

# 第4章

## 基本的な コンポーネント

Bootstrapにはフォームやリスト、ナビゲーションなどさまざまなコンポーネントが用意されており、定義済みクラスを要素に追加するだけで洗練されたUIを実装できます。さらに各コンポーネントにも数種類のパターンがあり、それらを組み合わせることでバリエーションも広がります。数が多いので仕様を覚えるのは大変ですが、わかりやすく連想しやすいクラス名が使われているので、ある程度使用していくうちに覚えることができるでしょう。本章では、Bootstrapで使用できる基本的なコンポーネントの使い方を解説します。

## 4

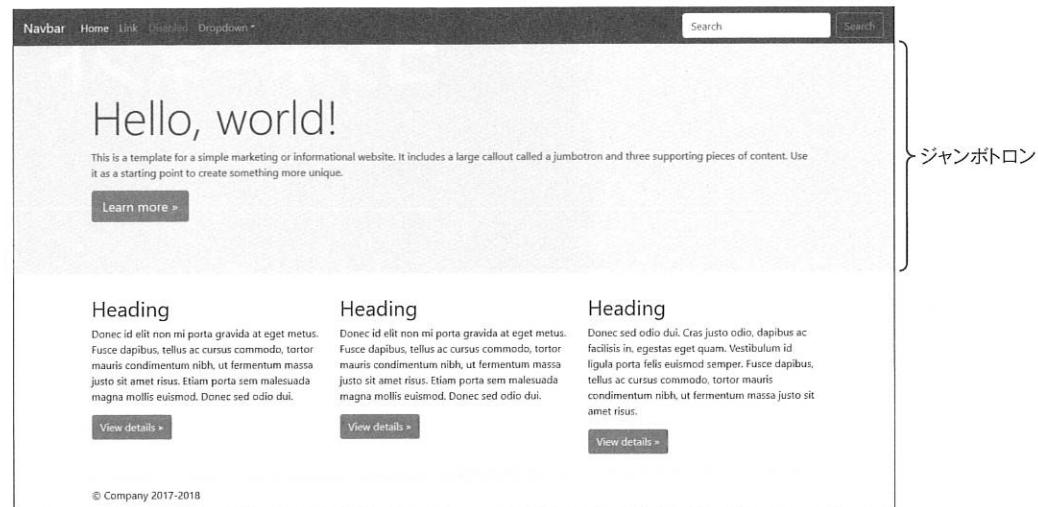
SECTION

## 1

## ジャンボトロン

Bootstrapには、Webサイトのトップページにおけるメインビジュアルなどの表示に便利なコンポーネントとして、**ジャンボトロン**(jumbotron)が用意されています(図4-1)。このコンポーネントは、親要素の幅いっぱいに広がるエリアを作成することができるため、一般的なコンテンツよりも大きく、インパクトのある表示をするのに最適です。

▼図4-1 Bootstrap公式サイトで公開されているジャンボトロンの例  
(<http://getbootstrap.com/docs/4.1/examples/>)



本節ではジャンボトロンを使用する方法を解説します。

### 4.1.1 基本的な使用例

ジャンボトロンを作成するには、div要素に**jumbotronクラス**を追加します。ジャンボトロンをメインビジュアルなどのエリアとして使う場合には、配置するテキストも大きく目立たせた方がバランスが良く見えます。次の例では、h1要素に見出しを目立たせるための**display-[ サイズ ] クラス**(P.50参照)を追加し、p要素にはテキストを目立たせるための**leadクラス**(P.52参照)を追加しています(リスト4-1、図4-2)。

## ▼リスト 4-1 ジャンボトロンの基本的な使用例 (jumbotron-basic.html)

```
<div class="container">
  <div class="jumbotron">
    <h1 class="display-3">Jumbotron</h1>
    <p class="lead">これはジャンボトロンのサンプルです。</p>
  </div>
</div>
```

▼図 4-2 ジャンボトロン適用前（左）、適用後（右）

**4.1.2 ジャンボトロンを全幅で表示する**

画面幅いっぱいのジャンボトロンを作成する場合は、**jumbotron** クラスを設定した div 要素に **jumbotron-fluid** クラスを追加します。ジャンボトロンは、固定幅コンテナ（P.23 参照）で使用されることを想定して角丸 (border-radius: 0.3rem;) になるよう設定されていますが、jumbotron-fluid クラスを追加されたジャンボトロンには角丸がなくなります（リスト 4-2、図 4-3）。

## ▼リスト 4-2 ジャンボトロンを全幅で表示する (jumbotron-fluid.html)

```
<div class="jumbotron jumbotron-fluid">
  <div class="container">
    <h1 class="display-3">Fluid jumbotron</h1>
    <p class="lead">これは全幅のジャンボトロンのサンプルです。</p>
  </div>
</div>
```

▼図 4-3 ジャンボトロンを全幅で表示する



Bootstrap の**アラート**は、エラーメッセージなど Web サイトからユーザーへのフィードバックメッセージを表示するためのコンポーネントです。本節では Bootstrap のアラートを使用する方法を解説します。

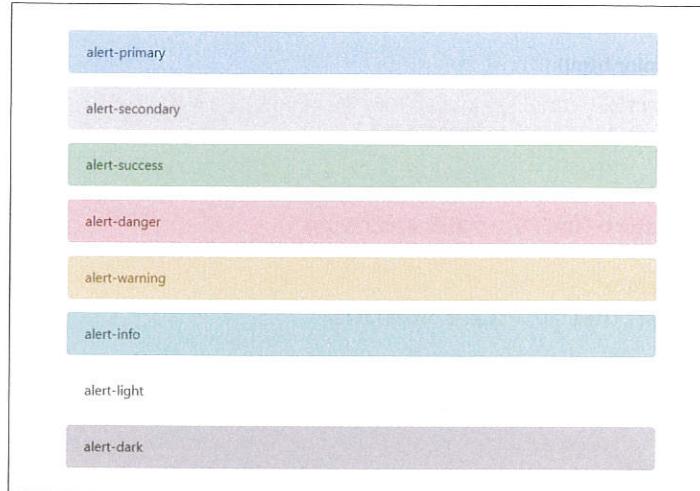
### 4.2.1 基本的な使用例

メッセージの枠となる親要素に **alert** クラスおよび **alert-{ 色の種類 } クラス** を追加することで、アラートを作成できます。色の種類には **primary** (青)、**secondary** (グレー) などコンテキストに対応した色の種類が入ります。またアクセシビリティへの配慮として属性 **role="alert"** を追加し、スクリーンリーダーなどの支援技術にこのコンポーネントの役割がアラートであることを伝えます (リスト 4-3、図 4-4)。

#### ▼リスト 4-3 アラートの基本的な使用例 (alerts-basic.html)

```
<div class="alert alert-primary" role="alert">
  alert-primary
</div>
<div class="alert alert-secondary" role="alert">
  alert-secondary
</div>
<div class="alert alert-success" role="alert">
  alert-success
</div>
<div class="alert alert-danger" role="alert">
  alert-danger
</div>
<div class="alert alert-warning" role="alert">
  alert-warning
</div>
<div class="alert alert-info" role="alert">
  alert-info
</div>
<div class="alert alert-light" role="alert">
  alert-light
</div>
<div class="alert alert-dark" role="alert">
  alert-dark
</div>
```

▼図 4-4 アラートの基本的な使用例



## NOTE

## コンテクストに対応した色の種類

Bootstrap では、**primary** は青、**secondary** はグレー、**success** は緑など、コンテクスト（文脈や意味）と対応した色の種類が、ソースファイル「\_variables.scss」（P.439 参照）に定義されており、CSS ファイル「bootstrap.css」（および「bootstrap.min.css」）内にコンパイルされています。アラートの他にも、バッジ（badge）やボタン（btn）などのコンポーネントの色を変更する場合は、{コンポーネント名}-{色の種類} クラスに表 4-1 の色の種類のいずれかを追加します。

▼表 4-1 コンテクストと色との対応

色の種類	コンテクスト	色
primary	主	青
secondary	副	グレー
success	成功	緑
info	情報	シアン
warning	警告	黄
danger	危険	赤
light	明	淡グレー
dark	暗	濃グレー

## 4.2.2 リンクの色

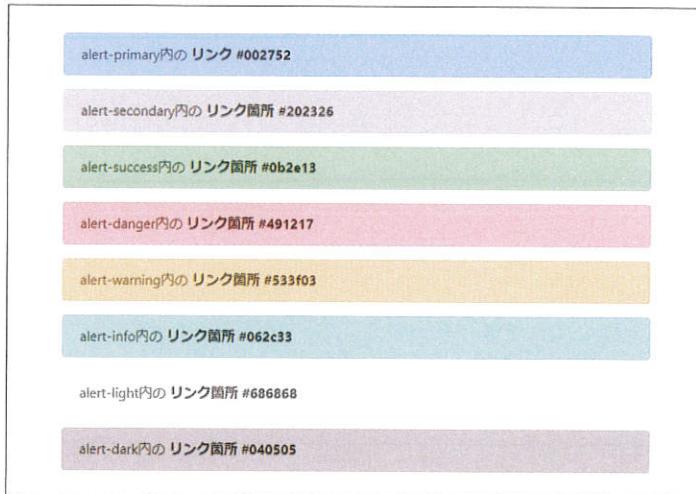
アラート内の a 要素に **alert-link** クラスを追加することで、リンク箇所の文字色を背景色に適した色や太さで

表示し、より読みやすくすることができます（リスト4-4、図4-5）。

▼リスト4-4 アラート内のリンク色（alerts-link-color.html）

```
<div class="alert alert-primary" role="alert">
  alert-primary内の <a href="#" class="alert-link">リンク #002752</a>
</div>
<div class="alert alert-secondary" role="alert">
  alert-secondary内の <a href="#" class="alert-link">リンク箇所 #202326</a>
</div>
<div class="alert alert-success" role="alert">
  alert-success内の <a href="#" class="alert-link">リンク箇所 #0b2e13</a>
</div>
<div class="alert alert-danger" role="alert">
  alert-danger内の <a href="#" class="alert-link">リンク箇所 #491217</a>
</div>
<div class="alert alert-warning" role="alert">
  alert-warning内の <a href="#" class="alert-link">リンク箇所 #533f03</a>
</div>
<div class="alert alert-info" role="alert">
  alert-info内の <a href="#" class="alert-link">リンク箇所 #062c33</a>
</div>
<div class="alert alert-light" role="alert">
  alert-light内の <a href="#" class="alert-link">リンク箇所 #686868</a>
</div>
<div class="alert alert-dark" role="alert">
  alert-dark内の <a href="#" class="alert-link">リンク箇所 #040505</a>
</div>
```

▼図4-5 アラート内のリンク色



1  
2  
3  
4  
5  
6  
7  
8  
9  
10

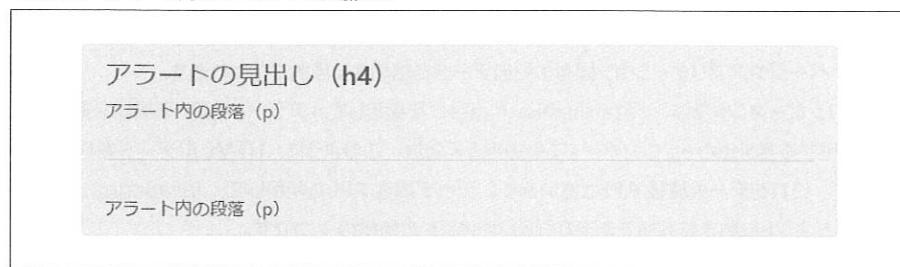
### 4.2.3 アラート内にコンテンツを追加する

アラート内には、見出し、段落、区切りなどさまざまなコンテンツを追加することができます。次の例では、`h4`要素に **alert-heading** クラスを追加して、アラートの見出しとして設定しています。なお `p` 要素に Spacing ユーティリティ (P.318 参照) の **mb-0** クラスを追加することで、下マージンを 0 にして余分な余白を削除しています (リスト 4-5、図 4-6)。

▼リスト 4-5 アラート内にコンテンツを追加する (alerts-content.html)

```
<div class="alert alert-success" role="alert">
  <h4 class="alert-heading">アラートの見出し (h4) </h4><!-- 見出し -->
  <p>アラート内の段落 (p) </p><!-- 段落 -->
  <hr><!-- 区切り -->
  <p class="mb-0">アラート内の段落 (p) </p><!-- 段落 -->
</div>
```

▼図 4-6 アラート内にコンテンツを追加する



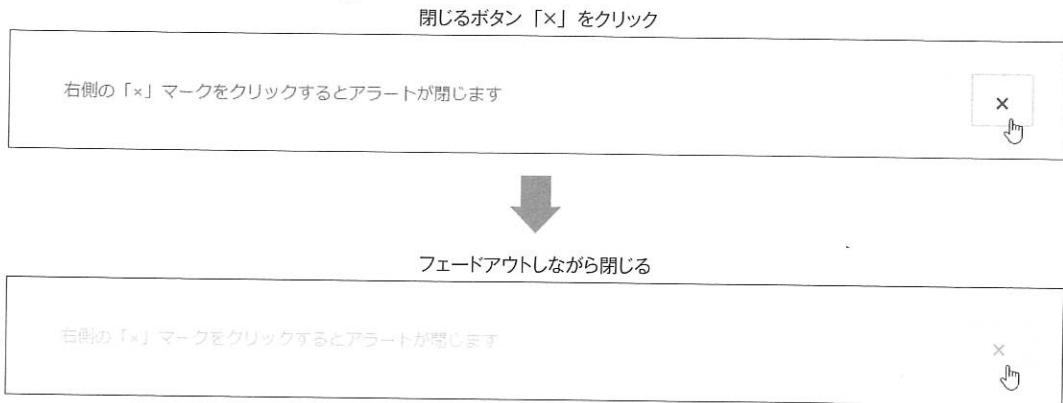
### 4.2.4 アラートを閉じる

表示されたアラートを閉じるには、`alert` クラスが設定された要素に、**alert-dismissible** クラスを追加します。アラートを閉じるボタンは、`button` 要素に **close** クラスを追加して作成します。また、JavaScript 経由でアラートを閉じる機能を有効化するために、`data-dismiss="alert"` を加えます (JavaScript の挙動については、第 7 章にて改めて解説します)。アラートを閉じる挙動をアニメーション化したい場合は、`alert` クラスが設定された要素に **fade** クラスおよび **show** クラスも追加します (リスト 4-6、図 4-7)。

▼リスト 4-6 アラートを閉じるボタンを追加する (alerts-close.html)

```
<div class="alert alert-warning alert-dismissible fade show" role="alert">
  右側の「×」マークをクリックするとアラートが閉じます
  <button type="button" class="close" data-dismiss="alert" aria-label="Close">
    <span aria-hidden="true">&times;</span>
  </button>
</div>
```

▼図4-7 アラート内にコンテンツを追加する



## NOTE

## 「data-」ではじまるデータ属性について

「data-」ではじまるデータ属性は、HTML5でカスタムデータ属性として定義されたものです。これは、適切な属性や要素がない場合に、ページやアプリケーション固有の独自データを格納する目的で使われます。

Bootstrapのアラートコンポーネントでは、`data-dismiss="alert"`を指定して、アラートを閉じる機能を追加しました。JavaScriptを利用するBootstrapコンポーネントのほとんどは、このように、HTMLにデータ属性を追加するだけで利用できます。これを**データ属性API**と言います。データ属性APIを利用して、JavaScriptコードを書かずに、簡単にコンポーネントに動きを追加できるのもBootstrapの特徴の1つです。

# 4 SECTION

## 3 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。

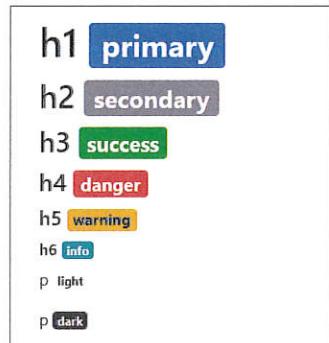
### 4.3.1 基本的な使用例

バッジを作成するには、span 要素に **badge** クラス、**badge-{ 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.111 参照) を入れて、バッジの背景色を設定します (リスト 4-7、図 4-8)。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調整されます。

▼リスト 4-7 バッジの基本的な使用例 (badge-basic.html)

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

▼図 4-8 バッジの基本的な使用例



### 4.3.2 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（P.115 参照）と同様に、ボタンやリンク内の span 要素に **badge** クラスおよび **badge-{ 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.233 参照）を追加した青色のボタン内に、**badge** クラスおよび **badge-light** 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」を表示するカウンターを作成しています（リスト 4-8、図 4-9）。

▼リスト 4-8 バッジでカウンターを作成する（badge-counter.html）

```
<button type="button" class="btn btn-primary">
  button <span class="badge badge-light">4</span>
</button>
```

▼図 4-9 バッジでカウンターを作成する



NOTE

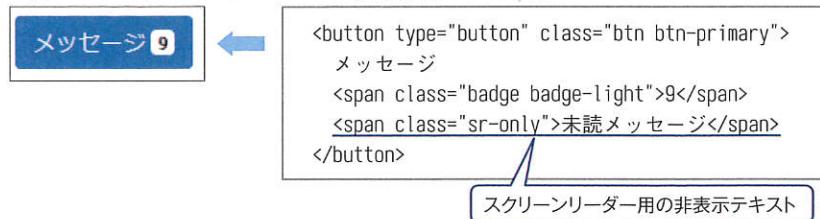
#### バッジのアクセシビリティ

バッジがラベルやカウンターの役割を果たしていることをスクリーンリーダーなどの支援技術に伝える場合は、テキストを非表示にするスクリーンリーダー用ユーティリティ（P.354 参照）の **sr-only** クラスを使って情報追加します。次の例では、バッジに表示された数字 9 が未読メッセージの数であることをスクリーンリーダーに伝えるための非表示テキストを加えています（リスト 4-9、図 4-10）。

▼リスト 4-9 バッジにスクリーンリーダー用の非表示テキストを加える（badge-sr-only.html）

```
<button type="button" class="btn btn-primary">
  メッセージ <span class="badge badge-light">9</span>
  <!-- sr-only クラスで非表示にしたスクリーンリーダー用テキスト -->
  <span class="sr-only">未読メッセージ</span>
</button>
```

▼図 4-10 バッジにスクリーンリーダー用の非表示テキストを加える



### 4.3.3 ピル型のバッジを作成する

丸みを帯びたピル型のバッジを作成する場合は、**badge** クラスおよび **badge-{ 色の種類 }** クラスが設定された要素に、**badge-pill** クラスを追加します。**badge-pill** クラスが追加されることで、角丸サイズと左右のパディングサイズが大きくなり、バッジが丸みを帯びます（リスト 4-10、図 4-11）。

▼リスト 4-10 ピル型のバッジを作成する（pill-badges.html）

```
<span class="badge badge-pill badge-primary">Primary</span>
<span class="badge badge-pill badge-secondary">Secondary</span>
<span class="badge badge-pill badge-success">Success</span>
<span class="badge badge-pill badge-danger">Danger</span>
<span class="badge badge-pill badge-warning">Warning</span>
<span class="badge badge-pill badge-info">Info</span>
<span class="badge badge-pill badge-light">Light</span>
<span class="badge badge-pill badge-dark">Dark</span>
```

▼図 4-11 ピル型のバッジを作成する



### 4.3.4 アクション付きのバッジを作成する

ホバーやフォーカスといったアクション付きのバッジを作成する場合、**a** 要素に **badge** クラスおよび **badge-{ 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）などコンテキストに対応した色の種類（P.111 参照）を入れて、バッジの背景色を設定します。次の例では、ホバー時に色が濃くなる緑色のバッジを作成しています（リスト 4-11、図 4-12）。

▼リスト 4-11 アクション付きのバッジを作成する（badge-links.html）

```
<a href="#" class="badge badge-success">badge-success</a>
```

▼図 4-12 アクション付きバッジの通常時（左）、アクション付きバッジのホバー時（右）



## 4

SECTION

# 4 プログレス

Bootstrap の**プログレス**は、プログレスバーを作成するためのコンポーネントです。**プログレスバー**とは、作業の経過・進捗状況を横長のバーで視覚的に示すもので、グラフィカルな UI の 1 つです。

Bootstrap では、シンプルなバー、ストライプ状のバー、アニメーション背景のバーなど、さまざまなプログレスバーを作成できるコンポーネントが用意されており、それらを積み重ねたり、テキストラベルを付けたりして使用することができます。本節では、Bootstrap のプログレスを使用する方法を解説します。

## 4.4.1 基本的な使用例

Bootstrap の**プログレス**でプログレスバーを作成するには、まず、コンポーネント全体を包括する要素に **progress** クラスを追加し、プログレスバーの最大値を示すブロックを作成します（リスト 4-12 ①）。次に、その子要素に **progress-bar** クラスを追加して、進捗状況を示すブロックを作成します（②）。このコンポーネントでは、HTML5 の progress 要素（進行状況を表す要素）は使用しません。

▼リスト 4-12 プログレスバーの基本の使い方 (progress-basic.html)

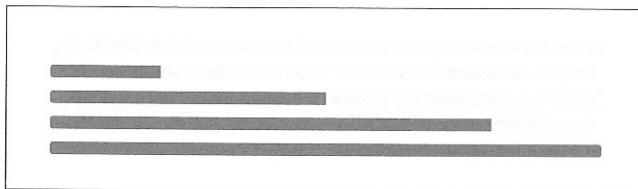
```
<div class="progress mb-3"> _____  
  <div class="progress-bar" role="progressbar" aria-valuenow="0" aria-valuemin="0" aria-valuemax="100"></div> _____  
</div>  
<div class="progress mb-3">  
  <div class="progress-bar" style="width: 20%" role="progressbar" aria-valuenow="25" aria-valuemin="0" aria-valuemax="100"></div>  
</div>  
<div class="progress mb-3">  
  <div class="progress-bar w-50" role="progressbar" aria-valuenow="50" aria-valuemin="0" aria-valuemax="100"></div>  
</div>  
<div class="progress mb-3">  
  <div class="progress-bar" style="width:80%" role="progressbar" aria-valuenow="75" aria-valuemin="0" aria-valuemax="100"></div>  
</div>  
<div class="progress">  
  <div class="progress-bar w-100" role="progressbar" aria-valuenow="100" aria-valuemin="0" aria-valuemax="100"></div>  
</div>
```

progress-bar クラスを指定した要素には、進捗状況を示すバーの長さ（幅）を指定する必要があります。style 属性を使用して style="width:20%"、style="width:80%" のように幅のスタイルを指定するか、Sizing ユーティリティ (P.314 参照) の w-{% 値} クラスを使用して 25%、50%、75%、100% の 4 段階で幅を指定します。

また、アクセシビリティへの配慮のために role 属性と aria-\* 属性の追加が必要です。属性 role="progressbar" を追加することで、スクリーンリーダーなどの支援技術にプログレスバーの役割であることを伝えます。aria-valuenow 属性には現在値、aria-valuemin には最小値、aria-valuemax 属性には最大値を指定し、同様の支援技術にプログレスバーの状態を伝えます。

なお、プログレスバーを複数並べる場合、バーと次のバーの間に余白がないので、progress クラスが設定された要素に Spacing ユーティリティ (P.318 参照) を追加して、バーの間隔を調整してあげると良いでしょう。図 4-13 の例では、最後のプログレスバー以外のバーに Spacing ユーティリティの mb-3 クラスを指定して、下マージンを設定しています。

▼図 4-13 基本のプログレスバー



#### 4.4.2 プログレスバーにテキストラベルを追加する

progress-bar クラスが設定された要素にテキストを配置すると、プログレスバーにテキストラベルが追加されます (リスト 4-13、図 4-14)。

▼リスト 4-13 プログレスバーにテキストラベルを追加する (progress-label.html)

```
<div class="progress">
  <div class="progress-bar w-25" role="progressbar" aria-valuenow="25" aria-valuemin="0" aria-valuemax="100">25%</div>
</div>
```

▼図 4-14 プログレスバーにテキストラベルを追加する



#### 4.4.3 プログレスバーの高さを変更する

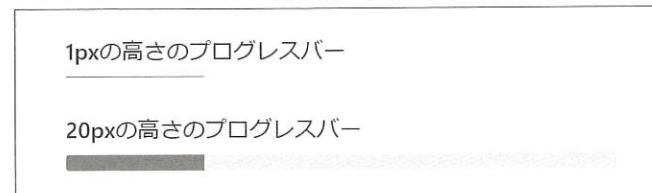
プログレスバーの高さ（太さ）は、初期設定として 1rem (= 16px) が定義されています。バーの高さを変

更するには、progress クラスを追加した要素に style 属性で height 値を設定します。その値を変更すると、内部の progress-bar クラスを指定した要素は自動的にそれに応じたサイズに変更されます（リスト 4-14、図 4-15）。

▼リスト 4-14 プログレスバーの高さを変更する（progress-height.html）

```
<h3 class="mb-3">1pxの高さのプログレスバー</h3>
<div class="progress mb-5" style="height: 1px;">
  <div class="progress-bar w-25" role="progressbar" aria-valuenow="25" aria-valuemin="0" aria-valuemax="100"></div>
</div>
<h3 class="mb-3">20pxの高さのプログレスバー</h3>
<div class="progress" style="height: 20px;">
  <div class="progress-bar w-25" role="progressbar" aria-valuenow="25" aria-valuemin="0" aria-valuemax="100"></div>
</div>
```

▼図 4-15 プログレスバーの高さを変更する



#### 4.4.4 プログレスバーの背景を変更する

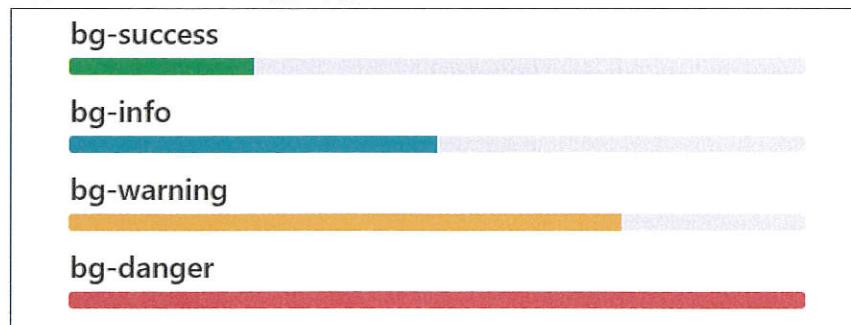
プログレスバーの外観を変更するには、progress-bar クラスが設定された要素に Color ユーティリティ（P.302 参照）の bg-{ 色の種類 } クラスを追加します。色の種類には **primary**（青）、**secondary**（グレー）などの色の種類があります。何も指定しない場合は、初期設定の bg-primary クラスが設定されて青色になります（リスト 4-15、図 4-16）。

▼リスト 4-15 プログレスバーの背景を変更する（progress-background.html）

```
<h3 class="mb-2">bg-success</h3>
<div class="progress mb-3">
  <div class="progress-bar bg-success" role="progressbar" style="width: 25%" aria-valuenow="25" aria-valuemin="0" aria-valuemax="100"></div>
</div>
<h3 class="mb-2">bg-info</h3>
<div class="progress mb-3">
  <div class="progress-bar bg-info" role="progressbar" style="width: 50%" aria-valuenow="50" aria-valuemin="0" aria-valuemax="100"></div>
</div>
<h3 class="mb-2">bg-warning</h3>
```

```
<div class="progress mb-3">
  <div class="progress-bar bg-warning" role="progressbar" style="width: 75%" aria-valuenow="75" ↪
    aria-valuemin="0" aria-valuemax="100"></div>
</div>
<h3 class="mb-2">bg-danger</h3>
<div class="progress mb-3">
  <div class="progress-bar bg-danger" role="progressbar" style="width: 100%" aria-valuenow="100" ↪
    aria-valuemin="0" aria-valuemax="100"></div>
</div>
```

▼図 4-16 プログレスバーの背景を変更する



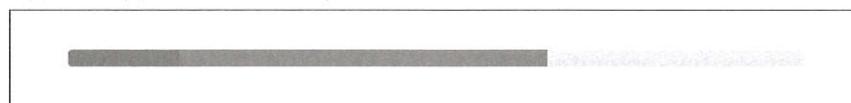
#### 4.4.5 複数のプログレスバーを重ねて表示する

必要に応じて、複数のプログレスバーを重ねて表示することができます。progress クラスが設定された要素の中に、progress-bar クラスを指定した要素を複数追加するだけで実装できます（リスト 4-16、図 4-17）。

▼リスト 4-16 複数のプログレスバーを積み重ねる（progress-multiple-bars.html）

```
<div class="progress">
  <div class="progress-bar" role="progressbar" style="width: 15%" aria-valuenow="15" ↪
    aria-valuemin="0" aria-valuemax="100"></div>
  <div class="progress-bar bg-success" role="progressbar" style="width: 30%" aria-valuenow="30" ↪
    aria-valuemin="0" aria-valuemax="100"></div>
  <div class="progress-bar bg-info" role="progressbar" style="width: 20%" aria-valuenow="20" ↪
    aria-valuemin="0" aria-valuemax="100"></div>
</div>
```

▼図 4-17 複数のプログレスバーを積み重ねる



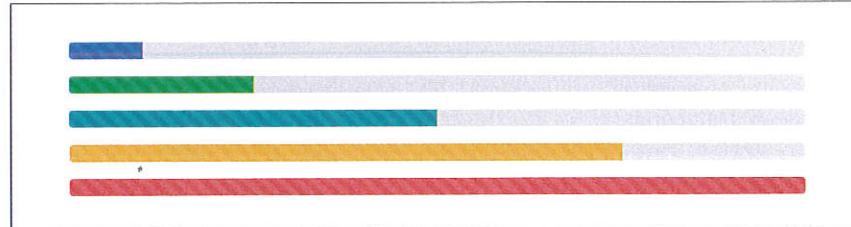
#### 4.4.6 プログレスバーをストライプにする

progress-bar クラスが設定された要素に、**progress-bar-striped クラス**を追加すると、プログレスバーの背景色に CSS グラデーションを使用したストライプの模様を適用することができます（リスト 4-17、図 4-18）。

▼リスト 4-17 プログレスバーをストライプにする（progress-stripe.html）

```
<div class="progress mb-3">
  <div class="progress-bar progress-bar-striped" role="progressbar" style="width: 10%" ↪
    aria-valuenow="10" aria-valuemin="0" aria-valuemax="100"></div>
</div>
<div class="progress mb-3">
  <div class="progress-bar progress-bar-striped bg-success" role="progressbar" style="width: 25%" ↪
    aria-valuenow="25" aria-valuemin="0" aria-valuemax="100"></div>
</div>
<div class="progress mb-3">
  <div class="progress-bar progress-bar-striped bg-info" role="progressbar" style="width: 50%" ↪
    aria-valuenow="50" aria-valuemin="0" aria-valuemax="100"></div>
</div>
<div class="progress mb-3">
  <div class="progress-bar progress-bar-striped bg-warning" role="progressbar" style="width: 75%" ↪
    aria-valuenow="75" aria-valuemin="0" aria-valuemax="100"></div>
</div>
<div class="progress">
  <div class="progress-bar progress-bar-striped bg-danger" role="progressbar" style="width: 100%" ↪
    aria-valuenow="100" aria-valuemin="0" aria-valuemax="100"></div>
</div>
```

▼図 4-18 プログレスバーをストライプにする



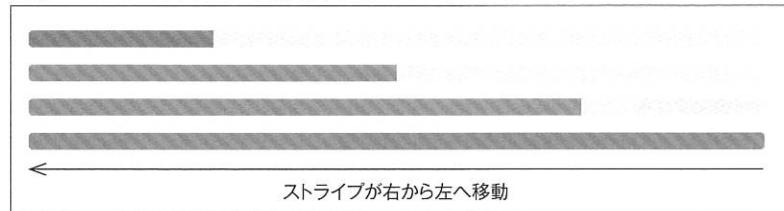
#### 4.4.7 プログレスバーのストライプをアニメーションにする

CSS グラデーションを使用したストライプの模様は、アニメーションで動かすこともできます。progress-bar クラスが設定された要素に **progress-bar-animated クラス**を追加すると、CSS3 アニメーションを使用してストライプが右から左に動くアニメーションを表示できます。ただし、CSS3 のアニメーションをサポートしていない Opera 12などのブラウザでは動作しません（リスト 4-18、図 4-19）。

## ▼リスト 4-18 プログレスバーのストライプをアニメーションさせる (progress-animation.html)

```
<div class="progress mb-3">
  <div class="progress-bar progress-bar-striped progress-bar-animated" role="progressbar" ↪
    aria-valuenow="25" aria-valuemin="0" aria-valuemax="100" style="width: 25%"></div>
</div>
<div class="progress mb-3">
  <div class="progress-bar progress-bar-striped progress-bar-animated bg-success" ↪
    role="progressbar" aria-valuenow="50" aria-valuemin="0" aria-valuemax="100" style="width: 50%"></div>
</div>
<div class="progress mb-3">
  <div class="progress-bar progress-bar-striped progress-bar-animated bg-info" ↪
    role="progressbar" aria-valuenow="75" aria-valuemin="0" aria-valuemax="100" style="width: 75%"></div>
</div>
<div class="progress mb-3">
  <div class="progress-bar progress-bar-striped progress-bar-animated bg-danger" ↪
    role="progressbar" aria-valuenow="100" aria-valuemin="0" aria-valuemax="100" style="width: 100%"></div>
</div>
</div>
```

## ▼図 4-19 プログレスバーのストライプをアニメーションさせる



1  
2  
3  
4  
5  
6  
7  
8  
9  
10

4

SECTION

5

# カード

Bootstrap の **カード** は、枠で囲まれたひとまとまりのコンテンツを作成するためのコンポーネントです。コンポーネント内には画像、テキスト、リストグループ、リンクなど、さまざまなコンテンツを含めることができます。またこのコンポーネントは flexbox をもとに構築されているため、レイアウトや他のコンポーネントとの組み合わせが容易です。本節では Bootstrap のカードを使用する方法を解説します。

なお、カードは Bootstrap 4 以降のコンポーネントで、Bootstrap 3 における **パネル** (Panels)、**サムネイル** (Thumbnails)、**ウェル** (Wells) が統廃合されたものです。

## 4.5.1 基本的な使用例

カードを作成するには、コンポーネント全体を包括する要素に **card クラス** を、枠の中に含まれる子要素に表 4-2 のクラスを追加します。

▼表 4-2 カード枠内の子要素に使用される主なクラス

クラス	概要
card-body	カードの本文エリアを作成
card-title	h 要素に追加し、カードの見出しを作成
card-subtitle	h 要素に追加し、カードの副見出しを作成
card-text	p 要素などのテキスト要素に追加し、カードのテキストを作成
card-link	a 要素に追加し、カードのリンクを作成
card-img-top	img 要素に追加し、カード上部に画像を配置
card-img-bottom	img 要素に追加し、カード下部に画像を配置
card-header	カードのヘッダーエリアを作成
card-footer	カードのフッターエリアを作成

それではカードの基本的な使用例を見てみましょう（リスト 4-19、図 4-20）。

▼リスト 4-19 基本的な使用例（card-basic.html）

```
<!-- カードの枠 : card -->
<div class="card" style="max-width: 25rem;">_____  

<!-- カード上部にレイアウトされる画像 : card-img-top -->
  _____  

<!-- カード本文 : card-body -->
  <div class="card-body">_____  

    <!-- カード見出し : card-title -->
```

```

<h4 class="card-title">カードの見出し</h4>-----④
<!-- カードの副見出し : card-subtitle -->
<h6 class="card-subtitle">カードの副見出し</h6>-----⑤
<!-- カードの内容文 : card-text -->
<p class="card-text">カードの内容文が入ります。</p>-----⑥
<a href="#" class="btn btn-primary">ボタン</a>
</div>
</div>

```

まずカードの枠を形作る親要素に **card クラス** を追加します（①）。カード上部にレイアウトされる画像には **img** 要素に **card-img-top クラス** を追加し（②）、カードの本文エリアとなる要素には **card-body** を追加します（③）。カードの見出しどころの要素には **card-title クラス** を追加し（④）、副見出しどころの要素には **card-subtitle** 追加します（⑤）。内容文となる要素には **card-text クラス** を追加します（⑥）。

なお、このコンポーネントには幅サイズが初期設定されていません。例では **card クラス** が設定された **div** 要素の **style** 属性に **max-width: 25rem** を追加して、幅サイズを 25rem 以下に設定しています（⑦）。

▼図 4-20 カードの基本的な使用例



## ■ カードに画像を配置する

カード内の **img** 要素に **card-img-top クラス** を追加してカード上部に画像を配置する方法は既に述べましたが、**img** 要素に **card-img-bottom クラス** を追加することでカード下部に画像を配置することもできます（リスト 4-20、図 4-21）。

▼リスト 4-20 カード内に画像を配置する（card-image.html）

```

<!-- 画像：上部 -->
<div class="card mb-3" style="max-width: 25rem;">
  <!-- カード上部に画像を配置 : card-img-top -->
  
  <div class="card-body">
    <p class="card-text">カードの内容文が入ります。</p>
  </div>
</div>

<!-- 画像：下部 -->

```

```
<div class="card" style="width: max-width: 25rem;">
  <div class="card-body">
    <p class="card-text">カードの内容文が入ります。</p>
  </div>
  <!-- カード下部に画像を配置 : card-img-bottom -->
  
</div>
```

▼図 4-21 カード内の画像配置



## ■ カードにリストグループを組み込む

カード内にリストグループ（P.180 参照）を組み込む場合、**list-group** クラスが設定された要素に **list-group-flush** クラスを追加します（リスト 4-21、図 4-22）。

▼リスト 4-21 カードにリストグループを組み込む (card-list-group.html)

```
<div class="card" style="width: max-width: 25rem;">
  <!-- カード内のリストグループ: list-group-flush -->
  <ul class="list-group list-group-flush">
    <li class="list-group-item">リスト01</li>
    <li class="list-group-item">リスト02</li>
    <li class="list-group-item">リスト03</li>
  </ul>
</div>
```

▼図 4-22 カード内のリストグループ



## カード内にコンテンツを複合する

カード内には複数のコンテンツを複合することができます。次の例では、カード内に画像、テキスト、リストグループなどを複合しています（リスト 4-22、図 4-23）。

▼リスト 4-22 カード内にコンテンツを複合する（card-kitchen-sink.html）

```
<div class="card" style="max-width: 25rem;">
  <!-- カード上部に画像を配置 : card-img-top -->
  
  <!-- カードの本文 : card-body -->
  <div class="card-body">
    <!-- カードの見出し : card-title -->
    <h4 class="card-title">カードの見出し</h4>
    <!-- カードの内容文 : card-text -->
    <p class="card-text">カードの内容文が入ります。</p>
  </div>
  <!-- カード内のリストグループ : list-group-flush -->
  <ul class="list-group list-group-flush">
    <li class="list-group-item">リスト01</li>
    <li class="list-group-item">リスト02</li>
    <li class="list-group-item">リスト03</li>
  </ul>
  <!-- カードの本文 : card-body -->
  <div class="card-body">
    <!-- カード内リンク : card-link -->
    <a href="#" class="card-link">カード内リンク</a>
    <a href="#" class="card-link">カード内リンク</a>
  </div>
</div>
```

▼図 4-23 カード内にコンテンツを複合する



1  
2  
3  
4  
5  
6  
7  
8  
9  
10

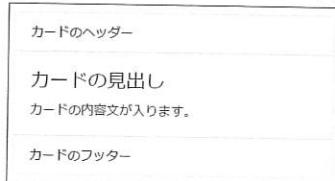
## ■ カードのヘッダーとフッターを作成する

カード内にヘッダーやフッターのエリアを作成するには、ヘッダーエリアとなる要素に **card-header クラス**、フッターエリアとなる要素に **card-footer クラス**を追加します（リスト 4-23、図 4-24）。

▼リスト 4-23 カードのヘッダーとフッターを作成する（card-header-footer.html）

```
<div class="card" style="max-width: 25rem;">
  <!-- カードのヘッダー : card-header -->
  <div class="card-header">
    カードのヘッダー
  </div>
  <!-- カードの本文 -->
  <div class="card-body">
    <h4 class="card-title">カードの見出し</h4>
    <p class="card-text">カードの内容文が入ります。</p>
  </div>
  <!-- カードのフッター : card-footer -->
  <div class="card-footer">
    カードのフッター
  </div>
</div>
```

▼図 4-24 カード内のヘッダーとフッターを作成する

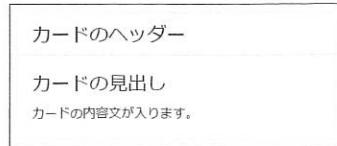


カード内の見出し要素に **card-header クラス**を追加して、見出し要素をヘッダーエリアとして設定することもできます（リスト 4-24、図 4-25）。

▼リスト 4-24 見出しに card-header クラスを使用する（card-header-h.html）

```
<div class="card" style="max-width: 25rem;">
  <!-- カードのヘッダー -->
  <h4 class="card-header">
    カードのヘッダー
  </h4>
  <!-- カードの本文 -->
  <div class="card-body">
    <h4 class="card-title">カードの見出し</h4>
    <p class="card-text">カードの内容文が入ります。</p>
  </div>
</div>
```

▼図 4-25 カード見出しに card-header クラスを使用する



## 4.5.2 カードのスタイルを変更する

カードには、背景、ボーダー、色をカスタマイズするためのさまざまなオプションがあります。

### カードの背景色や文字色を設定する

カードの色を変更するために、Color ユーティリティ (P.302 参照) の **text-{ 色の種類 } クラス** や **bg-{ 色の種類 } クラス** を使用することができます。色の種類には **primary** (青)、**secondary** (グレー) などが入ります。

次の例では、**text-white クラス** で文字色を白に、**bg-primary クラス** で背景色を青に、**bg-success クラス** で背景色を緑に調整しています (リスト 4-25、図 4-26)。

▼リスト 4-25 カードの背景色や文字色を設定する (card-background-color.html)

```
<!-- 背景色、文字色指定なし -->
<div class="card mb-3" style="max-width: 25rem;">
  <div class="card-header">カードのヘッダー</div>
  <div class="card-body">
    <h4 class="card-title">カードの見出し</h4>
    <p class="card-text">カードの内容文が入ります。</p>
  </div>
</div>

<!-- 背景色 : bg-primary、文字色 : text-white -->
<div class="card text-white bg-primary mb-3" style="max-width: 25rem;">
  ...中略...
</div>

<!-- 背景色 : bg-secondary、文字色 : text-white -->
<div class="card text-white bg-secondary mb-3" style="max-width: 25rem;">
  ...中略...
</div>

<!-- 背景色 : bg-success、文字色 : text-white -->
<div class="card text-white bg-success mb-3" style="max-width: 25rem;">
  ...中略...
</div>

<!-- 背景色 : bg-danger、文字色 : text-white -->
<div class="card text-white bg-danger mb-3" style="max-width: 25rem;">
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

```
…中略…  
/>
```

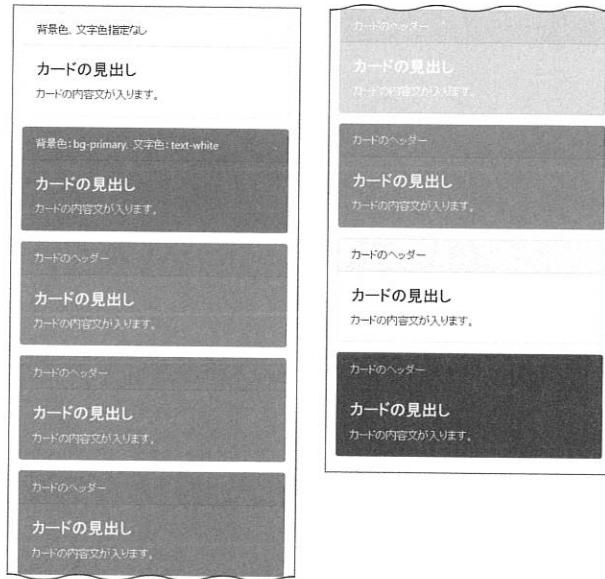
```
<!-- 背景色 : bg-warning、文字色 : text-white -->  
<div class="card text-white bg-warning mb-3" style="max-width: 25rem;">  
…中略…  
</div>
```

```
<!-- 背景色 : bg-info、文字色 : text-white -->  
<div class="card text-white bg-info mb-3" style="max-width: 25rem;">  
…中略…  
</div>
```

```
<!-- 背景色 : bg-light -->  
<div class="card bg-light mb-3" style="max-width: 25rem;">  
…中略…  
</div>
```

```
<!-- 背景色 : bg-dark、文字色 : text-white -->  
<div class="card text-white bg-darkmb-3" style="max-width: 25rem;">  
…中略…  
</div>
```

▼図 4-26 カードの背景色や文字色を設定する



## ■ カードのボーダー色を設定する

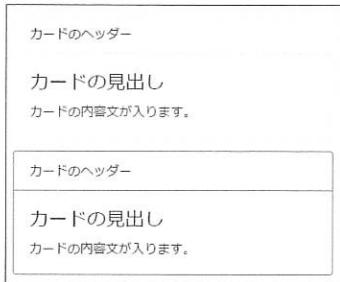
カードのボーダー色を設定するために、Border ユーティリティ（P.306 参照）の **border-{ 色の種類 } クラス**を使用することができます。色の種類には **primary**（青）、**secondary**（グレー）などが入ります。次の例では、card クラスが設定されたカードの枠となる要素に **border-primary クラス**を追加し、ボーダー色を青色に設定しています。また card-header クラスが設定されたカードのヘッダーエリアとなる要素にも **border-primary クラス**を追加し、カード内のボーダー色も青色に設定しています（リスト 4-26、図 4-27）。

▼リスト 4-26 カードのボーダー色を変更する（card-border.html）

```
<!-- カード枠のボーダー色：変更なし -->
<div class="card mb-3" style="max-max-width: 25rem">
    <!-- ヘッダーのボーダー色：変更なし -->
    <div class="card-header">カードのヘッダー</div>
    <div class="card-body">
        <h4 class="card-title">カードの見出し</h4>
        <p class="card-text">カードの内容文が入ります。</p>
    </div>
</div>

<!-- カード枠のボーダー色：青 border-primary -->
<div class="card border-primary mb-3" style="max-max-width: 25rem">
    <!-- ヘッダーのボーダー色：青 border-primary -->
    <div class="card-header border-primary">カードのヘッダー</div>
    <div class="card-body">
        <h4 class="card-title">カードの見出し</h4>
        <p class="card-text">カードの内容文が入ります。</p>
    </div>
</div>
```

▼図 4-27 カードのボーダー色を変更する



## ■ カードの背景色を除去する

カードのヘッダーやフッターには、初期設定で明るいグレーの背景色が設定されています。この背景色を除去する場合は、カードのヘッダーとして card-header クラスが設定された要素や、フッターとして card-footer 要素が設定された要素に、**bg-transparent クラス**を追加します（リスト 4-27、図 4-28）。

4  
5  
6  
7  
8  
9  
10

## ▼リスト4-27 カードの背景色を除去する (card-bg-transparent.html)

```
<div class="card mb-3" style="max-width: 25rem;">
  <!-- ヘッダーの背景色 -->
  <div class="card-header">ヘッダー初期設定の背景色</div>
  <div class="card-body">
    <h4 class="card-title">カードの見出し</h4>
    <p class="card-text">カードの内容文が入ります。</p>
  </div>
  <!-- フッターの背景色 -->
  <div class="card-footer">フッター初期設定の背景色</div>
</div>

<div class="card mb-3" style="max-width: 25rem;">
  <!-- ヘッダー背景色の除去 : bg-transparent -->
  <div class="card-header bg-transparent">背景色の除去 : bg-transparent</div>
  <div class="card-body">
    <h4 class="card-title">カードの見出し</h4>
    <p class="card-text">カードの内容文が入ります。</p>
  </div>
  <!-- フッター背景色の除去 : bg-transparent -->
  <div class="card-footer bg-transparent">背景色の除去 : bg-transparent</div>
</div>
```

## ▼図4-28 カードの背景色を除去する



### 4.5.3 カードのサイズを変更する

既に述べたように、カードには幅サイズが初期設定されていません。したがって幅サイズを設定しない場合は、コンポーネントの幅が親要素の100%のサイズになります。カードの幅サイズの設定を行うためには、style属性やCSSのwidthプロパティで幅を指定する方法の他に、グリッドレイアウトにカードを組み込んでレイアウトを行う方法や、カードにSizingユーティリティ(P.314参照)の**w-{% 値} クラス**を使用して幅指定を行う方法があります。

## ■ グリッドレイアウトに組み込む方法

card クラスが設定された要素をグリッドレイアウト（P.22 参照）の中に組み込んで、グリッドのカラムとして幅指定を行います。次の例では、**col-sm-6 クラス**が設定された div 要素の中にカードを組み込んで、画面幅 sm 以上で 6 列カラムが 2 つ並ぶレイアウトになるように幅を設定しています（リスト 4-28、図 4-29）。

▼リスト 4-28 グリッドレイアウトに組み込んで幅を指定する（card-grid.html）

```
<div class="row">
  <!-- カラム01 -->
  <div class="col-sm-6">
    <div class="card">
      <div class="card-body">
        …中略…
      </div>
    </div>
  </div>
  <!-- カラム02 -->
  <div class="col-sm-6">
    <div class="card">
      <div class="card-body">
        …中略…
      </div>
    </div>
  </div>
</div>
```

▼図 4-29 グリッドレイアウトに組み込んで幅を指定する



## ■ Sizing ユーティリティを使用する方法

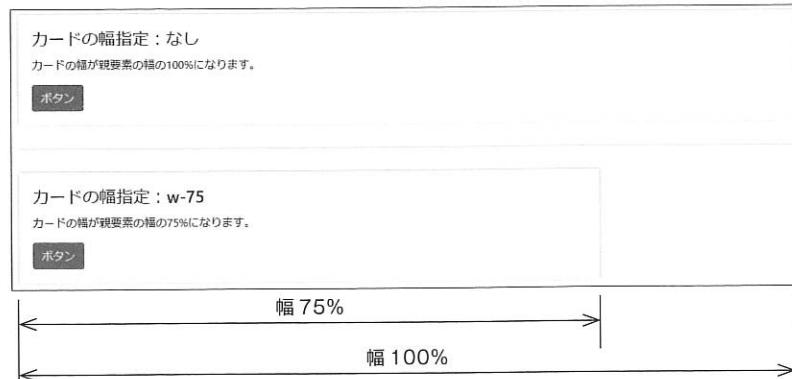
card クラスが設定された要素に Sizing ユーティリティ（P.314 参照）の **w-{% 値} クラス** を追加して幅指定を行います。% 値には、25、50、75、100 が入ります。次の例では **card クラス** が設定された要素に **w-75 クラス** を追加して、カードの幅が親要素の幅の 75 % になるように設定しています（リスト 4-29、図 4-30）。

▼リスト 4-29 Sizing ユーティリティを使用して幅を指定する（card-sizing.html）

```
<!-- Sizingユーティリティ : w-{%値} クラスによる幅指定 -->
<div class="card w-75">
  <div class="card-body">
    <h4 class="card-title">カードの幅指定 : w-75</h4>
```

```
<p class="card-text">カードの幅が親要素の幅の75%になります。</p>
<a href="#" class="btn btn-primary">ボタン</a>
</div>
</div>
```

▼図 4-30 Sizing ユーティリティを使用して幅を指定する



#### 4.5.4 カードのテキストを整列する

カード内のテキストを整列する場合は、`card` クラスが設定された要素に Text ユーティリティ (P.347 参照) の **text-right クラス** (右寄せ)、**text-center クラス** (中央揃え) などを追加します。初期設定では左揃えになっています (リスト 4-30、図 4-31)。

▼リスト 4-30 カードのテキストを整列する (card-text-align.html)

```
<!-- 初期設定 (左揃え) -->
<div class="card" style="max-width: 25rem">
  <div class="card-body">
    <h4 class="card-title">カードの見出し</h4>
    <p class="card-text">カードの内容文が入ります。</p>
    <a href="#" class="btn btn-primary">ボタン</a>
  </div>
</div>

<!-- 中央揃え : text-center -->
<div class="card text-center" style="max-width: 25rem">
  <div class="card-body">
    <h4 class="card-title">カードの見出し</h4>
    <p class="card-text">カードの内容文が入ります。</p>
    <a href="#" class="btn btn-primary">ボタン</a>
  </div>
</div>

<!-- 右揃え : text-right -->
```

```
<div class="card text-right" style="max-width: 25rem">
  <div class="card-body">
    <h4 class="card-title">カードの見出し</h4>
    <p class="card-text">カードの内容文が入ります。</p>
    <a href="#" class="btn btn-primary">ボタン</a>
  </div>
</div>
```

▼図 4-31 カードのテキストを整列する



### 4.5.5 カードにナビゲーションを組み込む

Bootstrap のナビゲーション (P.150 参照) をカードのヘッダー内に組み込むことができます。

#### タブ型のナビゲーション

カードのヘッダー内にタブ型ナビゲーション (P.153 参照) を作成します。カードのヘッダー内に、**nav クラス** および **nav-tabs クラス** が設定された **ul** 要素を配置し、**card-header-tabs クラス** を追加します (リスト 4-31、図 4-32)。

▼リスト 4-31 タブ型のナビゲーション (card-nav-tabs.html)

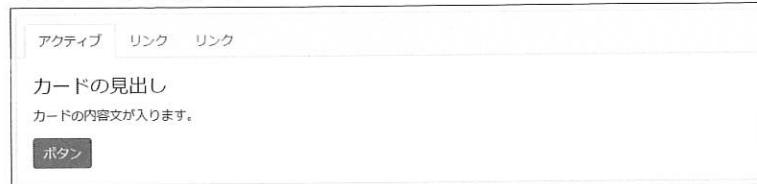
```
<div class="card">
  <!-- カードのヘッダー -->
  <div class="card-header">
    <!-- タブ型のナビゲーション : card-header-tabs -->
    <ul class="nav nav-tabs card-header-tabs">
      <li class="nav-item">
        <a class="nav-link active" href="#">アクティブ</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">リンク</a>
      </li>
    </ul>
  </div>
</div>
```

```

</li>
<li class="nav-item">
  <a class="nav-link" href="#">リンク</a>
</li>
</ul>
</div>
<!-- カードの本文 --&gt;
&lt;div class="card-body"&gt;
  ...中略...
&lt;/div&gt;
&lt;/div&gt;
</pre>

```

▼図 4-32 タブ型のナビゲーション



## ■ ピル型のナビゲーション

カードのヘッダー内にピル型ナビゲーション（P.154 参照）を作成します。カードのヘッダー内に、**nav** および **nav-pills** が設定された **ul** 要素を配置し、**card-header-pills** クラスを追加します（リスト 4-32、図 4-33）。

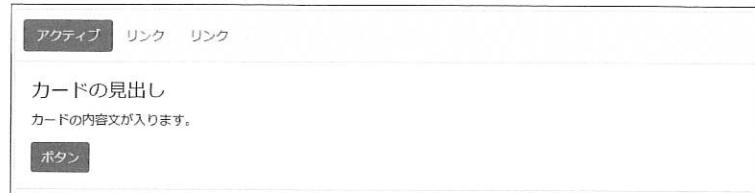
▼リスト 4-32 ピル型のナビゲーション（card-nav-pills.html）

```

<div class="card">
  <!-- カードのヘッダー --&gt;
  &lt;div class="card-header"&gt;
    <!-- ピル型のナビゲーションcard-header-pills --&gt;
    &lt;ul class="nav nav-pills card-header-pills"&gt;
      &lt;li class="nav-item"&gt;
        &lt;a class="nav-link active" href="#"&gt;アクティブ&lt;/a&gt;
      &lt;/li&gt;
      &lt;li class="nav-item"&gt;
        &lt;a class="nav-link" href="#"&gt;リンク&lt;/a&gt;
      &lt;/li&gt;
      &lt;li class="nav-item"&gt;
        &lt;a class="nav-link" href="#"&gt;リンク&lt;/a&gt;
      &lt;/li&gt;
    &lt;/ul&gt;
  &lt;/div&gt;
  <!-- カードの本文 --&gt;
  &lt;div class="card-body"&gt;
    ...中略...
  &lt;/div&gt;
&lt;/div&gt;
</pre>

```

▼図 4-33 ピル型のナビゲーション



1  
2  
3  
4  
5  
6  
7  
8  
9  
10

### 4.5.6 カードの画像とテキストを重ね合わせる

カード内のテキストブロックに **card-img-overlay** クラスを追加し、**card-img** クラスが設定された画像の上にテキストブロックを重ね合わせることができます（リスト 4-33、図 4-34）。

▼リスト 4-33 カードの画像とテキストを重ね合わせる（card-img-overlay.html）

```
<div class="card">
  
  <!-- 画像と重ね合わせるテキストブロック : card-img-overlay -->
  <div class="card-img-overlay">
    <h4 class="card-title">カードの見出し</h4>
    <p class="card-text">カードの内容文が入ります。</p>
  </div>
</div>
```

▼図 4-34 カードの画像とテキストを重ね合わせる



### 4.5.7 カードをレイアウトする

Bootstrap のカードには、**カードグループ**、**カードデッキ**、**カードカラム**といった一連のカードをレイアウトするオプションのコンポーネントが用意されています。

## カードグループによるレイアウト

カードグループは、一連のカードを連結して均一サイズでグループ化するコンポーネントです。カードグループを作成する場合は、一連のカードの親要素に **card-group クラス** を追加します（リスト 4-34 ①）。**card-group クラス** のスタイルには、flexbox のレイアウトを指定する **display:flex** が定義されています。また、カードグループ内の各カードのフッターとなる要素に **card-footer** を追加（②）すると、一連のフッターの縦位置が自動的に調整され、横並びに整列します（図 4-35）。

▼リスト 4-34 カードグループによるレイアウト（card-card-group.html）

```
<!-- カードグループ: card-group -->
<div class="card-group"> _____ ①
  <!-- カード01 -->
  <div class="card">
    
    <div class="card-body">
      ...中略...
    </div>
    <div class="card-footer"> _____
      ...中略...
    </div>
  </div>

  <!-- カード02 -->
  <div class="card">
    
    <div class="card-body">
      ...中略...
    </div>
    <div class="card-footer"> _____
      ...中略...
    </div>
  </div>

  <!-- カード03 -->
  <div class="card">
    
    <div class="card-body">
      ...中略...
    </div>
    <div class="card-footer"> _____
      ...中略...
    </div>
  </div>
```

▼図 4-35 カードグループによるレイアウト



1  
2  
3  
4  
5  
6  
7  
8  
9  
10

## ■ カードデッキによるレイアウト

**カードデッキ**は、一連のカードを連結せずに均一サイズでグループ化するコンポーネントです。カードデッキを作成する場合は、一連のカードの親要素に **card-deck** クラスを追加します（リスト 4-35 ①）。カードグループと同様、flexbox によるレイアウトが有効になり、一連のフッターの縦位置も自動調整されます（図 4-36）。

▼リスト 4-35 カードデッキによるレイアウト（card-card-deck.html）

```
<!-- カードデッキ : card-deck -->
<div class="card-deck"> ①

<!-- カード01 -->
<div class="card">
  
  <div class="card-body">
    ...中略...
  </div>
  <div class="card-footer">
    ...中略...
  </div>
</div>

<!-- カード02 -->
<div class="card">
  
  <div class="card-body">
    ...中略...
  </div>
  <div class="card-footer">
    ...中略...
  </div>
</div>

<!-- カード03 -->
<div class="card">
```

```

<div class="card-body">
    ...
</div>
<div class="card-footer">
    ...
</div>
</div>
```

▼図 4-36 カードデッキによるレイアウト



## ■ カードカラムによるレイアウト

**カードカラム**は、カードをタイル状の列にレイアウトするコンポーネントです。カードカラムを作成する場合は、一連のカードの親要素に **card-columns クラス** を追加します（リスト 4-36 ①）。カードカラムは、画面幅 sm 以上でカラム数が 3 になるように初期設定されています。カードの順序は上から下へ、左から右へ並べられます（図 4-37）。

▼リスト 4-36 カードカラムによるレイアウト（card-column.html）

```
<div class="card-columns"> ①
    <!-- カード01 -->
    <div class="card">
        ...
    </div>

    <!-- カード02 -->
    <div class="card p-3">
        ...
    </div>

    <!-- カード03 -->
    <div class="card">
```

```

...中略...
</div>

<!-- カード04 -->
<div class="card bg-secondary text-white text-center p-3">
...中略...
</div>

<!-- カード05 -->
<div class="card text-center">
...中略...
</div>

<!-- カード06 -->
<div class="card">
...中略...
</div>

<!-- カード07 -->
<div class="card p-3 text-right">
...中略...
</div>

<!-- カード08 -->
<div class="card">
...中略...
</div>

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

▼図 4-37 カードカラムによるレイアウト



Bootstrap のカードは flexbox を基本に構築されていますが、このカードカラムによるレイアウトは flexbox ではなく、段組カラム数を指定する CSS の **column-count プロパティ**によって定義されています。

カードカラムを作成する **card-columns クラス**には、リスト 4-37 のスタイルのように、Webkit 系ブラウザ（Chrome、Edge、Internet Explorer、Opera、Safari）およびその他のブラウザでの段組のカラム数が 3 になるよう定義されています（①②）。また、カラム同士の間隔が 1.25rem になるように定義されています（③④）。

▼リスト 4-37 カードカラムの card-columns クラスに定義されているスタイル

```
.card-columns .card {  
    margin-bottom: 0.75rem;  
}  
  
@media (min-width: 576px) {  
    .card-columns {  
        -webkit-column-count: 3; ①  
        column-count: 3; ②  
        -webkit-column-gap: 1.25rem; ③  
        column-gap: 1.25rem; ④  
    }  
    .card-columns .card {  
        display: inline-block;  
        width: 100%;  
    }  
}
```

4

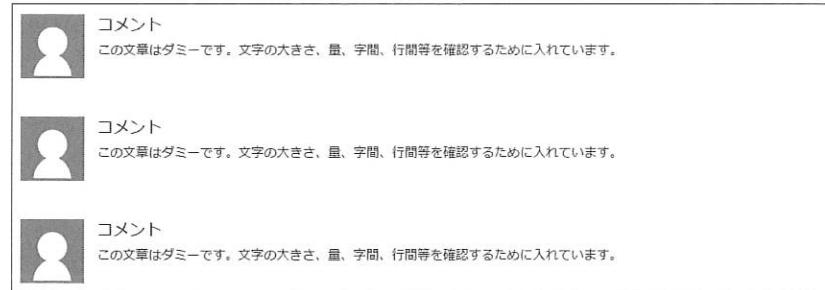
SECTION

6

# メディアオブジェクト

Bootstrap の**メディアオブジェクト**は、横並びの「画像」と「メディア本文」で構成されます。たとえばブログやTwitterのコメント欄のように、投稿者のアイコン画像とコメントのセットを繰り返して一覧表示させるようなレイアウトを行う場合には、このコンポーネントを使用すると便利です（図4-38）。

▼図4-38 メディアオブジェクトの例



## 4.6.1 基本的な使用例

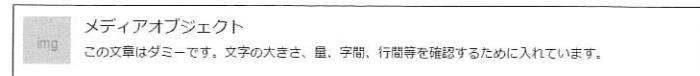
まず、div要素に**mediaクラス**を追加してメディアオブジェクトの外枠を作成します（リスト4-38①）。次に、メディア本文となる要素に**media-bodyクラス**を追加し、画像（img要素）とともにメディアオブジェクトの子要素として配置します（②）。

**mediaクラス**のスタイルには、flexboxのレイアウトを指定する**display:flex**が定義されています。そのため、メディアオブジェクトの子要素である画像とメディア本文は横並びにレイアウトされます（図4-39）。

▼リスト4-38 メディアオブジェクトの基本的な使用例（media-object-basic.html）

```
<!-- メディアオブジェクト -->
<div class="media"> _____❶
  <!-- 画像 -->
  
  <!-- メディア本文 -->
  <div class="media-body"> _____❷
    <h5>…</h5>
    …中略…
  </div>
</div>
```

▼図4-39 メディアオブジェクトの基本的な使用例



なお先の例では、img要素にSpacingユーティリティ(P.318参照)の**mr-3クラス**を追加することで、画像とメディア本文とのマージンサイズを調整しています。

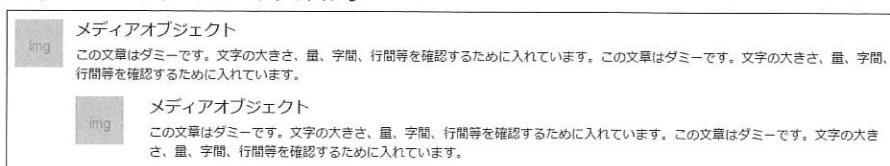
## 4.6.2 メディアオブジェクトの入れ子

**media-bodyクラス**が設定されたメディア本文の中に、**mediaクラス**を設定したメディアオブジェクトを配置することで、メディアオブジェクトを入れ子にすることができます。入れ子になったメディアオブジェクトは、ツリー状のコメント欄を作成する場合などに便利です(リスト4-39、図4-40)。

▼リスト4-39 メディアオブジェクトの入れ子(media-object-nest.html)

```
<!-- メディアオブジェクト -->
<div class="media">
  <!-- 画像 -->
  
  <!-- メディア本文 -->
  <div class="media-body">
    <h5>…</h5>
    …中略…
    <!-- メディアオブジェクトの入れ子 -->
    <div class="media mt-3">
      <!-- リンク画像 -->
      <a href="#"></a>
      <!-- メディア本文 -->
      <div class="media-body">
        <h5>…</h5>
        …中略…
      </div>
    </div>
  </div>
</div>
```

▼図4-40 メディアオブジェクトの入れ子



なお先の例では、img要素に**mr-3クラス**、mediaクラスが設定された入れ子のメディアオブジェクトに

**mt-3 クラス**といった Spacing ユーティリティ（P.318 参照）を追加することで、メディアオブジェクトの要素間のスペースが詰まりすぎないようにマージンサイズを調整しています。

### 4.6.3 メディアの位置合わせ

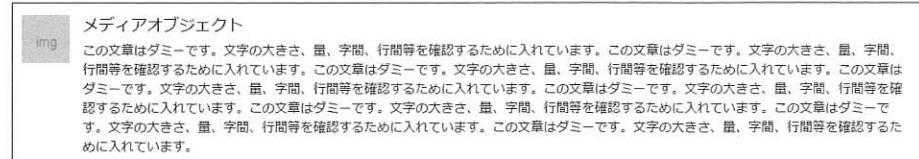
Flex ユーティリティ（P.322 参照）の **align-self-{ 整列方法 }** クラスを使用して、メディアオブジェクト内の画像などの要素を、上部、垂直方向中央、または下部に配置することができます。整列方法には、start（上部に配置）、center（垂直方向中央に配置）、end（下部に配置）があります。

次の例では、メディアオブジェクト内で画像を上部に配置するために、img 要素に **align-self-start** クラスを追加しています（リスト 4-40、図 4-41）。

▼リスト 4-40 メディアオブジェクトの上部に画像を配置（media-object-align-start.html）

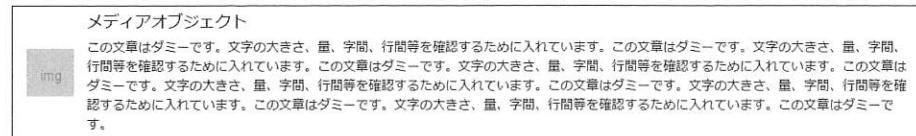
```
<!-- メディアオブジェクト -->
<div class="media">
  
  <!-- メディア本文 -->
  <div class="media-body">
    <h5>…</h5>
    …中略…
  </div>
</div>
```

▼図 4-41 align-self-start クラスで画像を上部に配置



img 要素に追加されたクラスを **align-self-center** に変更すると、画像がメディアオブジェクトの垂直方向中央に配置されます（図 4-42）。

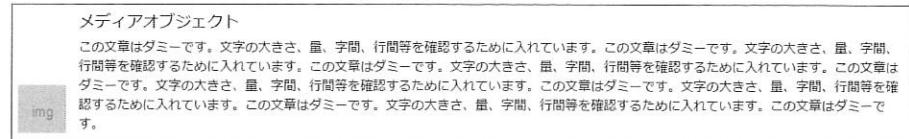
▼図 4-42 align-self-center クラスで画像を垂直方向中央に配置



img 要素に追加されたクラスを **align-self-end** に変更すると、画像がメディアオブジェクトの下部に配置されます（図 4-43）。

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

▼図 4-43 align-self-end クラスで画像を下部に配置



なお先の例では、img 要素に Spacing ユーティリティ（P.318 参照）の **mr-3 クラス**を追加することで、メディアオブジェクトの要素間のスペースが詰まりすぎないようにマージンサイズを調整しています。

#### 4.6.4 メディアオブジェクトの並べ替え

Flex ユーティリティ（P.322 参照）の **order-{ 順番 } クラス**を使用して、メディアオブジェクト内の画像とメディア本文の表示順序を入れ替えることができます。順番には、表示させたい順に 1～12 の数値がります。

次の例では、HTML の構造上は画像が先に、メディア本文が後に記述されていますが、画像に **order-2 クラス**、メディア本文に **order-1 クラス**を追加することで、表示上の順序を入れ替えています（リスト 4-41、図 4-44）。

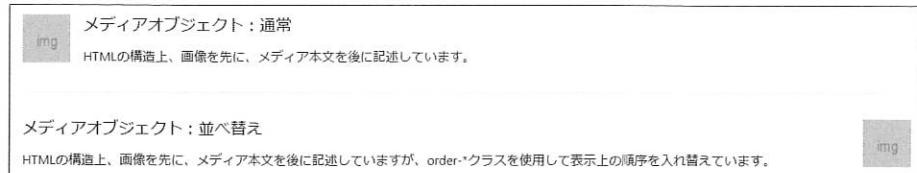
▼リスト 4-41 メディアオブジェクトの並べ替え例（media-object-order.html）

```
<!-- メディアオブジェクト：通常 -->
<div class="media">
  <!-- 画像 -->
  
  <!-- メディア本文 -->
  <div class="media-body">
    <h5>メディアオブジェクト：通常</h5>
    HTMLの構造上、画像を先に、メディア本文を後に記述しています。
  </div>
</div>

<hr>

<!-- メディアオブジェクト：並べ替え -->
<div class="media">
  <!-- 画像 -->
  
  <!-- メディア本文 -->
  <div class="media-body order-1">
    <h5>メディアオブジェクト：並べ替え</h5>
    HTMLの構造上、画像を先に、メディア本文を後に記述していますが、order-* クラスを使用して表示上の順序を入れ替えています。
  </div>
</div>
```

▼図 4-44 メディアオブジェクトの並べ替え例



なお先の例では、img 要素に **ml-3 クラス** や **mr-3 クラス** といった Spacing ユーティリティ (P.318 参照) を追加することで、メディアオブジェクトの要素間のスペースが詰まりすぎないようにマージンサイズを調整しています。

### 4.6.5 メディアオブジェクトをリストに組み込む

メディアオブジェクトは、リスト (ul 要素、ol 要素) に組み込んで使用することができます。ul 要素や ol 要素に **list-unstyled クラス** (P.61 参照) を追加してリストスタイルをリセットした上で、li 要素に **media クラス** を追加します。

次の例では、メディアオブジェクトを ul 要素に組み込んで使用しています (リスト 4-42、図 4-45)。

▼リスト 4-42 メディアオブジェクトをリストに組み込んだ例 (media-object-list.html)

```
<ul class="list-unstyled">
  <!-- メディアオブジェクト -->
  <li class="media mb-4">
    <!-- 画像 -->
    
    <!-- メディア本文 -->
    <div class="media-body">
      <h5>…</h5>
      …中略…
    </div>
  </li>

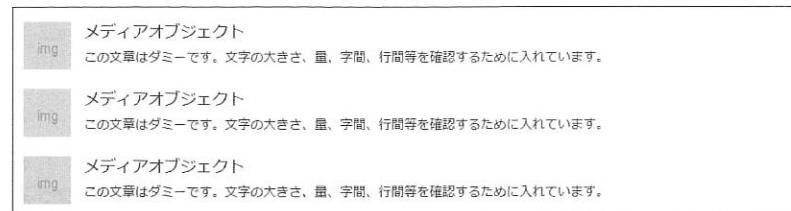
  <!-- メディアオブジェクト -->
  <li class="media mb-4">
    <!-- 画像 -->
    
    <!-- メディア本文 -->
    <div class="media-body">
      <h5>…</h5>
      …中略…
    </div>
  </li>

  <!-- メディアオブジェクト -->
  <li class="media mb-4">
```

```
<!-- 画像 -->

<!-- メディア本文 -->
<div class="media-body">
  <h5>…</h5>
  …中略…
</div>
</li>
</ul>
```

▼図 4-45 メディアオブジェクトをリストに組み込んだ例



なお先の例では、li要素に**mb-4 クラス**、img要素に**mr-3 クラス**といったSpacingユーティリティ(P.318 参照)を追加することで、メディアオブジェクトの要素間のスペースが詰まりすぎないようにマージンサイズを調整しています。