

Bootstrap4.xの使い方をBootstrap3.xからの変更箇所を交えて解説しています。

モーダル（Modal）

v4.0.0設定変更

v4.2.1追加

BootstrapのJavaScriptモーダルプラグインを使用して、ライトボックス、ユーザー通知、または完全カスタムコンテンツ用のダイアログをサイトに追加。

使い方（How it works）

Bootstrapモーダルコンポーネントを使い始める前に、メニューオプションが最近変更されたので、以下を読むこと。

- モーダルは、HTML、CSS、およびJavaScriptで構築。ドキュメント内の他の部分に配置され、`<body>` からスクロールを削除して、代わりにモーダルコンテンツをスクロールする。
- モーダルの"backdrop"（背景部分）をクリックすると、自動的にモーダルが閉じる。
- Bootstrapは一度に1つのモーダルウィンドウしかサポートしない。入れ子になったモーダルは、ユーザー経験が乏しいと思われるためサポートされていない。
- モーダルは、`position:fixed` を使用。これは、そのレンダリングに関してちょっと変わったことがある。可能であれば、他の要素からの干渉を避けるために、モーダルHTMLを最上位に配置すること。他の固定要素内で `.modal` を入れ子になったときに問題に遭遇する可能性がある。
- 再度、`position:fixed` のために、モバイルデバイス上のモーダル使用に関するいくつかの注意点がある。詳細については、[ブラウザ・サポートの解説](#)を参照すること。
- HTML5がそのセマンティクスをどのように定義するかによって、[autofocus HTML属性](#)は、Bootstrapのモーダルには何の影響も与えない。同じ効果を得るためにカスタムJavaScriptを使用：

```
$( '#myInput' ).trigger( 'focus' )
})
```

このコンポーネントのアニメーション効果は、`prefers-reduced-motion` メディアクエリに依存。詳細はアクセシビリティの[モーションを小さくする](#)を参照。

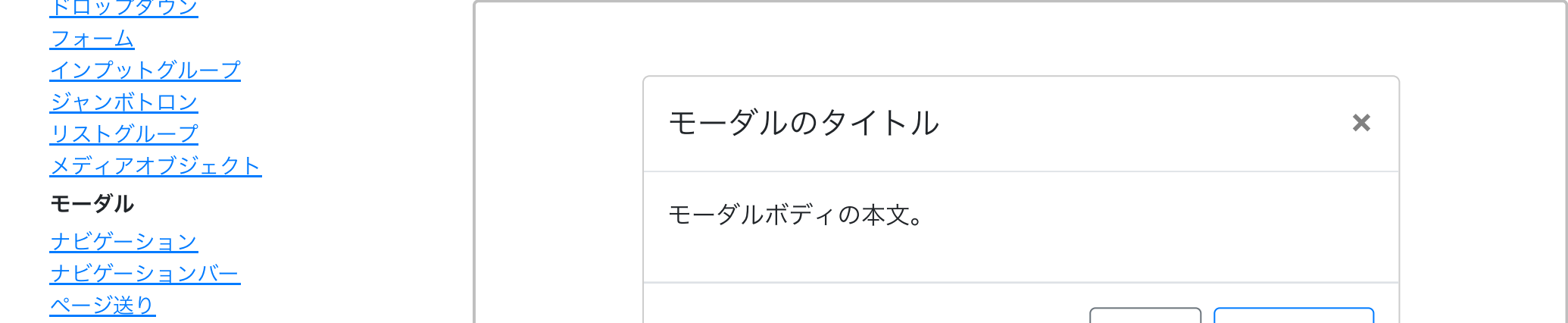
実例の見本と使用方法のガイドラインを参照。

実例（Examples）

モーダルコンテンツ（Modal components） v4.0.0設定変更

以下は静的モーダルの例（`position` と `display` が再定義されたことを意味する）。モーダルヘッダ、モーダルボディ（`padding` に必要）、モーダルフッタ（オプション）が含まれます。可能な限り、アクションを終了するモーダルヘッダを含めるか、別の明示的な却下アクションを提供することを推奨。

見本 モーダルを開いた状態で表示



コンポーネント

[アラート](#)

[バッジ](#)

[パンくずリスト](#)

[ボタン](#)

[ボタングループ](#)

[カード](#)

[カルーセル](#)

[折り畳み](#)

[ドロップダウン](#)

[フォーム](#)

[インプットグループ](#)

[ジャンボトロン](#)

[リストグループ](#)

[メディアオブジェクト](#)

モーダル

[ナビゲーション](#)

[ナビゲーションバー](#)

[ページ送り](#)

- [ポップオーバー](#)
- [プログレスバー](#)
- [スクロールスパイ](#)
- [スピナー](#)
- [トースト](#)
- [ツールチップ](#)

Bootstrap3.xの設定例

赤背景が変更箇所

```
<div class="modal" tabindex="-1"
role="dialog">
  <div class="modal-dialog"
role="document">
    <div class="modal-
content">
      <div
class="modal-header">
        <button
type="button" class="close" data-
dismiss="modal" aria-label="閉じる">
          <span aria-hidden="true">&times;</span>
        </button>
        <h4
class="modal-title">モーダルのタイトル</h4>
      </div><!--
/.modal-header -->
      <div
class="modal-body">
        <p>モーダ
ルボディの本文。</p>
      </div><!--
/.modal-body -->
      <div
class="modal-footer">
        <button
type="button" class="btn btn-default"
data-dismiss="modal">閉じる</button>
        <button
type="button" class="btn btn-primary">変
更を保存</button>
      </div><!--
/.modal-footer -->
    </div><!-- /.modal-
content -->
  </div><!-- /.modal-dialog -->
</div><!-- /.modal -->
```

Bootstrap4.xの設定例

緑背景が変更箇所

```
<div class="modal" tabindex="-1"
role="dialog">
  <div class="modal-dialog"
role="document">
    <div class="modal-
content">
      <div
class="modal-header">
        <h5
class="modal-title">モーダルのタイトル</h5>
        <button
type="button" class="close" data-
dismiss="modal" aria-label="閉じる">
          <span aria-hidden="true">&times;</span>
        </button>
      </div><!--
/.modal-header -->
      <div
class="modal-body">
        <p>モーダ
ルボディの本文。</p>
      </div><!--
/.modal-body -->
      <div
class="modal-footer">
        <button
type="button" class="btn btn-secondary"
data-dismiss="modal">閉じる</button>
        <button
type="button" class="btn btn-primary">変
更を保存</button>
      </div><!--
/.modal-footer -->
    </div><!-- /.modal-
content -->
  </div><!-- /.modal-dialog -->
</div><!-- /.modal -->
```

【設定】

- `div.modal > div.modal-dialog > div.modal-content > {div.modal-header > h5.modal-title 《タイトル》 + [button.close > <span> > &times;] 《閉じるアイコンボタン》}` 《ヘッダ》 + `div.modal-body` 《コンテンツ》 + `div.modal-footer` 《フッタ》
- ヘッダ (`.modal-header`) は、《タイトル》 (`.modal-title`) ⇒ 《閉じるアイコンボタン》 (`button.close`) の順に記載
- アイコンとなる `×` は、`&times;` と記述してエスケープ処理をする
- プラグインを使ってモーダルを閉じるため、閉じるアイコンボタンや閉じるボタン部分には `data-dismiss="modal"` を入れる

【アクセシビリティの設定】

- `div.modal` に、`role="dialog"` 属性（モーダルダイアログの役割を伝える）を入れる
- `div.modal-dialog` に、`role="document"` 属性（ドキュメントの役割を伝える）を入れる
- モーダルを閉じるボタンとなる `<button>` に、`aria-label` 属性（アイコンボタンのラベル）を入れる
- 閉じるボタンアイコンの `&times;` を `span[aria-hidden="true"]`（スクリーンリーダー等での読み上げをスキップさせる）で囲む

【Bootstrap3.xとの変更箇所】

- モーダルのヘッダ：《閉じるアイコンボタン》 ⇒ 《タイトル》の順から《タイトル》 ⇒ 《閉じるアイコンボタン》の順に
- モーダルのタイトルの大きさ：`<h4>` ⇒ `<h5>`

## モーダルの設定（Live demo）

下のボタンをクリックして、動作中のモーダルのデモを切り替える。ページの上からスライドしてフェードイン。

見本

ココを押すと表示

設定例

```
<!-- 切り替えボタンの設定 -->
<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#myModal">
  ココを押すと表示
</button>

<!-- モーダルの設定 -->
<div class="modal fade" id="myModal" tabindex="-1" role="dialog" aria-labelledby="myModalLabel">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalLabel">モーダルのタイトル
        <button type="button" class="close" data-dismiss="modal">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">
        <p>モーダルのコンテンツ文。</p>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-dismiss="modal">キャンセル
        <button type="button" class="btn btn-primary">変更を保存
      </div><!-- /.modal-footer -->
    </div><!-- /.modal-content -->
  </div><!-- /.modal-dialog -->
</div><!-- /.modal -->
```

コピー

### 【設定】

- `button.btn` 《切り替えボタン》 + `{div.modal.fade[role="dialog"] > div.modal-dialog[role="document"] > div.modal-content > {div.modal-header > h5.modal-title 《タイトル》 + [button.close[data-dismiss="modal"] [aria-label="閉じる"] > span[aria-hidden="true"] > &times;}` 《閉じるアイコンボタン》} 《ヘッダ》 + `div.modal-body` 《コンテンツ》 + `div.modal-footer` 《フッタ》} 《モーダル部分》
  - モーダルのプラグインを使用するため、
    - 切り替えボタンには、`data-toggle="modal"`（プラグインを有効にする）と、`data-target="#モーダルのID"`（または`href="#モーダルのID"`）
    - `div.modal`（ボタンのリンク先）には、`id="ID名"`と`tabindex="-1"`（escキーを押すとモーダルが閉じるようにする）
- をそれぞれ入れること

### 【アクセシビリティの設定】

- `h5.modal-title` に `id="ID名"` を入れ、`div.modal` に `aria-labelledby="タイトルのID"` 属性（タイトルをモーダルのラベルとして関連付ける）を入れること
- さらに `div.modal` に `aria-describedby` 属性を入れてモーダルダイアログの説明をすることも可能

## 長いコンテンツのスクロール（Scrolling long content）

モーダルがユーザーのビューポートやデバイスにとって長すぎると、モーダルはページ自体とは独立してスクロール。以下の見本で試して、どういう意味か確認すること。

見本

ココを押すと表示

### 設定例

<!-- 切り替えボタンの設定 -->

<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#exampleModal">ココを押すと表示</button>

</div>

<!-- モーダルの設定 -->

<div class="modal fade" id="exampleModalLong" tabindex="-1" role="dialog" aria-labelledby="exampleModalLongTitle"><div class="modal-dialog"><div class="modal-content"><div class="modal-header"><h5 class="modal-title" id="exampleModalLongTitle">モーダル</h5><button type="button" class="close" data-dismiss="modal"><span aria-hidden="true">&times;</span></button></div><div class="modal-body">...</div><div class="modal-footer"><button type="button" class="btn btn-secondary" data-dismiss="modal">キャンセル</button><button type="button" class="btn btn-primary">変更を保存</button></div></div></div></div>

`.modal-dialog-scrollable` を `.modal-dialog` に追加することでモーダル本体をスクロール可能にする。v4.3.0追加

### 見本

ココを押すと表示

### 設定例

<!-- 切り替えボタンの設定 -->

<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#exampleModal">ココを押すと表示</button>

</div>

<!-- モーダルの設定 -->

<div class="modal fade" id="exampleModalScrollable" tabindex="-1" role="dialog" aria-labelledby="exampleModalScrollableTitle"><div class="modal-dialog modal-dialog-scrollable"><div class="modal-content"><div class="modal-header"><h5 class="modal-title" id="exampleModalScrollableTitle">モーダル</h5><button type="button" class="close" data-dismiss="modal"><span aria-hidden="true">&times;</span></button></div><div class="modal-body">...</div><div class="modal-footer"><button type="button" class="btn btn-secondary" data-dismiss="modal">キャンセル</button><button type="button" class="btn btn-primary">変更を保存</button></div></div></div></div>

## 垂直方向に中央に配置（Vertically centered） v4.0.0-beta.3新設

file:///Users/user/Downloads/\_md5/モーダル〜Bootstrap4移行ガイド.html

4/11

`.modal-dialog-centered` を `.modal-dialog` に追加して、モーダルを垂直方向に中央に配置。

### 見本

垂直方向に中央揃えのモーダル

垂直方向にスクロール可能なモーダル

### 設定例

垂直方向に中央揃えのモーダル

```
<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#exampleModal">
  垂直方向に中央揃えのモーダル
</button>

<!-- モーダルの設定 -->
<div class="modal fade" id="exampleModalCenter" tabindex="-1" role="dialog" aria-label="Modal">
  <div class="modal-dialog modal-dialog-centered" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalCenterTitle">モーダルの設定
        <button type="button" class="close" data-dismiss="modal"><span aria-hidden="true">&times;</span>
      </div>
      <div class="modal-body">
        <p>モーダルのコンテンツ文。</p>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-dismiss="modal">キャンセル
        <button type="button" class="btn btn-primary">変更を保存
      </div>
    </div>
  </div>
</div>
```

【設定】

- `.modal-dialog` に `.modal-dialog-centered` を追加

## ツールチップとポップオーバー（Tooltips and popovers）

[ツールチップ](#)と[ポップオーバー](#)を必要に応じてモーダル内に配置。モーダルが閉じられると、内のツールヒントやポップオーバーも自動的に閉じられる。

### 見本

ココを押すと表示

### 設定例

```
HTML
class="modal-body">
  <h5>モーダル内のポップオーバー</h5>
  <p>この<a href="#" role="button" class="btn btn-secondary popover-test" title="モーダル内のポップオーバー">リンク</a>をクリックするとモーダル内のポップオーバーが表示されます。</p>
  <h5>モーダル内のツールチップ</h5>
  <p><a href="#" class="tooltip-test" title="ツールチップ1の表示">このリンク</a>と<a href="#" class="tooltip-test" title="ツールチップ2の表示">このリンク</a>をクリックするとモーダル内のツールチップが表示されます。</p>
</div>
```

### 実行コード

```
JSファイルの場合
$(document).ready(function() {
  $('[role="button"].popover-test').popover()
  $('[class="tooltip-test"]').tooltip()
})
```

【設定】

- モーダルの切替ボタンの部分にポップオーバーまたはツールチップ用のクラスを追加し、別途Javascriptで実行コードを作成する



## グリッドを使用（Using the grid）

`.modal-body` 内に `.container-fluid` を入れ子にして、モーダル内のBootstrapグリッドシステムを利用。次に、通常どおりのグリッドシステムクラスを使用。

見本

ココを押すと表示

設定例

```
<div class="modal-body">
  <div class="container-fluid">
    <p>モーダルのコンテンツ</p>
    <div class="row">
      <div class="col-md-4">.col-md-4</div>
      <div class="col-md-4 ml-auto">.col-md-4 .ml-auto</div>
    </div>
    <div class="row">
      <div class="col-md-3 ml-auto">.col-md-3 .ml-auto</div>
      <div class="col-md-2 ml-auto">.col-md-2 .ml-auto</div>
    </div>
    <div class="row">
      <div class="col-md-6 mx-auto">.col-md-6 .mx-auto</div>
    </div>
    <div class="row">
      <div class="col-sm-9">
        レベル 1: .col-sm-9
        <div class="row">
          <div class="col-8 col-sm-6">
            レベル 2: .col-8 .col-sm-6
          </div>
          <div class="col-4 col-sm-6">
            レベル 2: .col-4 .col-sm-6
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

コピー

【設定】

- `div.modal-body > div.container-fluid > div.row > div.col(-{breakpoint})-*` (`.m{l,r,x}(-{breakpoint})-auto`)

## 様々なモーダルコンテンツ（Varying modal contents）

若干異なる内容で同じモーダルを起動する複数のボタンが必要な場合、クリックしたボタンに応じてモーダルの内容を変更するには、`event.relatedTarget` 属性と[HTML data-\\*属性](#)（おそらく[jQuery.経由](#)）を使用。

以下に見本とHTMLとJavaScriptの設定例を示す。`relatedTarget` の詳細は、[モーダルイベントの解説](#)に記載。

見本

社長向けモーダルを開く

副社長向けモーダルを開く

重役向けモーダルを開く

設定例

```
HTMLton type="button" class="btn btn-primary" data-toggle="modal" data-target="#exampleModal">送信</button>
<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#exampleModal">送信</button>
<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#exampleModal">送信</button>
<div class="modal fade" id="exampleModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalLabel">新メッセージ</h5>
        <button type="button" class="close" data-dismiss="modal"><span aria-hidden="true">&times;</span></button>
      </div>
      <div class="modal-body">
        <form>
          <div class="form-group">
            <label for="recipient-name">受信者:</label>
            <input type="text" class="form-control" id="recipient-name">
          </div>
          <div class="form-group">
            <label for="message-text">メッセージ:</label>
            <textarea class="form-control" id="message-text">
          </div>
        </form>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-dismiss="modal">キャンセル</button>
        <button type="button" class="btn btn-primary">送信</button>
      </div>
    </div>
  </div>
</div>
</div>
```

```
JavaScriptleModal').on('show.bs.modal', function (event) {
  var button = $(event.relatedTarget) // モーダル切替えボタン
  var recipient = button.data('whatever') // data-* 属性から情報を抽出
  // 必要に応じて、ここでAJAXリクエストを開始可能(コールバックで更新することも可能)
  // モーダルの内容を更新。ここではjQueryを使用するが、代わりにデータ・バインディング・ライブラリまたは他
  var modal = $(this)
  modal.find('.modal-title').text(recipient + 'にメッセージを送る')
  modal.find('.modal-body input').val(recipient)
})
```

## アニメーションの削除（Remove animation）

フェードインではなく、いきなり表示されるモーダルにする場合は、モーダルのマークアップから `.fade` クラスを削除。

見本



設定例

```
<div class="modal" id="myModal" tabindex="-1" role="dialog" aria-labelledby="myModalLabel">
  ...
</div>
```

### 【設定】

- `.modal` で `.fade` をはずす

## 動的な高さのモーダル（Modals with dynamic heights）

モーダルの高さが開いている間に変更され、スクロールバーが表示された場合には、`$('#myModal').modal('handleUpdate')` を呼び出してモーダルの位置を再調整する必要がある。

## アクセシビリティ（Accessibility）

モーダルタイトルを参照する `role="dialog"` と `aria-labelledby="..."` を `.modal` に、そして `role="document"` を `.modal-dialog` 自体に必ず追加すること。さらに、モーダルダイアログの説明を `.modal` の `aria-describedby` で指定することも可能。

## YouTube動画の埋め込み（Embedding YouTube videos）

サンプル

YouTubeの動画をモーダルに埋め込むには、再生を自動的に停止するなどの追加のJavaScriptが必要。詳細は、この[Stack Overflowの便利な記事](#)に記載。

## モーダルの大きさ（Optional sizes） v4.2.1一部追加

モーダルには3つのオプションのサイズがあり、修飾クラスを使用して `.modal-dialog` に配置することが可能。これらのサイズは、狭いビューポートでの水平スクロールバーの出現を避けるために、特定のブレイクポイントで実行される。

| サイズ   | クラス                    | モーダルの最大幅 |
|-------|------------------------|----------|
| 小     | <code>.modal-sm</code> | 300px    |
| 標準（中） | なし                     | 500px    |
| 大     | <code>.modal-lg</code> | 800px    |
| 特大    | <code>.modal-xl</code> | 1140px   |

修飾クラスを持たないデフォルトモーダルは、"中"サイズのモーダルを構成。

### 大きさの種類

●特大：`.modal-xl` v4.2.1追加

より大きめのモーダル

●大きめ：`.modal-lg`

大きめのモーダル

●小さめ：`.modal-sm`

小さめのモーダル

※参考：標準

標準のモーダル

### 設定例

特大

```
button type="button" class="btn btn-primary" data-toggle="modal" data-target="#myModal"
<div class="modal fade" id="myModal" tabindex="-1" role="dialog" aria-labelledby="myModalLabel"
  <div class="modal-dialog modal-xl" role="document">
    <div class="modal-content">
      ...
    </div>
  </div>
</div>
```

大きめ

```
button type="button" class="btn btn-primary" data-toggle="modal" data-target="#myModal"
<div class="modal fade" id="myModal" tabindex="-1" role="dialog" aria-labelledby="myModalLabel"
  <div class="modal-dialog modal-lg" role="document">
    <div class="modal-content">
      ...
    </div>
  </div>
</div>
```



|   |   |
|---|---|
| 小さめ   | <code>on type="button" class="btn btn-primary" data-toggle="modal" data-target="#myModal" data-size="sm"&gt;コピー&lt;/button&gt;</code> |
| <code>&lt;div class="modal fade" id="myModal" tabindex="-1" role="dialog" aria-labelledby="myModalLabel"&gt;</code> |   |
| <code>  &lt;div class="modal-dialog modal-sm" role="document"&gt;</code>  |   |
| <code>    &lt;div class="modal-content"&gt;</code>  |   |
| <code>      ...</code>  |   |
| <code>    &lt;/div&gt;</code>   |   |
| <code>&lt;/div&gt;</code>   |   |
| <code>&lt;/div&gt;</code>   |   |

【設定】

- `.modal-dialog` に `.modal-{size}`（上記の「大きさの種類」から選択）を追加

【変更履歴】

- 【v4.2.1】 大きさの種類に特大（`.modal-xl`）が追加

## 使用方法（Usage）

モーダルプラグインは、必要に応じて、データ属性やJavaScriptを使用して、非表示のコンテンツを切り替える。また、`<body>` に `.modal-open` を追加してデフォルトのスクロール動作を再定義し、モーダルの外側をクリックしたときに表示されたモーダルを閉じるためのクリック領域を提供する `.modal-backdrop` を生成する。

## データ属性経由で（Via data attributes）

JavaScriptを書かずにモーダルを有効にする。特定のモーダルを起動させるためには、`data-target="#foo"` または `href="#foo"` とともに、ボタンのようなコントローラー要素に `data-toggle = "modal"` を設定すること。

|        |   |  |
|--------|---|--|
| ボタンで設定 | <code>type="button" data-toggle="modal" data-target="#myModal"&gt;モーダルを起動&lt;/button&gt;</code> |  |
| リンクで設定 | <code>toggle="modal" href="#myModal"&gt;モーダルを起動&lt;/a&gt;</code>                                |  |

## JavaScript経由で（Via JavaScript）

JavaScript1行でID `myModal` のモーダルを呼び出す：

|            |  |  |
|------------|--|--|
| JavaScript | <code>\$('#myModal').modal(options)</code> |  |
|------------|--|--|

## オプション（Options）

サンプル

v4.0.0一部変更

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可能。データ属性の場合、`data-backdrop=""` のように `data-` にオプション名を追加すること。

| 名前                | タイプ                                      | デフォルト | 説明   |
|-------------------|--|-------|--|
| backdrop          | boolean または string <code>'static'</code> | true  | モーダルの背景をオーバーレイ表示にする<br><code>true</code> ：有効／ <code>false</code> ：無効<br>または、クリック時にモーダルを閉じないようにするには <code>static</code> を指定 |
| keyboard          | boolean                                  | true  | escキーが押されたときにモーダルを閉じる<br><code>true</code> ：有効／ <code>false</code> ：無効   |
| focus<br>v4.0.0追加 | boolean                                  | true  | 初期化時にモーダルにフォーカスを移動<br><code>true</code> ：有効／ <code>false</code> ：無効  |
| show              | boolean                                  | true  | 初期化時にモーダルを表示<br><code>true</code> ：有効／ <code>false</code> ：無効  |

【Bootstrap3.xとの変更箇所】

- `remote` は、既にv3.3.0で廃止され、v4では削除

## メソッド（Methods）

### 非同期メソッドと遷移

すべてのAPIメソッドは、**非同期**で**遷移**を開始する。それらは移行が始まるとすぐに呼び出し元に戻るが、**終了する前に**呼び出し元に戻る。さらに、**遷移コンポーネントのメソッド呼び出しは無視される**。

詳細については、[Javascriptのドキュメント](#)に記載。

### `.modal(options)`

コンテンツをモーダルとしてアクティブ化する。オプションのオブション **object** を受け入れる。

```
JavaScript | al').modal({
  keyboard: false
})
```

### `.modal('toggle')`

モーダルを手動で切り替える。**モーダルが実際に表示または非表示になる前**（つまり、**shown.bs.modal** または **hidden.bs.modal** イベント発動前）に**呼び出し元に戻る**。

```
JavaScript | al').modal('toggle')
```

### `.modal('show')`

モーダルを手動で開く。**モーダルが実際に表示される前**（つまり、**shown.bs.modal** イベント発動前）に**呼び出し元に戻る**。

```
JavaScript | al').modal('show')
```

### `.modal('hide')`

モーダルを手動で非表示にする。**モーダルが実際に非表示になる前**（つまり、**hidden.bs.modal** イベントが発動する前）に**呼び出し元に戻る**。

```
JavaScript | al').modal('hide')
```

### `.modal('handleUpdate')`

モーダルの高さが開いているとき（つまり、スクロールバーが表示されているとき）に変更された場合は、モーダルの位置を手動で再調整する。

```
JavaScript | al').modal('handleUpdate')
```

### `.modal('dispose')` v4.0.0-beta.2追加

要素のモーダルを破棄。

## イベント（Events） サンプル v4.0.0一部廃止

Bootstrapのモーダルクラスは、モーダル機能に接続するためのいくつかのイベントを公開している。すべてのモーダルイベントはモーダル自体（つまり、**<div class="modal">**）で発動。

| イベントタイプ        | 説明   |
|----------------|--|
| show.bs.modal  | このイベントは、 <b>show</b> のインスタンス・メソッドが呼び出されると直ちに発動。クリックによって発動した場合、クリックされた要素はイベントの <b>relatedTarget</b> プロパティとして使用可能。   |
| shown.bs.modal | このイベントは、モーダルがユーザーに表示されたときに発動（完全にCSS 遷移が完了するまで待機）。クリックによって発動した場合、クリックされた要素はイベントの <b>relatedTarget</b> プロパティとして使用可能。 |

| イベントタイプ         | 説明  |
|-----------------|---|
| hide.bs.modal   | このイベントは、 <b>hide</b> のインスタンス・メソッドが呼び出されると直ちに発動。     |
| hidden.bs.modal | このイベントは、モーダルがユーザーから非表示になったときに発動（完全にCSS遷移が完了するまで待機）。 |

使用例

```
JavaScript modal').on('hidden.bs.modal', function (e) {  
    // 何かをする...  
})
```

- 【Bootstrap3.xとの変更箇所】
- loaded.bs.modal** は **remote** の廃止（v3.3.0）に伴い廃止