

Written by Dr. Thapanapong Rukkanchanunt

04 Markup Languages 3

OUTLINE

- Intro to CSS
- What's new in CSS3
- Syntax, Box Model, Link, and Navigation
- Media Query
- Responsive Web Design

What is CSS?

- CSS ย่อมาจาก Cascading Style Sheet คือภาษาที่ใช้อธิบายการแสดงผลของไฟล์ประเภท HTML บนหน้าจอ
- เราสามารถเรียกใช้งาน CSS ในไฟล์ HTML ได้หลากหลายรูปแบบเช่น
 - สร้างไฟล์ .css แล้วเรียกใช้งานด้วย `<link>` ในไฟล์ HTML
 - สร้างไฟล์ .css แล้วเชื่อมโยงด้วย `@component styleUrls` ใน Angular
 - เขียน CSS ภายใน `<style>` ในไฟล์ HTML
 - เขียน CSS ใน Attribute style ของ HTML Tag
- ในหนึ่งไฟล์ HTML เราสามารถมีได้หลาย Style

Road to CSS

- จุดมุ่งหมายของ HTML คือการอธิบายลักษณะของข้อมูลไม่ใช่การจัดรูปแบบของข้อมูล
- `<h1>` ใช้อธิบายว่าข้อความนี้เป็นหัวข้อ `<p>` ใช้อธิบายว่าข้อความนี้เป็นย่อหน้า
- ในสมัยที่ HTML 3.2 เริ่มใช้ครั้งแรก มีการเพิ่ม `` และ Attribute `color` ส่งผลให้
ผู้พัฒนาเว็บปวดหัวกันมาก เพราะต้องกำหนดรูปแบบฟอนต์และสีในทุก ๆ หน้า
- W3C สร้าง CSS ขึ้นมาเพื่อแก้ไขปัญหาดังกล่าว

What's new in CSS3?

- เวอร์ชันล่าสุดของ CSS คือเวอร์ชัน 3 เริ่มใช้ตั้งแต่ปี 2005
- โฟกัสที่การแสดงผลบนมือถือ รองรับการแสดงผลแบบ Responsive
- รองรับ Web Font นอกเหนือจาก Web Safe Font สามารถใช้งาน Font ที่ลงในเครื่อง Server หรือเรียกผ่าน Google Fonts, Typecast โดยตรงได้เลย
- เพิ่ม 2D/3D Animation, Transformation, และ Transition
- รองรับสีแบบ RGBA, HSL, HSLA และ Gradient Color อีกทั้งการกำหนด Opacity

CSS Syntax

- CSS ประกอบไปด้วย Selector และ Declaration Block
- Selector ชี้ไปยัง HTML Tag ที่ต้องการจัดรูปแบบ
- Declaration Block ประกอบไปด้วย Declaration ย่อย ๆ คั่นกันด้วย Semicolon
 - Declaration ประกอบไปด้วยชื่อ Property และค่าของ Property



Selector

- เราสามารถกำหนด Selector ได้หลายแบบ
 - Element Selector ใช้ชื่อ Tag โดยตรง เช่น `p {color:red}`
 - ID Selector ใช้ `#` นำหน้าชื่อไอดี เช่น `#red {color:red}` (ส่งผลต่อ `id="red"`)
 - Class Selector ใช้ `.` นำหน้าชื่อคลาส เช่น `.red {color:red}` (ส่งผลต่อ `class="red"`)
 - รวมกันระหว่าง Element และ Class เช่น `p.red {color:red}` (ส่งผลต่อ `<p class="red">`)
 - กำหนดทุก Tag ด้วย `*` เช่น `* {color:red}`
- ถ้าอยากกำหนดหลาย Selector ให้มีรูปแบบเดียวกัน เราสามารถใช้ `comma` คั่นแต่ละ Selector ได้เช่น `p, td, div {color:red}`

CSS Box Model

- เราสามารถมอง HTML ในรูปแบบ Box
- เราสามารถกำหนด Box ได้ดังนี้

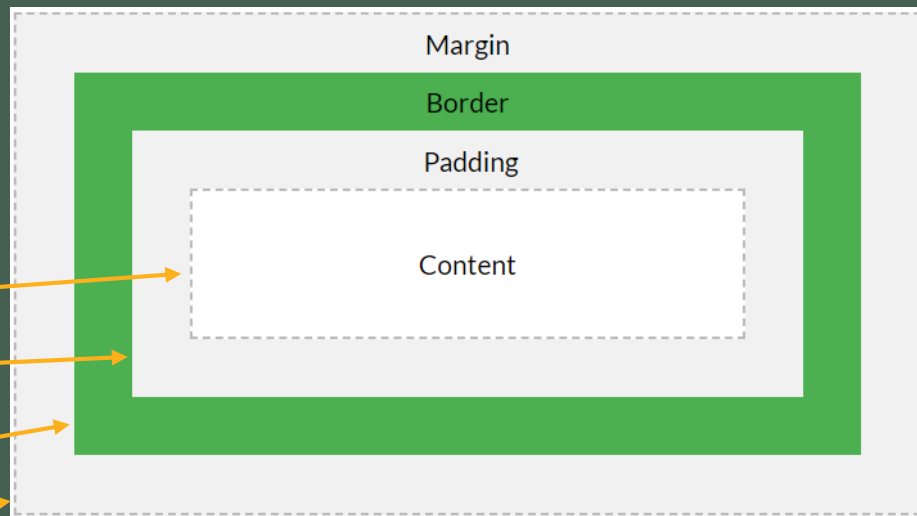
```
div {
```

```
width: 300px; height: 200px;
```

```
padding: 50px;
```

```
border: 15px solid green;
```

```
margin: 20px; }
```



Style Link State

- นอกการกำหนดรูปแบบของ Link โดยใช้ Element Selector a แล้ว เรายังสามารถกำหนดรูปแบบของ Link ในสถานะต่าง ๆ ได้
 - a:link กำหนด Link ปกติที่ยังไม่มีการคลิก
 - a:visited กำหนด Link ที่เคยคลิกแล้ว
 - a:hover กำหนด Link ตอนเอาเมาส์ไปวาง (แต่ยังไม่ได้คลิก)
 - a:active กำหนด Link ในขณะที่เกิดการคลิก
- ลำดับในการกำหนดรูปแบบมีความสำคัญ active ต้องมาหลัง hover ต้องมาหลัง link และ visited

Navigation with CSS

- Navigation คือ List ของ Link ซึ่งเราจะใช้ ``, `` และ `<a>` ในการกำหนดลักษณะ
- CSS มาตรฐานของการสร้าง Navigation คือการไม่แสดง bullet point และล้างรูปแบบ Default ใน margin และ padding

```
ul {  
  list-style-type: none;  
  margin: 0;  
  padding: 0;  
}
```



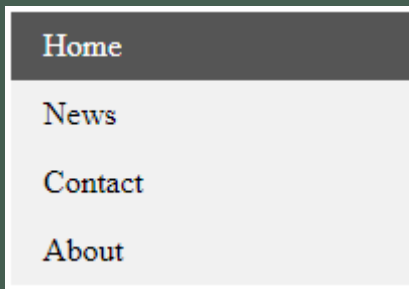
```
<ul>  
  <li><a href="#home">Home</a></li>  
  <li><a href="#news">News</a></li>  
  <li><a href="#contact">Contact</a></li>  
  <li><a href="#about">About</a></li>  
</ul>
```



[Home](#)
[News](#)
[Contact](#)
[About](#)

Vertical Navigation

- ถ้าเป็น Link เราจะคลิกได้แค่เฉพาะส่วนที่เป็นตัวอักษร แต่ถ้าเรากำหนด `display: block` ให้กับ `li` และ `a` จะทำให้เราสามารถคลิกพื้นที่รอบ ๆ ตัวอักษรได้ด้วย
- เมื่อนำสิ่งนี้มารวมกับการกำหนดสีพื้นหลังของ `a:hover` จะทำให้เราได้สิ่งที่คล้ายกับ Navigation Bar ที่มักจะพบในเว็บแอปต่าง ๆ



```
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  width: 200px;
  background-color: #f1f1f1;
}

li a {
  display: block;
  color: #000;
  padding: 8px 16px;
  text-decoration: none;
}

/* Change the link color on hover */
li a:hover {
  background-color: #555;
  color: white;
}
```

Full Height Fixed Width Navigation

- หากเนื้อหาในเว็บแอปของเรามีเยอะ (แสดงสินค้าหลายสิบชิ้น) แต่อยากให้ Navigation ไม่หายไปตอนเราเลื่อนลง เราสามารถทำได้โดยการกำหนด Fixed Position (ไม่ให้ขยับ), Height (กินพื้นที่แนวตั้งทั้งหมด) และ Overflow (เพิ่ม Scroll Bar เฉพาะตอนที่เนื้อหาเยอะ)

```
ul {  
  list-style-type: none;  
  margin: 0;  
  padding: 0;  
  width: 25%;  
  background-color: #f1f1f1;  
  position: fixed;  
  height: 100%;  
  overflow: auto;  
}
```

The screenshot shows a web application with a side navigation menu on the left and a main content area on the right. The navigation menu is a vertical list of links: Home (highlighted in green), News, Contact, and About. The main content area is titled "Fixed Full-height Side Nav" and contains several paragraphs of text, including instructions to scroll and observe the navigation menu's behavior. The navigation menu remains fixed in position and height as the page content is scrolled.

Home
News
Contact
About

Fixed Full-height Side Nav

Try to scroll this area, and see how the sidenav sticks to the page

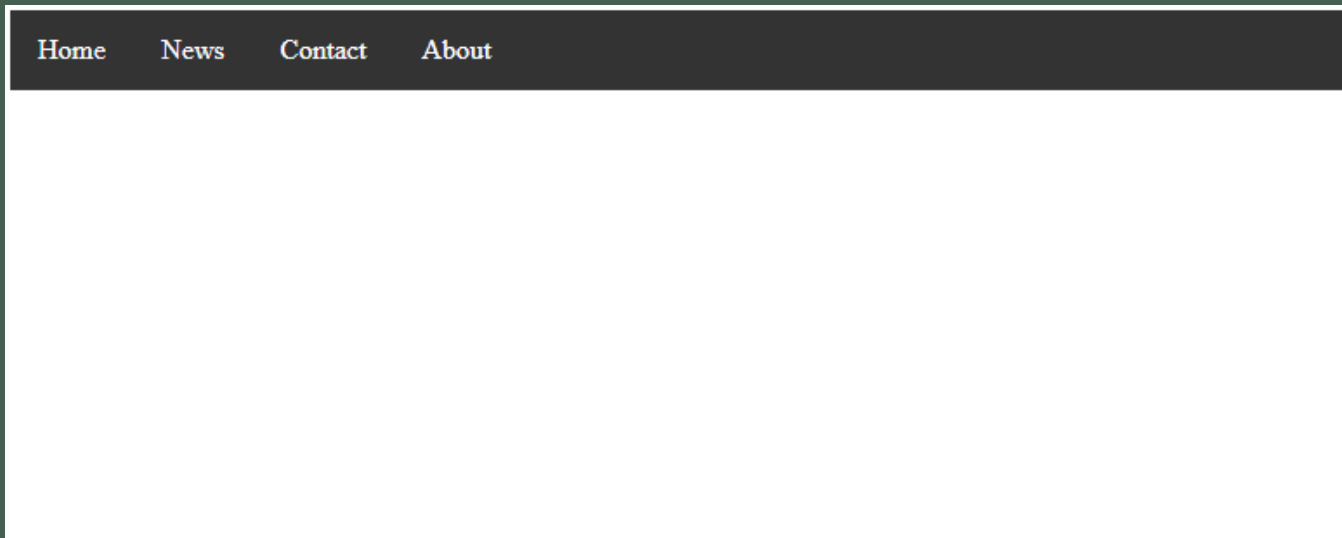
Notice that this div element has a left margin of 25%. This is because the side navigation is set to 25% width. If you remove the margin, the sidenav will overlay/sit on top of this div.

Also notice that we have set overflow:auto to sidenav. This will add a scrollbar when the sidenav is too long (for example if it has over 50 links inside of it).

Some text..
Some text..
Some text..
Some text..
Some text..
Some text..
Some text..

Horizontal Navigation

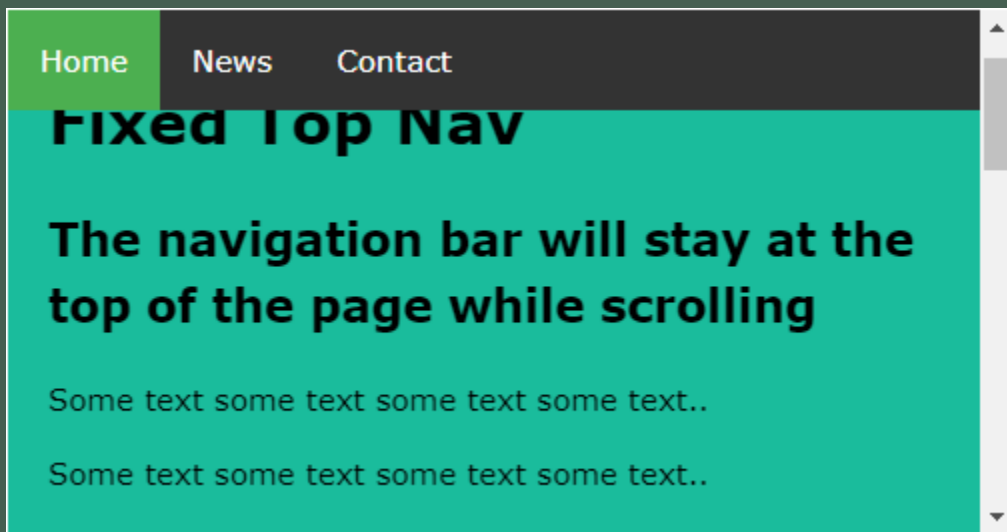
- หากเราต้องการทำ Navigation แนวนอน เราต้องเปลี่ยน display:block มาเป็น inline หรือกำหนด float เพิ่มเติม



```
ul {  
  list-style-type: none;  
  margin: 0;  
  padding: 0;  
  overflow: hidden;  
  background-color: #333;  
}  
  
li {  
  float: left;  
}  
  
li a {  
  display: block;  
  color: white;  
  text-align: center;  
  padding: 14px 16px;  
  text-decoration: none;  
}  
  
li a:hover {  
  background-color: #111;  
}
```

Fixed Top Fixed Bottom

- ถ้าอยากให้ Navigation อยู่ติดด้านบนหรือด้านล่าง ให้เรากำหนด position และ top หรือ bottom ตามความเหมาะสม



```
ul {  
  position: fixed;  
  top: 0;  
  width: 100%;  
}
```

Dropdown

- เราสามารถใช้ CSS ในการสร้างเมนูย่อยหรือ Dropdown จาก Tag ที่มีอยู่

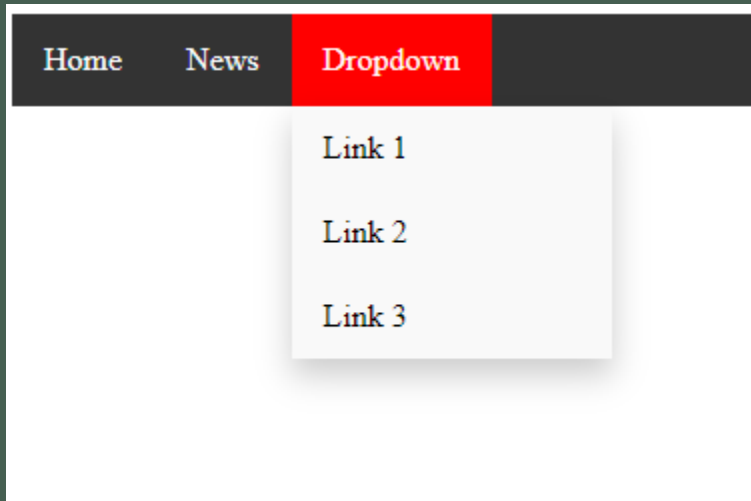
```
.dropdown {  
  position: relative;  
  display: inline-block;  
}  
  
.dropdown-content {  
  display: none;  
  position: absolute;  
  background-color: #f9f9f9;  
  min-width: 160px;  
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);  
  padding: 12px 16px;  
  z-index: 1;  
}  
  
.dropdown:hover .dropdown-content {  
  display: block;  
}
```

```
<div class="dropdown">  
  <span>Mouse over me</span>  
  <div class="dropdown-content">  
    <p>Hello World!</p>  
  </div>  
</div>
```

Mouse over me

Hello World!

```
<ul>
  <li><a href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li class="dropdown">
    <a href="#" class="dropbtn">Dropdown</a>
    <div class="dropdown-content">
      <a href="#">Link 1</a>
      <a href="#">Link 2</a>
      <a href="#">Link 3</a>
    </div>
  </li>
</ul>
```



```
li.dropdown {
  display: inline-block;
}

.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f9f9f9;
  min-width: 160px;
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
  z-index: 1;
}

.dropdown-content a {
  color: black;
  padding: 12px 16px;
  text-decoration: none;
  display: block;
  text-align: left;
}

.dropdown-content a:hover {background-color: #f1f1f1;}

.dropdown:hover .dropdown-content {
  display: block;
}
```


Transition

- เราสามารถเปลี่ยนค่าของ Property ใน CSS ตามระยะเวลาที่กำหนดได้

```
div {  
  width: 100px;  
  height: 100px;  
  background: red;  
  transition: width 2s, height 4s, background 3s;  
}  
  
div:hover {  
  width: 300px;  
  height: 300px;  
  background: blue;  
}
```



Note: This example does not work in Internet Explorer 9 and earlier versions.

Media Query

- Media Query สามารถเข้าถึงข้อมูลความสามารถของอุปกรณ์ที่ใช้เข้าเว็บแอป
 - ขนาดของ Viewport
 - ขนาดของอุปกรณ์
 - แนวตั้งหรือแนวนอน (Tablet มือถือ)
 - ความละเอียดของหน้าจอ
- Media Query จึงเหมาะแก่การกำหนดรูปแบบการแสดงผลในมือถือที่มีหลากหลายขนาดและความละเอียดหน้าจอ

```
@media not|only mediatype and (expressions) {  
  CSS-Code;  
}
```

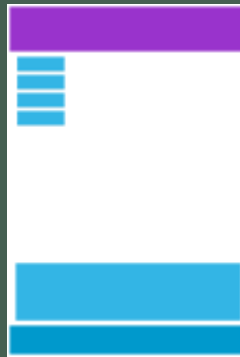
Media Query Example

```
body {  
  background-color: pink;  
}  
  
@media screen and (min-width: 480px) {  
  body {  
    background-color: lightgreen;  
  }  
}
```



Responsive Web Design

- Responsive Web Design คือการทำให้เว็บแอปที่เราสร้างขึ้นมาดูดี (เป็นไปตามที่คาดหวัง) ในทุกอุปกรณ์ที่จะเข้ามาใช้งาน
- Responsive Web Design ใช้แค่ HTML และ CSS โดยไม่มี JavaScript เข้ามาเกี่ยวข้อง



Viewport

- Viewport คือพื้นที่ในหน้าจอที่ผู้ใช้สามารถมองเห็นได้
- เราสามารถตั้งค่า Viewport โดยการใช้อะแท็ก <meta>

```
<meta name="viewport"
content="width=device-width, initial-
scale=1.0">
```



Caution with Viewport

- ผู้ใช้มักจะชินกับการเลื่อนในแนวตั้ง หากผู้ใช้จำเป็นต้องชมหน้าจอออก หรือเลื่อนในแนวนอน อาจจะทำให้แอปของเรามีภาพลักษณะที่ไม่ดี
- หลีกเลี่ยงการใช้ Element ที่มีการกำหนดขนาดแบบ Fixed และใหญ่กว่าพื้นที่หน้าจอ
- กำหนดขนาดแบบ Percentage มากกว่าการกำหนดขนาดในหน่วย pixel
- ควรใช้ Media Query ควบคู่ไปกับขนาดของ Viewport
- ควรดีไซน์สำหรับมือถือก่อน แล้วค่อยปรับมาเป็นอุปกรณ์ขนาดใหญ่ขึ้นมา

Grid View

- หน้าจอเว็บมักถูกแบ่งออกเป็นคอลัมน์ย่อย ๆ จำนวน 12 คอลัมน์
- คอลัมน์เหล่านี้มีความกว้างรวมกัน 100% ซึ่งจะปรับขนาดตามหน้าจออุปกรณ์
- ถ้าหากเรากำหนดขนาดของ Element ในหน่วยของคอลัมน์จะทำให้เว็บดูเป็นระเบียบ
- ถ้าต้องการใช้ Responsive Grid-view ในเพิ่มค่าใน box-sizing แล้วกำหนดขนาดของแต่ละคอลัมน์

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

```
* {  
  box-sizing: border-box;  
}
```

```
.col-1 {width: 8.33%;}  
.col-2 {width: 16.66%;}  
.col-3 {width: 25%;}  
.col-4 {width: 33.33%;}  
.col-5 {width: 41.66%;}  
.col-6 {width: 50%;}  
.col-7 {width: 58.33%;}  
.col-8 {width: 66.66%;}  
.col-9 {width: 75%;}  
.col-10 {width: 83.33%;}  
.col-11 {width: 91.66%;}  
.col-12 {width: 100%;}
```

Responsive Web Design with Media Query

- ใช้ Media Query ในการปรับเปลี่ยนขนาดของคอลัมน์ตามอุปกรณ์

```
/* For desktop: */
.col-1 {width: 8.33%;}
.col-2 {width: 16.66%;}
.col-3 {width: 25%;}
.col-4 {width: 33.33%;}
.col-5 {width: 41.66%;}
.col-6 {width: 50%;}
.col-7 {width: 58.33%;}
.col-8 {width: 66.66%;}
.col-9 {width: 75%;}
.col-10 {width: 83.33%;}
.col-11 {width: 91.66%;}
.col-12 {width: 100%;}

@media only screen and (max-width: 768px) {
  /* For mobile phones: */
  [class*="col-"] {
    width: 100%;
  }
}
```



More Information

- <https://www.w3schools.com/css/default.asp>