arial, sans-serif, "Apple Color Emoji", "Segoe UI Emoji", "Segoe UI Symbol";  
--font-family-monospace: "SFMono-Regular", Menlo, Monaco, Consolas, "Liberation Mono", "Courier New", monospace;  
}
```

「--」という接頭辞が付いているのが CSS 変数です。**--blue: #007bff;** のように、「--」という接頭辞の後に独自のプロパティ名を付けて宣言します。

作成した CSS 変数を利用する場合は、使いたい箇所で **var(-- 変数名)** を記述します。CSS 変数は Sass のように柔軟に利用できますが、コンパイルの必要はありません。リスト 10-34 の例では a 要素のスタイルを CSS 変数で設定しています。

▼リスト 10-34 CSS 変数の利用例

```
a {  
  color: var(--blue);  
}
```

また、リスト 10-35 の例では、メディアクエリでブレイクポイントの変数を使用しています。このように CSS 変数を利用することで、スタイルの一貫性を保つことができ、メンテナンス性も高くなるでしょう。

▼リスト 10-35 CSS 変数の利用例

```
.content-secondary {  
  display: none;  
}  
@media (min-width(var(--breakpoint-sm))) {  
  .content-secondary {  
    display: block;  
  }  
}
```

1
2
3
4
5
6
7
8
9
10



CSS 変数のソースファイル

CSS 変数の元々の設定内容はソースファイルの _root.scss ファイルで確認できます（リスト 10-36）。

▼リスト 10-36 CSS 変数の設定 (_root.scss)

```
:root {
  @each $color, $value in $colors {
    --#{$color}: #{$value};
  }
}

@each $color, $value in $theme-colors {
  --#{$color}: #{$value};
}

@each $bp, $value in $grid-breakpoints {
  --breakpoint-#{$bp}: #{$value};
}
```

最初の @each では、色の変数 \$color における、色と値 (\$color と \$value) のペアごとに、CSS 変数の宣言を書き出す処理をしています（①）。次の @each でも、テーマカラーの変数 \$theme-colors における色と値 (\$color と \$value) のペアごとに CSS 変数を作っています（②）。最後の @each では、グリッドのブレイクポイントの変数 \$grid-breakpoint におけるブレイクポイントと値 (\$bp と \$value) のペアごとに CSS 変数の宣言を書き出す処理をしています（③）。

10.3.2 ミックスインを利用する

これまでに、_variables.scss に設定されている変数の値を変更してコンパイルするだけの簡単なカスタマイズ方法を紹介しましたが、本項では、ミックスインを使用した、より応用的な方法を紹介します。

■ ミックスインの基本

ミックスインとは、簡単に言うと、汎用的なスタイルを定義しておき、それを他の場所で呼び出して使うことができる機能です。Bootstrap のミックスインのファイルは、scss フォルダ内の mixins フォルダに格納されています。この中で、角丸の設定が定義されている「_border-radius.scss」を見てみましょう（リスト 10-37）。

▼リスト 10-37 角丸を定義したミックスイン (_border-radius.scss)

```
@mixin border-radius($radius: $border-radius) {
  @if $enable-rounded {
    border-radius: $radius;
  }
}
```

…以下略

ミックスインの書式は、@ mixin の後に半角スペースを空けて、任意のミックスイン名を定義します。引数を使用する場合、ミックスイン名の直後に()を付けて、括弧内に引数を書き、続く{}内にスタイルを定義します。

@ mixin の半角スペースの後の「border-radius」がミックスイン名、()内の\$radius: \$border-radiusは、変数\$radiusに_variables.scssで定義されている\$border-radius（デフォルトで.25rem）を引数としてセットしています（①）。続く@if \$enable-roundedは、_variables.scssで定義されている\$enable-roundedの値（P.447 参照）がtrueの場合（②）、border-radius プロパティの値を\$radiusに設定する（③）という意味になります。

Sass の知識がないと少し難しいかもしれません、ここでは、border-radius というミックスインが定義されていることがわかれればOKです。

角丸ミックスインを追加する

Bootstrapには、角丸に設定するroundedクラスがありますが、右上、右下、左上、左下のように角を1つ1つ設定するクラスは定義されていません（P.308 参照）。そこで、_bordar-radius.scssに、四隅のそれぞれを角丸にするミックスインを追加してみましょう（リスト 10-38）。

▼リスト 10-38 bordar-radius.scss

```
// 左上の角丸
@mixin border-tl-radius($radius) {
  @if $enable-rounded {
    border-top-left-radius: $radius;
  }
}

// 右上の角丸
@mixin border-tr-radius($radius) {
  @if $enable-rounded {
    border-top-right-radius: $radius;
  }
}

// 左下の角丸
@mixin border-bl-radius($radius) {
  @if $enable-rounded {
    border-bottom-left-radius: $radius;
  }
}

// 右下の角丸
@mixin border-br-radius($radius) {
  @if $enable-rounded {
    border-bottom-right-radius: $radius;
  }
}
```

1
2
3
4
5
6
7
8
9
10

■ 角丸クラスを定義する

続いて、追加で作成したミックスインを利用したクラスを定義しましょう。css フォルダ内の custom.scss ファイルを開き、ファイルの末尾に次のコードを追加します（リスト 10-39）。

▼リスト 10-39 角丸クラスを定義する（custom.scss）

```
// 角丸クラスを定義
.rouded-tl {
  @include border-tl-radius(10px);
}
.rouded-tr {
  @include border-tr-radius(10px);
}
.rouded-bl {
  @include border-bl-radius(10px);
}
.rouded-br {
  @include border-br-radius(10px);
}
```

@include で、先ほど作成したミックスインを組み込んでいます。（）内の引数には 10px を指定していますが、値を空にした場合はデフォルトの .25rem が設定されます。

追記できたら、custom.scss を再度コンパイルし、custom.css を開いてみましょう。ファイルの最後に以下のコードが追加されていることが確認できるはずです（リスト 10-40）。

▼リスト 10-40 コンパイル後に追加生成されるクラス（custom.css）

```
.rouded-tl {
  border-top-left-radius: 10px; }

.rouded-tr {
  border-top-right-radius: 10px; }

.rouded-bl {
  border-bottom-left-radius: 10px; }

.rouded-br {
  border-bottom-right-radius: 10px; }
```

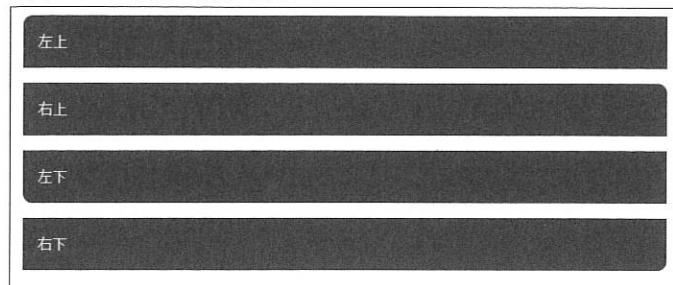
■ 作成したクラスを利用する

作成した rouded-tl、rouded-tr、rouded-bl、rouded-br クラスを、実際に使用してみましょう（リスト 10-41）。左上、右上、左下、右下の角をそれぞれ個別に角丸設定できるようになりました（図 10-19）。

▼リスト 10-41 角丸を個別に設定する (rounded.html)

```
<div class="container">
  <div class="rounded-tl p-3 mb-3 bg-dark text-white">左上</div>
  <div class="rounded-tr p-3 mb-3 bg-dark text-white">右上</div>
  <div class="rounded-bl p-3 mb-3 bg-dark text-white">左下</div>
  <div class="rounded-br p-3 mb-3 bg-dark text-white">右下</div>
</div>
```

▼図 10-19 角丸を個別に設定する

1
2
3
4
5
6
7
8
9
10

INDEX

■ 記号・数字

| | |
|-----------------------------|---------|
| :root | 454 |
| _reboot.scss | 68 |
| _type.scss | 68 |
| _variables.scss | 68 |
| - | 455 |
| <h1> ~ <h6> | 48 |
| < > | 64 |
| \$enable-caret | 447 |
| \$enable-gradients | 446,447 |
| \$enable-grid-classes | 447 |
| \$enable-print-styles | 447 |
| \$enable-rounded | 447 |
| \$enable-shadows | 447 |
| \$enable-transitions | 447 |
| 12カラム | 34 |

■ A

| | |
|-----------------------------|-------------------------------------|
| a | 103 |
| abbr | 57,105 |
| accordion | 277 |
| activate.bs.scrollspy | 299 |
| active | 153,179,180,185,191,236,251,263,295 |
| address | 104 |
| alert | 110 |
| alert-* | 110 |
| alert-dismissible | 113 |
| alert-heading | 113 |
| alert-link | 111 |
| align-content-around | 336 |
| align-content-between | 336 |
| align-content-center | 336 |
| align-content-end | 336 |
| align-content-start | 336 |
| align-items | 36 |
| align-items-baseline | 327 |
| align-items-center | 36,186,207,327 |
| align-items-end | 36,327 |
| align-items-start | 36,188,327 |
| align-items-stretch | 327 |
| align-self-* | 145 |
| align-self-baseline | 329 |
| align-self-center | 37,145,329 |
| align-self-end | 37,145,329 |
| align-self-start | 37,145,329 |
| align-self-stretch | 329 |
| aria-* | 119,179,190,274 |
| aria-controls | 274 |
| aria-expanded | 245,274 |
| aria-haspopup | 245 |
| aria-hidden | 392 |
| aria-label | 189,202,240,392 |
| aria-labelledby | 245 |
| aria-valuemax | 119 |

| | |
|---------------------|-----|
| aria-valuemin | 119 |
| aria-valuenow | 119 |

■ B

| | |
|------------------------------|--------------------|
| b | 54 |
| backdrop | 291 |
| badge | 115,187,430 |
| badge-* | 115,430 |
| badge-light | 116 |
| badge-pill | 117,187 |
| badge-primary | 187 |
| bg-* | 85,120,129,161,169 |
| bg-danger | 304 |
| bg-dark | 161,304 |
| bg-info | 304 |
| bg-light | 304,387 |
| bg-primary | 129,304 |
| bg-secondary | 304 |
| bg-success | 129,304,406 |
| bg-transparent | 131,304 |
| bg-warning | 304 |
| bg-white | 304 |
| blockquote | 58,105 |
| blockquote-footer | 59 |
| bootstrap.bundle.js | 16 |
| bootstrap.css | 16,68 |
| bootstrap.js | 16 |
| bootstrap-grid.css | 16 |
| bootstrap-reboot.css | 16 |
| border | 306,399 |
| Border | 306 |
| border-* | 131 |
| border-0 | 307 |
| border-bottom | 306 |
| border-bottom-0 | 307 |
| border-danger | 307 |
| border-dark | 307 |
| border-info | 307 |
| border-left | 306 |
| border-left-0 | 307 |
| border-light | 307 |
| border-primary | 131,307 |
| border-right | 306 |
| border-right-0 | 307 |
| border-secondary | 307 |
| border-success | 307 |
| border-top | 306 |
| border-top-0 | 307,399 |
| border-warning | 307 |
| border-white | 307 |
| boundary | 255 |
| box-sizing | 97 |
| box-sizing:border-box: | 97 |
| breadcrumb | 179,428 |

| | | | |
|----------------------------|-------------------------------------|----------------------|-------------------------|
| breadcrumb-item | 179,428 | collapse | 163,176,178,272,277,375 |
| btn | 181,229,233,238,245,284,389,391,431 | collapsing | 272 |
| btn-* | 168,229,233,238,245,284,389,391 | Color | 85,302 |
| btn-block | 235 | column-count | 142 |
| btn-group | 238,240,246 | container | 23,40,170,371,379 |
| btn-group-lg | 242 | container-fluid | 23,287 |
| btn-group-sm | 242 | CSS ソースマップファイル | 15 |
| btn-group-vertical | 243 | CSS 変数 | 454 |
| btn-lg | 235 | custom.css | 368 |
| btn-outline-* | 234 | custom.scss | 445 |
| btn-primary | 431 | custom-* | 216 |
| btn-sm | 235 | custom-checkbox | 216 |
| btn-toolbar | 240 | custom-control | 216 |
| button | 102 | custom-control-* | 216 |
| C | | | |
| caption | 87 | custom-control-input | 216 |
| card | 124,391,405 | custom-control-label | 217 |
| card-body | 124,391,406 | custom-file-input | 221 |
| card-columns | 140,142 | custom-radio | 216 |
| card-deck | 139 | custom-range | 222 |
| card-footer | 124,128,138,406 | custom-select | 218,231 |
| card-group | 138 | custom-select-lg | 219 |
| card-header | 124,128,405 | custom-select-sm | 219 |
| card-header-pills | 136 | | |
| card-header-tabs | 135 | | |
| card-img | 137 | | |
| card-img-bottom | 124 | | |
| card-img-overlay | 137 | | |
| card-img-top | 124 | | |
| card-link | 124 | | |
| card-subtitle | 124 | | |
| card-text | 124,406 | | |
| card-title | 124,391,406 | | |
| carousel | 263,379 | | |
| carousel-caption | 266,380 | | |
| carousel-control-next | 264,379 | | |
| carousel-control-next-icon | 264,381 | | |
| carousel-control-prev | 264,379 | | |
| carousel-control-prev-icon | 264,381 | | |
| carousel-fade | 266 | | |
| carousel-indicators | 265,379 | | |
| carousel-inner | 263,379 | | |
| carousel-item | 263,379 | | |
| CDN | 19 | | |
| checked | 217 | | |
| cite | 59 | | |
| clearfix | 339 | | |
| close | 113,355,392 | | |
| code | 64 | | |
| col | 25,26,35 | | |
| col-* | 26,29,63,203,384,389,401,409 | | |
| col-auto | 31,207 | | |
| col-form-label | 206,430 | | |
| col-form-label-lg | 206 | | |
| col-form-label-sm | 206 | | |
| col-lg | 35 | | |
| col-lg-* | 29 | | |
| col-md | 35 | | |
| col-md-* | 29,430 | | |
| col-md-auto | 31 | | |
| col-sm-* | 29,30 | | |
| col-sm-6 | 133 | | |
| col-xl-* | 29 | | |

D

| | |
|----------------------|-------------------------------------|
| d-block | 72,263,310 |
| d-flex | 46,186,188,310,322,391 |
| d-inline | 310 |
| d-inline-block | 310 |
| d-inline-flex | 310,322 |
| d-md-block | 33,266,380 |
| d-none | 33,266,310,380 |
| d-print | 162 |
| d-print-block | 313 |
| d-print-flex | 313 |
| d-print-inline | 313 |
| d-print-inline-block | 313 |
| d-print-inline-flex | 313 |
| d-print-none | 313 |
| d-print-table | 313 |
| d-print-table-cell | 313 |
| d-print-table-row | 313 |
| d-table | 310 |
| d-table-cell | 310 |
| d-table-row | 310 |
| danger | 111 |
| dark | 111 |
| data-* | 114,268 |
| data-dismiss | 392 |
| data-parent | 277 |
| data-ride | 263 |
| data-slide-to | 265 |
| data-spy | 295 |
| data-target | 265,295 |
| data-toggle | 274 |
| data-toggle="pill" | 160 |
| data-toggle="tab" | 160 |
| dd | 63 |
| del | 55 |
| disabled | 151,181,182,191,201,209,218,236,251 |
| display | 253 |
| Display | 33,46,310 |
| display: flex; | 138,143 |
| display-* | 50,108 |
| div | 165 |

| | |
|----------------------------|--------------------------|
| dl | 63 |
| dropdown | 158,245,375 |
| dropdown-divider | 158,251 |
| dropdown-header | 250 |
| dropdown-item | 158,230,245,375,420 |
| dropdown-item-text | 252 |
| dropdown-menu | 158,230,245,375 |
| dropdown-menu-right | 249 |
| dropdown-toggle | 158,230,245,375 |
| dropdown-toggle-split | 230,247 |
| dropleft | 247 |
| dropright | 247 |
| dropup | 247 |
| dt | 63 |
| E | |
| em | 30,57 |
| Embed | 356 |
| embed-responsive | 356,413 |
| embed-responsive-16by1 | 356 |
| embed-responsive-16by9 | 356,411 |
| embed-responsive-1by1 | 411 |
| embed-responsive-21by9 | 356,411 |
| embed-responsive-4by3 | 356,411 |
| embed-responsive-item | 356 |
| F | |
| fade | 113,285,392,399 |
| fieldset | 101 |
| figcaption | 93 |
| figure | 93 |
| figure-img | 93 |
| figure-caption | 93 |
| fixed-bottom | 171,173,344 |
| fixed-top | 171,172,343 |
| Flex | 36,46,322 |
| flex-column | 152,156,188,323 |
| flex-column-reverse | 323 |
| flex-fill | 331 |
| flex-grow-0 | 331 |
| flex-grow-1 | 331 |
| flexnowrap | 333 |
| flex-row | 323 |
| flex-row-reverse | 323 |
| flex-shrink-0 | 331 |
| flex-shrink-1 | 331 |
| flex-sm-fill | 156 |
| flex-sm-row | 156 |
| flex-wrap | 333 |
| flex-wrap-reverse | 333 |
| flexbox | 28,322 |
| Flexible Box | 322 |
| Flexible Box Layout Module | 28 |
| flex アイテム | 322 |
| flex コンテナ | 322 |
| flip | 254 |
| Float | 71,338 |
| float* | 338 |
| float-left | 71 |
| float-right | 71 |
| focus true | 291 |
| font-* | 57 |
| font-italic | 350 |
| font-size | 97 |
| font-weight | 52 |
| font-weight-bold | 350 |
| font-weight-normal | 350 |
| font-weight-light | 350 |
| form-check | 201,430 |
| form-check-inline | 202,431 |
| form-check-input | 201,430 |
| form-check-label | 201,430 |
| form-control | 167,198,223,224,430 |
| form-control-file | 198 |
| form-control-lg | 198 |
| form-control-plaintext | 200 |
| form-control-range | 199 |
| form-control-sm | 198 |
| form-group | 198,430 |
| form-inline | 163,167,208 |
| form-row | 204 |
| form-text | 209 |
| G | |
| Google | 8 |
| Google マップ | 411 |
| Grid System | 9 |
| H | |
| h-100 | 315 |
| h-25 | 315 |
| h-50 | 315 |
| h-75 | 315 |
| h-auto | 315 |
| h1 ~ h6 | 48 |
| help-block | 209 |
| hidden | 105 |
| hidden.bs.collapse | 281 |
| hidden.bs.modal | 292 |
| hide.bs.collapse | 281 |
| hide.bs.modal | 292 |
| I | |
| img-fluid | 69,72,93,380,389,391,402 |
| img-thumbnail | 70,72 |
| indeterminate | 217 |
| info | 111 |
| initialism | 57 |
| input | 102 |
| input-group | 168,223,226,241 |
| input-group-append | 224,225 |
| input-group-lg | 226 |
| input-group-prepend | 168,223,225 |
| input-group-sm | 226 |
| input-group-text | 168,223,224 |
| ins | 56 |
| interval | 268 |
| invalid-feedback | 211,214 |
| invalid-tooltip | 213 |
| invisible | 106,354 |
| is-invalid | 214 |
| is-valid | 214 |
| J | |
| JavaScript プラグイン | 260 |
| jQuery | 17 |
| jumbotron | 108 |
| jumbotron-fluid | 109 |

| | |
|-------------------------------|-------------------------|
| justify-content-* | 151 |
| justify-content-around | 38,325 |
| justify-content-between | 38,186,188,325,391 |
| justify-content-center | 38,151,193,325,417 |
| justify-content-end | 38,151,193,325,431 |
| justify-content-start | 38,325 |
| K | |
| kbd | 66 |
| keyboard | 268,291 |
| L | |
| label | 102,225 |
| lead | 52,108 |
| legend | 101 |
| li | 61 |
| light | 111 |
| list-group | 126,180,181 |
| list-group-flush | 126 |
| list-group-item | 180 |
| list-group-item-* | 184 |
| list-group-item-action | 182 |
| list-inline | 62 |
| list-inline-item | 62 |
| list-unstyled | 61,147 |
| M | |
| m-0 | 319 |
| m-1 | 319 |
| m-2 | 319 |
| m-3 | 319 |
| m-4 | 319 |
| m-5 | 319 |
| m-auto | 319 |
| map | 449 |
| mark | 54 |
| mb-* | 389,391,399,405,409,417 |
| mb-0 | 113,319 |
| mb-1 | 319 |
| mb-2 | 319 |
| mb-3 | 319 |
| mb-4 | 148,319 |
| mb-5 | 319 |
| mb-auto | 319 |
| media | 143 |
| media-body | 143 |
| mh-100 | 317 |
| ml-0 | 319 |
| ml-1 | 319 |
| ml-2 | 319 |
| ml-3 | 147,319 |
| ml-4 | 319 |
| ml-5 | 319 |
| ml-auto | 168,319,334 |
| ml-md-auto | 44 |
| modal | 284,392 |
| modal-body | 284,392 |
| modal-content | 284,392 |
| modal-dialog | 284,392 |
| modal-dialog-centered | 286,392 |
| modal-footer | 284,392 |
| modal-header | 284,392 |
| modal-lg | 289 |
| modal-sm | 289 |
| modal-title | 284,392 |
| mr-0 | 319 |
| mr-1 | 319 |
| mr-2 | 240,319 |
| mr-3 | 144,146,148,319 |
| mr-4 | 319 |
| mr-5 | 319 |
| mr-auto | 319,334,375 |
| mt-0 | 319 |
| mt-1 | 319 |
| mt-2 | 319 |
| mt-3 | 145,319 |
| mt-4 | 319 |
| mt-5 | 319 |
| mt-auto | 319 |
| mw-100 | 316 |
| mx-0 | 319 |
| mx-1 | 319 |
| mx-2 | 319 |
| mx-3 | 319 |
| mx-4 | 319 |
| mx-5 | 319 |
| mx-auto | 71,72,319,321,405 |
| my-0 | 319 |
| my-1 | 319 |
| my-2 | 319 |
| my-3 | 319 |
| my-4 | 319 |
| my-5 | 319 |
| my-auto | 319 |
| N | |
| nav | 135,150,157,399,417 |
| nav-fill | 154 |
| nav-item | 150,155,375,399,417 |
| nav-justified | 155 |
| nav-link | 150,375,399,417,420 |
| nav-pill | 154 |
| nav-pills | 136,158 |
| nav-tabs | 135,153,399 |
| navbar | 161,373 |
| navbar-* | 161 |
| navbar-brand | 163,177,373 |
| navbar-collapse | 163,175,176,375 |
| navbar-dark | 161,169 |
| navbar-expand | 175 |
| navbar-expand-* | 161,175 |
| navbar-expand-lg | 161 |
| navbar-light | 169 |
| navbar-nav | 163,375 |
| navbar-text | 163,168 |
| navbar-toggler | 163,175,177,374 |
| navbar-toggler-icon | 163,374 |
| no-gutters | 39 |
| Node.js | 439 |
| node-sass | 439,443 |
| Normalize.css | 96 |
| Normalize CSS | 96 |
| novalidate | 211 |
| O | |
| Object-Oriented CSS | 434 |
| offset | 253 |
| offset-0 | 43 |

| | |
|-----------------------|--|
| offset-md-* | 43 |
| offset-sm-* | 43 |
| ol | 61 |
| OOCSS | 434 |
| order-* | 42, 146, 334, 401 |
| order-first | 42 |
| P | |
| p-* | 401 |
| p-0 | 320 |
| p-1 | 320 |
| p-2 | 320 |
| p-3 | 320 |
| p-4 | 320 |
| p-5 | 320 |
| page-item | 189 |
| pagination | 189 |
| pagination-lg | 193 |
| pagination-sm | 193 |
| parent | 279 |
| pause | 268 |
| pb-0 | 320 |
| pb-1 | 320 |
| pb-2 | 320 |
| pb-3 | 320 |
| pb-4 | 320 |
| pb-5 | 320 |
| pl-0 | 320 |
| pl-1 | 320 |
| pl-2 | 320 |
| pl-3 | 320 |
| pl-4 | 320 |
| pl-5 | 320 |
| Popper.js | 260 |
| Position | 342 |
| position-absolute | 342 |
| position-fixed | 342 |
| position-relative | 342 |
| position-static | 202, 342 |
| position-sticky | 342 |
| pr-0 | 320 |
| pr-1 | 320 |
| pr-2 | 320 |
| pr-3 | 320 |
| pr-4 | 320 |
| pr-5 | 320 |
| pre | 65, 100 |
| pre-scrollable | 65 |
| primary | 111 |
| progress | 118 |
| progress-bar | 118 |
| progress-bar-animated | 122 |
| progress-bar-striped | 122 |
| pt-0 | 320 |
| pt-1 | 320 |
| pt-2 | 320 |
| pt-3 | 320 |
| pt-4 | 320 |
| pt-5 | 320 |
| px | 30 |
| px-0 | 320 |
| px-1 | 320 |
| px-2 | 320 |
| px-3 | 320 |
| px-4 | 320 |
| px-5 | 320 |
| py-* | 370 |
| py-0 | 320 |
| py-1 | 320 |
| py-2 | 320 |
| py-3 | 320 |
| py-4 | 320 |
| py-5 | 320 |
| R | |
| readonly | 199 |
| Reboot | 16, 96 |
| rem | 30, 97 |
| required | 211 |
| Reset CSS | 96 |
| Responsive Web Design | 4 |
| ride | 268 |
| role | 119, 179, 234, 240, 274 |
| rounded | 308 |
| rounded-0 | 308 |
| rounded-bottom | 308 |
| rounded-circle | 308 |
| rounded-left | 308 |
| rounded-right | 308 |
| rounded-top | 308 |
| row | 24, 40, 63, 384, 389, 401, 409, 430 |
| S | |
| s | 55 |
| samp | 67 |
| Sass | 16, 439 |
| Sass 变数 | 447 |
| secondary | 111 |
| select | 102 |
| Sizing | 32, 133, 314 |
| shadow | 357 |
| shadow-lg | 357 |
| shadow-none | 357 |
| shadow-sm | 357 |
| show | 113, 272, 278, 291, 399 |
| show.bs.collapse | 281 |
| show.bs.modal | 292 |
| shown.bs.collapse | 281 |
| shown.bs.modal | 292 |
| slid.bs.carousel | 271 |
| slide | 263, 379 |
| slide.bs.carousel | 271 |
| small | 53 |
| Spacing | 46, 318, 334 |
| sr-only | 116, 186, 190, 191, 208, 232, 264, 354 |
| sr-only-focusable | 354 |
| Starter template | 368 |
| Sticky | 342 |
| sticky-top | 171, 174, 346, 373, 421 |
| strong | 54 |
| success | 111 |
| SVG | 70 |
| T | |
| tab-content | 160, 399 |
| tab-pane | 399 |
| tabindex | 191, 237 |
| table | 73, 88, 100, 402, 410 |

| | |
|---------------------|--------------------------|
| table-bordered | 78 |
| table-borderless | 79 |
| table-dark | 75 |
| table-hover | 80,410 |
| table-responsive | 88 |
| table-responsive-* | 90 |
| table-responsive-sm | 90 |
| table-sm | 81 |
| table-striped | 77,402 |
| td | 84 |
| Text | 57,60,71,347 |
| text-* | 57,129 |
| text-black-50 | 302 |
| text-body | 302 |
| text-capitalize | 349 |
| text-center | 60,71,94,134,348,371,405 |
| text-danger | 302 |
| text-dark | 302,405 |
| text-info | 302 |
| text-justify | 347,406 |
| text-left | 94,348 |
| text-lg-center | 348 |
| text-lg-left | 348 |
| text-lg-right | 348 |
| text-light | 302 |
| text-lowercase | 349 |
| text-monospace | 351 |
| text-muted | 50,209,302 |
| text nowrap | 348 |
| text-primary | 302 |
| text-right | 60,94,134,348 |
| text-secondary | 302 |
| text-sm-center | 156,348 |
| text-sm-left | 348 |
| text-sm-right | 348 |
| text-success | 302 |
| text-truncate | 63,349 |
| text-uppercase | 349 |
| text-warning | 302 |
| text-white | 129,302,406 |
| text-white-50 | 302 |
| text-xl-center | 348 |
| text-xl-left | 348 |
| text-xl-right | 348 |
| textarea | 102,103 |
| th | 84 |
| thead | 76 |
| thead-dark | 76 |
| thead-light | 76 |
| toggle | 279 |
| Twitter Blueprint | 11 |

■ U

| | |
|-----------|------------|
| u | 56 |
| ul | 61,150,165 |
| unchecked | 217 |
| util.js | 260 |

■ V

| | |
|----------------|------------|
| valid-feedback | 211 |
| valid-tooltip | 213 |
| var | 66,455 |
| visible | 354 |
| Visibility | 46,106,354 |

■ W

| | |
|---------------------|----------------|
| w-* | 119,132 |
| w-100 | 32,188,263,314 |
| w-25 | 314 |
| w-50 | 314 |
| w-75 | 314,405 |
| w-auto | 314 |
| WAI-ARIA | 234,278 |
| warning | 111 |
| was-validated | 212 |
| Web アプリケーションフレームワーク | 2 |
| wrap | 268 |

■ 行

| | |
|--------------|---------------------|
| アクセシビリティ | 116,157,186,232,354 |
| アコードイオン | 275 |
| アシスティブテクノロジー | 57 |
| アスペクト比 | 356,411 |
| アラート | 110 |
| 暗色テーブル | 75 |
| イタリック体 | 57 |
| 入れ子 | 45 |
| インジケーター | 262,265 |
| インプラウザデザイン | 404 |
| 引用文 | 58 |
| 引用元 | 59 |
| オンラインキリスト | 53 |
| オンラインリスト | 62 |
| ウェル | 124 |
| オフセット値 | 253 |
| 折り畳み | 272 |

■ 力行

| | |
|-------------|-------------|
| カード | 124,386,404 |
| カードカラム | 140 |
| カードグループ | 138 |
| カードデッキ | 139 |
| 開閉パネル | 272 |
| カスタマイズ用 CSS | 434 |
| カスタムフォーム | 216 |
| ガター | 22,453 |
| 可変幅コンテナ | 23 |
| カラム | 22,25,453 |
| カルーセル | 262,377 |
| 疑似セレクタ | 454 |
| キャプション | 87,262 |
| 行 | 25 |
| 強調したいテキスト | 57 |
| 切り替えボタン | 373 |
| グリッドシステム | 5,9,22,25 |
| グリッドレイアウト | 287 |
| グレースケール | 448 |
| 罫線付きのテーブル | 78 |
| 罫線なしのテーブル | 79 |
| コード | 64 |
| コードブロック | 65 |
| 交差軸 | 324 |
| 構造 | 434 |
| 固定幅コンテナ | 23 |
| コンテナ | 23,25 |
| コンテナの変更 | 452 |
| コントローラー | 262 |
| コンポーネント | 9 |

■ サ行

| | |
|-----------|---------------------|
| 再起動 | 96 |
| 細目 | 53 |
| 削除されたテキスト | 55 |
| サブコンポーネント | 162 |
| サムネイル | 124 |
| サムネイル画像 | 70 |
| サンプル出力 | 67 |
| 支援技術 | 264,274 |
| 指定幅カラム | 26 |
| 縞模様のテーブル | 77,396 |
| 斜体 | 57 |
| ジャンボトロン | 108,377 |
| 主軸 | 323 |
| 書式 | 453 |
| スキン | 434 |
| スクリーンリーダー | 116,157,186,232,354 |
| スクロールスパイ | 293 |
| 図表キャプション | 94 |
| スプリットボタン | 246 |

■ タ行

| | |
|----------------|-------------|
| ダイアログ | 283 |
| タブ型ナビゲーション | 153,396 |
| チェックボックス | 200 |
| 注釈 | 53 |
| 次へ送る | 264 |
| データ属性 | 268 |
| データ属性 API | 114,261,279 |
| テーブル | 73 |
| テーマカラー | 449 |
| 定義リスト | 63,383 |
| テンプレート | 2 |
| 等幅カラム | 26 |
| 取り消し線 | 55 |
| ドロップダウン | 166,230,244 |
| ドロップダウンナビゲーション | 157 |

■ ナ行

| | |
|------------|-------------------------|
| ナビゲーション | 135,150,165,293,374,416 |
| ナビゲーションバー | 161,372 |
| 入力グループ | 167,223 |
| ネスト | 45 |
| ノーマライズ CSS | 96 |

■ ハ行

| | |
|-------------|-----------------|
| 背景色のグラデーション | 446 |
| ハイライト表示 | 54 |
| バッジ | 115,423 |
| バディング | 450 |
| パネル | 124 |
| パンくずリスト | 179,423,427 |
| ハンバーガーアイコン | 374 |
| ビル型ナビゲーション | 154 |
| フォーム | 167,196,423,428 |
| フォームグループ | 198 |
| フォントウェイト | 52 |
| フォントサイズ | 52 |
| フォントファミリー | 453 |
| 太字 | 54 |
| 部品化 | 2 |
| ブランド | 163,373 |
| フルードイメージ | 5 |
| フルードグリッド | 5 |

| | |
|-------------|-----------------------------|
| フルスクリーン表示 | 356 |
| フレームワーク | 2 |
| ブレイクポイント | 7,29,42,156,342,348,436,449 |
| ブレイクポイントの変更 | 452 |
| フレックスアイテム | 46 |
| プログレス | 118 |
| プログレスバー | 118 |
| ページネーション | 189 |
| ヘルプカーソル | 57 |
| 変数 | 66,454 |
| ボタン | 229,233,386 |
| ボタングループ | 240 |
| ボタンツールバー | 240 |

■ マ行

| | |
|----------------|---------|
| マージン | 450 |
| マウスオーバー表示のテーブル | 80,407 |
| 前に戻る | 264 |
| マルチクラス | 435 |
| ミックスイン | 456 |
| 未読メッセージ | 116 |
| 無効なテキスト | 55 |
| メディアプロジェクト | 143 |
| メディアクエリ | 6,455 |
| モーダル | 283,386 |
| モーダルウインドウ | 283 |
| 文字参照 | 64 |
| モバイルファースト | 30,311 |
| モバイルフレンドリー | 8 |

■ ヤ行・ラ行・ワ行

| | |
|------------------|-------------|
| ユーザーインプット | 66 |
| ユーティリティクラス | 35 |
| ライブラリ | 2 |
| ラジオボタン | 200 |
| ラベル付けされたテキスト | 56 |
| リード | 52 |
| リキッドレイアウト | 5 |
| リスト | 61 |
| リストグループ | 180,293,296 |
| リストマーカー | 61 |
| リセット CSS | 96 |
| リセットスタイル | 96 |
| 略称 | 57 |
| ルート要素 | 97 |
| レスポンシブ Web デザイン | 4,35 |
| レスポンシブ画像 | 69 |
| レスポンシブ対応のテーブル | 88 |
| レスポンシブ対応のナビゲーション | 156 |
| レンジ入力 | 199 |
| ワイドフレーム | 361 |

■著者略歴

WINGS プロジェクト (<https://wings.msn.to/>)

有限会社 WINGS プロジェクトが運営する、テクニカル執筆コミュニティ（代表：山田祥寛）。主に Web 開発分野の書籍／記事執筆、翻訳、講演などを幅広く手がける。2018 年 5 月時点での登録メンバーは約 55 名で、現在も執筆メンバーを募集中。興味のある方は、どしどし応募頂きたい。著書、記事多数。

RSS : <https://wings.msn.to/contents/rss.php> / Facebook : facebook.com/WINGSProject / Twitter : @yyamada (公式)

宮本 麻矢（みやもと まや）

第 1 章、第 2 章、第 7 章、第 8 章、第 10 章担当。

WINGS プロジェクト所属のフリーライター。専門学校在学中、Web デザインコンペで入賞したことをきっかけに、Web デザインの世界へ。卒業後、文具メーカーにて Web 開発を担当、2013 年退職。現在は Web サイトの構築やコンサルティング業務を行うかたわら、執筆活動をしている他、職業訓練校やスクールにて Web や DTP に関するトレーニングを行っている。

朝平 文彦（あさひら ふみひこ）

第 3 章、第 4 章、第 5 章、第 6 章、第 9 章担当。

神戸芸術工科大学大学院芸術工学研究科（総合デザイン専攻）修了。

建築デザイン、都市計画コンサルタントを経て、現在は Web やグラフィックなどのヴィジュアルコミュニケーション分野のコンサルティング・制作業務を行っている。また、大学や職業訓練校の講師としてクリエイター育成にも尽力している。

■監修者紹介

山田 祥寛（やまだ よしひろ）

千葉県鎌ヶ谷市在住のフリーライター。Microsoft MVP for Visual Studio and Development Technologies。執筆コミュニティ「WINGS プロジェクト」の代表でもある。

主な著書に『改訂新版 JavaScript 本格入門』『Angular アプリケーションプログラミング』『Ruby on Rails 5 アプリケーションプログラミング』（以上、技術評論社）、『独習シリーズ（C#・サーバサイド Java・PHP・ASP.NET）』（翔泳社）、『はじめての Android アプリ開発 第 2 版』（秀和システム）、『書き込み式 SQL のドリル 改訂新版』（日経 BP 社）など。

カバーデザイン ◆ 菊池祐（株式会社ライラック）

本文デザイン ◆ 株式会社トップスタジオ

本文レイアウト ◆ 株式会社トップスタジオ

編集担当 ◆ 青木宏治

ブートストラップ **Bootstrap 4**

かいはつ きょうかしょ フロントエンド開発の教科書

2018年9月8日 初版 第1刷発行

2020年4月29日 初版 第3刷発行

著者 ウィングス WINGSプロジェクト 宮本 麻矢、

朝平 文彦

監修者 山田 祥寛

発行者 片岡 巍

発行所 株式会社技術評論社

東京都新宿区市谷左内町 21-13

電話 03-3513-6150 販売促進部

03-3513-6160 書籍編集部

印刷所 港北出版印刷株式会社

定価はカバーに表示してあります

本書の一部または全部を著作権法の定める範囲を越え、無断で複写、複製、転載、テープ化、ファイルに落とすことを禁じます。

© 2018 WINGS プロジェクト

造本には細心の注意を払っておりますが、万一、乱丁（ページの乱れ）や落丁（ページの抜け）がございましたら、小社販売促進部までお送りください。送料小社負担にてお取り替えいたします。

ISBN978-4-297-10020-9 C3055

Printed in Japan

■ご質問について

本書の内容に関するご質問は、下記の宛先までFAXか書面、もしくは弊社Webサイトの電子メールでお送りください。お電話によるご質問、および本書に記載されている内容以外のご質問には、いっさいお答えできません。あらかじめご了承ください。

宛先：〒162-0846

東京都新宿区市谷左内町 21-13

株式会社技術評論社 書籍編集部

『Bootstrap 4 フロントエンド開発の教科書』係

FAX : 03-3513-6167

Web : <https://gihyo.jp/book>

※ご質問の際に記載いただきました個人情報は、ご質問の返答以外での目的には使用いたしません。参照後は速やかに削除させていただきます。