

CSS

全域的 CSS 設置、樣式與可擴充的類別以增強基礎的 HTML 元素，還有優秀的格線系統。

概觀

深入瞭解 Bootstrap 3 基礎架構的關鍵內容，包含我們讓 Web 開發變的更好、更快、更穩健的方法。

HTML5 doctype

Bootstrap 3 所使用的 HTML 元素與 CSS 屬性都需要使用到 HTML5 doctype，將它包含在所有專案中頁面的第一行。

```
<!DOCTYPE html>
<html lang="zh-Hant">
  ...
</html>
```

譯者註

`<html lang="zh">` 裡的 `lang` 屬性用來標記網頁所使用的語系。中文有三種設法，一、賦予 `zh` 值，表示為通用中文。二、賦予 `zh-Hant` 值，表示為正體中文。三、賦予 `zh-Hant-TW` 值，表示為台灣所使用的正體中文。更多資訊請參考：

1. HTML Language Code Reference
(http://www.w3schools.com/tags/ref_language_codes.asp)
2. HTML ISO Country Codes Reference
(http://www.w3schools.com/tags/ref_country_codes.asp)

3. 格式細節可參考 RFC 4646 (<http://www.ietf.org/rfc/rfc4646.txt>) 的說明。

行動優先

在 Bootstrap 2 裡，我們在框架核心面加入選擇性的行動友善樣式。在 Bootstrap 3 裡，我們改寫了整個專案，並從一開始就將行動友善放入其中。這不是簡單增加一些選擇性的行動樣式，而且直接置於框架核心當中。事實上，Bootstrap 3 是行動優先（Mobile First）。行動優先的樣式融合在框架的每個角落，而不是散落於各個文件。

為了確保適當的呈現和觸控縮放效果，加入可視區域（viewport）的 meta 標籤到頁面的 <head>。

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

你也能在可視區域的 meta 標籤中加入 user-scalable=no 設置，即可取消行動設備上的縮放功能。取消縮放功能意味著使用者只能滑動螢幕，結果就是，你的網站看起來更像是原生應用程式般的感覺。總之，我們不建議所有網站都使用此方式，因此，請自行斟酌。

```
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
```

版面與連結

Bootstrap 3 進行了基本的全域呈現、版面和連結樣式的設置。分別是：

- 為 body 設置 background-color: #fff;
- 使用 @font-family-base、@font-size-base 和 @line-height-base 屬性（attribute）作為我們版面的基礎
- 經由 @link-color 設置全域的連結顏色，並且當連結處於 :hover 時才會呈現下劃線

這些樣式設置都能在 scaffolding.less 中找到。

譯者註

Bootstrap Sass 專案中是在 _scaffolding.scss 檔案，並且 Sass 是使用 \$ 符號，變數名稱一樣。

Normalize.css

為了改善跨瀏覽器的呈現效果，我們使用了 Normalize.css (<http://necolas.github.io/normalize.css/>)，此專案由 Nicolas Gallagher (<http://twitter.com/necolas>) 和 Jonathan Neal (http://twitter.com/jon_neal) 開發維護。

容器

Bootstrap 3 需要一個 .container 容器元素來包裝網站內容與容納我們的格線系統。你可以在專案裡從下面範例的二個容器之間進行選擇。注意，由於內距（padding）和其他屬性的原因，這二個容器並不能互動巢狀套用。

.container 用於自適應且固定寬度的容器。

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

.container-fluid 用於 100% 寬度的容器，橫跨可視區域的全部寬度。

```
<div class="container-fluid">  
  ...  
</div>
```

格線系統

Bootstrap 3 包含自適應功能，以行動優先的流動格線系統，並且隨著設備或可視區域大小縮放，格線系統會自動擴大最大至 12 個欄位。它包含易於佈局的預定義類別選項，以及用於產生更有語義佈局的強大 mixins。

前言

格線系統使用一系列的行 (row · 橫向) 和列 (column · 直向) 來建立出網頁的佈局並容納你的內容。以下是 Bootstrap 3 的工作原理：

譯者註

考慮 Bootstrap 3 中的 .row 與 .col-*-* 類別的使用，文件中的「行、列」並不強烈選擇翻譯，讀者應習慣使 row 與 column 來記憶。

下文中的"中縫 (gutter)"是出版界用詞。原指書藉跨頁之間白邊。

- row 必須放在 .container (固定寬度) 或 .container-fluid (100% 寬度) 容器中，以便適當的進行對齊和內距 (padding) 調整。
- 使用 row 來建立水平群組的 column 。
- 內容應該放置在 column 之內，並且僅只有 column 可能作為 row 的直接子元素。
- 預定義好的格線類別，像是 .row 和 .col-xs-4 等等，可以用來快速建立格線佈局。而 Less 的 mixin 也能用來建立更多語義化佈局。
- column 經由 padding 建立中縫 (gutter) (column 內容之間的間距)。透過為 .row 設定負的邊距 (margin) 值，以偏移 row 裡第一個 column 與最後一個 column 的內距 (padding) 值。
- 負的邊距 (margin) 值也就是為什麼下面的範例會向外突出的原因。它就是這樣與非格線內容對齊格線中 column 的內容。
- 格線 column 建立方法是透過指定你希望的 12 可用的 column 數字範圍。舉例來說，三個相等的 column 應使用三個 .col-xs-4 來設置。

- 如果超過 12 個 column 放在單一個 row 中，每個群組額外的 column 將作為一個單元包裝到新的一行。
- 格線類別的套用，會使用設備螢幕寬度大於或等於來當判斷點，而且針對較小螢幕設備進行了格線類別的重寫。因此，套用 .col-md- 到元素上時，不僅僅是影響中螢幕設備的樣式，也會影響大螢幕設備的樣式（如果未套用 .col-lg- 的話）。

看看以下這些範例，並且應用這些原則到你的程式碼中。

媒體查詢

我們在 Less 檔案中使用以下的媒體查詢建立格線系統的關鍵判斷點。

```
/* 超小螢幕設備（手機，小於 768px） */
/* 沒有媒體查詢，因為這在 Bootstrap 3 是預設的。 */

/* 小螢幕設備（平板，768px（含）以上） */
@media (min-width: @screen-sm-min) { ... }

/* 中螢幕設備（一般桌面，992px（含）以上） */
@media (min-width: @screen-md-min) { ... }

/* 大螢幕設備（大型桌面，1200px（含）以上） */
@media (min-width: @screen-lg-min) { ... }
```

我們偶爾會在媒體查詢中包含 max-width 設置，將 CSS 的影響限制在更窄的設備之內。

```
@media (max-width: @screen-xs-max) { ... }
@media (min-width: @screen-sm-min) and (max-width: @screen-sm-max) { ... }
@media (min-width: @screen-md-min) and (max-width: @screen-md-max) { ... }
@media (min-width: @screen-lg-min) { ... }
```

格線選項

請參考以下方便的表格，以瞭解 Bootstrap 3 的格線系統是如何橫跨多種設備運作。

	超小螢幕設備 手機 (<768px)	小螢幕設備 平板 (≥768px)	中螢幕設備 桌面 (≥992px)	大螢幕設備 大桌面 (≥1200px)
格線行為	總是水平佈局	開始是折疊的，當超過判斷點時回復水平佈局		
容器寬度	無（自動）	750px	970px	1170px
類別前綴	.col-xs-	.col-sm-	.col-md-	.col-lg-
column 數	12			
column 寬度	自動	~62px	~81px	~97px
中縫寬度	30px (column 左右邊各 15px)			
可巢狀套用	是			

	超小螢幕設備 手機 (<768px)	小螢幕設備 平板 (≥768px)	中螢幕設備 桌面 (≥992px)	大螢幕設備 大桌面 (≥1200px)
位移 (Offsets)	是			
column 排序	是			

範例：由堆疊到水平排列

使用單一組 `.col-md-*` 格線類別，你就能建立一個基本的格線系統，在手機設備與平板設備（介於超小螢幕到小螢幕的範圍）上一開始是堆疊方式呈現，在桌面（中螢幕）設備會變成水平佈局。格線 `column` 必須放在任何 `.row` 之內。

.col-md-8

.col-md-4

.col-md-4

.col-md-4

```
.col-md-4
```

```
.col-md-6
```

```
.col-md-6
```

```
<div class="row">
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
</div>
<div class="row">
  <div class="col-md-8">.col-md-8</div>
  <div class="col-md-4">.col-md-4</div>
</div>
<div class="row">
  <div class="col-md-4">.col-md-4</div>
  <div class="col-md-4">.col-md-4</div>
  <div class="col-md-4">.col-md-4</div>
</div>
<div class="row">
  <div class="col-md-6">.col-md-6</div>
  <div class="col-md-6">.col-md-6</div>
</div>
```

範例：流動容器

將最外層的 `.container` 修改為 `.container-fluid`，就可以將任何固定寬度格線佈局轉換為 100% 寬度格線佈局。

```
<div class="container-fluid">
  <div class="row">
    ...
  </div>
</div>
```

範例：行動與桌面

不希望你的 `column` 在小螢幕設備只是堆疊在一起？那就在你的 `column` 加入超小螢幕和中螢幕設備

的格線類別，即 .col-xs-* 和 .col-md-*。參考以下範例，以更瞭解它們是如何運作。

.col-xs-12 .col-md-8

.col-xs-6 .col-md-4

.col-xs-6 .col-md-4

.col-xs-6 .col-md-4

.col-xs-6

.col-xs-6

```
<!-- 製作一個 100% 寬度和兩個 50% 寬度，在行動設備上堆疊的 column -->
<div class="row">
```

```
  <div class="col-xs-12 col-md-8">.col-xs-12 .col-md-8</div>
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
</div>
```

```
<!-- 一開始，在行動設備上 column 會是 50% 寬度，提升至桌面大小後，寬度會平均為
33.3% -->
```

```
<div class="row">
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
</div>
```

```
<!-- 在行動或桌面設備，column 寬度總是 50% 寬度 -->
```

```
<div class="row">
  <div class="col-xs-6">.col-xs-6</div>
  <div class="col-xs-6">.col-xs-6</div>
</div>
```

範例：行動、平板、桌面

基於上面範例，加入平板使用的 .col-sm-* 類別，建置更動態與強大佈局。

.col-xs-12 .col-sm-6 .col-md-8

.col-xs-6 .col-md-4

.col-xs-6 .col-sm-4

.col-xs-6 .col-sm-4

.col-xs-6 .col-sm-4

```

<div class="row">
  <div class="col-xs-12 col-sm-6 col-md-8">.col-xs-12 .col-sm-6 .col-
md-8</div>
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
</div>
<div class="row">
  <div class="col-xs-6 col-sm-4">.col-xs-6 .col-sm-4</div>
  <div class="col-xs-6 col-sm-4">.col-xs-6 .col-sm-4</div>
  <!-- 選項：如果他們內容有不符合的高度，清除 XS column -->
  <div class="clearfix visible-xs-block"></div>
  <div class="col-xs-6 col-sm-4">.col-xs-6 .col-sm-4</div>
</div>

```

範例：column 換行

如果超過 12 個 column 放在單一個 row 中，每個群組額外的 column 將作為一個單元包裝到新的一行。

.col-xs-9

.col-xs-4

因為 $9 + 4 = 13 > 12$ ，此 4 column 寬度的 div 會被包裝到新的一行作為一個連續的單元。

.col-xs-6

後續 column 會順著新行繼續

```

<div class="row">
  <div class="col-xs-9">.col-xs-9</div>
  <div class="col-xs-4">.col-xs-4<br>因為  $9 + 4 = 13 > 12$ ，此 4 column 寬度的 div 會被包裝到新的一行作為一個連續的單元。</div>
  <div class="col-xs-6">.col-xs-6<br>後續 column 會順著新行繼續</div>
</div>

```

重設自適應 column

隨著四層 (tier) 的格線可使用，你一定會碰到問題，其中，在一些明確的判斷點，當你其中一個 column 比其他還高時，你的 column 清除的不完全正確。要修正此問題，組合使用 .clearfix 和我們的自適應工具類別。

.col-xs-6 .col-sm-3

重置你的可視區域大小，或是在你的手機瀏覽此範例。

.col-xs-6 .col-sm-3

.col-xs-6 .col-sm-3

.col-xs-6 .col-sm-3

```

<div class="row">
  <div class="col-xs-6 col-sm-3">.col-xs-6 .col-sm-3</div>
  <div class="col-xs-6 col-sm-3">.col-xs-6 .col-sm-3</div>

  <!-- 僅在所需的可視區域加入額外的 clearfix -->
  <div class="clearfix visible-xs-block"></div>

  <div class="col-xs-6 col-sm-3">.col-xs-6 .col-sm-3</div>
  <div class="col-xs-6 col-sm-3">.col-xs-6 .col-sm-3</div>
</div>

```

另外，在自適應判斷點進行 column 清除時，你也許需要重置位移（ offsets ）、 push 或 pull 。查看這此實作的格線範例 ([/bs3/Examples?name=grid](#)) 。

```

<div class="row">
  <div class="col-sm-5 col-md-6">.col-sm-5 .col-md-6</div>
  <div class="col-sm-5 col-sm-offset-2 col-md-6 col-md-offset-0">.col-sm-5
  .col-sm-offset-2 .col-md-6 .col-md-offset-0</div>
</div>

<div class="row">
  <div class="col-sm-6 col-md-5 col-lg-6">.col-sm-6 .col-md-5 .col-
  lg-6</div>
  <div class="col-sm-6 col-md-5 col-md-offset-2 col-lg-6 col-lg-
  offset-0">.col-sm-6 .col-md-5 .col-md-offset-2 .col-lg-6 .col-lg-
  offset-0</div>
</div>

```

column 位移

使用 .col-md-offset-* 類別可以將 column 向右進行位移。這些類別是使用 * 選擇器為當下的 column 增加左邊距（ margin-left ）的值。舉例說明， .col-md-offset-4 類別將會把 .col-md-4 向右位移 4 個 column 的寬度。

.col-md-4

.col-md-4 .col-md-offset-4

.col-md-3 .col-md-offset-3

.col-md-3 .col-md-offset-3

.col-md-6 .col-md-offset-3

```

<div class="row">
  <div class="col-md-4">. col-md-4</div>
  <div class="col-md-4 col-md-offset-4">. col-md-4 . col-md-offset-4</div>
</div>
<div class="row">
  <div class="col-md-3 col-md-offset-3">. col-md-3 . col-md-offset-3</div>
  <div class="col-md-3 col-md-offset-3">. col-md-3 . col-md-offset-3</div>
</div>
<div class="row">
  <div class="col-md-6 col-md-offset-3">. col-md-6 . col-md-offset-3</div>
</div>

```

巢狀 column

為了巢狀加入你的內容到預設的格線系統中，加入一行新的 `.row` 並設置 `.col-sm-*` 的 `column` 在已存在的 `.col-sm-*` `column` 內。巢狀的 `row` 該包含一組含 12 或更少的 `column` (它並不要求你使用所有可用的 12 個 `column`) 。

Level 1: `.col-sm-9`

Level 2: `.col-xs-8 .col-sm-6`

Level 2: `.col-xs-4 .col-sm-6`

```

<div class="row">
  <div class="col-sm-9">
    Level 1: .col-sm-9
    <div class="row">
      <div class="col-xs-8 col-sm-6">
        Level 2: .col-xs-8 .col-sm-6
      </div>
      <div class="col-xs-4 col-sm-6">
        Level 2: .col-xs-4 .col-sm-6
      </div>
    </div>
  </div>

```

column 排序

透過 `.col-md-push-*` 和 `.col-md-pull-*` 修飾類別，可以很輕鬆改變內建格線系統的 `column` 排序。

`.col-md-9 .col-md-push-3`

`.col-md-3 .col-md-pull-9`

```
<div class="row">
  <div class="col-md-9 col-md-push-3".col-md-9 .col-md-push-3</div>
  <div class="col-md-3 col-md-pull-9".col-md-3 .col-md-pull-9</div>
</div>
```

Less mixin 和變數

除了使用在快速佈局的預建置的格線類別外，Bootstrap 3 也包含 Less 變數和 mixin 來幫助你快速產生簡單、語義化的佈局。

變數

變數用來決定 column 的數量、中縫寬度和媒體查詢的判斷點（從此處開始浮動 column）。我們使用這些變數產生預定義好的格線類別文件（如前所示），以及自訂 mixin（如下列出）。

```
@grid-columns:          12;
@grid-gutter-width:     30px;
@grid-float-breakpoint: 768px;
```

Mixin

Mixin 與格線變數一起使用，用來為每種 column 產生語義的 CSS。

```
// 建立一個用於一系列的 column 包裝容器
.make-row(@gutter: @grid-gutter-width) {
    // 清除浮動的 column
    .clearfix();

    @media (min-width: @screen-sm-min) {
        margin-left: (@gutter / -2);
        margin-right: (@gutter / -2);
    }

    // 負的邊距 (margin) 值套用至 row 以對齊 column 的內容
    .row {
        margin-left: (@gutter / -2);
        margin-right: (@gutter / -2);
    }
}

// 產生超小螢幕 column
.make-xs-column(@columns; @gutter: @grid-gutter-width) {
    position: relative;
    // 當 column 為空，預防折疊行為
    min-height: 1px;
    // 經由 padding 填入中縫
    padding-left: (@gutter / 2);
    padding-right: (@gutter / 2);

    // 基於可用 column 的數量計算寬度
    @media (min-width: @grid-float-breakpoint) {
        float: left;
        width: percentage((@columns / @grid-columns));
    }
}

// 產生小螢幕 column
.make-sm-column(@columns; @gutter: @grid-gutter-width) {
    position: relative;
    // 當 column 為空，預防折疊行為
    min-height: 1px;
    // 經由 padding 填入中縫
    padding-left: (@gutter / 2);
    padding-right: (@gutter / 2);

    // 基於可用 column 的數量計算寬度
    @media (min-width: @screen-sm-min) {
        float: left;
        width: percentage((@columns / @grid-columns));
    }
}

// 產生小螢幕 column 位移
.make-sm-column-offset(@columns) {
    @media (min-width: @screen-sm-min) {
```

```

        margin-left: percentage((@columns / @grid-columns));
    }
}

.make-sm-column-push(@columns) {
    @media (min-width: @screen-sm-min) {
        left: percentage((@columns / @grid-columns));
    }
}

.make-sm-column-pull(@columns) {
    @media (min-width: @screen-sm-min) {
        right: percentage((@columns / @grid-columns));
    }
}

// 產生中螢幕 column
.make-md-column(@columns; @gutter: @grid-gutter-width) {
    position: relative;
    // 當 column 為空，預防折疊行為
    min-height: 1px;
    // 經由 padding 填入中縫
    padding-left: (@gutter / 2);
    padding-right: (@gutter / 2);

    // 基於可用 column 的數量計算寬度
    @media (min-width: @screen-md-min) {
        float: left;
        width: percentage((@columns / @grid-columns));
    }
}

// 產生中螢幕 column 位移
.make-md-column-offset(@columns) {
    @media (min-width: @screen-md-min) {
        margin-left: percentage((@columns / @grid-columns));
    }
}

.make-md-column-push(@columns) {
    @media (min-width: @screen-md-min) {
        left: percentage((@columns / @grid-columns));
    }
}

.make-md-column-pull(@columns) {
    @media (min-width: @screen-md-min) {
        right: percentage((@columns / @grid-columns));
    }
}

// 產生大螢幕 column
.make-lg-column(@columns; @gutter: @grid-gutter-width) {
    position: relative;
    // 當 column 為空，預防折疊行為
    min-height: 1px;
    // 經由 padding 填入中縫
}

```

使用範例

你可以修改變數值或是僅使用 mixin 所提供預設值。以下範例使用預設的設定值去建立一個含有間距兩個 column (two-column) 牆局。

```
.wrapper {  
  .make-row();  
}  
.content-main {  
  .make-lg-column(8);  
}  
.content-secondary {  
  .make-lg-column(3);  
  .make-lg-column-offset(1);  
}
```

```
<div class="wrapper">  
  <div class="content-main">...</div>  
  <div class="content-secondary">...</div>  
</div>
```

排版

標題

所有 HTML 的標題元素，`<h1>` 到 `<h6>` 均可使用。另外還提供 `.h1` 到 `.h6` 類別，當你想要搭配標題字型樣式並且在行內元素 (inline) 顯示你的文字。

EXAMPLE

h1. Bootstrap heading

Semibold 36px

h2. Bootstrap heading

Semibold 30px

h3. Bootstrap heading

Semibold 24px

h4. Bootstrap heading

Semibold 18px

h5. Bootstrap heading Semibold 14px

h6. Bootstrap heading Semibold 12px

```
<h1>h1. Bootstrap heading</h1>
<h2>h2. Bootstrap heading</h2>
<h3>h3. Bootstrap heading</h3>
<h4>h4. Bootstrap heading</h4>
<h5>h5. Bootstrap heading</h5>
<h6>h6. Bootstrap heading</h6>
```

在任何主標題內使用 `<small>` 標籤或 `.small` 類別，以產生更輕小的副標題效果。

EXAMPLE

h1. Bootstrap heading Secondary text

h2. Bootstrap heading Secondary text

h3. Bootstrap heading Secondary text

h4. Bootstrap heading Secondary text

h5. Bootstrap heading Secondary text

h6. Bootstrap heading Secondary text

```
<h1>h1. Bootstrap heading <small>Secondary text</small></h1>
<h2>h2. Bootstrap heading <small>Secondary text</small></h2>
<h3>h3. Bootstrap heading <small>Secondary text</small></h3>
<h4>h4. Bootstrap heading <small>Secondary text</small></h4>
<h5>h5. Bootstrap heading <small>Secondary text</small></h5>
<h6>h6. Bootstrap heading <small>Secondary text</small></h6>
```

頁面主體

Bootstrap 3 的全域設置中，預設 `font-size` 是 14px，`line-height` 是 1.428。這些屬性值被套用至 `<body>` 和 `p`（段落）元素上。此外，`<p>` 設置了一個 1/2 行高（預設 10px）的底部邊距（`margin`）值。

EXAMPLE

Nullam quis risus eget urna mollis ornare vel eu leo. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nullam id dolor id nibh ultricies vehicula.

Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec ullamcorper nulla non metus auctor fringilla. Duis mollis, est non commodo luctus, nisi erat porttitor ligula, eget lacinia odio sem nec elit. Donec ullamcorper nulla non metus auctor fringilla.

Maecenas sed diam eget risus varius blandit sit amet non magna. Donec id elit non mi porta gravida at eget metus. Duis mollis, est non commodo luctus, nisi erat porttitor ligula, eget lacinia odio sem nec elit.

```
<p>...</p>
```

譯者註

英文和中文在閱讀的感受上所有不同，開發者請自行測試並調整之。

EXAMPLE

葡萄美酒夜光杯，欲飲琵琶馬上催。

醉臥沙場君莫笑，古來征戰幾人回。

前導主體

製作一個讓段落突出的效果（字型加大加粗），僅需要在段落加入 .lead 類別。

EXAMPLE

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Duis mollis, est non commodo luctus.

```
<p class="lead">...</p>
```

使用 Less 建置

在 variables.less 文件中定義了兩個 LESS 變數來決定佈局大小：@font-size-base 和 @line-height-base。第一個變數定義 font-size 基準，第二個變數定義 line-height 基準。我們使用這些變數和一些簡單的數學計算去建立其他所有頁面元素的 margin、padding 和 line-height。自定義這些變數可以改變 Bootstrap 3 預設樣式。

行內文字元素

標記文字

由於其他方面的關聯性，醒目提示一段的文字內容。使用 `<mark>` 標籤。

EXAMPLE

你可以使用 `mark` 標籤以 醒目提示 文字內容。

你可以使用 `mark` 標籤以 `<mark>`醒目提示`</mark>` 文字內容。

刪除文字

使用 `` 標籤表明此文字區塊已被刪除。

EXAMPLE

此行文字是要被視為已刪除文字。

``此行文字是要被視為已刪除文字。``

刪除線效果

使用 `<s>` 用於指出此文字區塊已經不再和段落有相關聯。

EXAMPLE

This line of text is meant to be treated as no longer accurate.

`<s>`This line of text is meant to be treated as no longer accurate.`</s>`

插入文字

使用 `<ins>` 標籤，用來指出被額外加入至文件的文字。

EXAMPLE

此行文字是要被視為已新加入的文字。

`<ins>`此行文字是要被視為已新加入的文字。`</ins>`

譯者註

`` 和 `<s>` 就呈現上的效果是一模一樣，主要差異是在語義上。html5 Doctor有篇不錯的 Comparing and contrasting ins, del, and s (<http://html5doctor.com/ins-del-s/>) 文章可以參考。

底線文字

使用 `<u>` 標籤以呈現底線文字效果。

EXAMPLE

此行文將會呈現底線效果。

`<u>此行文將會呈現底線效果。</u>`

利用 HTML 的預設加強重要性的標籤並加入少量樣式。

較小文字

對於不那麼重要的行內 (`inline`) 或區塊 (`block`) 元素裡的文字，使用 `<small>` 標籤包圍的文字，將被設置為父容器的 85% 大小。標題元素中巢狀的 `<small>` 元素，將依標題的 `font-size` 來設置。

你也可以在行內元素使用 `.small` 類別來代替任何 `<small>` 元素。

EXAMPLE

此行文字是會被視為印刷精美的文字。

`<small>此行文字是會被視為印刷精美的文字。</small>`

粗體

透過加重 `font-weight` 值 (加粗字體)，用於強調某一片段文字。

EXAMPLE

此片段文字呈現粗體文字效果。

`呈現粗體文字效果`

斜體

使用斜體來強調某一段文字。

EXAMPLE

此片段文字呈現斜體文字效果。

`呈現斜體文字效果`

譯者註

就筆者本身的經驗，`em` 元素預設的斜體效果，在中文閱讀上不太是加分效果。一般會透過 `font-style:normal;` 來取消樣式，並且另外進行顏色或粗體來強調。

可選元素

在 HTML5 中可放心使用 `` 和 `<i>`。`` 用於加粗字體或短句，沒有額外重要的語義。`<i>` 常用於發言或技術術語等。

對齊類別

透過文字對齊類別，可以輕鬆將元件裡的文字重新對齊。

EXAMPLE

靠左對齊文字。

置中對齊文字。

靠右對齊文字。

平均對齊文字。

不換行文字。

```
<p class="text-left">靠左對齊文字。</p>
<p class="text-center">置中對齊文字。</p>
<p class="text-right">靠右對齊文字。</p>
<p class="text-justify">平均對齊文字。</p>
<p class="text nowrap">不換行文字。</p>
```

譯者註

1. `.text-justify` 的範例不是很好（不管官方或我翻譯後的），可另外參考 w3schools 的 `text-align` (http://www.w3schools.com/cssref/pr_text_text-align.asp) 範例。
2. Bootstrap 3 預設的 `.text-justify` 類別僅對英文有效，如果想要中文有一樣的效果，在原始碼的 `.text-justify` 類別附加一個 `text-justify:inter-ideograph;` 屬性設定。

轉換類別

透過文字轉換類別，將元件裡的英文文字進行大小寫轉換。

EXAMPLE

lowercased text. (全小寫)

UPPERCASED TEXT. (全大寫)

Capitalized Text. (字首大寫)

```
<p class="text-lowercase">Lowercased text. ( 全小寫 ) </p>
<p class="text-uppercase">Uppercased text. ( 全大寫 ) </p>
<p class="text-capitalize">Capitalized text. ( 字首大寫 ) </p>
```

縮寫

當滑鼠停留在縮寫或縮寫詞上時會顯示完整內容，Bootstrap 3 增強 HTML `<abbr>` 元素的樣式。縮寫元素帶有 `title` 屬性，外觀會帶有淡淡的虛底線，滑鼠移至縮寫詞上面時會帶有問號指標。

基本縮寫

想看完整的內容，可以把滑鼠停留在縮寫詞上，但 `<abbr>` 元素必須包含 `title` 屬性。

EXAMPLE

An abbreviation of the word attribute is attr (attribute).

```
<abbr title="attribute">attr</abbr>
```

字首縮寫

加入 `.initialism` 類別到縮寫裡，會使用更小一些的 `font-size`。

EXAMPLE

HTML (HYPERTEXT MARKUP LANGUAGE) is the best thing since sliced bread.

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

地址

讓連絡資訊以最接近日常使用格式呈現。在每行結尾加入 `
` 以保留格式。

EXAMPLE

Twitter, Inc.

795 Folsom Ave, Suite 600

San Francisco, CA 94107

P: (Phone) (123) 456-7890

Full Name
first.last@example.com (mailto:#)

```
<address>
  <strong>Twitter, Inc.</strong><br>
  795 Folsom Ave, Suite 600<br>
  San Francisco, CA 94107<br>
  <abbr title="Phone">P:</abbr> (123) 456-7890
</address>
```

```
<address>
  <strong>Full Name</strong><br>
  <a href="mailto:#">first.last@example.com</a>
</address>
```

引用

在你的文件中引用其他來源的內容。

預設引用

將任何 HTML (HyperText Markup Language) 包含在 <blockquote> 其中即為引用。直接的引用建議用 <p>。

EXAMPLE

少小離家老大回，鄉音無改鬢毛催。兒童相見不相識，笑問客從何處來。

```
<blockquote>
  <p>少小離家老大回，鄉音無改鬢毛催。兒童相見不相識，笑問客從何處來。</p>
</blockquote>
```

譯者註

<blockquote> 是區塊元素，通常是整段或較長內容的引用。<q> 則是行內元素，合適短句、短言的引用，另外，主流瀏覽器會自動在被 <q> 所引用的文字前後加上左右雙引號。

引用選項

在標準 <blockquote> 裡，可以很簡單的改變樣式和內容。

說明來源

加入 <footer> 以標識來源。將來源名稱放在 <cite>。

EXAMPLE

少小離家老大回，鄉音無改鬢毛催。兒童相見不相識，笑問客從何處來。

—回鄉偶書，作者：賀知章

```
<blockquote>
```

```
  <p>少小離家老大回，鄉音無改鬢毛催。兒童相見不相識，笑問客從何處來。</p>
```

```
  <footer>回鄉偶書，作者：賀知章</cite></footer>
```

```
</blockquote>
```

譯者註

<q> 使用 `cite` 屬性來註明來源。<code> 標籤的 `title` 屬性與 <q> 的 `cite` 屬性也可以是 URL (Uniform Resource Locator) 網址。例如，引用 Wiki 的資料。

換邊對齊

加入 `.blockquote-reverse` 類別到 `blockquote` 標籤中，那麼內容會靠右對齊呈現。

EXAMPLE

少小離家老大回，鄉音無改鬢毛催。兒童相見不相識，笑問客從何處來。

回鄉偶書，作者：賀知章—

```
<blockquote class="blockquote-reverse">
  ...
</blockquote>
```

清單

無序清單

清單中的項目順序不是那麼明顯或重要。

EXAMPLE

- Lorem ipsum dolor sit amet
- Consectetur adipiscing elit
- Integer molestie lorem at massa
- Facilisis in pretium nisl aliquet
- Nulla volutpat aliquam velit
 - Phasellus iaculis neque
 - Purus sodales ultricies

- Vestibulum laoreet porttitor sem
 - Ac tristique libero volutpat at
 - Faucibus porta lacus fringilla vel
 - Aenean sit amet erat nunc
 - Eget porttitor lorem
-

```
<ul>
  <li>...</li>
</ul>
```

有序清單

清單中的項目順序是明顯或重要的。

EXAMPLE

1. Lorem ipsum dolor sit amet
 2. Consectetur adipiscing elit
 3. Integer molestie lorem at massa
 4. Facilisis in pretium nisl aliquet
 5. Nulla volutpat aliquam velit
 6. Faucibus porta lacus fringilla vel
 7. Aenean sit amet erat nunc
 8. Eget porttitor lorem
-

```
<ol>
  <li>...</li>
</ol>
```

移除清單樣式

移除清單項目預設 `list-style` 樣式和左邊距 (`margin`) 值 (僅對直接子元素有效)。這僅會套用到子清單項目上，意思是，你必須將所有巢狀清單都加入此類別才能得到相同的樣式。

EXAMPLE

 Lorem ipsum dolor sit amet
 Consectetur adipiscing elit
 Integer molestie lorem at massa
 Facilisis in pretium nisl aliquet
 Nulla volutpat aliquam velit

- Phasellus iaculis neque
- Purus sodales ultricies
- Vestibulum laoreet porttitor sem
- Ac tristique libero volutpat at

 Faucibus porta lacus fringilla vel
 Aenean sit amet erat nunc
 Eget porttitor lorem

```
<ul class="list-style">  
  <li>...</li>  
</ul>
```

行內清單

透過設置 `display: inline-block;` 和一些內距 (`padding`) ，將所有清單項目放在同一行上顯示。

EXAMPLE

`Lorem ipsum Phasellus iaculis Nulla volutpat`

```
<ul class="list-inline">  
  <li>...</li>  
</ul>
```

術語清單

一個專業術語和相關描述的清單。

EXAMPLE

URI (Uniform Resource Identifier)

在電腦術語中，統一資源標識是一個用於標識某一互聯網資源名稱的字符串。

URL (Universal Resource Locator)

統一資源定位器，有時也被俗稱為網頁地址（網址）。

如同在網路上的門牌，是 Internet 上標準的資源的地址（Address）。

URN (Uniform resource name)

統一資源名稱是統一資源標識的歷史名字，它使用urn:作為URI scheme。

```
<dl>  
  <dt>專業術語</dt>  
  <dd>相關描述</dd>  
</dl>
```

水平佈局術語清單

讓專業術語與描述在 `<dl>` 同一行並行（水平）呈現。一開始會如同預設 `<dl>` 是上下堆疊樣式，但隨著瀏覽者逐漸展開畫面而轉換水平佈局。

EXAMPLE

Description lists

A description list is perfect for defining terms.

Euismod

Vestibulum id ligula porta felis euismod semper eget lacinia odio sem nec elit.

Donec id elit non mi porta gravida at eget metus.

Malesuada porta

*Etiam porta sem malesuada magna mollis euismod.
Felis euismod semper eget lacinia
Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum
massa justo sit amet risus.*

```
<dl class="dl-horizontal">  
  <dt>...</dt>  
  <dd>...</dd>  
</dl>
```

自動截斷

透過 `text-overflow` 屬性，讓水平佈局術語清單會將過長無法在左邊 `column` 中完全呈現的專業術語截去一部分（例如，上面範例的第四個 `<dt>` 原始內容為：「*Felis euismod semper eget lacinia*」）。而在較窄的可視區域中，會改變回預設堆疊（垂直）佈局方式。

程式碼

行內程式碼

使用 `<code>` 來包裝一小段行內程式碼。

EXAMPLE

For example, `<section>` should be wrapped as inline.

For example, `<code><section></code>` should be wrapped as inline.

譯者註

`<pre>` 標籤裡 Tab 會被完整呈現，請先預覽確定程式碼排版如你所願。

使用者輸入

使用 `<kbd>` 標籤來指出這是要經由鍵盤來輸入的內容。

EXAMPLE

To switch directories, type `cd` followed by the name of the directory.

To edit settings, press `ctrl + ,`

To switch directories, type `<kbd>cd</kbd>` followed by the name of the

directory.

To edit settings, press `<kbd><ctrl>+<kb>`,
</kb></kb>

程式碼區塊

使用 `<pre>` 來呈現多行程式碼。為了能正確呈現，請務必將程式碼中的所有角括號進行轉義。

EXAMPLE

```
<p>Sample text here...</p>
```

```
<pre>&lt;p&gt;Sample text here...&lt;/p&gt;</pre>
```

你也選擇加入 `.pre-scrollable` 類別，它會把該區塊最大高度設置為 350px 且帶有一個滑動的 Y 軸。

變數

使用 `<var>` 標籤來指出這是一個變數。

EXAMPLE

$$y = mx + b$$

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

範例輸出

使用 `<samp>` 來指出這個區塊是由程式所輸出的內容。

EXAMPLE

此行文字是要被視為由電腦程式所輸出的文字。

```
<samp>此行文字是要被視為由電腦程式所輸出的文字。</samp>
```

表格

基本範例

為任何 `<table>` 增加基本類別 `.table`，會有少量的內距（padding）和水平分隔線等基本樣式。這看起來像是多餘的，但考慮到表格相當廣泛與其他外掛配合使用，例如，日曆和日期選擇器，我們偏向於獨立出我們自訂的表格樣式。

EXAMPLE

Optional table caption.

#	First Name	Last Name	Username
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

```
<table class="table">
  ...
</table>
```

條紋式 row

使用 `.table-striped` 類別，可以為表格的 `<tbody>` 加入像斑馬條紋的樣式。

跨瀏覽器相容性

條紋式表格是經由 `:nth-child` CSS 選擇器來實現，它不被 Internet Explorer 8 支援。

EXAMPLE

#	First Name	Last Name	Username
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

```
<table class="table table-striped">
  ...
</table>
```

邊框表格

加入 `.table-bordered` 類別，為表格和所有儲存格（cell）的四邊都加入邊框效果。

EXAMPLE

#	First Name	Last Name	Username
1	Mark	Otto	@mdo
	Mark	Otto	@TwBootstrap
2	Jacob	Thornton	@fat
3	Larry the Bird		@twitter

```
<table class="table table-bordered">
  ...
</table>
```

移入 row

加入 `.table-hover` 類別，可以為表格 `<tbody>` 裡的 row 加入對滑鼠移入（hover）移出作顏色反差的效果。

EXAMPLE

#	First Name	Last Name	Username
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry the Bird		@twitter

```
<table class="table table-hover">
  ...
</table>
```

緊湊表格

加入 `.table-condensed` 類別，將表格儲存格（cell）的內距（padding）減半，讓表格更加緊湊。

EXAMPLE

#	First Name	Last Name	Username
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry the Bird		@twitter

```
<table class="table table-condensed">
  ...
</table>
```

狀態類別

使用狀態類別可以為表格的 row 或單獨的儲存格 (cell) 設置顏色。

類別	說明
.active	當滑鼠移入某個 row 或儲存格 (cell) 時設置顏色。
.success	指出一個成功或積極的動作。
.info	
.warning	指出一個需要注意的警告。
.danger	指出一個危險或潛在有害的行為。

EXAMPLE

#	Column heading	Column heading	Column heading
1	Column content	Column content	Column content
2	Column content	Column content	Column content
3	Column content	Column content	Column content
4	Column content	Column content	Column content
5	Column content	Column content	Column content
6	Column content	Column content	Column content
7	Column content	Column content	Column content
8	Column content	Column content	Column content

#	Column heading	Column heading	Column heading
9	Column content	Column content	Column content

```
<!-- 在 row 設置 -->
<tr class="active">...</tr>
<tr class="success">...</tr>
<tr class="warning">...</tr>
<tr class="danger">...</tr>
<tr class="info">...</tr>

<!-- 在儲存格 (`td` 或 `th` ) 設置 -->
<tr>
  <td class="active">...</td>
  <td class="success">...</td>
  <td class="warning">...</td>
  <td class="danger">...</td>
  <td class="info">...</td>
</tr>
```

傳達意義至輔助技術

使用顏色加入到表格的 row 或單獨的儲存格 (cell) 僅是提供視覺效果上的含義，這不會被傳達至輔助技術的使用者 - 例如：螢幕閱讀器。請確保顏色所表示的訊息從內容本身上是顯而易見的（在相關 row / cell 可見的文字），或是通過其他的方式來傳達，例如，額外 .sr-only 類別的隱藏文字。

自適應表格

將任何 .table 包裝在 .table-responsive 之內，以建立自適應表格，它能讓表格在小螢幕設備（小於 768px）進行水平式的滑動。當你在任何大於 768px 的螢幕寬度觀看時，並不會和平常有何不同。

Firefox 和 fieldset 元素

Firefox 在 fieldset 樣式有些尷尬，它在涉及 width 的設置對自適應表格會造成干擾。這無法在非 Firefox 特定的 hack 被覆寫，因此，我們不在 Bootstrap 3 裡提供。可以自行使用以下程式碼修正：

```
@-moz-document url-prefix() {
  fieldset { display: table-cell; }
}
```

更多資訊，請參考 [這篇 Stack Overflow 回答](#)

(<http://stackoverflow.com/questions/17408815/fieldset-resizes-wrong-appears-to-have-unremovable-min-width-min-content/17863685#17863685>)。

EXAMPLE

| # | Table heading |
|---|---------------|---------------|---------------|---------------|---------------|---------------|
| 1 | Table cell |
| 2 | Table cell |
| 3 | Table cell |

| # | Table heading |
|---|---------------|---------------|---------------|---------------|---------------|---------------|
| 1 | Table cell |
| 2 | Table cell |
| 3 | Table cell |

```
<div class="table-responsive">
  <table class="table">
    ...
  </table>
</div>
```

表單

基本範例

一些單獨的 form 控制項會自動被賦予一些全域樣式。所有設置了 .form-control 的文字類元素，如 `<input>`、`<textarea>` 和 `<select>` 元素都會預設為 `width: 100%;`。將 `label` 標籤與 form 控制項使用 `.form-group` 包裝在一起以取得最佳的排列組合。

EXAMPLE

電子郵件

輸入電子郵件

密碼

密碼

檔案上傳

在此示範區塊層級輔助說明文字。

勾選我

```
<form role="form">
  <div class="form-group">
    <label for="exampleInputEmail1">電子郵件</label>
    <input type="email" class="form-control" id="exampleInputEmail1"
placeholder="輸入電子郵件">
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">密碼</label>
    <input type="password" class="form-control" id="exampleInputPassword1"
placeholder="Password">
  </div>
  <div class="form-group">
    <label for="exampleInputFile">檔案上傳</label>
    <input type="file" id="exampleInputFile">
    <p class="help-block">在此示範區塊層級輔助說明文字。</p>
  </div>
  <div class="checkbox">
    <label>
      <input type="checkbox" checked=""> 勾選我
    </label>
  </div>
  <button type="submit" class="btn btn-default">送出</button>
</form>
```

不要混合 form 群組與 input 群組

不要直接將 form 群組與 input 群組 (/bs3/Components#input-groups)混用。建議方式是將 input 群組巢狀放入 form 群組之內。

行內表單

將 .form-inline 加入 <form> 元素可以得到靠左對齊且行內區塊 (inline-block) 的控制項。僅套用於可視區域至少 768px 寬度的表單。

譯者註

可視區域小於 768px 寬度時，表單會堆疊方式呈現。

需要自訂寬度

預設 Bootstrap 3 裡的 `Input` 、`select` 和 `textarea` 元素寬度設置為 100%。要使用行內表單，你必須在內部使用的 `form` 控制項上去設定寬度。

譯者註

`.form-inline` 預設寬度為 `width: auto`，基本上是可以將多個控制項在同一行內呈現，但依照佈局上的需求，就必須去自訂寬度。

總是加入 `label`

如果你沒有為每一個輸入類控制項加入 `label` 標籤，那會對螢幕閱讀器造成麻煩。對於這些行內表單，你可以使用 `.sr-only` 類別來隱藏。此外，替代性的做法是提供相關標籤給輔助技術，例如 `aria-label` 、`aria-labelledby` 或 `title`。如果以上都不存在，那麼螢幕閱讀器可能會改使用 `placeholder` 屬性。如果存在，要注意使用 `placeholder` 來替代使用其他標記方法是不妥當的。

EXAMPLE

輸入電子郵件

@ 輸入電子郵件

密碼

記住我

登入

```
<form class="form-inline" role="form">
  <div class="form-group">
    <div class="input-group">
      <label class="sr-only" for="exampleInputEmail2">電子郵件</label>
      <div class="input-group-addon">@</div>
      <input type="email" class="form-control" id="exampleInputEmail2"
placeholder="輸入電子郵件">
    </div>
  </div>
  <div class="form-group">
    <label class="sr-only" for="exampleInputPassword2">密碼</label>
    <input type="password" class="form-control" id="exampleInputPassword2"
placeholder="密碼">
  </div>
  <div class="checkbox">
    <label>
      <input type="checkbox"> 記住我
    </label>
  </div>
  <button type="submit" class="btn btn-default">登入</button>
</form>
```

水平佈局表單

透過為 form 加入 .form-horizontal 類別並且使用 Bootstrap 3 預定義好的格線類別，可以將 label 和 form 控制項進行水平佈局。這會改變 .form-group 的行為，將會以格線系統的 row 來呈現，因此，不需要再加入額外的 .row 設置。

EXAMPLE

電子郵件

輸入電子郵件

密碼

密碼

記住我

登入

```
<form class="form-horizontal" role="form">
  <div class="form-group">
    <label for="inputEmail3" class="col-sm-2 control-label">電子郵件</label>
    <div class="col-sm-10">
      <input type="email" class="form-control" id="inputEmail3" placeholder="輸入電子郵件">
    </div>
  </div>
  <div class="form-group">
    <label for="inputPassword3" class="col-sm-2 control-label">密碼</label>
    <div class="col-sm-10">
      <input type="password" class="form-control" id="inputPassword3" placeholder="密碼">
    </div>
  </div>
  <div class="form-group">
    <div class="col-sm-offset-2 col-sm-10">
      <div class="checkbox">
        <label>
          <input type="checkbox" /> 記住我
        </label>
      </div>
    </div>
  </div>
  <div class="form-group">
    <div class="col-sm-offset-2 col-sm-10">
      <button type="submit" class="btn btn-default">登入</button>
    </div>
  </div>
</form>
```

支援的控制項

form 慄局範例中展示了支援的準標 form 控制項。

input

最常見的 form 控制項、基於文字的 input 欄位，也包含支援所有 HTML5 輸入類型：text、password、datetime、datetime-local、date、month、time、week、number、email、url、search、tel 和 color。

需要類型 (Type) 告警

僅有在正確的宣告其 type 時 input 控制項才能賦予完全正確的樣式。

EXAMPLE

Text input

```
<input type="text" class="form-control" placeholder="Text input">
```

input 群組

如果需要在基於文字的 `<input>` 元素之前或之後整合文字或按鈕的話，請參考 `input` 群組 ([/bs3/Components#input-groups](#))。

textarea

form 控制項支援多行文字輸入的 `<textarea>`。依據需要改變 `row` 屬性。

EXAMPLE

```
Textarea
```

```
<textarea class="form-control" rows="3"></textarea>
```

複選和單選

checkbox 類型是由一組清單中選擇一個或多個，而 radio 類型是由多個選項中選出一個。

checkbox 或 radio 如設置了 `disabled` 屬性都能被賦予合適的樣式。如果 checkbox 或 radio 與 `<label>` 一起使用並且希望當滑鼠移入時一併顯示"不允許"的圖示，將 `.disabled` 類別加至 `.radio`、`.radio-inline`、`.checkbox`、`.checkbox-inline` 類別處，或是加至 `<fieldset>` 元素。

預設(堆疊樣式)

EXAMPLE

Option one is this and that—be sure to include why it's great

Option two is disabled

Option one is this and that—be sure to include why it's great

Option two can be something else and selecting it will deselect option one

Option three is disabled

```

<div class="checkbox">
  <label>
    <input type="checkbox" value="">
    Option one is this and that—be sure to include why it's great
  </label>
</div>
<div class="checkbox disabled">
  <label>
    <input type="checkbox" value="" disabled>
    Option two is disabled
  </label>
</div>

<div class="radio">
  <label>
    <input type="radio" name="optionsRadios" id="optionsRadios1" value="option1" checked>
    Option one is this and that—be sure to include why it's great
  </label>
</div>
<div class="radio">
  <label>
    <input type="radio" name="optionsRadios" id="optionsRadios2" value="option2">
    Option two can be something else and selecting it will deselect option one
  </label>
</div>
<div class="radio disabled">
  <label>
    <input type="radio" name="optionsRadios" id="optionsRadios3" value="option3" disabled>
    Option three is disabled
  </label>
</div>

```

行內 checkbox 和 radio

使用 `.checkbox-inline` 和 `.radio-inline` 類別，可以將一系列的 checkbox 和 radio 控制在同一行上來表現。

EXAMPLE

1 2 3

1 2 3

```
<label class="checkbox-inline">
  <input type="checkbox" id="inlineCheckbox1" value="option1"> 1
</label>
<label class="checkbox-inline">
  <input type="checkbox" id="inlineCheckbox2" value="option2"> 2
</label>
<label class="checkbox-inline">
  <input type="checkbox" id="inlineCheckbox3" value="option3"> 3
</label>

<label class="radio-inline">
  <input type="radio" name="inlineRadioOptions" id="inlineRadio1"
value="option1"> 1
</label>
<label class="radio-inline">
  <input type="radio" name="inlineRadioOptions" id="inlineRadio2"
value="option2"> 2
</label>
<label class="radio-inline">
  <input type="radio" name="inlineRadioOptions" id="inlineRadio3"
value="option3"> 3
</label>
```

無 label 文字的 checkbox 和 radio

如果你未包含任何文字在 `<label>` 之中，`input` 的定位會正如你所期待。目前只適用於非行內的 `checkbox` 和 `radio` 元素。還有，不要忘記仍然為輔助技術提供某種形式的標籤（例如，使用 `aria-label`）。

EXAMPLE

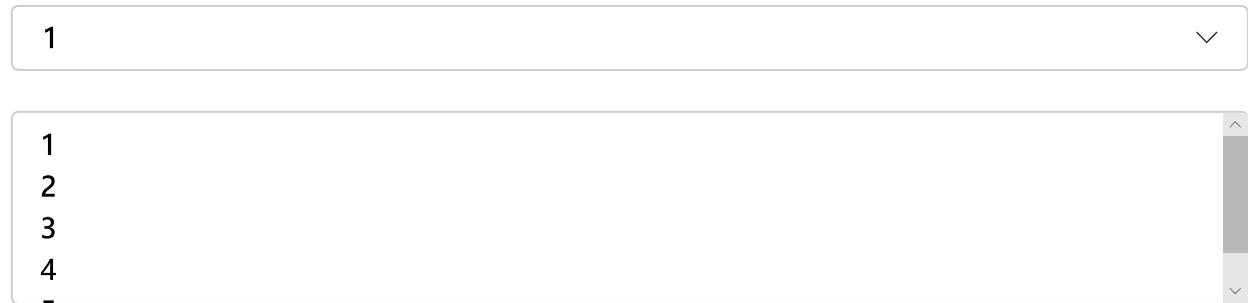


```
<div class="checkbox">
  <label>
    <input type="checkbox" id="blankCheckbox" value="option1" aria-label="..."/>
  </label>
</div>
<div class="radio">
  <label>
    <input type="radio" name="blankRadio" id="blankRadio1" value="option1" aria-label="..."/>
  </label>
</div>
```

下拉選單

使用預設的單選或是加入 `multiple` 以呈現多選。

EXAMPLE



A screenshot of a dropdown menu. The menu is open and displays the number '1'. There are four other options below it: '2', '3', '4', and a separator line '-'. To the right of the menu, there is a vertical scroll bar with a grey track and a white slider.

```
<select class="form-control">
  <option>1</option>
  <option>2</option>
  <option>3</option>
  <option>4</option>
  <option>5</option>
</select>

<select multiple class="form-control">
  <option>1</option>
  <option>2</option>
  <option>3</option>
  <option>4</option>
  <option>5</option>
</select>
```

靜態控制

當你需要在水平佈局的表單中將純文字放置在 `label` 之後，請在 `<p>` 元素加入 `.form-control-static` 類別。

EXAMPLE

Email

email@example.com

Password



A screenshot of a password input field. The field is rectangular with rounded corners and a thin black border. Inside the field, the word 'Password' is written in a small, dark font. The input field is positioned on a light grey background.

```
<form class="form-horizontal" role="form">
  <div class="form-group">
    <label class="col-sm-2 control-label">Email</label>
    <div class="col-sm-10">
      <p class="form-control-static">email@example.com</p>
    </div>
  </div>
  <div class="form-group">
    <label for="inputPassword" class="col-sm-2 control-label">Password</label>
    <div class="col-sm-10">
      <input type="password" class="form-control" id="inputPassword"
placeholder="Password">
    </div>
  </div>
</form>
```

EXAMPLE

email@example.com

Password

Confirm identity

```
<form class="form-inline" role="form">
  <div class="form-group">
    <label class="sr-only">Email</label>
    <p class="form-control-static">email@example.com</p>
  </div>
  <div class="form-group">
    <label for="inputPassword2" class="sr-only">Password</label>
    <input type="password" class="form-control" id="inputPassword2"
placeholder="Password">
  </div>
  <button type="submit" class="btn btn-default">Confirm identity</button>
</form>
```

焦點狀態

我們將某些 form 控制項預設的 outline 樣式移除，然後在 :focus 賦予 box-shadow 屬性。

EXAMPLE

Demonstrative focus state

展示 :focus 狀態

在此文件中，為了展示 .form-control 的 :focus 狀態效果，在上面的範例中的 input 是使用自訂樣式。

禁用狀態

加入 disabled 布林屬性到 input 元素可以預防使用者輸入，並且會觸發一個稍微不同的外觀。

EXAMPLE

Disabled input here...

```
<input class="form-control" id="disabledInput" type="text" placeholder="Disabled input here..." disabled>
```

禁用 fieldset

將 disabled 屬性加入 <fieldset> 元素，那麼就會禁用所有 <fieldset> 之內的控制項。

關於 <a> 連結功能的警告

我們嘗試使用 pointer-events: none 樣式設置去禁用 按鈕樣式的連結功能，但此 CSS 屬性並未成為標準且在 Opera 18 與 Internet Explorer 11 (或更低版本) 並不是完全支援。較安全性的作法是，使用自訂的 JavaScript 去禁用連結的功能。

跨瀏覽器相容性

雖然 Bootstrap 3 會將這些樣式套用到所有瀏覽器，但 Internet Explorer 9 (或更低版本) 實際上並不支援 <fieldset> 元素中設置 disabled 屬性。較安全性的作法是，使用自訂的 JavaScript 去禁用 fieldset。

EXAMPLE

禁用 input

Disabled input

禁用下拉選單

Disabled select



禁用 checkbox

禁用按鈕

```
<form role="form">
  <fieldset disabled>
    <div class="form-group">
      <label for="disabledTextInput">禁用 input</label>
      <input type="text" id="disabledTextInput" class="form-control" placeholder="Disabled input">
    </div>
    <div class="form-group">
      <label for="disabledSelect">禁用下拉選單</label>
      <select id="disabledSelect" class="form-control">
        <option>Disabled select</option>
      </select>
    </div>
    <div class="checkbox">
      <label>
        <input type="checkbox" /> 禁用 checkbox
      </label>
    </div>
    <button type="submit" class="btn btn-primary">禁用按鈕</button>
  </fieldset>
</form>
```

唯讀狀態

在 `input` 元素加入 `readonly` 布林屬性，可預防使用者輸入且樣式是以禁用方式呈現。

EXAMPLE

Readonly input here...

```
<input class="form-control" type="text" placeholder="Readonly input here..." readonly>
```

驗證狀態

Bootstrap 3 的 `form` 控制項含有驗證狀態樣式，例如：錯誤狀態、警告狀態和成功狀態。要使用它們，只需要在父元素上加入 `.has-warning`、`.has-error` 或 `.has-success`。任何 `.control-label`、`.form-control` 和 `.help-block` 內的元素都能接受驗證狀態樣式。

傳達驗證狀態至輔助技術

使用這些驗證狀態樣式來表示 `form` 控制項的狀態僅是提供視覺效果，這不會被傳達至輔助技術的使用者 - 例如：螢幕閱讀器。

請確保還提供了視覺狀態額外的資訊。例如，你可以在 form 控制項的 `<label>` 包含關於狀態本身的提示文字（例如下面程式碼範例的情況），或將其他的元素與 form 控制項使用 `aria-describedby` 關聯與狀態有關的驗證狀態資訊（參考下一節中的範例）。如果有錯誤，你也可以在 form 控制項上使用 `aria-invalid="true"` 屬性。

EXAMPLE

Input with success

Input with warning

Input with error

- Checkbox with success
- Checkbox with warning
- Checkbox with error

```
<div class="form-group has-success">
  <label class="control-label" for="inputSuccess1">Input with success</label>
  <input type="text" class="form-control" id="inputSuccess1">
</div>
<div class="form-group has-warning">
  <label class="control-label" for="inputWarning1">Input with warning</label>
  <input type="text" class="form-control" id="inputWarning1">
</div>
<div class="form-group has-error">
  <label class="control-label" for="inputError1">Input with error</label>
  <input type="text" class="form-control" id="inputError1">
</div>
<div class="has-success">
  <div class="checkbox">
    <label>
      <input type="checkbox" id="checkboxSuccess" value="option1">
      Checkbox with success
    </label>
  </div>
</div>
<div class="has-warning">
  <div class="checkbox">
    <label>
      <input type="checkbox" id="checkboxWarning" value="option1">
      Checkbox with warning
    </label>
  </div>
</div>
<div class="has-error">
  <div class="checkbox">
    <label>
      <input type="checkbox" id="checkboxError" value="option1">
      Checkbox with error
    </label>
  </div>
</div>
```

選擇性圖示

你也可以加入選擇性的回饋圖示 (icon)。只需要設置 .has-feedback 與正確的 Glyphicons 圖示。

回饋圖示僅能正常運作基於文字 `<input class="form-control">` 元素上。

圖示、label 和 input 群組

對於不帶有 `label` 標籤的 `input` 元素和帶有外掛 `input` 群組 ([bs3/Components#input-groups](#))，均需要手動調整回饋圖示的定位。為了無障礙 (accessibility) 原因，我們強烈建議為所有 `input` 元素都加入對應的 `label` 標籤。如果你希望去預防 `label` 被顯示出來，記得使用前面提過的 `sr-only` 類別即可。如果你必須做沒有 `label` 標籤的設計，那麼請調整回饋圖示的 `top` 值。至於 `input` 群組，依照實際環境的寬度調整 `right` 到適當的像素 (pixel) 值。

傳達圖示意義到輔助技術

為確保輔助技術 – 例如，螢幕閱讀器 – 能正確傳達圖示的含意，應加入額外含有 `.sr-only` 類別的隱藏文字並明確與使用到 `aria-describedby` 的 `form` 控制項相關聯。或者確保在其他的表單上有傳達圖示的意義（例如，一個特定文字輸入欄位的警告），例如，改變 `form` 控制項實際關聯的 `<label>` 的文字。

雖然下面的範例在它們各自 `form` 控制項的 `<label>` 都含有驗證狀態的訊息，但上面提到的技術（使用 `.sr-only` 類別的文字和 `aria-describedby`）包含在內是用於說明的目的。

EXAMPLE

Input with success



Input with warning



Input with error



```
<div class="form-group has-success has-feedback">
  <label class="control-label" for="inputSuccess2">Input with success</label>
  <input type="text" class="form-control" id="inputSuccess2" aria-
describedby="inputSuccess2Status">
  <span class="glyphicon glyphicon-ok form-control-feedback" aria-
hidden="true"></span>
  <span id="inputSuccess2Status" class="sr-only">(success)</span>
</div>
<div class="form-group has-warning has-feedback">
  <label class="control-label" for="inputWarning2">Input with warning</label>
  <input type="text" class="form-control" id="inputWarning2" aria-
describedby="inputWarning2Status">
  <span class="glyphicon glyphicon-warning-sign form-control-feedback" aria-
hidden="true"></span>
  <span id="inputWarning2Status" class="sr-only">(warning)</span>
</div>
<div class="form-group has-error has-feedback">
  <label class="control-label" for="inputError2">Input with error</label>
  <input type="text" class="form-control" id="inputError2" aria-
describedby="inputError2Status">
  <span class="glyphicon glyphicon-remove form-control-feedback" aria-
hidden="true"></span>
  <span id="inputError2Status" class="sr-only">(error)</span>
</div>
```

在水平佈局和行內的 forms 設置選擇性圖示

EXAMPLE

Input with success

```
<form class="form-horizontal" role="form">
  <div class="form-group has-success has-feedback">
    <label class="control-label col-sm-3" for="inputSuccess3">Input with
success</label>
    <div class="col-sm-9">
      <input type="text" class="form-control" id="inputSuccess3" aria-
describedby="inputSuccess3Status">
      <span class="glyphicon glyphicon-ok form-control-feedback" aria-
hidden="true"></span>
      <span id="inputSuccess3Status" class="sr-only">(success)</span>
    </div>
  </div>
</form>
```

EXAMPLE

Input with success

```
<form class="form-inline" role="form">
  <div class="form-group has-success has-feedback">
    <label class="control-label" for="inputSuccess4">Input with success</label>
    <input type="text" class="form-control" id="inputSuccess4" aria-
describedby="inputSuccess4Status">
    <span class="glyphicon glyphicon-ok form-control-feedback" aria-
hidden="true"></span>
    <span id="inputSuccess4Status" class="sr-only">(success)</span>
  </div>
</form>
```

隱藏的 .sr-only 標籤與選擇性圖示

加入 .sr-only 類別至 label 元素裡，可以讓 form 控制項裡的 label 不可見。當圖示加入，Bootstrap 3 會自動調整圖示的定位。

EXAMPLE

```
<div class="form-group has-success has-feedback">
  <label class="control-label sr-only" for="inputSuccess5">Hidden label</label>
  <input type="text" class="form-control" id="inputSuccess5" aria-
describedby="inputSuccess5Status">
  <span class="glyphicon glyphicon-ok form-control-feedback" aria-
hidden="true"></span>
  <span id="inputSuccess5Status" class="sr-only">(success)</span>
</div>
```

控制大小

設置高度，使用像是 `.input-lg` 的類別。設置寬度，使用格線 `column` 的類別，例如：`.col-lg-*`。

高度大小

建立較高或較短的 `form` 控制項，以符合按鈕大小。

EXAMPLE

`.input-lg`

Default input

`.input-sm`

`.input-lg`

Default select

`.input-sm`

```
<input class="form-control input-lg" type="text" placeholder=".input-lg">
<input class="form-control" type="text" placeholder="Default input">
<input class="form-control input-sm" type="text" placeholder=".input-sm">

<select class="form-control input-lg">...</select>
<select class="form-control">...</select>
<select class="form-control input-sm">...</select>
```

譯者註

這裡的短指上下距離。

水平佈局 form 群組大小

為 `.form-horizontal` 加入 `.form-group-lg` 或 `.form-group-sm` 設置，可以快速調整 label 和 form 控制項的大小。

EXAMPLE

Large label

Large input

Small label

Small input

```
<form class="form-horizontal" role="form">
  <div class="form-group form-group-lg">
    <label class="col-sm-2 control-label" for="formGroupInputLarge">Large
    label</label>
    <div class="col-sm-10">
      <input class="form-control" type="text" id="formGroupInputLarge"
      placeholder="Large input">
    </div>
  </div>
  <div class="form-group form-group-sm">
    <label class="col-sm-2 control-label" for="formGroupInputSmall">Small
    label</label>
    <div class="col-sm-10">
      <input class="form-control" type="text" id="formGroupInputSmall"
      placeholder="Small input">
    </div>
  </div>
</form>
```

column 大小

把 `input` 元素包裝進格線 `column` 或任何父元素之中，就能輕鬆設置期待的寬度。

EXAMPLE

`.col-xs-`

`.col-xs-3`

`.col-xs-4`

```
<div class="row">
  <div class="col-xs-2">
    <input type="text" class="form-control" placeholder=".col-xs-2">
  </div>
  <div class="col-xs-3">
    <input type="text" class="form-control" placeholder=".col-xs-3">
  </div>
  <div class="col-xs-4">
    <input type="text" class="form-control" placeholder=".col-xs-4">
  </div>
</div>
```

輔助文字

在 form 控制項裡以區塊層級呈現輔助文字。

與 form 控制項關聯的輔助文字

輔助文字應該明確與它使用的 `aria-describedby` 屬性的 form 控制項相關聯 - 這能確保輔助技術 - 例如：螢幕閱讀器 - 在使用者取得焦點或進入 form 控制項時宣讀輔助文字。

EXAMPLE

Input with help text

A block of help text that breaks onto a new line and may extend beyond one line.

```
<label class="sr-only" for="inputHelpBlock">Input with help text</label>
<input type="text" id="inputHelpBlock" class="form-control" aria-
describedby="helpBlock">
...
<span id="helpBlock" class="help-block">A block of help text that breaks onto a
new line and may extend beyond one line.</span>
```

按鈕

選項

使用以下任何的按鈕類別，可以快速建立具有樣式的按鈕。

EXAMPLE

Default

Primary

Success

Info

Warning

Danger

Link

<!-- 標準按鈕 -->

```
<button type="button" class="btn btn-default">Default</button>
```

<!-- 提供額外視覺上的重量和識別一組按鈕中主要的操作項目 -->

```
<button type="button" class="btn btn-primary">Primary</button>
```

<!-- 指出成功或積極的操作 -->

```
<button type="button" class="btn btn-success">Success</button>
```

<!-- 訊息提示方面的操作 -->

```
<button type="button" class="btn btn-info">Info</button>
```

<!-- 指出應謹慎採取此一行動 -->

```
<button type="button" class="btn btn-warning">Warning</button>
```

<!-- 指出危險或潛在負面作用的行動 -->

```
<button type="button" class="btn btn-danger">Danger</button>
```

<!-- 淡化一個按鈕，使它看起來像是一個連結並同時保持按鈕行為 -->

```
<button type="button" class="btn btn-link">Link</button>
```

傳達意義到輔助技術

使用顏色加入到按鈕僅是提供視覺效果上的含義，這不會被傳達至輔助技術的使用者 - 例如：螢幕閱讀器。請確保顏色所表示的訊息從內容本身上是顯而易見的（在相關按鈕可見的文字）・或是通過其他的方式來傳達，例如，額外 .sr-only 類別的隱藏文字。

大小

要想更大或更小的按鈕嗎？加入 .btn-lg 、 .btn-sm 或 .btn-xs 可以取得其他的大小。

EXAMPLE

Large button

Large button

Default button

Default button

Small button

Small button

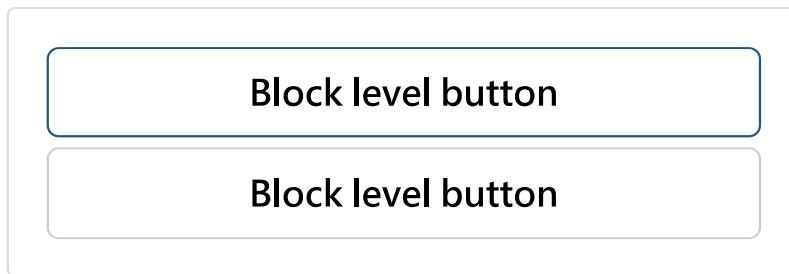
Extra small button

Extra small button

```
<p>
  <button type="button" class="btn btn-primary btn-lg">Large button</button>
  <button type="button" class="btn btn-default btn-lg">Large button</button>
</p>
<p>
  <button type="button" class="btn btn-primary">Default button</button>
  <button type="button" class="btn btn-default">Default button</button>
</p>
<p>
  <button type="button" class="btn btn-primary btn-sm">Small button</button>
  <button type="button" class="btn btn-default btn-sm">Small button</button>
</p>
<p>
  <button type="button" class="btn btn-primary btn-xs">Extra small
button</button>
  <button type="button" class="btn btn-default btn-xs">Extra small
button</button>
</p>
```

透過加入 `.btn-block` 類別，可以建立一個區塊層級的按鈕 — 它可以延伸至父元素的 100% 寬度。

EXAMPLE



```
<button type="button" class="btn btn-primary btn-lg btn-block">Block level
button</button>
<button type="button" class="btn btn-default btn-lg btn-block">Block level
button</button>
```

啟用狀態

當按鈕是啟用狀態，那麼會以被壓下去的效果來表現（更深的背景，更深的邊框、向內的陰影）。對於 `<button>` 元素，這是透過 `:active` 來完成。對於 `<a>` 元素，它是透過 `.active` 類別來完成。無論如何，你也能使用程式碼方式來處理啟用狀態，在 `<button>` 套用 `.active` 類別。

按鈕元素

無需額外加入 `:active`，因為它是一個偽類別（pseudo-class），但是，如果你需要要求一致的外觀，加入 `.active` 類別。

EXAMPLE

Primary button

Button

```
<button type="button" class="btn btn-primary btn-lg active">Primary  
button</button>  
<button type="button" class="btn btn-default btn-lg active">Button</button>
```

連結元素

加入 `.active` 類別到 `<a>` 元素中，可以讓 `<a>` 元素以按鈕方式呈現。

EXAMPLE

Primary link

Link

```
<a href="#" class="btn btn-primary btn-lg active" role="button">Primary link</a>  
<a href="#" class="btn btn-default btn-lg active" role="button">Link</a>
```

禁用狀態

透過將按鈕的背景顏色減少 `opacity` (透明度)，讓按鈕看起來無法點擊。

按鈕元素

加入 `disabled` 屬性到 `<button>` 元素中。

EXAMPLE

Primary button

Button

```
<button type="button" class="btn btn-lg btn-primary" disabled="disabled">Primary  
button</button>  
<button type="button" class="btn btn-default btn-lg"  
disabled="disabled">Button</button>
```

跨瀏覽器相容性

如果你將 `disabled` 屬性加到 `<button>` 元素，Internet Explorer 9 (含以下版本) 將會文字以灰色並帶有討厭的文字陰影，目前我們還無法解決。

連結元素

加入 `disabled` 屬性到 `<a>` 按鈕元素中。

EXAMPLE

Primary link

Link

```
<a href="#" class="btn btn-primary btn-lg disabled" role="button">Primary  
link</a>  
<a href="#" class="btn btn-default btn-lg disabled" role="button">Link</a>
```

在此，我們把 `.disabled` 類別當成工具類別使用，如同 `.active` 類別一樣，因此前綴不是必要的。

連結功能警告

`.disabled` 類別嘗試使用 `pointer-events: none` 樣式設置去禁用 `<a>` 的連結功能，但此 CSS 屬性並未成為標準且在 Opera 18 與 Internet Explorer 11 (或更低版本) 並不是完全支援。此外，即使瀏覽器支援 `pointer-events: none`，鍵盤的巡覽依然不影響，也就是說，使用鍵盤和輔助技術的使用者仍然能夠啟用這些連結。較安全性的作法是，使用自訂的 JavaScript 去禁用連結的功能。

特定情境的用法

雖然按鈕類別可以套用至 `<a>` 和 `<button>` 元素上，但僅有 `<button>` 元素是被巡覽列和巡覽元件所支援。

按鈕標籤

在 `<a>`、`<button>` 或 `<input>` 元素套用按鈕類別。

EXAMPLE

Link

Button

Input

Submit

```
<a class="btn btn-default" href="#" role="button">Link</a>  
<button class="btn btn-default" type="submit">Button</button>  
<input class="btn btn-default" type="button" value="Input">  
<input class="btn btn-default" type="submit" value="Submit">
```

作為按鈕的連結

如果 `<a>` 元素被用來充當按鈕 - 觸發頁面內的功能，而不是巡覽至其他文件或當前頁面的其

他章節 - 那他們應該給於一個合適的 `role="button"` 設置。

跨瀏覽器呈現

作為最佳實踐，我們強烈建議盡可能的使用 `<button>` 元素以確保在跨瀏覽器上能有一致的呈現。

此外，我們還發現了一個 Firefox <30 (https://bugzilla.mozilla.org/show_bug.cgi?id=697451) 版本上的臭蟲，它會阻止我們基於 `<input>` 元素建立的按鈕所設置的 `line-height` 屬性，導致它們不完全符合 Firefox <30 版本中與其他按鈕一致的高度。

圖片

自適應圖片

在 Bootstrap 3 中，透過加入 `.img-responsive` 類別，可以讓圖片擁有良好的自適應功能。這僅套用 `max-width: 100%;` 和 `height: auto;` 屬性到圖片上，讓圖片可以在父元素中進行更好效果的縮放。

SVG 圖片 和 IE 8-10

在 Internet Explorer 8-10，設置 `.img-responsive` 類別的 SVG 圖片呈現出來會有不對稱的大小問題。要修正此問題，在出現問題的地方加入 `width: 100% \9;` 即可。Bootstrap 3 並未自動套用上述屬性，因為它會造成其他圖片格式的混亂。

```

```

圖片形狀

為 `` 元素加入以下類別，可以很輕鬆的讓圖片以不同樣式來呈現。

跨瀏覽器相容性

牢記此事，Internet Explorer 8 不支援 CSS3 的圓角屬性。

EXAMPLE



```



```

輔助類別

情境顏色

透過重點工具類別使用少數的顏色來傳達含義。這些類別也能套用至連結並會讓移入效果 (hover) 變的較暗，就像預設的連結樣式。

EXAMPLE

Fusce dapibus, tellus ac cursus commodo, tortor mauris nibh.
Nullam id dolor id nibh ultricies vehicula ut id elit.
Duis mollis, est non commodo luctus, nisi erat porttitor ligula.
Maecenas sed diam eget risus varius blandit sit amet non magna.
Etiam porta sem malesuada magna mollis euismod.
Donec ullamcorper nulla non metus auctor fringilla.

```
<p class="text-muted">...</p>
<p class="text-primary">...</p>
<p class="text-success">...</p>
<p class="text-info">...</p>
<p class="text-warning">...</p>
<p class="text-danger">...</p>
```

特別處理

有些情況，情境重點文字類別無法套至特別的其他選擇器上。多數情況下，一個不錯的變通方

法是在類別裡用 `` 包裝你的文字。

傳達意義到輔助技術

使用顏色僅是提供視覺效果上的含義，這不會被傳達至輔助技術的使用者 - 例如：螢幕閱讀器。請確保顏色所表示的訊息從內容本身上是顯而易見的（情境顏色只是用於加強已存在的文字或標記的含義），或是通過其他的方式來傳達，例如，額外 `.sr-only` 類別的隱藏文字。

情境背景

類似於情境文字顏色類別，透過任何情境類別都能輕鬆地設置元素的背景。連結元件在移入效果會變較暗，這與文字類別相同。

EXAMPLE

`Nullam id dolor id nibh ultricies vehicula ut id elit.`

`Duis mollis, est non commodo luctus, nisi erat porttitor ligula.`

`Maecenas sed diam eget risus varius blandit sit amet non magna.`

`Etiam porta sem malesuada magna mollis euismod.`

`Donec ullamcorper nulla non metus auctor fringilla.`

```
<p class="bg-primary">...</p>
<p class="bg-success">...</p>
<p class="bg-info">...</p>
<p class="bg-warning">...</p>
<p class="bg-danger">...</p>
```

特別處理

有些情況，情境重點文字類別無法套至特別的其他選擇器上。多數情況下，一個不錯的變通方法是在類別的 `` 包裝你的文字。

傳達意義到輔助技術

與情境顏色相同，確保顏色意義上傳達並也傳達你的整體安排，而不只是一個純粹的表象。

關閉圖示

使用一般關閉圖示來消除內容，像是互動視窗（modal）或警告視窗（alert）。

EXAMPLE

×

```
<button type="button" class="close"><span aria-hidden="true">&times;</span><span class="sr-only">Close</span></button>
```

三角括號

使用三角括號來指出此處有下拉選單的功能與選單方向。注意，預設三角括號在向上選單（/bs3/Components#btn-dropdowns-dropup）會自動以向上三角型呈現。

EXAMPLE

▼

```
<span class="caret"></span>
```

快速浮動

透過類別可以快速讓元素向左或向右浮動。!important 包含在類別設置之中以避免特別的問題。類別也能進行混合使用。

```
<div class="pull-left">...</div>
<div class="pull-right">...</div>
```

```
// Classes
.pull-left {
  float: left !important;
}
.pull-right {
  float: right !important;
}

// Usage as mixins
.element {
  .pull-left();
}
.another-element {
  .pull-right();
}
```

不要在巡覽列使用

在巡覽列要使用工具類別來對齊元件，使用 `.navbar-left` 或 `.navbar-right` 來替代。更多資訊請參考巡覽列文件 ([/bs3/Components#navbar-component-alignment](#))。

置中內容區塊

透過設置元素 `display: block` 和 `margin`，將會在置中位置且以區塊模式呈現。可作為 Less 的 `mixin` 和類別。

```
<div class="center-block">...</div>
```

```
// 作為類別
.center-block {
  display: block;
  margin-left: auto;
  margin-right: auto;
}

// 作為 mixin
.element {
  .center-block();
}
```

清除浮動

在父元素上加入 `.clearfix` 就能輕鬆清除 `float` 效果。這裡普遍使用 Nicolas Gallagher 的 micro clearfix (<http://nicolasgallagher.com/micro-clearfix-hack/>) 方法。也可使用在 Less 的 `mixin`。

```
<!-- 作為類別使用 -->
<div class="clearfix">...</div>
```

```
// Mixin本身
.clearfix() {
  &:before,
  &:after {
    content: " ";
    display: table;
  }
  &:after {
    clear: both;
  }
}

// 作為 Mixin
.element {
  .clearfix();
}
```

顯示和隱藏內容

使用 `.show` 和 `.hidden` 類別，可以讓元素顯示或隱藏（包含螢幕閱讀器）。這些類別使用 `!important` 以避免不必要的衝突，像是快速浮動。他們僅在區塊層級進行切換時有效。也可使用在 Less 的 mixin。

`.hide` 還是可以使用，但它在螢幕閱讀器不總是有效果，而且從 v3.0.1 開始不推薦使用。建議使用 `.hidden` 或 `.sr-only` 替代。

此外，`.invisible` 類別僅能針對可見的元素進行隱藏，意思是，元素的 `display` 屬性不會有任何異動，而且元素在 DOM 內仍然有浮動及佈局上的效果。

```
<div class="show">...</div>
<div class="hidden">...</div>
```

```
// 類別的用法
.show {
  display: block !important;
}
.hidden {
  display: none !important;
  visibility: hidden !important;
}
.invisible {
  visibility: hidden;
}

// 作為 mixin
.element {
  .show();
}
.another-element {
  .hidden();
}
```

螢幕閱讀器和鍵盤巡覽內容

使用 `.sr-only` 類別，可以對除了螢幕閱讀器以外的所有設備隱藏元素。結合使用 `.sr-only` 和 `.sr-only-focusable` 類別，當它取得焦點時，再一次顯示元素（例如，僅能使用鍵盤的使用者）。請參考無障礙最佳做法 (/bs3/GettingStarted#accessibility)。也可使用在 Less 的 mixin。

```
<a class="sr-only sr-only-focusable" href="#content">Skip to main
content</a>
```

```
// Usage as a Mixin
.skip-navigation {
  .sr-only();
  .sr-only-focusable();
}
```

圖片替換

.text-hide 工具類別或 mixin，幫助替換元素的文字內容與背景圖片。

```
<h1 class="text-hide">Custom heading</h1>
```

```
// 作為 mixin
.heading {
  .text-hide();
}
```

自適應工具類別

為了快速進行行動友善的開發，使用以下的工具類別，能依照設備的可視區域與媒體查詢來顯示或隱藏內容。當列印時，工具類別也能輔助進行內容佈局（顯示或隱藏）。

嘗試使用這些類別在有限的基礎上，並避免建立在同一網站上建立完全不同的版本。相反的，使用他們來輔助每一種設備的呈現。

可用類別

透過單獨使用或組合這些可用類別，可以針對不同可視區域進行內容的顯示或隱藏切換。

	超小螢幕設備 手機 (<768px)	小螢幕設備 平板 ($\geq 768px$)	中螢幕設備 桌面 ($\geq 992px$)	大螢幕設備 大桌面 ($\geq 1200px$)
.visible-xs-*	顯示	隱藏	隱藏	隱藏
.visible-sm-*	隱藏	顯示	隱藏	隱藏
.visible-md-*	隱藏	隱藏	顯示	隱藏
.visible-lg-*	隱藏	隱藏	隱藏	顯示
.hidden-xs	隱藏	顯示	顯示	顯示
.hidden-sm	顯示	隱藏	顯示	顯示
.hidden-md	顯示	顯示	隱藏	顯示
.hidden-lg	顯示	顯示	顯示	隱藏

從 v3.2.0 開始，.visible-*-* 的類別在每一個可視區域的判斷點上都有三種變化，每一種變化都是針對 CSS 的 display 屬性，列表如下：

類別群組	CSS display 值
.visible-*-block	display: block;
.visible-*-inline	display: inline;
.visible-*-inline-block	display: inline-block;

以超小螢幕 (xs) 設備為例，可用的 .visible-*-* 類別有：.visible-xs-block 、.visible-xs-inline 和 .visible-xs-inline-block 。

.visible-xs 、.visible-sm 、.visible-md 和 .visible-lg 類別依然存在，但從 v3.2.0 開始不再推薦使用。除了切換 <table> 相關元素的特別情況下，它們大致上與 .visible-*-block 類別相同。

列印類別

與正規的自適應類別相似，使用以下類別在列印時進行內容的切換。

類別	瀏覽器	列印
.visible-print-block .visible-print-inline .visible-print-inline-block	隱藏	顯示
.hidden-print	顯示	隱藏

.visible-print 類別依然存在，但從 v3.2.0 開始不再推薦使用。除了切換 <table> 相關元素的特別情況下，它們大致上與 .visible-print-block 類別相同。

測試情境

調整你的瀏覽器大小或在不同設備載入頁面，以測試這些自適應工具類別。

在...可見

帶有綠色勾選標記的元素是指出你目前當下可視區域會被顯示的區域。

✓ 在超小螢幕顯示

小螢幕

中螢幕

大螢幕

✓ 在超小螢幕和小螢幕顯示

中螢幕和大螢幕

✓ 在超小螢幕中螢幕顯示

小螢幕和大螢幕

✓ 在超小螢幕和大螢幕顯示

小螢幕和中螢幕

在...隱藏

這裡，帶有綠色勾選標記的元素是指出你目前當下可視區域會被隱藏的區域。

✓ 在超小螢幕隱藏

小螢幕

中螢幕

大螢幕

✓ 在超小螢幕和小螢幕隱藏

中螢幕和大螢幕

✓ 在超小螢幕和中螢幕隱藏

小螢幕和大螢幕

✓ 在超小螢幕和大螢幕隱藏

小螢幕和中螢幕

使用 Less

Bootstrap 3 的 CSS 是透過 Less 建置的，Less 是一個預處理器，額外提供像是變數、mixin 和函數等功能以編譯出 CSS。那些希望使用 Less 原始碼來替代已編譯 CSS 檔案開發者，Bootstrap 3 框架含有大量的變數和 mixin 可供使用。

格線變數和 mixin 已含在格線系統小節。

編譯 Bootstrap 3

Bootstrap 3 至少有兩種使用方式：使用已編譯 CSS 檔案或使用 Less 原始碼。要使用 Less 原始碼進行編譯，請參考旅程開始的編譯 CSS 和 JavaScript (/bs3/GettingStarted#grunt) 小節，以瞭解如何設置你的開發環境和執行所需的指令。

第三方編譯工具也許能編譯 Bootstrap 3，但它們不是核心團隊所支援的對象。

變數

變數用於整個專案中，以此來集中與共享通用的值，像是顏色、間距、字型等。完整的分類，請參考客製化 (<http://getbootstrap.com/customize/#less-variables-section>)一節。

顏色

輕鬆使用兩種顏色方案：灰階和表達語意。灰階顏色提供快速存取常用的黑色陰影。而表達語意含有各種顏色，分配給有意義的情境值。

EXAMPLE

```
@gray-darker:  lighten(#000, 13.5%); // #222  
@gray-dark:    lighten(#000, 20%);   // #333  
@gray:         lighten(#000, 33.5%); // #555  
@gray-light:   lighten(#000, 46.7%); // #777  
@gray-lighter: lighten(#000, 93.5%); // #eee
```

EXAMPLE

```
@brand-primary: #428bca;  
@brand-success: #5cb85c;  
@brand-info:    #5bc0de;  
@brand-warning: #f0ad4e;  
@brand-danger:  #d9534f;
```

可以使用以上任何顏色變數或在你的專案中重新分配它們到更有意義的變數。

```
// 照常使用  
.masthead {  
  background-color: @brand-primary;  
}  
  
// 在 Less 重新分配變數  
@alert-message-background: @brand-info;  
.alert {  
  background-color: @alert-message-background;  
}
```

基架

少數的變數來快速自訂網站的骨架中的關鍵元素。

```
// Scaffolding  
@body-bg:    #fff;  
@text-color: @black-50;
```

連結

賦予你的連結一個輕鬆的樣式，合適的顏色並且僅有一個值。

```
// 變數
@link-color:      @brand-primary;
@link-hover-color: darken(@link-color, 15%);

// Usage
a {
  color: @link-color;
  text-decoration: none;

  &:hover {
    color: @link-hover-color;
    text-decoration: underline;
  }
}
```

注意 `@link-hover-color` 使用的功能。Less 裡另一個很棒的工具，能自動正確建立滑鼠移入時的顏色。你可以使用 `darken`、`lighten`、`saturate` 和 `desaturate`。

排版

輕鬆設置你的版型，文字大小、行距和更多的快速變數。Bootstrap 3 使用這些內容以及提供容易排版的 mixin。

```
@font-family-sans-serif: "Helvetica Neue", Helvetica, Arial, sans-serif;
@font-family-serif:      Georgia, "Times New Roman", Times, serif;
@font-family-monospace:  Menlo, Monaco, Consolas, "Courier New", monospace;
@font-family-base:       @font-family-sans-serif;

@font-size-base:          14px;
@font-size-large:         ceil(@font-size-base * 1.25)); // ~18px
@font-size-small:         ceil(@font-size-base * 0.85)); // ~12px

@font-size-h1:             floor(@font-size-base * 2.6)); // ~36px
@font-size-h2:             floor(@font-size-base * 2.15)); // ~30px
@font-size-h3:             ceil(@font-size-base * 1.7)); // ~24px
@font-size-h4:             ceil(@font-size-base * 1.25)); // ~18px
@font-size-h5:             @font-size-base;
@font-size-h6:             ceil(@font-size-base * 0.85)); // ~12px

@line-height-base:        1.428571429; // 20/14
@line-height-computed:   floor(@font-size-base * @line-height-base)); // ~20px

@headings-font-family:   inherit;
@headings-font-weight:   500;
@headings-line-height:   1.1;
@headings-color:         inherit;
```

圖示

兩個快速變數用於自定義你的圖示的位置和檔案名稱。

```
@icon-font-path:      ".../fonts/";  
@icon-font-name:     "glyphicon-halflings-regular";
```

元件

在 Bootstrap 3 的整體元件中，利用一些預設變數設置一些通用的值。以下是最常使用的配置。

```
@padding-base-vertical:      6px;  
@padding-base-horizontal:    12px;  
  
@padding-large-vertical:     10px;  
@padding-large-horizontal:   16px;  
  
@padding-small-vertical:     5px;  
@padding-small-horizontal:   10px;  
  
@padding-xs-vertical:        1px;  
@padding-xs-horizontal:      5px;  
  
@line-height-large:          1.33;  
@line-height-small:          1.5;  
  
@border-radius-base:         4px;  
@border-radius-large:        6px;  
@border-radius-small:        3px;  
  
@component-active-color:     #fff;  
@component-active-bg:        @brand-primary;  
  
@caret-width-base:           4px;  
@caret-width-large:          5px;
```

瀏覽器供應商 mixin

瀏覽器供應商 mixin 是 mixin，以協助支援多個瀏覽器在編譯 CSS 時能包含相關的瀏覽器供應商前綴（vendor prefixes）。

Box 大小

在單獨的 mixin 中重新設置你的元件 box 模型（box model）。更多細節，請參考 Mozilla 的 box-sizing 文件 (<https://developer.mozilla.org/en-US/docs/CSS/box-sizing>)。

從 v3.2.0 版開始此 mixin 不再推薦，改採用 autoprefixer。為了保持向後相容，Bootstrap 3 將會持續在 mixin 內部使用直到 Bootstrap v4。

```
.box-sizing(@box-model) {  
    -webkit-box-sizing: @box-model; // Safari <= 5  
    -moz-box-sizing: @box-model; // Firefox <= 19  
    box-sizing: @box-model;  
}
```

圓角

今日，所有主流瀏覽器都支援不必加瀏覽器供應商前綴的 `border-radius` 屬性。因此，就不再提供 `.border-radius` 的 mixin，但 Bootstrap 3 含有在物件的特定面快速設置兩個角的方式。

```
.border-top-radius(@radius) {  
    border-top-right-radius: @radius;  
    border-top-left-radius: @radius;  
}  
.border-right-radius(@radius) {  
    border-bottom-right-radius: @radius;  
    border-top-right-radius: @radius;  
}  
.border-bottom-radius(@radius) {  
    border-bottom-right-radius: @radius;  
    border-bottom-left-radius: @radius;  
}  
.border-left-radius(@radius) {  
    border-bottom-left-radius: @radius;  
    border-top-left-radius: @radius;  
}
```

Box 陰影

如果你的目標使用者是使用最新最好的瀏覽器和設備，請確保僅使用 `box-shadow` 屬性本身。如果你需要支援到舊 Android (v4 之前) 和 iOS 設備 (iOS 5 之前)，從不再推薦 mixin 中選取所需的 `-webkit` 首綴。

此 mixin 從 v3.1.0 開始不再推薦。因為，Bootstrap 3 不正式支援過時平台所不支援的標準屬性。為了保持向後相容，Bootstrap 3 將會持續在 mixin 內部使用直到 Bootstrap v4。

請務必在 box 陰影使用 `rgba()` 的顏色設置，這樣能盡可能無縫的融入背景之中。

```
.box-shadow(@shadow: 0 1px 3px rgba(0,0,0,.25)) {  
    -webkit-box-shadow: @shadow; // iOS <4.3 & Android <4.1  
    box-shadow: @shadow;  
}
```

轉場效果

多個 mixin 的靈活性。在一處設置所有轉場資訊，或依據需求單獨指定延遲和持續時間。

從 v3.2.0 版開始此 mixin 不再推薦，改採用 autoprefixer。為了保持向後相容，Bootstrap 3 將會持續在 mixin 內部使用直到 Bootstrap v4。

```
.transition(@transition) {
  -webkit-transition: @transition;
  transition: @transition;
}
.transition-property(@transition-property) {
  -webkit-transition-property: @transition-property;
  transition-property: @transition-property;
}
.transition-delay(@transition-delay) {
  -webkit-transition-delay: @transition-delay;
  transition-delay: @transition-delay;
}
.transition-duration(@transition-duration) {
  -webkit-transition-duration: @transition-duration;
  transition-duration: @transition-duration;
}
.transition-timing-function(@timing-function) {
  -webkit-transition-timing-function: @timing-function;
  transition-timing-function: @timing-function;
}
.transition-transform(@transition) {
  -webkit-transition: -webkit-transform @transition;
  -moz-transition: -moz-transform @transition;
  -o-transition: -o-transform @transition;
  transition: transform @transition;
}
```

轉換效果

旋轉、縮放、平移或傾斜任何物件。

從 v3.2.0 版開始此 mixin 不再推薦，改採用 autoprefixer。為了保持向後相容，Bootstrap 3 將會持續在 mixin 內部使用直到 Bootstrap v4。

```

.rotate(@degrees) {
  -webkit-transform: rotate(@degrees);
  -ms-transform: rotate(@degrees); // IE9 only
  transform: rotate(@degrees);
}

.scale(@ratio; @ratio-y...) {
  -webkit-transform: scale(@ratio, @ratio-y);
  -ms-transform: scale(@ratio, @ratio-y); // IE9 only
  transform: scale(@ratio, @ratio-y);
}

.translate(@x; @y) {
  -webkit-transform: translate(@x, @y);
  -ms-transform: translate(@x, @y); // IE9 only
  transform: translate(@x, @y);
}

.skew(@x; @y) {
  -webkit-transform: skew(@x, @y);
  -ms-transform: skewX(@x) skewY(@y); // See
https://github.com/twbs/bootstrap/issues/4885; IE9+
  transform: skew(@x, @y);
}

.translate3d(@x; @y; @z) {
  -webkit-transform: translate3d(@x, @y, @z);
  transform: translate3d(@x, @y, @z);
}

.rotateX(@degrees) {
  -webkit-transform: rotateX(@degrees);
  -ms-transform: rotateX(@degrees); // IE9 only
  transform: rotateX(@degrees);
}

.rotateY(@degrees) {
  -webkit-transform: rotateY(@degrees);
  -ms-transform: rotateY(@degrees); // IE9 only
  transform: rotateY(@degrees);
}

.perspective(@perspective) {
  -webkit-perspective: @perspective;
  -moz-perspective: @perspective;
  perspective: @perspective;
}

.perspective-origin(@perspective) {
  -webkit-perspective-origin: @perspective;
  -moz-perspective-origin: @perspective;
  perspective-origin: @perspective;
}

.transform-origin(@origin) {
  -webkit-transform-origin: @origin;
  -moz-transform-origin: @origin;
  -ms-transform-origin: @origin; // IE9 only
  transform-origin: @origin;
}

```

動畫效果

在單一個 mixin 的一個宣告中使用所有 CSS3 的動畫屬性和其他 mixin 的單獨屬性。

從 v3.2.0 版開始此 mixin 不再推薦，改採用 autoprefixer。為了保持向後相容，Bootstrap 3 將會持續在 mixin 內部使用直到 Bootstrap v4。

```
.animation(@animation) {
  -webkit-animation: @animation;
  animation: @animation;
}
.animation-name(@name) {
  -webkit-animation-name: @name;
  animation-name: @name;
}
.animation-duration(@duration) {
  -webkit-animation-duration: @duration;
  animation-duration: @duration;
}
.animation-timing-function(@timing-function) {
  -webkit-animation-timing-function: @timing-function;
  animation-timing-function: @timing-function;
}
.animation-delay(@delay) {
  -webkit-animation-delay: @delay;
  animation-delay: @delay;
}
.animation-iteration-count(@iteration-count) {
  -webkit-animation-iteration-count: @iteration-count;
  animation-iteration-count: @iteration-count;
}
.animation-direction(@direction) {
  -webkit-animation-direction: @direction;
  animation-direction: @direction;
}
```

透明度

為所有瀏覽器設定透明度（ opacity ），並提供 filter 退路給 IE8 。

```
.opacity(@opacity) {
  opacity: @opacity;
  // IE8 filter
  @opacity-ie: (@opacity * 100);
  filter: ~"alpha(opacity=@{opacity-ie})";
}
```

Placeholder 文字

為 form 控制項內每個欄位提供相關設置。

```
.placeholder(@color: @input-color-placeholder) {  
  &:-moz-placeholder { color: @color; } // Firefox  
  &:-ms-input-placeholder { color: @color; } // Internet Explorer 10+  
  &:-webkit-input-placeholder { color: @color; } // Safari and Chrome  
}
```

Column

在單一元素內透過 CSS 產生 column。

```
.content-columns(@width; @count; @gap) {  
  -webkit-column-width: @width;  
  -moz-column-width: @width;  
  column-width: @width;  
  -webkit-column-count: @count;  
  -moz-column-count: @count;  
  column-count: @count;  
  -webkit-column-gap: @gap;  
  -moz-column-gap: @gap;  
  column-gap: @gap;  
}
```

漸層

輕鬆地把任何兩種顏色做背景漸層。更進階的應用如設定方向、使用三種顏色或放射狀的漸層。在單一 mixin 你可以得到所需的所有前綴語法。

```
#gradient > .vertical(#333; #000);  
#gradient > .horizontal(#333; #000);  
#gradient > .radial(#333; #000);
```

你還可以指定標準的兩種顏色做角度的線性漸層：

```
#gradient > .directional(#333; #000; 45deg);
```

如果你需要像霓虹燈般條紋樣式的漸層，那也很簡單。僅需要指定一個單一的顏色，我們會疊加一個半透明的白色條紋。

```
#gradient > .striped(#333; 45deg);
```

加大賭注，改用三種顏色。在 mixin 設置第一個顏色，然後第二個顏色，之後設定一個百分比，比如 25%，接下來第三個顏色：

```
#gradient > .vertical-three-colors(#777; #333; 25%; #000);  
#gradient > .horizontal-three-colors(#777; #333; 25%; #000);
```

當心！如果你曾經需要刪除漸層，一定要確保刪除你可能加入的任何特定 IE 的 filter。你可以在 background-image: none; 之後使用 .reset-filter() mixin。

工具類 mixin

工具類 mixin 是 mixin 結合原本不同的 CSS 屬性來實現特定目標或任務。

清除浮動

忘記加入 `class="clearfix"` 到任何元素和改為在合適地方加入 `.clearfix()` mixin。使用 Nicolas Gallager (<http://twitter.com/necolas>) 的 micro clearfix (<http://nicolasgallagher.com/micro-clearfix-hack/>) 方法。

```
// Mixin
.clearfix() {
  &:before,
  &:after {
    content: " ";
    display: table;
  }
  &:after {
    clear: both;
  }
}

// Usage
.container {
  .clearfix();
}
```

水平置中

任何元素都能快速的以父元素為中心點對齊。需要對 `width` 或 `max-width` 進行設置。

```
// Mixin
.center-block() {
  display: block;
  margin-left: auto;
  margin-right: auto;
}

// Usage
.container {
  width: 940px;
  .center-block();
}
```

大小輔助

更輕鬆地指定物件的大小。

```
// Mixins
.size(@width; @height) {
  width: @width;
  height: @height;
}
.square(@size) {
  .size(@size; @size);
}

// Usage
.image { .size(400px; 300px); }
.avatar { .square(48px); }
```

可調大小的 textarea

輕鬆的為任何 textarea 或其他任何元素進行重調大小的組態。預設為正常瀏覽器的行為 (both) 。

```
.resizable(@direction: both) {
  // Options: horizontal, vertical, both
  resize: @direction;
  // Safari fix
  overflow: auto;
}
```

截斷文字

在單一 mixin，輕鬆在有省略符號 (ellipsis) 的地方截斷文字。元素必須是 `block` 或 `inline-block` 層級。

```
// Mixin
.text-overflow() {
  overflow: hidden;
  text-overflow: ellipsis;
  white-space: nowrap;
}

// Usage
.branch-name {
  display: inline-block;
  max-width: 200px;
  .text-overflow();
}
```

視網膜圖片

指定兩個圖片路徑和 `@1x` 圖片尺寸，然後 Bootstrap 3 將會將提 `@2x` 的媒體查詢。如果你有很多的圖片服務，可以考慮在單一個媒體查詢中手動撰寫你的視網膜圖片 CSS 。

```

.img-retina(@file-1x; @file-2x; @width-1x; @height-1x) {
  background-image: url("@{file-1x}");

  @media
    only screen and (-webkit-min-device-pixel-ratio: 2),
    only screen and (  min--moz-device-pixel-ratio: 2),
    only screen and (      -o-min-device-pixel-ratio: 2/1),
    only screen and (          min-device-pixel-ratio: 2),
    only screen and (              min-resolution: 192dpi),
    only screen and (              min-resolution: 2dppx) {
      background-image: url("@{file-2x}");
      background-size: @width-1x @height-1x;
    }
}

// Usage
.jumbotron {
  .img-retina("/img/bg-1x.png", "/img/bg-2x.png", 100px, 100px);
}

```

使用 Sass

雖然 Bootstrap 3 是基於 Less 建置的，我們也提供了一個官方 Sass 移植版 (<https://github.com/twbs/bootstrap-sass>)。我們維護它在一個獨立的 GitHub 儲存庫中，並處理轉換 Script 的更新。

包含的內容

雖然 Sass 移植版有一個獨立儲存庫，並針對不同的使用者，這個專案的內容和 Bootstrap 3 主專案有很大的不同。這是為了確保 Sass 移植版與其他許多基於 Sass 的系統盡可能保持相容。

Path	Description
lib/	Ruby gem code (Sass configuration, Rails and Compass integrations)
tasks/	Converter scripts (turning upstream Less to Sass)
test/	Compilation tests
templates/	Compass package manifest
vendor/assets/	Sass, JavaScript, and font files
Rakefile	Internal tasks, such as rake and convert

訪問 Sass 移植版 GitHub 儲存庫 (<https://github.com/twbs/bootstrap-sass>)，以瞭解這些檔案的作業。Visit the Sass port's GitHub repository (<https://github.com/twbs/bootstrap-sass>) to see these files in action.

安裝

關於如何安裝與使用 Bootstrap 3 的 Sass 移植版，請參考 GitHub 儲存庫的 readme 檔案 (<https://github.com/twbs/bootstrap-sass>)。它包含最新原始碼，包含使用 Rails、Compass 和標準 Sass 專案資訊。

Sass 的 Bootstrap (<https://github.com/twbs/bootstrap-sass>)

1 Comment kkbruce.tw

 Login ▾

 Recommend  Share

從最好的優先排列 ▾



Join the discussion...



tony • 一年前

「表單」內容有錯誤：

```
<input type="密碼" class="form-control" id="exampleInputPassword1"  
placeholder="Password">
```

應該是：

```
<input type="Password" class="form-control" id="exampleInputPassword1"  
placeholder="密碼">
```



^ | v · Reply · Share ›

設計與建置由有世界大愛的 @mdo (<http://twitter.com/mdo>) 和 @fat (<http://twitter.com/fat>)。
並由 陳傳興 (/About) 翻譯。

由核心團隊 (<https://github.com/twbs?tab=members>) 與 我們的貢獻者
(<https://github.com/twbs/bootstrap/graphs/contributors>) 的幫助下維護。

程式碼依 MIT (<https://github.com/twbs/bootstrap/blob/master/LICENSE>) 授權，英文文件依 CC BY 3.0 (<http://creativecommons.org/licenses/by/3.0/>) 授權。

Code licensed under MIT (<https://github.com/twbs/bootstrap/blob/master/LICENSE>),
documentation under CC BY 3.0 (<http://creativecommons.org/licenses/by/3.0/>).

翻譯版本 v3.3.1 · GitHub (<https://github.com/twbs/bootstrap>) · 範例
(/bs3/Gettingstarted#examples) · v2.3.2 (/bs2) · 關於 (/bs3/About) · 官方精選
(<http://expo.getbootstrap.com>) · 官方Blog (<http://blog.getbootstrap.com>) · 問題
(<https://github.com/twbs/bootstrap/issues>) · 發行版本
(<https://github.com/twbs/bootstrap/releases>)