

第 8 章

ユーティリティ

Bootstrap には、何度も同じスタイルを指定しなくても良いように、繰り返し使われる利用頻度の高いスタイルがユーティリティクラスとして用意されています。これらユーティリティクラスは、コンポーネントを配置した後の微調整に大変役立ちます。定義されている数が多いため、最初は覚えるのが大変ですが、CSS プロパティがわかる人なら概ね予想が付くクラス名になっています。そうではない人も、ある程度使用していくうちに覚えることができるでしょう。

Color ユーティリティ

Color ユーティリティを使用して、要素に文字色や背景色を設定することができます。Bootstrapでは、primary（重要）、success（成功）、danger（危険）などコンテクスト（文脈や意味）と対応したテーマカラーが用意されており、色で情報を伝えたいときなどにも便利です。これらのクラスを使って文字色や背景色を指定すると、WebサイトやWebアプリケーションで一貫性を保った配色することができます。

8.1.1 文字色を設定するクラス

文字色を指定するには、**text-{ 色の種類 } クラス**を使用します。Bootstrapで定義済みの文字色クラスは表8-1のとおりです。

▼表8-1 文字色を設定するクラス一覧

| クラス | 文字色の種類 |
|----------------|-----------------------------|
| text-primary | 青系 |
| text-secondary | グレー |
| text-success | 緑系 |
| text-danger | 赤系 |
| text-warning | 黄系 |
| text-info | 水色系 |
| text-light | 明るいグレー |
| text-dark | 暗いグレー |
| text-body | Bootstrapでbody要素に設定されている文字色 |
| text-muted | グレー |
| text-white | 白 |
| text-black-50 | 50%透過の黒 |
| text-white-50 | 50%透過の白 |

これらのクラスは、リンクのホバー時やフォーカス時でもうまく機能するようにスタイルが設定されていますが、text-bodyクラス、text-mutedクラス、text-whiteクラス、text-black-50クラス、text-white-50クラスについては、リンク用のスタイルは設定されていません（そもそもa要素の設定でホバー時に下線が付くのみとなっています）。デフォルトでは、text-secondaryクラスとtext-mutedクラスは同じ色のグレーが定義されていますが、text-secondaryは、ホバー時に若干文字色が濃くなるのに対し、text-muteは変化しません。紙面では確認ができないほど微妙な変化なので、サンプルで確認してください（リスト8-1、図8-1）。

▼リスト 8-1 文字色を設定するクラスの例 (color.html)

```
<p><a href="#" class="text-primary">text-primary</a></p>
<p><a href="#" class="text-secondary">text-secondary</a></p>
<p><a href="#" class="text-success">text-success</a></p>
<p><a href="#" class="text-danger">text-danger</a></p>
<p><a href="#" class="text-warning">text-warning</a></p>
<p><a href="#" class="text-info">text-info</a></p>
<p><a href="#" class="text-light bg-dark">text-light</a></p>
<p><a href="#" class="text-dark" text-dark</a></p>
<p><a href="#" class="text-muted">text-muted</a></p>
<p><a href="#" class="text-white bg-dark">White link</a></p>
```

▼図 8-1 文字色を設定するクラスの例



NOTE

色のみで情報を伝えない

色を使用すると重要度や緊急性など、情報を視覚的に伝えることができますが、スクリーンリーダーなどの支援技術には、こうした情報は伝わりません。色で示される情報をコンテンツ自体で明示にするか、またはスクリーンリーダー用の sr-only クラスを使用して代替手段を入れるようにしましょう (P.354 参照)。

8.1.2 背景色を設定するクラス

文字色同様、**bg-{ 色の種類 }** クラスを利用して背景色を指定できます。Bootstrap で定義済みの背景色クラスは表 8-2 のとおりです。

▼表 8-2 背景色を設定するクラス一覧

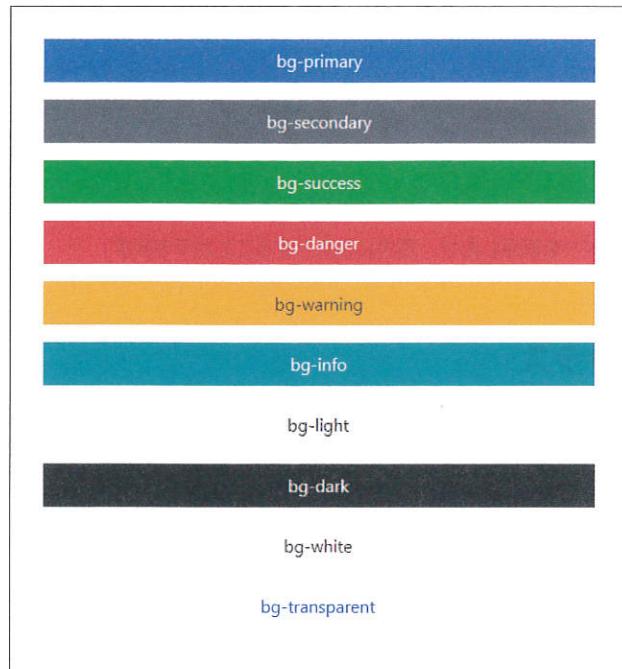
| クラス | 背景色の種類 |
|----------------|---------|
| bg-primary | 青系 |
| bg-secondary | グレー系 |
| bg-success | 緑系 |
| bg-danger | 赤系 |
| bg-warning | 黄系 |
| bg-info | 水色系 |
| bg-light | 明るいグレー系 |
| bg-dark | 暗いグレー系 |
| bg-white | 白 |
| bg-transparent | 透過 |

アンカー要素に指定した場合は、文字色クラスと同様にホバー時に色が濃くなります。背景色ユーティリティでは、文字色は定義されていないため、場合によっては文字色クラスと併用する必要があります。サンプルでは、背景色の明度に合わせて文字色クラスも指定しています。また、アンカー要素の背景色をわかりやすくするため、Display ユーティリティ (P.310 参照) の d-block クラスでブロックレベルの表示にし、Spacing ユーティリティ (P.318 参照) の py-2 クラスでパディングを付けて間隔を広げています (リスト 8-2、図 8-2)。

▼リスト 8-2 背景色を設定するクラスの例 (bg-color.html)

```
<p><a href="#" class="bg-primary text-white d-block py-2">bg-primary</a></p>
<p><a href="#" class="bg-secondary text-white d-block py-2">bg-secondary</a></p>
<p><a href="#" class="bg-success text-white d-block py-2">bg-success</a></p>
<p><a href="#" class="bg-danger text-white d-block py-2">bg-danger</a></p>
<p><a href="#" class="bg-warning text-dark d-block py-2">bg-warning</a></p>
<p><a href="#" class="bg-info text-white d-block py-2">bg-info</a></p>
<p><a href="#" class="bg-light text-dark d-block py-2">bg-light</a></p>
<p><a href="#" class="bg-dark text-white d-block py-2">bg-dark</a></p>
<p><a href="#" class="bg-white text-dark d-block py-2">bg-white</a></p>
<p><a href="#" class="bg-transparent d-block py-2">bg-transparent</a></p>
```

▼図 8-2 背景色を設定するクラスの例

**NOTE****グラデーション背景の設定**

Bootstrap では、グラデーション背景のクラスも用意されていますが、デフォルトではこの機能が無効になっています。有効にするには Sass オプションでカスタマイズする必要があります (P.446 参照)。

1
2
3
4
5
6
7
8
9
10

Border ユーティリティ

Border ユーティリティを使用して、要素にボーダーを表示したり、角丸にしたりすることができます。

8.2.1 ボーダーを追加するクラス

要素の上下左右のすべての辺にボーダーを表示するには **border クラス**を追加します。辺を選んで指定する場合は、**border-top**のような形式の **border-{ 辺 }** クラスを追加します。Bootstrap で定義済みのボーダー追加クラスは表 8-3 のとおりです。

▼表 8-3 ボーダーを表示するクラス一覧

| クラス | ボーダーを表示する辺 |
|---------------|------------|
| border | 全辺 |
| border-top | 上ボーダー |
| border-right | 右ボーダー |
| border-bottom | 下ボーダー |
| border-left | 左ボーダー |

デフォルトでは、ボーダー色は薄いグレー (#dee2e6)、1px の太さの実線で定義されています。サンプルではわかりやすいように背景色を付けています（リスト 8-3、図 8-3）。

▼リスト 8-3 ボーダーを表示するクラスの例（border.html）

```
<span class="border">border</span><!-- 全辺ボーダー -->
<span class="border-top">border-top</span><!-- 上ボーダー -->
<span class="border-right">border-right</span><!-- 右ボーダー -->
<span class="border-bottom">border-bottom</span><!-- 下ボーダー -->
<span class="border-left">border-left</span><!-- 左ボーダー -->
```

▼図 8-3 ボーダーを追加するクラスの例

8.2.2 ボーダーを削除するクラス

既にボーダーの表示が設定されている要素から、すべてのボーダーを削除するには **border-0**、辺を選んで削除するには **border-{ 辺 }-0** クラスを使用します。Bootstrap で定義済みのボーダー削除クラスは表 8-4 のとおりです。

▼表 8-4 ボーダーを削除するクラス一覧

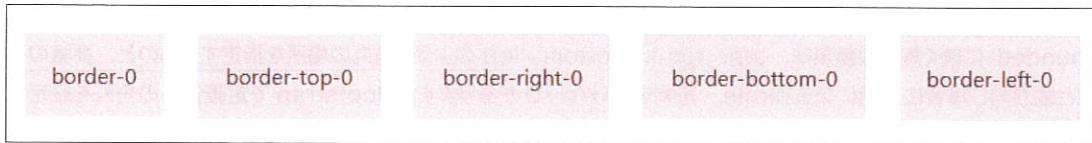
| クラス | ボーダーを削除する辺 |
|-----------------|------------|
| border-0 | 全辺のボーダー |
| border-top-0 | 上ボーダー |
| border-right-0 | 右ボーダー |
| border-bottom-0 | 下ボーダー |
| border-left-0 | 左ボーダー |

サンプルでは、border クラスで要素の全辺にボーダーを指定した上で、指定のボーダーを削除しています（リスト 8-4、図 8-4）。

▼リスト 8-4 ボーダーを削除するクラスの例（border-0.html）

```
<span class="border border-0">border-0</span><!-- 全ボーダー削除 -->
<span class="border border-top-0">border-top-0</span><!-- 上ボーダー削除 -->
<span class="border border-right-0">border-right-0</span><!-- 右ボーダー削除 -->
<span class="border border-bottom-0">border-bottom-0</span><!-- 下ボーダー削除 -->
<span class="border border-left-0">border-left-0</span><!-- 左ボーダー削除 -->
```

▼図 8-4 ボーダーを削除するクラスの例



8.2.3 ボーダー色を設定するクラス

ボーダー色の指定には、border クラスが設定された要素に **border-{ 色の種類 } クラス** を追加します。Bootstrap で定義済みのボーダー色クラスは表 8-5 のとおりです。

▼表 8-5 ボーダー色を設定するクラス一覧

| クラス | ボーダー色の種類 |
|------------------|----------|
| border-primary | 青系 |
| border-secondary | グレー系 |
| border-success | 緑系 |
| border-danger | 赤系 |
| border-warning | 黄系 |

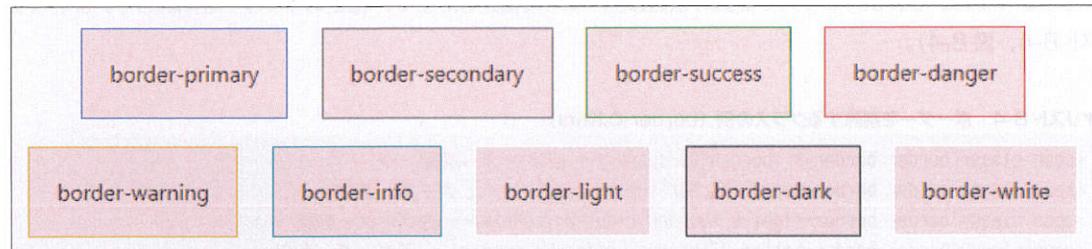
| クラス | ボーダー色の種類 |
|--------------|----------|
| border-info | 水色系色 |
| border-light | 明るいグレー系 |
| border-dark | 暗いグレー系 |
| border-white | 白 |

サンプルでは、白ボーダー（border-white クラス）がわかるように body 要素にグレーの背景色を付けています（リスト 8-5、図 8-5）。

▼リスト 8-5 ボーダー色を設定するクラスの例（border-color.html）

```
<span class="border border-primary">border-primary</span>
<span class="border border-secondary">border-secondary</span>
<span class="border border-success">border-success</span>
<span class="border border-danger">border-danger</span>
<span class="border border-warning">border-warning</span>
<span class="border border-info">border-info</span>
<span class="border border-light">border-light</span>
<span class="border border-dark">border-dark</span>
<span class="border border-white">border-white</span>
```

▼図 8-5 ボーダー色クラスの例



8.2.4 角丸を設定するクラス

要素に角丸を設定するには、**rounded** クラスまたは **rounded-{ 角丸の種類 }** クラスを追加します。rounded- に続く角丸の種類は、top、right、bottom、left のように角丸の場所を指定するものと、要素の高さと幅が同じ場合に正円になる circle、角丸を付けない 0 があります。Bootstrap で定義済みの角丸を設定するクラスは表 8-6 のとおりです。

▼表 8-6 角丸を設定するクラス一覧

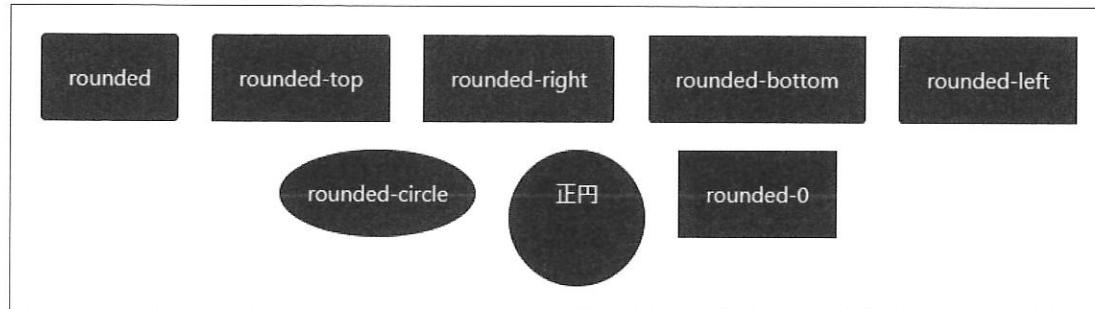
| クラス | 角丸の種類 |
|----------------|-------|
| rounded | 全辺を角丸 |
| rounded-top | 上が角丸 |
| rounded-right | 右が角丸 |
| rounded-bottom | 下が角丸 |
| rounded-left | 左が角丸 |
| rounded-circle | 円 |
| rounded-0 | 角丸なし |

rounded クラスは、CSS の border-radius プロパティを使っています。border- と名前がありますが、ボーダーを設定していなくても、背景色の設定されている要素や img 要素の四隅も丸くなります（リスト 8-6、図 8-6）。

▼リスト 8-6 角丸を設定するクラスの例（rounded.html）

```
<span class="rounded">rounded</span>
<span class="rounded-top">rounded-top</span>
<span class="rounded-right">rounded-right</span>
<span class="rounded-bottom">rounded-bottom</span>
<span class="rounded-left">rounded-left</span>
<span class="rounded-circle" id="circle">rounded-circle</span>
<span class="rounded-circle" id="circle" style="border-radius: 50%; width: 100px; height: 100px; background-color: black; color: white; text-align: center; line-height: 100px; margin: 10px;">正円</span><!-- 幅と高さを同じ（100px）に設定 --&gt;
&lt;span class="rounded-0"&gt;rounded-0&lt;/span&gt;</pre>
```

▼図 8-6 角丸を設定するクラスの例



1
2
3
4
5
6
7
8
9
10

8 SECTION 3 Display ユーティリティ

Display ユーティリティは、要素の表示形式をレスポンシブに切り替えるために使います。

8.3.1 表示形式を設定するクラス

このユーティリティクラスには **d-{ 表示形式 }** が定義されており、表示形式には `display` プロパティで指定できる値を入れます。たとえば、`d-none` クラスは「`display: none;`」、`d-block` クラスは「`display: block;`」を設定できるクラスです。

また、ブレイクポイントの接頭辞 `sm`、`md`、`lg`、`xl` と組み合わせた **d-{ ブレイクポイント }-{ 表示形式 }** の記法をとったクラスでは、ブレイクポイントごとに表示形式（または非表示）を切り替えることができます。たとえば、`d-sm-none` クラスを指定すると、画面幅が Small 以上では表示されません（表 8-7）。このブレイクポイントによる表示／非表示の切り替えについては次項で詳しく解説します。

▼表 8-7 表示形式を設定するクラスの一覧

| クラス | レスポンシブ対応 | 機能 | 定義済みスタイル |
|-----------------------------|---|-----------------------|-------------------------------------|
| <code>d-none</code> | <code>d-{sm,md,lg,xl}-none</code> | 表示しない | <code>display: none;</code> |
| <code>d-inline</code> | <code>d-{sm,md,lg,xl}-inline</code> | インライン表示 | <code>display: inline;</code> |
| <code>d-inline-block</code> | <code>d-{sm,md,lg,xl}-inline-block</code> | インラインブロック表示 | <code>display: inline-block;</code> |
| <code>d-block</code> | <code>d-{sm,md,lg,xl}-block</code> | ブロック表示 | <code>display: block;</code> |
| <code>d-table</code> | <code>d-{sm,md,lg,xl}-table</code> | テーブル表示 | <code>display: table;</code> |
| <code>d-table-cell</code> | <code>d-{sm,md,lg,xl}-table-cell</code> | テーブルセル表示 | <code>display: table-cell;</code> |
| <code>d-table-row</code> | <code>d-{sm,md,lg,xl}-table-row</code> | テーブル行表示 | <code>display: table-row;</code> |
| <code>d-flex</code> | <code>d-{sm,md,lg,xl}-flex</code> | flex コンテナとして表示 | <code>display: flex;</code> |
| <code>d-inline-flex</code> | <code>d-{sm,md,lg,xl}-inline-flex</code> | インラインの flex コンテナとして表示 | <code>display: inline-flex;</code> |

次の例では、`div` 要素に **d-inline** クラスを指定しています。本来ブロック表示される `div` 要素がインライン表示になり、改行されずに並んで配置されます。

その逆で、インライン表示される `span` 要素に **d-block** クラスを指定するとブロック表示になり、改行されて配置されます（リスト 8-7、図 8-7）。

▼リスト 8-7 Display ユーティリティクラスの例（display.html）

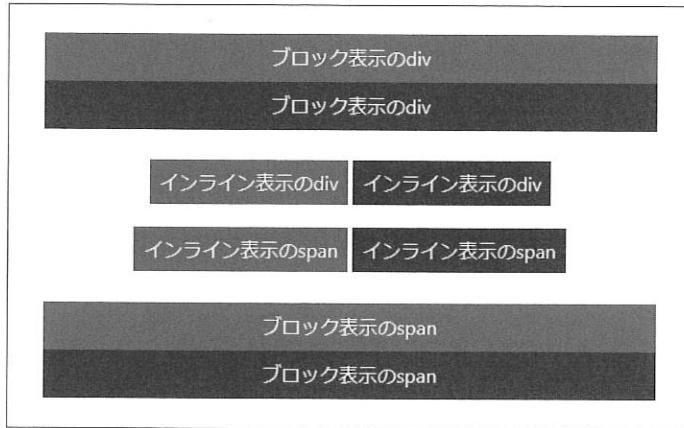
```
<div class="container text-center">
  <div class="bg-primary text-white">ブロック表示のdiv</div>
```

```

<div class="bg-dark text-white">ブロック表示のdiv</div>
</div>
<div class="container text-center">
  <div class="d-inline bg-primary text-white">インライン表示のdiv</div>
  <div class="d-inline bg-dark text-white">インライン表示のdiv</div>
</div>
<div class="container text-center">
  <span class="bg-primary text-white">インライン表示のspan</span>
  <span class="bg-dark text-white">インライン表示のspan</span>
</div>
<div class="container text-center">
  <span class="d-block bg-primary text-white">ブロック表示のspan</span>
  <span class="d-block bg-dark text-white">ブロック表示のspan</span>
</div>

```

▼図 8-7 表示形式を設定するクラスの例



8.3.2 要素の表示／非表示を設定するレスポンシブなクラス

Bootstrap における開発では、画面幅ごとに表示および非表示を切り替えるレスポンシブなクラスが頻繁に使用されます。前項のとおり、要素を非表示にするには `d-none` クラスまたは、`d-{sm,md,lg,xl}-none` クラスのいずれかを使用します。要素を表示するには、`d-block` または `d-{sm,md,lg,xl}-block` クラスを使用します。指定された画面幅の範囲でのみ要素を表示するには、これらクラスを組み合わせて使用します。

Bootstrap はモバイルファーストの方針で設定されているため、基本となる画面幅は最小の `xs` (Extra Small) サイズです。`d-xs-none` というクラスは存在せず、`d-none` クラスを指定すると、`xs ~ xl`までの全画面幅において表示されないしくみです。

`xs` の画面幅のみ非表示にしたい場合は、まず `d-none` クラスで全画面幅を非表示にした上で、`d-sm-none` クラスを追加します。これにより、`Small` 以降の画面幅で表示されるようになります。

その逆で、`d-block` を指定すると `xs ~ xl`までの画面幅で表示されます。ただし、わざわざ `d-block` クラスを指定しなくても、デフォルトで表示されている状態なので、非表示のスタイル指定がされてない限りこのクラスは不

要です（表8-8）。

▼表8-8 指定された画面幅の範囲で表示／非表示を切り替えるクラスの設定例

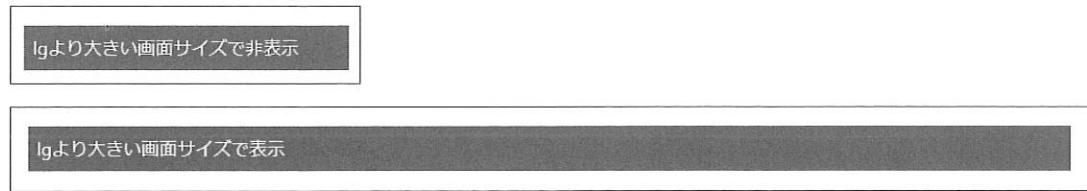
| 画面幅 | クラス |
|---------|-------------------------------------|
| すべて非表示 | d-none |
| xsのみ非表示 | d-none および d-sm-block |
| smのみ非表示 | d-sm-none および d-md-block |
| mdのみ非表示 | d-md-none および d-lg-block |
| lgのみ非表示 | d-lg-none および d-xl-block |
| xlのみ非表示 | d-xl-none |
| すべて表示 | d-block（デフォルト） |
| xsのみ表示 | d-block（デフォルト）および d-sm-none |
| smのみ表示 | d-none および d-sm-block および d-md-none |
| mdのみ表示 | d-none および d-md-block および d-lg-none |
| lgのみ表示 | d-none および d-lg-block および d-xl-none |
| xlのみ表示 | d-none および d-xl-block |

次のコードは、lgのブレイクポイントで表示・非表示を切り替えた例です（リスト8-8、図8-8）。

▼リスト8-8 要素の表示／非表示を設定するレスポンシブなクラスの例（d-none.html）

```
<div class="d-lg-none bg-primary text-white">lgより大きい画面幅で非表示</div>
<div class="d-none d-lg-block bg-secondary text-white">lgより大きい画面幅で表示</div>
```

▼図8-8 要素の表示／非表示を設定するレスポンシブなクラスの例（上：lgサイズ未満、下：lgサイズ以上）



8.3.3 印刷時の表示／非表示を設定するクラス

印刷時に要素の表示／非表示を切り替えるには、**d-print-{ 表示形式 } クラス**を使用します（表8-9）。たとえば、ナビゲーションバーや広告など、印刷不要の要素によく使用されます。

▼表 8-9 印刷時の表示／非表示を設定するクラス一覧

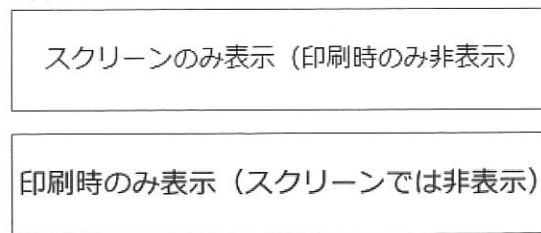
| クラス | 説明 |
|----------------------|------------------|
| d-print-none | 印刷時に非表示 |
| d-print-inline | 印刷時にインライン表示 |
| d-print-inline-block | 印刷時にインラインブロック表示 |
| d-print-block | 印刷時にブロック表示 |
| d-print-table | 印刷時にテーブル表示 |
| d-print-table-row | 印刷時にテーブル行表示 |
| d-print-table-cell | 印刷時にテーブルセル表示 |
| d-print-flex | 印刷時にフレックス表示 |
| d-print-inline-flex | 印刷時にインラインフレックス表示 |

次のコードは、d-print-none クラスを指定して印刷時に非表示になる要素と、その逆に、d-none と d-print-block クラスを指定して印刷時のみ表示する要素の例です（リスト 8-9、図 8-9）。

▼リスト 8-9 印刷時の表示／非表示を設定するクラスの例（d-print.html）

```
<div class="d-print-none">スクリーンのみ表示（印刷時のみ非表示）</div>
<div class="d-none d-print-block">印刷時のみ表示（スクリーンでは非表示）</div>
```

▼図 8-9 印刷時の表示／非表示を設定するクラスの例（上：PC 画面、下：印刷プレビュー画面）

1
2
3
4
56
78
9
10

Sizing ユーティリティ

Sizing ユーティリティを使用して、要素の幅や高さを設定することができます。このユーティリティにはデフォルト時、25%、50%、75%、100%の幅と自動幅および高さを設定するクラスが定義されています。設定を変更するにはP.451を参照してください。

8.4.1 幅を設定するクラス

要素の幅を%値で設定するには**w-{%値}クラス**を使用します。%値には、25、50、75、100、autoがあります（表8-10）。

▼表8-10 幅を設定するクラス一覧

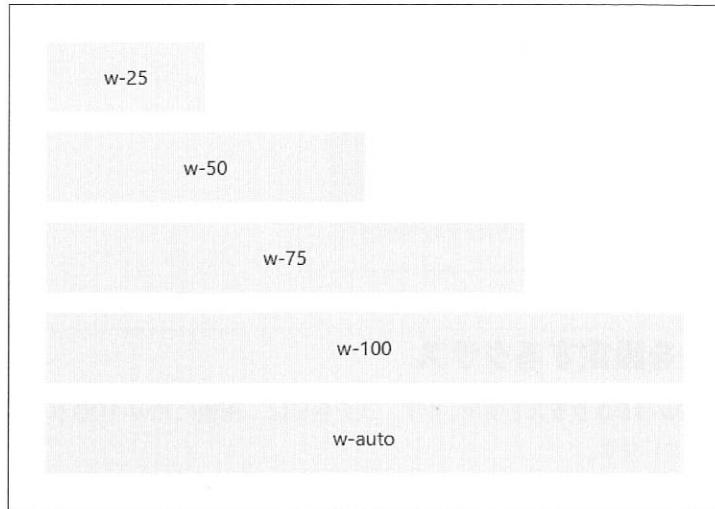
| クラス | 説明 |
|--------|----------------|
| w-25 | 25%の幅を設定する |
| w-50 | 50%の幅を設定する |
| w-75 | 75%の幅を設定する |
| w-100 | 100%の幅を設定する |
| w-auto | 自動（初期値）の幅を設定する |

親要素を基準とするので、次のサンプルではコンテナ（container）を基準に、25%、50%、75%、100%の幅で表示されます（リスト8-10、図8-10）。

▼リスト8-10 幅を設定するクラスの例（sizing-w.html）

```
<div class="container">
  <div class="w-25">w-25</div>
  <div class="w-50">w-50</div>
  <div class="w-75">w-75</div>
  <div class="w-100">w-100</div>
  <div class="w-auto">w-auto</div>
</div>
```

▼図 8-10 幅を設定するクラスの例



8.4.2 高さを設定するクラス

要素の高さを % 値で設定するには **h-{% 値} クラス** を指定します。% 値には、25、50、75、100、auto が入ります（表 8-11）。

▼表 8-11 高さを設定するクラス一覧

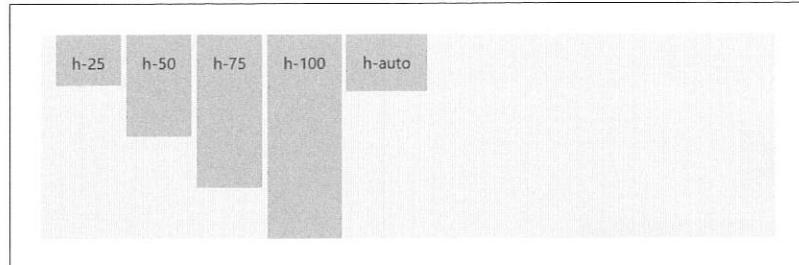
| クラス | 説明 |
|--------|-----------------|
| h-25 | 25% の高さを設定する |
| h-50 | 50% の高さを設定する |
| h-75 | 75% の高さを設定する |
| h-100 | 100% の高さを設定する |
| h-auto | 自動（初期値）の高さを設定する |

親要素を基準とするので、次のサンプルでは親要素となるコンテナ（container）に 200px の高さを指定しています。コンテナの各子要素には、d-inline-block を指定して、横並びに配置し、それぞれクラスを指定して、25%、50%、75%、100% の高さで表示しています（リスト 8-11、図 8-11）。

▼リスト 8-11 高さを設定するクラスの例 (sizing-y.html)

```
<div class="container">
  <div class="h-25 d-inline-block">h-25</div>
  <div class="h-50 d-inline-block">h-50</div>
  <div class="h-75 d-inline-block">h-75</div>
  <div class="h-100 d-inline-block">h-100</div>
  <div class="h-auto d-inline-block">h-auto</div>
</div>
```

▼図 8-11 高さを設定するクラスの例



8.4.3 最大幅 100% を設定するクラス

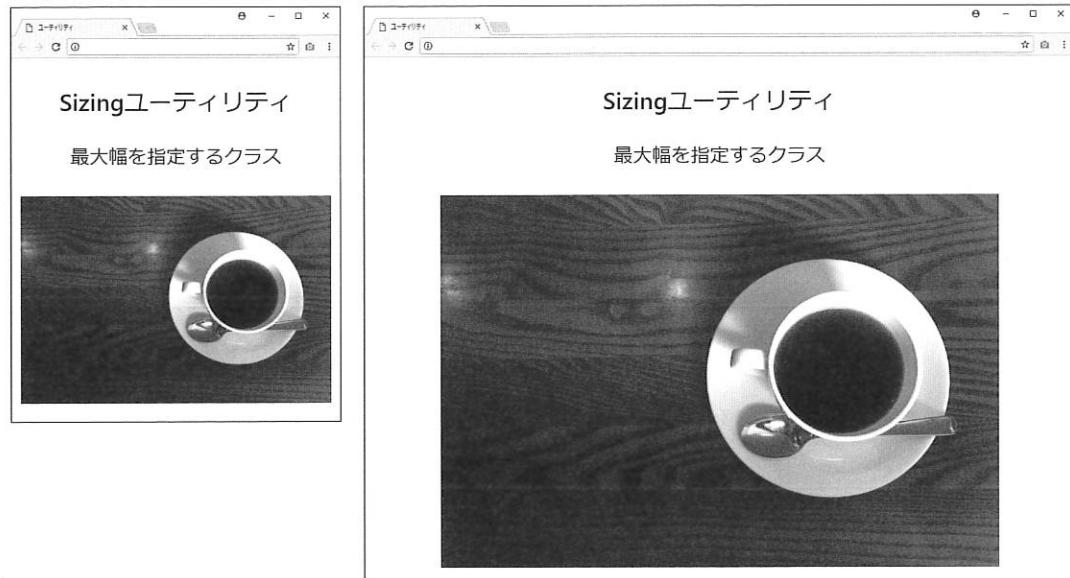
要素に最大幅 100% を指定するには **mw-100** クラスを使用します。次の例では、画像に mw-100 クラスを指定して最大幅 100% になるよう設定しています。

ブラウザのサイズを拡大、縮小してもコンテナ内に収まるため、横スクロールが出ることはありません（リスト 8-12、図 8-12）。

▼リスト 8-12 最大幅を設定するクラスの例 (sizing-mw.html)

```
<div class="container">
  
</div>
```

▼図 8-12 最大幅を設定するクラスの例



8.4.4 最大高 100% を設定するクラス

同様に、要素に最大高を指定するには **mh-100 クラス**を使用します。

次の例では、コンテナに高さ 200px を指定しています。画像の元のサイズは縦 800px ありますが、コンテナの高さの 200px に収まるよう表示されます（リスト 8-13、図 8-13）。

▼リスト 8-13 最大幅を設定するクラスの例（sizing-mh.html）

```
<div class="container">
  
</div>
```

▼図 8-13 最大高を設定するクラスの例



1
2
3
4
5
6
7
8
9
10

8

SECTION

5

Spacing ユーティリティ

Spacing ユーティリティを使用して、要素やコンポーネント間のマージンおよびパディングを設定することができます。

8.5.1 Spacing ユーティリティの記法

Spacing ユーティリティは、マージンやパディングを設定するユーティリティクラスですが、適用する辺、サイズ、ブレイクポイントなどを組み合わせて使用できるため、たくさんの種類があります。その記法は { プロパティ } { 辺 }-{ ブレイクポイント }-{ サイズ } と少々複雑な形式になっています。

たとえば、mt-md-3という形式のクラス名の場合は、{ マージン }{ 上 }-{ Medium 以上 }-{ 3 }で、Medium 以上の画面幅で上マージンを3(1rem)に設定するという意味になります。1つ1つ見ていきましょう。

■ プロパティ

{ プロパティ }には、m(マージン)またはp(パディング)のどちらかが入ります。

■ 辺

{ 辺 }には、top(上辺)、bottom(下辺)、left(左辺)、right(右辺)の頭文字を取ったt、b、l、rまたは、左右のx、上下のyが入ります。{ 辺 }を付けない場合は、4辺すべてにマージンまたはパディングを設定します。

■ ブレイクポイント

{ ブレイクポイント }には、各ブレイクポイントに対応したクラス接頭辞、sm(Small)、md(Medium)、lg(Large)、xl(Extra large)が入ります。

■ サイズ

{ サイズ }には、0~5までの数字またはautoが入ります。「0」は0、「1」は0.25rem、「2」は0.5rem、「3」は1rem、「4」は1.5rem、「5」は3rem分のマージンまたはパディングが定義されています（「rem」はルート要素のfont-sizeの高さを1とする単位）。また、autoは、マージンをautoに設定するクラスです。これはBootstrapのデフォルトの設定ですが、この内容はSassを使って編集することができます。詳しくは、P.450を参照してください。

上記をまとめた Spacing ユーティリティ一覧が表8-12と表8-13になります。

▼表 8-12 マージンに関するユーティリティ一覧

| クラス | 定義済みスタイル | レスポンシブ対応 |
|---------|-----------------|-----------------------|
| m-0 | 全マージンを 0 | m-{sm,md/lg,xl}-0 |
| m-1 | 全マージンを 0.25rem | m-{sm,md/lg,xl}-1 |
| m-2 | 全マージンを 0.5rem | m-{sm,md/lg,xl}-2 |
| m-3 | 全マージンを 1rem | m-{sm,md/lg,xl}-3 |
| m-4 | 全マージンを 1.5rem | m-{sm,md/lg,xl}-4 |
| m-5 | 全マージンを 3rem | m-{sm,md/lg,xl}-5 |
| m-auto | 全マージンを auto | m-{sm,md/lg,xl}-auto |
| mt-0 | 上マージンを 0 | mt-{sm,md/lg,xl}-0 |
| mt-1 | 上マージンを 0.25rem | mt-{sm,md/lg,xl}-1 |
| mt-2 | 上マージンを 0.5rem | mt-{sm,md/lg,xl}-2 |
| mt-3 | 上マージンを 1rem | mt-{sm,md/lg,xl}-3 |
| mt-4 | 上マージンを 1.5rem | mt-{sm,md/lg,xl}-4 |
| mt-5 | 上マージンを 3rem | mt-{sm,md/lg,xl}-5 |
| mt-auto | 上マージンを auto | mt-{sm,md/lg,xl}-auto |
| mr-0 | 右マージンを 0 | mr-{sm,md/lg,xl}-0 |
| mr-1 | 右マージンを 0.25rem | mr-{sm,md/lg,xl}-1 |
| mr-2 | 右マージンを 0.5rem | mr-{sm,md/lg,xl}-2 |
| mr-3 | 右マージンを 1rem | mr-{sm,md/lg,xl}-3 |
| mr-4 | 右マージンを 1.5rem | mr-{sm,md/lg,xl}-4 |
| mr-5 | 右マージンを 3rem | mr-{sm,md/lg,xl}-5 |
| mr-auto | 右マージンを auto | mr-{sm,md/lg,xl}-auto |
| mb-0 | 下マージンを 0 | mb-{sm,md/lg,xl}-0 |
| mb-1 | 下マージンを 0.25rem | mb-{sm,md/lg,xl}-1 |
| mb-2 | 下マージンを 0.5rem | mb-{sm,md/lg,xl}-2 |
| mb-3 | 下マージンを 1rem | mb-{sm,md/lg,xl}-3 |
| mb-4 | 下マージンを 1.5rem | mb-{sm,md/lg,xl}-4 |
| mb-5 | 下マージンを 3rem | mb-{sm,md/lg,xl}-5 |
| mb-auto | 下マージンを auto | mb-{sm,md/lg,xl}-auto |
| ml-0 | 左マージンを 0 | ml-{sm,md/lg,xl}-0 |
| ml-1 | 左マージンを 0.25rem | ml-{sm,md/lg,xl}-1 |
| ml-2 | 左マージンを 0.5rem | ml-{sm,md/lg,xl}-2 |
| ml-3 | 左マージンを 1rem | ml-{sm,md/lg,xl}-3 |
| ml-4 | 左マージンを 1.5rem | ml-{sm,md/lg,xl}-4 |
| ml-5 | 左マージンを 3rem | ml-{sm,md/lg,xl}-5 |
| ml-auto | 左マージンを auto | ml-{sm,md/lg,xl}-auto |
| mx-0 | 左右マージンを 0 | mx-{sm,md/lg,xl}-0 |
| mx-1 | 左右マージンを 0.25rem | mx-{sm,md/lg,xl}-1 |
| mx-2 | 左右マージンを 0.5rem | mx-{sm,md/lg,xl}-2 |
| mx-3 | 左右マージンを 1rem | mx-{sm,md/lg,xl}-3 |
| mx-4 | 左右マージンを 1.5rem | mx-{sm,md/lg,xl}-4 |
| mx-5 | 左右マージンを 3rem | mx-{sm,md/lg,xl}-5 |
| mx-auto | 左右マージンを auto | mx-{sm,md/lg,xl}-auto |
| my-0 | 上下マージンを 0 | my-{sm,md/lg,xl}-0 |
| my-1 | 上下マージンを 0.25rem | my-{sm,md/lg,xl}-1 |
| my-2 | 上下マージンを 0.5rem | my-{sm,md/lg,xl}-2 |
| my-3 | 上下マージンを 1rem | my-{sm,md/lg,xl}-3 |
| my-4 | 上下マージンを 1.5rem | my-{sm,md/lg,xl}-4 |
| my-5 | 上下マージンを 3rem | my-{sm,md/lg,xl}-5 |
| my-auto | 上下マージンを auto | my-{sm,md/lg,xl}-auto |

1
2
3
4
5
6
7
8
9
10

▼表 8-13 パディングに関するユーティリティ一覧

| クラス | 定義済みスタイル | レスポンシブ対応 |
|------|-----------------|--------------------|
| p-0 | 全パディングを0 | p-{sm,md,lg,xl}-0 |
| p-1 | 全パディングを0.25rem | p-{sm,md,lg,xl}-1 |
| p-2 | 全パディングを0.5rem | p-{sm,md,lg,xl}-2 |
| p-3 | 全パディングを1rem | p-{sm,md,lg,xl}-3 |
| p-4 | 全パディングを1.5rem | p-{sm,md,lg,xl}-4 |
| p-5 | 全パディングを3rem | p-{sm,md,lg,xl}-5 |
| pt-0 | 上パディングを0 | pt-{sm,md,lg,xl}-0 |
| pt-1 | 上パディングを0.25rem | pt-{sm,md,lg,xl}-1 |
| pt-2 | 上パディングを0.5rem | pt-{sm,md,lg,xl}-2 |
| pt-3 | 上パディングを1rem | pt-{sm,md,lg,xl}-3 |
| pt-4 | 上パディングを1.5rem | pt-{sm,md,lg,xl}-4 |
| pt-5 | 上パディングを3rem | pt-{sm,md,lg,xl}-5 |
| pr-0 | 右パディングを0 | pr-{sm,md,lg,xl}-0 |
| pr-1 | 右パディングを0.25rem | pr-{sm,md,lg,xl}-1 |
| pr-2 | 右パディングを0.5rem | pr-{sm,md,lg,xl}-2 |
| pr-3 | 右パディングを1rem | pr-{sm,md,lg,xl}-3 |
| pr-4 | 右パディングを1.5rem | pr-{sm,md,lg,xl}-4 |
| pr-5 | 右パディングを3rem | pr-{sm,md,lg,xl}-5 |
| pb-0 | 下パディングを0 | pb-{sm,md,lg,xl}-0 |
| pb-1 | 下パディングを0.25rem | pb-{sm,md,lg,xl}-1 |
| pb-2 | 下パディングを0.5rem | pb-{sm,md,lg,xl}-2 |
| pb-3 | 下パディングを1rem | pb-{sm,md,lg,xl}-3 |
| pb-4 | 下パディングを1.5rem | pb-{sm,md,lg,xl}-4 |
| pb-5 | 下パディングを3rem | pb-{sm,md,lg,xl}-5 |
| pl-0 | 左パディングを0 | pl-{sm,md,lg,xl}-0 |
| pl-1 | 左パディングを0.25rem | pl-{sm,md,lg,xl}-1 |
| pl-2 | 左パディングを0.5rem | pl-{sm,md,lg,xl}-2 |
| pl-3 | 左パディングを1rem | pl-{sm,md,lg,xl}-3 |
| pl-4 | 左パディングを1.5rem | pl-{sm,md,lg,xl}-4 |
| pl-5 | 左パディングを3rem | pl-{sm,md,lg,xl}-5 |
| px-0 | 左右パディングを0 | px-{sm,md,lg,xl}-0 |
| px-1 | 左右パディングを0.25rem | px-{sm,md,lg,xl}-1 |
| px-2 | 左右パディングを0.5rem | px-{sm,md,lg,xl}-2 |
| px-3 | 左右パディングを1rem | px-{sm,md,lg,xl}-3 |
| px-4 | 左右パディングを1.5rem | px-{sm,md,lg,xl}-4 |
| px-5 | 左右パディングを3rem | px-{sm,md,lg,xl}-5 |
| py-0 | 上下パディングを0 | py-{sm,md,lg,xl}-0 |
| py-1 | 上下パディングを0.25rem | py-{sm,md,lg,xl}-1 |
| py-2 | 上下パディングを0.5rem | py-{sm,md,lg,xl}-2 |
| py-3 | 上下パディングを1rem | py-{sm,md,lg,xl}-3 |
| py-4 | 上下パディングを1.5rem | py-{sm,md,lg,xl}-4 |
| py-5 | 上下パディングを3rem | py-{sm,md,lg,xl}-5 |

8.5.2 自動マージンの応用

mx-auto クラスは、左右マージンを auto に設定するクラスです。これを使用すると、固定幅のブロックレベルコンテンツを水平方向にセンタリングすることができます（リスト 8-14、図 8-14）。

▼リスト 8-14 自動マージンによる水平方向中央揃え（mx-auto.html）

```
<div class="mx-auto">mx-auto</div>
```

▼図 8-14 mx-auto クラスで要素を水平方向にセンタリングした例



Bootstrap 3 では、ブロックレベルの中央寄せの方法として、center-block クラスが使用されていました。Bootstrap 4 では、center-block クラスは廃止され、新しく mx-auto クラスが定義されたので注意しましょう。

また、左マージンや右マージンだけを自動で配置する ml-auto や mr-auto クラスは、2つ以上の要素を左右に分離して配置する場合に、flexbox (P.322 参照) の柔軟性を発揮できる非常に便利なクラスです。flexbox における左右の自動マージンの具体的な使用方法については、P.334 を参照してください。

8

SECTION

6

Flex ユーティリティ

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

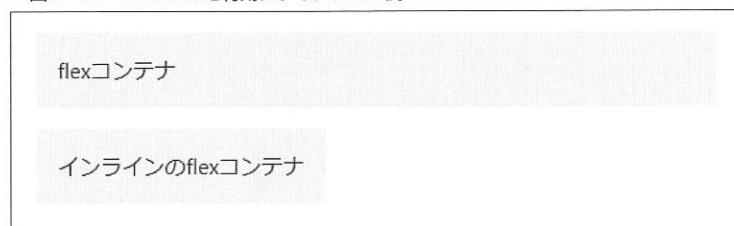
8.6.1 flexbox を有効にするクラス

flexbox を利用するには、要素に Display ユーティリティ (P.310 参照) の **d-flex クラス** (`display:flex`) または **d-inline-flex クラス** (`display:inline-flex`) を追加します。これにより、要素は **flex コンテナ**となり、またその子要素は自動的に **flex アイテム**となります。flex アイテムとなった子要素は、flex コンテナ内で任意の方向にレイアウトすることが可能になります (リスト 8-15、図 8-15)。

▼リスト 8-15 flexbox を有効にするクラスの例 (d-flex.html)

```
<div class="d-flex">d-flex</div>
<div class="d-inline-flex">d-inline-flex</div>
```

▼図 8-15 flexbox を有効にするクラスの例



また各ブレイクポイントに対応する接頭辞 sm、md、lg、xl 付きの **d-{sm,md,lg,xl}-flex クラス**、および **d-{sm,md,lg,xl}-inline-flex クラス**も定義されています。たとえば、**d-sm-flex クラス**を指定すると、Small 以上の画面幅で表示形式を flex コンテナとして設定することができます (表 8-14)。

▼表 8-14 flexbox を有効にするクラス一覧

| クラス | レスポンシブ対応クラス | 説明 |
|---------------|-----------------------------|-----------------------|
| d-flex | d-{sm,md,lg,xl}-flex | flex コンテナをブロックレベルにする |
| d-inline-flex | d-{sm,md,lg,xl}-inline-flex | flex コンテナをインラインレベルにする |

そもそも Bootstrap 4 のレイアウトは flexbox をもとに構築されており、ほとんどのコンポーネントにおいても flexbox が使えるようになっていますが、すべての要素の表示形式が「display:flex」になっているわけではありません（この表示形式による多くの不必要的上書きや、キーブラウザの挙動に起こる予期せぬ変更を避けるため）。flexbox のサイジング、位置合わせ、スペーシングなど、その他の Flex ユーティリティを使用できるようにするには、このユーティリティクラスが必要になります。

8.6.2 flex コンテナの主軸方向を設定するクラス

flex-{ 方向 } クラスを指定すると、flex コンテナの**主軸**の方向を設定できます。flex アイテムはこの主軸の方向に沿って配置されます。{ 方向 } は、row（左から右）、reverse（右から左）、column（上から下）、reverse（下から上）から選択できます。

また、各ブレイクポイントに対応する接頭辞 sm、md、lg、xl 付きの **flex-{sm,md,lg,xl}-{ 方向 }** クラスも定義されています。たとえば、flex コンテナに **flex-md-reverse** クラスを指定すると、Small 以上の画面幅で flex アイテムを右から左に配置します（表 8-15）。

▼表 8-15 flex コンテナの主軸方向を設定するクラス一覧

| クラス | レスポンシブ対応クラス | 説明 |
|---------------------|-----------------------------------|---------|
| flex-row | flex-{sm,md,lg,xl}-row | 左から右に配置 |
| flex-row-reverse | flex-{sm,md,lg,xl}-row-reverse | 右から左に配置 |
| flex-column | flex-{sm,md,lg,xl}-column | 上から下に配置 |
| flex-column-reverse | flex-{sm,md,lg,xl}-column-reverse | 下から上に配置 |

ほとんどのケースでは、ブラウザのデフォルトは flex-direction:row（左から右に配置）であるため、このクラスを省略できます。しかし、レスポンシブレイアウトなどで、この値を明示的に設定しなければならない場合にこれを使用します（リスト 8-16、図 8-16）。

▼リスト 8-16 flex コンテナの主軸方向を設定するクラスの例 (flex-direction.html)

```
<h3>左から右 (flex-row) </h3>
<div class="d-flex flex-row">
  <div>flexアイテム01</div>
  <div>flexアイテム02</div>
  <div>flexアイテム03</div>
</div>
<h3>右から左 (flex-row-reverse) </h3>
<div class="d-flex flex-row-reverse">
  <div>flexアイテム01</div>
  <div>flexアイテム02</div>
  <div>flexアイテム03</div>
</div>
<h3>上から下 (flex-column) </h3>
```

1
2
3
4
5
6
7
8
9
10

```
<div class="d-flex flex-column">
  <div>flexアイテム01</div>
  <div>flexアイテム02</div>
  <div>flexアイテム03</div>
</div>
<h3>下から上 (flex-column-reverse) </h3>
<div class="d-flex flex-column-reverse">
  <div>flexアイテム01</div>
  <div>flexアイテム02</div>
  <div>flexアイテム03</div>
</div>
```

▼図 8-16 flex コンテナの主軸方向を設定するクラスの例

左から右 (flex-row)

| | | |
|------------|------------|------------|
| flexアイテム01 | flexアイテム02 | flexアイテム03 |
|------------|------------|------------|

右から左 (flex-row-reverse)

| | | |
|------------|------------|------------|
| flexアイテム03 | flexアイテム02 | flexアイテム01 |
|------------|------------|------------|

上から下 (flex-column)

| |
|------------|
| flexアイテム01 |
|------------|

| |
|------------|
| flexアイテム02 |
|------------|

| |
|------------|
| flexアイテム03 |
|------------|

下から上 (flex-column-reverse)

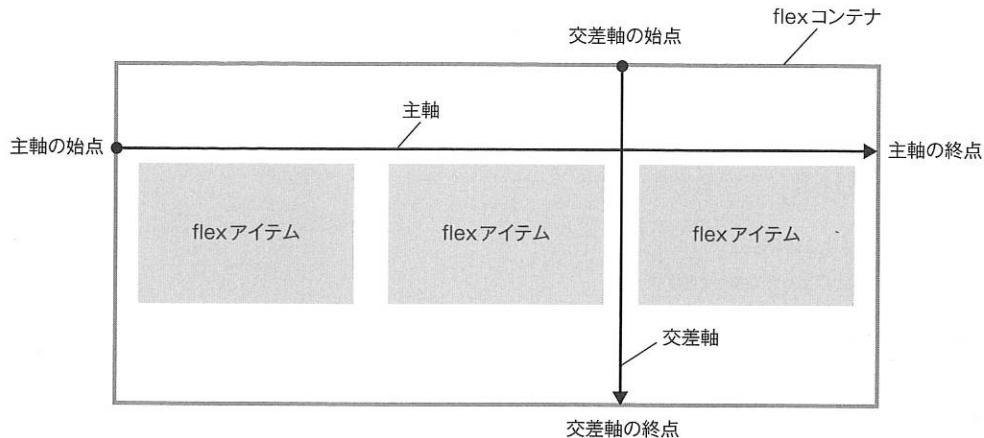
| |
|------------|
| flexアイテム03 |
|------------|

| |
|------------|
| flexアイテム02 |
|------------|

| |
|------------|
| flexアイテム01 |
|------------|

flexbox は 2 つの軸を持っており、主軸に対して垂直な軸を**交差軸**と言います。flex アイテムは、flex コンテナの主軸に沿って平行に配置されますが、それぞれの軸の始点や終点も flex アイテムのレイアウトの基準として使われます。図 8-17 は、flex-row の場合の主軸と交差軸、それぞれの軸の始点と終点を示したものです。flex-column の場合は、主軸が垂直、交差軸が水平になります。

▼図 8-17 主軸方向が flex-row の場合の主軸と交差軸



8.6.3 主軸方向の整列をするクラス

justify-content-{ 整列方法 } クラスを使用すると、flex コンテナの主軸に沿って、flex アイテムをどのように配置するかを設定できます。主軸方向が row (デフォルト) の場合は、水平方向の配置を制御します。{ 整列方法 } は、start (始点)、end (終点)、center (中央)、between (両端から均等)、around (等間隔) から選択できます。ブラウザのデフォルトは start です。

また各ブレイクポイントに対応する接頭辞 sm、md、lg、xl 付きの **justify-content-{sm,md,lg,xl}-{ 整列方法 } クラス**も定義されています。主軸方向の整列をするクラス一覧は表 8-16 のとおりです。

▼表 8-16 主軸方向の整列をするクラス一覧

| クラス | レスポンシブ対応クラス | 説明 |
|-------------------------|---------------------------------------|------------------------|
| justify-content-start | justify-content-{sm,md,lg,xl}-start | 主軸の始点に揃える (デフォルト) |
| justify-content-end | justify-content-{sm,md,lg,xl}-end | 主軸の終点に揃える |
| justify-content-center | justify-content-{sm,md,lg,xl}-center | 主軸の中央に揃える |
| justify-content-between | justify-content-{sm,md,lg,xl}-between | flex アイテムを両端から均等に揃える |
| justify-content-around | justify-content-{sm,md,lg,xl}-around | 主軸に対し flex アイテムを等間隔に配置 |

justify-content-between と **justify-content-around** クラスの違いがわかりにくいのですが、**justify-content-between** クラスは、最初の flex アイテムを主軸の始点、最後の flex アイテムを終点に揃え、残りを均等に配置するのに対し、**justify-content-around** クラスは、主軸に対しすべての flex アイテムを等間隔に配置します（リスト 8-17、図 8-18）。

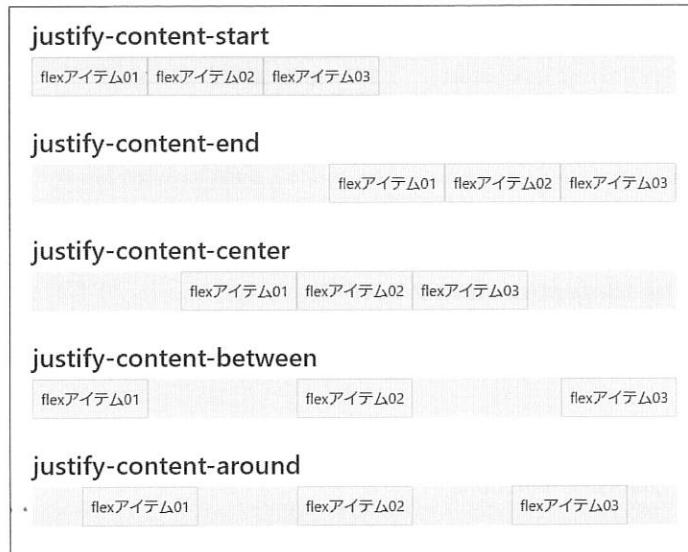
▼リスト 8-17 主軸方向の整列をするクラスの例 (justify-content.html)

```
<h3>justify-content-start</h3>
<div class="d-flex justify-content-start">
  <div>flexアイテム01</div>
```

1
2
3
4
5
6
7
8
9
10

```
<div>flexアイテム02</div>
<div>flexアイテム03</div>
</div>
<h3>justify-content-end</h3>
<div class="d-flex justify-content-end">
  <div>flexアイテム01</div>
  <div>flexアイテム02</div>
  <div>flexアイテム03</div>
</div>
<h3>justify-content-center</h3>
<div class="d-flex justify-content-center">
  <div>flexアイテム01</div>
  <div>flexアイテム02</div>
  <div>flexアイテム03</div>
</div>
<h3>justify-content-between</h3>
<div class="d-flex justify-content-between">
  <div>flexアイテム01</div>
  <div>flexアイテム02</div>
  <div>flexアイテム03</div>
</div>
<h3>justify-content-around</h3>
<div class="d-flex justify-content-around">
  <div>flexアイテム01</div>
  <div>flexアイテム02</div>
  <div>flexアイテム03</div>
</div>
```

▼図 8-18 主軸方向の整列をするクラスの例



8.6.4 交差軸方向の整列をするクラス

align-items-{ 整列方法 } クラスを使用して、交差軸の flex アイテムの配置を設定できます。主軸方向が row (デフォルト) の場合は、垂直方向の配置を制御します。{ 整列方法 } は、stretch (交差軸に合わせて伸縮)、start (始点)、end (終点)、center (中央)、baseline (ペースライン) から選択できます。デフォルトは stretch です。

また、各ブレイクポイントの接頭辞 sm、md、lg、xl 付きの **align-items-{sm,md,lg,xl}-{ 整列方法 }** クラスも定義されています (表 8-17、リスト 8-18、図 8-19)。

▼表 8-17 交差軸方向の整列をするクラス一覧

| クラス | レスポンシブ対応クラス | 説明 |
|----------------------|------------------------------------|---------------------------|
| align-items-start | align-items-{sm,md,lg,xl}-start | 交差軸の始点に配置 |
| align-items-end | align-items-{sm,md,lg,xl}-end | 交差軸の終点に配置 |
| align-items-center | align-items-{sm,md,lg,xl}-center | 交差軸の中央に配置 |
| align-items-baseline | align-items-{sm,md,lg,xl}-baseline | flex アイテム内のペースラインが揃うように配置 |
| align-items-stretch | align-items-{sm,md,lg,xl}-stretch | 交差軸の幅 (高さ) に合わせて伸縮して配置 |

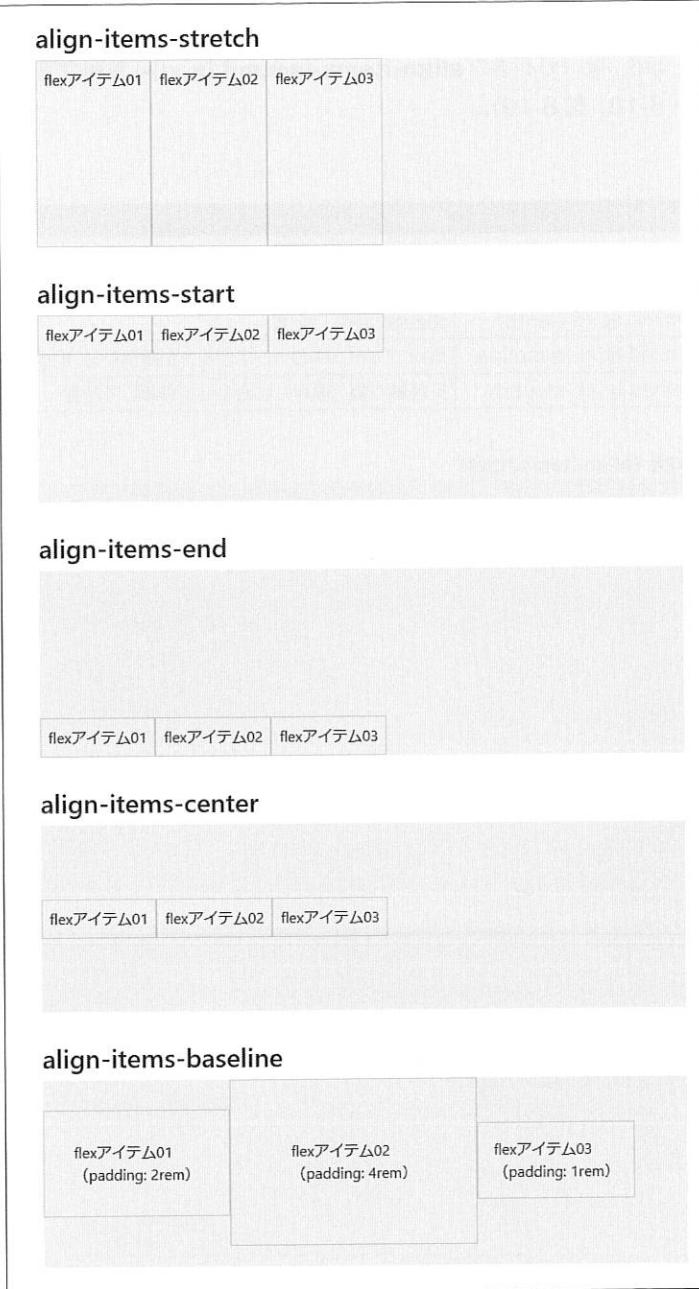
▼リスト 8-18 交差軸方向の整列をするクラスの例 (align-items.html)

```
<h3>align-items-stretch</h3>
<div class="d-flex align-items-stretch" style="height: 200px">
  <div>flexアイテム01</div>
  <div>flexアイテム02</div>
  <div>flexアイテム03</div>
</div>
<h3>align-items-start</h3>
<div class="d-flex align-items-start" style="height: 200px">
  <div>flexアイテム01</div>
  <div>flexアイテム02</div>
  <div>flexアイテム03</div>
</div>
<h3>align-items-end</h3>
<div class="d-flex align-items-end" style="height: 200px">
  <div>flexアイテム01</div>
  <div>flexアイテム02</div>
  <div>flexアイテム03</div>
</div>
<h3>align-items-center</h3>
<div class="d-flex align-items-center" style="height: 200px">
  <div>flexアイテム01</div>
  <div>flexアイテム02</div>
  <div>flexアイテム03</div>
</div>
<h3>align-items-baseline</h3>
<div class="d-flex align-items-baseline" style="height: 200px">
  <div style="padding: 2rem">flexアイテム01 <br> (padding: 2rem) </div>
```

1
2
3
4
5
6
7
8
9
10

```
<div style="padding: 4rem">flexアイテム02 <br> (padding: 4rem) </div>
<div style="padding: 1rem">flexアイテム03 <br> (padding: 1rem) </div>
</div>
```

▼図 8-19 交差軸方向の整列をするクラスの例



8.6.5 flex アイテムを交差軸上で個別に整列するクラス

align-self-{ 整列方法 } クラスを使用して、flex アイテムの交差軸上の配置を個別に変更できます。{ 整列方法 } は、前項の align-items と同じ stretch (ブラウザのデフォルト)、start、end、center、baseline から選択可能です。また各ブレイクポイントに対応する接頭辞 sm、md、lg、xl 付きの **align-self-{sm,md,lg,xl}-{ 整列方法 } クラス**も定義されています。flex アイテムを交差軸上で個別に整列するクラス一覧は表 8-18 のとおりです。

▼表 8-18 flex アイテムを交差軸上で個別に整列するクラス一覧

| クラス | レスポンシブ対応クラス | 説明 |
|---------------------|-----------------------------------|---------------------------|
| align-self-start | align-self-{sm,md,lg,xl}-start | 交差軸の始点に配置 |
| align-self-end | align-self-{sm,md,lg,xl}-end | 交差軸の終点に配置 |
| align-self-center | align-self-{sm,md,lg,xl}-center | 交差軸の中央に配置 |
| align-self-baseline | align-self-{sm,md,lg,xl}-baseline | flex アイテム内のベースラインが揃うように配置 |
| align-self-stretch | align-self-{sm,md,lg,xl}-stretch | 交差軸の幅 (高さ) に合わせて伸縮して配置 |

前項の align-items が flex コンテナに指定するのに対し、このクラスは個々の flex アイテムに指定します。また align-item より優先されます（リスト 8-19、図 8-20）。

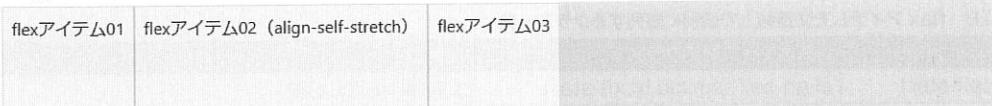
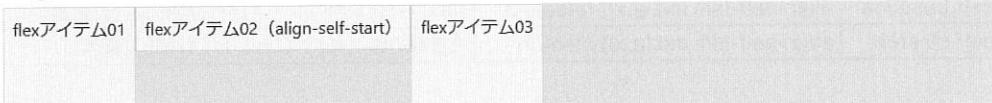
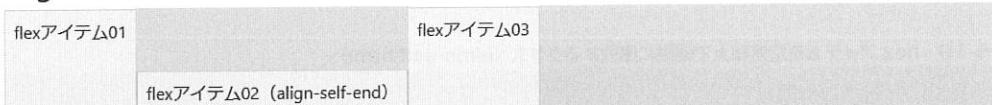
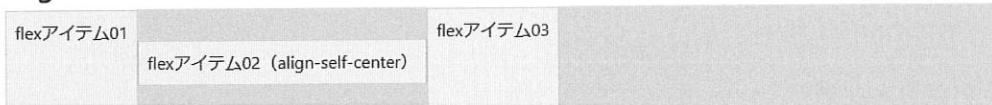
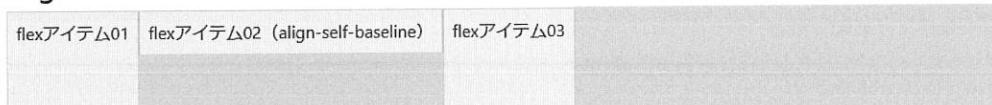
▼リスト 8-19 flex アイテムを交差軸上で個別に整列するクラス（align-self.html）

```
<h3>align-self-stretch</h3>
<div class="d-flex" style="height: 100px">
  <div>flexアイテム01</div>
  <div class="align-self-stretch">flexアイテム02 (align-self-stretch) </div>
  <div>flexアイテム03</div>
</div>
<h3>align-self-start</h3>
<div class="d-flex" style="height: 100px">
  <div>flexアイテム01</div>
  <div class="align-self-start">flexアイテム02 (align-self-start) </div>
  <div>flexアイテム03</div>
</div>
<h3>align-self-end</h3>
<div class="d-flex" style="height: 100px">
  <div>flexアイテム01</div>
  <div class="align-self-end">flexアイテム02 (align-self-end) </div>
  <div>flexアイテム03</div>
</div>
<h3>align-self-center</h3>
<div class="d-flex" style="height: 100px">
  <div>flexアイテム01</div>
  <div class="align-self-center">flexアイテム02 (align-self-center) </div>
  <div>flexアイテム03</div>
</div>
```

1
2
3
4
5
6
7
8
9
10

```
<h3>align-self-baseline</h3>
<div class="d-flex" style="height: 100px">
  <div>flexアイテム01</div>
  <div class="align-self-baseline">flexアイテム02 (align-self-baseline) </div>
  <div>flexアイテム03</div>
</div>
```

▼図 8-20 flex アイテムを交差軸上で個別に整列するクラス

align-self-stretch**align-self-start****align-self-end****align-self-center****align-self-baseline**

8.6.6 flex コンテナ全幅に渡って等幅で整列するクラス

flex-fill クラスを使用して、flex コンテナ全幅に渡って flex アイテムが等幅で並ぶように設定することができます。また各ブレイクポイントに対応する接頭辞 sm、md、lg、xl 付きの **flex-{sm,md,lg,xl}-fill** クラスも定義されています（表 8-19、リスト 8-20、図 8-21）。

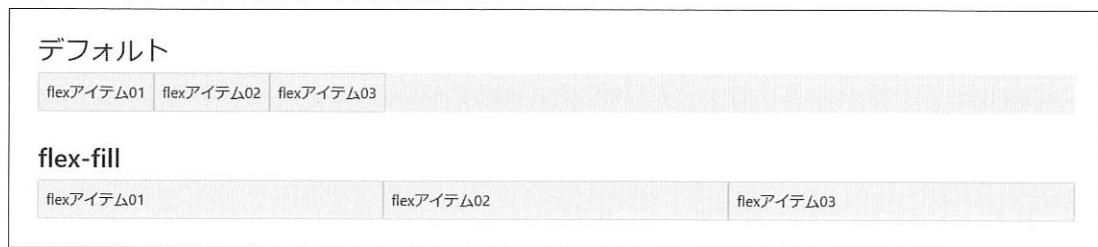
▼表 8-19 flex コンテナ全幅に渡って等幅で整列するクラス一覧

| クラス | レスポンシブ対応クラス | 説明 |
|-----------|-------------------------|----------------------|
| flex-fill | flex-{sm,md,lg,xl}-fill | flex コンテナ全幅に渡って等幅で整列 |

▼リスト 8-20 flex コンテナ全幅に渡って等幅で整列するクラス (flex-fill.html)

```
<div class="d-flex">
  <div class="flex-fill">flexアイテム01</div>
  <div class="flex-fill">flexアイテム02</div>
  <div class="flex-fill">flexアイテム03</div>
</div>
```

▼図 8-21 flex コンテナ全幅に渡って等幅で整列するクラス



8.6.7 flex アイテムの幅の伸縮を指定するクラス

flex-grow-* クラスを使用して、flex アイテムの幅を伸長し、flex コンテナの余白領域を埋めることができます。* には「0」（伸長しない）または「1」（伸長する）が入ります。その逆に、**flex-shrink-*** クラスを使用して、flex アイテムの幅を縮小することができます。* には「0」（縮小しない）または「1」（縮小する）が入ります（表 8-20）。

▼表 8-20 flex アイテムの伸縮を指定するクラス一覧

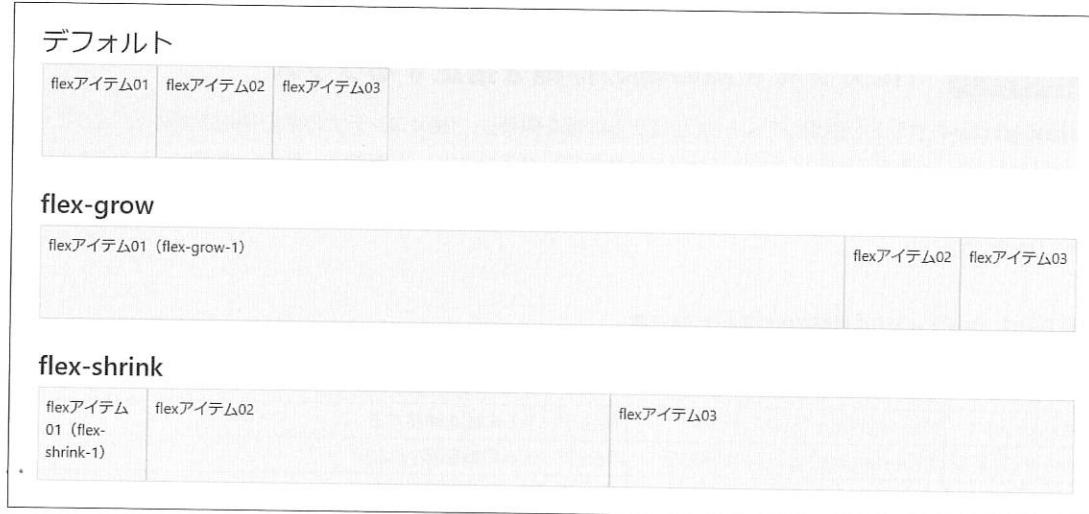
| クラス | レスポンシブ対応クラス | 説明 |
|---------------|-----------------------------|-------------------|
| flex-grow-0 | flex-{sm,md,lg,xl}-grow-0 | flex アイテムの幅を伸長しない |
| flex-grow-1 | flex-{sm,md,lg,xl}-grow-1 | flex アイテムの幅を伸長する |
| flex-shrink-0 | flex-{sm,md,lg,xl}-shrink-0 | flex アイテムの幅を縮小しない |
| flex-shrink-1 | flex-{sm,md,lg,xl}-shrink-1 | flex アイテムの幅を縮小する |

次の例（リスト8-21、図8-22）では、flexアイテムにflex-grow-1クラスを追加して幅を伸長し、flexコンテナの余白領域を埋めています（①）。また、flexアイテムにflex-shrink-1クラスを追加して幅を縮小し、w-100クラスが設定された他のflexコンテナのスペースを増やすことで、flexコンテナの余白領域を埋めています（②）。

▼リスト8-21 flexアイテムの伸縮を指定するクラス（flex-grow-shrink.html）

```
<h3>デフォルト</h3>
<div class="d-flex" style="height: 100px">
  <div class="p-2">flexアイテム01</div>
  <div class="p-2">flexアイテム02</div>
  <div class="p-2">flexアイテム03</div>
</div>
<!-- flexアイテムの幅を伸長する -->
<h3>flex-grow</h3>
<div class="d-flex" style="height: 100px">
  <div class="p-2 flex-grow-1">flexアイテム01 (flex-grow-1) </div> ①
  <div class="p-2">flexアイテム02</div>
  <div class="p-2">flexアイテム03</div>
</div>
<!-- flexアイテムの幅を縮小する -->
<h3>flex-shrink</h3>
<div class="d-flex" style="height: 100px">
  <div class="p-2 flex-shrink-1">flexアイテム01 (flex-shrink-1) </div> ②
  <div class="p-2 w-100">flexアイテム02</div>
  <div class="p-2 w-100">flexアイテム03</div>
</div>
```

▼図8-22 flexアイテムの伸縮を指定するクラス



8.6.8 flex アイテムの折り返しを設定するクラス

flex アイテムが flex コンテナ内でどのように折り返すかを指定するには、**flex-nowrap クラス**（折り返しなし）、**flex-wrap クラス**（折り返し）または**flex-wrap-reverse クラス**（逆方向へ折り返し）を選択して設定することができます（表 8-21、リスト 8-22、図 8-23）。

▼表 8-21 flex アイテムの折り返しを設定するクラス一覧

| クラス | レスポンシブ対応クラス | 説明 |
|-------------------|----------------------------------|----------|
| flex-nowrap | flex-[sm,md,lg,xl]-nowrap | 折り返しなし |
| flex-wrap | flex-[sm,md,lg,xl]-wrap | 折り返し |
| flex-wrap-reverse | fflex-[sm,md,lg,xl]-wrap-reverse | 逆方向へ折り返し |

▼リスト 8-22 flex アイテムの折り返しを設定するクラス（flex-wrap.html）

```
<h3>flex-nowrap</h3>
<div class="d-flex flex-nowrap">
  <div>flexアイテム01</div>
  ...中略...
  <div>flexアイテム12</div>
</div>
<h3>flex-wrap</h3>
<div class="d-flex flex-wrap">
  ...中略...
</div>
<h3>flex-wrap-reverse</h3>
<div class="d-flex flex-wrap-reverse">
  ...中略...
</div>
```

▼図 8-23 flex アイテムの折り返しを設定するクラス

| flex-nowrap | | | | | | | | | | | |
|-------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| flexアイテム01 | flexアイテム02 | flexアイテム03 | flexアイテム04 | flexアイテム05 | flexアイテム06 | flexアイテム07 | flexアイテム08 | flexアイテム09 | flexアイテム10 | flexアイテム11 | flexアイテム12 |

| flex-wrap | | | | | | | |
|------------|------------|------------|------------|------------|------------|------------|--|
| flexアイテム01 | flexアイテム02 | flexアイテム03 | flexアイテム04 | flexアイテム05 | flexアイテム06 | flexアイテム07 | |
| flexアイテム08 | flexアイテム09 | flexアイテム10 | flexアイテム11 | flexアイテム12 | | | |

| flex-wrap-reverse | | | | | | |
|-------------------|------------|------------|------------|------------|------------|------------|
| flexアイテム08 | flexアイテム09 | flexアイテム10 | flexアイテム11 | flexアイテム12 | | |
| flexアイテム01 | flexアイテム02 | flexアイテム03 | flexアイテム04 | flexアイテム05 | flexアイテム06 | flexアイテム07 |

8.6.9 特定の flex アイテムの表示順序を入れ替えるクラス

特定の flex アイテムの表示順序を変更するには、**order-{ 順番 } クラス** 使用します。{ 順番 } は、flex アイテムを並べたい順に 0 ~ 12 の数値を設定可能です。同じ順番のクラスを指定された要素は、ソースコード内の記載順に配置されます。

また、各ブレイクポイントに対応する接頭辞 sm、md、lg、xl 付きの **order-{sm,md,lg,xl}-{ 順番 } クラス** も定義されていますが、ブレイクポイントの接頭辞が設定されていない要素が優先的に先に表示されます。設定する際は、{sm,md,lg,xl} を揃えるようにしてください。

特定の flex アイテムの表示順序を入れ替えるクラス一覧は表 8-22 のとおりです。

▼表 8-22 特定の flex アイテムの表示順序を入れ替えるクラス一覧

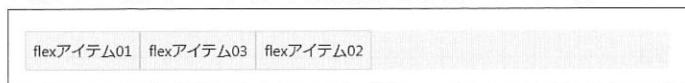
| クラス | レスポンシブ対応クラス | 説明 |
|----------------|------------------------------|---------------|
| order-{0 ~ 12} | order-{sm,md,lg,xl}-{0 ~ 12} | 値が小さい要素から順に配置 |

次の例では、HTML 上では、flex アイテム 01、flex アイテム 02、flex アイテム 03 の順で書かれていますが、**order-1**、**order-2**、**order-3** で指定された順に表示されます（リスト 8-23、図 8-24）。

▼リスト 8-23 特定の flex アイテムの表示順序を入れ替えるクラスの例（flex-order.html）

```
<div class="d-flex">
  <div class="order-1">flexアイテム01</div>
  <div class="order-3">flexアイテム02</div>
  <div class="order-2">flexアイテム03</div>
</div>
```

▼図 8-24 特定の flex アイテムの表示順序を入れ替えるクラスの例



8.6.10 自動マージンで flex アイテムを分離する

Flex ユーティリティの整列と Spacing ユーティリティの自動マージン（P.321 参照）を混在させると、flexbox の柔軟な機能を利用できます。

たとえば、ある flex アイテム以降の flex アイテムを右または左に分離したい場合に便利です。右側にアイテムを分離したい場合は、**mr-auto クラス** を右に分離したい直前のアイテムに入れます。左側にアイテムを分離したい場合は、**ml-auto クラス** を左に分離したい直後のアイテムに入れます（リスト 8-24、図 8-25）。

ただし、Internet Explorer 10、11 では、親要素がデフォルト以外の justify-content の値を持つ flex アイテムの自動マージンを正しくサポートしていないので注意してください。

▼リスト 8-24 自動マージンで flex アイテムを分離した例 (mr-auto.html)

```
<h3>デフォルト（自動マージンなし）</h3>
<div class="d-flex">
  <div>Flexアイテム1</div>
  <div>Flexアイテム2</div>
  <div>Flexアイテム3</div>
</div>
<h3>右側に分離（mr-auto）</h3>
<div class="d-flex">
  <div class="mr-auto">Flexアイテム1</div>
  <div>Flexアイテム2</div>
  <div>Flexアイテム3</div>
</div>
<h3>左側に分離（ml-auto）</h3>
<div class="d-flex">
  <div>Flexアイテム1</div>
  <div>Flexアイテム2</div>
  <div class="ml-auto p-2">Flexアイテム3</div>
</div>
```

▼図 8-25 自動マージンで flex アイテムを分離した例



8.6.11 複数行における flex アイテムの交差軸の整列をするクラス

align-content-{ 整列方法 } クラスを使用して、flex アイテムを交差軸上に整列できます。注意点として、單一行の flex アイテムは制御できません。{ 整列方法 } は、start、end、center、between、around、stretch から選択可能です。また、各ブレイクポイントに対応する接頭辞 sm、md、lg、xl 付きの **align-content-{sm,md,lg,xl}-{ 整列方法 }** クラスも定義されています（表 8-23）。

▼表 8-23 複数行における flex アイテムの交差軸の整列をするクラス

| クラス | レスポンシブ対応クラス | 説明 |
|-----------------------|-------------------------------------|------------------|
| align-content-start | align-content-{sm,md,lg,xl}-start | 行を交差軸の始点に揃えて配置 |
| align-content-end | align-content-{sm,md,lg,xl}-end | 行を交差軸の終点に揃えて配置 |
| align-content-center | align-content-{sm,md,lg,xl}-center | 行を交差軸の中央に配置 |
| align-content-between | align-content-{sm,md,lg,xl}-center | 行を両端から均等に並べる |
| align-content-around | align-content-{sm,md,lg,xl}-around | 行を等間隔に並べる |
| align-content-stretch | align-content-{sm,md,lg,xl}-stretch | 行を交差軸いっぱいに伸縮して配置 |

次の例では、これらユーティリティの動きを確認するために、flex-wrap: wrap を強制し、flex アイテムの数を増やしています（リスト 8-25、図 8-26）。

▼リスト 8-25 複数行における flex アイテムの交差軸の整列をするクラス（align-content.html）

```

<h3>align-content-start</h3>
<div class="d-flex align-content-start flex-wrap" style="height: 200px">
  <div>flexアイテム01</div>
  <div>flexアイテム02</div>
  <div>flexアイテム03</div>
  <div>flexアイテム04</div>
  <div>flexアイテム05</div>
  <div>flexアイテム06</div>
  <div>flexアイテム07</div>
  <div>flexアイテム08</div>
  <div>flexアイテム09</div>
  <div>flexアイテム10</div>
  <div>flexアイテム11</div>
  <div>flexアイテム12</div>
</div>
<h3>align-content-end</h3>
<div class="d-flex align-content-end flex-wrap" style="height: 200px">
  ...中略...
</div>
<h3>align-content-center</h3>
<div class="d-flex align-content-center flex-wrap" style="height: 200px">
  ...中略...
</div>
<h3>align-content-between</h3>
<div class="d-flex align-content-between flex-wrap" style="height: 200px">
  ...中略...
</div>
<h3>align-content-around</h3>
<div class="d-flex align-content-around flex-wrap" style="height: 200px">
  ...中略...
</div>
<h3>align-content-stretch</h3>
<div class="d-flex align-content-stretch flex-wrap" style="height: 200px">
  ...中略...
</div>

```

▼図 8-26 複数行における flex アイテムの交差軸の整列をするクラス

| align-content-start | | | | |
|-----------------------|------------|------------|------------|------------|
| flexアイテム01 | flexアイテム02 | flexアイテム03 | flexアイテム04 | flexアイテム05 |
| flexアイテム06 | flexアイテム07 | flexアイテム08 | flexアイテム09 | flexアイテム10 |
| flexアイテム11 | flexアイテム12 | | | |
| | | | | |
| align-content-end | | | | |
| flexアイテム01 | flexアイテム02 | flexアイテム03 | flexアイテム04 | flexアイテム05 |
| flexアイテム06 | flexアイテム07 | flexアイテム08 | flexアイテム09 | flexアイテム10 |
| flexアイテム11 | flexアイテム12 | | | |
| | | | | |
| align-content-center | | | | |
| flexアイテム01 | flexアイテム02 | flexアイテム03 | flexアイテム04 | flexアイテム05 |
| flexアイテム06 | flexアイテム07 | flexアイテム08 | flexアイテム09 | flexアイテム10 |
| flexアイテム11 | flexアイテム12 | | | |
| | | | | |
| align-content-between | | | | |
| flexアイテム01 | flexアイテム02 | flexアイテム03 | flexアイテム04 | flexアイテム05 |
| flexアイテム06 | flexアイテム07 | flexアイテム08 | flexアイテム09 | flexアイテム10 |
| flexアイテム11 | flexアイテム12 | | | |
| | | | | |
| align-content-around | | | | |
| flexアイテム01 | flexアイテム02 | flexアイテム03 | flexアイテム04 | flexアイテム05 |
| flexアイテム06 | flexアイテム07 | flexアイテム08 | flexアイテム09 | flexアイテム10 |
| flexアイテム11 | flexアイテム12 | | | |
| | | | | |
| align-content-stretch | | | | |
| flexアイテム01 | flexアイテム02 | flexアイテム03 | flexアイテム04 | flexアイテム05 |
| flexアイテム06 | flexアイテム07 | flexアイテム08 | flexアイテム09 | flexアイテム10 |
| flexアイテム11 | flexアイテム12 | | | |
| | | | | |

1
2
3
4
5
6
7
8
9
10

8 SECTION

7 Float ユーティリティ

Float ユーティリティを使用すると、要素を包含ブロックの左または右に沿うように配置し、他の要素がその周りを回り込むように設定できます。このユーティリティはブレイクポイントと組み合わせることもできます。

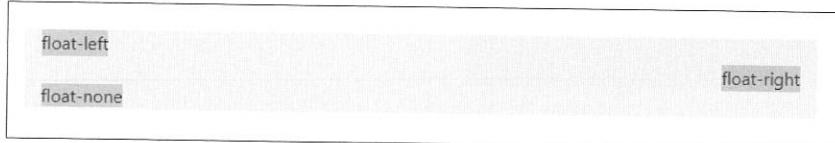
8.7.1 フロートを設定するクラス

Float ユーティリティは、float プロパティを設定できるクラスです。フロートを設定したい要素に **float-{ 配置の種類 } クラス** を設定します。{ 配置の種類 } には、left (左)、right (右)、none (なし) のいずれかを選択します（リスト 8-26、図 8-27）。

▼リスト 8-26 フロートを設定するクラスの例（float.html）

```
<div class="container">
  <span class="float-left">float-left</span><br>
  <span class="float-right">float-right</span><br>
  <span class="float-none">float-none</span>
</div>
```

▼図 8-27 フロートを設定するクラスの例



8.7.2 ブレイクポイントでフロートを切り替えるクラス

各ブレイクポイントに対応したクラス接頭辞 sm、md、lg、xl と組み合わせた **float-{ ブレイクポイント }-{ 配置の種類 } クラス** も使用できます。

次の例では、各ブレイクポイントで要素を右寄せにしています（リスト 8-27、図 8-28）。

▼リスト 8-27 ブレイクポイントでフロートを切り替えるクラスの例（float-responsive.html）

```
<div class="container">
  <span class="float-sm-right">SM (small)以上で右寄せ</span>
</div>
<div class="container">
```

```
<span class="float-md-right">MD (medium)以上で左寄せ</span>
</div>
<div class="container">
  <span class="float-lg-right">LG (large)以上で左寄せ</span>
</div>
<div class="container">
  <span class="float-xl-right">XL (extra-large)以上で左寄せ</span>
</div>
```

▼図 8-28 ブレイクポイントでフロートを切り替えるクラスの例（上：Small 未満、下：Small）

SM(small)以上で左寄せ
 MD (medium)以上で左寄せ
 LG (large)以上で左寄せ
 XL (extra-large)以上で左寄せ

| | |
|------------------------|-----------------|
| MD (medium)以上で左寄せ | SM(small)以上で左寄せ |
| LG (large)以上で左寄せ | |
| XL (extra-large)以上で左寄せ | |

8.7.3 Clearfix ユーティリティ

clearfix ユーティリティは、フロートを指定した要素の親要素に設定し、フロートを解除するのに使用します。

次の例では、子要素に右フロートをかけています。親要素には、背景色クラス bg-danger を指定していますが、中の要素が浮いた（フロートした）状態になっているため、親ボックスの高さが算出されず、背景色が敷かれていません（リスト 8-28、図 8-29）。

▼リスト 8-28 フロートを解除していない例（clearfix-before.html）

```
<div class="w-50 mx-auto bg-danger">
  <div class="float-right bg-success">右フロート</div>
</div>
```

▼図 8-29 フロートを解除していない例

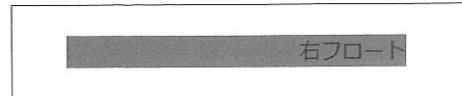
右フロート

その他、マージンが効かない、後続の要素の配置が崩れるといったことも、フロートを解除しなかった場合に起こりがちな問題です。こういったケースでは、親要素に clearfix クラスを指定してフロートを解除します（リスト 8-29、図 8-30）。

▼リスト 8-29 clearfix でフロートを解除した例 (clearfix-after.html)

```
<div class="w-50 mx-auto bg-danger clearfix">
  <div class="float-right bg-success">右フロート</div>
</div>
```

▼図 8-30 clearfix でフロートを解除した例



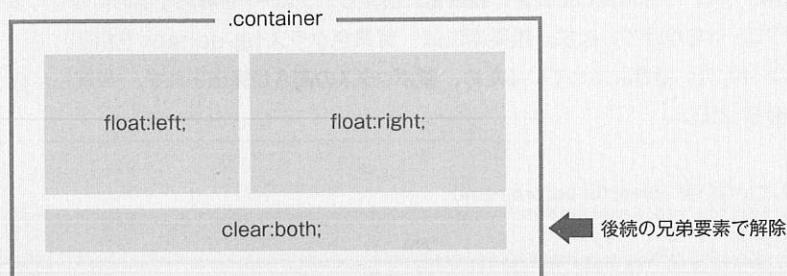
clearfix の使いどころ

次の例のように、フロートをかけた要素の後に、フロートを解除できる兄弟要素があれば clearfix は必要ありません (リスト 8-30、図 8-31)。

▼リスト 8-30 clearfix クラスが不要のケース

```
<div class="container">
  <div class="float-left">左フロート</div>
  <div class="float-right">右フロート</div>
  <div style="clear:both">フロート解除</div> <!-- ここで解除 -->
</div>
```

▼図 8-31 clearfix クラスが不要のケース



次の例のように、後続の兄弟要素がない場合に、親要素に clearfix クラスを指定します (リスト 8-31)。

▼リスト 8-31 clearfix クラスが必要になるケース

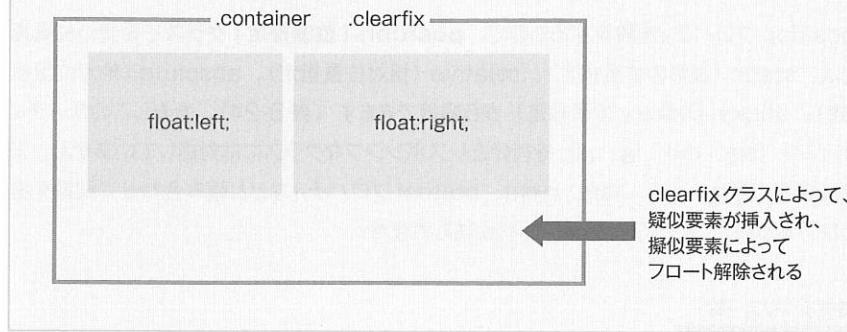
```
<div class="container clearfix">
  <div class="float-left">左フロート</div>
  <div class="float-right">右フロート</div>
</div>
```

clearfix クラスは、after 疑似要素を使ったもので、要素の後ろに指定した内容を追加して、フロートを解除するしくみになっています（リスト 8-32、図 8-32）。

▼リスト 8-32 clearfix クラスに定義されているスタイル

```
.clearfix::after {  
    display: block;  
    content: "";  
    clear: both;  
}
```

▼図 8-32 clearfix クラスが必要になるケース



1
2
3
4
5
6
7
8
9
10

8

SECTION

8

Position ユーティリティ

Position ユーティリティを使用して、相対位置や絶対位置など、要素の位置指定を行うことができます。

8.8.1 要素の位置指定をするクラス

このユーティリティは、position プロパティを設定するもので、**position-{ 位置指定 } クラス**で要素の位置指定を設定します。位置指定は、static (通常の位置指定)、relative (相対位置指定)、absolute (絶対位置指定)、fixed (固定位置指定)、sticky (Sticky 位置指定) から選択できます（表 8-24）。ただしこのユーティリティクラスは、ブレイクポイント (sm、md、lg、xl) を付けたレスポンシブなクラスには対応していません。また実際のところ位置指定を設定する際は、top、left、right、bottom プロパティなどと組み合わせて配置を指定することが多いため、このクラスの使いどころは限られているかもしれません。

▼表 8-24 要素の位置指定をするクラス一覧

| クラス | 説明 |
|-------------------|-------------|
| position-static | 通常の位置指定 |
| position-relative | 相対位置指定 |
| position-absolute | 絶対位置指定 |
| position-fixed | 固定位置指定 |
| position-sticky | Sticky 位置指定 |

NOTE

Sticky とは？

Sticky 位置指定とは、CSS3 で position プロパティに新たに追加された値で、相対位置指定と固定位置指定を組み合わせたものです。指定したしきい値に達するまで相対位置指定として振る舞い、しきい値に達したら固定位置指定として扱われます。

たとえば sticky といいクラス名を付けた要素に、リスト 8-33 のようなスタイルを指定した場合、上端から 100px の位置に到達するまでは、position: relative のように振る舞い、上端から 100px の位置に到達するとそこで固定されます。

▼リスト 8-33 position: sticky; の例

```
.sticky {
  position: sticky;
  top: 10px; /* 上端から100pxのところで固定 */
```

}

ただし、Internet Explorer 10、11など、position: sticky; に未対応のブラウザもあるので注意してください。リスト 8-34 のように @supports クエリを使用すると、サポートしているブラウザにのみ position: sticky; を、未対応ブラウザには position: fixed を適用することができます。

▼リスト 8-34 サポートブラウザにのみ position: sticky; を適用

```
@supports (position: sticky) { /* サポートブラウザ用 */
  .sticky {
    position: sticky;
  }
}
@supports not (position: sticky) { /* 未サポートブラウザ用 */
  .sticky {
    position: fixed;
  }
}
```

8.8.2 最上部に固定するクラス

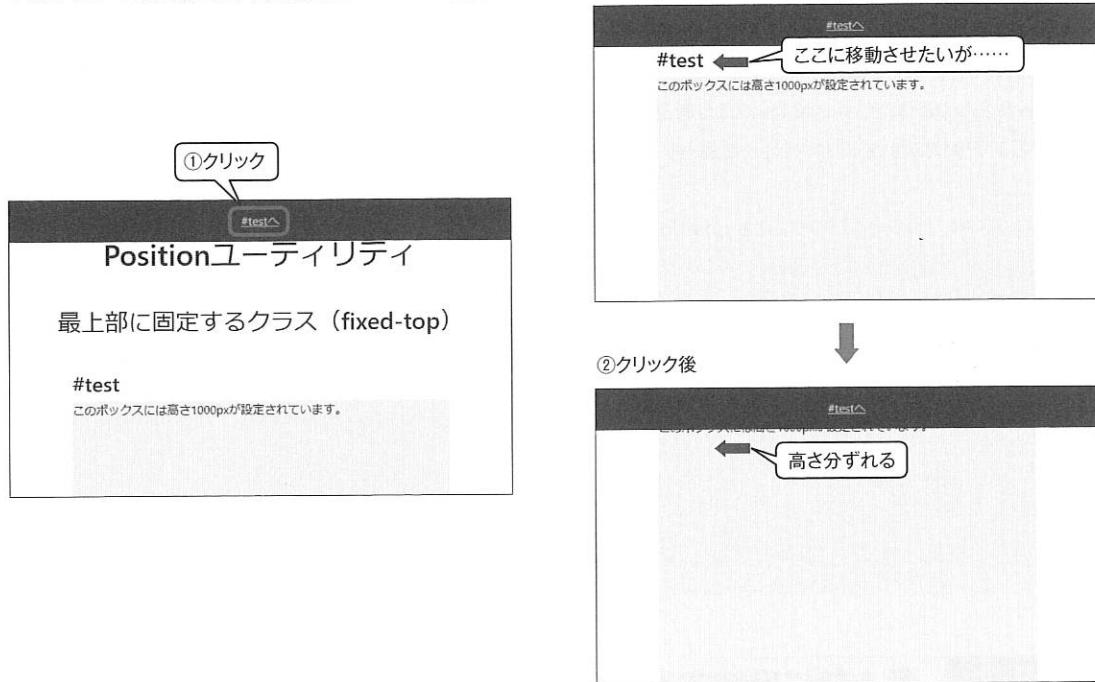
fixed-top クラスを使用すると、要素を画面の最上部に固定できます。たとえば、ナビゲーションバーの固定などで使用します。リスト 8-35 のサンプルでは、fixed-top クラスを指定した要素の前に、h1 要素や h2 要素などがありますが、それらの見出し要素よりも上の最上部に要素が固定配置されます。

▼リスト 8-35 fixed-top クラスの例 (fixed-top.html)

```
<h1 class="my-5 text-center">Positionユーティリティ</h1>
<h2 class="my-5 text-center">最上部に固定するクラス (fixed-top) </h2>
<div class="container">
  <div class="fixed-top py-3 px-3 bg-dark text-center" id="nav"><a href="#test" class="text-light">#testへ</a></div>
  <h3 id="test">#test</h3>
  <div class="box">このボックスには高さ 1000px が設定されています。</div>
</div>
```

ただし、固定位置指定で要素を配置すると、その要素の下に他の要素が隠れてしまう場合があります。サンプルの「#test へ」のアンカーリンクをクリックすると、test という ID 名を付けた h3 要素が、最上部に固定された要素の下に隠れているのを確認できます（図 8-33）。

▼図 8-33 要素を最上部に固定配置したことによる影響



そのようなときには、追加 CSS で調整が必要です。今回の場合は、バーの高さを 56px で指定しているので、対象の要素の上マージンに、その高さ分のネガティブマージン -56px を指定します。このままだと前の要素との間隔が 56px 分詰まってしまいますので、上パディングを追加します（リスト 8-36）。

▼リスト 8-36 固定配置によるズレを調整する追加 CSS (fixed-top2.html)

```
#test {
    margin-top: -56px; /* 固定した要素の高さ分のネガティブマージンを設定 */
    padding-top: 56px; /* 打消し用の上パディング */
}
```

8.8.3 最下部に固定するクラス

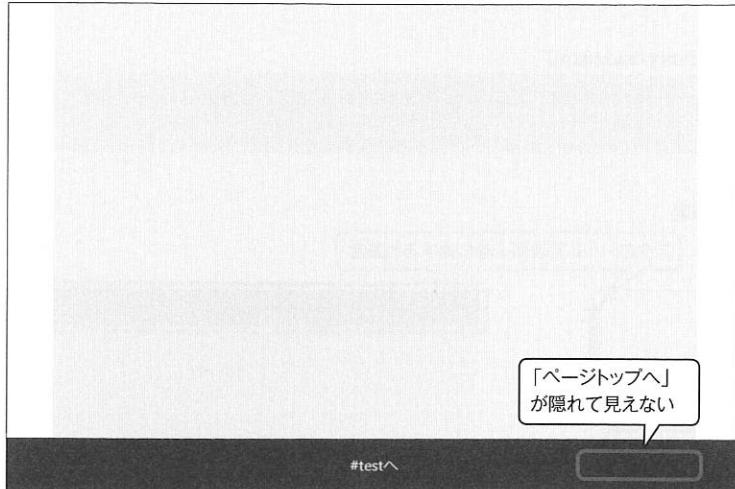
fixed-bottom クラスを使用すると、要素を画面の最下部に固定できます。こちらも前項と同じように、要素を固定位置指定したことによる影響を考慮する必要があります。次の例では、ページトップへのアンカーリンクを画面下部に設置していますが、固定配置した要素の下に隠れてしまっています（リスト 8-37、図 8-34）。

▼リスト 8-37 fixed-bottom クラスの例 (fixed-bottom.html)

```
<div class="fixed-bottom py-3 px-3 bg-dark text-center" id="nav"><a href="#test" class="text-light">#testへ</a></div>
<h3 id="test">#test</h3>
```

```
<div class="box">このボックスには高さ1000pxが設定されています。</div>
<p class="pagetop text-right"><a href="#top">ページトップへ</a></p>
```

▼図 8-34 要素を最下部に固定配置したことによる影響

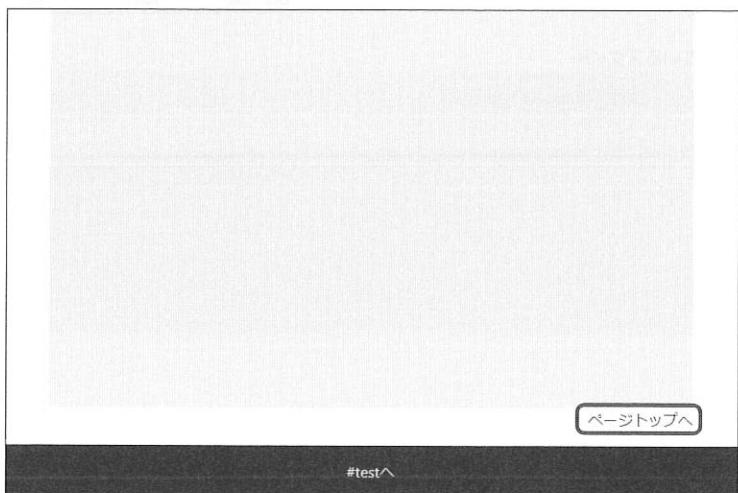


今回は、固定配置した要素の高さ分の下パディングを追加して調整します（リスト 8-38、図 8-35）。

▼リスト 8-38 固定配置によるズレを調整する追加 CSS (fixed-bottom2.html)

```
.pagetop {
    padding-bottom: 56px; /* 固定した要素の高さ分、下パディングを追加 */
}
```

▼図 8-35 追加 CSS で調整後



1
2
3
4
5
6
7
8
9
10

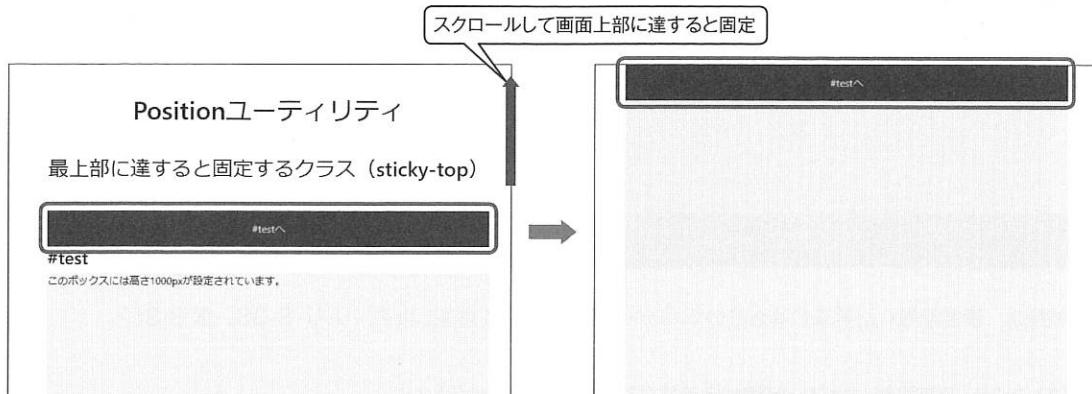
8.8.4 最上部に達すると固定するクラス

sticky-top クラスを使用すると、要素がスクロールして最上部に達すると固定されます。このクラスも、ナビゲーションバーの固定などで使用できます（リスト 8-39、図 8-36）。

▼リスト 8-39 fixed-bottom クラスの例（sticky-top.html）

```
<div class="sticky-top py-3 px-3 bg-dark text-center" id="nav"><a href="#test" class="text-light">#testへ</a></div>
```

▼図 8-36 スクロールして最上部に達すると固定



sticky-top ユーティリティは、CSS の position: sticky を使用していますが、これはすべてのブラウザで完全にサポートされているわけではありません。たとえば、Internet Explorer 10、11 は position: sticky; を position: relative; としてレンダリングします。そのため、Bootstrap では、スタイルを @support クエリで囲み、適切にレンダリングできるブラウザのみに適用する方法を採用しています（リスト 8-40）。

▼リスト 8-40 sticky-top クラスに定義されているスタイル

```
@supports ((position: -webkit-sticky) or (position: sticky)) {
  .sticky-top {
    position: -webkit-sticky;
    position: sticky;
    top: 0;
    z-index: 1020;
  }
}
```

8

9

Text ユーティリティ

Text ユーティリティを使用して、文字の整列、折り返し、太さなどを設定できます。

8.9.1 文字の均等割り付けを設定するクラス

文字を均等割り付けするには、**text-justify** クラスを指定します（リスト 8-41、図 8-37）。

▼リスト 8-41 文字の均等割り付けの例（text-justify.html）

```
<p class="text-justify">Bootstrapは、HTML/CSS/JavaScriptから構成されるもっとも有名なWebアプリケーションフレームワークの1つです。Bootstrapは、HTML/CSS/JavaScriptから構成されるもっとも有名なWebアプリケーションフレームワークの1つです。・・・略</p>
```

▼図 8-37 text-justify 設定前（左）と設定後（右）

Bootstrapは、HTML/CSS/JavaScriptから構成される最も有名なWebアプリケーションフレームワークのひとつです。Bootstrapは、HTML/CSS/JavaScriptから構成される最も有名なWebアプリケーションフレームワークのひとつです。Bootstrapは、HTML/CSS/JavaScriptから構成される最も有名なWebアプリケーションフレームワークのひとつです。Bootstrapは、HTML/CSS/JavaScriptから構成される最も有名なWebアプリケーションフレームワークのひとつです。Bootstrapは、HTML/CSS/JavaScriptから構成される最も有名なWebアプリケーションフレームワークのひとつです。

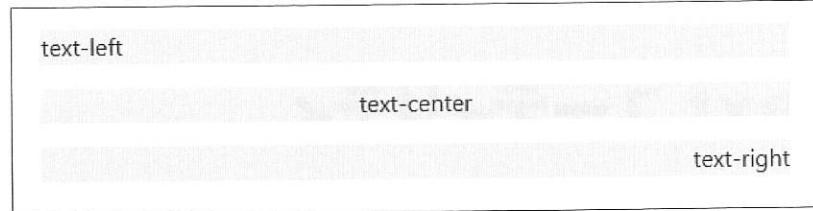
8.9.2 文字の左寄せ／右寄せ／中央揃えを設定するクラス

文字を左寄せにするには **text-left** クラス、右寄せにするには **text-right** クラス、中央寄せにするには **text-center** クラスを使用します（リスト 8-42、図 8-38）。

▼リスト 8-42 文字の左寄せ／中央揃え／右寄せを設定するクラスの例（text-align.html）

```
<p class="text-left">text-left</p>
<p class="text-center">text-center</p>
<p class="text-right">text-right</p>
```

▼図 8-38 字の左寄せ／中央揃え／右寄せを設定するクラスの例



また、text-sm-right のように、ブレイクポイント sm、md、lg、xl を組み合わせたレスポンシブなクラスも利用可能です（表 8-25）。

▼表 8-25 字の左寄せ／中央揃え／右寄せを設定するクラス一覧

| クラス | 説明 |
|----------------|----------------|
| text-left | すべての画面サイズで左寄せ |
| text-sm-left | sm 以上で左寄せ |
| text-lg-left | lg 以上で左寄せ |
| text-xl-left | xl 以上で左寄せ |
| text-right | すべての画面サイズで右寄せ |
| text-sm-right | sm 以上で右寄せ |
| text-lg-right | lg 以上で右寄せ |
| text-xl-right | xl 以上で右寄せ |
| text-center | すべての画面サイズで中央揃え |
| text-sm-center | sm 以上で中央揃え |
| text-lg-center | lg 以上で中央揃え |
| text-xl-center | xl 以上で中央揃え |

8.9.3 文字を折り返さないよう設定するクラス

text nowrap クラスを使用すると、テキストが折り返されないように設定することができます。次の例では、幅の狭いテーブルの見出しセルに、text nowrap クラスを指定しています。この例のようにテーブルのセルで使用されることが多いクラスです（リスト 8-43、図 8-39）。

▼リスト 8-43 文字を折り返さないよう設定するクラスの例（text nowrap.html）

```
<table class="w-50 mx-auto">
  <tr>
    <th class="text nowrap">おりかえしたくない</th><td>折り返しても良い</td>
  </tr>
</table>
```

▼図 8-39 text nowrap クラス指定前（左）と指定後（右）

| | |
|-------------------------|------------------------|
| おりかえしたく 折り返しても ない 良い | おりかえしたくない 折り返し ても良い |
|-------------------------|------------------------|

8.9.4 長いテキストを省略記号で表すクラス

設定された幅より長いテキストを丸めて省略記号で表すには、**text-truncate クラス**を使用します。丸められる要素の表示形式はインラインブロック表示（display:inline-block;）またはブロック表示（display: inline-block）である必要があります（リスト 8-44、図 8-40）。

▼リスト 8-44 テキストを省略記号で表すクラス（text-truncate.html）

```
<p class="text-truncate w-25">Bootstrapは、HTML/CSS/JavaScriptから構成されるもっとも有名なWebアプリケーションフレームワークの1つです。</p>
```

▼図 8-40 テキストを省略記号で表すクラス

| |
|-----------------------------------|
| Bootstrapは、HTML/CSS/JavaScript... |
|-----------------------------------|

8.9.5 文字を大文字や小文字に変換するクラス

文字を大文字や小文字に変換するには、**text-{ 文字変換の種類 } クラス**を使用します。{ 文字変換の種類 } は、lowercase（すべて小文字）、uppercase（すべて大文字）、capitalize（最初の文字が大文字）から選択します（表 8-26）。

▼表 8-26 文字を大文字や小文字に変換するクラス一覧

| クラス | 説明 |
|-----------------|-----------|
| text-lowercase | すべて小文字 |
| text-uppercase | すべて大文字 |
| text-capitalize | 最初の文字を大文字 |

text-capitalize クラスは、最初の文字だけを変換し、他の文字には影響しません。サンプルでどのように変換されるか確認してください（リスト 8-45、図 8-41）。

▼リスト 8-45 文字を大文字や小文字に変換するクラス（text-transform.html）

```
<p class="text-lowercase">Text-Lowercase（すべて小文字）</p>
<p class="text-uppercase">text-Uppercase（すべて大文字）</p>
<p class="text-capitalize">text-Capitalize（最初の文字を大文字）</p>
```

▼図 8-41 文字を大文字や小文字に変換するクラス

text-lowercase (すべて小文字)
 TEXT-UPPERCASE (すべて大文字)
 Text-Capitalize (最初の文字を大文字)

8.9.6 文字の太さとイタリック体を設定するクラス

文字の太さを設定するには、**font-weight-{ 文字の太さ }** クラスを使用します。{ 文字の太さ } は、bold (太字)、normal (通常)、light (細字) から選択します。また、文字を斜体にするには **font-italic** クラスを指定します (表 8-27)。

▼表 8-27 文字の太さとイタリック体を設定するクラス一覧

| クラス | 説明 |
|--------------------|--------------|
| font-weight-bold | 文字を太字にする |
| font-weight-normal | 文字を通常にする |
| font-weight-light | 文字を細字にする |
| font-italic | 文字をイタリック体にする |

次のコードは、文字の太さやイタリック体を設定するユーティリティクラスを使用した例です (リスト 8-46、図 8-42)。

▼リスト 8-46 文字の太さとイタリック体を設定するクラス (text-font.html)

```
<p class="font-weight-bold">太字 (font-italic) </p>
<p class="font-weight-normal">通常 (font-weight-normal) </p>
<p class="font-weight-light">細字 (font-weight-light) </p>
<p class="font-italic">イタリック体 (font-italic) </p>
```

▼図 8-42 テキストを省略記号で表すクラス

太字 (font-italic)
 通常 (font-weight-normal)
 細字 (font-weight-light)
 イタリック体 (font-italic)

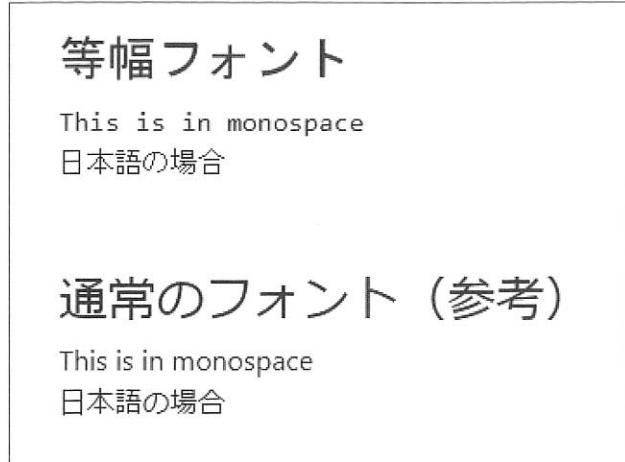
8.9.7 等幅フォントを指定するクラス 4.1

テキストを等幅フォントに変更するには **text-monospace** クラスを指定します（リスト 8-47、図 8-43）。

▼リスト 8-47 等幅フォントを指定するクラス（text-monospace.html）

```
<div class="container mb-5">
  <h3 class="text-monospace">等幅フォント</h3>
  <p class="text-monospace">This is in monospace<br>日本語の場合</p>
</div>
<div class="container">
  <h3>通常のフォント（参考）</h3>
  <p>This is in monospace<br>日本語の場合</p>
</div>
```

▼図 8-43 等幅フォントを指定する



1
2
3
4
5
6
7
8
9
10

Vertical align ユーティリティ

Vertical align ユーティリティを使えば、インラインレベルおよびテーブルのセル要素の垂直方向の配置を簡単に変更することができます。プロックレベル要素には使用できない点に注意してください。使用できるクラスは表 8-28 のとおりです。

▼表 8-28 垂直方向の整列クラス一覧

| クラス | 説明 | スタイル |
|-------------------|-----------|------------------------------|
| align-baseline | ベースライン揃え | vertical-align: baseline; |
| align-top | 上端揃え | vertical-align: top; |
| align-middle | 中央揃え | vertical-align: middle; |
| align-bottom | 下端揃え | vertical-align: bottom; |
| align-text-top | テキストの上端揃え | vertical-align: text-top; |
| align-text-bottom | テキストの下端揃え | vertical-align: text-bottom; |

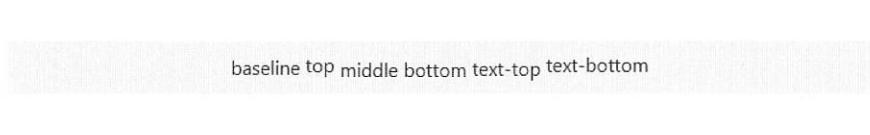
8.10.1 インライン要素の垂直方向の整列

次の例では、インライン要素に Vertical align ユーティリティを設定しています。サンプルではわかりやすいように親要素にボーダーや高さを設定しています（リスト 8-48、図 8-44）。

▼リスト 8-48 インライン要素の垂直方向の整列の例（vertical-align-inline.html）

```
<div class="box">
  <span class="align-baseline">baseline</span><!-- ベースライン -->
  <span class="align-top">top</span><!-- 上揃え -->
  <span class="align-middle">middle</span><!-- 中央揃え -->
  <span class="align-bottom">bottom</span><!-- 下揃え -->
  <span class="align-text-top">text-top</span><!-- テキストの上揃え -->
  <span class="align-text-bottom">text-bottom</span><!-- テキストの下揃え -->
</div>
```

▼図 8-44 インライン要素の垂直方向の整列の例



baseline top middle bottom text-top text-bottom

8.10.2 テーブルセルの垂直方向の整列

次の例では、インライン要素に Vertical align ユーティリティを設定しています。サンプルではわかりやすいように親要素にボーダーや高さを設定しています（リスト 8-49、図 8-45）。

▼リスト 8-49 テーブルセルの垂直方向の整列の例（vertical-align-table.html）

```
<table style="height: 100px;" class="mx-auto">
  <tbody>
    <tr>
      <td class="align-baseline">baseline</td><!-- ベースライン -->
      <td class="align-top">top</td><!-- 上揃え -->
      <td class="align-middle">middle</td><!-- 上揃え -->
      <td class="align-bottom">bottom</td><!-- 上揃え -->
      <td class="align-text-top">text-top</td><!-- テキストの上揃え -->
      <td class="align-text-bottom">text-bottom</td><!-- テキストの下揃え -->
    </tr>
  </tbody>
</table>
```

▼図 8-45 テーブルセルの垂直方向の整列の例

| | | | | | |
|----------|-----|--------|--|----------|-------------|
| baseline | top | middle | | text-top | text-bottom |
| | | bottom | | | |

水平方向の配置を設定したい場合は Float ユーティリティ（P.338 参照）を、flexbox の整列については、P.325 を参照してください。

8

SECTION

11

その他のユーティリティクラス

前節までに説明したユーティリティクラスは、どのようなサイトにも利用できる使用頻度の高いものですが、それ以外にも、Bootstrapにはさまざまなクラスがユーティリティとしてまとめられています。本節では、一般的ではないかもしれません、限定的に使用される他のユーティリティクラスを紹介します。

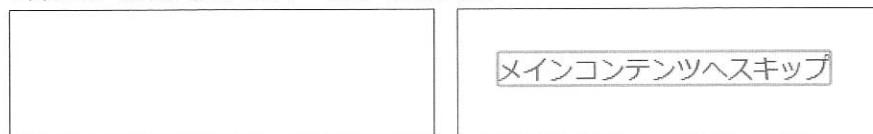
8.11.1 スクリーンリーダー用ユーティリティ

sr-only クラスを使用すると、スクリーンリーダーを除くすべてのデバイスにおいて、要素を非表示にします。また、**sr-only** クラスと **sr-only-focusable クラス**を組み合わせると、たとえばキーボード利用のみのユーザーなどに対し、フォーカス時に要素を再表示させることができます。**sr-only** クラスは、アクセシビリティのための設定として、よく利用されるクラスです（リスト 8-50、図 8-46）。

▼リスト 8-50 スクリーンリーダー用クラスの例（sr-only.html）

```
<a class="sr-only sr-only-focusable" href="#content">メインコンテンツへスキップ</a>
```

▼図 8-46 通常時（左）、tab キーでフォーカス時（右）



8.11.2 Visibility ユーティリティ

Visibility ユーティリティは、要素の表示形式（display プロパティ）を変更せずに、可視性（visibility）だけを切り替える場合に使います。このユーティリティクラスには **visible**（visibility:visible）と **invisible**（visibility:hidden）が定義されています。要素自体の表示を消す（display:none）のではなく、見た目上で要素を隠す（visibility:hidden）だけなので、スクリーンリーダーには表示されます（表 8-29、リスト 8-51、図 8-47）。

▼表 8-29 Visibility ユーティリティ

| クラス | 説明 | スタイル |
|-----------|-----------|----------------------|
| visible | 要素を表示する | visibility: visible; |
| invisible | 要素を非表示にする | visibility: visible; |

▼リスト 8-51 invisible と d-none クラスの違い (visibility.html)

```
<ol>
  <li>リストその1</li>
  <li class="invisible">リストその2</li><!-- ボックスを保ったまま非表示 -->
  <li>リストその3</li>
  <li class="d-none">リストその4</li><!-- ボックスごと非表示 -->
  <li>リストその5</li>
</ol>
```

▼図 8-47 invisible と d-none クラスの違い

1. リストその1
3. リストその3
4. リストその5

8.11.3 クローズアイコンユーティリティ

Bootstrap では、モーダルやアラートなどのコンテンツを閉じる「クローズアイコン」を次のようなコードで実装できます。button 要素に **close クラス** を追加します。アイコンとなる「×」は、乗算記号「×」の文字実体参照です。span 要素で囲んで button 要素の中に配置します。アクセシビリティの設定として、button 要素に aria-label 属性でアイコンボタンのラベルを付けます。span 要素には aria-hidden="true" を指定し、スクリーンリーダーなどの読み上げをスキップするよう指定します。例の場合は「閉じる」が読み上げられ「×」は読み上げられません（リスト 8-54、図 8-49）。このクラスはクローズアイコン表示を実装するもので閉じる機能はありません。これを機能させるにはアラート（P.113）やモーダル（P.283）を参照してください。

▼リスト 8-54 クローズアイコンの例 (close.html)

```
<button type="button" class="close" aria-label="閉じる">
  <span aria-hidden="true">&times;</span>
</button>
```

▼図 8-49 クローズアイコンの例



8.11.4 Embed ユーティリティ

Embed ユーティリティを使用して、レスポンシブなビデオやスライドショーなどをアスペクト比を保ったまま設定できます。子要素として配置できるのは、`iframe`、`embed`、`video`、`object`要素のいずれかです。

埋め込むコンテンツを囲み、**embed-responsive クラス**、およびアスペクト比を持つ**embed-responsive-{ アスペクト比 } クラス**を指定します。`{ アスペクト比 }`は、`16by9` (`16:9`) のように指定します。指定できるアスペクト比は表 8-30 のとおりです。

▼表 8-30 Embed ユーティリティで設定できるアスペクト比

| クラス | アスペクト比 |
|-------------------------------------|-------------------|
| <code>embed-responsive-21by9</code> | <code>21:9</code> |
| <code>embed-responsive-16by9</code> | <code>16:9</code> |
| <code>embed-responsive-4by3</code> | <code>4:3</code> |
| <code>embed-responsive-16by1</code> | <code>1:1</code> |

例では、`16:9` のアスペクト比を持つ動画を埋め込んでいます。フルスクリーン表示に対応するために `iframe` 要素に `allowfullscreen` を指定しています（この指定がない場合、YouTube では「全画面表示はご利用いただけません」という表示が出ます）。子要素には、**embed-responsive-item クラス**を指定しています。これは厳密には必須ではありませんが、子要素を明示的にするクラスとして推奨されています（リスト 8-55、図 8-50）。

▼リスト 8-55 埋め込みクラスの例（`embed.html`）

```
<div class="embed-responsive embed-responsive-16by9">
  <iframe class="embed-responsive-item" src="https://www.youtube.com/embed/zp0ULjyy-n8?rel=0" allowfullscreen></iframe>
</div>
```

▼図 8-50 埋め込みクラスの例



8.11.5 Shadows ユーティリティ 4.1.0

ボックスに影を追加または削除するには、Shadows ユーティリティを使用します。影のサイズは、shadow（標準）、shadow-sm（小さめ）、shadow-lg（大きめ）から選択できます。また、影を消したい時は、shadow-none クラスを指定します。

▼表 8-31 ボックスの影を設定するクラス一覧

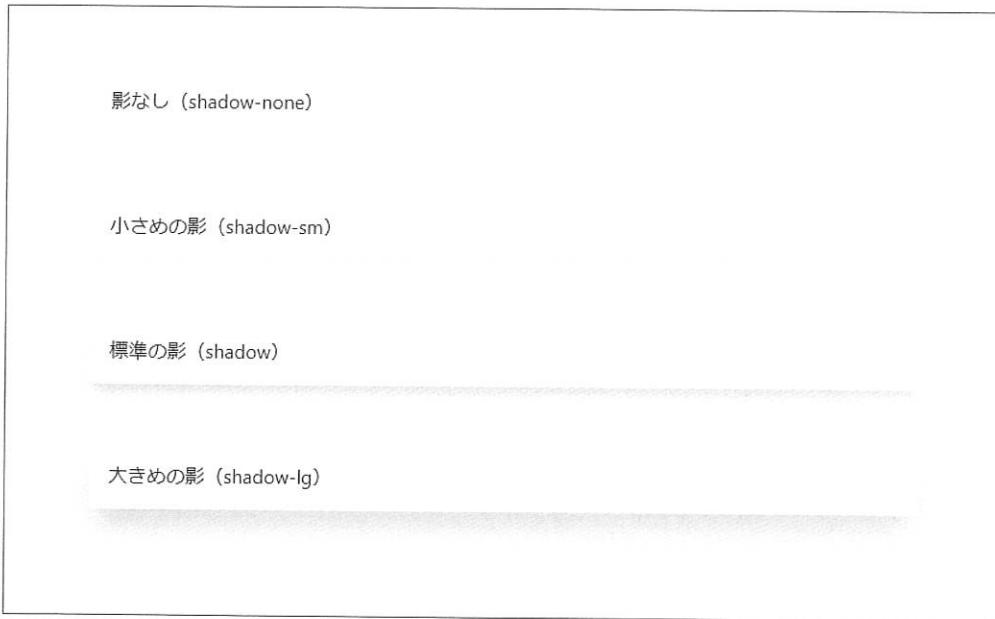
| クラス | 説明 |
|-------------|-------|
| shadow-none | 影なし |
| shadow-sm | 小さめの影 |
| shadow | 標準の影 |
| shadow-lg | 大きめの影 |

次のコードは、Shadows ユーティリティを使用してボックスに影を設定した例です。

▼リスト 8-56 Shadows ユーティリティの例 (shadow.html)

```
<div class="shadow-none p-3 mb-5 bg-light rounded">影なし (shadow-none) </div>
<div class="shadow-sm p-3 mb-5 bg-light rounded">小さめの影 (shadow-sm) </div>
<div class="shadow p-3 mb-5 bg-light rounded">標準の影 (shadow) </div>
<div class="shadow-lg p-3 mb-5 bg-light rounded">大きめの影 (shadow-lg) </div>
```

▼図 8-51 Shadows ユーティリティの例



 NOTE

コンポーネントの影を有効にする

Bootstrapでは、コンポーネントの影はデフォルトでは無効になっています。Shadows ユーティリティを使用して影を追加または削除する他に、Sass変数の\$enable-shadowsを使って有効にすることも可能です（P.447参照）。