2.1 Bootstrap のグリッドシステム

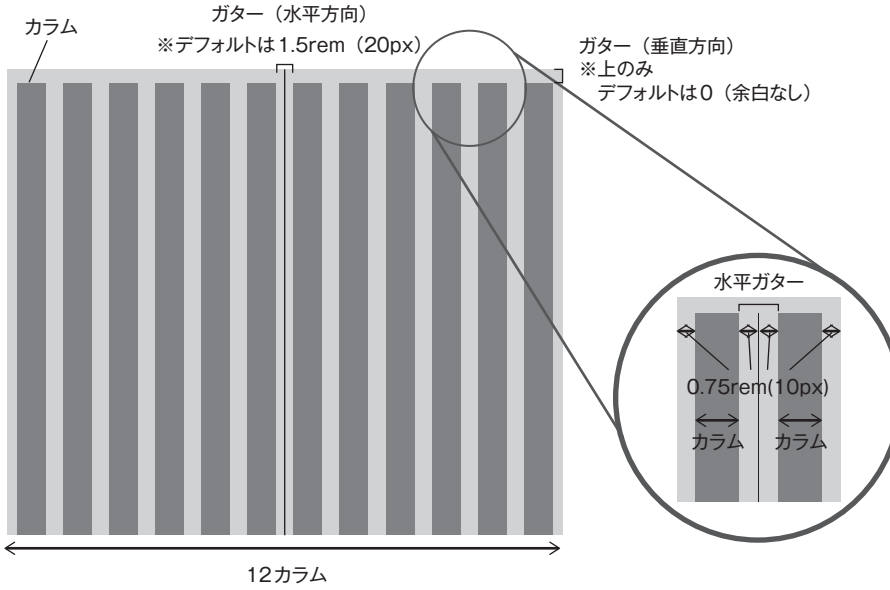
Bootstrap のレイアウト方法には、ページの幅を 12 列に分割した**グリッドシステム**が採用されています。この機能は、ページを 2 段組みや 3 段組みにしたり、コンポーネントを横に並べて配置したりする場合などに活用できます。それでは、Bootstrap のグリッドシステムを構成する要素を基本的なところから見ていきましょう。

2.1.1 カラム（column：列）とガター（gutter：溝）

12 に分割されたグリッドの列を**カラム**（column：列）と言い、各カラムの間の余白を**ガター**（gutter：溝）と言います。ガターは、水平方向に対するものと垂直方向に対するものがあります。水平方向にはカラムの左右に均等にガターが設けられ、垂直方向にはカラムの上にのみガターが設けられます。

デフォルトでは、各カラムの左右内側（padding）には 0.75rem（10px）のガターが設定されているため、コンテンツ間の左右の余白は 1.5rem（20px）となります。各カラムの上外側（margin-top）にもガターが設定されていますが、大きさが 0 なので、カラム間の上下には余白が発生しません（図 2-1）。

▼図 2-1　カラムとガター



水平、垂直方向ともにガターの大きさは変更できます。変更した場合、垂直方向にも余白を作ることができます。

```
    <!-- 背景色：Dark -->
    <tr class="table-dark">
      <th scope="row">table-dark</th>
      …中略…
    </tr>
  </tbody>
</table>
```

▼図 3-40　テーブル行に色を付ける

背景色用のクラス	見出しセル	見出しセル	見出しセル
なし（標準）	データセル	データセル	データセル
table-primary	データセル	データセル	データセル
table-secondary	データセル	データセル	データセル
table-success	データセル	データセル	データセル
table-danger	データセル	データセル	データセル
table-warning	データセル	データセル	データセル
table-info	データセル	データセル	データセル
table-light	データセル	データセル	データセル
table-dark	データセル	データセル	データセル

個々のセルに色を付ける

個々のセルに色付けを行う場合は、th 要素または td 要素に背景色用クラスを追加します（リスト 3-73、図 3-41）。

▼リスト 3-73　個々のセルに色を付ける（table-bg-td.html）

```
<table class="table">
  <tr>
    <th colspan="2">th（標準） </th>
  </tr>
  <tr>
    <!-- 背景色：Primary -->
    <td class="table-primary">td.table-primary</td>
    <!-- 背景色：Secondary -->
    <td class="table-secondary">td.table-secondary</td>
  </tr>
  <tr>
    <!-- 背景色：Success -->
    <td class="table-success">td.table-success </td>
    <!-- 背景色：Danger -->
    <td class="table-danger">td.table-danger</td>
```

4.2 バッジ

Bootstrap の**バッジ**はカウンター表示やラベリングを行うためのコンポーネントです。本節では Bootstrap のバッジを使用する方法を解説します。

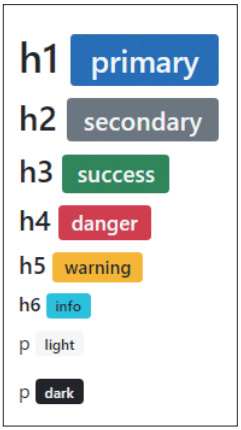
4.2.1 基本的な使用例

バッジを作成するには、span 要素に **badge クラス**、**bg-{色の種類}クラス**、**text-{色の種類}クラス**を追加します。色の種類には **primary**（青）、**secondary**（グレー）などコンテクストに対応した色の種類（P.121 参照）を入れて、バッジの背景色および文字色を設定します（リスト 4-6、図 4-6）。なおバッジのサイズは、親要素のフォントサイズと一致するように調整されます。

▼リスト 4-6　バッジの基本的な使用例（badge-basic.html）

```
<h1>h1 <span class="badge bg-primary text-light">primary</span></h1>
<h2>h2 <span class="badge bg-secondary text-light">secondary</span></h2>
<h3>h3 <span class="badge bg-success text-light">success</span></h3>
<h4>h4 <span class="badge bg-danger text-light">danger</span></h4>
<h5>h5 <span class="badge bg-warning text-dark">warning</span></h5>
<h6>h6 <span class="badge bg-info text-dark">info</span></h6>
<p>p <span class="badge bg-light text-dark">light</span></p>
<p>p <span class="badge bg-dark text-light">dark</span></p>
```

▼図 4-6　バッジの基本的な使用例



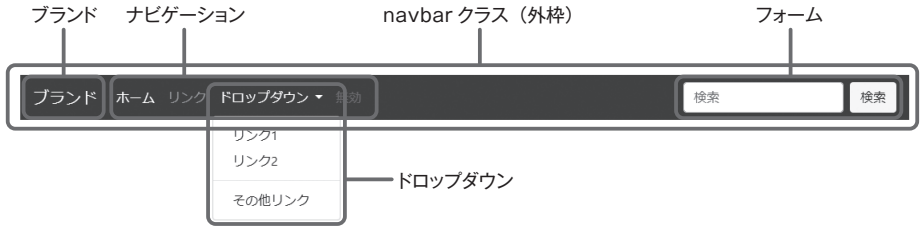
4.2.2 カウンターを作成する

バッジを使用して、ボタンやリンクに付けるカウンターを作成します。バッジの基本的な使用例と同様に、ボタンやリンク内の span 要素に **badge クラス**および **bg-{色の種類}クラス**、**text-{色の種類}クラス**を追加します。次の例では、button 要素に btn クラスおよび btn-primary クラス（P.244 参照）を追加した青色のボタン内に、**badge クラス**および **bg-light クラス**、**text-dark クラス**を追加した span 要素を配置し、明るいグレーの背景色に「4」を暗色で表示するカウンターを作成しています（リスト 4-7、図 4-7）。

5.2 ナビゲーションバー

Bootstrap の**ナビゲーションバー**は、ロゴマークやナビゲーション、検索フォームなど複数のコンテンツを含んだナビゲーションバーを作成するコンポーネントです。本節では、Bootstrap のナビゲーションバーと、そこに含まれるコンテンツを作成するサブコンポーネントの使用法を解説します。ナビゲーションバーは、中に組み込むサブコンポーネントの種類が多くバリエーションも豊富です。その分、コードも非常に長くなるため、最初は覚えるのが大変ですが、コンポーネントの構造を把握しながら覚えていくようにしましょう（図 5-15）。

▼図 5-15　サブコンポーネントを配置したナビゲーションバーの例



5.2.1 外枠の作成

まず nav 要素または div 要素に **navbar クラス**を追加して、ナビゲーションバーの外枠を作成します。ナビゲーションバーの外枠となる要素には、navbar クラスを含め表 5-2 のようなクラスを追加します。

▼表 5-2　ナビゲーションバーの外枠に使用するクラス

クラス	概要
navbar	ナビゲーションバーの外枠を作成
navbar-expand-{ブレイクポイント}	ブレイクポイントでナビゲーション項目の表示を切り替え
navbar-{背景色の種類}	背景色が dark の場合は明るい文字色、light の場合は暗い文字色を設定
bg-{色の種類}	primary（青）、secondary（グレー）など背景色を設定
container-fluid	サブコンポーネントを入れるコンテナ

次の例では、navbar クラスを設定した nav 要素に **navbar-expand-lg クラス**を追加して、画面幅 lg 以上になるとナビゲーション項目が横に広がって表示され、画面幅 md 以下になると折り畳まれてアイコン表示になるナビゲーションバーを作成しています。また **bg-dark クラス**および **navbar-dark クラス**を追加して、ナビゲーションバーの背景色が暗いときにサブコンポーネントの文字色を明るくして読みやすくなるように設定しています（リスト 5-17）。さらに、サブコンポーネントは **container-fluid クラス**を追加した div 要素で囲みます。なおサブコンポーネントについては次項より解説します。

6.3 ボタン

Bootstrap にはさまざまなサイズ、状態のボタンがコンポーネントとして用意されており、定義済みクラスを要素に追加するだけで洗練された UI を実装することができます（図 6-39）。本節では、Bootstrap で**ボタン**を使用する方法を解説します。

▼図 6-39　通常のボタンとボタンコンポーネントとの比較（button-compare.html）

6.3.1 基本的な使用例

ボタンのコンポーネントは、button 要素や a 要素、input 要素に **btn クラス**と **btn-{ 色の種類 } クラス**を追加して作成します。アラート（P.120 参照）と同様、色の種類には **primary**（青）、**secondary**（グレー）などコンテクストに対応した色の種類が入ります。

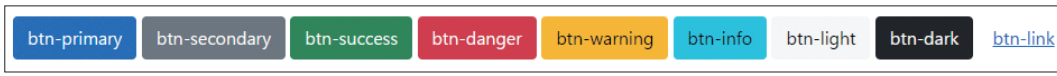
button 要素でボタンを作成する

次の例では、button 要素に **btn クラス**と **btn-{ 色の種類 } クラス**を追加してボタンを作成しています（リスト 6-40、図 6-40）。

▼リスト 6-40　ボタンの基本的な使用例（button-basic.html）

```
<button type="button" class="btn btn-primary">btn-primary</button>
<button type="button" class="btn btn-secondary">btn-secondary</button>
<button type="button" class="btn btn-success">btn-success</button>
<button type="button" class="btn btn-danger">btn-danger</button>
<button type="button" class="btn btn-warning">btn-warning</button>
<button type="button" class="btn btn-info">btn-info</button>
<button type="button" class="btn btn-light">btn-light</button>
<button type="button" class="btn btn-dark">btn-dark</button>
<button type="button" class="btn btn-link">btn-link</button>
```

▼図 6-40　ボタンの基本的な使用例



7.1 Bootstrap の JavaScript プラグイン

Bootstrap の JavaScript プラグインを利用すると、コンポーネントに動きを追加することができます。プラグインは、Bootstrap が提供する個別の *.js ファイルを必要に応じて組み込むか、一括版を使用すれば利用できます。

7.1.1 Bootstrap の JavaScript ファイル

Bootstrap の個別の JavaScript ファイルは、ソースファイルの js/dist/ 内にあります。またそれらを結合した一括版にも 2 種類あり、ドロップダウン、ポップオーバー、ツールチップで必要になる Popper.js が組み込まれている bootstrap.bundle.js（もしくはその圧縮版の bootstrap.bundle.min.js）と、Popper.js が含まれていない bootstrap.js（もしくはその軽量版の bootstrap.min.js）があります（表 7-1）。

▼表 7-1　Bootstrap の JavaScript ファイル

		JavaScript プラグイン	説明
bootstrap.bundle.js bootstrap.bundle.min.js	bootstrap.js bootstrap.min.js	alert.js	アラートメッセージ
		button.js	コントロールボタン
		carousel.js	カルーセル
		collapse.js	折り畳み、ナビゲーションバーの切り替え
		dropdown.js	ドロップダウン
		modal.js	モーダル
		offcanvas.js	オフキャンバス
		popover.js	ポップオーバー
		scrollspy.js	スクロールスパイ
		tab.js	タブ、リストグループ切り替えパネル
		toast.js	トースト
		tooltip.js	ツールチップ
		base-component.js	基本コンポーネント。他の JavaScript ファイルと一緒に読み込む必要あり
		dom/*.js	DOM に関する機能。他の JavaScript ファイルと一緒に読み込む必要あり
		popper.js	ドロップダウン、ポップオーバー、ツールチップ使用時に必要

8.6 Flex ユーティリティ

Flex ユーティリティは、CSS3 の **Flexible Box**（フレキシブルボックス）または **flexbox**（フレックスボックス）と呼ばれるレイアウトモードを使用できるユーティリティです。flexbox を使ったレイアウトでは、従来の float を使ったレイアウトより、より柔軟に要素を配置することができます。

8.6.1 flexbox を有効にするクラス

flexbox を利用するには、要素に Display ユーティリティ（P.358 参照）の **d-flex クラス**（display:flex）または **d-inline-flex クラス**（display:inline-flex）を追加します。これにより、要素は **flex コンテナ**となり、またその子要素は自動的に **flex アイテム**となります。flex アイテムとなった子要素は、flex コンテナ内で任意の方向にレイアウトすることが可能になります（リスト 8-25、図 8-19）。

▼リスト 8-25　flexbox を有効にする（d-flex.html）

```
<div class="d-flex">flexコンテナ</div>
<div class="d-inline-flex">インラインのflexコンテナ</div>
```

▼図 8-19　flexbox を有効にする

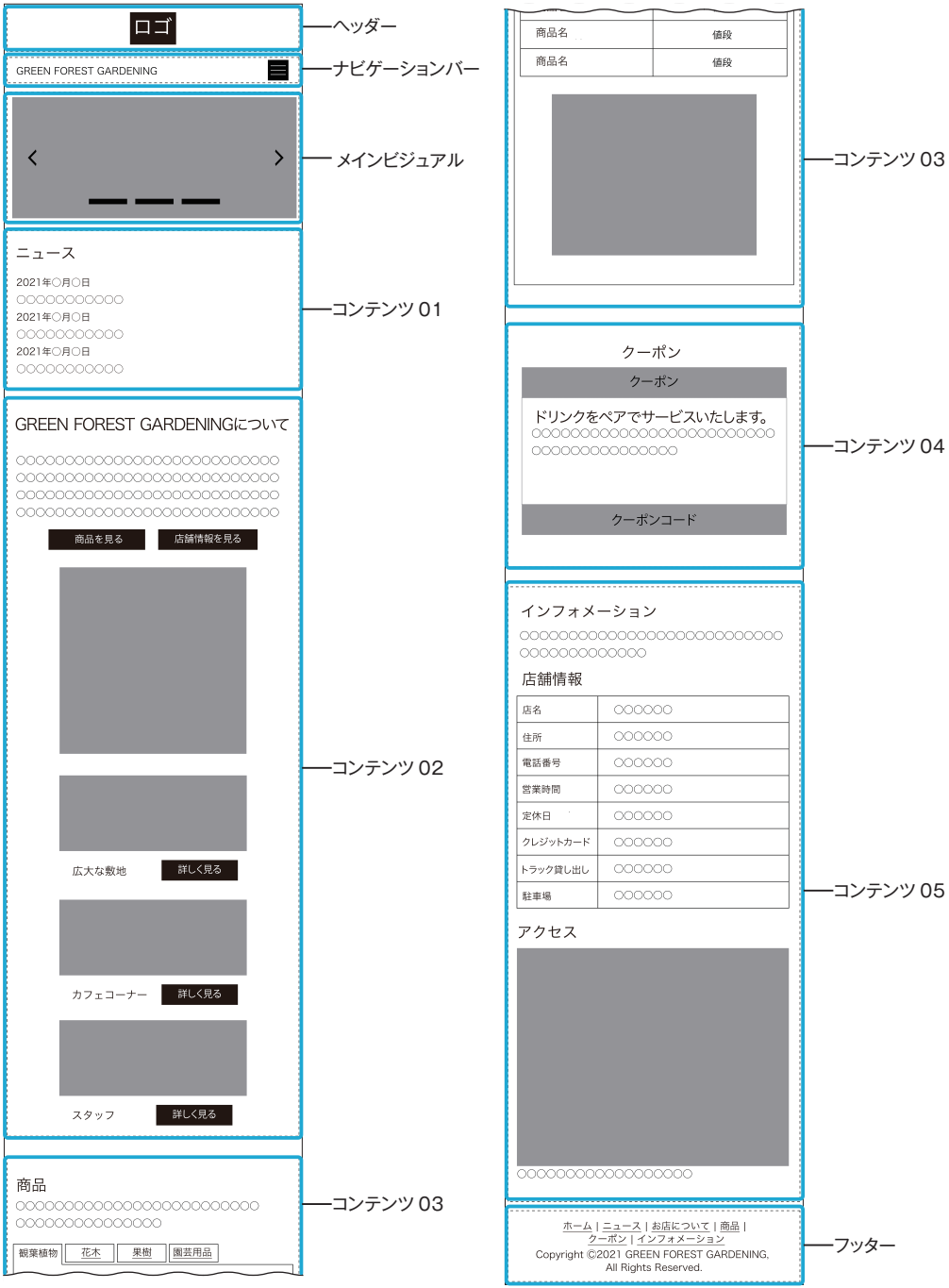
また各ブレイクポイントに対応する接頭辞 sm、md、lg、xl、xxl 付きの **d-{ブレイクポイント}-flex クラス**、および **d-{ブレイクポイント}-inline-flex クラス**も定義されています。たとえば、**d-sm-flex クラス**を指定すると、Small 以上の画面幅で表示形式を flex コンテナとして設定することができます（表 8-17）。

▼表 8-17　flexbox を有効にするクラス一覧

クラス	レスポンシブ対応クラス	説明
d-flex	d-{ブレイクポイント}-flex	flex コンテナをブロックレベルにする
d-inline-flex	d-{ブレイクポイント}-inline-flex	flex コンテナをインラインレベルにする

デスクトップ PC（画面幅 md 以上）での閲覧時は図 9-3 のように表示され、モバイル端末（画面幅 sm 以下）での閲覧時は図 9-4 のようにレイアウトが変更される設計です。

▼図 9-4　トップページのワイヤーフレーム（モバイル端末用）



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。すでに 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、ここでは、sass（Dart Sass、機能としての Sass と紛らわしいので以降は Dart Sass とします）を使った Sass の導入方法を説明します。Dart 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アプリケーションです。Dart Sass は Dart 言語で書かれており、スタンドアロンのプログラムや CLI（Command Line Interface）を備えています。JavaScript のコードにコンパイルされた npm（Node Package Manager）パッケージ版もあります。Bootstrap 4 ではサーバーサイドの JavaScript 実行環境である Node.js で動く node-sass（LibSass）がよく使われていたため、ここでは同じく Node.js で動く Dart Sass の npm パッケージ版を使用します。

Node.js をインストールする

まずは、Dart Sass を利用するために、Node.js をインストールしましょう。ここでは、Windows を例に手順を説明します。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 10-2）。

NOTE

macOS への Node.js のインストール

macOS で公式サイトにアクセスすると macOS（x64）版のダウンロード画面が表示されるので、pkg ファイルをダウンロードすると GUI でのインストールが可能です（推奨版／最新版については Windows 同様）。

なお macOS では、Nodebrew を使用したインストールも可能です。Nodebrew を使うと Node.js のバージョン管理が容易になります。Nodebrew のインストールには、Homebrew というパッケージ管理ツールも必要です。Homebrew → Nodebrew → Node.js の 3 回のインストールになりますが、これらは Web 開発現場ではよく用いられるツールなので、この機会に触れてみるとよいでしょう。

macOS（または Linux）用パッケージマネージャー — Homebrew

https://brew.sh/index_ja.html

付録：Bootstrap 5 移行ガイド

ここでは、Bootstrap 4 のユーザーを対象に、Bootstrap 5 にスムースに移行できるようにするためのポイントを紹介します。

11.0.1 全般

削除 jQuery が不要になりました。ただし jQuery を検出すれば従来どおりに jQuery のイベントが発生するので、JavaScript コードの変更はひとまず不要です。

削除 HTML のメタタグにあった「shrink-to-fit=no」は、iOS 9.0 がサポート外となったので Bootstrap 5 では削除されています。

変更 RTL（Right to Left）のサポートにより、方向を表すいくつかのクラス名が次のように変更になっています。

変更前	変更後
left-*, right-*	start-*, end-*
*-left, *-right	*-start, *-end
ml-*, pl-*	ms-*, ps-*
mr-*, pr-*	me-*, pe-*

削除 Internet Explorer がサポート対象外となりました。これによりいくつかのクラスの定義が簡潔になっています。

11.0.2 基本的なレイアウト

新規 ブレイクポイント xxl が追加されました。最大サイズの拡張を考慮する必要があります。

新規 行あたりの列数を指定できる row-col-* クラスが設けられました。

変更 ガターは垂直方向にも設けられました（既定値は 0）。水平方向のガターの既定値は 30px から 1.5rem に変更となりました。ガターの大きさは g-*、gx-*、gy-* クラスで変更できるようになりました。

削除 上記に伴いガターサイズを 0 にする no-gutters クラスは廃止されました。

変更 項目並べ替えのための order-* クラスは、範囲が 1 ～ 12 から 1 ～ 5 に変更されました。

変更 オフセットのための offset-* クラスは、範囲が 0 ～ 12 から 0 ～ 11 に変更されました。