

# 第 6 章

## フォームとボタンの コンポーネント

本章では、Bootstrap のフォームおよびボタンのコンポーネントについて解説します。Bootstrap では、リブートによって体裁の整えられた基本的なスタイルのフォームを利用できますが、それをさらに拡張したカスタムフォームと呼ばれる独自のスタイルも利用することができます。また、フォームでよく利用される入力検証機能や、JavaScript を利用した機能拡張についても見ていきましょう。

Bootstrap の**フォーム**は、リブートで設定されたフォームのスタイル（P.101 参照）を拡張し、一貫性のあるスタイルの入力コントロールやレイアウトを作成するためのコンポーネントです。本節では、Bootstrap のフォームを使用する方法を解説します。

### 6.1.1 基本的な使用例

まずは、input 要素や select 要素、textarea 要素などで作成された入力コントロール（以下、「入力コントロール」）の例を見てみましょう（リスト 6-1、図 6-1）。

▼リスト 6-1 基本的な使用例（form-basic.html）

```
<form>
  <!-- メールアドレス入力 -->
  <div class="form-group">③
    <label for="email1">メールアドレス</label>
    <input type="email" class="form-control" id="email1" aria-describedby="emailHelp" ↵
placeholder="メールアドレスを入力">①
    <small id="emailHelp" class="form-text text-muted">あなたに関する個人情報を収集←
    することはありません。</small>
  </div>
  <!-- パスワード入力 -->
  <div class="form-group">③
    <label for="password1">パスワード</label>
    <input type="password" class="form-control" id="password1" placeholder="パスワードを入力">①
  </div>
  <!-- プルダウンメニュー -->
  <div class="form-group">③
    <label for="select1">プルダウンメニュー</label>
    <select class="form-control" id="select1">――
      <option>1</option>
      <option>2</option>
      <option>3</option>
      <option>4</option>
      <option>5</option>
    </select>
  </div>
  <!-- 複数選択のプルダウンメニュー -->
  <div class="form-group">
```

```

<label for="select2">複数選択のプルダウンメニュー</label>
<select multiple class="form-control" id="select2"> ①
  <option>1</option>
  <option>2</option>
  <option>3</option>
  <option>4</option>
  <option>5</option>
</select>
</div>

<div class="form-group"> ②
  <label for="textarea1">複数行のテキスト入力欄</label>
  <textarea class="form-control" id="textarea1" rows="3"></textarea> ③
</div>

<div class="form-group"> ④
  <label for="file1">ファイルを選択</label>
  <input type="file" class="form-control-file" id="file1"> ⑤
</div>

<div class="form-check">
  <input type="checkbox" class="form-check-input" id="check1">
  <label class="form-check-label" for="check1">チェックする</label>
</div>

<button type="submit" class="btn btn-primary">送信</button>
</form>

```

▼図 6-1 基本的な使用例

メールアドレス  
メールアドレスを入力  
あなたに該する個人情報を収集することはありません。

パスワード  
パスワードを入力

プルダウンメニュー  
1

複数選択のプルダウンメニュー  
1  
2  
3  
4

複数行のテキスト入力欄

ファイルを選択  
[ファイルを選択] 選択されていません

チェックする

送信

入力コントロールは、**form-control クラス**を使用して、一般的な外観、フォーカス状態、サイズなどをスタイルリングします（①）。また、入力コントロールにファイル選択の機能を持たせる場合は、form-control クラスの代わりに**form-control-file クラス**を追加します（②）。

これらの入力コントロールとlabel要素を、**form-group クラス**を追加したdiv要素で囲み、グループ化して**フォームグループ**を作成します（③）。form-group クラスのスタイルには「margin-bottom:1rem」が定義されており、フォームグループ間に余白が付いてフォーム全体が見やすくなります。

## ■ 入力コントロールのサイズを調整する

入力コントロールの高さを大きくする場合は、**form-control クラス**が設定された要素に**form-control-lg クラス**を追加します。小さくする場合は**form-control-sm クラス**を追加します（リスト6-2、図6-2）。

### ▼リスト6-2 入力コントロールのサイズを調整する（form-control-sizing.html）

```
<label for="input1">大サイズのテキスト入力欄</label>
<input class="form-control form-control-lg" id="input1" type="text" placeholder="form-control-lg">
…中略…
<label for="input2">標準サイズのテキスト入力欄</label>
<input class="form-control" id="input2" type="text" placeholder="Default size">
…中略…
<label for="input3">小サイズのテキスト入力欄</label>
<input class="form-control form-control-sm" id="input3" type="text" placeholder="form-control-sm">
…中略…
<label for="select1">大サイズのプルダウンメニュー</label>
<select class="form-control form-control-lg" id="select1">
  <option>form-control-lg</option>
</select>
…中略…
<label for="select2">標準サイズのプルダウンメニュー</label>
<select class="form-control" id="select2">
  <option>Default size</option>
</select>
…中略…
<label for="select3">小サイズのプルダウンメニュー</label>
<select class="form-control form-control-sm" id="select3">
  <option>form-control-sm</option>
</select>
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

▼図 6-2 入力コントロールのサイズを調整する

大サイズのテキスト入力欄  
form-control-lg

標準サイズのテキスト入力欄  
Default size

小サイズのテキスト入力欄  
form-control-sm

大サイズのプルダウンメニュー  
form-control-lg

標準サイズのプルダウンメニュー  
Default size

小サイズのプルダウンメニュー  
form-control-sm

## ■ レンジ入力を作成する 4.1

水平方向にスクロール可能なレンジ入力を作成するには、**form-control-range** クラスを使用します。レンジ入力とは、input 要素の type 属性で type="range" を指定するもので、だいたいこれくらい……といった範囲を入力するためのものです。たとえば、音量や画面の明るさ調節などの範囲などに使用できます（リスト 6-3、図 6-3）。

▼リスト 6-3 レンジ入力欄を作成する (form-range.html)

```
<div class="form-group">
  <label for="formControlRange">レンジ入力の例</label>
  <input type="range" class="form-control-range" id="formControlRange">
</div>
```

▼図 6-3 レンジ入力を作成する

レンジ入力の例

## ■ 読み取り専用のテキストを表示する

入力コントロールに **readonly** 属性を追加すると、入力値の変更ができない読み取り専用のテキストを表示できます。読み取り専用入力コントロールは、無効状態の入力と同様、薄い色のテキストで表示されますが、カーソルの表示は標準のまま保持されます（リスト 6-4、図 6-4）。

▼リスト 6-4 読み取り専用のテキストを表示する (form\_READONLY.html)

```
<input class="form-control" type="text" id="input1" placeholder="読み取り専用のテキスト" readonly>
```

▼図 6-4 読み取り専用のテキストを表示する

読み取り専用のテキストを表示

読み取り専用のテキスト

## ■ 読み取り専用テキストの枠を非表示にする

入力コントロールの枠を非表示にし、読み取り専用テキストだけを表示させる場合は、input要素に **form-control-plaintext** クラスを追加します。このクラスは、枠が非表示になった場合にもマージンとパディングのサイズを整えて、入力コントロールのレイアウトを保持します（リスト 6-5、図 6-5）。

▼リスト 6-5 読み取り専用テキストの枠を非表示にする（form-control-plaintext.html）

```
<input type="text" readonly class="form-control-plaintext" id="staticEmail" value="email@example.com">
...中略...
```

▼図 6-5 読み取り専用テキストの枠を非表示にする

メールアドレス email@example.com

パスワード パスワードを入力

## 6.1.2 チェックボックスとラジオボタン

続いて、選択肢の中から複数項目の選択が可能なチェックボックスや、1項目のみ選択が可能なラジオボタンを作成する例を見てみましょう（リスト 6-6、図 6-6）。

▼リスト 6-6 チェックボックスとラジオボタン（form-check-stacked.html）

```
<!-- チェックボックス -->
<div class="form-check"> _____①
  <input class="form-check-input" type="checkbox" value="" id="check1"> _____③
  <label class="form-check-label" for="check1"> _____②
    チェック1
  </label>
</div>
<div class="form-check"> _____①
  <input class="form-check-input" type="checkbox" value="" id="check2" disabled> _____③
  <label class="form-check-label" for="check2"> _____②
    チェック2（無効）
  </label>
</div>
<hr>
<!-- ラジオボタン -->
<div class="form-check"> _____①
```

```

<input class="form-check-input" type="radio" name="radios" id="radios1" value="option1" checked> ③
<label class="form-check-label" for="radios1"> _____ ②
    オプション1
</label>
</div>
<div class="form-check"> _____ ①
    <input class="form-check-input" type="radio" name="radios" id="radios2" value="option2"> ③
    <label class="form-check-label" for="radios2"> _____ ②
        オプション2
    </label>
</div>
<div class="form-check"> _____ ①
    <input class="form-check-input" type="radio" name="radios" id="radios3" value="option3" disabled> ③
    <label class="form-check-label" for="radios3"> _____ ②
        オプション3 (無効)
    </label>
</div>

```

▼図 6-6 チェックボックスとラジオボタン



input 要素と label 要素を div 要素で囲んで **form-check クラス** を追加します（①）。これにより、各項目は縦並びの配置でスタイルリングされます。label 要素には **form-check-label クラス** を追加します（②）。input 要素には **form-check-input クラス** を追加します（③）。ラベルと入力コントロールは、label 要素の for 属性値と入力コントロールの id 属性値を一致させることで関連付けます。

また、選択無効な項目を設定する場合は、form-check-input クラスが設定された要素に **disabled** 属性を追加します。

## NOTE

**ラベルと入力コントロールの関連付け**

HTML のフォームでは、ラベルと入力コントロールを関連付ける方法は 2 通りあります。1 つ目は、label 要素の for 属性値と入力コントロールの id 属性値を一致させる方法です（リスト 6-7）。

▼リスト 6-7 label 要素の for 属性値と入力コントロールの id 属性値を一致させる

```

<input type="radio" name="name" id="name">
<label for="name">ラベル</label>

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

2つ目は、label要素の子要素として入力コントロールを内包する方法です（リスト6-8）。

▼リスト6-8 label要素の子要素として入力コントロールを内包する

```
<label>
  <input type="radio" name="name">ラベル
</label>
```

Bootstrapのチェックボックスとラジオボタンでは、input要素とlabel要素を分けた簡潔で利用しやすい構造を提供できるように、1つ目の方法が採用されています。

## ■ 選択項目を横並びに変更する

選択項目を横並びのレイアウトに変更する場合は、form-checkクラスが設定されたdiv要素に**form-check-inline**クラスを追加します（リスト6-9、図6-7）。

▼リスト6-9 選択項目を横並びに変更する（form-check-inline.html）

```
<div class="form-check form-check-inline">
  <input class="form-check-input" type="checkbox" id="checkbox1" value="option1">
  <label class="form-check-label" for="checkbox1">1</label>
</div>
<div class="form-check form-check-inline">
  <input class="form-check-input" type="checkbox" id="checkbox2" value="option2">
  <label class="form-check-label" for="checkbox2">2</label>
</div>
<div class="form-check form-check-inline">
  <input class="form-check-input" type="checkbox" id="checkbox3" value="option3" disabled>
  <label class="form-check-label" for="checkbox3">3</label>
</div>
```

▼図6-7 選択項目を横並びに変更する



□ 1 □ 2 □ 3

## ■ ラベル表示のない項目を作成する

ラベル表示のない選択項目を作成する場合は、form-check-inputクラスが設定された要素に**position-static**クラスを追加します。ただし、スクリーンリーダーなどの支援技術に対してラベル付けを行う**aria-label**属性を追加して、アクセシビリティに配慮しましょう（リスト6-10、図6-8）。

▼リスト6-10 ラベル表示のない項目を作成する（form-check-without-label.html）

```
<div class="form-check">
  <input class="form-check-input position-static" type="checkbox" id="blankCheckbox" value="option1" aria-label="...">
```

```
</div>
<div class="form-check">
  <input class="form-check-input position-static" type="radio" name="blankRadio" id="blankRadio1" value="option1" aria-label="...">
</div>
```

▼図 6-8 ラベル表示のない項目を作成する



### 6.1.3 レイアウトを調整する

Bootstrap のフォームでは、ほぼすべての入力コントロールのスタイルに **display:block** と **width:100%** が定義されています。そのため、コンポーネントの初期設定では入力コントロールが縦並びになります。このレイアウトに変更や調整を行う方法として、フォームにグリッドレイアウト（P.22 参照）を組み込む方法や、レイアウト調整のために定義されたクラスを追加する方法があります。

#### ■ グリッドレイアウトを組み込む

まず、フォームをグリッドレイアウトで配置する方法を見ていきましょう。`form-group` クラスが設定された要素に **col-\* クラス** や **col-{ ブレイクポイント }-\* クラス** を追加して、`row` クラスで囲み、グリッドのカラムとして幅指定を行います。次の例では、姓の入力欄と名の入力欄が横並びのカラムになるように幅指定を行っています（リスト 6-11、図 6-9）。

▼リスト 6-11 グリッドレイアウトを組み込む（form-grid.html）

```
<form>
  <div class="row">
    <div class="col">
      <input type="text" class="form-control" placeholder="姓">
    </div>
    <div class="col">
      <input type="text" class="form-control" placeholder="名">
    </div>
  </div>
</form>
```

▼図 6-9 グリッドレイアウトを組み込む



## ■ コンパクトなカラム間隔のフォームを作成する

グリッドレイアウトのフォームのカラム間隔をコンパクトにする場合は、div要素にrowクラスを追加する代わりに、**form-row**クラスを追加します。

次の例では、メールアドレスの欄とパスワードの欄が、画面幅md以上で6列カラムとして2つ横並びになるようにレイアウトを設定しています。また、国と郵便番号と都道府県の欄は、画面幅md以上で4列、2列、6列カラムとして3つ横並びになるように設定しています（リスト6-12、図6-10）。

▼リスト6-12 コンパクトなカラム間隔のフォーム（form-row.html）

```
<form>
  <div class="form-row">
    <div class="form-group col-md-6">
      <label for="inputEmail">メールアドレス</label>
      <input type="email" class="form-control" id="inputEmail" placeholder="メールアドレスを入力">
    </div>
    <div class="form-group col-md-6">
      <label for="inputPassword">パスワード</label>
      <input type="password" class="form-control" id="inputPassword" placeholder="パスワードを入力">
    </div>
  </div>
  <div class="form-row">
    <div class="form-group col-md-4">
      <label for="inputState">国</label>
      <select id="inputState" class="form-control">
        <option selected>Choose...</option>
        <option>...</option>
      </select>
    </div>
    <div class="form-group col-md-2">
      <label for="inputZip">郵便番号</label>
      <input type="text" class="form-control" id="inputZip">
    </div>
    <div class="form-group col-md-6">
      <label for="inputCity">都道府県</label>
      <input type="text" class="form-control" id="inputCity">
    </div>
  </div>
  ...中略...
</form>
```

▼図 6-10 コンパクトなカラム間隔のフォーム

メールアドレス パスワード  
メールアドレスを入力 パスワードを入力  
国 郵便番号 都道府県  
Choose...  
住所1 市町村  
住所2 マンション名  
□ チェックする  
サインイン

## ■ 水平配置のフォームを作成する

フォーム全体のレイアウトだけではなく、label要素と入力コントロールのレイアウトにも、グリッドレイアウトを組み込むことができます。これによって、ラベルと入力コントロールを水平に配置することができます（リスト6-13、図6-11）。

▼リスト6-13 水平配置のフォームを作成する（form-horizontal-form.html）

```
<form>
  <div class="form-group row"> _____ ①
    <label for="inputEmail" class="col-sm-2 col-form-label">メールアドレス</label> _____ ②
    <div class="col-sm-10"> _____
      <input type="email" class="form-control" id="inputEmail" placeholder="メールアドレスを入力">
    </div>
  </div>
  <div class="form-group row"> _____ ①
    <label for="inputPassword" class="col-sm-2 col-form-label">パスワード</label> _____ ②
    <div class="col-sm-10"> _____
      <input type="password" class="form-control" id="inputPassword" placeholder="パスワードを入力">
    </div>
  </div>
  <fieldset class="form-group">
    <div class="row"> _____ ①
      <legend class="col-form-label col-sm-2 pt-0">ラジオボタン</legend> _____ ②
      <div class="col-sm-10"> _____
        <div class="form-check">
          <input class="form-check-input" type="radio" name="radios" id="radios1" value="option1" checked>
          <label class="form-check-label" for="radios1">
            オプション1
          </label>
        </div>
        ...中略...
      </div>
    </div>
  </fieldset>
```

```

        </div>
        </div>
</fieldset>
<div class="form-group row"> _____ ①
    <div class="col-sm-2">チェックボックス</div> _____ ②
    <div class="col-sm-10">
        <div class="form-check">
            <input class="form-check-input" type="checkbox" id="check1">
            <label class="form-check-label" for="check1">
                チェックする
            </label>
        </div>
    </div>
<div class="form-group row">
    <div class="col-sm-10">
        <button type="submit" class="btn btn-primary">サインイン</button>
    </div>
</div>
</form>

```

▼図 6-11 水平配置のフォームを作成する

The screenshot shows a login form with the following structure:

- メールアドレス:** Input field with placeholder "メールアドレスを入力".
- パスワード:** Input field with placeholder "パスワードを入力".
- ラジオボタン:** Radio button group with three options: "オプション1" (selected), "オプション2", and "オプション3 (無効)".
- チェックボックス:** Checkbox labeled "チェックする".
- サインイン:** Primary button labeled "サインイン".

まずフォームグループに **row クラス** を追加して行を作ります（①）。次に、ラベルと入力コントロールに **col-{ ブレイクポイント }-\* クラス** を追加して幅を指定し、ラベルと入力コントロールを水平に配置します。label 要素に **col-form-label クラス** を追加すると、ラベルを関連する入力コントロールの高さに合わせてレイアウトできます（②）。

また、legend 要素に **col-form-label クラス** を追加して、label 要素と同様のスタイルで表示することができます。

## ■ ラベルのサイズを調整する

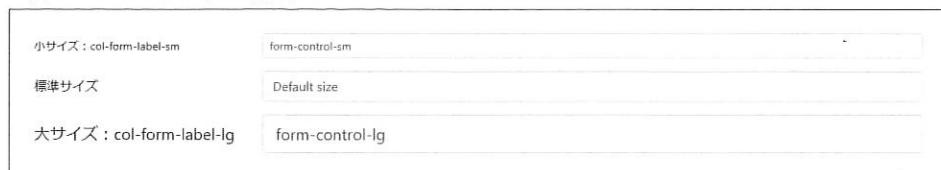
入力コントロールのサイズ調整（P.198 参照）に合わせて、ラベルのサイズを調整します。ラベルのサイズを大きくする場合は、label 要素に **col-form-label-lg クラス** を追加します。小さくする場合は、label 要素に **col-form-label-sm クラス** を追加します（リスト 6-14、図 6-12）。

▼リスト 6-14 ラベルサイズを調整する（form-label-sizing.html）

```
<label for="labelSm" class="col-sm-3 col-form-label col-form-label-sm">小サイズ：col-form-label-→
sm</label>
```

```
…中略…
<label for="labelDefault" class="col-sm-3 col-form-label">標準サイズ</label>
…中略…
<label for="labelLg" class="col-sm-3 col-form-label col-form-label-lg">大サイズ: col-form-label-→
lg</label>
```

▼図 6-12 ラベルサイズを調整する



## ■ 自動サイズ調整のカラムを作成する

内容によって幅サイズが自動調整されるカラムを作成する場合は、**col-\* クラス**や**col-{ ブレイクポイント }-\* クラス**の代わりに**col-auto クラス**を追加します。また隣り合うラベルや入力コントロールを垂直方向中央に揃える場合は、**row クラス**または**form-row クラス**が設定された

要素にFlexユーティリティ(P.322参照)の**align-items-center クラス**を追加します(リスト6-15、図6-13)。

▼リスト 6-15 自動サイズ調整のカラムを作成する (form-auto-sizing.html)

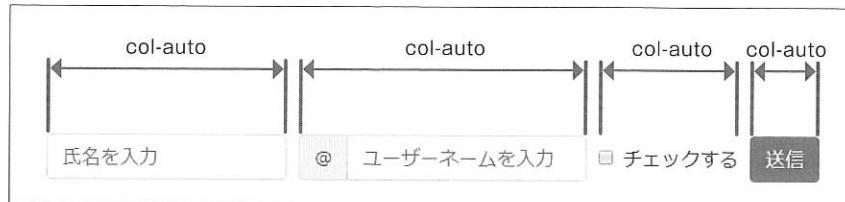
```
<form>
  <div class="form-row align-items-center">
    <div class="col-auto">
      <label class="sr-only" for="inputName">氏名</label>
      <input type="text" class="form-control mb-2" id="inputName" placeholder="氏名を入力">
    </div>
    <div class="col-auto">
      <label class="sr-only" for="inputUsername">ユーザー名</label>
      <div class="input-group mb-2">
        <div class="input-group-prepend">
          <div class="input-group-text">@</div>
        </div>
        <input type="text" class="form-control" id="inputUsername" placeholder="ユーザー名を入力">
      </div>
    </div>
    <div class="col-auto">
      <div class="form-check mb-2">
        <input class="form-check-input" type="checkbox" id="check">
        <label class="form-check-label" for="check">
          チェックする
        </label>
      </div>
    </div>
  </div>
```

```

    <button type="submit" class="btn btn-primary mb-2">送信</button>
  </div>
</div>
</form>

```

▼図 6-13 自動サイズ調整のカラムを作成する



## ■ インラインのフォームを作成する

form 要素に **form-inline** クラスを追加すると、画面幅 576px 以上の際に、一連のラベル、フォームの入力コントロール、およびボタンをインライン表示にすることができます。form-inline クラスのスタイルには、flexbox のレイアウトを指定する **display:flex** が定義されているため、Flex ユーティリティ (P.322 参照) や Spacing ユーティリティ (P.318 参照) を使った整列が可能になります。

また、ラベルを非表示にしたい場合は、label 要素にスクリーンリーダー用ユーティリティ (P.354 参照) の **sr-only** クラスを追加して、スクリーンリーダー用の非表示テキストにします。スクリーンリーダーなどの支援技術のユーザーにとって、label 要素は入力コントロールに必要な要素です。削除してしまわないようにしましょう (リスト 6-16、図 6-14)。

▼リスト 6-16 インラインのフォームを作成する (form-inline.html)

```

<form class="form-inline">
  <label class="sr-only" for="name">氏名</label>
  <input type="text" class="form-control mb-2 mr-sm-2" id="name" placeholder="氏名を入力">
  <label class="sr-only" for="username">ユーザー名</label>
  <div class="input-group mb-2 mr-sm-2">
    <div class="input-group-prepend">
      <div class="input-group-text">@</div>
    </div>
    <input type="text" class="form-control" id="username" placeholder="ユーザー名を入力">
  </div>
  <div class="form-check mb-2 mr-sm-2">
    <input class="form-check-input" type="checkbox" id="check">
    <label class="form-check-label" for="check">
      チェックする
    </label>
  </div>
  <button type="submit" class="btn btn-primary mb-2">送信</button>
</form>

```

▼図 6-14 インラインのフォームを作成する

### 6.1.4 ヘルプテキストを表示する

入力コントロールの下に表示するヘルプテキスト（補助説明文）を作成する場合は、テキスト要素に **form-text** クラスを追加します。form-text クラスは、Bootstrap 3 で **help-block** と呼ばれていたクラスが変更されたものです。次の例では、small 要素に form-text クラスと Color ユーティリティ（P.302 参照）の **text-muted** クラスを追加して、小さくグレーで表示されるヘルプテキストを作成しています（リスト 6-17、図 6-15）。

▼リスト 6-17 ヘルプテキストを表示する（form-text.html）

```
<form>
  <label for="inputPassword">パスワード</label>
  <input type="password" id="inputPassword" class="form-control" aria-describedby="passwordHelpBlock">
    <small id="passwordHelpBlock" class="form-text text-muted">パスワードは8~20文字で、文字と数字を含み、スペース、特殊文字、または絵文字を含むことはできません。</small>
</form>
```

▼図 6-15 ヘルプテキストを表示する

なお、form-text クラスのスタイルには **display:block** と **margin-top:0.25rem** が定義されているため、入力コントロールと縦並びの配置でヘルプテキストが表示されます。入力コントロールとヘルプテキストを横並びにインライン表示したい場合は、form-text クラスを追加せずにヘルプテキストを作成してください。

また、入力コントロールには属性 **aria-describedby=" (ヘルプテキストの ID) "** を追加し、スクリーンリーダーなどの支援技術に対して、ヘルプテキストと入力コントロールとの関連を伝え、アクセシビリティに配慮しましょう。

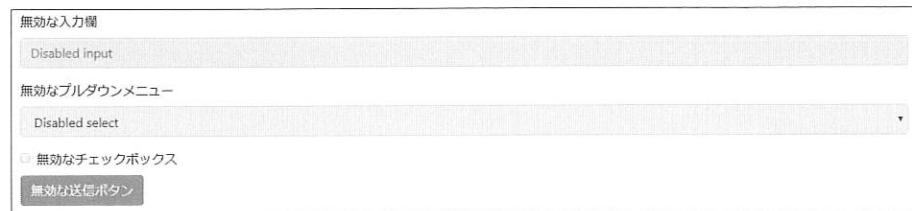
### 6.1.5 一連のフォームグループをまとめて無効にする

ユーザーが操作できない入力コントロールを作成する場合、input 要素に **disabled** 属性を追加して選択を無効にできることは既に述べました（P.201 参照）。もし、グループ化されている一連の入力コントロールを一度にまとめて無効にする場合は、fieldset 要素に disabled 属性を追加します。disabled 属性が追加された要素の内容はグレー表示されます（リスト 6-18、図 6-16）。

▼リスト 6-18 一連のフォームグループをまとめて無効にする (form-disabled-fieldset.html)

```
<form>
  <fieldset disabled>
    <div class="form-group">
      <label for="disabledTextInput">無効な入力欄</label>
      <input type="text" id="disabledTextInput" class="form-control" placeholder="Disabled input">
    </div>
    <div class="form-group">
      <label for="disabledSelect">無効なプルダウンメニュー</label>
      <select id="disabledSelect" class="form-control">
        <option>Disabled select</option>
      </select>
    </div>
    <div class="form-check">
      <input class="form-check-input" type="checkbox" id="disabledCheck" disabled>
      <label class="form-check-label" for="disabledCheck">無効なチェックボックス</label>
    </div>
    <button type="submit" class="btn btn-primary">無効な送信ボタン</button>
  </fieldset>
</form>
```

▼図 6-16 一連のフォームグループをまとめて無効にする



ただし、a要素にはdisabled属性がサポートされていないため、この方法ではフォーム内のa要素で作られたボタンを無効化することができません。a要素のボタンを無効化する場合は、「無効状態のボタンを作成する」(P.236)を参照してください。

またInternet Explorer 11以降においては、fieldset要素でdisabled属性が完全にはサポートされていないため、JavaScriptをカスタムで追加するなどの対応が必要です。

### 6.1.6 フォームの入力検証機能を使う

ユーザーの入力データを検証してフィードバックを返す機能（以下、「入力検証機能」）については、HTML5に実装されているブラウザデフォルトの入力検証機能を使用するか、Bootstrapのフォームの検証機能を使用するかを選択することができます。ここでは、Bootstrapのフォームの入力検証機能を使用する方法について説明します。

## ■ Bootstrap のフォームの入力検証機能を使用する

まず、form 要素に **novalidate** 属性を追加して、ブラウザデフォルトの入力検証機能を無効化します。入力必須を指定する項目の入力コントロールには **required** 属性を追加しておきましょう。また、フィードバックの文章は、**valid-feedback** クラスや **invalid-feedback** クラスを追加した要素で囲みます。入力値が妥当な場合のフィードバックには valid-feedback クラスを使用し、緑色で表示します。妥当でない場合のフィードバックには invalid-feedback クラスを使用し、赤色で表示します（リスト 6-19）。

▼リスト 6-19 Bootstrap のフォームの入力検証機能を使用する（form-validation-custom.html）

```
<form class="needs-validation" novalidate>
  <div class="form-row">
    <div class="col-md-6 mb-3">
      <label for="validation01">姓</label>
      <input type="text" class="form-control" id="validation01" placeholder="First name" value="山田" required>
      <div class="valid-feedback">
        入力済！
      </div>
    </div>
    <div class="col-md-6 mb-3">
      <label for="validation02">名</label>
      <input type="text" class="form-control" id="validation02" placeholder="Last name" value="太郎" required>
      <div class="valid-feedback">
        入力済！
      </div>
    </div>
  </div>
  <div class="form-row">
    <div class="col-md-6 mb-3">
      <label for="validation03">市町村</label>
      <input type="text" class="form-control" id="validation03" placeholder="市町村名を入力" required>
      <div class="invalid-feedback">
        市町村名を入力してください
      </div>
    </div>
    <div class="col-md-3 mb-3">
      <label for="validation04">都道府県</label>
      <input type="text" class="form-control" id="validation04" placeholder="都道府県名を入力" required>
      <div class="invalid-feedback">
        国名を入力してください
      </div>
    </div>
    <div class="col-md-3 mb-3">
      <label for="validation05">郵便番号</label>
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

```

<input type="text" class="form-control" id="validation05" placeholder="郵便番号を入力" required>
<div class="invalid-feedback">
    郵便番号を入力してください
</div>
</div>
<button class="btn btn-primary" type="submit">送信する</button>
</form>

```

次に、JavaScriptコードを追加します。Bootstrapは、入力検証の結果が妥当であるかどうかを、CSSの擬似クラス`:invalid`と`:valid`として表します。`:invalid`および`:valid`スタイルは、親要素であるform要素の`was-validated`クラスに対して付与されます。

そこで、無効な入力で送信しようとした場合に送信をキャンセルし、`was-validated`クラスを追加するしくみをJavaScriptで実装します。これによって、ユーザーが無効な入力で送信しようとすると、入力コントロールにメッセージが表示されるようになります（リスト6-20、図6-17）。

▼リスト6-20 無効な入力がある場合にフォームの送信を無効にする（form-validation-custom.html）

```

<script>
(function() {
    'use strict';
    window.addEventListener('load', function() {
        // 入力検証のスタイルを適用するフォームを取得
        var forms = document.getElementsByClassName('needs-validation');
        // ループして帰順を防ぐ
        var validation = Array.prototype.filter.call(forms, function(form) {
            form.addEventListener('submit', function(event) {
                // バリデーターが通らない場合、イベントをキャンセルして、was-validatedクラスを追加
                if (form.checkValidity() === false) {
                    event.preventDefault();
                    event.stopPropagation();
                }
                form.classList.add('was-validated');
            }, false);
        });
    }, false);
})();
</script>

```

▼図 6-17 フォームの入力検証機能を使用したフィードバック表示例

また、フィードバックをツールチップのスタイルで表示するには、フィードバックの文章に設定されている valid-feedback クラスを **valid-tooltip クラス** に、invalid-feedback クラスを **invalid-tooltip クラス** に置き換えます。ただし、ツールチップの配置には、親要素に **position: relative** のスタイルが設定されている必要があります。次の例では、親要素に追加されている col-{ ブレイクポイント }-\* クラスに position: relative があらかじめ設定されています（リスト 6-21、図 6-18）。

▼リスト 6-21 フィードバックをツールチップのスタイルで表示する（form-validation-tooltip.html）

```
<form class="needs-validation" novalidate>
  <div class="row">
    <div class="col-md-6 mb-3">
      <label for="validation01">姓</label>
      <input type="text" class="form-control" id="validation01" placeholder="First name" value="山田" required>
      <div class="valid-tooltip">
        入力済！
      </div>
    </div>
    <div class="col-md-6 mb-3">
      <label for="validation02">名</label>
      <input type="text" class="form-control" id="validation02" placeholder="Last name" value="太郎" required>
      <div class="valid-tooltip">
        入力済！
      </div>
    </div>
  </div>
  <div class="row">
    <div class="col-md-6 mb-3">
      <label for="validation03">市町村</label>
      <input type="text" class="form-control" id="validation03" placeholder="市町村名を入力" required>
      <div class="invalid-tooltip">
        市町村名を入力してください
      </div>
    </div>
  </div>
</form>
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

```

<label for="validation04">都道府県</label>
<input type="text" class="form-control" id="validation04" placeholder="都道府県名を入力" required>
<div class="invalid-tooltip">
    国名を入力してください
</div>
</div>
<div class="col-md-3 mb-3">
    <label for="validation05">郵便番号</label>
    <input type="text" class="form-control" id="validation05" placeholder="郵便番号を入力" required>
    <div class="invalid-feedback">
        郵便番号を入力してください
    </div>
    </div>
    <button class="btn btn-primary" type="submit">送信する</button>
</form>

```

▼図 6-18 ツールチップのスタイルで表示したフィードバック

The screenshot shows a form with two columns. The left column has fields for '姓' (last name) containing '山田' and '市町村名を入力' (enter town/village name) containing '太郎'. Below these are two buttons: '入力済!' (Input completed!) and '市町村名を入力してください' (Please enter town/village name). The right column has fields for '名' (first name) containing '太郎' and '郵便番号を入力' (enter zip code) containing '郵便番号を入力してください' (Please enter zip code). Below these are two buttons: '入力済!' (Input completed!) and '郵便番号を入力してください' (Please enter zip code).

## ■ サーバー側で入力検証を行う場合

フォームの入力検証を JavaScript ではなく、サーバー側で行う場合は、無効な入力があった場合の表示に **is-invalid クラス**、有効な入力の表示に **is-valid クラス** を付与します。これらは、クライアント側で検証する際に使用される :valid や :invalid に相当するスタイルで、サーバー側でこれらのスタイルを付与することで、検証結果に応じたスタイルを適用できます。

なお、フィードバックメッセージの表示には、これまでと同じ、**invalid-feedback クラス** を利用します（リスト 6-22、図 6-19）。

▼リスト 6-22 サーバー側での検証を行う (form-validation-server-side.html)

```

<form>
    <div class="form-row">
        <div class="col-md-6 mb-3">
            <label for="validationServer01">姓</label>
            <input type="text" class="form-control is-valid" id="validationServer01" placeholder="First name" value="山田" required>
            <div class="invalid-feedback">
                姓を入力してください
            </div>
        </div>
    </div>

```

```

</div>
<div class="col-md-6 mb-3">
    <label for="validationServer02">名</label>
    <input type="text" class="form-control is-valid" id="validationServer02" placeholder="Last name" value="太郎" required>
</div>
</div>
<div class="form-row">
    <div class="col-md-6 mb-3">
        <label for="validationServer03">市町村</label>
        <input type="text" class="form-control is-invalid" id="validationServer03" placeholder="市町村名を入力" required>
        <div class="invalid-feedback">
            市町村名を入力してください
        </div>
    </div>
    <div class="col-md-3 mb-3">
        <label for="validationServer04">都道府県</label>
        <input type="text" class="form-control is-invalid" id="validationServer04" placeholder="都道府県名を入力" required>
        <div class="invalid-feedback">
            都道府県名を入力してください
        </div>
    </div>
    <div class="col-md-3 mb-3">
        <label for="validationServer05">郵便番号</label>
        <input type="text" class="form-control is-invalid" id="validationServer05" placeholder="郵便番号を入力" required>
        <div class="invalid-feedback">
            郵便番号を入力してください
        </div>
    </div>
    <button class="btn btn-primary" type="submit">送信する</button>
</form>

```

▼図 6-19 サーバー側で入力検証を行う場合のフィードバック表示例

姓	名	
山田	太郎	
市町村	都道府県	郵便番号
市町村名を入力	都道府県名を入力	郵便番号を入力
市町村名を入力してください		
都道府県名を入力してください		
郵便番号を入力してください		
<input type="button" value="送信する"/>		

## 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

Bootstrap のフォームには、より一貫性のあるスタイルにカスタマイズするためのクラスとして、**custom-{ 入力コントロールの種類 } クラス**や**custom-control-{ 機能 } クラス**が用意されています。入力コントロールの種類には **checkbox** や **radio** などが、機能には **input**（入力）や **indicator**（表示）、**description**（説明）などがあります。Bootstrap では、これらのクラスでスタイル設定されたフォームを**カスタムフォーム**と呼んでいます。

なお、Bootstrap のフォームの入力検証機能（P.210 参照）は、カスタムフォームの入力コントロールにも利用できます。

### ■ チェックボックスとラジオボタンを作成する

次の例では、カスタムフォームのチェックボックスとラジオボタンを作成しています（リスト 6-23、図 6-20）。

▼リスト 6-23 チェックボックスとラジオボタンを作成する（form-custom-input.html）

```
<form>
  <!-- カスタムフォームのチェックボックス -->
  <div class="custom-control custom-checkbox mb-3"> ①
    <input type="checkbox" class="custom-control-input" id="customCheck1"> ③
    <label class="custom-control-label" for="customCheck1">チェックする</label>
  </div>
  <!-- カスタムフォームのラジオボタン -->
  <div class="custom-control custom-radio"> ②
    <input type="radio" id="option1" name="option" class="custom-control-input"> ③
    <label class="custom-control-label" for="option1">オプション1</label> ④
  </div>
  <div class="custom-control custom-radio">
    <input type="radio" id="option2" name="option" class="custom-control-input"> ③
    <label class="custom-control-label" for="option2">オプション2</label> ④
  </div>
</form>
```

▼図 6-20 通常のチェックボックスとラジオボタン（左）、カスタムフォームのチェックボタンとラジオボタン（右）



カスタムフォームのチェックボックスを作成する場合は、親要素となるdiv要素に**custom-control クラス**と**custom-checkbox クラス**を追加します（①）。同様に、ラジオボタンを作成する場合は、div要素に**custom-control クラス**と**custom-radio クラス**を追加します（②）。

次に、初期設定の入力コントロール部分を非表示にして、カスタムフォームの入力コントロール部分を新たに作成します。まず、input要素に**custom-control-input クラス**を追加します。custom-control-input クラ

スのスタイルには、色の透明度を指定する **opacity: 0;** が定義されているため、初期設定の入力コントロールが透明化されて非表示になります（③）。

次に、label 要素に **custom-control-label** クラスを追加して、カスタムフォームの入力コントロールを新たに作成します（④）。

## 【カスタムフォームのチェックボックスに不確定状態(indeterminate)を指定する】

HTML5 では、チェックボックスの input 要素に表 6-1 の属性を追加することで、3 つの選択状態を表示することができます。

▼表 6-1 チェックボックスに追加する属性と状態の表示

属性	状態
checked	選択済み
unchecked	未選択
indeterminate	不確定。選択か未選択かはつきりしない中間の状態

ただし、Bootstrap のカスタムフォームのチェックボックスでは、input 要素に indeterminate 属性を追加しても不確定の状態を表示することができません。カスタムフォームのチェックボックスに不確定の状態を表示する場合は、次のような jQuery を使用し、:indeterminate 擬似クラスを使ってスタイルリングする必要があります（リスト 6-24）。

▼リスト 6-24 不確定の表示に使用する jQuery (form-custom-indeterminate.html)

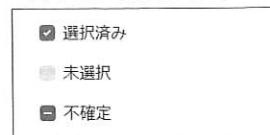
```
$('.indeterminate-box').prop('indeterminate', true)
```

次の例では、カスタムフォームのチェックボックスに、「選択済み」、「未選択」、「不確定」の状態を表示させています（リスト 6-25、図 6-21）。

▼リスト 6-25 カスタムフォームのチェックボックスの 3 つの選択状態 (form-custom-indeterminate.html)

```
<form>
  <!-- カスタムフォームのチェックボックス -->
  <div class="custom-control custom-checkbox mb-3">
    <input type="checkbox" class="custom-control-input" id="customCheck1" checked>
    <label class="custom-control-label" for="customCheck1">選択済み</label>
  </div>
  <div class="custom-control custom-checkbox mb-3">
    <input type="checkbox" class="custom-control-input" id="customCheck2" unchecked>
    <label class="custom-control-label" for="customCheck2">未選択</label>
  </div>
  <div class="custom-control custom-checkbox mb-3">
    <input type="checkbox" class="custom-control-input indeterminate-box" id="customCheck3">
    <label class="custom-control-label" for="customCheck3">不確定</label>
  </div>
</form>
```

▼図 6-21 カスタムフォームのチェックボックスの3つの選択状態を表示する



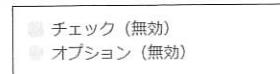
## ■ チェックボックスとラジオボタンを無効にする

カスタムフォームのチェックボックスやラジオボタンも、input要素に**disabled**属性を追加することで選択を無効にすることができます（リスト6-26、図6-22）。

▼リスト 6-26 カスタムフォームのチェックボックスとラジオボタンを無効にする（form-custom-input-disabled.html）

```
<form>
  <!-- チェックボックスを無効にする -->
  <div class="custom-control custom-checkbox">
    <input type="checkbox" class="custom-control-input" id="customCheckDisabled" disabled>
    <label class="custom-control-label" for="customCheckDisabled">
      チェック（無効）
    </label>
  </div>
  <!-- ラジオボタンを無効にする -->
  <div class="custom-control custom-radio">
    <input type="radio" name="radioDisabled" id="customRadioDisabled" class="custom-control-input" disabled>
    <label class="custom-control-label" for="customRadioDisabled">
      オプション（無効）
    </label>
  </div>
</form>
```

▼図 6-22 カスタムフォームのチェックボックスとラジオボタンを無効にする



## ■ カスタムフォームのプルダウンメニューを作成する

カスタムフォームのプルダウンメニューを作成する場合は、select要素に**custom-select**クラスを追加します（リスト6-27、図6-23）。

▼リスト 6-27 カスタムフォームのプルダウンメニューを作成する（form-custom-select.html）

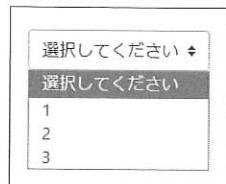
```
<form>
  <select class="custom-select">
    <option selected>選択してください</option>
    <option value="1">1</option>
```

```

<option value="2">2</option>
<option value="3">3</option>
</select>
</form>

```

▼図 6-23 カスタムフォームのプルダウンメニューを作成する



また、プルダウンメニューのサイズを変更することもできます。プルダウンメニューのサイズを大きくする場合は、custom-select クラスが設定された要素に **custom-select-lg クラス** を追加します。プルダウンメニューのサイズを小さくする場合は **custom-select-sm クラス** を追加します（リスト 6-28、図 6-24）。

▼リスト 6-28 プルダウンメニューのサイズを変更する（form-custom-select-sizing.html）

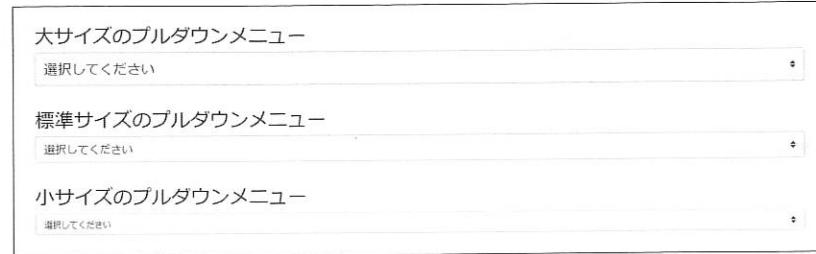
```

<div class="container">
  <h3>大サイズのプルダウンメニュー</h3>
  <form>
    <select class="custom-select custom-select-lg">
      <option selected>選択してください</option>
      ...中略...
      </select>
    </form>
  </div>
  <div class="container">
    <h3>標準サイズのプルダウンメニュー</h3>
    <form>
      <select class="custom-select">
        <option selected>選択してください</option>
      ...中略...
      </select>
    </form>
  </div>
  <div class="container">
    <h3>小サイズのプルダウンメニュー</h3>
    <form>
      <select class="custom-select custom-select-sm">
        <option selected>選択してください</option>
      ...中略...
      </select>
    </form>
  </div>

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

▼図 6-24 プルダウンメニューのサイズを変更する



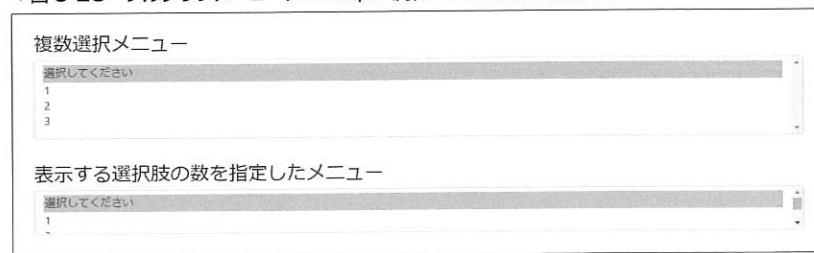
## ■ プルダウンメニューに追加できる属性

カスタムフォームのプルダウンメニューには、multiple 属性、size 属性を追加することができます。複数選択メニューを作成するには、custom-select クラスが設定された要素に multiple 属性を追加します。表示する選択肢の数を指定するには、size 属性を追加します。次の例では属性 size="2" を追加して、表示する選択肢を 2 つに指定しています（リスト 6-29、図 6-25）。

▼リスト 6-29 プルダウンメニューに multiple 属性と size 属性を追加する (form-custom-select-attr.html)

```
<div class="container">
<h3>複数選択メニュー</h3>
<form>
  <select class="custom-select" multiple>
    <option selected>選択してください</option>
    ...中略...
  </select>
</form>
</div>
<div class="container">
<h3>表示する選択肢の数を指定したメニュー</h3>
<form>
  <select class="custom-select" size="2">
    <option selected>選択してください</option>
    ...中略...
  </select>
</form>
</div>
```

▼図 6-25 プルダウンメニューに multiple 属性と size 属性を追加する



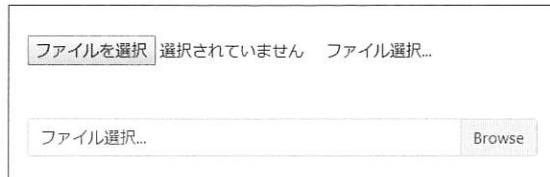
## ■ カスタムフォームのファイル選択を作成する

カスタムフォームのファイル選択を作成するには、**custom-file-input** クラスを追加します（リスト 6-30、図 6-26）。

▼リスト 6-30 カスタムフォームのファイル選択を作成する（form-custom-file.html）

```
<form>
  <!-- デフォルトのファイル選択 -->
  <div class="default-file mb-5">
    <input type="file" id="defaultFile" lang="ja">
    <label for="defaultFile">ファイル選択...</label>
  </div>
  <!-- カスタムフォームのファイル選択 -->
  <div class="custom-file">
    <input type="file" class="custom-file-input" id="customFile" lang="ja">
    <label class="custom-file-label" for="customFile">ファイル選択...</label>
  </div>
</form>
```

▼図 6-26 デフォルトのファイル選択（上）、カスタムフォームのファイル選択（下）



カスタムフォームのファイル選択では、「ファイルを選択」のボタンが「Browse」になります。これを日本語で「参照」の文字列に変えるには、CSS を次のように変更します。セレクタがやや複雑ですが、「custom-file-input クラスを持ち、lang 属性が ja の要素」の後に続く「custom-file-label クラスを持つ要素の後」に「参照」という文字列を追加、という意味になります（リスト 6-31、図 6-27）。

▼リスト 6-31 「Browse」を「参照」にする（form-custom-file-ja.html）

```
.custom-file-input:lang(ja) ~ .custom-file-label::after {
  content: "参照";
}
```

▼図 6-27 「Browse」を「参照」にする



- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

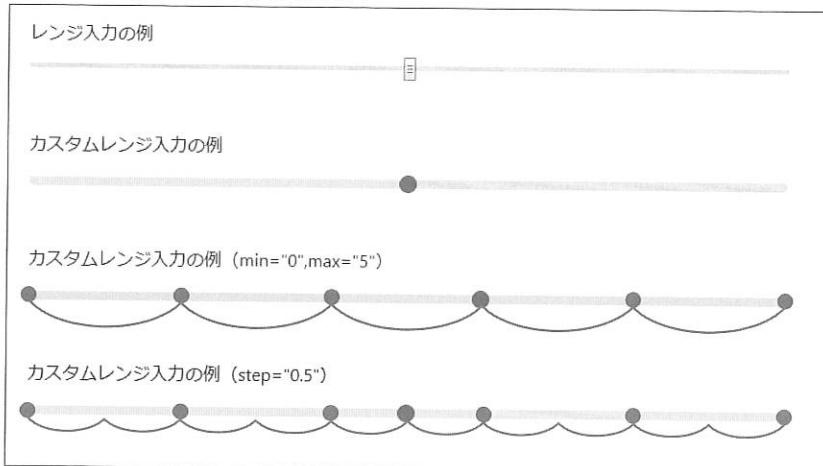
## ■ カスタムレンジ入力を作成する 4.1

属性 type="range" が設定された input 要素に **custom-range** クラスを使用すると、カスタムフォームのレンジ入力コントロール（以下、「カスタムレンジ入力」）を作成できます。トラック（背景）は、グレーのバー、つまみ（値）は円形でスタイルリングされます（リスト 6-32、図 6-28）。

▼リスト 6-32 カスタムレンジ入力を作成する（form-range-custom.html）

```
<div class="form-group mb-5">
  <label for="formControlRange">レンジ入力の例</label>
  <input type="range" class="form-control-range" id="defaultRange">
</div>
<div class="form-group mb-5">
  <label for="customRange1">カスタムレンジ入力の例</label>
  <input type="range" class="custom-range" id="customRange1">
</div>
<div class="form-group mb-5">
  <label for="customRange2">カスタムレンジ入力の例（min="0",max="5"）</label>
  <input type="range" class="custom-range" min="0" max="5" id="customRange2"> ①
</div>
<div class="form-group mb-5">
  <label for="customRange3">カスタムレンジ入力の例（step="0.5"）</label>
  <input type="range" class="custom-range" min="0" max="5" step="0.5" id="customRange3"> ②
</div>
```

▼図 6-28 カスタムレンジ入力を作成する



レンジ入力は、0～100 の範囲の値を持っています。この範囲を変更するには、min 属性および max 属性を使用します。例では、0～5 までに設定しています（①）。つまみを動かすと 5 分割でスナップすることを確認できます。またレンジ入力は、デフォルトで整数値でスナップします。これを変更するには step 属性で値を指定します。例では、step="0.5" を指定して、ステップ数を 2 倍にしています（②）。

## 6

SECTION

## 2

## 入力グループ

Bootstrap の **入力グループ** は、フォームの入力コントロールとテキストやボタンなどをグループ化して、入力欄を拡張するコンポーネントです。Bootstrap の定義済みクラスを追加するだけで、テキスト入力欄の両側に一体的なスタイルでアドオン（追加機能：たとえば「@」や「URL」など）を配置できます（図 6-29）。他のコンポーネントと同様に、アクセシビリティに配慮しながら入力グループを作成する方法を見ていきましょう。

▼図 6-29 入力グループのアドオンの例 (`input-group-basic-addons.html`)



### 6.2.1 基本的な使用例

入力コントロールの隣にアドオンを配置する場合、アドオンと入力コントロールを、**input-group クラス** を追加した `div` 要素で囲みます。

#### ■ 入力欄の前にアドオンを配置する

入力コントロールの前にアドオンを配置する場合、`span` 要素に **input-group-text クラス** を追加してアドオン部分を作成し、**input-group-prepend クラス** を追加した `div` 要素で囲って、入力コントロールの前に配置します。入力コントロールは `input` 要素に **form-control クラス** を追加して作成します。

また入力コントロールには、属性 `aria-describedby=" (アドオンの ID) "` や `aria-label=" (代替テキスト) "` などを追加し、スクリーンリーダーなどの支援技術に対して、アドオンと入力コントロールとの関連や役割を伝え、アクセシビリティに配慮しましょう（リスト 6-33、図 6-30）。

## ▼リスト 6-33 入力コントロールの前にアドオンを配置 (input-group-basic-addon1.html)

```
<div class="input-group">
  <div class="input-group-prepend">
    <span class="input-group-text" id="basic-addon1">@</span>
  </div>
  <input type="text" class="form-control" placeholder="ユーザー名" aria-label="ユーザー名" aria-describedby="basic-addon1">
</div>
```

## ▼図 6-30 入力コントロールの前にアドオンを配置

textarea 要素による複数行の入力コントロールにアドオンを配置する場合も同様です（リスト 6-34、図 6-31）。

## ▼リスト 6-34 テキストエリアの前にアドオンを配置 (input-group-basic-addon2.html)

```
<div class="input-group">
  <div class="input-group-prepend">
    <span class="input-group-text">テキストエリア用</span>
  </div>
  <textarea class="form-control" aria-label="テキストエリア用"></textarea>
</div>
```

## ▼図 6-31 テキストエリアの前にアドオンを配置

**■ 入力コントロールの後にアドオンを配置する**

入力コントロールの後にアドオンを配置する場合、span 要素に **input-group-text クラス**を追加してアドオン部分を作成し、**input-group-append クラス**を追加した div 要素で囲って、入力コントロールの後に配置します。入力コントロールは input 要素に **form-control クラス**を追加して作成します。前に配置する場合と同様、入力コントロールには、属性 **aria-describedby=" (アドオンの ID) "** や **aria-label=" (代替テキスト) "**などを追加しておきましょう（リスト 6-35、図 6-32）。

## ▼リスト 6-35 入力欄の後にアドオンを配置 (input-group-basic-addon3.html)

```
<div class="input-group">
  <input type="text" class="form-control" placeholder="ユーザー名" aria-label="ユーザー名" aria-describedby="basic-addon2">
  <span class="input-group-text" id="basic-addon2">@</span>
</div>
```

```
aria-describedby="basic-addon3">
<div class="input-group-append">
<span class="input-group-text" id="basic-addon3">@example.com</span>
</div>
</div>
```

▼図 6-32 入力欄の後にアドオンを追加

A screenshot of a web form. It features a horizontal input field with a placeholder 'ユーザー名'. To the right of the input field is a small rectangular box containing the text '@example.com'. This illustrates how an addon is placed to the right of an input field.

## ■ 入力コントロールの前後にアドオンを配置する

入力コントロールの前後両側にアドオンを配置することもできます。前に配置するアドオンは **input-group-prepend** クラスを追加した div 要素で囲み、後ろに配置するアドオンは **input-group-append** クラスを追加した div 要素で囲みます（リスト 6-36、図 6-33）。

▼リスト 6-36 入力コントロールの前後両側にアドオンを配置 (input-group-basic-addon4.html)

```
<!-- 入力コントロールの前後両側にアドオンを追加 -->
<div class="input-group">
<div class="input-group-prepend">
<span class="input-group-text">&yen;</span>
</div>
<input type="text" class="form-control" aria-label="金額">
<div class="input-group-append">
<span class="input-group-text">.00</span>
</div>
</div>
```

▼図 6-33 入力コントロールの前後両側にアドオンを配置

A screenshot of a web form featuring a text input field. On the far left of the input field is a small '\$' symbol. On the far right is a '.00' symbol, representing a decimal separator. This shows how two addons can be placed side-by-side around a central input field.

## ■ ラベル付きの入力グループ

入力グループにラベルを作成する場合は、入力グループの外に label 要素を配置します。なおラベルと入力コントロールは、label 要素の for 属性値と入力コントロールの id 属性値を一致させることで関連付けます（リスト 6-37、図 6-34）。

▼リスト 6-37 ラベル付きの入力グループ (input-group-basic-addon5.html)

```
<label for="basic-url">サイトのURL</label>
<div class="input-group">
```

```
<div class="input-group-prepend">
  <span class="input-group-text" id="basic-addon5">https://example.com/users/</span>
</div>
<input type="text" class="form-control" id="basic-url" aria-describedby="basic-addon5">
</div>
```

▼図 6-34 ラベル付きの入力グループ

サイトのURL  
https://example.com/users/

## 6.2.2 入力グループのサイズ調整

入力グループのサイズを小さくする場合は、**input-group** クラスを設定した親要素に **input-group-sm** クラスを追加します。大きくする場合は、**input-group-lg** クラスを追加します。入力グループ内の内容は自動的にサイズ調整されるため、繰り返しサイズ調整をする必要はありません（リスト 6-38、図 6-35）。

▼リスト 6-38 入力グループのサイズ調整 (input-group-sizing.html)

```
<!-- 小サイズの入力グループ -->
<div class="input-group input-group-sm mb-3">
  <div class="input-group-prepend">
    <span class="input-group-text" id="inputGroup-sizing-sm">Small</span>
  </div>
  <input type="text" class="form-control" aria-label="Small" aria-describedby="inputGroup-sizing-sm">
</div>
...中略...
<!-- 大サイズの入力グループ -->
<div class="input-group input-group-lg">
  <div class="input-group-prepend">
    <span class="input-group-text" id="inputGroup-sizing-lg">Large</span>
  </div>
  <input type="text" class="form-control" aria-label="Large" aria-describedby="inputGroup-sizing-sm">
</div>
```

▼図 6-35 入力グループのサイズ調整

Small  
Default  
Large

### 6.2.3 チェックボックスやラジオボタンのアドオン

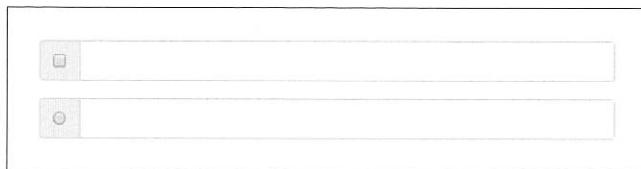
入力グループのアドオンは、テキストだけでなくチェックボックスやラジオボタンも利用できます（リスト 6-39、図 6-36）。

▼リスト 6-39 チェックボックスやラジオボタン付きのアドオン（input-group-checkbox-radio.html）

```
<!-- チェックボックス付きテキスト入力欄 -->
<div class="input-group mb-3">
  <div class="input-group-prepend">
    <div class="input-group-text">
      <input type="checkbox" aria-label="次のテキスト入力用のチェックボックス">
    </div>
  </div>
  <input type="text" class="form-control" aria-label="チェックボックス付きテキスト入力欄">
</div>

<!-- ラジオボタン付きテキスト入力欄 -->
<div class="input-group">
  <div class="input-group-prepend">
    <div class="input-group-text">
      <input type="radio" aria-label="次のテキスト入力用のラジオボタン">
    </div>
  </div>
  <input type="text" class="form-control" aria-label="ラジオボタン付きテキスト入力欄">
</div>
```

▼図 6-36 チェックボックスやラジオボタンのアドオン



### 6.2.4 複数の入力コントロール

入力グループ内に複数の input 要素を含めて、複数の入力コントロールを一体的なスタイルでグループ化することができます。次の例では、氏名の入力欄を、姓と名とに分けた入力グループとして作成しています。

ただし、フォームの入力検証機能（P.210 参照）は、1 つの入力欄に、複数の入力コントロールを含めて扱うことができません。入力検証の対象とする場合は、1 つの入力グループに含める input 要素は 1 つまでにしましょう（リスト 6-40、図 6-37）。

## ▼リスト 6-40 複数の入力コントロール (input-group-multiple-inputs.html)

```
<div class="input-group">
  <div class="input-group-prepend">
    <span class="input-group-text" id="...>姓名</span>
  </div>
  <input type="text" class="form-control" placeholder="姓">
  <input type="text" class="form-control" placeholder="名">
</div>
```

▼図 6-37 複数の入力コントロール

### 6.2.5 複数のアドオンを組み合わせる

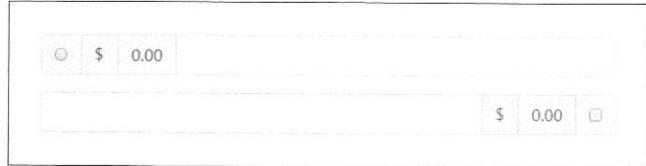
ここまで紹介したテキストやチェックボックスなどを組み合わせて、複数のアドオンを設置することも可能です（リスト 6-41、図 6-38）。

## ▼リスト 6-41 複数のアドオンを組み合わせる (input-group-multiple.html)

```
<div class="input-group mb-3">
  <div class="input-group-prepend">
    <div class="input-group-text">
      <input type="radio" aria-label="次のテキスト入力用のラジオボタン">
    </div>
    <span class="input-group-text">&yen;</span>
    <span class="input-group-text">0.00</span>
  </div>
  <input type="text" class="form-control" aria-label="金額">
</div>

<div class="input-group">
  <input type="text" class="form-control" aria-label="金額">
  <div class="input-group-append">
    <span class="input-group-text">&yen;</span>
    <span class="input-group-text">0.00</span>
    <div class="input-group-text">
      <input type="checkbox" aria-label="前のテキスト入力用のチェックボックス">
    </div>
  </div>
</div>
```

▼図 6-38 複数のアドオンを組み合わせる



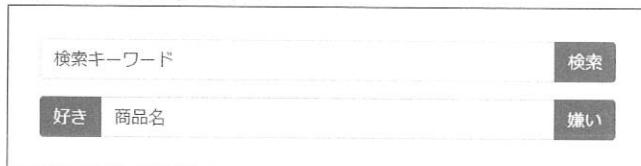
## 6.2.6 ボタン付きアドオン

入力グループのアドオンには、ボタン（P.233 参照）のコンポーネントを組み込むこともできます。ボタン付きアドオンは、検索フォームなどを作成する場合に便利です。ボタンは、button 要素に **btn クラス** および **btn-{色の種類} クラス** を追加して作成します。色の種類には **primary**（青）、**secondary**（グレー）などが入ります（リスト 6-42、図 6-39）。ボタンについての詳細は「ボタン」（P.233）を参照してください。

▼リスト 6-42 ボタン付きアドオン (input-group-button.html)

```
<div class="input-group mb-3">
  <input type="text" class="form-control" placeholder="検索キーワード" aria-label="検索キーワード" ↪
  aria-describedby="basic-addon1">
  <div class="input-group-append">
    <button class="btn btn-secondary" type="button">検索</button>
  </div>
</div>
<div class="input-group mb-3">
  <div class="input-group-prepend">
    <button class="btn btn-secondary" type="button">好き</button>
  </div>
  <input type="text" class="form-control" placeholder="商品名" aria-label="商品名" ↪
  aria-describedby="basic-addon2">
  <div class="input-group-append">
    <button class="btn btn-secondary" type="button">嫌い</button>
  </div>
</div>
```

▼図 6-39 ボタン付きアドオン



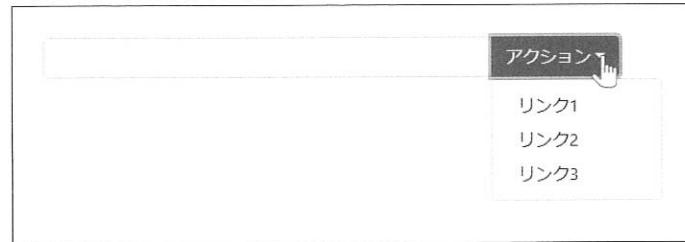
## 6.2.7 ドロップダウン付きアドオン

ボタン付きアドオンと同様に、入力グループのアドオンにドロップダウン（P.244 参照）のコンポーネントを組み込むこともできます。ドロップダウンは、button 要素に **dropdown-toggle クラス** と属性 **data-toggle="dropdown"** を追加して作成します。ドロップダウンの中のメニュー部分は div 要素に **dropdown-menu クラス** を、リンク項目は a 要素に **dropdown-item クラス** を追加して作成します（リスト 6-43、図 6-40）。ドロップダウンについての詳細は「ドロップダウン」（P.244）を参照してください。

▼リスト 6-43 ドロップダウン付きアドオン（input-group-dropdown.html）

```
<div class="input-group">
  <input type="text" class="form-control" aria-label="ドロップダウン付きテキスト入力欄">
  <div class="input-group-append">
    <button class="btn btn-secondary dropdown-toggle" type="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">アクション</button>
    <div class="dropdown-menu">
      <a class="dropdown-item" href="#">リンク1</a>
      <a class="dropdown-item" href="#">リンク2</a>
      <a class="dropdown-item" href="#">リンク3</a>
    </div>
  </div>
</div>
```

▼図 6-40 ドロップダウン付きアドオン



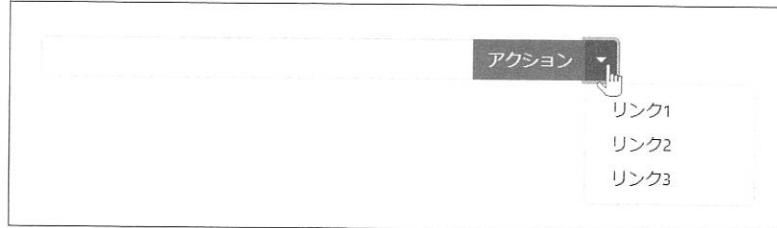
## 6.2.8 スプリットボタンのアドオン

前項のドロップダウン付きアドオンのボタン部分をアレンジし、ボタン内に切り替え用のキャラlettアイコン（▼）とボタン本体とを分離したスプリットボタンにすることもできます。スプリットボタンのアドオンを作成する場合は、ボタン付きアドオン（P.229 参照）と同様のボタンを配置した後に、**dropdown-toggle クラス** と **dropdown-toggle-split クラス** を追加したドロップダウンを配置します（リスト 6-44、図 6-41）。

▼リスト 6-44 スプリットボタンのアドオン (input-group-segmented-buttons.html)

```
<div class="input-group">
  <input type="text" class="form-control" aria-label="スプリットボタン付きテキスト入力欄">
  <div class="input-group-append">
    <button type="button" class="btn btn-secondary">アクション</button>
    <button type="button" class="btn btn-secondary dropdown-toggle dropdown-toggle-split" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
      <span class="sr-only">ドロップダウン切り替え</span>
    </button>
    <div class="dropdown-menu">
      <a class="dropdown-item" href="#">リンク1</a>
      <a class="dropdown-item" href="#">リンク2</a>
      <a class="dropdown-item" href="#">リンク3</a>
    </div>
  </div>
</div>
```

▼図 6-41 スプリットボタンのアドオン



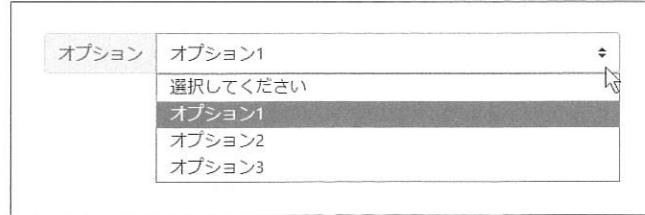
### 6.2.9 カスタムフォームの組み込み

入力グループにカスタムフォーム (P.216 参照) のプルダウンメニューを組み込むことができます。次の例では、入力コントロールとして select 要素に **custom-select** クラスを追加したカスタムフォームのプルダウンを使用しています (リスト 6-45、図 6-42、図 6-43)。

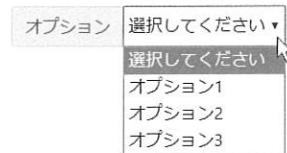
▼リスト 6-45 カスタムフォームのプルダウンを組み込む (input-group-custom-select.html)

```
<div class="input-group mb-3">
  <div class="input-group-prepend">
    <label class="input-group-text" for="inputGroupSelect01">オプション</label>
  </div>
  <select class="custom-select" id="inputGroupSelect01">
    <option selected>選択してください</option>
    <option value="1">オプション1</option>
    <option value="2">オプション2</option>
    <option value="3">オプション3</option>
  </select>
</div>
```

▼図 6-42 カスタムフォームのプルダウンを組み込む



▼図 6-43 標準のプルダウンを組み込んだ場合



なお、カスタムフォーム以外のプルダウンは入力グループとしてサポートされません。select 要素に custom-select クラスを設定しない標準のプルダウンを組み込んだ場合は、サイズやレイアウト、角丸といった入力グループのスタイルが維持されません。

#### 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。

6  
SECTION  
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6-44）。本節では、Bootstrapでボタンを使用する方法を解説します。

▼図6-44 通常のボタンとボタンコンポーネントとの比較 (button-compare.html)

通常のボタン ボタンコンポーネント

## 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.110参照）と同様、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あります。

### 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6-46、図6-45）。

▼リスト6-46 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6-45 ボタンの基本的な使用例



## a要素やinput要素でボタンを作成する

ボタンのコンポーネントは、button要素にクラスを追加して作成するように設計されていますが、a要素やinput要素で作成することもできます。ブラウザによっては若干異なるレンダリングが適用されることがあります。

次の例では、a要素とinput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。a要素でボタンを作成する場合には、アクセシビリティへの配慮として属性**role="button"**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ボタンであることを伝えましょう（リスト6-47、図6-46）。

▼リスト6-47 a要素やinput要素でボタンを作成する(button-tags.html)

```
<a class="btn btn-primary" href="#" role="button">a要素ボタン</a>
<input class="btn btn-primary" type="button" value="inputボタン">
<input class="btn btn-primary" type="submit" value="submitボタン">
<input class="btn btn-primary" type="reset" value="resetボタン">
```

▼図6-46 a要素やinput要素でボタンを作成する



### role属性

role属性は、W3Cが勧告しているアクセシビリティに関する仕様**WAI-ARIA**（Web Accessibility Initiative Accessible Rich Internet Applications）で定義されている属性の1つで、要素の役割を伝えるための属性です。

たとえば、要素に**role="button"**を設定した場合は、その要素がボタンの役割を担っていることをブラウザに伝えることができます。

## 6.3.2 アウトラインボタンを作成する

背景色のないアウトラインで表現されたボタンを作成する場合は、ボタンの要素に設定されたbtn-{ 色の種類 }クラスを**btn-outline-{ 色の種類 } クラス**に置き換えます（リスト6-48、図6-47）。背景色の設定のないアウトラインボタンは、背景画像がある場合でも邪魔になりにくいデザインです。

▼リスト6-48 アウトラインボタンを作成する(button-outline-buttons.html)

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

```
<button type="button" class="btn btn-outline-light">outline-light</button>
<button type="button" class="btn btn-outline-dark">outline-dark</button>
```

▼図 6-47 アウトラインボタンを作成する

`outline-primary` `outline-secondary` `outline-success` `outline-danger` `outline-warning` `outline-info` `outline-dark`



#### アウトラインボタンに背景色を指定するには?

アウトラインボタンに背景色を合わせて指定するには、`btn-{ 色の種類 }` クラスではなく、`bg-{ 色の種類 }` クラスを使用します。

▼リスト 6-49 アウトラインボタンに背景色を指定する

```
<button type="button" class="btn btn-outline-primary bg-warning">アウトラインボタン+背景色</button>
```

### 6.3.3 ボタンサイズを変更する

ボタンのサイズを大きくする場合は、ボタンの要素に **btn-lg** クラスを追加します。小さくする場合は **btn-sm** クラスを追加します（リスト 6-50、図 6-48）。

▼リスト 6-50 ボタンサイズを変更する（button-sizes.html）

```
<button class="btn btn-primary btn-lg">大サイズ</button>
<button class="btn btn-primary">通常サイズ</button>
<button class="btn btn-primary btn-sm">小サイズ</button>
```

▼図 6-48 ボタンサイズを変更する

`大サイズ` `通常サイズ` `小サイズ`

### 6.3.4 ブロックレベルのボタンを作成する

親要素の全幅にまたがるブロックレベルのボタンを作成する場合は、ボタンの要素に **btn-block** クラスを追加します（リスト 6-51、図 6-49）。

▼リスト 6-51 ブロックレベルのボタンを作成する（button-btn-block.html）

```
<button type="button" class="btn btn-primary btn-lg btn-block">ブロックレベルボタン</button>
```

▼図 6-49 ブロックレベルのボタンを作成する

ブロックレベルボタン

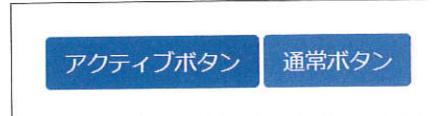
### 6.3.5 アクティブ状態のボタンを作成する

ボタンにアクティブな状態の外観を強制的に作成する場合は、ボタンの要素に **active クラス**を追加します。また、アクセシビリティへの配慮として属性 **aria-pressed="true"** を追加し、スクリーンリーダーなどの支援技術にこのボタンがアクティブな状態であることを伝えましょう（リスト 6-52、図 6-50）。

▼リスト 6-52 アクティブ状態のボタンを作成する（button-active-state.html）

```
<a href="#" class="btn btn-primary btn-lg active" role="button" aria-pressed="true">アクティブボタン</a>
<a href="#" class="btn btn-primary btn-lg" role="button">通常ボタン</a>
```

▼図 6-50 アクティブ状態のボタンを作成する



### 6.3.6 無効状態のボタンを作成する

クリックできない無効な状態のボタンを作成する場合は、ボタンの要素に **disabled 属性**を追加します（リスト 6-53、図 6-51）。

▼リスト 6-53 無効状態のボタンを作成する（button-disabled-state.html）

```
<button type="button" class="btn btn-lg btn-primary" disabled>無効ボタン</button>
<button type="button" class="btn btn-lg btn-primary">通常ボタン</button>
```

▼図 6-51 無効状態のボタンを作成する



ただし、a 要素は disabled 属性をサポートしていません。そのため、a 要素で無効な状態のボタンを作成する場合は、ボタンの要素に **disabled クラス**を追加します。また、アクセシビリティへの配慮として属性 **aria-disabled="true"** を追加し、スクリーンリーダーなどの支援技術にこのボタンが無効な状態であることを伝えます（リスト 6-54、図 6-52）。

▼リスト 6-54 a 要素で無効状態のボタンを作成する（button-disabled-anchor.html）

```
<a href="#" class="btn btn-primary btn-lg disabled" role="button" aria-disabled="true">無効ボタン</a>
<a href="#" class="btn btn-primary btn-lg" role="button">通常ボタン</a>
```

▼図 6-52 a 要素で無効状態のボタンを作成する



NOTE

**disabled クラスに定義されているスタイル**

**disabled** クラスには、a 要素のリンク機能を無効にするスタイル **pointer-events: none** が定義されています。このプロパティはまだ標準化されていないため、ブラウザによってはボタンが無効化されません（2017年12月現在）。また、このプロパティをサポートしているブラウザであっても、キーボードナビゲーション（マウスを使用しないキーボードのみの操作）は対象外となります。キーボードナビゲーションにおいてもボタンを無効化する場合は、タブキーでの移動によるフォーカスを受けないようにする属性 **tabindex="-1"** を追加し、JavaScript を使用して無効化する必要があります。

**6.3.7 切り替えボタンを作成する**

切り替えボタンを作成するには、ボタン要素に属性 **data-toggle="button"** を追加します。また、ブラウザの入力補完機能を無効にするために属性 **autocomplete="off"** を追加します。

ボタンをあらかじめプッシュ状態に切り替えておく場合は、ボタン要素に **active** クラスを追加し、支援技術にボタンが押されている状態であることを伝えるための属性 **aria-pressed="true"** を追加します。反対に、未プッシュ状態にしておくときは、属性 **aria-pressed="false"** を追加します（リスト 6-55、図 6-53）。

▼リスト 6-55 切り替えボタンを作成する (button-toggle.html)

```
<button type="button" class="btn btn-primary active" data-toggle="button" aria-pressed="true" ↪
  autocomplete="off">プッシュ状態</button>
<button type="button" class="btn btn-primary" data-toggle="button" aria-pressed="false" ↪
  autocomplete="off">未プッシュ状態</button>
```

▼図 6-53 切り替えボタンを作成する



### 6.3.8 チェックボックスとラジオボタンを作成する

フォームの入力コントロールであるチェックボックスやラジオボタンの外観を、ボタンのスタイルに変更することができます。チェックボックスもラジオボタンも、フォームのlabel要素に、ボタン用の**btnクラス**と**btn-{ 色の種類 }クラス**を追加します。また一連のボタンの親要素には、ボタングループ（P.240 参照）用の**btn-groupクラス**と、属性**data-toggle="buttons"**を追加します。

#### ■ チェックボックスをスタイル変更する

次の例では、チェックボックスの外観をボタンのスタイルに変更しています（リスト6-56、図6-54）。

▼リスト6-56 チェックボックスをボタンのスタイルに変更する（button-checkbox.html）

```
<div class="btn-group" data-toggle="buttons">
  <label class="btn btn-secondary active">
    <input type="checkbox" checked autocomplete="off"> チェックボックス1（選択済み）
  </label>
  <label class="btn btn-secondary">
    <input type="checkbox" autocomplete="off"> チェックボックス2
  </label>
  <label class="btn btn-secondary">
    <input type="checkbox" autocomplete="off"> チェックボックス3
  </label>
</div>
```

▼図6-54 チェックボックスをボタンコンポーネントのスタイルに変更する



□チェックボックス1（選択済み） ■チェックボックス2 ■チェックボックス3

#### ■ ラジオボタンをスタイル変更する

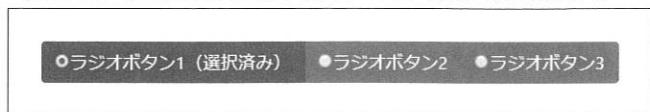
次の例では、ラジオボタンの外観をボタンのスタイルに変更しています（リスト6-57、図6-55）。

▼リスト6-57 ラジオボタンをボタンのスタイルに変更する（button-radio.html）

```
<div class="btn-group" data-toggle="buttons">
  <label class="btn btn-secondary active">
    <input type="radio" checked autocomplete="off"> ラジオボタン1（選択済み）
  </label>
  <label class="btn btn-secondary">
    <input type="radio" autocomplete="off"> ラジオボタン2
  </label>
  <label class="btn btn-secondary">
    <input type="radio" autocomplete="off"> ラジオボタン3
  </label>
</div>
```

&lt;/div&gt;

▼図 6-55 ラジオボタンをボタンコンポーネントのスタイルに変更する



### 6.3.9 メソッド

JavaScriptでボタンの動作を指定するメソッドには表 6-2 の 2つがあります。

▼表 6-2 JavaScript でボタンの動作を指定するメソッド

メソッド	概要
\$(().button('toggle'))	ボタンのプッシュ状態を切り替え、アクティビ化された外観を与える
\$(().button('dispose'))	ボタンを破棄する

次の例では、ボタンをクリックするとプッシュ状態の表示に切り替わり、ボタンのテキストが「押してね」から「押したね」に変化します（リスト 6-58、図 6-56）。

▼リスト 6-58 クリックするとボタンをプッシュ状態に切り替える（button-method.html）

```
<button type="button" class="btn btn-primary" aria-pressed="true" autocomplete="off" id="toggle-btn">押してね</button>
...
<script>
  $("#toggle-btn").on('click',function(){
    $(this).button('toggle').text('押したね');
  });
</script>
```

▼図 6-56 クリックするとボタンをプッシュ状態に切り替える：クリック前（左）、クリック後（右）



Bootstrap の**ボタングループ**は、一連のボタンのコンポーネントをグループ化して一体的に表示するためのコンポーネントです。本節では**ボタングループ**を使用する方法を解説します。

### 6.4.1 基本的な使用例

ボタンのコンポーネントをグループ化した**ボタングループ**を作成する場合は、div 要素に **btn-group クラス** を追加して一連のボタンを囲みます。またスクリーンリーダーなどの支援技術に対して、コンポーネントの役割がグループであることを伝える属性 **role="group"** と、ラベル付けを行う **aria-label 属性** を追加してアクセシビリティに配慮します（リスト 6-59、図 6-57）。

▼リスト 6-59 ボタングループの基本的な使用例 (btn-group-basic.html)

```
<div class="btn-group" role="group" aria-label="基本的な使用例">
  <button type="button" class="btn btn-secondary">左ボタン</button>
  <button type="button" class="btn btn-secondary">中ボタン</button>
  <button type="button" class="btn btn-secondary">右ボタン</button>
</div>
```

▼図 6-57 ボタングループの基本的な使用例

### 6.4.2 ボタンツールバーを作成する

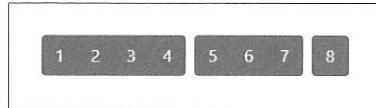
複数のボタングループを組み込んだより複雑なボタングループを、**ボタンツールバー**として簡単に作成することができます。

ボタンツールバーを作成する場合は、div 要素に **btn-toolbar クラス** を追加して、一連のボタングループの要素を囲みます。またスクリーンリーダーなどの支援技術に対して、コンポーネントの役割がツールバーであることを伝える属性 **role="toolbar"** と、ラベル付けを行う **aria-label 属性** を追加してアクセシビリティに配慮します。なお次の例では、ボタングループ間が詰まり過ぎて見にくくならないように、Spacing ユーティリティ（P.318 参照）の **mr-2 クラス** を使用してマージンを調整しています（リスト 6-60、図 6-58）。

▼リスト 6-60 ボタンツールバーを作成する (btn-group-btn-toolbar.html)

```
<div class="btn-toolbar" role="toolbar" aria-label="ボタンツールバー">
  <!-- ボタングループ1 -->
  <div class="btn-group mr-2" role="group" aria-label="ボタングループ1">
    ...中略...
  </div>
  <!-- ボタングループ2 -->
  <div class="btn-group mr-2" role="group" aria-label="ボタングループ2">
    ...中略...
  </div>
  <!-- ボタングループ3 -->
  <div class="btn-group" role="group" aria-label="ボタングループ3">
    ...中略...
  </div>
</div>
```

▼図 6-58 ボタンツールバーを作成する



## ■ 入力グループを組み込む

ボタンツールバーには、ボタングループだけでなく入力グループ（P.223 参照）のコンポーネントを組み込むこともできます。ボタングループと同様に、div 要素に **btn-toolbar クラス** を設定したボタンツールバーで、div 要素に **input-group クラス** を設定した入力グループを囲みます（リスト 6-61、図 6-59）。

▼リスト 6-61 入力グループを組み込む (btn-group-mix-groups.html)

```
<!-- ボタンツールバー -->
<div class="btn-toolbar mb-3" role="toolbar" aria-label="ボタンツールバー">
  <!-- ボタングループ -->
  <div class="btn-group mr-2" role="group" aria-label="ボタングループ">
    ...中略...
  </div>
  <!-- 入力グループ -->
  <div class="input-group">
    <div class="input-group-prepend">
      <div class="input-group-text" id="btnGroupAddon">@</div>
    </div>
    <input type="text" class="form-control" placeholder="入力グループの例" aria-label="入力グループの例" aria-describedby="btnGroupAddon">
  </div>
</div>
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

▼図 6-59 入力グループを組み込む



### 6.4.3 サイズを変更する

ボタングループのサイズを大きくする場合は、ボタングループの要素に **btn-group-lg** クラスを追加します。小さくする場合は **btn-group-sm** クラスを追加します（リスト 6-62、図 6-60）。

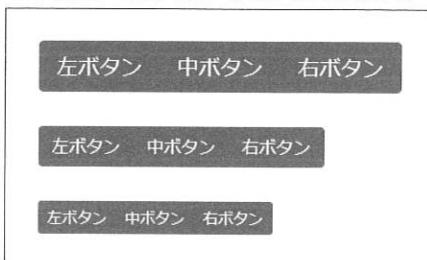
▼リスト 6-62 ボタングループのサイズを変更する (btn-group-sizing.html)

```
<!-- 大サイズ -->
<div class="btn-group btn-group-lg" role="group" aria-label="グループ1">
...中略...
</div>

<!-- 標準サイズ -->
<div class="btn-group" role="group" aria-label="グループ2">
...中略...
</div>

<!-- 小サイズ -->
<div class="btn-group btn-group-sm" role="group" aria-label="グループ3">
...中略...
</div>
```

▼図 6-60 ボタングループのサイズを変更する



### 6.4.4 ドロップダウンメニューを入れ子にする

ボタングループの中にドロップダウン（P.244 参照）を含める場合は、切り替えボタンとドロップダウンメニューをグループ化するボタングループを作成して、全体のボタングループの入れ子にします（リスト 6-63、図 6-61）。

▼リスト 6-63 ドロップダウンメニューを入れ子にする (btn-group-nesting.html)

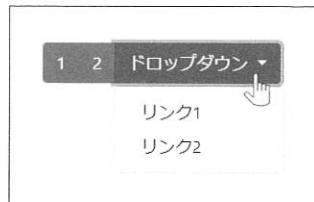
```
<div class="btn-group" role="group" aria-label="ドロップダウンを含むボタングループ">
```

```

<button type="button" class="btn btn-secondary">1</button>
<button type="button" class="btn btn-secondary">2</button>
<div class="btn-group" role="group">
  <!-- 切り替えボタン -->
  <button id="drop1" type="button" class="btn btn-secondary dropdown-toggle" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
    ドロップダウン
  </button>
  <!-- ドロップダウンメニュー -->
  <div class="dropdown-menu" aria-labelledby="drop1">
    <a class="dropdown-item" href="#">リンク1</a>
    <a class="dropdown-item" href="#">リンク2</a>
  </div>
</div>

```

▼図 6-61 ドロップダウンメニューを入れ子にする



### 6.4.5 垂直方向のボタングループを作成する

垂直方向に縦並びになるボタングループを作成する場合は、ボタングループの要素に設定された `btn-group` クラスを **btn-group-vertical** クラスに置き換え、垂直方向にグループ化します（リスト 6-64、図 6-62）。

ただし垂直方向にグループ化した場合、ボタン内に切り替え用のキャレットアイコン（▼）とボタン本体とを分離したスプリットボタンのドロップダウン（P.246 参照）は使えません。

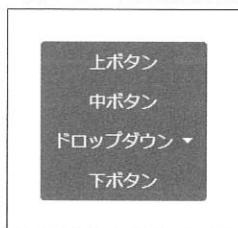
▼リスト 6-64 垂直方向のボタングループを作成する (btn-group-vertical.html)

```

<div class="btn-group-vertical">
  ...中略...
</div>

```

▼図 6-62 垂直方向のボタングループ





## 5 ドロップダウン

Bootstrap の**ドロップダウン**は、ボタンをクリックすることでドロップダウンメニューを表示するコンポーネントです。本節では**ドロップダウン**を使用する方法を解説します。なおこのコンポーネントは、サードパーティの JavaScript ライブラリ Popper.js 上に構築されています。各種 JavaScript を読み込む際は bootstrap.js の前に popper.js (または軽量版の popper.min.js) を読み込むか、popper.js を組み込み済みの bootstrap.bundle.js を使用する必要があります (P.16 参照)。

### 6.5.1 基本的な使用例

ドロップダウンの基本的な使用例を見ていきましょう。このコンポーネントは、button 要素または a 要素で作成した切り替えボタンと、ドロップダウンメニューとで構成されています。

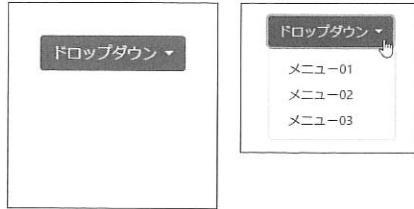
#### ■ 切り替えボタンに button 要素を使用したドロップダウン

次の例は、ドロップダウンの切り替えボタンに button 要素を使用した例です (リスト 6-65、図 6-63)。

▼リスト 6-65 ドロップダウンの基本的な使用例 (dropdown-basic-button.html)

```
<!-- ドロップダウン -->
<div class="dropdown"> _____①
  <!-- 切り替えボタン -->
  <button class="btn btn-secondary dropdown-toggle" type="button" id="dropdownMenuButton" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false"> _____②
    ドロップダウン
  </button>
  <!-- ドロップダウンメニュー -->
  <div class="dropdown-menu" aria-labelledby="dropdownMenuButton"> _____③
    <a class="dropdown-item" href="#">メニュー01</a> _____②
    <a class="dropdown-item" href="#">メニュー02</a> _____②
    <a class="dropdown-item" href="#">メニュー03</a> _____②
  </div>
</div>
```

▼図 6-63 通常時（左）、ドロップダウン表示時（右）



ドロップダウンを作成する場合は、**dropdown クラス**を設定した親要素で、切り替えボタンとドロップダウンメニューとを囲みます（①）。

切り替えボタンは、button 要素または a 要素に **btn クラス**と **btn-{ 色の種類 } クラス**を設定した基本的なボタン（P.233 参照）のコンポーネントに、**dropdown-toggle クラス**と属性 **data-toggle="dropdown"** を追加して作成します（②）。

ドロップダウンメニューは、まず **dropdown-menu クラス**を設定した親要素でメニューの枠を作成します（③）。次に **dropdown-item クラス**を設定した子要素でメニューの項目を作成します（④）。

#### NOTE

#### ドロップダウンのアクセシビリティ

ドロップダウンを作成する場合、アクセシビリティへの配慮として **aria-\* 属性**を追加し、スクリーンリーダーなどの支援技術に対してコンポーネントの状態を伝えましょう。

切り替えボタンには表 6-3 の属性を追加します。

▼表 6-3 切り替えボタンに追加する属性

属性	概要
<code>aria-haspopup="true"</code>	ポップアップが含まれていることを伝える
<code>aria-expanded="false"</code>	切り替えが非表示の状態であることを伝える

ドロップダウンメニューには表 6-4 の属性を追加します。

▼表 6-4 ドロップダウンメニューに追加する属性

属性	概要
<code>aria-labelledby=" (切り替えボタンの ID) "</code>	値が要素のラベルとして関連付いていることを伝える

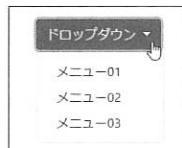
### 切り替えボタンに a 要素を使用したドロップダウン

次の例では、a 要素を使用してドロップダウンの切り替えボタンを作成しています。先の例と同様に、ボタンの要素に **dropdown-toggle クラス**と属性 **data-toggle="dropdown"** を追加して切り替えボタンを作成することができます。ただしアクセシビリティへの配慮として、コンポーネントの役割がボタンであることをスクリーンリーダーなどの支援技術に伝える属性 **role="button"** を追加する必要があります（リスト 6-66、図 6-64）。

## ▼リスト 6-66 a 要素を使用したドロップダウンの切り替えボタン (dropdown-basic-link.html)

```
<!-- 切り替えボタン -->
<a class="btn btn-secondary dropdown-toggle" href="#" role="button" id="dropdownMenuLink" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
  ドロップダウン
</a>
```

▼図 6-64 切り替えボタンに a 要素を使用したドロップダウン



## ■ ボタングループに組み込むドロップダウン

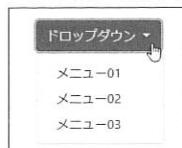
dropdown クラスの代わりに **btn-group** クラスを使用して、ボタングループ（P.240 参照）にドロップダウンを組み込むことができます（リスト 6-67、図 6-65）。詳しくは「ボタングループ」（P.240）を参照してください。

## ▼リスト 6-67 ボタングループに組み込むドロップダウン (dropdown-btn-group.html)

```
<!-- ボタングループ -->
<div class="btn-group">
  <!-- 切り替えボタン -->
  <button type="button" class="btn btn-secondary dropdown-toggle" data-toggle="dropdown" data-haspopup="true" aria-expanded="false">
    ドロップダウン
  </button>
  <!-- ドロップダウンメニュー -->
  <div class="dropdown-menu">
    <a class="dropdown-item" href="#">メニュー-01メニュー-02メニュー-03

```

▼図 6-65 ボタングループに組み込むドロップダウン



## ■ スプリットボタンのドロップダウン

ボタン内に切り替え用のキャレットアイコン（▼）とボタン本体と分離した**スプリットボタン**のドロップダウンを作成することができます（リスト 6-68、図 6-66）。

▼リスト 6-68 スプリットボタンのドロップダウン (dropdown-toggle-split.html)

```

<!-- ボタングループ -->
<div class="btn-group">
  <!-- ボタン本体 -->
  <button type="button" class="btn btn-secondary"> ①
    スプリットボタン
  </button>
  <!-- 切り替え用アイコン -->
  <button type="button" class="btn btn-secondary dropdown-toggle dropdown-toggle-split" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false"> ②
    <span class="sr-only">ドロップダウン切り替え</span> ③
  </button>
  <!-- ドロップダウンメニュー -->
  <div class="dropdown-menu">
    ...中略...
  </div>
</div>

```

▼図 6-66 通常時（左）、ドロップダウン表示時（右）



まずボタングループ内に **btn クラス** と **btn-{ 色の種類 } クラス** を設定した基本的なボタンのコンポーネントを、スプリットボタンの本体として配置します（①）。次に切り替え用アイコンとなるボタンのコンポーネントを配置し、**dropdown-toggle クラス** と **dropdown-toggle-split クラス** を追加します（②）。またアクセシビリティへの配慮として、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** を使用し、スクリーンリーダー用の非表示テキストで切り替えアイコンの役割を記述しておきましょう（③）。

### 6.5.2 ドロップダウン方向を変更する

ドロップダウン方向を下から上に変更する場合は、**btn-group** クラスが設定された要素に **dropup クラス** を追加します。同様に、右方向に変更する場合は **dropright クラス** を、左方向に変更する場合は **dropleft クラス** を追加します。ドロップダウン方向の変更は、スプリットボタンでも同様に設定することができます。

次の例では、**dropup クラス** を追加して、ドロップダウン方向を上に変更しています（リスト 6-69、図 6-67）。

▼リスト 6-69 上方向へのドロップ (dropdown-dropup.html)

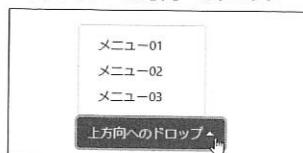
```

<!-- 上方向へのドロップ -->
<div class="btn-group dropup">

```

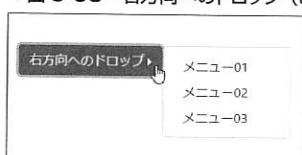
```
<!-- 切り替えボタン -->
<button type="button" class="btn btn-secondary dropdown-toggle" data-toggle="dropdown" ↪
aria-haspopup="true" aria-expanded="false">上方向へのドロップ</button>
<!-- ドロップダウンメニュー -->
<div class="dropdown-menu" aria-labelledby="dropdownMenuLink">
…中略…
</div>
</div>
```

▼図 6-67 上方向へのドロップ



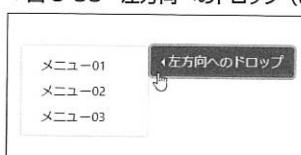
同様に **dropright** クラスを追加して、ドロップダウン方向を右に変更できます（図 6-68）。

▼図 6-68 右方向へのドロップ (dropdown-dropright.html)



また **dropleft** クラスを追加して、ドロップダウン方向を左に変更できます（図 6-69）。

▼図 6-69 左方向へのドロップ (dropdown-dropleft.html)



### 6.5.3 メニュー項目のリンクに使用できる要素

ドロップダウンメニューの各項目には、a 要素だけでなく button 要素も使用することができます（Bootstrap 3 では a 要素のみが使用できます）。次の例では、ドロップダウンのメニュー項目に button 要素を使用しています（リスト 6-70、図 6-70）。

▼リスト 6-70 メニュー項目に button 要素を使用する (dropdown-items-button.html)

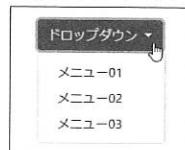
```
<!-- ドロップダウン -->
<div class="dropdown">
<!-- 切り替えボタン -->
```

```

<button class="btn btn-secondary dropdown-toggle" type="button" id="dropdownMenuButton" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
  ドロップダウン
</button>
<!-- ドロップダウンメニュー -->
<div class="dropdown-menu" aria-labelledby="dropdownMenuButton">
  <button class="dropdown-item" type="button">メニュー-01</button>
  <button class="dropdown-item" type="button">メニュー-02</button>
  <button class="dropdown-item" type="button">メニュー-03</button>
</div>
</div>

```

▼図 6-70 メニュー項目に button 要素を使用する



#### 6.5.4 メニューの位置揃えを変更する

ドロップダウンの初期設定では、ドロップダウンメニューと切り替えボタンは左揃えに配置されます。これを右揃えに変更する場合は、dropdown-menu クラスが設定された要素に **dropdown-menu-right クラス**を追加します（リスト 6-71、図 6-71）。

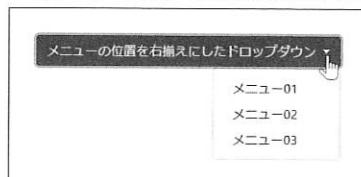
▼リスト 6-71 メニューの位置をボタンと右揃えにする（dropdown-alignment.html）

```

<!-- ボタングループ -->
<div class="btn-group">
  <!-- 切り替えボタン -->
  <button type="button" class="btn btn-secondary dropdown-toggle" data-toggle="dropdown" data-hasspopup="true" aria-expanded="false">
    メニューの位置を右揃えにしたドロップダウン
  </button>
  <!-- ドロップダウンメニュー -->
  <div class="dropdown-menu dropdown-menu-right">
    ...中略...
  </div>
</div>

```

▼図 6-71 メニューの位置をボタンと右揃えにする



## 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

ドロップダウンメニュー内には、メニュー項目以外にもさまざまな要素を組み込むことができます。

次の例では、ドロップダウンメニュー内に見出しやフォーム、区切り線などを組み込んでいます（リスト6-72、図6-72）。

▼リスト6-72 ドロップダウンメニューにさまざまな要素を追加する（dropdown-includes.html）

```
<!-- ボタングループ -->
<div class="btn-group">
  <!-- 切り替えボタン -->
  <button type="button" class="btn btn-secondary dropdown-toggle" data-toggle="dropdown" data-aria-haspopup="true" data-aria-expanded="false">
    ドロップダウン
  </button>
  <!-- ドロップダウンメニュー -->
  <div class="dropdown-menu">
    <h6 class="dropdown-header">ドロップダウンメニューの見出し</h6> -----①
    <form class="px-4 py-3"> -----②
      ...中略...
    </form>
    <div class="dropdown-divider"></div> -----③
    <a class="dropdown-item" href="#">初めての方</a>
    <a class="dropdown-item" href="#">パスワードをお忘れの方</a>
  </div>
</div>
```

▼図6-72 ドロップダウンメニューにさまざまな要素を追加する



ドロップダウンメニュー内に見出しを配置する場合は、見出し要素に **dropdown-header クラス** を追加します（①）。

ドロップダウンメニュー内にフォームを配置する場合は、特に追加するべきクラスはありません（②）。ただし、必要に応じて Spacing ユーティリティ（P.318 参照）を設定してマージンやパディングを調整し、フォームが読みやすくなるようにレイアウトを整えましょう。この例では **px-4 クラス** で水平方向のパディングを、**py-3 クラス**

で垂直方法のパディングをサイズ調整しています。

ドロップダウンメニューの各項目を分ける区切り線を作成する場合は、div要素に**dropdown-dividerクラス**を追加します（❸）。

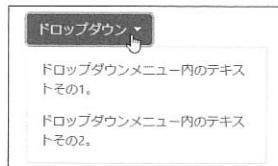
### 6.5.6 ドロップダウンメニューに自由形式のテキストを配置する 4.1

ドロップダウンメニュー内には、p要素など自由形式のテキストを配置できます。メニュー幅を制限するには、サイズを指定するスタイルを追加する必要があります。次の例では、ドロップダウンメニューにSpacingユーティリティ（P.318参照）のp-3クラスを指定して1rem分のパディングを付けています。また、属性style="max-width: 300px;"を指定し、メニューの最大幅が300pxになるよう制限しています（リスト6-73、図6-73）。

▼リスト6-73 メニュー項目に自由形式のテキストを配置する（dropdown-text.html）

```
<div class="dropdown-menu p-3 text-muted" style="max-width: 300px;">
  <p>ドロップダウンメニュー内のテキストその1。</p>
  <p class="mb-0">ドロップダウンメニュー内のテキストその2。</p>
</div>
```

▼図6-73 メニュー項目に自由形式のテキストを配置する



### 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

メニュー項目に、クリックできない無効な状態のスタイルを設定する場合は、dropdown-itemクラスが設定された要素に**disabledクラス**を追加します。また、アクティブな状態のスタイルを設定する場合は、**activeクラス**を追加します（リスト6-74、図6-74）。

▼リスト6-74 メニュー項目に無効やアクティブの状態を設定する（dropdown-disabled-active.html）

```
<!-- メニュー -->
<div class="dropdown-menu">
  <a class="dropdown-item" href="#">メニュー01</a>
  <a class="dropdown-item disabled" href="#">メニュー02（無効）</a>
  <a class="dropdown-item active" href="#">メニュー03（アクティブ）</a>
</div>
```

▼図 6-74 メニュー項目に無効やアクティブの状態を設定する



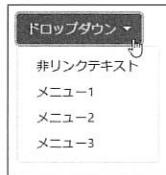
### 6.5.8 ドロップダウンにリンクなしのメニュー項目を追加する 4.1

ドロップダウンメニューの項目に **dropdown-item-text** クラスを追加して、リンクなしのメニュー項目を作成できます（リスト 6-75、図 6-75）。

▼リスト 6-75 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（dropdown-no-link.html）

```
<div class="dropdown-menu">
  <span class="dropdown-item-text">非リンクテキスト</span>
  <a class="dropdown-item" href="#">メニュー-1</a>
  <a class="dropdown-item" href="#">メニュー-2</a>
  <a class="dropdown-item" href="#">メニュー-3</a>
</div>
```

▼図 6-75 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



## 6

SECTION

## 6

# ドロップダウンの JavaScript 使用

既述のように、ドロップダウンのコンポーネントを利用するには、データ属性経由で `data-toggle="dropdown"` を指定します。本節では、ドロップダウンコンポーネントを JavaScript コードで呼び出しますが、この場合でも、`data-toggle="dropdown"` は必要となりますので注意してください。

## 6.6.1

## ドロップダウンのオプション

ドロップダウンで定義されているオプションは表 6-5 のとおりです。オプションは、データ属性または JavaScript を使用して渡すことができます。データ属性の場合、`data-offset="10"` のように `data-` にオプション名を追加します。

▼表 6-5 ドロップダウンのオプション

オプション名	値	説明
offset	px 値や % 値など。単位なしは px と見なされる	ドロップダウンを配置する位置を指定
flip	true、false	メニューを表示する領域がない場合には反対側に表示
boundary	"scrollParent"、"viewport"、"window"、または任意の DOM 要素	ドロップダウンメニューのオーバーフローを制約する境界
reference	"toggle"、"parent" または任意の DOM 要素	ドロップダウンメニューを表す要素
display	"dynamic"、"static"	デフォルトでは Popper.js で、ドロップダウンを動的に位置決めして配置 (dynamic)。static でこれを無効にする

### offset

`offset` はターゲットに対するドロップダウンのオフセット値（配置する位置）を設定するオプションです。データ属性を使ってオフセット値を指定するには、オプション名「`offset`」の前に「`data-`」を付けて指定します。`data-offset="10"` のように値が 1 つの場合は横軸を指定したことになり、この場合は右へ 10px ずらして配置されます。`data-offset="10, 20"` を指定すると右へ 10px、下へ 20px ずらした位置に配置されます（リスト 6-76、図 6-76）。

▼リスト 6-76 ドロップダウンのオフセット値を設定する (dropdown-option-offset.html)

```
<div class="dropdown">
  <button type="button" class="btn btn-secondary dropdown-toggle" id="dropdownMenuButton" ↪
    data-toggle="dropdown" data-offset="10, 20" aria-haspopup="true" aria-expanded="false">↪
    ドロップダウンボタン</button>
  <div class="dropdown-menu" aria-labelledby="dropdownMenuButton">
```

```
<a class="dropdown-item" href="#">メニュー1
```

```
<a class="dropdown-item" href="#">メニュー2
```

```
<a class="dropdown-item" href="#">メニュー3
```

```
</div>
```

```
</div>
```

▼図 6-76 ドロップダウンのオフセット値を設定する



また JavaScript コードでオプションを渡す場合は、リスト 6-77 のような書式で記述します。

▼リスト 6-77 ドロップダウンのオフセット値を設定する (dropdown-option-offset.js.html)

```
<script>
$(function(){
    $('#dropdownMenuButton').dropdown({offset:'10, 20'})
});
</script>
```

## ■ flip

`flip` は、ドロップダウンメニューを表示させるスペースがない場合に、反対側に表示させるオプションです。`true` または `false` で設定します（リスト 6-78、図 6-77）。

▼リスト 6-78 ドロップダウンメニューを表示させるスペースがない場合に、反対側に表示させる (dropdown-option-flip.html)

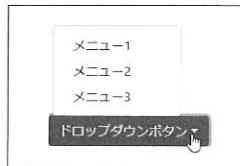
```
<div class="box" style="overflow: auto">
<br><br><br><br><br>
<div class="dropdown">
    <button class="btn btn-secondary dropdown-toggle" data-toggle="dropdown" data-flip="true">⇨
        ドロップダウンボタン
    </button>
    <div class="dropdown-menu">
        <a class="dropdown-item" href="#">メニュー1
```

```
<a class="dropdown-item" href="#">メニュー2
```

```
<a class="dropdown-item" href="#">メニュー3
```

```
</div>
</div>
</div>
```

▼図 6-77 ドロップダウンメニューを表示させるスペースがない場合に、反対側に表示させる



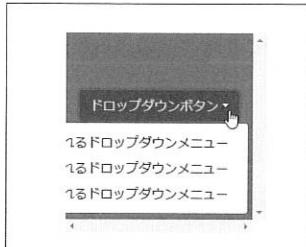
## boundary

boundary は、ドロップダウンメニューの表示領域の境界を設定します。次の例では、「scrollParent」を指定し、ドロップダウンメニューの表示をスクロールルウィンドウ内に制約しています。スクロールルウィンドウからはみ出る部分は表示されません（リスト 6-79、図 6-78）。

▼リスト 6-79 ドロップダウンメニューのオーバーフローを制約する境界を設定する（dropdown-option-boundary.html）

```
<div class="bg-info text-center" style="width:250px;height:250px;overflow:scroll;">
<br><br><br>
<div class="dropdown">
<button class="btn btn-secondary dropdown-toggle" data-toggle="dropdown" data-boundary=<-->
"scrollParent">ドロップダウンボタン</button>
<div class="dropdown-menu">
<a href="#" class="dropdown-item">スクロールルウィンドウ幅で制約されるドロップダウンメニュー</a>
<a href="#" class="dropdown-item">スクロールルウィンドウ幅で制約されるドロップダウンメニュー</a>
<a href="#" class="dropdown-item">スクロールルウィンドウ幅で制約されるドロップダウンメニュー</a>
</div>
</div>
</div>
```

▼図 6-78 ドロップダウンメニューのオーバーフローを制約する境界を設定する



### 6.6.2 ドロップダウンのメソッド

ドロップダウンで定義されているメソッドは表 6-6 のとおりです。

▼表 6-6 ドロップダウンのメソッド

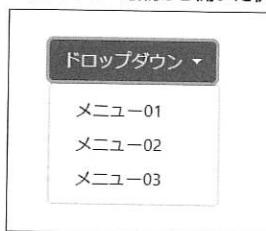
メソッド	説明
<code>\$(()).dropdown('toggle')</code>	ドロップダウンを開閉
<code>\$(()).dropdown('update')</code>	要素のドロップダウンの位置を更新
<code>\$(()).dropdown('dispose')</code>	要素のドロップダウンを破棄

次の例では、`$(()).dropdown('toggle')` を使用しています。通常はメニューが閉じられていてクリックすると開いた状態になりますが、最初からメニューが開いています（リスト 6-80、図 6-79）。

▼リスト 6-80 ドロップダウンのメソッド：`$(()).dropdown('toggle')` (dropdown-method.html)

```
<script>
$(function(){
    $('#dropdownMenuButton').dropdown('toggle');
});
</script>
```

▼図 6-79 最初から開いた状態のドロップダウンメニュー



### 6.6.3 ドロップダウンのイベント

ドロップダウンで定義されているイベントは表 6-7 のとおりです。ドロップダウンを開く直前や直後に何らかの処理を行いたい場合に使用します。

▼表 6-7 ドロップダウンのイベント

イベント	説明
<code>show.bs.dropdown</code>	ドロップダウンを開く直前に発動
<code>shown.bs.dropdown</code>	ドロップダウンを開いた直後に発動
<code>hide.bs.dropdown</code>	ドロップダウンを閉じる直前に発動
<code>hidden.bs.dropdown</code>	ドロップダウンが閉じた直後に発動

これらのイベントでは、`relatedTarget` プロパティに、`dropdown` クラスを指定した要素のオブジェクトが格納されます。次の例では、ドロップダウンを開く直前にボタンの文字列を取得し、アラートに表示します（リスト 6-81、図 6-80）。

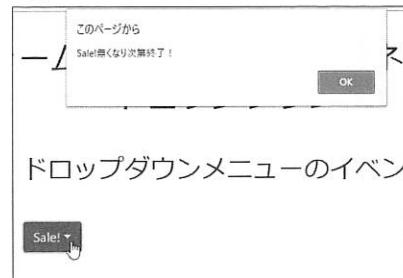
## ▼リスト 6-81 ドロップダウンのイベント: show.bs.dropdown (dropdown-event.html)

```

<div class="dropdown">
  <button class="btn btn-primary dropdown-toggle" type="button" id="dropdownExample" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">Sale!</button>
  <div class="dropdown-menu" aria-labelledby="dropdownExample">
    <a class="dropdown-item" href="#">メニュー01</a>
    <a class="dropdown-item" href="#">メニュー02</a>
    <a class="dropdown-item" href="#">メニュー03</a>
  </div>
</div>
…中略…
<script>
$(function(){
  $(".dropdown").on("show.bs.dropdown", function(event){
    var x = $(event.relatedTarget).text(); // ボタンのテキストを取得
    alert(x + 'なくなり次第終了！');
  });
});
</script>

```

▼図 6-80 ドロップダウンメニュー項目が表示される前にアラートを表示



1  
2  
3  
4  
5  
6  
7  
8  
9  
10

## COLUMN 公式サイトで最新情報を確認する

本書の執筆時点でのBootstrapは「v4.1.1」が最新版（2018年4月30日リリース）ですが、ここに至るまでメジャー、マイナーに関わらず頻繁なアップデートが行われてきました。本章まで紹介してきたコンポーネントの仕様についても、細かな改善が度々行われています。つまり、Bootstrapを正しく活用していくためには、最新情報をこまめにチェックする必要があると言えるでしょう。では、Bootstrapのバージョンなど最新情報を確認する方法を見ていきましょう。

### 公式サイトでバージョンを確認する

Bootstrap公式サイト（<https://getbootstrap.com/>）のナビゲーションバー上で、現在の最新バージョンを確認することができます。また、過去バージョンのドキュメントと表示を切り替えることも可能です。

▼図 6-81 現在の最新バージョンを確認する



なお、メジャーアップデートされた内容については、公式サイトの「Documentation > Migration (<https://getbootstrap.com/docs/4.1/migration/>)」で確認することができます。

▼図 6-82 現在の最新バージョンを確認する

