

# 第3章

## 基本的な スタイリング

Bootstrap では、タイポグラフィ、コード表示、画像、テーブル、図表といった要素について、マージンサイズやパディングサイズ、フォントなどに関する基本的なスタイルがあらかじめ定義され、クラスとして用意されています。これらの定義済みクラスを選択し、要素に追加していくことで、Bootstrap で作成するサイトデザインの全体的な統一を図ります。またこれらのスタイルは、Bootstrap 特有の `Reboot` と呼ばれるリセットスタイルを基礎として構築されているのも特徴です。

この章では、Bootstrap における基本的な要素のスタイリングの特徴や方法を説明していきます。

## 3

SECTION

## 1

## タイポグラフィ

Bootstrapには、基本的な表示形式、タイポグラフィ、およびリンクスタイルが設定されています。この節では、Bootstrapにおけるタイポグラフィ（グローバル設定、見出し、本文、リストなど）の使用方法を解説します。

## 3.1.1 見出し

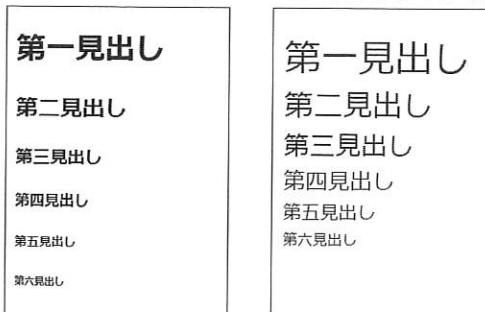
すべての見出し要素 `<h1> ~ <h6>` に Bootstrap の基本的なスタイルが定義されています。また、定義済みクラスとして **h1 クラス**～**h6 クラス** が用意されており、見出しではない文字要素に見出しと同じスタイルを適用することができます。

まずは見出し要素に Bootstrap のスタイルを適用しない場合と適用した場合とを見比べてみましょう。Bootstrap 独自のスタイルの方が、ブラウザのデフォルトスタイルよりも見やすく整っていることがわかります（リスト 3-1、図 3-1）。

▼リスト 3-1 見出し `<h1> ~ <h6>` (`typography-heading.html`)

```
<h1>第一見出し</h1>
<h2>第二見出し</h2>
<h3>第三見出し</h3>
<h4>第四見出し</h4>
<h5>第五見出し</h5>
<h6>第六見出し</h6>
```

▼図 3-1 Bootstrap のスタイル適用前（左）、スタイル適用後（右）



Bootstrapでは見出し要素に、`h1～h6`要素および**h1 クラス**～**h6 クラス**のマージンサイズ（リスト 3-2 ①）や、見出しレベルごとの文字サイズが定義されています（②）。

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

## ▼リスト3-2 見出し要素 h1 ~ h6 に定義されているスタイル

```

h1, h2, h3, h4, h5, h6 {
    margin-top: 0; _____
    margin-bottom: 0.5rem; _____
}
…中略…
h1, h2, h3, h4, h5, h6,
.h1, .h2, .h3, .h4, .h5, .h6 {
    margin-bottom: 0.5rem; _____
    font-family: inherit;
    font-weight: 500;
    line-height: 1.2;
    color: inherit;
}

h1, .h1 { _____
    font-size: 2.5rem;
}
h2, .h2 { _____
    font-size: 2rem;
}
h3, .h3 { _____
    font-size: 1.75rem;
}
h4, .h4 { _____
    font-size: 1.5rem;
}
h5, .h5 { _____
    font-size: 1.25rem;
}
h6, .h6 { _____
    font-size: 1rem;
}

```

次に、**h1 クラス～h6 クラス**を使用してみます（リスト3-3、図3-2）。

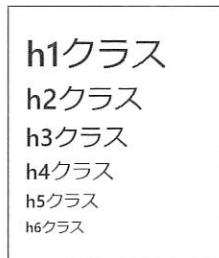
## ▼リスト3-3 h1 クラス～h6 クラス (typography-heading-class.html)

```

<p class="h1">h1クラス</p>
<p class="h2">h2クラス</p>
<p class="h3">h3クラス</p>
<p class="h4">h4クラス</p>
<p class="h5">h5クラス</p>
<p class="h6">h6クラス</p>

```

▼図3-2 h1クラス～h6クラス



以上のように見出し以外の文字要素（違うレベルの見出し要素も含む）にも、見出しと同じスタイルを適用することができます。

### 3.1.2 見出しに副見出しを付ける

Textユーティリティ（P.347参照）の **text-muted** クラスを使用して、小さい副見出しを作成できます。タイトルとなる文にサブタイトル文を添えるような場合に使用できます（リスト3-4、図3-3）。

▼リスト3-4 text-mutedクラスで副見出しを作成（typography-text-muted.html）

```
<h3>
  主見出し<small class="text-muted">副見出し (small.text-muted) </small>
</h3>
```

▼図3-3 text-mutedクラスで副見出しを作成

### 3.1.3 見出しを目立たせる

通常の見出しスタイルよりもさらに見出しを目立たせる必要がある場合は、**display-{ サイズ }** クラスを使用して、テキストを大きく表示することができます。{ サイズ }には、1～4の数値を設定でき、display-1がもっとも大きいサイズになります（リスト3-5、図3-4）。

▼リスト3-5 display-{ サイズ }クラスで見出しを目立たせる（typography-display.html）

```
<h1 class="display-1">h1.display-1</h1>
<h1 class="display-2">h1.display-2</h1>
<h1 class="display-3">h1.display-3</h1>
<h1 class="display-4">h1.display-4</h1>
<h1>h1</h1>
```

▼図 3-4 `display-{ サイズ }` クラス

```
h1.display-1  
h1.display-2  
h1.display-3  
h1.display-4  
h1 (標準)
```

`display-{ サイズ }` クラスには、フォントサイズ、フォントウェイト（文字の太さ）、行高さが 4 段階で定義されています（リスト 3-6）。

▼リスト 3-6 `display-{ サイズ }` クラスに定義されているスタイル

```
.display-1 {  
    font-size: 6rem;  
    font-weight: 300;  
    line-height: 1.2;  
}  
.display-2 {  
    font-size: 5.5rem;  
    font-weight: 300;  
    line-height: 1.2;  
}  
.display-3 {  
    font-size: 4.5rem;  
    font-weight: 300;  
    line-height: 1.2;  
}  
.display-4 {  
    font-size: 3.5rem;  
    font-weight: 300;  
    line-height: 1.2;  
}
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

### 3.1.4 リード

見出しへなく通常の段落内のテキストを目立たせる必要がある場合は、**lead クラス**を使用して、テキストを大きく表示することができます（リスト3-7、図3-5）。

▼リスト3-7 leadクラスでリード文を作成（typography-lead.html）

```
<p class="lead">目立たせたい段落（p.lead）</p>
<p>標準的な段落（p）</p>
```

▼図3-5 leadクラスでリード文を作成

目立たせたい段落（p.lead）

標準的な段落（p）

なお**lead**クラスには、**フォントサイズ 1.25rem、フォントウェイト 300**が定義されています（リスト3-8）。

▼リスト3-8 leadクラスに定義されているスタイル

```
.lead {
  font-size: 1.25rem;
  font-weight: 300;
}
```

NOTE

#### フォントウェイトについて

**font-weight**は、フォントのウェイト（太さ）を指定するためのCSSプロパティで、表3-1の値が使用できます。実際に表示されるフォントのウェイトは使用しているfont-familyに依存し、normalおよびboldしか使用できないフォントもあります。

▼表3-1 font-weightプロパティで設定できる値

値	説明
normal	通常のフォントウェイト。400に相当
bold	太字のフォントウェイト。700に相当
lighter	フォントウェイトを継承値より一段階細く設定
bolder	フォントウェイトを継承値より一段階太く設定
100, 200, 300, 400, 500, 600, 700, 800, 900	normalとbold以上のものを提供するフォントに対し、より細かく設定できる数値のフォントウェイト

一般的なウェイト名と数値の対応表は表3-2のとおりです。leadクラスは、font-weight:300なので、normalより少し細めのフォントウェイトになることが想定されています。

▼表 3-2 一般的なウェイト名との対応

値	一般的なウェイト名
100	Thin (Hairline)
200	Extra Light (Ultra Light)
300	Light
400	Normal
500	Medium
600	Semi Bold (Demi Bold)
700	Bold
800	Extra Bold (Ultra Bold)
900	Black (Heavy)

### 3.1.5 インラインテキスト要素

この項では、インラインのテキスト要素のスタイルに用意されているクラスを見ていきましょう。

#### ■ small 要素と small クラス

**small 要素**を使って、テキストを細目・注釈を表すテキストとしてマークアップし、小さなサイズで表示することができます。また、定義済みクラスとして **small クラス**が用意されており、細目・注釈以外の文字要素に small 要素と同じスタイルを適用することができます（リスト 3-9、図 3-6）。

▼リスト 3-9 small 要素および small クラス (typography-small.html)

```
<p><small>「small要素」</small>を使って、テキストを細目・注釈を表すテキストとして小さなサイズで表示させることができます。</small></p>
<p><span class="small">「smallクラス」</span>を使用してsmall要素とスタイルを一致させることもできます。</p>
```

▼図 3-6 small 要素および small クラス

「small要素」を使って、テキストを細目・注釈を表すテキストとして小さなサイズで表示させることができます。

「smallクラス」を使用してsmall要素とスタイルを一致させることもできます。

なお Bootstrap における small 要素および **small クラス**には、フォントサイズが 80 %、フォントウェイトは標準値が定義されています（リスト 3-10）。

▼リスト 3-10 small 要素および small クラスに定義されているスタイル

```
small, .small {
  font-size: 80%;
```

```
font-weight: 400;
}
```

## ■ strong 要素と b 要素

**strong** 要素を使って、テキストを**重要なテキスト**としてマークアップし、太字で表示することができます。また、**b** 要素を使ってテキストを**強調表示するテキスト**としてマークアップし、太字で表示することができます（リスト3-11、図3-7）。

### ▼リスト3-11 strong要素とb要素 (typography-strong.html)

```
<p>strong要素を使って、テキストを<strong>重要なテキスト</strong>として太字で表示することができます。</p>
<p>b要素を使ってテキストを<b>強調表示するテキスト</b>として太字で表示することができます。</p>
```

### ▼図3-7 strong要素とb要素

strong要素を使って、テキストを**重要なテキスト**として太字で表示させることができます。

b要素を使ってテキストを**強調表示するテキスト**として太字で表示させることができます。

なお Bootstrap における strong 要素および b 要素には、フォントウェイトが継承した値より 1 段階太く (+100) 定義されています（リスト3-12①）。

### ▼リスト3-12 strong要素およびb要素に定義されているスタイル

```
b,
strong {
  font-weight: bolder; ①
}
```

## ■ mark要素

**mark** 要素を使って、テキストをハイライト表示することができます。また、定義済みクラスとして**mark クラス**が用意されており、mark 要素以外の文字要素に mark 要素と同じスタイルを適用することができます（リスト3-13、図3-8）。

### ▼リスト3-13 mark要素およびmarkクラス (typography-mark.html)

```
<p>mark要素を使って、テキストを<mark>ハイライト表示</mark>することができます。</p>
<p>定義済みクラス「mark」を使用して<span class="mark">mark要素とスタイルを一致</span>させることもできます。</p>
```

▼図 3-8 mark 要素および mark クラス

mark要素を使って、テキストをハイライト表示させることができます。  
定義済みクラス「.mark」を使用して mark要素とスタイルを一致させることもできます。

なお Bootstrap における mark 要素および **mark クラス**には、リスト 3-14 のようなスタイルでパディングサイズや背景色が定義されています（リスト 3-14）。

▼リスト 3-14 mark 要素および mark クラスに定義されているスタイル

```
mark,
.mark {
  padding: 0.2em;
  background-color: #fcf8e3;
}
```

## ■ del 要素

**del 要素**のデフォルトスタイルを使って、**削除されたテキスト**としてマークアップし、取り消し線を表示することができます（リスト 3-15、図 3-9）。

▼リスト 3-15 del 要素 (typography-del.html)

```
<p>del要素を使って、テキストを<del>削除されたテキスト</del>として取り消し線を表示することができます。</p>
```

▼図 3-9 del 要素

del要素のデフォルトスタイルを使って、テキストを削除されたテキストとして取り消し線を表示させることができます。

## ■ s 要素

**s 要素**のデフォルトスタイルを使って、テキストを**無効なテキスト**としてマークアップし、取り消し線を表示することができます（リスト 3-16、図 3-10）。

▼リスト 3-16 s 要素 (typography-s.html)

```
<p>s要素のデフォルトスタイルを使って、テキストを<s>無効なテキスト</s>として取り消し線を表示することができます。</p>
```

▼図 3-10 s 要素

s要素のデフォルトスタイルを使って、テキストを無効なテキストとして取り消し線を表示させることができます。

## ■ ins 要素

ins要素のデフォルトスタイルを使って、テキストを後から挿入されたテキストとしてマークアップし、下線を表示することができます（リスト3-17、図3-11）。

▼リスト3-17 ins要素（typography-ins.html）

```
<p>ins要素のデフォルトスタイルを使って、テキストを<ins>後から挿入されたテキスト</ins>として下線を表示することができます。</p>
```

▼図3-11 ins要素

ins要素のデフォルトスタイルを使って、テキストを後から挿入されたテキストとして下線を表示させることができます。

## ■ u 要素

u要素のデフォルトスタイルを使って、テキストをラベル付けされたテキストとしてマークアップし、下線を表示することができます（リスト3-18、図3-12）。

▼リスト3-18 u要素（typography-u.html）

```
<p>u要素のデフォルトスタイルを使って、テキストを<u>ラベル付けされたテキスト</u>として下線を表示することができます。</p>
```

▼図3-12 u要素の使用例

u要素のデフォルトスタイルを使って、テキストをラベル付けされたテキストとして下線を表示させることができます。

## em 要素

**em 要素**を使って、テキストを強調したいテキストとしてマークアップし、イタリック体または斜体で表示することができます。em 要素も Bootstrap 独自のスタイルではなく、ブラウザのデフォルトスタイルとなっているため、ブラウザによっては 123 や abc など英数字のみに適用されます（リスト 3-18、図 3-13）。

### ▼リスト 3-19 em 要素 (typography-em.html)

```
<p><em>em要素のデフォルトスタイルを使って、テキストを<em>強調したいテキスト (Emphasis text) </em>として</p>
イタリック体または斜体で表示することができます。（ブラウザによっては123やabcなど英数字のみ適用）</p>
```

### ▼図 3-13 em 要素

em要素のデフォルトスタイルを使って、テキストを強調したいテキスト (Emphasis text) としてイタリック体または斜体で表示させることができます。（ブラウザによっては123やabcなど英数字のみ適用）

## 3.1.6 Text ユーティリティ

テキストの整列、折り返し、太さなど、タイポグラフィに関する詳細な設定変更が必要な場合は、Text ユーティリティ（P.347 参照）の **text-{ プロパティ } クラス**、**font-{ プロパティ } クラス** を使用します。

## 3.1.7 略語

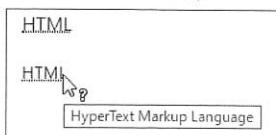
略称、略語を表す **abbr 要素**には、マウスポインターを合わせると説明文が展開表示されます。略語には、デフォルト設定として下線が付いており、マウスポインターを合わせると、追加の文脈表示や、アシスティブテクノロジー（障害を持つ人々を支援するための技術）のユーザーのためのヘルプカーソルが得られます（リスト 3-20、図 3-14）。

また、**initialism クラス**を追加して、少し小さいフォントサイズの略語を表現できます。

### ▼リスト 3-20 abbr 要素 (typography-abbr.html)

```
<p><abbr title="HyperText Markup Language">HTML</abbr></p>
<p><abbr title="HyperText Markup Language" class="initialism">HTML</abbr></p>
```

### ▼図 3-14 abbr 要素



なおBootstrapにおけるabbr要素には、アンダーラインやマウスポインターの種類、下ボーダーの削除などが定義されています（リスト3-21）。また、Webkit系ブラウザ（Chrome、Edge、Internet Explorer、Opera、Safari）で正しく下線装飾されるようにスタイルを追加し（①）、ヘルプカーソルの表示（②）を設定しています。

▼リスト3-21 abbr要素に定義されているスタイル

```
abbr[title],  
abbr[data-original-title] {  
    text-decoration: underline;  
    -webkit-text-decoration: underline dotted; ❶  
    text-decoration: underline dotted;  
    cursor: help; ❷  
    border-bottom: 0;  
}
```

**initialism**クラスには、リスト3-22のようなスタイルで文字サイズを90%に縮小し、すべて大文字で表示されるように定義しています（①）。

▼リスト3-22 initialismクラスに定義されているスタイル

```
.initialism {  
    font-size: 90%;  
    text-transform: uppercase; ❶  
}
```

### 3.1.8 引用文

別ソースのコンテンツをブロック単位で引用する場合、**blockquote**要素でマークアップし、**blockquote**クラスを追加します（リスト3-23、図3-15）。

▼リスト3-23 blockquote要素とblockquoteクラス（typography-blockquote.html）

```
<p>引用文の例を見てみましょう</p>  
<blockquote class="blockquote">  
    <p class="mb-0">基本の引用文です。</p>  
</blockquote>
```

▼図3-15 blockquote要素とblockquoteクラスの使用例

引用文の例を見てみましょう。

基本の引用文です。

また、blockquote 要素のデフォルトスタイルは字下げ設定されているのが一般的ですが、リブート（P.96 参照）と呼ばれる Bootstrap 独自のリセットスタイルにより左マージンが 0 にリセットされているため、字下げはされません（リスト 3-24）。

▼リスト 3-24 blockquote 要素に定義されているスタイル

```
blockquote {
    margin: 0 0 1rem;
}
```

また **blockquote クラス**には、リスト 3-24 のようなスタイルでマージンサイズやフォントサイズが定義されています（リスト 3-25）。

▼リスト 3-25 blockquote クラスに定義されているスタイル

```
.blockquote {
    margin-bottom: 1rem;
    font-size: 1.25rem;
}
```

### 3.1.9 引用元の表示

引用文の引用元を特定する表示を行うには、blockquote 要素内の footer 要素に **blockquote-footer クラス**を追加し、引用元の名前を **cite 要素**としてマークアップします（リスト 3-26、図 3-16）。

▼リスト 3-26 blockquote-footer クラスと cite 要素（typography-blockquote-footer.html）

```
<blockquote class="blockquote">
    <p class="mb-0">基本の引用文です。</p>
    <footer class="blockquote-footer">文章の<cite title="引用元の名前">引用元</cite>が明示できます。→
    </footer>
</blockquote>
```

▼図 3-16 blockquote-footer クラスと cite 要素

引用文の例を見てみましょう。

基本の引用文です。

— 文章の引用元が明示できます。

なお **blockquote-footer クラス**には、リスト 3-27 のようなスタイルでブロックボックスとして表示し、フォントサイズを 80 %、文字色をグレーに定義されています（①）。また要素の前には—を挿入し、引用元であることがわかりやすく表示されるように定義されています（②）。

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

## ▼リスト3-27 blockquote-footerクラスに定義されているスタイル

```
.blockquote-footer {
  display: block;
  font-size: 80%;
  color: #868e96; ①
}

.blockquote-footer::before {
  content: "\2014 \00A0"; ②
}
```

**3.1.10 引用文の位置合わせ**

blockquote要素の配置を変更するには、必要に応じて **Textユーティリティ** (P.347 参照) を使用します。

**■ 引用文の位置合わせを中心揃えにする**

次の例では、**text-center**クラスを利用して引用文の位置合わせを中心揃えにしています（リスト3-28、図3-17）。

## ▼リスト3-28 引用の位置合わせを中心揃えにする (typography-blockquote-center.html)

```
<blockquote class="blockquote text-center">
  <p class="mb-0">基本の引用文です。</p>
  <footer class="blockquote-footer">文章の<cite title="引用元の名前">引用元</cite>が明示できます。 ↪
  </footer>
</blockquote>
```

▼図3-17 引用文の位置合わせを中心揃えにする

引用文の例を見てみましょう。

基本の引用文です。  
— 文章の引用元が明示できます。

**■ 引用文の位置合わせを右揃えにする**

次の例では、**text-right**クラスを利用して引用文の位置合わせを右揃えにしています（リスト3-29、図3-18）。

▼リスト 3-29 引用の位置合わせを右揃えにする (typography-blockquote-right.html)

```
<blockquote class="blockquote text-right">
  <p class="mb-0">基本の引用文です。</p>
  <footer class="blockquote-footer">文章の<cite title="引用元の名前">引用元</cite>が明示できます。←
</footer>
</blockquote>
```

▼図 3-18 引用文の位置合わせを右揃えにする

引用文の例を見てみましょう。

— 基本の引用文です。  
— 文章の引用元が明示できます。

### 3.1.11 リスト

**リスト要素 (ul, ol)** に **list-unstyled** クラスを追加して、li 要素にデフォルトで設定されているリストマークと左バディングを削除することができます。ネストされた li 要素のリストマーカーを削除するには、ネストされた li 要素にも **list-unstyled** クラスも追加する必要があります（リスト 3-30、図 3-19）。

#### ▼リスト 3-30 list-unstyled クラス (typography-list-unstyled.html)

```
<ul class="list-unstyled">
  <li>リストマーカーのないリスト項目</li>
  <li>リストマーカーのないリスト項目
    <ul>
      <li>ネストされたリスト項目</li>
      <li>ネストされたリスト項目</li>
      <li>ネストされたリスト項目</li>
    </ul>
  </li>
  <li>リストマーカーのないリスト項目</li>
  <li>リストマーカーのないリスト項目</li>
</ul>
```

▼図 3-19 list-unstyled クラス

- リストマーカーの無いリスト項目
- リストマーカーの無いリスト項目
  - ネストされたリスト項目
  - ネストされたリスト項目
  - ネストされたリスト項目
- リストマーカーの無いリスト項目
- リストマーカーの無いリスト項目

**list-unstyled** クラスには、左パディングなし、リストマーカーなしの設定が定義されています（リスト 3-31）。

## ▼リスト3-31 list-unstyledクラスに定義されているスタイル

```
.list-unstyled {
  padding-left: 0;
  list-style: none;
}
```

**3.1.12 インラインリスト**

リスト要素に **list-inline** クラス、li要素に **list-inline-item** クラスを組み合わせることで、リスト項目を横並びにして間にパディングを適用することができます（リスト3-32、図3-20）。

## ▼リスト3-32 インラインリスト（typography-list-inline.html）

```
<ul class="list-inline">
  <li class="list-inline-item">インラインリスト項目</li>
  <li class="list-inline-item">インラインリスト項目</li>
  <li class="list-inline-item">インラインリスト項目</li>
</ul>
```

▼図3-20 list-inlineクラスとlist-inline-itemクラスの使用例

インラインリスト項目 インラインリスト項目 インラインリスト項目

**list-inline** クラスと **list-inline-item** クラスには、次のようなスタイルでパディングサイズやリストマーカーの設定、表示形式が定義されています（リスト3-33）。

## ▼リスト3-33 list-inlineクラスとlist-inline-itemクラスに定義されているスタイル

```
.list-inline {
  padding-left: 0;
  list-style: none;
}

.list-inline-item {
  display: inline-block;
}

.list-inline-item:not(:last-child) {
  margin-right: 5px;
}
```

### 3.1.13 定義リスト

定義リスト（dl、dt、dd）にグリッドシステム用の **row** クラスおよび **col-\*** クラス（P.24、25 参照）を追加して、用語（dt 要素）と説明（dd 要素）を水平に揃えることができます。長い用語を途中で省略する場合には **text-truncate** クラスを追加して、テキストを省略記号で切り捨てるすることができます（リスト 3-34、図 3-21）。

▼リスト 3-34 定義リスト (typography-dl.html)

```
<dl class="row">
  <dt class="col-sm-3">用語1</dt>
  <dd class="col-sm-9">用語1の説明。</dd>
  <dt class="col-sm-3 text-truncate">用語2の長いテキストは切り捨てて省略</dt>
  <dd class="col-sm-9">用語2の説明。</dd>
  <dt class="col-sm-3">用語3</dt>
  <dd class="col-sm-9">
    <dl class="row">
      <dt class="col-sm-4">ネストされた用語4</dt>
      <dd class="col-sm-8">ネストされた用語4の説明</dd>
    </dl>
  </dd>
</dl>
```

▼図 3-21 定義リスト

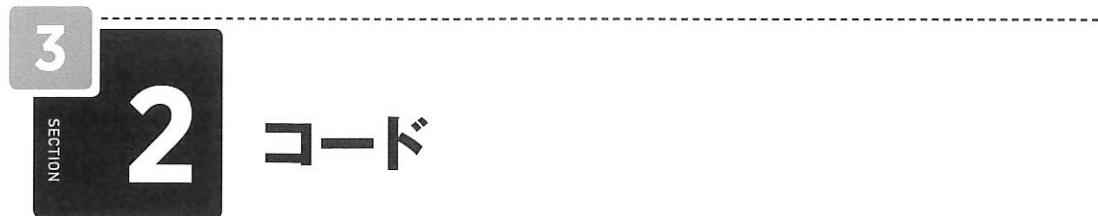
<b>用語1</b>	用語1の説明。
<b>用語2の長いテキ...</b>	用語2の説明。
<b>用語3</b>	<b>ネストされた用語4</b> ネストされた用語4の説明

**text-truncate** クラスは、リスト 3-35 のようなスタイルで、ボックスに収まらない内容を非表示（①）にして省略記号…で表示（②）するように定義されています。またスペース、タブ、改行を半角スペースで表示し、自動改行しないように定義されています（③）。

▼リスト 3-35 text-truncate クラスに定義されているスタイル

```
.text-truncate {
  overflow: hidden; ①
  text-overflow: ellipsis; ②
  white-space: nowrap; ③
}
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10



Bootstrap で定義済みのスタイル、クラスを使用して、インラインのコード表記や複数行のコードブロックを表示します。

### 3.2.1 インラインのコード表記

インラインのコード表示部分を `code` 要素でマークアップします。<>は文字参照（&lt; と &gt;）を使って置き換える必要があります（リスト 3-36、図 3-22）。

▼リスト 3-36 インラインのコード表記（code-inline.html）

インラインのコード表示部分を<code>&lt;code&gt;</code>タグで囲みます。

▼図 3-22 インラインのコード表記

インラインのコード表示部分を `<code>` タグで囲みます。

Bootstrap における `code` 要素は、リスト 3-37 のようなスタイルでフォント設定が定義されています。

▼リスト 3-37 `code` 要素に定義されたスタイル

```
pre,  
code,  
kbd,  
samp {  
    font-family: SFMono-Regular, Menlo, Monaco, Consolas, "Liberation Mono", "Courier New", monospace;  
    font-size: 1em;  
}  
…中略…  
code {  
    font-size: 87.5%;  
    color: #e83e8c;  
    word-break: break-word;  
}  
…中略…
```

```
pre code {
  font-size: inherit;
  color: inherit;
  word-break: normal;
}
a > code {
  color: inherit;
}
```

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

### 3.2.2 コードブロックの表記

複数行のコードは pre 要素でマークアップします。<> は文字参照を使って置き換える必要があります。必要に応じて、**pre-scrollable クラス**を追加すると、最大高さが 350px に設定され、縦スクロールバーが表示されます（リスト 3-38、図 3-23）。

▼リスト 3-38 コードブロック（複数行のコード）の表記（code-pre.html）

```
<pre><code>&lt;p&gt;サンプルテキスト</p&gt;&lt;p&gt;2行目のサンプルテキスト</p&gt;</code></pre>
```

▼図 3-23 コードブロック（複数行のコード）の表記

```
<p>サンプルテキスト</p>
<p>2行目のサンプルテキスト</p>
```

Bootstrap における pre 要素、code 要素には、リスト 3-39 のようなスタイルでフォント設定やマージンサイズ、表示方法が定義されています。

また Internet Explorer 11、Edge で自動非表示になるスクロールバーを強制的に表示にする設定（①）も定義されています。

▼リスト 3-39 pre 要素、code 要素に定義されたスタイル

```
pre,
code,
kbd,
samp {
  font-family: SFMono-Regular, Menlo, Monaco, Consolas, "Liberation Mono", "Courier New", monospace;
  font-size: 1em;
}

pre {
  margin-top: 0;
  margin-bottom: 1rem;
  overflow: auto;
  -ms-overflow-style: scrollbar; ①
```

```

}
…中略…
pre {
  display: block;
  font-size: 87.5%;
  color: #212529;
}

pre code {
  font-size: inherit;
  color: inherit;
  word-break: normal;
}

```

### 3.2.3 変数の表記

`var` 要素のデフォルトスタイルを使って、変数をイタリック体で表記します（リスト 3-40、図 3-24）。

▼リスト 3-40 変数の表記（code-var.html）

```
<var>y</var> = <var>m</var><var>x</var> + <var>b</var>
```

▼図 3-24 変数の表記



$y = mx + b$

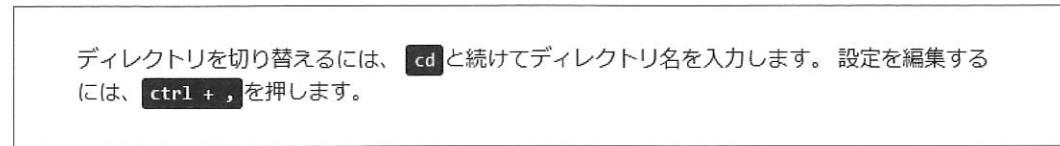
### 3.2.4 ユーザーインプットの表記

`kbd` 要素を使用して、通常キーボードから入力される入力テキストを表記します（リスト 3-41、図 3-25）。

▼リスト 3-41 ユーザーインプットの表記（code-kbd.html）

```
ディレクトリを切り替えるには、<kbd>cd</kbd>と続けてディレクトリ名を入力します。設定を編集するには、←<kbd><kbd>ctrl</kbd></kbd> + <kbd>, </kbd></kbd>を押します。
```

▼図 3-25 ユーザーインプットの表記



ディレクトリを切り替えるには、**cd**と続けてディレクトリ名を入力します。設定を編集するには、**ctrl + ,**を押します。

Bootstrap における `kbd` 要素には、リスト 3-42 のようなスタイルでフォント設定やパディングサイズ、背景色や角丸形状が定義されています。

▼リスト 3-42 `kbd` 要素に定義されているスタイル

```
pre,
code,
kbd,
samp {
    font-family: SFMono-Regular, Menlo, Monaco, Consolas, "Liberation Mono", "Courier New", monospace;
    font-size: 1em;
}
...中略...
kbd {
    padding: 0.2rem 0.4rem;
    font-size: 87.5%;
    color: #fff;
    background-color: #212529;
    border-radius: 0.2rem;
}

kbd kbd {
    padding: 0;
    font-size: 100%;
    font-weight: 700;
}
```

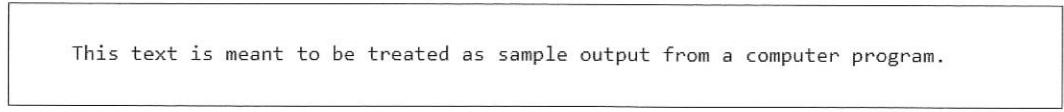
### 3.2.5 サンプル出力

`samp` 要素を使用して、プログラムからのサンプル出力を表示します（リスト 3-43、図 3-26）。

▼リスト 3-43 サンプル出力 (code-samp.html)

```
<samp>This text is meant to be treated as sample output from a computer program.</samp>
```

▼図 3-26 サンプル出力



```
This text is meant to be treated as sample output from a computer program.
```

Bootstrap における `samp` 要素には、リスト 3-44 のようなスタイルでフォント設定が定義されています。等幅フォントでフォントサイズは 1em に定義され、コード表記らしいスタイルになっています。

## ▼リスト3-44 samp要素に定義されているスタイル

```
pre,
code,
kbd,
samp {
  font-family: SFMono-Regular, Menlo, Monaco, Consolas, "Liberation Mono", "Courier New", monospace;
  font-size: 1em;
}
```

**COLUMN 定義済みのスタイルの記述場所とカスタマイズ**

本章で例示している定義済みのスタイルは、「Bootstrap の導入」(P.14 参照) でダウンロードした **bootstrap.css** に記述されています。このファイルは元々、ソースコード版 Bootstrap (P.444 参照) 内の複数の Sass ファイルをコンパイルしたものです。例えばタイポグラフィに関するスタイルは、**\_type.scss** や **\_reboot.scss** といった Sass ファイルに定義されており、その設定値は **\_variables.scss** に変数としてまとめられています。この変数の値を変更することで、Bootstrap 全体のタイポグラフィのスタイルをカスタマイズすることも可能になります。

▼リスト45 ソースファイルに定義されたスタイルの例 (**\_reboot.scss**)

```
pre,
code,
kbd,
.samp {
  font-family: $font-family-monospace;
  font-size: 1em; // Correct the odd `em` font sizing in all browsers.
```

▼リスト46 「**\_variables.scss**」にまとめられている変数の例

```
$font-family-monospace: SFMono-Regular, Menlo, Monaco, Consolas, "Liberation Mono", ↵
  "Courier New", monospace !default;
```

詳しくは、「Sass を使ってカスタマイズする」(P.439) を参照してください。

# 3 SECTION

# 3 画像

Bootstrap では、画像をレスポンシブに対応させるためのクラスも定義されています。その内容を見ていきましょう。

## 3.3.1 レスポンシブ画像

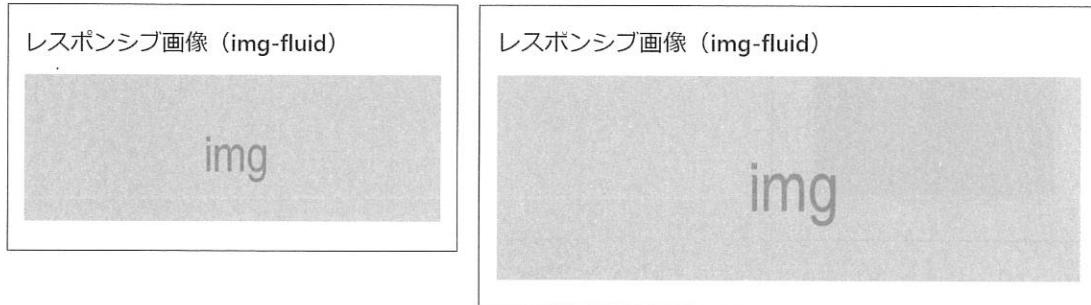
画像をレスポンシブ対応させるには、img 要素に **img-fluid クラス**を追加します（リスト 3-47、図 3-27）。

▼リスト 3-47 レスponsiv 画像 (img-fluid.html)

```
<h3 class="mb-4">レスポンシブ画像 (img-fluid) </h3>

```

▼図 3-27 ウィンドウを縮小（左）、拡大（右）させても縦横比を維持したままフィット



**img-fluid クラス**には、親要素にフィットするように**最大幅：100%、高さ：自動**のスタイルで幅や高さが定義されています（リスト 3-48）。

▼リスト 3-48 img-fluid クラスに定義されているスタイル

```
.img-fluid {
  max-width: 100%;
  height: auto;
}
```

NOTE

### SVG画像とInternet Explorer 10

Internet Explorer 10では、SVG画像に**img-fluid**クラスを追加した場合、サイズが不均衡になります。これを修正するには、必要に応じてスタイル **width: 100% !important;** を追加してください（他の画像フォーマットのサイズが不適切になるため、Bootstrapでは自動的には適用しません）。

### 3.3.2 サムネイル画像

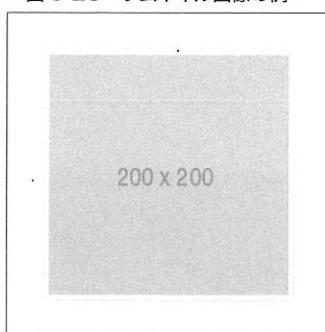
Bootstrapでは、サムネイル用の画像のスタイルとして1px幅の角丸枠が採用されています。このサムネイル画像を作成する場合は、img要素に**img-thumbnail**クラスを追加します（リスト3-49、図3-28）。

▼リスト3-49 サムネイル画像 (img-thumbnail.html)

```

```

▼図3-28 サムネイル画像の例



**img-thumbnail**クラスには、パディングサイズや背景色、枠線、角丸表示、幅や高さが定義されており、サムネイルらしいスタイルとなっています（リスト3-50）。

▼リスト3-50 img-thumbnailクラスに定義されているスタイル

```
.img-thumbnail {
  padding: 0.25rem;
  background-color: #fff;
  border: 1px solid #dee2e6;
  border-radius: 0.25rem;
  max-width: 100%;
  height: auto;
}
```

角丸枠の画像には、Borderユーティリティ（P.306参照）の**rounded**クラスを使用することもできます。