

## บรรณานุกรม

- Narisa. เรื่องความรู้เกี่ยวกับ Web Application [ออนไลน์]. จาก <http://narisara212.blogspot.com/2014/01/web-application-search-engine.html>. สืบค้นเมื่อวันที่ 3 มกราคม 2558
- Suranart Niamcome. เรื่อง Bootstrap คืออะไร [ออนไลน์]. จาก <http://www.siamhtml.com/bootstrap-คืออะไร-สอนวิธีใช้/>. สืบค้นเมื่อวันที่ 14 กรกฎาคม 2557
- Cloud-Tutorial. เรื่อง สอนใช้ Bootstrap [ออนไลน์]. จาก <http://cloud-tutorial.com/บทความ/>. สืบค้นเมื่อวันที่ 19 มกราคม 2558
- Bootstrap. เรื่อง Bootstrap Framework [ออนไลน์]. จาก <http://getbootstrap.com/getting-started/>

## ภาคผนวก ก

### Bootstrap Framework

สามารถเข้าไปดาวน์โหลดไฟล์ที่ <http://getbootstrap.com/getting-started/#download> ซึ่งจะประกอบไปด้วยไฟล์ CSS, Fonts, Javascript

การใช้งาน Bootstrap

```
<link href="css/bootstrap.min.css" rel="stylesheet">
```

เพิ่ม Style Sheets ของ Bootstrap เข้าไปที่ <head>...</head>

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link href="css/bootstrap.min.css" rel="stylesheet">
<link href="css/bootstrap-responsive.min.css" rel="stylesheet">
```

ถ้าต้องการให้เว็บเป็นแบบ Responsive ให้ใส่ bootstrap-responsive.min.css เข้าไป

```
<body>
.
.
.
<script src="js/jquery.js"></script>
<script src="js/bootstrap.min.js"></script>
</body>
```

เมื่อต้องการใช้ javascript ที่ Bootstrap เตรียมมาให้ได้นั้นต้องใส่ส่วนนี้ลงไป

การใช้งาน Bootstrap นั้น ต้องเริ่มต้นจากการสร้าง Container และตามด้วยจัดแบ่งสัดส่วน Column ด้วย หลักการ Grid System

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



รูปที่ ก.1 ตัวอย่างการจัดแบ่ง Column ด้วยหลักการ Grid System

### Grid System ของ Bootstrap

ปกติแล้ว grid system ที่ Bootstrap ให้มาจะเป็นแบบ 12 คอลัมน์ ซึ่งจะใช้ร่วมกับ container ที่เพิ่งสร้างขึ้นมา โดยแต่ละคอลัมน์จะมีความกว้างเท่ากันคือ 60px และจะอยู่ห่างกัน 20px จะได้ว่า  $(60 * 12) + (20 * 11) = 940$  หรือความกว้างของ container การจะสร้าง layouts ด้วย Bootstrap ให้สร้าง html element ที่มี class “row” ขึ้นมาก่อนถึงจะสามารถสร้างแต่ละคอลัมน์ขึ้นมาได้ด้วยการใส่ html element ที่มี class “span\*” เข้าไปข้างใน โดย \* ที่อยู่ข้างหลัง span นั้นจะเป็นตัวเลขที่จะระบุว่าต้องการให้คอลัมน์นี้ “กาง” แค่ไหน สมมติว่าจะสร้าง layouts แบบ 12 คอลัมน์ ให้ใส่โค้ดดังนี้

## 12 คอลัมน์

```
<div class="row">
  <div class="span1">Column 1</div>
  <div class="span1">Column 2</div>
  <div class="span1">Column 3</div>
  <div class="span1">Column 4</div>
  <div class="span1">Column 5</div>
  <div class="span1">Column 6</div>
  <div class="span1">Column 7</div>
  <div class="span1">Column 8</div>
  <div class="span1">Column 9</div>
  <div class="span1">Column 10</div>
  <div class="span1">Column 11</div>
  <div class="span1">Column 12</div>
</div>
```

จากโค้ดด้านบนเราได้ใส่ HTML elements ที่มี class ชื่อ “span1” รวมทั้งหมด 12 elements ด้วยกัน เมื่อพรีวิวดู container จะถูกแบ่งออกเป็นคอลัมน์ต่างๆ ที่มีความกว้างเท่ากันทั้งหมดจำนวน 12 คอลัมน์ ด้วยกัน

## 2 คอลัมน์

ถ้าต้องการแบ่ง 2 คอลัมน์ และให้คอลัมน์ซ้ายใหญ่กว่าคอลัมน์ขวาดังนี้

```
<div class="row">
  <div class="span8">Main</div>
  <div class="span4">Sidebar</div>
</div>
```

จะเห็นว่าความกว้างของทั้ง 2 คอลัมน์นี่จะเป็น  $8+4 = 12$  พอดี ที่จำเป็นต้องกำหนดความกว้างให้รวมกันเท่ากับ 12 เพราะว่า grid system ของ bootstrap นั้นเป็นแบบ 12 แล้วเท่ากับว่า 2 คอลัมน์นี้จะกางเต็ม container พอดี

## 3 คอลัมน์

ถ้าต้องการ 3 คอลัมน์ ก็กำหนดให้ 3 คอลัมน์มีผลรวมเท่ากับ 12 ตามเดิม

```
<div class="row">
  <div class="span8">Main</div>
  <div class="span4">Sidebar</div>
</div>
```

จะเห็นว่าความกว้างของทั้ง 2 คอลัมน์นี่จะเป็น  $8+4 = 12$  พอดี การที่เราจำเป็นต้องกำหนดความกว้างให้รวมกันเท่ากับ 12 เพราะว่า grid system ของ bootstrap นั้นเป็นแบบ 12 แล้วเท่ากับว่า 2 คอลัมน์นี้จะกางเต็ม container พอดี

## Offsetting columns

ในบางครั้ง ถ้าต้องการเพิ่มระยะห่างระหว่างคอลัมน์ เราสามารถทำได้ด้วยการเพิ่ม class ที่ชื่อ “offset\*” เข้าไป โดยจะหมายถึงจำนวนของคอลัมน์ที่จะกลายมาเป็นระยะห่างดังนี้

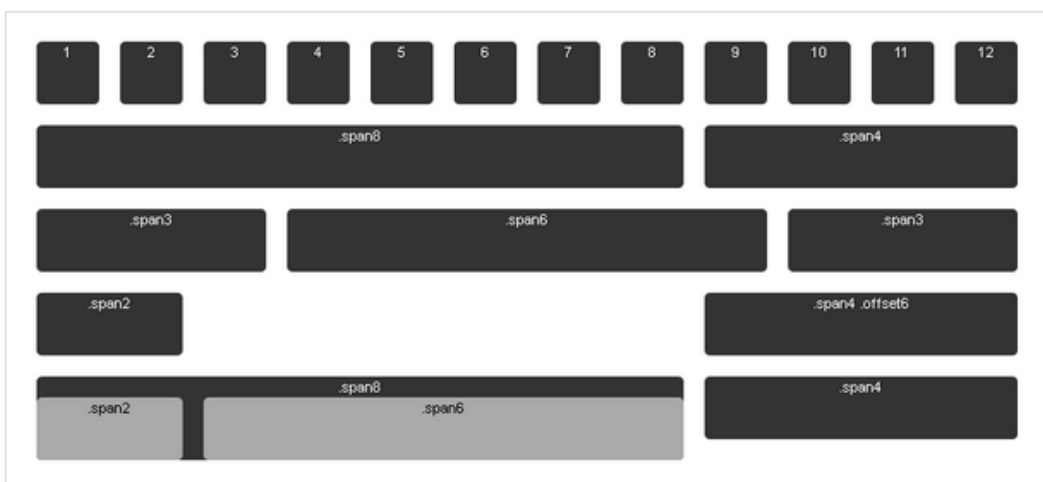
```
<div class="row">
  <div class="span2">Logo</div>
  <div class="span4 offset6">Search</div>
</div>
```

เมื่อพีริวิว จะพบว่าทั้ง 2 คอลัมน์นี้ยังคงกางเต็ม container อยู่ เนื่องจาก  $2 + 4 + 6 = 12$

## Nesting columns

```
<div class="row">
  <div class="span8">
    Main
    <div class="row">
      <div class="span2">Menu</div>
      <div class="span6">Content</div>
    </div>
  </div>
  <div class="span4">Sidebar</div>
</div>
```

นอกจากนั้นยังสามารถสร้างคอลัมน์ซ้อนภายในคอลัมน์ได้ด้วยตามตัวอย่างนี้ จะเห็นว่าได้สร้างคอลัมน์ “span2” กับ “span6” ซ้อนข้างในคอลัมน์ “span8” ซึ่งคอลัมน์ที่อยู่ภายในทั้ง 2 คอลัมน์ จะกางเต็มคอลัมน์ข้างนอกพอดีเนื่องจาก  $2 + 6 = 8$



รูปที่ ก.2 รูปแบบของการจัดแบ่ง Column ด้วย Grid System

## Class Container Fluid

Bootstrap ได้เตรียม fluid grid system มาให้ด้วย fluid grid system เป็นการที่ใช้พื้นที่ของ browser แบบเต็มพื้นที่ ด้วยการเปลี่ยน จาก container ปกติมาเป็น container-fluid ในส่วนของ class ใน tag <div> ดังนี้

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

```
<div class="row-fluid">
  <div class="span8">
    <div class="row-fluid">
      <div class="span3">Menu</div>
      <div class="span9">Content</div>
    </div>
  </div>
  <div class="span4">Sidebar</div>
</div>
```

เมื่อเปลี่ยนเป็น fluid grid system แล้วต้องเปลี่ยน Class ให้เป็น row-fluid เพื่อให้สอดคล้องกันด้วย



รูปที่ ก.3 ตัวอย่างการกำหนดให้แสดงเป็น fluid grid system ให้แสดงเต็มพื้นที่ของ Web Browser

## Responsive Design

ถ้าต้องการทำให้เป็น Responsive Web นั้น Bootstrap ได้เตรียม features รองรับไว้แล้ว แต่ต้องเปิดใช้งาน responsive features ด้วยการเพิ่มโค้ดด้านล่างนี้ไว้ที่ <head>

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<link href="css/bootstrap-responsive.min.css" rel="stylesheet">
```

สไตลชีท “bootstrap-responsive.min.css” จะเข้ามาช่วยจัดการในเรื่องของ responsive แบบอัตโนมัติ ไม่ว่าจะเป็นขนาดของ margin และ padding รวมไปถึงการเรียงกันของ grid ที่ viewport ขนาดต่างๆ หรือต้องการให้ html elements ซ่อน/แสดง บนอุปกรณ์แบบไหน เราก็สามารถทำได้ง่ายๆ ด้วยการเพิ่ม class ที่ Bootstrap เตรียมไว้ให้ตามตารางนี้

Class	Phones (767px and below)	Tablets (979px to 768px)	Desktops (Default)
.visible-phone	Visible	Hidden	Hidden
.visible-tablet	Hidden	Visible	Hidden
.visible-desktop	Hidden	Hidden	Visible
.hidden-phone	Hidden	Visible	Visible
.hidden-tablet	Visible	Hidden	Visible
.hidden-desktop	Visible	Visible	Hidden

รูปที่ ก.4 ตารางแสดง Class ที่จะซ่อน หรือ แสดงเนื้อหาต่างๆ ในแต่ละอุปกรณ์

การใช้งานแค่ให้เพิ่ม class เข้าไปที่ html element ที่ต้องการจะ ซ่อน/แสดง ดังนี้

```
<div class="row">
  <div class="span3">Secondary</div>
  <div class="span6">Primary</div>
  <div class="span3 visible-desktop">Sidebar</div>
</div>
```

จากตัวอย่างนี้จะได้ว่า คอลัมน์ “Sidebar” จะแสดงต่อเมื่อได้เปิดผ่านทาง Desktop เท่านั้น ในขณะที่คอลัมน์อื่นๆ จะแสดงในทุกๆ อุปกรณ์

## Base CSS ของ Bootstrap

หลังจากที่เราได้แบ่งหน้าเว็บออกเป็นส่วนๆ โดยใช้ grid system แล้ว จะลองนำส่วนอื่นๆ ที่ Bootstrap เตรียมให้มาใช้งาน สำหรับ Base CSS นั้น จะเป็นการตกแต่งสไตล์ให้กับ html elements พื้นฐาน ไม่ว่าจะเป็น tables, forms, buttons, icons และ images สามารถเรียกเรียกใช้ได้ง่าย ถ้าต้องการจะใช้สไตล์สำหรับ <table> ที่ Bootstrap เตรียมไว้ให้ ให้เพิ่ม class “table” ไว้ที่ <table>

```
<table class="table">
  <thead>
    <tr>
      <th>#</th>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Username</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Mark</td>
      <td>Otto</td>
      <td>@mdo</td>
    </tr>
    <tr>
      <td>2</td>
      <td>Jacob</td>
      <td>Thornton</td>
      <td>@fat</td>
    </tr>
    <tr>
      <td>3</td>
      <td>Larry</td>
      <td>the Bird</td>
      <td>@twitter</td>
    </tr>
  </tbody>
</table>
```

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

รูปที่ ก.5 ตัวอย่างการใช้งาน <table> ค่า default ของ Bootstrap

หากไม่พอใจสไตล์แบบ default ที่ Bootstrap ให้มานั้น สามารถปรับแต่งเพิ่มเติมได้ด้วยการเพิ่ม optional classes ต่อเข้าไปที่ <table> ดังนี้

- .table-striped** – ทำให้สีพื้นหลังของแต่ละแถวสลับกันไปเรื่อยๆ
- .table-bordered** – ใส่ borders และ rounded corners ให้กับ table
- .table-hover** – เมื่อ hover ที่ row ไหน จะไฮไลต์สีพื้นหลังที่ row นั้นๆ
- .table-condensed** – ลด padding ของแต่ละ cell ลงครึ่งหนึ่ง

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

รูปที่ ก.6 ตัวอย่างการใช้งาน table-striped ใน <table>



## Components ของ Bootstrap

Bootstrap ได้เตรียมสไตลส์สำหรับสิ่งใช้บ่อยๆ เช่น navigation, breadcrumbs รวมไปถึง pagination มาด้วย Components จะต่างกับ Base CSS ตรงที่มันจะเป็นกลุ่มของ html elements ต่างๆ ที่มารวมตัวกันเพื่อทำหน้าที่ใดหน้าที่หนึ่งโดยเฉพาะ ส่วนวิธีใช้ให้ copy โค้ด html ของ Component ไปวางในตำแหน่งที่ต้องการจะให้แสดงผล ตัวอย่างการใช้ Components ที่มีชื่อว่า “Navs” ถ้าปกติอยากจะสร้าง navigation ขึ้นมาสักอันหนึ่ง โค้ด html จะเป็นดังนี้

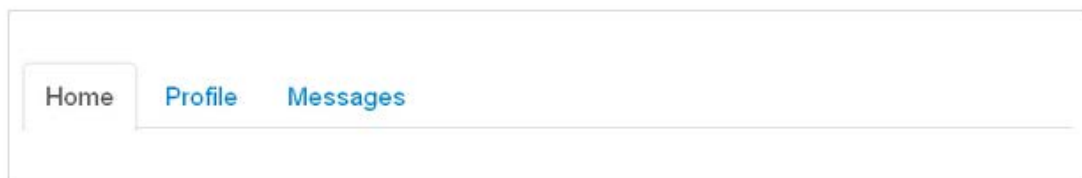
```
<ul>
  <li>
    <a href="#">Home</a>
  </li>
  <li>
    <a href="#">Profile</a>
  </li>
  <li>
    <a href="#">Messages</a>
  </li>
</ul>
```

หากต้องการจะใช้สไตลส์ของ navigation นี้ ให้ทำการเพิ่ม class ที่ชื่อ “nav” เข้าไปที่ <ul>

```
<ul class="nav">
  .
  .
  .
</ul>
```

ซึ่ง Bootstrap ได้มีรูปแบบของ navigation ให้เลือกดังนี้

- .nav-tabs** – navigation แบบ tab
- .nav-pills** – navigation แบบ button
- .nav-lists** – navigation แบบ list



รูปที่ ก.7 ตัวอย่างการใช้งาน nav-tabs ใน <nav>