

ウェブ開発の知識とコツをしっかりマスター

Bootstrap 4

フロントエンド開発の教科書

WINGSプロジェクト

宮本麻矢、朝平文彦【著】

山田祥寛【監修】



CSSフレームワークやグリッドデザインの基本から
独自テーマによるカスタマイズ方法まで

Bootstrap 4の
全機能がこの1冊でわかる！

一步先を目指す
Webデザイナー&
Webエンジニアに
おすすめ！

技術評論社



はじめに

スマートフォンやタブレットの普及に伴い、Web ページをさまざまなデバイスに合わせて表示する「レスポンシブ Web デザイン」は、今や必須の技術となりました。従来の Web 制作のフローでは、各デバイス向けに行うデザインや開発の工数が増大するため、制作サイドにとって膨大な手間が発生してしまいます。また、発注者サイドにとっても制作コストが大きく膨らむことになってしましました。

そんな中、Web 制作を生業とする私たちの救いとなったのが、あらかじめ用意された簡単なクラスを HTML に追加するだけで、レスポンシブな Web ページをあっという間に作ることができる「Web アプリケーションフレームワーク」でした。

本書で扱う「Bootstrap」は、数あるフレームワークの中でも、もっとも人気のあるツールの 1 つです。本書では、HTML、CSS などの言語についてひととおり理解している方を対象として、「デザインは得意ではない」というエンジニアの方や、「コーディングが苦手」というデザイナーの方にも、従来のフローに代わる新しい制作フロー「インブラウザデザイン」（ブラウザ上で直接デザインする手法）を実践することができるようになることを目指しています。

「Bootstrap」を使用するためには、まず覚えなければならないルールが少なからずあります。私たちは、特にこの最初のハードルを下げるために可能な限り丁寧な解説を心がけました。本書が Web 制作に携わる多くの皆様の助けになることを心から祈っております。

なお、本書に関するサポートサイトを以下の URL で公開しています。Q & A 掲示板はじめ、サンプルのダウンロードサービス、本書に関する FAQ 情報、オンライン公開記事などの情報を掲載していますので、あわせてご利用ください。

<https://wings.msn.to/>

最後にはなりましたが、叱咤激励しながら無理を調整いただいた監修の山田祥寛さん、奈美さん、技術評論社、関係者ご一同に心から感謝いたします。

2018 年 7 月吉日 宮本麻矢 朝平文彦

9

SECTION

9

コンテンツ 04 (Coupon) の作成

カフェのクーポンチケットを紹介するコンテンツ 04 「Coupon」を作成します。

このコンテンツには、このコンテンツでは、**カード** (P.124 参照) を使用したクーポンチケットを表示します。

9.9.1 コンテンツ 04 のレイアウト

まず、ワイヤーフレームでコンテンツ 04 のレイアウトを再確認しておきましょう。画面幅 md 以上（デスクトップ PC での閲覧時）と画面幅 sm 以下（モバイル端末での閲覧時）とで、特にレイアウトの変更はありません（図 9-38）。

▼図 9-38 コンテンツ 04 のレイアウト



9.9.2 コンテンツ 04 の構成

ではコンテンツ 04 を作成していきましょう。このコンテンツでは、カードの基本的な使い方や、配色や文字のユーティリティ（P.301 参照）を活用した外観の調整方法を確認できます。Bootstrap で定義されているたくさんのユーティリティの活用は、**インプラウザデザイン**^{*1} を行う上でとても有効です（リスト 9-20）。

▼リスト 9-20 コンテンツ 04 の作成 (contents-04.html)

```
</-- コンテンツ04 -->
<div class="py-4 bg-light"> _____
  <section id="coupon"> _____
    <div class="container"> _____
      <h3 class="text-center mb-3">Coupon</h3> _____
    </div> _____
  </section> _____
</div> _____
```

404

*1 ワイヤーフレームと記述コンテンツをもとに、コーディングしながらブラウザ上で直接デザインする手法です。

コードリスト

HTMLやCSS、JavaScript のソースコードを表します。紙面では、Bootstrap の使用方法を理解するためのコードを抜粋しているので、全体を確認するにはダウンロードサンプルから対応するファイルを確認してください。

脚注

本文で説明しきれなかった補足情報を表します。本文中の番号と対応していますので、あわせて確認してください。

NOTE

本文の説明に加えて知つておきたい注意点や追加情報を表します。

コラム

本文中で扱っているテーマと直接の関係はありませんが、関連している情報として知つておきたい参考情報を表します。

NOTE フレームワーク導入のデメリット

ここまでフレームワーク導入の利点を挙げましたが、デメリットもあります。フレームワークはとても便利ですが、フレームワーク特有のルールを見る必要があります。それなりに学習コストがかかります。またルールに沿って書くことになるため、自由にコーディングできるわけではありません。個人で開発しているような小規模なプロジェクトでは、フレームワークの制約に縛られず自由に設計する方が、総合的に良いケースもあります。学習コストが開発コストを上回るようなら、フレームワーク導入にこだわる必要はありません。

COLUMN 入力グループのアクセシビリティ

入力グループについては、スクリーンリーダーなどの支援技術に対して、その機能が伝えられているかどうかをよく確認する必要があります。たとえば、label 要素を使用してラベルを付ける (`sr-only` クラスを使用して非表示にする)、あるいは `aria-describedby`、`aria-label`、`aria-labelledby` 属性などを使用する方法があります。本書ではなるべく例を示すようにしていますが、どのような追加情報が必要かは、実装しているインターフェイスによって異なります。コードをそのまま使うのではなく内容を理解してアクセシビリティのためのコードを書くようにしましょう。



CONTENTS

はじめに	iii
本書の読み方	iv

第1章 イントロダクション 1

1.1 Web アプリケーションフレームワークの基本	2
1.1.1 Web アプリケーションフレームワークとは	2
1.1.2 フレームワーク導入の利点	3
1.2 Bootstrap の特徴	4
1.2.1 レスポンシブ Web デザインに対応	4
1.2.2 グリッドシステムによるレイアウト	9
1.2.3 デザイン性に優れたコンポーネント	9
1.3 Bootstrap の歴史	11
1.3.1 Bootstrap の誕生と歩み	11
1.3.2 バージョンによる違い	12
1.4 Bootstrap の導入	14
1.4.1 Bootstrap の導入に必要な環境	14
1.4.2 ダウンロードして利用する	14
1.4.3 HTML の雑形	17
1.4.4 CDN を利用する	19

第2章 Bootstrap のレイアウト 21

2.1 Bootstrap のグリッドシステム	22
2.1.1 カラム (column : 列) とガター (gutter : 溝)	22
2.1.2 コンテナ (container : 箱)	23
2.1.3 row クラス (行)	24
2.1.4 グリッドシステムの使い方	25
2.2 列の自動レイアウト	26
2.2.1 等幅カラム	26
2.2.2 指定幅カラム	26
2.2.3 1 カラムのみ幅を設定 (残りのカラムは自動的に等幅)	28
2.3 レスポンシブなグリッドシステム	29
2.3.1 5 段階のレイアウト制御	29
2.3.2 ブレイクポイントによる切り替え	30
2.3.3 可変幅カラム	31
2.3.4 等幅カラムを複数行に分割	32
2.3.5 行の分割をブレイクポイントで切り替え	33
2.3.6 複数クラスの組み合わせ	34
2.4 カラムの整列	36
2.4.1 行単位での垂直方向の整列	36
2.4.2 カラム単位での垂直方向の整列	37
2.4.3 水平方向の整列	38
2.4.4 ガターの削除	39
2.4.5 カラムの折り返し	41
2.5 カラムの並べ替え	42
2.5.1 order-* クラスで並べ替え	42
2.5.2 カラムのオフセット	43
2.5.3 グリッドレイアウトの入れ子 (ネスト)	45
2.6 レイアウトのためのユーティリティ	46
2.6.1 Display ユーティリティ	46
2.6.2 Visibility ユーティリティ	46
2.6.3 Flex ユーティリティ	46
2.6.4 Spacing ユーティリティ	46

3.1	タイポグラフィ	48
3.1.1	見出し	48
3.1.2	見出しに副見出しを付ける	50
3.1.3	見出しを目立たせる	50
3.1.4	リード	52
3.1.5	インラインテキスト要素	53
3.1.6	Text ユーティリティ	57
3.1.7	略語	57
3.1.8	引用文	58
3.1.9	引用元の表示	59
3.1.10	引用文の位置合わせ	60
3.1.11	リスト	61
3.1.12	インライнст	62
3.1.13	定義リスト	63
3.2	コード	64
3.2.1	インラインのコード表記	64
3.2.2	コードブロックの表記	65
3.2.3	変数の表記	66
3.2.4	ユーザーインプットの表記	66
3.2.5	サンプル出力	67
3.3	画像	69
3.3.1	レスポンシブ画像	69
3.3.2	サムネイル画像	70
3.3.3	画像の位置合わせ	71
3.4	テーブル	73
3.4.1	テーブルの基本スタイリング	73
3.4.2	暗色テーブル	75
3.4.3	テーブルヘッドのオプション	76
3.4.4	縞模様のテーブル	77
3.4.5	罫線付きのテーブル	78
3.4.6	罫線なしのテーブル 4.1	79
3.4.7	テーブル行のマウスオーバー表示	80
3.4.8	テーブルのコンパクト化	81
3.4.9	テーブル行・セルの色付け	82
3.4.10	キャプション	87
3.4.11	レスポンシブ対応のテーブル	88
3.5	図表	93
3.5.1	図表の基本的なスタイリング	93
3.5.2	図表キャプションの位置合わせ	94
3.6	Reboot による初期設定	96
3.6.1	Reboot によって初期化されているスタイル	97
3.6.2	全要素へのリブート設定	97
3.6.3	body 要素へのリブート設定	98
3.6.4	見出しと段落へのリブート設定	98
3.6.5	リストへのリブート設定	99
3.6.6	整形済みのテキストへのリブート設定	100
3.6.7	テーブルへのリブート設定	100
3.6.8	フォームへのリブート設定	101
3.6.9	リンクへのリブート設定	103
3.6.10	その他要素へのリブート設定	104
3.6.11	HTML5 の hidden 属性	105

4.1	ジャンボトロン	108
4.1.1	基本的な使用例	108
4.1.2	ジャンボトロンを全幅で表示する	109
4.2	アラート	110
4.2.1	基本的な使用例	110

4.2.2	リンクの色	111
4.2.3	アラート内にコンテンツを追加する.....	113
4.2.4	アラートを閉じる.....	113
4.3	バッジ.....	115
4.3.1	基本的な使用例.....	115
4.3.2	カウンターを作成する.....	116
4.3.3	ビル型のバッジを作成する.....	117
4.3.4	アクション付きのバッジを作成する.....	117
4.4	プログレス.....	118
4.4.1	基本的な使用例.....	118
4.4.2	プログレスバーにテキストラベルを追加する.....	119
4.4.3	プログレスバーの高さを変更する.....	119
4.4.4	プログレスバーの背景を変更する.....	120
4.4.5	複数のプログレスバーを重ねて表示する.....	121
4.4.6	プログレスバーをストライプにする.....	122
4.4.7	プログレスバーのストライプをアニメーションにする.....	122
4.5	カード.....	124
4.5.1	基本的な使用例.....	124
4.5.2	カードのスタイルを変更する.....	129
4.5.3	カードのサイズを変更する.....	132
4.5.4	カードのテキストを整列する.....	134
4.5.5	カードにナビゲーションを組み込む.....	135
4.5.6	カードの画像とテキストを重ね合わせる.....	137
4.5.7	カードをレイアウトする.....	137
4.6	メディアオブジェクト.....	143
4.6.1	基本的な使用例.....	143
4.6.2	メディアオブジェクトの入れ子.....	144
4.6.3	メディアの位置合わせ.....	145
4.6.4	メディアオブジェクトの並べ替え.....	146
4.6.5	メディアオブジェクトをリストに組み込む.....	147

第 5 章 ナビゲーションのコンポーネント

149

5.1	ナビゲーション	150
5.1.1	基本的な使用例.....	150
5.1.2	ナビゲーションに使用できるスタイル.....	151
5.1.3	レスポンシブ対応のナビゲーション.....	156
5.1.4	ドロップダウンナビゲーション.....	157
5.1.5	ナビゲーションの JavaScript 使用例.....	159
5.2	ナビゲーションバー	161
5.2.1	外枠の作成	161
5.2.2	サブコンポーネントの作成	162
5.2.3	ナビゲーションバーの配色	169
5.2.4	ナビゲーションバーの幅の設定	170
5.2.5	ナビゲーションバーの配置	171
5.2.6	レスポンシブ対応の設定	175
5.3	パンくずリスト	179
5.3.1	基本的な使用例	179
5.4	リストグループ	180
5.4.1	基本的な使用例	180
5.4.2	リスト項目をアクティブ状態にする	180
5.4.3	リスト項目を無効状態にする	181
5.4.4	リンク付きリストグループ	181
5.4.5	ボタンのリストグループ	182
5.4.6	リストグループの背景色を変更する	184
5.4.7	リンク付きリストグループの背景色を変更する	184
5.4.8	バッジ付きリストグループ	186
5.4.9	カスタムコンテンツのリストグループ	187
5.4.10	棒なしのリストグループ 4.1	188
5.5	ページネーション	189

5.5.1	基本的な使用例.....	189
5.5.2	ページネーションにアイコンを使用する.....	190
5.5.3	リンクに無効状態や現在位置であることを示す.....	190
5.5.4	ページネーションのサイズを変更する.....	193
5.5.5	ページネーションの配置.....	193

第6章 フォームとボタンのコンポーネント

195

6.1	フォーム	196
6.1.1	基本的な使用例.....	196
6.1.2	チェックボックスとラジオボタン.....	200
6.1.3	レイアウトを調整する.....	203
6.1.4	ヘルプテキストを表示する.....	209
6.1.5	一連のフォームグループをまとめて無効にする.....	209
6.1.6	フォームの入力検証機能を使う.....	210
6.1.7	Bootstrap 独自にスタイル設定されたフォームを使用する.....	216
6.2	入力グループ	223
6.2.1	基本的な使用例.....	223
6.2.2	入力グループのサイズ調整.....	226
6.2.3	チェックボックスやラジオボタンのアドオン	227
6.2.4	複数の入力コントロール.....	227
6.2.5	複数のアドオンを組み合わせる.....	228
6.2.6	ボタン付きアドオン	229
6.2.7	ドロップダウン付きアドオン	230
6.2.8	スプリットボタンのアドオン	230
6.2.9	カスタムフォームの組み込み	231
6.3	ボタン	233
6.3.1	基本的な使用例.....	233
6.3.2	アウトラインボタンを作成する.....	234
6.3.3	ボタンサイズを変更する.....	235
6.3.4	ブロックレベルのボタンを作成する.....	235
6.3.5	アクティブ状態のボタンを作成する.....	236
6.3.6	無効状態のボタンを作成する.....	236
6.3.7	切り替えボタンを作成する.....	237
6.3.8	チェックボックスとラジオボタンを作成する.....	238
6.3.9	メソッド	239
6.4	ボタングループ	240
6.4.1	基本的な使用例.....	240
6.4.2	ボタンツールバーを作成する.....	240
6.4.3	サイズを変更する.....	242
6.4.4	ドロップダウンメニューを入れ子にする.....	242
6.4.5	垂直方向のボタングループを作成する.....	243
6.5	ドロップダウン	244
6.5.1	基本的な使用例.....	244
6.5.2	ドロップダウン方向を変更する.....	247
6.5.3	メニュー項目のリンクに使用できる要素	248
6.5.4	メニューの位置揃えを変更する.....	249
6.5.5	ドロップダウンメニューにさまざまな要素を組み込む	250
6.5.6	ドロップダウンメニューに自由形式のテキストを配置する 4.1	251
6.5.7	ドロップダウンのメニュー項目に無効やアクティブの状態を設定する	251
6.5.8	ドロップダウンにリンクなしのメニュー項目を追加する 4.1	252
6.6	ドロップダウンの JavaScript 使用	253
6.6.1	ドロップダウンのオプション	253
6.6.2	ドロップダウンのメソッド	255
6.6.3	ドロップダウンのイベント	256

第7章 JavaScript を利用したコンポーネント

259

7.1	Bootstrap の JavaScript プラグイン	260
7.1.1	Bootstrap の JavaScript ファイル	260
7.1.2	Bootstrap のデータ属性 API	261

7.2	カルーセル.....	262
7.2.1	基本的な使用例.....	262
7.2.2	コントローラーを表示させる.....	263
7.2.3	インジケーターを表示させる.....	265
7.2.4	スライドのキャプションを表示させる.....	265
7.2.5	フェードで遷移させる 4.1	266
7.3	カルーセルの JavaScript 使用.....	268
7.3.1	カルーセルのメソッド.....	269
7.3.2	カルーセルのイベント.....	270
7.4	折り畳み.....	272
7.4.1	基本的な使用例.....	272
7.4.2	複数の要素の表示と非表示とを切り替える.....	274
7.4.3	アコーディオンを作成する.....	275
7.5	折り畳みの JavaScript 使用.....	279
7.5.1	折り畳みのメソッド.....	281
7.5.2	折り畳みのイベント.....	281
7.6	モーダル.....	283
7.6.1	基本的な使用例 4.1	283
7.6.2	サイズのオプション 4.1	289
7.7	モーダルの JavaScript 使用.....	291
7.7.1	モーダルのメソッド.....	291
7.7.2	モーダルのイベント.....	292
7.8	スクロールスパイ.....	293
7.8.1	基本的な使用例.....	293
7.8.2	body 要素以外の要素での使用例.....	296
7.9	スクロールスパイの JavaScript 使用.....	299
7.9.1	スクロールスパイのメソッド.....	299
7.9.2	スクロールスパイのイベント.....	299

第 8 章 ユーティリティ

301

8.1	Color ユーティリティ	302
8.1.1	文字色を設定するクラス	302
8.1.2	背景色を設定するクラス	304
8.2	Border ユーティリティ	306
8.2.1	ボーダーを追加するクラス	306
8.2.2	ボーダーを削除するクラス	307
8.2.3	ボーダー色を設定するクラス	307
8.2.4	角丸を設定するクラス	308
8.3	Display ユーティリティ	310
8.3.1	表示形式を設定するクラス	310
8.3.2	要素の表示／非表示を設定するレスポンシブなクラス	311
8.3.3	印刷時の表示／非表示を設定するクラス	312
8.4	Sizing ユーティリティ	314
8.4.1	幅を設定するクラス	314
8.4.2	高さを設定するクラス	315
8.4.3	最大幅 100% を設定するクラス	316
8.4.4	最大高 100% を設定するクラス	317
8.5	Spacing ユーティリティ	318
8.5.1	Spacing ユーティリティの記法	318
8.5.2	自動マージンの応用	321
8.6	Flex ユーティリティ	322
8.6.1	flexbox を有効にするクラス	322
8.6.2	flex コンテナの主軸方向を設定するクラス	323
8.6.3	主軸方向の整列をするクラス	325
8.6.4	交差軸方向の整列をするクラス	327
8.6.5	flex アイテムを交差軸上で個別に整列するクラス	329
8.6.6	flex コンテナ全幅に渡って等幅で整列するクラス	331

8.6.7	flex アイテムの幅の伸縮を指定するクラス	331
8.6.8	flex アイテムの折り返しを設定するクラス	333
8.6.9	特定の flex アイテムの表示順序を入れ替えるクラス	334
8.6.10	自動マージンで flex アイテムを分離する	334
8.6.11	複数行における flex アイテムの交差軸の整列をするクラス	335
8.7	Float ユーティリティ	338
8.7.1	フロートを設定するクラス	338
8.7.2	ブレイクポイントでフロートを切り替えるクラス	338
8.7.3	clearfix ユーティリティ	339
8.8	Position ユーティリティ	342
8.8.1	要素の位置指定をするクラス	342
8.8.2	最上部に固定するクラス	343
8.8.3	最下部に固定するクラス	344
8.8.4	最上部に達すると固定するクラス	346
8.9	Text ユーティリティ	347
8.9.1	文字の均等割り付けを設定するクラス	347
8.9.2	文字の左寄せ／右寄せ／中央揃えを設定するクラス	347
8.9.3	文字を折り返さないよう設定するクラス	348
8.9.4	長いテキストを省略記号で表すクラス	349
8.9.5	文字を大文字や小文字に変換するクラス	349
8.9.6	文字の太さとイタリック体を設定するクラス	350
8.9.7	等幅フォントを指定するクラス 4.1	351
8.10	Vertical align ユーティリティ	352
8.10.1	インライン要素の垂直方向の整列	352
8.10.2	テーブルセルの垂直方向の整列	353
8.11	その他のユーティリティクラス	354
8.11.1	スクリーンリーダー用ユーティリティ	354
8.11.2	Visibility ユーティリティ	354
8.11.3	クローズアイコンユーティリティ	355
8.11.4	Embed ユーティリティ	356
8.11.5	Shadows ユーティリティ 4.10	357

第 9 章 Bootstrap でモックアップを作る

359

9.1	サイト概要とファイルの準備	360
9.1.1	サイト概要	360
9.1.2	ワイヤーフレームの確認	361
9.1.3	使用する主なコンポーネント	365
9.2	新規ファイル作成	368
9.2.1	head 要素の修正	368
9.2.2	基本構造の入力	369
9.3	ヘッダーの作成	371
9.4	ナビゲーションバーの作成	372
9.4.1	ナビゲーションバーのレイアウト	372
9.4.2	ナビゲーションバーの基本構成	372
9.4.3	サブコンポーネントの組み込み	373
9.4.4	ナビゲーションバーの完成図	376
9.5	メインビジュアルの作成	377
9.5.1	メインビジュアルのレイアウト	377
9.5.2	メインビジュアルの基本構成	378
9.5.3	インジケーターの組み込み	379
9.5.4	各スライドの組み込み	379
9.5.5	各コントローラーの組み込み	381
9.5.6	メインビジュアルの完成図	381
9.6	コンテンツ 01 (News) の作成	383
9.6.1	コンテンツ 01 のレイアウト	383
9.6.2	コンテンツ 01 の構成	383
9.6.3	コンテンツ 01 の完成図	384
9.7	コンテンツ 02 (About) の作成	386

9.7.1	コンテンツ 02 のレイアウト	386
9.7.2	コンテンツ 02 の構成	388
9.7.3	コンテンツ 02 の完成図	394
9.8	コンテンツ 03 (Menu) の作成	396
9.8.1	コンテンツ 03 のレイアウト	396
9.8.2	コンテンツ 03 の構成	397
9.8.3	コンテンツ 03 の完成図	402
9.9	コンテンツ 04 (Coupon) の作成	404
9.9.1	コンテンツ 04 のレイアウト	404
9.9.2	コンテンツ 04 の構成	404
9.9.3	コンテンツ 04 の完成図	406
9.10	コンテンツ 05 (Information) の作成	407
9.10.1	コンテンツ 05 のレイアウト	407
9.10.2	コンテンツ 05 の構成	408
9.10.3	左側セクションにテーブルを作成	410
9.10.4	右側セクションに Google マップを埋め込み	411
9.10.5	コンテンツ 05 の完成図	414
9.11	フッターの作成	416
9.11.1	フッターのレイアウト	416
9.11.2	フッターの構成	416
9.11.3	フッターの完成図	417
9.12	リンクの設定と追加 CSS の作成	419
9.12.1	ナビゲーションバーのリンク	419
9.12.2	コンテンツ 02 のリンク	420
9.12.3	フッターのリンク	421
9.12.4	ページ内リンクの位置調整	421
9.13	下層ページ (Contact) の作成	423
9.13.1	ファイルの準備	423
9.13.2	下層ページのレイアウト	426
9.13.3	下層ページの構成	427
9.13.4	パンくずリストの作成	428
9.13.5	フォームの作成	428
9.13.6	下層ページの完成図	431

第 10 章 Bootstrap のカスタマイズ

433

10.1	Bootstrap のオリジナルスタイルを上書きする	434
10.1.1	カスタマイズ用 CSS を参照する	434
10.1.2	Bootstrap の CSS 設計の方針	434
10.1.3	クラス名の付け方のポイント	435
10.1.4	スタイルを上書きする際の注意点	436
10.2	Sass を使ってカスタマイズする	439
10.2.1	Sass の利用環境を整える	439
10.2.2	SCSS ファイルの準備	443
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	446
10.2.4	Sass 変数を上書きする	447
10.2.5	基本の配色を変更する	448
10.2.6	body の背景色、文字色、リンク色を変更する	449
10.2.7	Spacing ユーティリティを変更する	450
10.2.8	Sizing ユーティリティを変更する	451
10.2.9	ブレイクポイントを変更する	452
10.2.10	コンテナを変更する	452
10.2.11	グリッドのカラム数やガター幅を変更する	453
10.2.12	書式を変更する	453
10.3	CSS 変数を利用する	454
10.3.1	Bootstrap で定義されている CSS 変数	454
10.3.2	ミックスインを利用する	456
	索引	460

第 1 章

イントロダクション

本章では、Bootstrap を学ぶ上で知っておきたい基礎知識について解説します。そもそも Web アプリケーションフレームワークとは何なのか、その概念について触れ、Bootstrap の特徴やそれを支える技術、Bootstrap がここまで発展した背景、バージョンによる違いなど、Bootstrap の概要について説明します。本章の最後には、Bootstrap の導入方法として、ダウンロードして利用する方法と CDN で利用する方法を説明します。

1

SECTION

1

Web アプリケーション フレームワークの基本

本書のテーマであるBootstrapは、HTML、CSS、Javascriptで構成されるフロントエンドのWebアプリケーションフレームワークです。この「フレームワーク」とは、そもそもどういったものなのでしょうか？まずはフレームワークについて簡単に触れておきましょう。

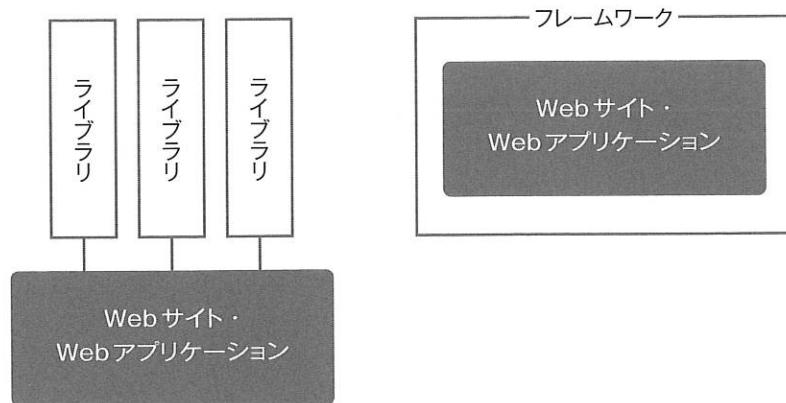
1.1.1

Web アプリケーションフレームワークとは

フレームワークとは、枠組み、骨組み、骨格などを意味する言葉です。WebサイトやWebアプリケーションを構築する上で、その骨組み（土台）を提供したり、よく使われる汎用的な機能をまとめて提供したりするものを**Web アプリケーションフレームワーク**と言います。

フレームワークと似たものに**ライブラリ**があります。ライブラリは、よく使われる機能を**部品化**し、それらを集めパッケージ化したものです。ライブラリが単にコードの再利用を目的としているのに対し、フレームワークは設計レベルの再利用を目的としています。ライブラリは**部品**の集まりですから、部品をどのように組み合わせ、最終的にどのようなWebサイト・Webアプリケーションにするのかは、制作者に委ねられています。一方のフレームワークでは、全体の骨組みが**テンプレート**として用意されており、制作者はテンプレートに肉付けの作業をしていきます。既にWebサイト・Webアプリケーションの土台ができあがっているので、あとは枠にはめていくだけです。部分的に機能を提供してくれるのがライブラリ、全体的に面倒を見てくれるのがフレームワークと捉えればわかりやすいかもしれません（図1-1）。

▼図1-1 ライブラリとフレームワーク



1.1.2 フレームワーク導入の利点

フレームワークを導入することには、次のようなメリットがあります。

① 速く開発できる

フレームワークの最大のメリットは、生産性の向上です。基盤となる枠組みができているので、ゼロからコードを書く必要がなく、開発スピードが格段にアップします。

② コードの書き方を統一できる

コードの書き方には、個人のクセが出てしまいがちです。複数人のチームで開発する場合、コードに一貫性がなく、共同作業がやっかいになるケースがあります。フレームワークを利用すると、自分ルールではなく、フレームワークのルールでコードを書く必要があるため、誰が書いてもコードに一貫性を持たせることができます。

③ メンテナンスしやすい

コードに一貫性があるということは、メンテナンスしやすいということでもあります。他人の書いたコードを解読するのは面倒な物ですが、フレームワーク上のルールでコーディングしたものは、該当箇所を特定しやすくなります。

④ 技術トレンドに対応しやすい

新しい技術が目まぐるしく出てくる中、開発者がそれをキャッチアップしていくのは大変な労力を伴います。しかし、フレームワークにはそうした技術的なトレンドが積極的に取り入れられています。フレームワークを利用することで、トレンドにも臨機応変に対応することができます。

NOTE

フレームワーク導入のデメリット

ここまでフレームワーク導入の利点を挙げましたが、デメリットもあります。フレームワークはとても便利ですが、フレームワーク特有のルールを覚える必要があります、それなりに学習コストがかかります。またルールに沿って書くことになるため、自由にコーディングできるわけではありません。個人で開発しているような小規模なプロジェクトでは、フレームワークの制約に縛られず自由に設計する方が、総合的に良いケースもあります。学習コストが開発コストを上回るようなら、フレームワーク導入にこだわる必要はありません。

1

SECTION

2

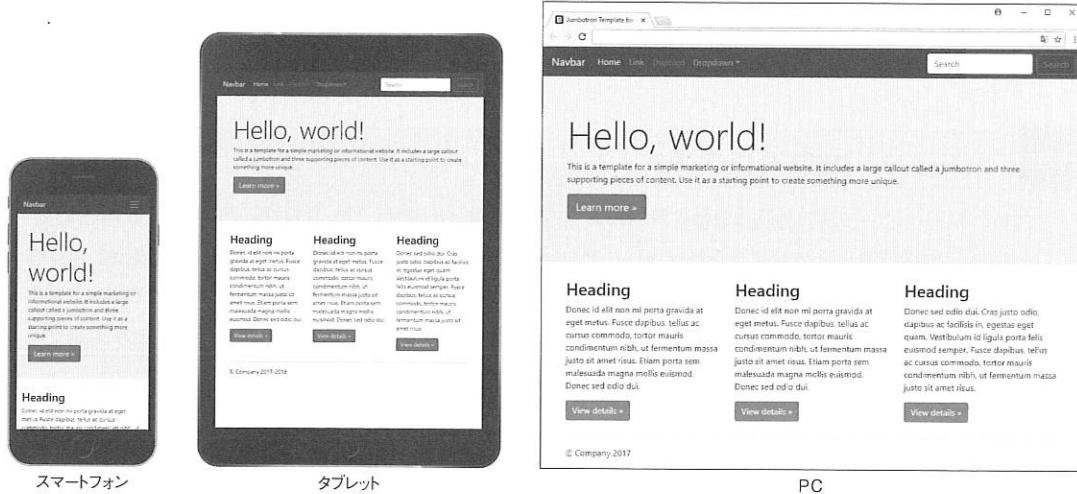
Bootstrapの特徴

本書で扱うBootstrapは、数あるフレームワークの中でも、もっとも人気のあるWebアプリケーションフレームワークの1つです。世界中のプロジェクトで利用されており、実績も確かです。なぜ世界中で支持されているのか、その特徴を見ていきましょう。

1.2.1 レスポンシブ Web デザインに対応

現在、Web制作において、マルチデバイス対応の手法の主流となっているのが、**レスポンシブ Web デザイン (Responsive Web Design)** です。レスポンシブ Web デザインは、アメリカの Web デザイナー、イーサン・マルコッテ (Ethan Marcotte) 氏が提唱したマルチデバイス対応 Web ページの制作手法です。簡単に言うと、1つの Web ページ (HTML ファイル) で、スマートフォン、タブレット、PC などあらゆるデバイスに合わせた最適なレイアウト表示をする手法です (図 1-2)。

▼図 1-2 レスポンシブ Web デザインの例

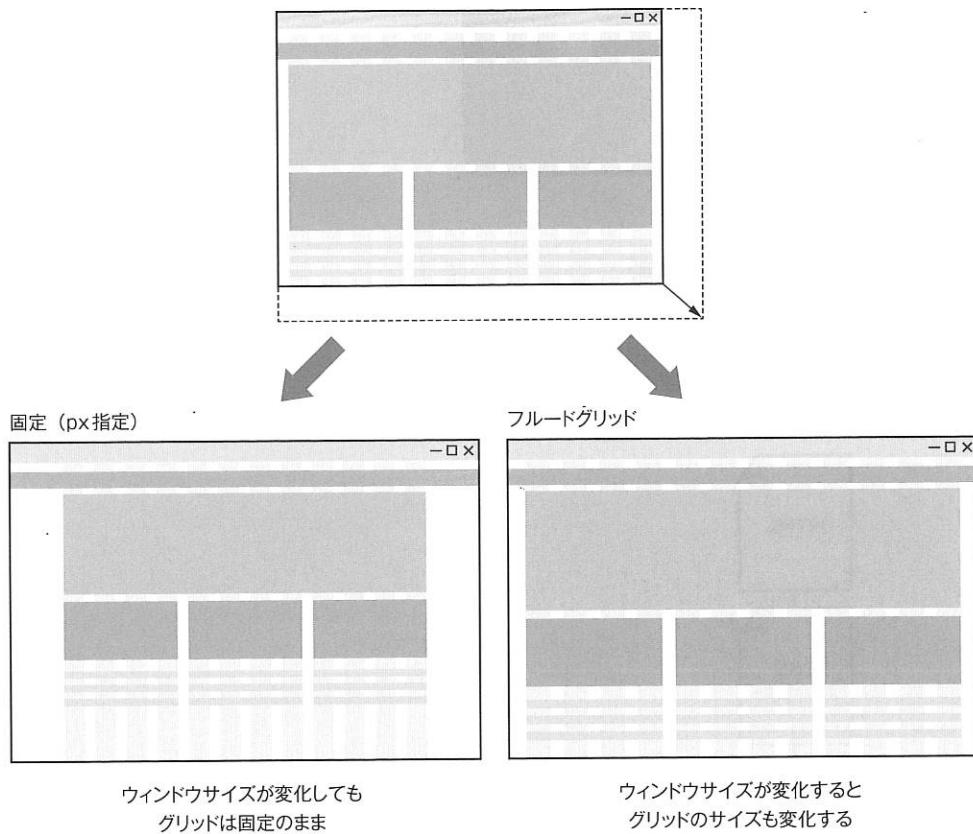


レスポンシブ Web デザインは、**フルードグリッド**、**フルードイメージ**、**メディアクエリ**などの技術で構成されています。

1 フルードグリッド (Fluid Grid : 可変グリッド)

グリッドシステムによる Web デザインと、ブラウザ幅の変化に応じて内容の大きさを変えるリキッドレイアウトを組み合わせた手法を**フルードグリッド**と言います。ブラウザの表示領域に応じてグリッドの横幅が変化するように、Web ページの横幅を 100 %として、グリッドのサイズ相対値（%、em、rem など）で指定します（図 1-3）。

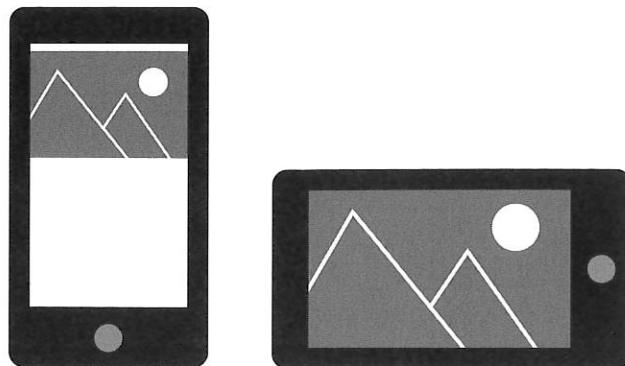
▼図 1-3 固定のグリッド（左）とフルードグリッド（右）



2 フルードイメージ (Fluid Image : サイズが変化する画像メディア)

レイアウト幅に応じて、画像サイズを拡大縮小させるテクニックを**フルードイメージ**と言います。フルードグリッドと組み合わせることで、画像サイズをグリッド幅に応じて変化させることができます。CSS のみで実装できます（図 1-4）。

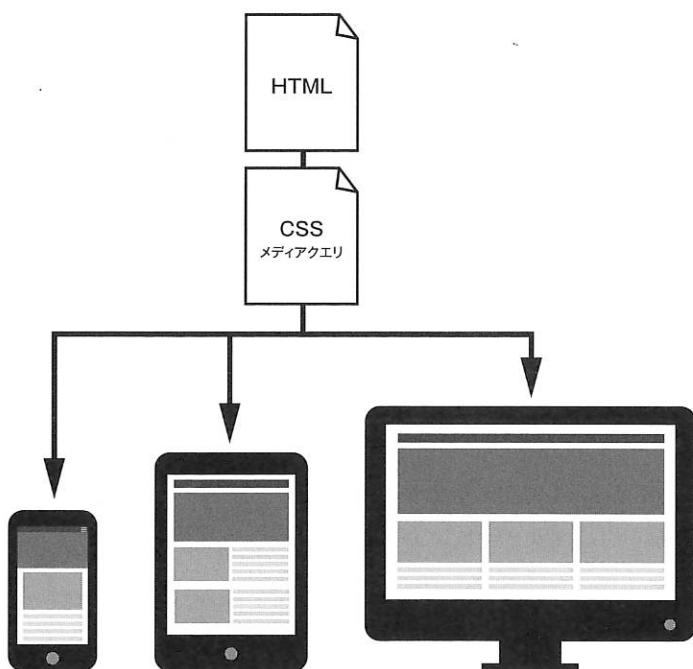
▼図 1-4 フルードイメージ：画面サイズに応じて画像サイズが変化する



3 メディアクエリ (Media Queries)

メディアクエリは、画面幅、画像解像度、デバイスの向きなどの条件に合わせて、スタイルを切り替えることができるCSS3から導入された技術です（図1-5）。

▼図 1-5 メディアクエリの切り替えイメージ



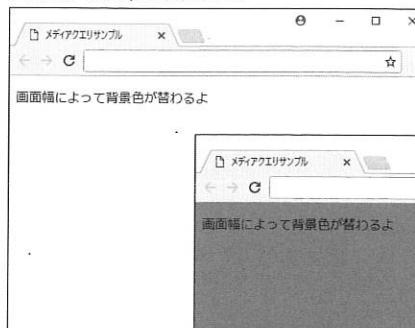
メディアクエリには、いくつかの書き方がありますが、ここでは簡単な例を見てみましょう。次のコードでは、@media に続く()内で、min-width:768pxを指定し、続く{}内でbodyの背景色が赤になるようにスタイルを指定しています（リスト1-1）。画面幅が指定した幅よりも大きいときにスタイルが適用されるので、この例では、画面幅が768px以上になると、背景色が赤になります（図1-6）。サンプルをブラウザで表示して、ウィンドウサイズを変えて、スタイルが切り替わる様子を試してみてください。

▼リスト1-1 メディアクエリの例（media-queries.html）

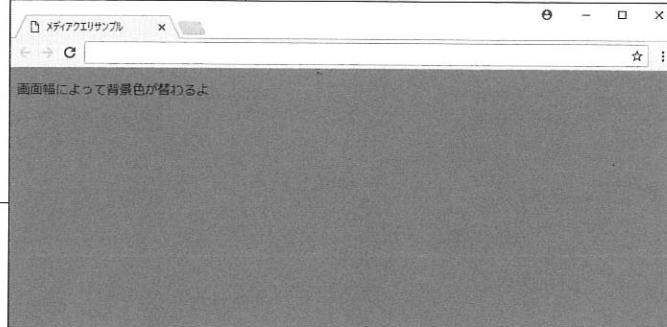
```
<style>
@media (min-width:768px){
    body { background-color: red; }
}
</style>
```

▼図1-6 画面幅768px以上で背景色が赤になる

画面幅が768px未満のとき



画面幅が768px以上のとき



このように、レスポンシブWebデザインでは、ある画面サイズ（例では768px）を境目として適用させるスタイルを切り替えますが、この境目を**ブレイクポイント**と言います。

フルードグリッド、フルードイメージ、メディアクエリ、これら3つの技術を駆使してレスポンシブWebデザインを実装するのは、それなりに開発コストがかかります。エンジニアにとっては簡単なことかもしれません、デザイナーにとってレスポンシブWebデザインは、少々敷居が高く感じられるかもしれません。

しかし、Bootstrapを利用すれば、あらかじめ用意された簡単なクラスを追加するだけで、レスポンシブに対応したモバイルファーストなWebページをあつという間に作ることができます。

NOTE

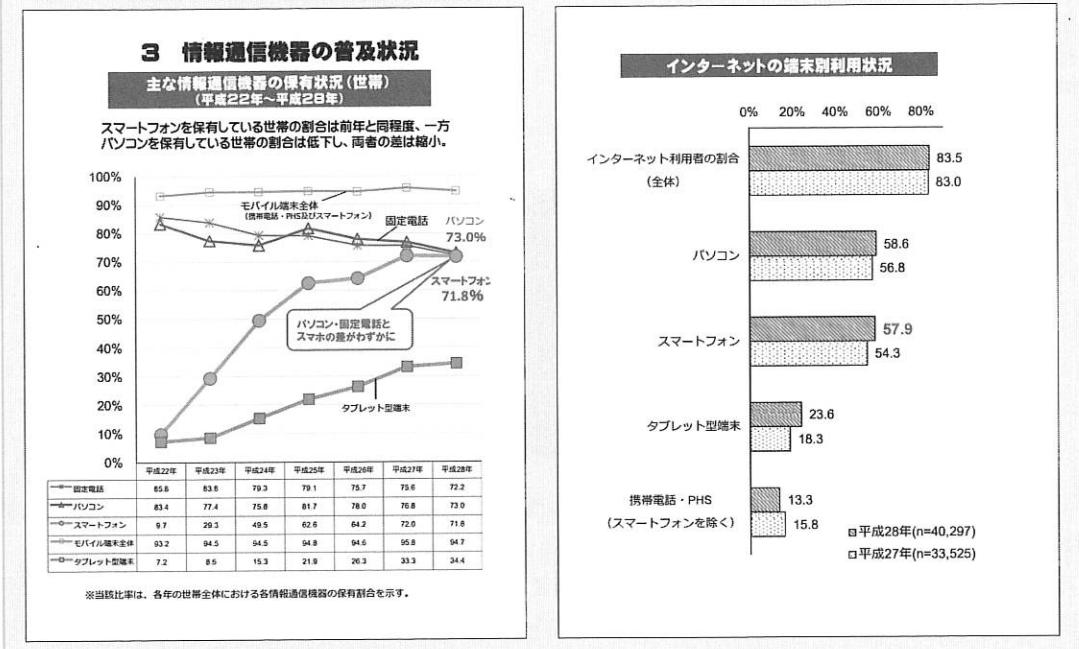
モバイルフレンドリーの時代

最近のWebページは、スマートフォンで閲覧されることが前提となっています。総務省「平成28年度通信利用動向調査」によると、パソコンの保有率は平成22年の85.8%から平成28年では72.2%と減少。一方のスマートフォン保有率は、平成22年の9.7%から平成28年には71.8%と急増しています。また、同調査の「インターネットの端末別利用状況」を見ても、パソコン58.6%、スマートフォン57.9%とどちらも僅差となっています（図1-7）。

パソコンだけでなくスマートフォンやタブレットを使ったインターネット利用者が増えたことで、Web開発の現場でも、デバイスごとにデザインを最適化することが求められるようになりました。

また、2015年に代表的な検索エンジンサービスのGoogleが、モバイル検索結果のランキング要素の1つとして、モバイルフレンドリー（モバイル対応）であるかどうかを使用すると発表しました。そこで推奨されたのがレスポンシブWebデザインです。

▼図1-7 総務省「平成28年度通信利用動向調査」



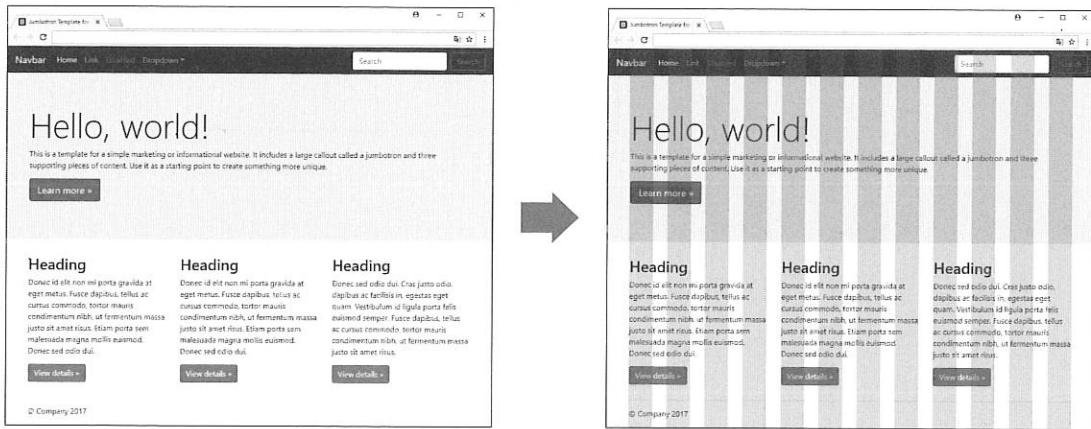
1.2.2 グリッドシステムによるレイアウト

グリッドシステム (Grid System) とは、スイスのグラフィックデザイナー、ヨゼフ・ミューラー・ブロックマン (Josef Müller-Brockmann) 氏が提唱したレイアウトデザインの手法です。具体的には、グリッド（縦横線の格子）状のガイドラインを下地に、画像や文字をブロックごとに配置する手法を指します。統一感や一貫性といったレイアウトのクオリティを維持するために使用されてきました。

近年は Web ページのレイアウト手法としても用いられる機会が増え、Bootstrap をはじめとする多くの CSS フレームワークのレイアウト手法としても採用されています。

Bootstrap では、レイアウトを縦に 12 分割した 12 カラムのグリッドシステムが採用されており、横一行の合計が 12 カラムになるようにレイアウトをしていきます。グリッドシステムとレスポンシブ Web デザインを組み合わせて、モバイルでは 12 カラム分、デスクトップでは 4 カラム分といった、画面に応じたコンテンツの配置が可能になります（図 1-8）。

▼図 1-8 グリッドシステムの例：12 列のグリッドに沿って配置されている



1.2.3 デザイン性に優れたコンポーネント

Bootstrap では、フォームやボタン、ナビゲーションなど、GUI (Graphical User Interface) のパーツも事前に用意されています。このような要素（パーツ、部品）を**コンポーネント**と呼びます。

洗練されたデザインのコンポーネントをゼロから作ろうとすると、かなりの時間がかかります。Bootstrap を利用すれば、HTML のタグにクラスを追加するだけで、体裁の整ったコンポーネントの利用が可能になります。

デザインの不得手なエンジニアでも、簡単に見栄えのいい Web サイトや Web アプリケーションを作ることができます（図 1-9）。

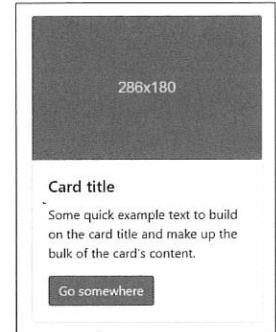
▼図 1-9 Bootstrap のコンポーネントの例
ナビゲーションバー



フォーム

A screenshot of a Bootstrap form. It includes fields for "Email address" (placeholder "Enter email") and "Password" (placeholder "Password"). Below these fields is a checkbox labeled "Check me out" and a "Submit" button.

カード



ドロップダウン



1

3

SECTION

Bootstrapの歴史

Bootstrapは、何度かのバージョンアップを経て、世界中でもっとも有名なWebフレームワークの1つになりました。本節ではBootstrapの誕生した背景から、これまでの歴史を見していくことにしましょう。

1.3.1

Bootstrapの誕生と歩み

元々 **Twitter Blueprint** という名称だったBootstrapは、2010年半ばにアメリカのTwitter社のMark Otto氏とJacob Thornton氏によって作されました。かつてのTwitter社のインターフェイス開発には、さまざまなライブラリが使用されていたため、一貫性がなく、メンテナンスにかかる負担は大きなものでした。そこで、内部ツール間の統一を図るフレームワークとしてTwitter Blueprintが誕生したのです。

少数グループによる数か月の開発期間を経て、Twitter開発チームは、ハッカソンスタイル^{*1}のHack Weekと呼ばれる開発イベントを開催。Twitter社の多くの開発者がこれに参加し、プロジェクトは急激に大きくなりました。オープンソースとして公開される前から、社内ツール開発のスタイルガイドとして広く使われるようになります(図1-10)。

▼図1-10 最初のHack Weekのアナウンス。現在でもBootstrapのAboutページからリンクされ、閲覧できる(https://blog.twitter.com/engineering/en_us/a/2010/hack-week.html)



*1 ハッカソン(hackathon) : hack(ハック) + marathon(マラソン)からの造語。エンジニア、デザイナー、プロジェクトマネージャーなどの技術者が集まってチームを作り、与えられたテーマに対して、それぞれの技術やアイデアを持ち寄り、会場にこもって集中的に開発を行うイベントのことです。

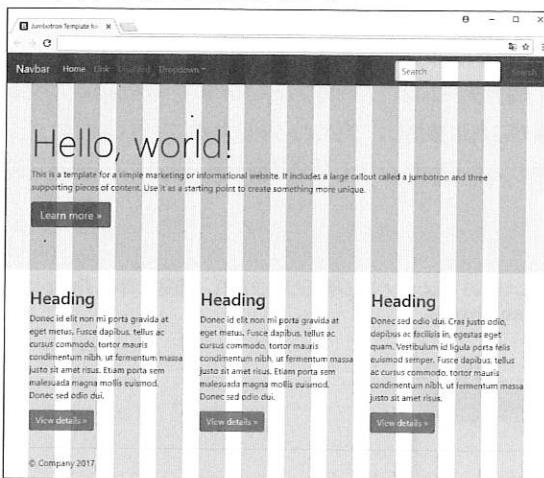
その後、「Twitter Blueprint」から「Twitter Bootstrap」に改名され、2011年8月19日にオープンソースのプロジェクトとしてリリースされます。現在は、Mark Otto 氏と Jacob Thornton 氏、コア開発者の少数グループだけでなく、多くの貢献者が集まつた大きなコミュニティによって維持されています。

オープンソースとなった当初は、Twitter 社から提供されていて、Twitter Bootstrap と呼ばれていましたが、プロジェクトのメイン開発者である Mark Otto 氏と Jacob Thornton 氏が Twitter 社を退社した後は、Twitter 社から独立したプロジェクトとなり、名称も「Twitter Bootstrap」から「Bootstrap」に変更されました。

1.3.2 バージョンによる違い

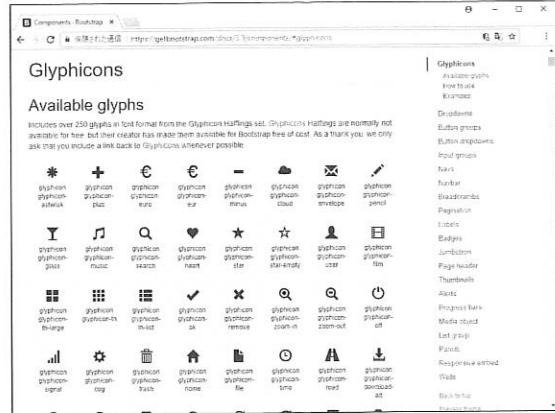
2012年1月31日にリリースされたBootstrap 2では、レスポンシブな12カラムのグリッドシステム（図1-11）が採用されました（P.22参照）。

▼図 1-11 グリッドシステムのイメージ



その他、グリフアイコンのサポート、既存コンポーネントに変更が加わり、新しいコンポーネントも追加されました（図1-12）。

▼図 1-12 グリフアイコン（バージョン 4 で廃止）



2013年8月19日には、コンポーネントにフラットデザインを使用したBootstrap 3がリリースされ、このバージョンでは、モバイルファーストのアプローチが採用されました（図 1-13）。

▼図 1-13 フラットデザインのボタン



そして、2018年1月18日、執筆時現在最新版であるBootstrap 4がリリースされました。Bootstrap 4では、コンポーネント、jQuery プラグイン、ドキュメントなど、Bootstrap 3のほぼすべてが書き直されました。中でも大きな変更点は次のとおりです。

- CSS プリプロセッサが Less から Sass に
- Internet Explorer 8、Internet Explorer 9、iOS 6 がサポート対象外に
- flexbox がサポートされ、非 flexbox がサポート対象外に
- 単位がピクセルから rem に
- パネル、サムネイルが廃止され、新たにカードが追加される
- グリフアイコンが廃止される
- ページャーコンポーネントが廃止される

細かな変更点については、Bootstrap 公式サイトの「Migrating to v4」のページ (<https://getbootstrap.com/docs/4.1/migration/>) を参照してください。

1

SECTION

4

Bootstrapの導入

前置きはこのくらいにして、Bootstrapを利用する準備をしていきましょう。Bootstrapを導入するには、いくつかの方法がありますが、本書では、すぐに使えるように、コンパイル済みのCSSとJavaScriptをダウンロードする方法を採用します。

1.4.1 Bootstrapの導入に必要な環境

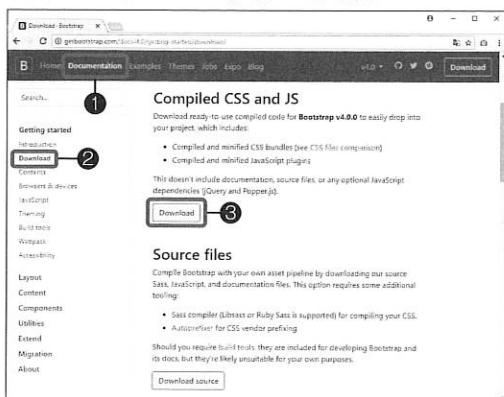
Bootstrapを導入するにあたっては、これまでのフロントエンドの開発環境があればそれで充分です。特別に必要となるものはありません。必要なアプリケーションは、HTML、CSS、Javascriptを編集するテキストエディタと、表示を確認するWebブラウザです。コードの編集には、Adobe Brackets、Sublime Textなどのテキストエディタでも、Adobe DreamweaverなどのWebオーサリングツールでもOKです。普段使っているものを使ってください。

1.4.2 ダウンロードして利用する

それではBootstrapをダウンロードする方法を見ていきましょう。Bootstrapの公式サイト(<http://getbootstrap.com/>)にアクセスし、[Documentation]の中の[Download] (<http://getbootstrap.com/docs/4.1/getting-started/download/>)に移動します。

このページの[Compiled CSS and JS]の下にある[Download]ボタンをクリックし、ファイルのダウンロードを実行します(図1-14)。

▼図1-14 [Download]ボタンをクリック



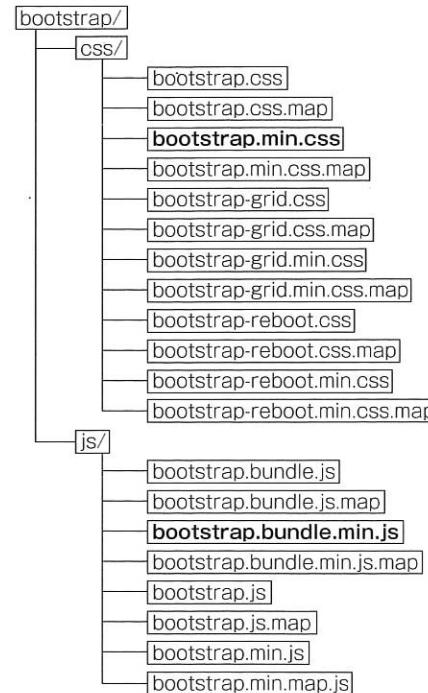
ダウンロードされたファイルは ZIP 形式で圧縮されています。これを解凍すると、css フォルダーおよび js フォルダーができます。

■ ディレクトリ構成

ダウンロード後、解凍してきたフォルダー内には、css フォルダーにはコンパイルされた CSS ファイル、js フォルダーには JavaScript ファイルが格納されています。同じ名前で、「min」が付いているものは、同名ファイルの内容から、改行やコメント、スペースなどを排除して圧縮（Minify）した軽量版です。（図 1-15）。通常は、軽量版を使うと良いでしょう。

いろいろなファイルがありますが、Bootstrap の全機能に対応した CSS ファイルが bootstrap.css（とその軽量版の bootstrap.min.css）、すべての JavaScript プラグインを利用できるのが、bootstrap.bundle.js（とその軽量版の bootstrap.bundle.min.js）です。初心者の方やベーシックな使い方をする場合は、この 2つを使うのが一般的です。本書でも、これらの軽量版の「bootstrap.min.css」と「bootstrap.bundle.min.js」を使用します。

▼図 1-15 ダウンロードファイルのディレクトリ構成



また、拡張子が .map のファイルは、**CSS ソースマップファイル**と言って、Sass などを使用していた場合に、オリジナルのソースとコンパイル後のソースを紐づけてくれるファイルです。開発時に使用するもので、コンパイラを使わない場合には必要ありません。

COLUMN Sassとコンパイル

Bootstrap のダウンロードページでは、Sass のソースファイルもダウンロードできるようになっています。Sass (Syntactically Awesome Stylesheets) とは、簡単に言うと CSS を効率的に記述・管理できるもので、CSS 拡張メタ言語とも呼ばれます。Sass には、「.sass」と「.scss」の2つの形式の拡張子がありますが、Bootstrap の Sass は .scss 形式です。これまでの CSS に、変数や計算式を使ったプログラミング風の書き方をした .scss を CSS 形式の .css に変換することをコンパイルと言います。

■ CSS ファイルの内容

他にもいろいろな CSS ファイルがありますが、これは、Bootstrap のどの機能を使うかによって選択します（表 1-1）。詳しくはこれから学んでいきますが、Bootstrap では、レイアウトするための機能、見出しやリストなどのコンテンツの体裁を整える機能、ボタンやナビゲーションなどのコンポーネント、配置や表示に関するユーティリティなど、さまざまな機能を利用できます。これら Bootstrap の機能のうち、たとえば、グリッドシステムを使ったレイアウトの部分だけを使用したい場合は、bootstrap-grid.css（またはその軽量版の bootstrap-grid.min.css）を選びます。

また、ブラウザごとのスタイルの相違をなくすことを目的として、ブラウザのデフォルトスタイルを独自のスタイルで再定義することを、Bootstrap では **Reboot** と言います（詳しくは P.96 参照）。この機能のみを使いたい場合には、bootstrap-reboot.css（またはその軽量版の bootstrap-reboot.min.css）を使用します。このように、用途によって必要なものだけを選ぶことで Web サイトの軽量化を図ることができます。

▼表 1-1 CSS ファイルの内容

CSS ファイル	レイアウト	コンテンツ	コンポーネント	ユーティリティ
bootstrap.css (bootstrap.min.css)	含まれる	含まれる	含まれる	含まれる
bootstrap-grid.css (bootstrap-grid.min.css)	グリッドシステムのみ	含まれていない	含まれていない	Flex ユーティリティのみ
bootstrap-reboot.css (bootstrap-reboot.min.css)	含まれていない	Reboot のみ	含まれていない	含まれていない

■ JavaScript ファイルの内容

同様に、JavaScript ファイルにも全機能を利用できる bootstrap.bundle.js（とその軽量版の bootstrap.bundle.min.js）と、一部の JavaScript プラグインで必要な popper.js を含まないバージョンの bootstrap.js（とその軽量版の bootstrap.min.js）があります（表 1-2）。その他、ソースファイル版をダウンロードすると、プラグインを個別に選ぶこともできます（P.260 参照）。本書では、全部入りの軽量版 bootstrap.bundle.min.js を使用します。

▼表 1-2 JavaScript ファイルの内容

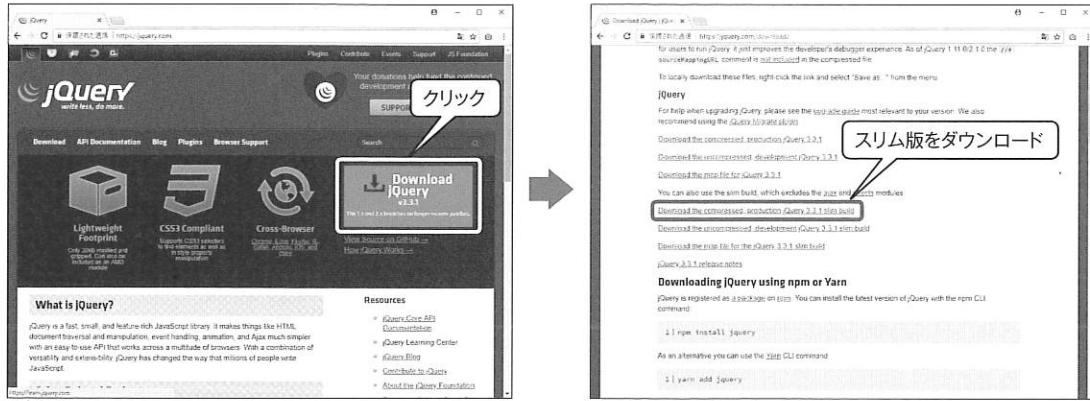
JavaScript ファイル	popper.js
bootstrap.bundle.js (bootstrap.bundle.min.js)	含まれる
bootstrap.js (bootstrap.min.js)	含まれない

jQuery の入手

Bootstrap の JavaScript は、**jQuery** というライブラリを必要とします。jQuery 3.0からは、標準版に加え、標準版から ajax や effects、非推奨コードを除いたスリム版が利用できるようになりました。Bootstrap はどちらの版もサポートしていますが、本書ではスリム版を使用します。

では jQuery の公式サイト (<https://jquery.com/>) から [Download jQuery] をクリックしダウンロードページに移動しましょう。ダウンロードページには、いろいろな種類のファイルがありますが、[Download the compressed, production jQuery X.X.X slim build (X.X.Xにはバージョンを表す数字が入ります)] をクリックして、jQuery のスリム版をダウンロードしてください（図 1-16）。

▼図 1-16 jQuery のスリム版をダウンロード

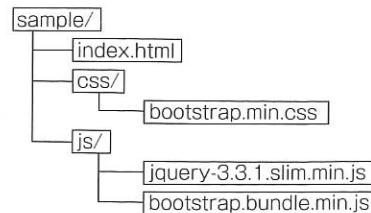


本書の執筆時点では「jQuery 3.3.1」が最新版です。jQuery にも空白や改行などを取り除いて圧縮した compress 版 (jquery-3.3.1.slim.js) と、オリジナルの uncompressrd 版 (jquery-3.3.1.slim.min.js) があります。本書では、compress 版の jquery-3.3.1.slim.min.js を使用します。

1.4.3 HTML の雛形

Bootstrap で Web サイトを構築する場合は、ダウンロードした CSS ファイルや JavaScript ファイル、jQuery ライブラリを HTML から読み込む必要があります。図 1-17 のように作業用フォルダー（ここでは sample フォルダー）を作りその中に、css フォルダー、js フォルダーを作ります。css フォルダーには、ダウンロードしてきた bootstrap.min.css を、js フォルダーには、jquery-X.X.X.slim.min.js と bootstrap.bundle.min.js を格納します。これから作る HTML ファイル (index.html) は、sample フォルダーに保存します。

▼図 1-17 本書のサンプルの構成



HTML の雛形は、Bootstrap の公式サイトの [Documentation] > [Getting started] > [Introduction] (<http://getbootstrap.com/docs/4.1/getting-started/introduction/>) に「Starter template」として紹介されています。本書で使用するサンプルは、この「Starter template」を少し改良したものを使用します（リスト 1-2）。

▼リスト 1-2 HTML の雛形 (index.html)

```

<!doctype html> _____ ①
<html lang="ja"> _____ ②
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"> _____ ③
    <link rel="stylesheet" href="css/bootstrap.min.css"> _____ ④
    <title>Hello, world!</title>
  </head>
  <body>
    <h1>Hello, world!</h1>
    <script src="js/jquery-3.3.1.slim.min.js"></script>
    <script src="js/bootstrap.bundle.min.js"></script> _____ ⑤
  </body>
</html>
  
```

① HTML5 でコードを書く

Bootstrap は、HTML5 に準拠しているため、最初に HTML5 の doctype を書きましょう。

② 言語を日本語にする

2 行目の html 要素の lang 属性の値を日本語の ja にします。

③ viewport メタタグを書く

適切なレンダリングを実現するためメタタグで viewport を設定しましょう。「width=device-width, initial-scale=1」は、「デバイス幅に合わせる、ズーム倍率 1」という設定です。「shrink-to-fit=no」は、iOS 9.0 の Safari で、initial-scale が適用されないのを回避するための設定です。

④ CSS を読み込む

CSS を読み込むために、<head> 内の他のすべてのスタイルシートの前に、Bootstrap のスタイルシート (bootstrap.min.css) を読み込む <link> を記述します。

⑤各種 Javascript を読み込む

各種 Javascript ファイルを読み込みます。最初に、Bootstrap の JavaScript プラグインを動かすためのライブラリである **jQuery (jquery-3.3.1.slim.min.js)** を読み込みます。次に、Bootstrap (**bootstrap.bundle.min.js**) を読み込みます。読み込む順番に注意してください。本書では、すべての機能が使える一括版の bootstrap.bundle.min.js を使用しますが、JavaScript プラグインを個別に読み込む場合は、最初に jQuery、次に Popper.js (そのプラグインに必要な場合)、最後に JavaScript プラグインの順番で読み込みます (P.260 参照)。

以上で Bootstrap を利用する準備が整いましたが、内容がないため、今はブラウザで見ても何も変化はありません。そこで h1 要素に Bootstrap 定義済みクラスを追加してみましょう (リスト 1-3)。

▼リスト 1-3 HTML の雛形 (index.html)

```
<h1 class="text-success">Hello, world!</h1>
```

ブラウザで表示させ、h1 要素が緑色になれば OK です (図 1-18)。もし、このようにならないときは、css ファイルが正しい場所にあるかどうかを確認してください。

▼図 1-18 HTML の雛形の実行結果



1.4.4 CDN を利用する

ファイルをダウンロードして利用する他、Stack Path が無償で提供している Bootstrap CDN^{*2} を利用することもできます。リスト 1-4 のサンプルは、前項で紹介した Bootstrap 公式サイトの「Starter template」をベースにしたもので、前項の雛形との違いはファイルの参照先が CDN になっているという点と、CDN からのクロスサイトスクリプティング (XSS) を防ぐため integrity 属性と crossorigin 属性があるという点です。

*2 CDN (Content Delivery Network):コンテンツ配信ネットワークの略。インターネット上にキャッシュサーバーを分散配置し、エンドユーザーにもっとも近い経路にあるキャッシュサーバーからコンテンツを高速で安定して配信するしくみです。

▼リスト 1-4 Bootstrap CDN による設定例 (starter-cdn.html)

```
<!doctype html>
<html lang="ja">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css" integrity="sha384-WskhaSGFgHYWDcbwN70/dfYBj47jz9qbsMId/iRN3ewGhXQFZCSftd1LZCfmhktB" crossorigin="anonymous">
<title>Hello, world!</title>
</head>
<body>
<h1>Hello, world!</h1>
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965Dz0rT7abK41JStQIAqVgRVzbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js" integrity="sha384-ZMP7rVo3mIykV+2+9J3UJ46jBk0WLAdn689aCwoqbBJiSnjAK/l8WvCWPIPM49" crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js" integrity="sha384-smHYKdLADwkXOn1EmN1qk/HfnUcbVRZyYmZ4qpPea6sjB/pTJ0euyQp0Mk8ck+5T" crossorigin="anonymous"></script>
</body>
</html>
```

NOTE

integrity 属性と crossorigin 属性

CDN からのクロスサイトスクリプティング (XSS) を防ぐため、ブラウザが取得するリソース (ファイル) をハッシュ化して、その値を link 要素や script 要素の integrity 属性に指定することで、整合性を確認できるようにするものを Subresource Integrity と言います。integrity 属性がある場合、ブラウザは読み込み時に整合性をチェックし、一致した場合にのみロードされるしくみです。また、crossorigin 属性に「anonymous」が指定された場合は、リクエストには cookie やクライアントサイドの SSL 証明書、HTTP 認証などのユーザー認証情報は利用されません。