

目錄

Bootstrap4 介紹	1.1
相關資源網站	1.2
安裝與開始	1.3
CSS 基礎知識	1.4
容器container 應用	1.5
網格系統 應用	1.6
斷點	1.6.1
版本斷點比較	1.6.2
欄位排序	1.6.3
欄位推移	1.6.4
巢狀欄位	1.6.5
Flexbox 方向性	1.6.6
間隔工具	1.7
按鈕(button)	1.8
不同的呈現(display)方式	1.9
容器對齊方式	1.10
文字(背景bgcolor)對齊與樣式	1.11
導覽列	1.12
參考資源	1.13

Bootstrap 4 介紹

- 世界上最受歡迎的框架用於建立響應式、行動優先的網站
- 用於做RWD網站設計

相關資源網站

一、BootStrap相關資源

- BootStrap4官網：<https://getbootstrap.com/>
- 正體中文手冊：<http://bootstrap.hexschool.com/docs/4.0/getting-started/introduction/>

二、相關套件與資源網站

- jquery：<https://jquery.com/>
- popper.js：<https://popper.js.org/>
- fontawesome：<http://fontawesome.io/>

三、相關資源網站

- CSS參考手冊：<http://css.doyoe.com/>

安裝與開始

一、安裝資源

- 新建專案資料夾
- 下載最新版 Bootstrap 4 : [Bootstrap 4下載](#)
- 將下載的 js , css 資料夾拉出 放到 專案資料夾 下
- BootStrap 4拿掉了圖示，可到 [fontawesome下載](#)
- 解壓後，將 fonts 及 css 複製到 專案資料夾 裡面
- 下載 jquery : [jquery下載](#)，並存入 js 資料夾中
- 下載 popper.js : [popper下載](#)，存入 js 資料夾中

二、引入index.html內

- 可以使用網址引入
- CSS :

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-Gn5384xqQ1aowXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">
```

- js :

```
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCKRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous">
</script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous">
</script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-JZR6Spejh4U02d8j0t6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmY1"
crossorigin="anonymous">
</script>
```

- 建立index.html

```
<html lang="zh-Hant-TW">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-
```

```

to-fit=no">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Doc</title>
    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <link rel="stylesheet" href="css/fontawesome.css">
    <link rel="stylesheet" href="css/my.css">
    <!-- JavaScript -->
    <script src="js/jquery-3.2.1.min.js"></script>
    <script src="js/popper.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
</head>
<body>
    <h1>Hello, world!</h1>
</body>
</html>

```

- 引入順序請勿隨便調整
- **width=device-width**：表示寬度是設備螢幕的寬度
- **initial-scale=1**：表示初始的縮放比例，可以讓網頁的寬度自動適應手機螢幕的寬度
- 在iOS9更新中更改了**initial-scale**的用途，故使用**shrink-to-fit=no**來替代
- 其中的 **css/my.css** 是用來放置自定義的CSS內容，例如想要讓頁面用微軟正黑體，加入字型設定即可：

```

body{
    font-family: '微軟正黑體';
}

```

CSS 基礎知識

一、關於CSS

- Bootstrap是一套CSS框架，簡單了解CSS用法是必要的
- CSS：Cascading Style Sheets，簡稱CSS，中文通常稱為層疊樣式表或串接樣式表
- 可用來替網頁上的各種元件設定其外觀
- 套用樣式的三種方法：

	行內樣式	內部頁面樣式	外部連結樣式
範圍	最小	頁中	許多頁
優先	最優先		
用法	<code><標籤 style="CSS;"></code>	<code><style type='text/css'> CSS; </style></code>	<code><link href="CSS檔.css" rel="stylesheet"></code>

- 頁面可同時套用數個css檔或設定，重複的以最後讀到的為主
- 並可以用import來引入其他的CSS檔：如：`@import url("my.css");`

	標籤挑選器	ID挑選器	類別挑選器
樣式表	<code>p {CSS屬性: 值;}</code>	<code>#home{CSS屬性: 值;}</code>	<code>.row{CSS屬性: 值;}</code>
網頁	<code><p></p></code>	<code><div id="home"></div></code>	<code><div class="row"></div></code>

- 基本上，BootStrap就是一堆類別挑選器的組合

container(容器) 應用

一、基本概念

- 要用 Bootstrap 的網格系統來排版，一定要使用容器（container）
- 頁面中可有多個容器，容器可巢狀使用（容器中還有容器）
- 容器有兩種，一種是寬度有上限的容器：

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

- 一種是全螢幕容器：

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

- 沒加容器之前。網頁內容會緊貼著邊緣，加上之後，預設就有邊距出現
- 可參考：

<http://bootstrap.hexschool.com/docs/4.0/layout/overview/#%E5%AE%B9%E5%99%A8>

grid(網格) 系統應用

一、使用網格的注意事項

- 網格系統以 **flexbox (CSS3)** 為基礎，可以做到自動等寬等高的欄位、自動平均欄位、垂直置中等以往很難做的功能。
- 同時也代表對 **IE9 (以下)** 不再支援，至少要用**IE10**才行。
- 網格系統必須在容器 (**.container**) 中才有作用

二、指定欄寬用法

- 使用範例：

```
<div class="container">
  <div class="row">
    <div class="col-sm-5">左邊五欄</div>
    <div class="col-sm-7">右邊七欄</div>
  </div>
</div>
```

- **.row** 代表一個橫列，確保裡面的欄位對齊、排序。
- **.col-sm** 是斷點前綴詞，後面接的是欄位寬度設定，預設滿版為**12**欄
- 若超過**12**欄，自動新增一個 **.row**

三、自動等寬用法

- 使用等寬欄位

```
<div class="container">
  <div class="row">
    <div class="col">col 等寬 1/5</div>
    <div class="col">col 等寬 2/5</div>
    <div class="col">col 等寬 3/5</div>
    <div class="col">col 等寬 4/5</div>
    <div class="col">col 等寬 5/5</div>
  </div>
  <div class="row">
    <div class="col-sm">col-sm 等寬 1/5</div>
    <div class="col-sm">col-sm 等寬 2/5</div>
    <div class="col-sm">col-sm 等寬 3/5</div>
    <div class="col-sm">col-sm 等寬 4/5</div>
    <div class="col-sm">col-sm 等寬 5/5</div>
  </div>
</div>
```


- 上方是不管螢幕多大，都分成五欄
- 下方是當螢幕在 576 以上時，才分五欄，576 以下的時候，自動調成單欄式。

四、指定欄寬+自動欄寬

- 可以指定某幾欄的欄寬，其餘的自行自動平均

```
<div class="container">
  <div class="row">
    <div class="col-sm">自動平均分配</div>
    <div class="col-sm-7">指定佔 7 欄</div>
    <div class="col-sm">自動平均分配</div>
  </div>
</div>
```

五、根據內容多寡自動調整欄寬

- 不確定該站多少欄寬，讓電腦自行根據內容多寡來判斷：

```
<div class="container">
  <div class="row">
    <div class="col-sm">自動平均分配</div>
    <div class="col-sm-auto">根據中間內容多多寡來判斷欄寬</div>
    <div class="col-sm">自動平均分配</div>
  </div>
</div>
```

六、不同斷點的欄寬搭配

- 在同一個欄位根據不同斷點給予不同大小

```
<div class="container">
  <div class="row">
    <div class="col">自動平均分配</div>
    <div class="col-sm-6 col-md-7 col-lg-8">根據不同螢幕大小來指定中間欄位欄寬</div>
    <div class="col">自動平均分配</div>
  </div>
</div>
```

斷點

一、什麼是斷點？

	col	col-sm	col-md	col-lg	col-xl
改變範圍	< 576 px	>= 576 px	>= 768 px	>= 992 px	>= 1200 px
硬體裝置	any	手機	平板	桌機螢幕	超寬螢幕

- 寫：.col-sm-3，意思就是當螢幕 $\geq 576\text{px}$ 時，就會將此區域的寬度設為螢幕的 $3/12$ 。
- 同時寫：col-sm-4 col-md-3。意思就是當螢幕 $\geq 768\text{px}$ 時，就會將此區域的寬度設為螢幕的 $3/12$ ，若螢幕 < 768 但 $\geq 576\text{px}$ 時，就會將此區域的寬度設為螢幕的 $4/12$ 。

Bootstrap 3 及 Bootstrap 4 的斷點比較

斷點差異比較表(以 **px** 為單位)

	575 以下	576 ~ 767	768 ~ 991	992 ~ 1199	1200 以上
Bootstrap 3	col-xs	col-xs	col-sm	col-md	col-lg
Bootstrap 4	col-	col-sm	col-md	col-lg	col-xl
對應裝置	any	手機	平板	桌機螢幕	超寬螢幕

欄位排序

常見的例子是三欄式畫面（左側邊+主內容+右側邊），
在手機觀看，變成單欄式時，希望變成：（主內容→左側邊→右側邊）的順序。

- 用 `.order` 調整：

```
<div class="container">
  <div class="row">
    <div class="col-sm-7 order-sm-2">
      在中間的主內容區
    </div>
    <div class="col-sm order-sm-1">
      在左邊的側邊欄
    </div>
    <div class="col-sm order-sm-3">
      在右邊的側邊欄
    </div>
  </div>
</div>
```

請依照在單欄式情況下，希望呈現的先後順序來寫，然後再利用 `order` 來調整位置。
在三欄式時（未自適應前），會按照 `order` 的順序來排，
在單欄式時，就會忽略 `order`，依照語法順序來排。

欄位推移

- Bootstrap 4 裡面已經沒辦法用 `.col-offset-x` 或者 `.offset-sm-x` 來跳過幾欄
- 可利用下表將剩下的空位騰出來。

.mr-auto	.mx-auto	.ml-auto
將右邊剩下的空位騰出	空位平均分散到左右兩邊	空位放到左邊

- 使用範例：

```
<div class="container">
  <div class="row">
    <div class="col-2">col 等寬 1/3</div>
    <div class="col-2 ml-auto">col 等寬 2/3</div>
    <div class="col-2">col 等寬 3/3</div>
  </div>
</div>
```

.mr-auto

col 等

寬 1/3

col 等

寬 2/3

col 等

寬 3/3

.mx-auto

col 等

寬 1/3

col 等

寬 2/3

col 等

寬 3/3

.ml-auto

col 等

寬 1/3

col 等

寬 2/3

col 等

寬 3/3

巢狀欄位

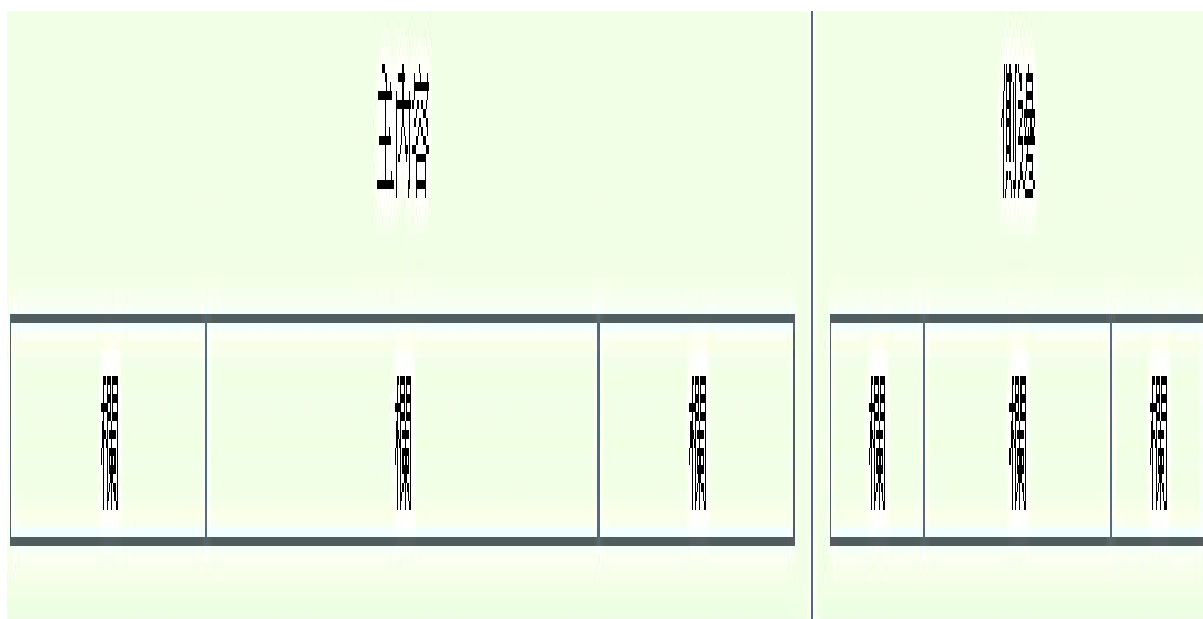
一、使用網格的注意事項

- .row 裡面還可以有 .row，子 .row 會以父 .row 的寬度為依據來進行寬度調整
- 使用範例：
- html:

```
<div class="container">
  <div class="row">
    <div class="col-sm-8">
      主內容
      <div class="row">
        <div class="col-sm-3">欄</div>
        <div class="col-sm-6">欄</div>
        <div class="col-sm-3">欄</div>
      </div>
    </div>
    <div class="col-sm-4">
      側邊
      <div class="row">
        <div class="col-sm-3">欄</div>
        <div class="col-sm-6">欄</div>
        <div class="col-sm-3">欄</div>
      </div>
    </div>
  </div>
</div>
```

- CSS:

```
div.col-sm-8, div.col-sm-4, div.col-sm-3, div.col-sm-6{
  background-color: #f1ffe7;
  border:1px solid #88a0a8;
  text-align: center;
}
```



Flexbox方向性

- 水平：.flex-row（預設值）
- 反向水平：.flex-row-reverse
- 垂直：.flex-column
- 反向垂直：.flex-column-reverse
- 範例：
- html：

```
<div class="container">
  <div class="d-flex flex-row my-1">
    <div class="p-1">col 1/3</div>
    <div class="p-1">col 2/3</div>
    <div class="p-1">col 3/3</div>
  </div>
  <div class="d-flex flex-row-reverse my-1">
    <div class="p-1">col 1/3</div>
    <div class="p-1">col 2/3</div>
    <div class="p-1">col 3/3</div>
  </div>
  <div class="d-flex flex-column my-1">
    <div class="p-1">col 1/3</div>
    <div class="p-1">col 2/3</div>
    <div class="p-1">col 3/3</div>
  </div>
  <div class="d-flex flex-column-reverse my-1">
    <div class="p-1">col 1/3</div>
    <div class="p-1">col 2/3</div>
    <div class="p-1">col 3/3</div>
  </div>
</div>
```

- CSS：

```
body{
  background-color: #88a0a8;
}
div.container{
  background-color: #86bbbd;
}
div.row{
  background-color: #c2e7da;
}
div.p-1, div.col{
  background-color: #f1ffe7;
  border: 1px solid #88a0a8;
```

}

col 1/3	col 2/3	col 3/3
---------	---------	---------

col 3/3	col 2/3	col 1/3
---------	---------	---------

col 1/3

col 2/3

col 3/3

col 3/3

col 2/3

col 1/3

間隔工具

- 間隔工具的組成為「屬性邊緣-尺寸」：ml-3
- 屬性設定
 - m - 這個 class 會設定 margin
 - p - 這個 class 會設定 padding
- 邊緣設定
 - t - 這個 class 會設定 margin-top 或 padding-top
 - b - 這個 class 會設定 margin-bottom 或 padding-bottom
 - l - 這個 class 會設定 margin-left 或 padding-left
 - r - 這個 class 會設定 margin-right 或 padding-right
 - x - 這個 class 會設定 *-left* 和 *-right*
 - y - 這個 class 會設定 *-top* 和 *-bottom*
 - 空白 - 如果邊緣 class 未加入則會設定 margin 或 padding 在元素的四個邊緣
- 尺寸設定
 - 0 - 這個 class 會設定 margin 或 padding 的樣式值為 0
 - 1 - (預設時) 這個 class 會設定 margin 或 padding 於 \$spacer * .25
 - 2 - (預設時) 這個 class 會設定 margin 或 padding 於 \$spacer * .5
 - 3 - (預設時) 這個 class 會設定 margin 或 padding 於 \$spacer
 - 4 - (預設時) 這個 class 會設定 margin 或 padding 於 \$spacer * 1.5
 - 5 - (預設時) 這個 class 會設定 margin 或 padding 於 \$spacer * 3
- 使用範例：

```
<div class="container">
  <div class="row">
    <div class="col-2">col 等寬 1/3</div>
    <div class="col-2 pl-5">col 等寬 2/3</div>
    <div class="col-2">col 等寬 3/3</div>
  </div>
</div>
```

按鈕

.btn 設計 `<button>` 元素使用。但也可在 `<a>` 或 `<input>` 元素上使用

標籤	使用方法
a	<code>Link</code>
button	<code><button class="btn btn-primary" type="submit">Button</button></code>
input:button	<code><input class="btn btn-primary" type="button" value="Input"></code>
input:submit	<code><input class="btn btn-primary" type="submit" value="Submit"></code>
input:reset	<code><input class="btn btn-primary" type="submit" value="reset"></code>

按鈕顏色樣式

- 改變按鈕顏色

顏色	對應的class
藍色	btn-primary
灰色	btn-secondary
綠色	btn-success
紅色	btn-danger
黃色	btn-warning
淺藍色	btn-info
白色	btn-light
黑色	btn-dark
原生	btn-link

- 若要改變外框顏色：加上 `outline` 即可
- 例子：`btn-outline-primary`

按鈕大小樣式

小	大	區塊
btn-sm	btn-lg	btn-block

- 範例：

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

啟用與停用

- 啟用：active
- 停用：disabled

狀態	標籤	用法	範例
啟用	a	class	<code>link</code>
啟用	button	class	<code><button type="button" class="btn btn-lg btn-primary active">button</button></code>
啟用	input	class	<code><input type="button" class="btn btn-secondary btn-lg active" value="input" ></code>
停用	a	class	<code>link</code>
停用	button	標籤	<code><button type="button" class="btn btn-lg btn-primary" disabled>button</button></code>
停用	input	標籤	<code><input type="button" class="btn btn-secondary btn-lg active" disabled value="input" ></code>

切換狀態

- 增加 data-toggle="button"
- 如預先需要切換按鈕，則須手動將 .active class 和 aria-pressed="true" 添加到 `<button>`
- 範例：

```
<button type="button" class="btn btn-primary" data-toggle="button" aria-pressed="false" autocomplete="off">
Single toggle
</button>
```

自行定義顏色管理工具

- 自行定義按鈕工具：<http://blog.koalite.com/bbg/>
- 定義出來的按鈕可以放在 `css/my.css` 中
- 若希望底色是透明的，可自行將背景顏色取代為 `transparent`

不同的呈現(**display**)方式

一、套用**display**方法

- xs : .d-{value}
- sm, md, lg, xl : .d-{breakpoint}-{value}
- value 可換成：
 - none
 - inline
 - inline-block
 - block
 - table
 - table-cell
 - table-row
 - flex
 - inline-flex
- 範例：

```
<!--  
  d-inline = display:inline : 在同一行  
  p-2 : 對內(padding)四邊距為$spacer * .5  
  bg : 背景顏色  
  text-white : 文字顏色  
-->  
<div class="d-inline p-2 bg-primary text-white">d-inline</div>  
<div class="d-inline p-2 bg-dark text-white">d-inline</div>
```

隱藏元素

- 例子：.d-none .d-md-block .d-lg-none 將隱藏所有螢幕尺寸的元素，除了中型(lg)和大型(xl)設備

螢幕大小	class
Hidden on all	.d-none
Hidden only on xs	.d-none .d-sm-block
Hidden only on sm	.d-sm-none .d-md-block
Hidden only on md	.d-md-none .d-lg-block
Hidden only on lg	.d-lg-none .d-xl-block
Hidden only on xl	.d-xl-none
Visible on all	.d-block

Visible only on xs	.d-block .d-sm-none
Visible only on sm	.d-none .d-sm-block .d-md-none
Visible only on md	.d-none .d-md-block .d-lg-none
Visible only on lg	.d-none .d-lg-block .d-xl-none
Visible only on xl	.d-none .d-xl-block

運用在列印

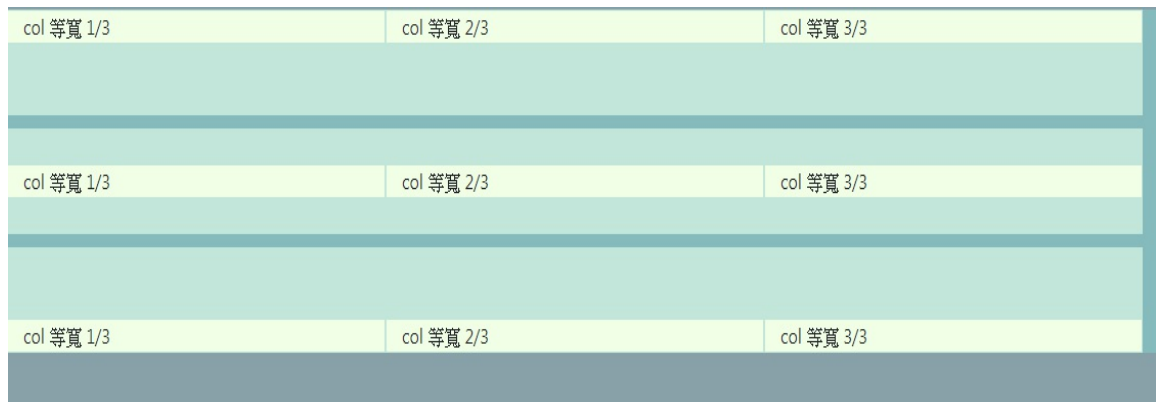
- 例子：

```
<div class="d-print-none">只出現在螢幕,但不印出來</div>
<div class="d-none d-print-block">只有列印才會看到</div>
<div class="d-none d-lg-block d-print-block">只有 lg螢幕以上 和 列印出來 才會顯示</div>
</div>
```

class
.d-print-none
.d-print-inline
.d-print-inline-block
.d-print-block
.d-print-table
.d-print-table-row
.d-print-table-cell
.d-print-flex
.d-print-inline-flex

容器對齊方式

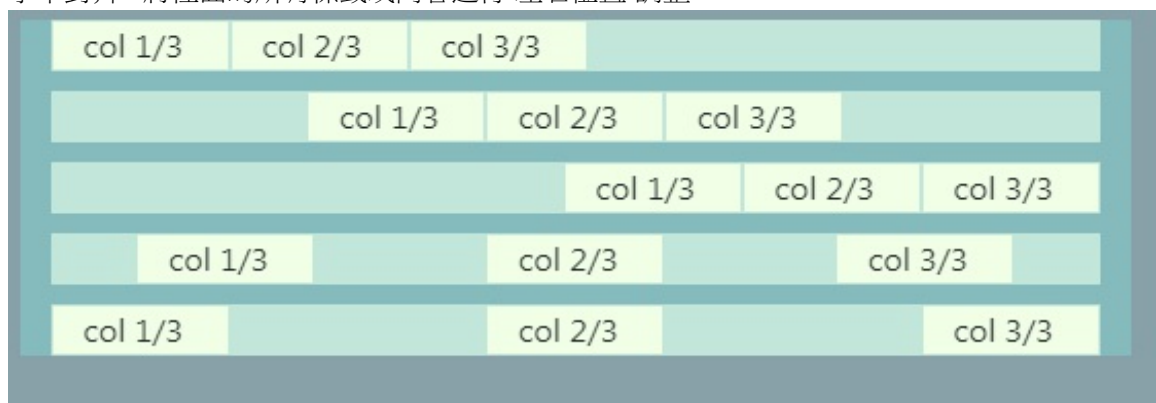
- 全部垂直對齊：將裡面的所有標籤或內容進行 上下位置 調整



- 個別垂直對齊：將自身標籤進行 上下位置 調整



- 水平對齊：將裡面的所有標籤或內容進行 左右位置 調整



對齊方式	位置	用法
全部垂直靠上對齊	outside	align-items-start
全部垂直置中對齊	outside	align-items-center
全部垂直靠下對齊	outside	align-items-end
個別垂直靠上對齊	inside	align-self-start

個別垂直置中對齊	inside	align-self-center
個別垂直靠下對齊	inside	align-self-end
水平靠左對齊	outside	justify-content-start
水平置中對齊	outside	justify-content-center
水平靠右對齊	outside	justify-content-end
水平分散對齊(有左右間隔)	outside	justify-content-around
水平分散對齊(無左右間隔)	outside	justify-content-between

- 範例：

```
<div class="container">
  <div class="row (outside)">
    <div class="col (inside)">col 等寬 1/3</div>
    <div class="col (inside)">col 等寬 2/3</div>
    <div class="col (inside)">col 等寬 3/3</div>
  </div>
</div>
```

內容(文字，圖案)對齊

- 前面講的是 `<div>` 容器本身的對齊，這裡講的是 `<div>` 裡面內容的對齊。
- 例子：`<div class="col-sm text-right"></div>`
- 中間可加：`sm,md,lg,xl => text-sm-left`

對齊方式	用法
靠左對齊	<code>text-left</code>
置中對齊	<code>text-center</code>
靠右對齊	<code>text-right</code>

- 注意，這種對齊僅對行內元件（`inline`）有效，如果內容是區塊元件（`block`），如 `<div>`，那就要用別的方式

text-wrap 和 Overflow

- 防止換行可搭配 `.text-nowrap`

```
<div class="text-nowrap" style="width: 8rem;">
  This text should overflow the parent.
</div>
```

- 更長的內容，可增加 `.text-truncate`，截掉多餘內容改用 ...。需要使用 `display: inline-block or display: block`

```
<!-- Praeterea iter... -->
<div class="row">
  <div class="col-2 text-truncate">
    Praeterea iter est quasdam res quas ex communi.
  </div>
</div>

<!-- Praeterea iter est q... -->
<span class="d-inline-block text-truncate" style="max-width: 150px;">
  Praeterea iter est quasdam res quas ex communi.
</span>
```

文字字體變換

- 將元件中的文字內容轉變樣式
- 例子：

```
<p class="text-lowercase">Lowercased text.</p>
```

```

<p class="text-uppercase">Uppercased text.</p>
<p class="text-capitalize">CapiTaliZed text.</p>

<p class="font-weight-bold">Bold text.</p>
<p class="font-weight-normal">Normal weight text.</p>
<p class="font-weight-light">Light weight text.</p>
<p class="font-italic">Italic text.</p>

```

- 注意 `text-capitalize` 僅改變第一個字母，其它字母的大小寫不受影響

文字顏色變換

- 下表為文字顏色總表
- 注意，`.text-white`、`.text-muted` 沒有連結樣式(focus時不會變色)

顏色	用法
藍	<code>.text-primary</code>
灰	<code>.text-secondary</code>
綠	<code>.text-success</code>
紅	<code>.text-danger</code>
黃	<code>.text-warning</code>
淺藍	<code>.text-info</code>
亮	<code>.text-light</code>
黑	<code>.text-dark</code>
淺灰	<code>.text-muted</code>
白	<code>.text-white</code>

背景顏色

- 下表為背景顏色總表

顏色	用法
藍	<code>.bg-primary</code>
灰	<code>.bg-secondary</code>
綠	<code>.bg-success</code>
紅	<code>.bg-danger</code>
黃	<code>.bg-warning</code>
淺藍	<code>.bg-info</code>
亮	<code>.bg-light</code>

黑	.bg-dark
白	.bg-white

背景漸層：(**bg-gradient**-後面跟**bg**一樣)

- 當 `$enable-gradients` 設置為`true`時，您將可以使用 `.bg-gradient-` 的通用類別。
預設情況下，`$enable-gradients` 是停用的
在您開始使用 **Bootstrap** 時進行自定義。 [了解我們的 Sass 選項](#)

導覽列

- `.navbar-expand{-sm|-md|-lg|-xl}`
是指 `{-sm|-md|-lg|-xl}` 範圍以上才會直接展開選項
- `.navbar-brand` 專案名稱。
- `.navbar-toggler` 用於折疊插件和其他 `navigation toggling` 行為。
- `.navbar-nav` 提供導航（包括下拉清單）。
- `.form-inline` 任何表單控制元件和操作。
- `.navbar-text` 垂直居中的文本字串。
- `.collapse.navbar-collapse` 外層中斷點群組和隱藏導覽列內容。

`.navbar-brand` 製作自己的專案品牌

- 可放導覽或折疊插件下面,小型螢幕裝置會改變頁面配置
 - 可參考：<http://bootstrap.hexschool.com/docs/4.0/components/navbar/#toggler>
- 範例：

```
<nav class="navbar navbar-light bg-light">
  <a class="navbar-brand" href="#">
    
    Bootstrap
  </a>
</nav>
```

`.navbar-toggler` 用於折疊插件

- 配合下面的導覽而先介紹 `.navbar-toggler`
- 導覽列可以使用 `.navbar-toggler`、`.navbar-collapse`、`.navbar-expand{-sm|-md|-lg|-xl}` 來改變。結合其他通用類別，選擇顯示或隱藏特定元素。
- 不需要折疊的導覽列，在導覽列中加入 `.navbar-expand`。總是折疊的導覽列，不加任何 `.navbar-expand`
- `toggler` 是為了讓導覽資訊在螢幕較小的裝置時變成下拉式選單的關鍵
- 可以看到下面範例,`data-target`指定要收縮

裡面的內容,所以id必須要一樣
- 範例：

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="#">Navbar</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse"
    data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
    aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
```

```

<div class="collapse navbar-collapse" id="navbarSupportedContent">
  <!-- 可將導覽放入 -->
</div>
</nav>

```

.navbar-nav 導覽

- 導覽的連結建立在 .nav 選項上
- 在導覽列中的導覽元件，將佔用更多的水平空間，以保持導覽列內容安全對齊
- 用 .active 表示當前頁面。可用於 .nav-link 或 .nav-item
- 範例：(兩者效果依樣)

```

<!-- 範例 1 -->
<ul class="navbar-nav">
  <li class="nav-item active">
    <a class="nav-link" href="#">Home
      <span class="sr-only">(current)</span>
    </a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Features</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Pricing</a>
  </li>
  <li class="nav-item">
    <a class="nav-link disabled" href="#">Disabled</a>
  </li>
</ul>

<!-- 範例 2 -->
<div class="navbar-nav">
  <a class="nav-item nav-link active" href="#">Home
    <span class="sr-only">(current)</span>
  </a>
  <a class="nav-item nav-link" href="#">Features</a>
  <a class="nav-item nav-link" href="#">Pricing</a>
  <a class="nav-item nav-link disabled" href="#">Disabled</a>
</div>

```

下拉式選單：請確保 .nav-item 和 .nav-link 使用單獨及巢狀的元素

- 範例：

```

<li class="nav-item dropdown">
  <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink"

```

```

    data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
    Dropdown link
  </a>
  <div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
    <a class="dropdown-item" href="#">Action</a>
    <a class="dropdown-item" href="#">Another action</a>
    <a class="dropdown-item" href="#">Something else here</a>
  </div>
</li>

```

表單

- 在導覽列中放置各種表單控制元件，這是透過 `.form-inline` 水平放置的表單
- 範例：

```

<nav class="navbar navbar-light bg-light">
  <form class="form-inline">
    <input class="form-control mr-sm-2" type="search" placeholder="Search" aria-label="Search">
    <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>
  </form>
</nav>

```

- 根據需要將行內表單的內容透過通用類別對齊。(可參考：容器對齊方式)

```

<!-- justify-content-between : 水平分散對齊(無左右間隔) -->
<nav class="navbar navbar-light bg-light justify-content-between">
  <!-- 裡面內容會自動對齊 -->
</nav>

```

文字

- 導覽列可在 `.navbar-text` 包含一些文本。`class` 會調整文本字串的垂直對齊和水平間距
- 範例：

```

<nav class="navbar navbar-light bg-light">
  <span class="navbar-text">
    Navbar text with an inline element
  </span>
</nav>

```

顏色方案

- `.navbar-light` 選淺色背景顏色，或者 `.navbar-dark` 用深色背景顏色
- 更多顏色：

- <https://work.smarchal.com/twbscolor/css/e74c3cc0392becf0f1ffbbbc0>
- <https://stackoverflow.com/questions/18529274/change-navbar-color-in-twitter-bootstrap>
- 範例：

```
<nav class="navbar navbar-dark bg-dark">
  <!-- Navbar content -->
</nav>

<nav class="navbar navbar-dark bg-primary">
  <!-- Navbar content -->
</nav>

<nav class="navbar navbar-light" style="background-color: #e3f2fd;">
  <!-- Navbar content -->
</nav>
```

容器

- 可將導覽列包裹在 `.container` 中，放置在頁面中央或內容的中間 固定或靜態於頂的導覽列 位置
- 範例：

```
<div class="container">
  <nav class="navbar navbar-expand-lg navbar-light bg-light">
    <a class="navbar-brand" href="#">Navbar</a>
  </nav>
</div>
```

- `.container` 在導覽列中，其水平 `padding` 在低於指定的 `.navbar-expand{-sm|-md|-lg|-xl}` 中斷點處被移除 這確保了當您的導覽列折疊時，不會產生多餘的 `padding`
- 範例：

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="container">
    <a class="navbar-brand" href="#">Navbar</a>
  </div>
</nav>
```

定位

- 預設：

```
<nav class="navbar navbar-light bg-light">
  <a class="navbar-brand" href="#">Default</a>
</nav>
```

- 固定在上方(相對位置)

```
<nav class="navbar fixed-top navbar-light bg-light">  
  <a class="navbar-brand" href="#">Fixed top</a>  
</nav>
```

- 固定在下方(相對位置)

```
<nav class="navbar fixed-bottom navbar-light bg-light">  
  <a class="navbar-brand" href="#">Fixed top</a>  
</nav>
```

參考資源

- 實作入門：https://campus-xoops.tn.edu.tw/modules/tad_book3/html_all.php?tbsn=32
- 正體中文手冊：<http://bootstrap.hexschool.com/docs/4.0/getting-started/introduction/>