



# CSS (CASCADING STYLE SHEETS)

030523124 & 030523224 : Web Application Development  
Dr. Phollakrit Wongsantisuk

# CSS (Cascading Style Sheets)

---

CSS เป็นคำสั่งที่ใช้ในการจัดรูปแบบเอกสารเว็บเพจในลักษณะต่างๆ เช่น สี พื้นหลัง เส้นขอบ ขนาด โครงร่าง หรือตัวอักษร เป็นต้น แม้ว่าใน HTML นั้นจะมีทั้งแท็กและแอตทริบิวต์บางส่วนที่สามารถใช้จัดรูปแบบเอกสารอยู่แล้ว แต่ไม่ครอบคลุมการใช้งานครบทุกกรณี นอกจากนี้ยังขาดความยืดหยุ่นและไม่สามารถจัดรูปแบบในลักษณะที่ซับซ้อนได้ ซึ่งในปัจจุบันเราจะเน้นการจัดรูปแบบด้วย CSS มากกว่าที่จะใช้แท็กแอตทริบิวต์ของ HTML ดังนั้น จึงเกิดการแยกหน้าที่กันอย่างชัดเจนคือ HTML ใช้สำหรับการจัดโครงสร้างของเว็บเพจ ส่วน CSS ใช้สำหรับการจัดรูปแบบของเว็บเพจ นอกจากนี้ยังเป็นพื้นฐานไปสู่การใช้ Bootstrap

# พรีออปเพอร์เตอร์ของ CSS

พรีออปเพอร์เตอร์ (Property) ของ CSS คือสิ่งที่ใช้กำหนดคุณลักษณะต่างๆ ของอีลิเมนต์ HTML เช่น ทำให้ข้อความเป็นสีฟ้า + พื้นหลังเป็นสีเหลือง หรือทำให้ฟอนต์มีขนาดตามที่ต้องการ เป็นต้น สำหรับหลักการเบื้องต้นเกี่ยวกับพรีออปเพอร์เตอร์ที่เราควรทราบ มีดังนี้

- **การกำหนดค่าของพรีออปเพอร์เตอร์ของ CSS**

รูปแบบการกำหนดค่าของพรีออปเพอร์เตอร์จะมี 2 ส่วนคือ ชื่อพรีออปเพอร์เตอร์ และค่าของมัน โดยคั่นระหว่างชื่อและค่าด้วยเครื่องหมาย Colon (:) คือ

ชื่อพรีออปเพอร์เตอร์ : ค่าที่กำหนด

กรณีที่มีมากกว่า 1 พรีออปเพอร์เตอร์ ให้คั่นแต่ละพรีออปเพอร์เตอร์ด้วยเครื่องหมาย Semicolon (;) ในรูปแบบดังต่อไปนี้

พรีออปเพอร์เตอร์-1 : ค่าที่กำหนด; พรีออปเพอร์เตอร์-2 : ค่าที่กำหนด; พรีออปเพอร์เตอร์-3 : ค่าที่กำหนด; ...

# พรีออปเพอร์เตอร์ของ CSS

---

แนวทางการกำหนดพรีออปเพอร์เตอร์ เช่น

```
color : white; background-color : black; font-size : 12pt;
```

ถ้าไม่เขียนรวมไว้ในบรรทัดเดียวกัน เราอาจเขียนแยกกันไว้คนละบรรทัดกันก็ได้ เนื่องจากมีเครื่องหมาย ; คั่นระหว่างพรีออปเพอร์เตอร์อยู่แล้ว เช่น

```
color : white;  
background-color : black;  
font-size : 12pt;
```

# พร็อพเพอร์ตี้ของ CSS

- พร็อพเพอร์ตี้พื้นฐานที่ควรรู้จัก

เราควรรู้จักกับพร็อพเพอร์ตี้พื้นฐานบางส่วนของ CSS ล่วงหน้าไว้ก่อน เช่น

color	กำหนดสีข้อความ โดยในเบื้องต้นให้ระบุค่าเป็นชื่อสีในภาษาอังกฤษไปก่อน เช่น green, blue, red, yellow, black, white, gray, pink, brown, orange เป็นต้น
background-color	กำหนดสีพื้นหลัง โดยระบุค่าเป็นชื่อสีเช่นเดียวกับ color
font-size	กำหนดขนาดของฟอนต์ ซึ่งในเบื้องต้นให้กำหนดค่าเป็นตัวเลข ตามด้วยหน่วย pt หรือ px โดยตัวเลขและหน่วยให้เขียนติดกัน เช่น font-size: 10pt หรือ font-size: 12px
font-weight	กำหนดขนาดน้ำหนักของฟอนต์ มักใช้เมื่อต้องการทำให้ข้อความเป็นตัวหนา โดยระบุค่าเป็น bold เช่น font-weight: bold
font-style	กำหนดรูปแบบฟอนต์ โดยทั่วไปมักกำหนดเมื่อต้องการทำให้ข้อความเป็นตัวเอียง โดยระบุค่า italic เช่น font-style: italic
text-decoration	สำหรับตกแต่งข้อความ มักใช้เมื่อต้องการขีดเส้นใต้ โดยระบุค่าเป็น underline เช่น text-decoration: underline

# ขอบเขตของการกำหนดสไตล์แบบต่างๆ

---

การกำหนดสไตล์แบ่งออกเป็น 3 แบบคือ

1. สไตล์แบบ Inline จะมีผลเฉพาะกับอิลิเมนต์ที่กำหนดสไตล์ให้กับมัน
2. สไตล์แบบ Internal จะมีผลกับทุกอิลิเมนต์ภายในเพจนั้น ที่ตรงกับข้อกำหนด
3. สไตล์แบบ External นำเข้าสไตล์จากไฟล์ภายนอก และจะมีผลกับทุกอิลิเมนต์ภายในเพจนั้น ที่ตรงกับข้อกำหนด

สำหรับรายละเอียดการกำหนดสไตล์แต่ละแบบมีดังต่อไปนี้

- **การกำหนดสไตล์แบบ Inline**

สไตล์แบบ Inline คือการกำหนดรูปแบบเอาไว้ที่แท็กซึ่งเราต้องการจัดรูปแบบ โดยระบุพรีอเพอร์ตีและค่าของมันให้กับแอตทริบิวต์ style ซึ่งรูปแบบคือ

`<แท็ก style="พรีอเพอร์ตี-1 : ค่าที่กำหนด; พรีอเพอร์ตี-2 : ค่าที่กำหนด; ...">`

# ขอบเขตของการกำหนดสไตล์แบบต่างๆ

...

<body>

<p style="font-size: 18pt; font-weight: bold; color: red">

HTML ใช้สำหรับการจัดโครงสร้างของเว็บเพจ

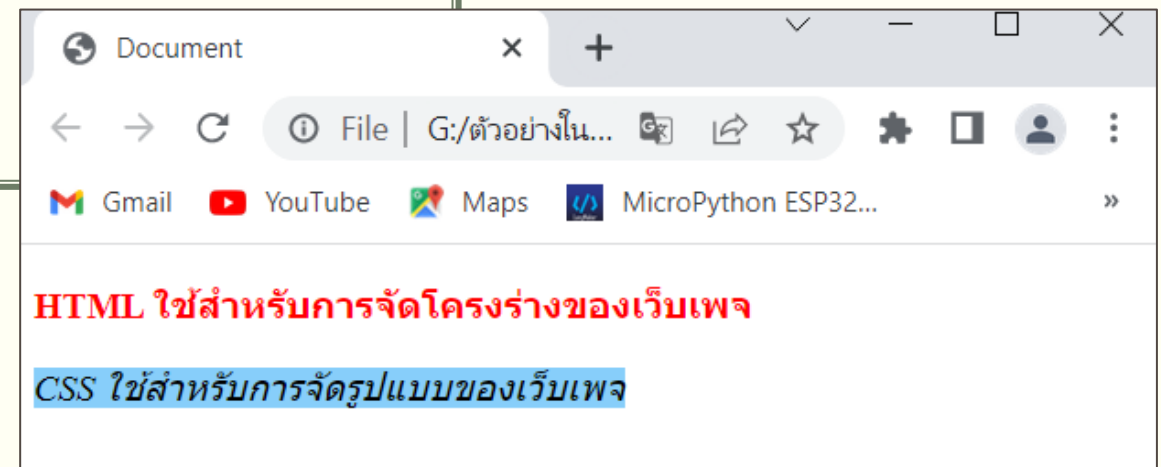
</p>

<span style="background-color: lightskyblue; font-style: italic">

CSS ใช้สำหรับการจัดรูปแบบของเว็บเพจ

</span>

</body>



# ขอบเขตของการกำหนดสไตล์แบบต่างๆ

- **การกำหนดสไตล์แบบ Internal**

สไตล์แบบ Internal จะกำหนดเอาไว้ที่ส่วนหัวของเอกสาร เพื่อให้มีผลกับหลายๆ อิลิเมนต์ โดยไม่ต้องเขียนซ้ำซ้อนกันอีก ดังรูปแบบต่อไปนี้

```
<style>
  ซีเล็คเตอร์ {
    พร็อพเพอร์ตี้-1 : ค่าที่กำหนด;
    พร็อพเพอร์ตี้-2 : ค่าที่กำหนด;
    ...
  }
</style>
```

สิ่งที่ควรรู้เพิ่มเติมเกี่ยวกับสไตล์แบบ Internal คือ

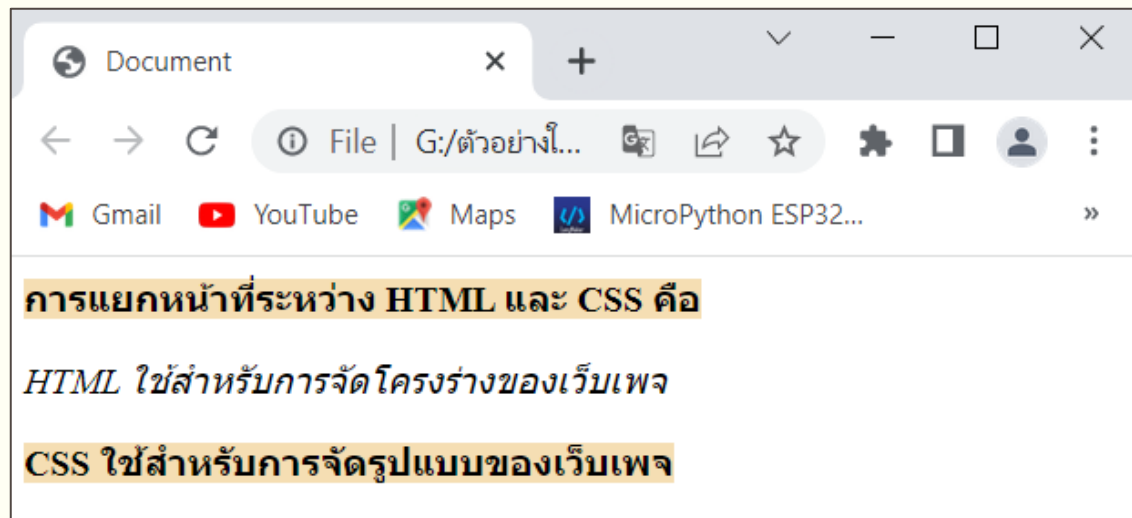
- สไตล์แบบ Internal ต้องกำหนดด้วยแท็ก <style> และให้อยู่ระหว่างแท็ก <head>...</head>
- ซีเล็คเตอร์ (Selector) เป็นตัวที่ใช้กำหนดว่า จะให้สไตล์นั้นมีผลกับอิลิเมนต์ใด โดยมีซีเล็คเตอร์หลายรูปแบบ เช่น Type, ID, Class เป็นต้น
- พร็อพเพอร์ตี้และค่าของมัน ต้องกำหนดไว้ในวงเล็บ { } เสมอ
- เราอาจเขียนพร็อพเพอร์ตี้รวมไว้ในบรรทัดเดียวกัน หรือแยกบรรทัดกันก็ได้



# ขอบเขตของการกำหนดสไตล์แบบต่างๆ

```
<head>
...
<style>
  div { font-size: 12pt; }
  p { font-style: italic; }
  span {
    background-color: wheat;
    font-weight: bold;
  }
</style>
</head>
```

```
<body>
  <div>
    <span>การแยกหน้าที่ระหว่าง HTML และ CSS คือ</span>
    <p>HTML ใช้สำหรับการจัดโครงสร้างของเว็บเพจ</p>
    <span>CSS ใช้สำหรับการจัดรูปแบบของเว็บเพจ</span>
  </div>
</body>
```

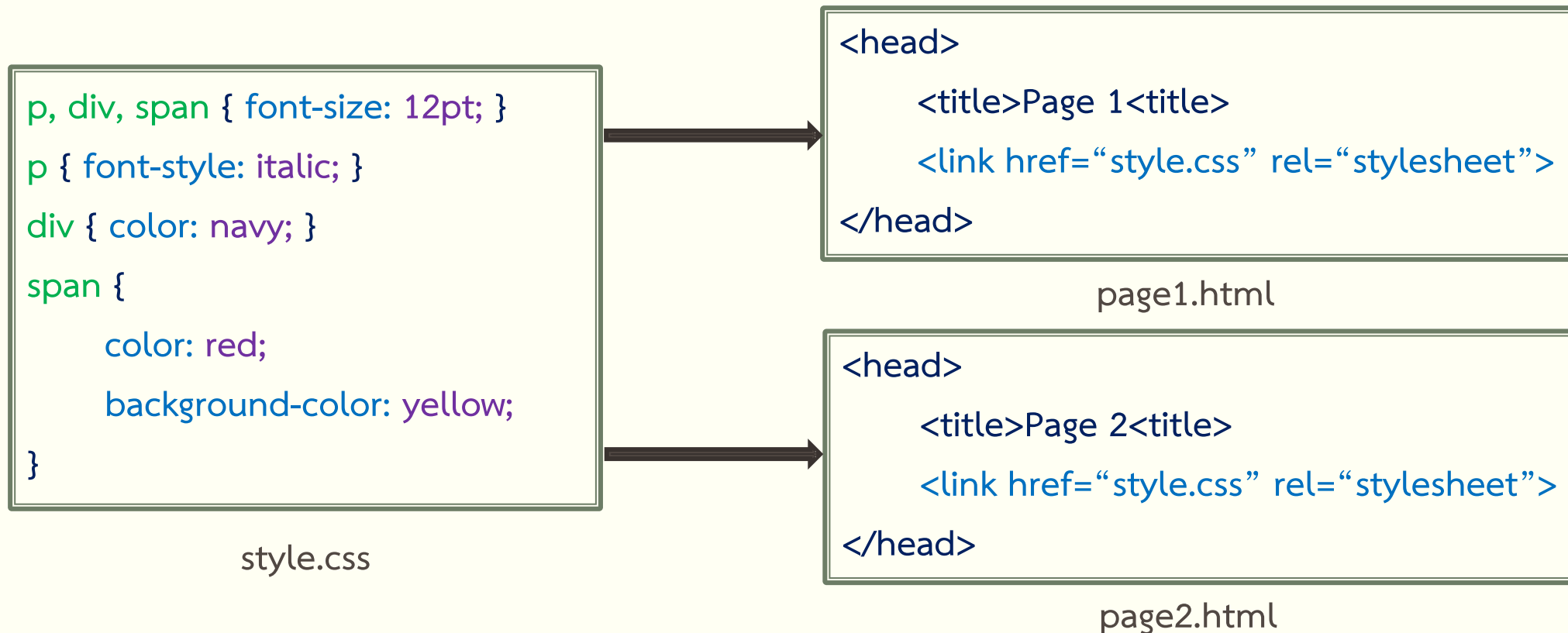


# ขอบเขตของการกำหนดสไตล์แบบต่างๆ

- **การกำหนดสไตล์แบบ External**

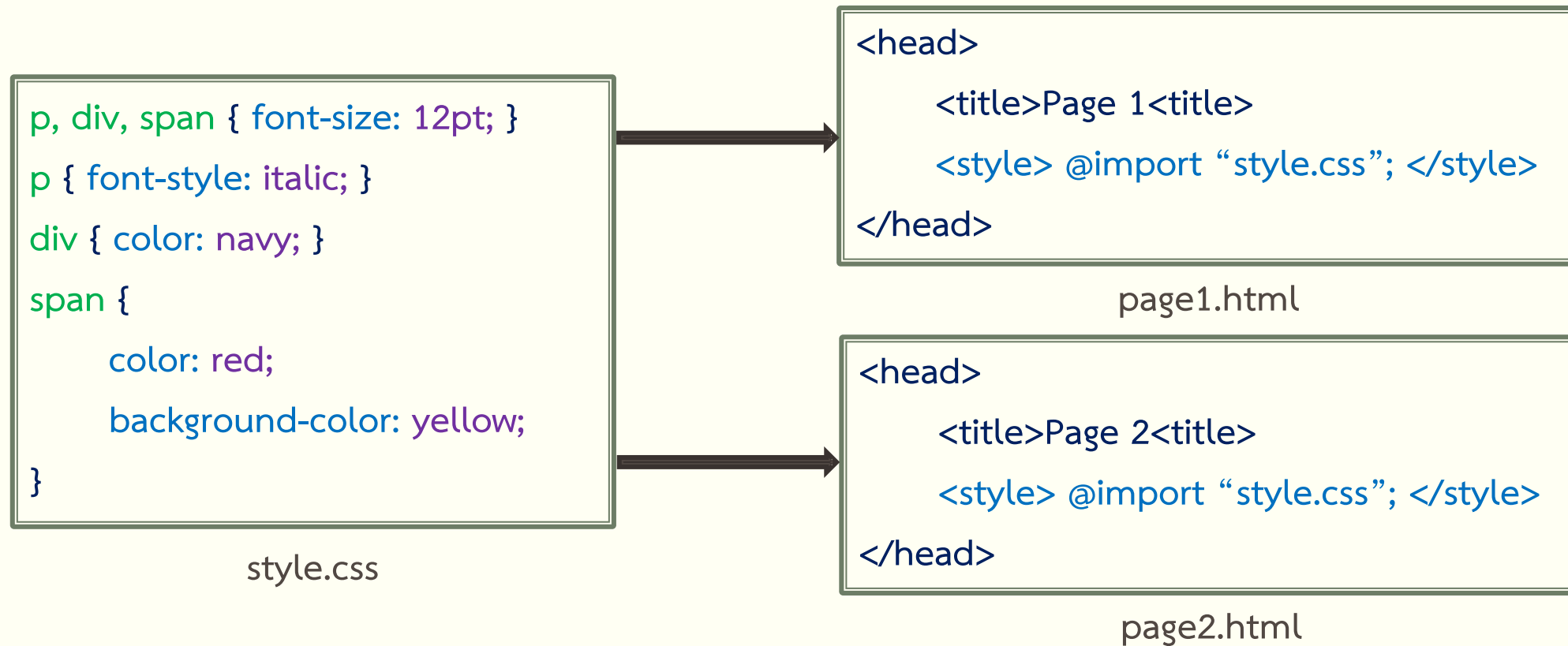
สไตล์แบบ External มักใช้กับกรณีที่เราจำเป็นต้องใช้สไตล์แบบเดียวกันในหลายๆ पेज โดยแยกเฉพาะส่วนสไตล์ที่ซ้ำกันไปสร้างไว้ที่ไฟล์ภายนอก แล้วเมื่อต้องการใช้ที่เพจใด ก็ทำการเชื่อมโยงกับไฟล์นั้น ซึ่งมีให้เลือก 2 วิธีคือ

- วิธีที่ 1 เชื่อมโยงโดยวางแท็ก <link> ไว้ที่ส่วนหัวของเพจซึ่งต้องการใช้สไตล์



# ขอบเขตของการกำหนดสไตล์แบบต่างๆ

- วิธีที่ 2 เชื่อมโยงด้วยคีย์เวิร์ด @import โดยวางไว้ในแท็ก <style> ของเพจที่ต้องการใช้สไตล์



# ขอบเขตของการกำหนดสไตล์แบบต่างๆ

---

สิ่งที่ควรรู้เพิ่มเติมเกี่ยวกับสไตล์แบบ External คือ

- โค้ดของส่วนสไตล์ที่แยกไปสร้างไว้ในไฟล์ภายนอกนั้น ต้องไม่มีแท็ก `<style>` รวมอยู่ด้วย
- ไฟล์ของสไตล์ภายนอก ให้กำหนดส่วนขยายเป็น `.css`
- สำหรับกรณีการใช้แท็ก `<link>` มีแอตทริบิวต์ที่สำคัญ 2 อย่างคือ
  - **href** สำหรับการกำหนดชื่อไฟล์ หากไฟล์ CSS กับไฟล์เว็บเพจที่เรียกใช้งานอยู่คนละไดเรกทอรี ให้ระบุเส้นทางลงไปด้วย เช่น `<link href="../../css/style.css">` ซึ่งกรณีใช้ `@import` ก็ทำเช่นเดียวกัน
  - **rel** สำหรับกำหนดชนิดข้อมูลที่เราจะเชื่อมโยง สำหรับกรณีไฟล์ CSS ให้กำหนดค่าเป็น `rel="stylesheet"`
- เราสามารถเชื่อมโยงไฟล์ CSS จากภายนอกเข้ามาในเพจเดียวกันจำนวนกี่ไฟล์ก็ได้ โดยแต่ละไฟล์ให้กำหนดด้วยแท็ก `<link>` หรือคีย์เวิร์ด `@import` แยกจากกัน เช่น

```
<head>
...
<link href="style.css" rel="stylesheet">
<link href="plugin.css" rel="stylesheet">
</head>
```

# ซีเล็คเตอร์รูปแบบต่างๆ

ซีเล็คเตอร์ (Selector) คือ ตัวเลือกสำหรับการกำหนดสไตล์ โดยซีเล็คเตอร์เป็นการเลือกอิลิเมนต์ที่สอดคล้องกับกฎ (Rule) หรือเงื่อนไข (Condition) เพื่อแสดงผลตามขอบเขตของลักษณะที่ซีเล็คเตอร์นั้นๆ กำหนดไว้เฉพาะนั่นเอง โดยซีเล็คเตอร์มีโครงสร้าง ดังนี้

```
Selector {  
    Property-1 : Value;  
    Property-2 : Value;  
    ...  
}
```

โดยที่

- Selector คือ ชื่อของตัวเลือกสำหรับการกำหนดสไตล์
- Property คือ คุณสมบัติที่ต้องการกำหนดการแสดงผล
- Value คือ ค่าของคุณสมบัติ (Property)
- เครื่องหมาย { } คือ ขอบเขตของ Selector นั้นๆ
- เครื่องหมาย : คือ การกำหนดค่าของ Property
- เครื่องหมาย ; คือ การจบการกำหนดค่าของ Property
- ชุดของ Property: Value; เรียกว่า Declaration

# ซีเล็คเตอร์รูปแบบต่างๆ

---

ซีเล็คเตอร์นั้นมีหลายประเภท โดยสามารถจำแนกเป็น 6 ประเภทหลัก ดังนี้

- 1) ซีเล็คเตอร์อเนกประสงค์ (Universal selector)
- 2) ซีเล็คเตอร์ชนิดอิลิเมนต์ (Element selector)
- 3) ซีเล็คเตอร์ชนิดคลาส (Class selector)
- 4) ซีเล็คเตอร์ชนิดเอกลักษณ์ (ID selector)
- 5) ซีเล็คเตอร์ชนิดแอททริบิวต์ (Attribute selector)
- 6) ซีเล็คเตอร์เทียม (Pseudo-selector)

# ซีเล็กเตอร์เนกประสงค์ (Universal selector)

---

ซีเล็กเตอร์เนกประสงค์ (Universal selector) คือ ตัวเลือกที่ไม่มีการระบุอิลิเมนต์เฉพาะ ทำให้คุณลักษณะของสไตล์ที่อยู่ภายใต้ขอบเขตของซีเล็กเตอร์นี้ส่งผลกระทบต่อทุกๆ อิลิเมนต์

รูปแบบของซีเล็กเตอร์เนกประสงค์ (Universal selector) มีดังนี้

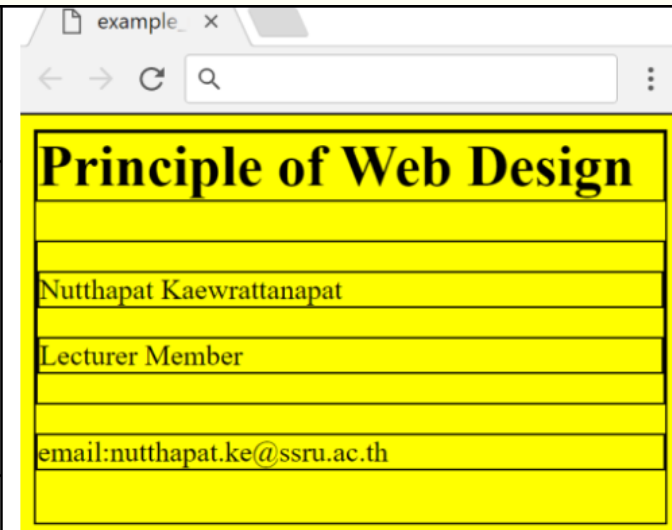
```
* {  
    Property : Value;  
}
```

โดยที่เครื่องหมาย \* (Asterisk) คือ การเลือกทุกๆ อิลิเมนต์ให้แสดงผลตามสไตล์ที่อยู่ภายใต้ขอบเขตของซีเล็กเตอร์นี้

# ซีเล็กเตอร์อเนกประสงค์ (Universal selector)

ตัวอย่างการใช้ซีเล็กเตอร์อเนกประสงค์ (Universal selector)

```
<html>
  <head>
    <meta charset="UTF-8">
    <style type="text/css">
      * {
        background-color: yellow;
        border: 1px solid black;
      }
    </style>
  </head>
  <body>
    <h1>Principle of Web Design</h1>
    <div>
      <p>Nutthapat Kaewrattanapat</p>
      <p>Lecturer Member</p>
    </div>
    <p>email:nutthapat.ke@ssru.ac.th</p>
  </body>
</html>
```





# ซีเล็คเตอร์ชนิดอิลิเมนต์ (Element selector)

---

ซีเล็คเตอร์ชนิดอิลิเมนต์ (Element Selector) คือ ซีเล็คเตอร์ที่ช่วยให้สามารถกำหนดการแสดงผลเฉพาะบางอิลิเมนต์ได้ โดยตัวเลือกชนิดอิลิเมนต์มีข้อกำหนดหลายลักษณะ ได้แก่

1. ซีเล็คเตอร์ชนิดอิลิเมนต์เดี่ยว (Element selector)
2. ซีเล็คเตอร์ชนิดหลายอิลิเมนต์ (Multiple element selector)
3. ซีเล็คเตอร์ชนิดสืบทอดให้ซีเล็คเตอร์ย่อยถัดไป (Descendant selector)
4. ซีเล็คเตอร์ชนิดสืบทอดให้อิลิเมนต์ย่อยแบบเรียงลำดับ (Sub selector)
5. ซีเล็คเตอร์ชนิดสืบทอดให้อิลิเมนต์ภายนอกแบบเรียงลำดับ (Sequential selector)
6. ซีเล็คเตอร์ชนิดสืบทอดให้อิลิเมนต์ถัดไปทั้งหมด (Preceded selector)

# ซีเล็คเตอร์ชนิดอิลิเมนต์ (Element selector)

---

- ซีเล็คเตอร์ชนิดอิลิเมนต์เดี่ยว (Element selector)


ซีเล็คเตอร์ชนิดอิลิเมนต์เดี่ยว (Element selector) คือ การกำหนดการแสดงผลเฉพาะบางอิลิเมนต์เท่านั้น

รูปแบบของซีเล็คเตอร์ชนิดอิลิเมนต์เดี่ยว (Element selector) มีดังนี้

```
Element {  
    Property : Value;  
}
```

# ซีเล็กเตอร์ชนิดอิลิเมนต์ (Element selector)

ตัวอย่างการใช้ซีเล็กเตอร์อิลิเมนต์เดียว (Element selector)

<pre>&lt;html&gt; &lt;head&gt; &lt;style type="text/css"&gt; h1 {   background-color: pink;   color: white; } p {   background-color: blue;   color: white; } &lt;/style&gt; &lt;/head&gt; &lt;body&gt; &lt;h1&gt;Profile&lt;/h1&gt; &lt;div&gt; &lt;p&gt;Mr.Nutthapat Kaewrattanapat&lt;/p&gt; &lt;p&gt;I am work for Suan Sunandha Rajabhat University&lt;/p&gt;&lt;/div&gt; &lt;p&gt;email:nutthapat.ke@ssru.ac.th&lt;/p&gt; &lt;/body&gt; &lt;/html&gt;</pre>	
---	--

# ซีเล็คเตอร์ชนิดอิลิเมนต์ (Element selector)

---

- ซีเล็คเตอร์ชนิดหลายอิลิเมนต์ (Multiple element selector)


ซีเล็คเตอร์ชนิดหลายอิลิเมนต์ (Multiple element selector) คือ การกำหนดสไตร์ที่มีลักษณะเหมือนกันให้แก่อิลิเมนต์หลายตัว โดยสามารถระบุชื่ออิลิเมนต์มากกว่า 1 ชื่อและคั่นระหว่างชื่ออิลิเมนต์ด้วยเครื่องหมาย คอมม่า ( , )

รูปแบบของซีเล็คเตอร์ชนิดหลายอิลิเมนต์ (Multiple element selector) มีดังนี้

```
Element1, Element2, ElementN {  
    Property : Value;  
}
```

# ซีเล็คเตอร์ชนิดอิลิเมนต์ (Element selector)

ตัวอย่างการใช้ซีเล็คเตอร์ชนิดอิลิเมนต์ (Multiple element selector)

<pre>&lt;html&gt; &lt;head&gt; &lt;style type="text/css"&gt; h1,p {     background-color: limegreen;     color: white; } &lt;/style&gt; &lt;/head&gt; &lt;body&gt; &lt;h1&gt;Profile&lt;/h1&gt; &lt;div&gt; &lt;p&gt;Mr.Nutthapat Kaewrattanapat&lt;/p&gt; &lt;p&gt;I am work for Suan Sunandha     Rajabhat University&lt;/p&gt; &lt;/div&gt; &lt;p&gt;email:nutthapat.ke@ssru.ac.th&lt;/p&gt; &lt;/body&gt; &lt;/html&gt;</pre>	
---	--

# ซีเล็กเตอร์ชนิดอิลิเมนต์ (Element selector)

---

- ซีเล็กเตอร์ชนิดสืบทอดให้ซีเล็กเตอร์ย่อยถัดไป (Descendant selector)

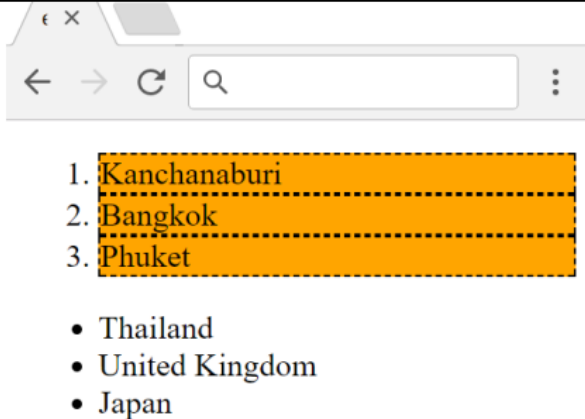
ซีเล็กเตอร์ชนิดสืบทอดให้ซีเล็กเตอร์ย่อยถัดไป (Descendant selector) คือ การกำหนดสไตล์ให้มีผลกับซีเล็กเตอร์ถัดไปตามลำดับ

รูปแบบของซีเล็กเตอร์ชนิดสืบทอดให้ซีเล็กเตอร์ย่อยถัดไป (Descendant selector) มีดังนี้

```
ElementParent ElementTarget {  
    Property : Value;  
}
```

# ซีเล็กเตอร์ชนิดอิลิเมนต์ (Element selector)

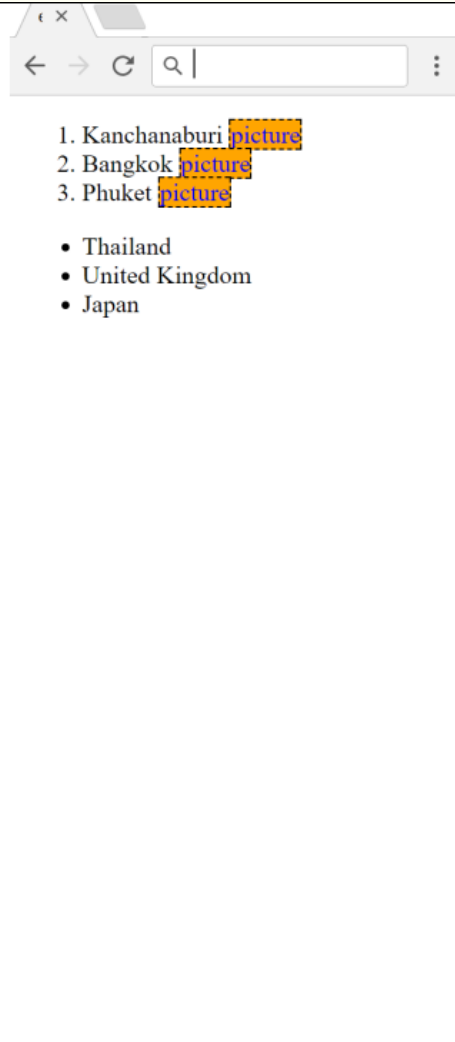
ตัวอย่างการใช้ซีเล็กเตอร์ชนิดสืบทอดให้ซีเล็กเตอร์ย่อยถัดไป (Descendant selector) แบบที่ 1

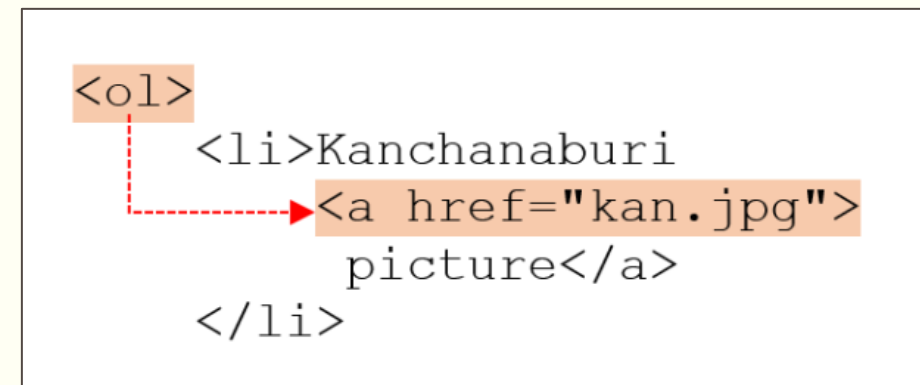
<pre>&lt;html&gt; &lt;head&gt; &lt;style type="text/css"&gt; ol li{     background-color: orange;     border: 1px dashed black; } &lt;/style&gt; &lt;/head&gt; &lt;body&gt; &lt;ol&gt; &lt;li&gt;Kanchanaburi&lt;/li&gt; &lt;li&gt;Bangkok&lt;/li&gt; &lt;li&gt;Phuket&lt;/li&gt; &lt;/ol&gt; &lt;ul&gt; &lt;li&gt;Thailand&lt;/li&gt; &lt;li&gt;United Kingdom&lt;/li&gt; &lt;li&gt;Japan&lt;/li&gt; &lt;/ul&gt; &lt;/body&gt; &lt;/html&gt;</pre>	
---	--



# ชีเล็กเตอร์ชนิดอิลิเมนต์ (Element selector)

ตัวอย่างการใช้ชีเล็กเตอร์ชนิดสืบทอดให้ชีเล็กเตอร์ย่อยถัดไป (Descendant selector) แบบที่ 2

<pre>&lt;html&gt; &lt;head&gt; &lt;style type="text/css"&gt; ol a{   background-color: orange;   border: 1px dashed black;   text-decoration:none; } &lt;/style&gt; &lt;/head&gt; &lt;body&gt; &lt;ol&gt;   &lt;li&gt;Kanchanaburi     &lt;a href="kan.jpg"&gt;       picture&lt;/a&gt;&lt;/li&gt;   &lt;li&gt;Bangkok     &lt;a href="bkk.jpg"&gt;       picture&lt;/a&gt;&lt;/li&gt;   &lt;li&gt;Phuket     &lt;a href="phu.jpg"&gt;       picture&lt;/a&gt;&lt;/li&gt; &lt;/ol&gt; &lt;ul&gt;   &lt;li&gt;Thailand&lt;/li&gt;   &lt;li&gt;United Kingdom&lt;/li&gt;   &lt;li&gt;Japan&lt;/li&gt; &lt;/ul&gt; &lt;/body&gt; &lt;/html&gt;</pre>	
---	---





# ซีเล็กเตอร์ชนิดอิลิเมนต์ (Element selector)

---

- ซีเล็กเตอร์ชนิดสืบทอดให้อิลิเมนต์ย่อยแบบเรียงลำดับ (Sub selector)


ซีเล็กเตอร์ชนิดสืบทอดให้อิลิเมนต์ย่อยแบบเรียงลำดับ (Sub selector) คือ การกำหนดการแสดงผลของอิลิเมนต์เป้าหมาย (Target) โดยอิลิเมนต์เป้าหมายจะถูกเลือกแสดงผลได้ก็ต่อเมื่ออิลิเมนต์ก่อนหน้า (Parent) มีการเรียงลำดับตามที่กำหนดไว้เท่านั้น

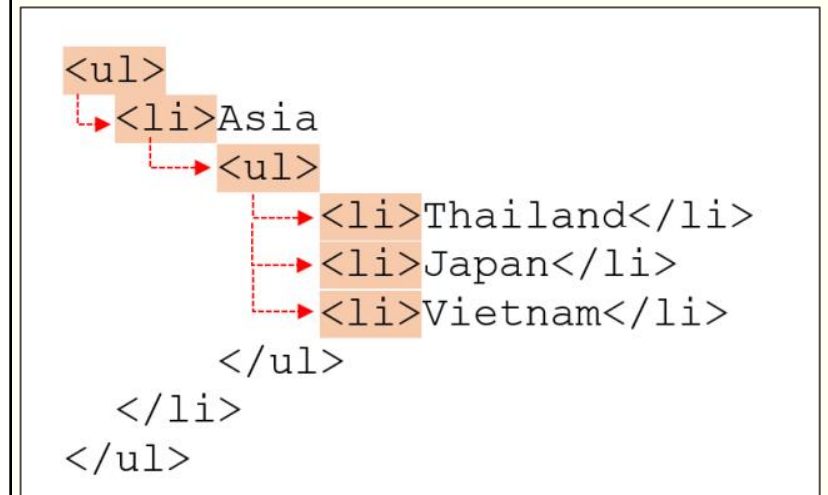
รูปแบบของซีเล็กเตอร์ชนิดสืบทอดให้อิลิเมนต์ย่อยแบบเรียงลำดับ (Sub selector) มีดังนี้

```
ElementParent > ElementTarget {  
    Property : Value;  
}
```

# ชีเล็กเตอร์ชนิดอิลิเมนต์ (Element selector)

ตัวอย่างการใช้ชีเล็กเตอร์ชนิดสืบทอดให้อิลิเมนต์ย่อยแบบเรียงลำดับ (Sub selector)

<pre>&lt;html&gt; &lt;head&gt; &lt;style type="text/css"&gt; ul&gt;li&gt;ul&gt;li {     background-color:aquamarine; } &lt;/style&gt; &lt;/head&gt; &lt;body&gt;   &lt;ul&gt;     &lt;li&gt;Asia       &lt;ul&gt;         &lt;li&gt;Thailand&lt;/li&gt;         &lt;li&gt;Japan&lt;/li&gt;         &lt;li&gt;Vietnam&lt;/li&gt;       &lt;/ul&gt;     &lt;/li&gt;   &lt;/ul&gt; &lt;/body&gt; &lt;/html&gt;</pre>	
---	--



# ซีเล็กเตอร์ชนิดอิลิเมนต์ (Element selector)

---

- ซีเล็กเตอร์ชนิดสืบทอดให้อิลิเมนต์ภายนอกแบบเรียงลำดับ (Sequential selector)


ซีเล็กเตอร์ชนิดสืบทอดให้อิลิเมนต์ภายนอกแบบเรียงลำดับ (Sequential selector) หรือ อาจเรียกว่า ซีเล็กเตอร์ชนิดสืบทอดให้อิลิเมนต์ที่ติดกัน (Adjacent selector) คือ การเรียงลำดับอิลิเมนต์ภายนอกเพื่อกำหนดการแสดงผลของอิลิเมนต์เป้าหมาย (Target)

รูปแบบของซีเล็กเตอร์ชนิดสืบทอดให้อิลิเมนต์ภายนอกแบบเรียงลำดับ (Sequential selector) มีดังนี้

```
Element + ElementTarget {  
    Property : Value;  
}
```

# ซีเล็คเตอร์ชนิดอิลิเมนต์ (Element selector)

ตัวอย่างการใช้ซีเล็คเตอร์ชนิดสืบทอดให้อิลิเมนต์ภายนอกแบบเรียงลำดับ (Sequential selector)

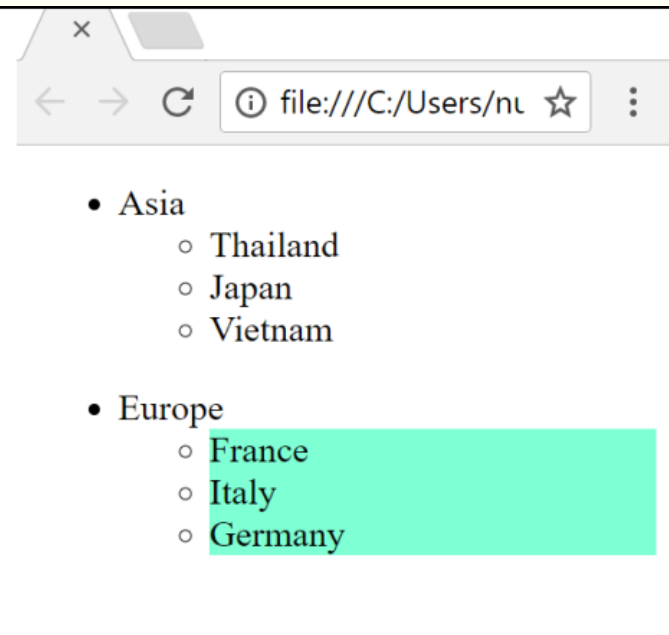
<pre>&lt;html&gt; &lt;head&gt; &lt;style type="text/css"&gt; ul+ul{   background-color:aquamarine; } &lt;/style&gt; &lt;/head&gt; &lt;body&gt;   &lt;ul&gt;     &lt;li&gt;Asia       &lt;ul&gt;         &lt;li&gt;Thailand&lt;/li&gt;         &lt;li&gt;Japan&lt;/li&gt;         &lt;li&gt;Vietnam&lt;/li&gt;       &lt;/ul&gt;     &lt;/li&gt;   &lt;/ul&gt;   &lt;ul&gt;     &lt;li&gt;Europe       &lt;ul&gt;         &lt;li&gt;France&lt;/li&gt;         &lt;li&gt;Italy&lt;/li&gt;         &lt;li&gt;Germany&lt;/li&gt;       &lt;/ul&gt;     &lt;/li&gt;   &lt;/ul&gt; &lt;/body&gt; &lt;/html&gt;</pre>	
--	---

```
<ul>
  <li>Asia
    <ul>
      <li>Thailand</li>
      <li>Japan</li>
      <li>Vietnam</li>
    </ul>
  </li>
</ul>
<ul>
  <li>Europe
    <ul>
      <li>France</li>
      <li>Italy</li>
      <li>Germany</li>
    </ul>
  </li>
</ul>
```

# ซีเล็กเตอร์ชนิดอิลิเมนต์ (Element selector)

กรณีต้องการให้ซีเล็กเตอร์เลือกอิลิเมนต์ย่อยร่วมด้วย สามารถกระทำได้โดยการประยุกต์ใช้ซีเล็กเตอร์อิลิเมนต์ภายนอกแบบเรียงลำดับ (Sequential selector) กับ ซีเล็กเตอร์อิลิเมนต์ย่อยแบบเรียงลำดับ (Sub selector)

ตัวอย่างการประยุกต์ใช้ซีเล็กเตอร์อิลิเมนต์ภายนอกแบบเรียงลำดับ (Sequential selector) กับ ซีเล็กเตอร์อิลิเมนต์ย่อยแบบเรียงลำดับ (Sub selector)

<pre>&lt;html&gt; &lt;head&gt; &lt;style type="text/css"&gt; ul+ul&gt;li&gt;ul&gt;li {     background-color:aquamarine; } &lt;/style&gt; &lt;/head&gt; &lt;body&gt;   &lt;ul&gt;     &lt;li&gt;Asia       &lt;ul&gt;</pre>	 <p>The screenshot shows a web browser window with a file path in the address bar. The content displays a list of countries under two main categories: Asia and Europe. The 'Europe' category is highlighted with a light blue background, and the 'Germany' item is highlighted with a light blue background.</p> <ul style="list-style-type: none"><li>• Asia<ul style="list-style-type: none"><li>◦ Thailand</li><li>◦ Japan</li><li>◦ Vietnam</li></ul></li><li>• Europe<ul style="list-style-type: none"><li>◦ France</li><li>◦ Italy</li><li>◦ Germany</li></ul></li></ul>
--	--

# ชีล็กเตอร์ชนิดอิลิเมนต์ (Element selector)

```
<li>Thailand</li>
<li>Japan</li>
<li>Vietnam</li>
</ul>
</li>
</ul>
<ul>
<li>Europe
  <ul>
    <li>France</li>
    <li>Italy</li>
    <li>Germany</li>
  </ul>
</li>
</ul>
</body>
</html>
```

Diagram illustrating the structure of the HTML code with selectors highlighted:

```
<ul>
  <li>Asia
    <ul>
      <li>Thailand</li>
      <li>Japan</li>
      <li>Vietnam</li>
    </ul>
  </li>
</ul>
<ul>
  <li>Europe
    <ul>
      <li>France</li>
      <li>Italy</li>
      <li>Germany</li>
    </ul>
  </li>
</ul>
```

The diagram shows the following selectors highlighted in orange:

- Sequential selector:** Indicated by a red dashed arrow pointing to the first `<ul>` tag.
- Sub selector:** Indicated by blue dashed arrows pointing to the nested `<ul>` tags within the `<li>Europe` and `<li>Asia` elements.

# ซีเล็กเตอร์ชนิดอิลิเมนต์ (Element selector)

- ซีเล็กเตอร์ชนิดสืบทอดให้อิลิเมนต์ถัดไปทั้งหมด (Preceded selector)

ซีเล็กเตอร์ชนิดสืบทอดให้อิลิเมนต์ถัดไปทั้งหมด (Preceded selector) คือ การเลือกอิลิเมนต์ถัดไปทั้งหมดให้แสดงผลตามที่กำหนด โดยมีความแตกต่างจากซีเล็กเตอร์ชนิดสืบทอดให้อิลิเมนต์ภายนอกแบบเรียงลำดับ (Sequential selector) เนื่องจากซีเล็กเตอร์ชนิดสืบทอดให้อิลิเมนต์ภายนอกแบบเรียงลำดับ (Sequential selector) จะส่งผลกับอิลิเมนต์เดียวเท่านั้น แต่ซีเล็กเตอร์ชนิดสืบทอดให้อิลิเมนต์ถัดไปทั้งหมด (Preceded selector) จะส่งผลกับอิลิเมนต์เป้าหมาย (Target) ทุกๆ ตัว ที่อยู่ถัดจากอิลิเมนต์ก่อนหน้า (Preceded)

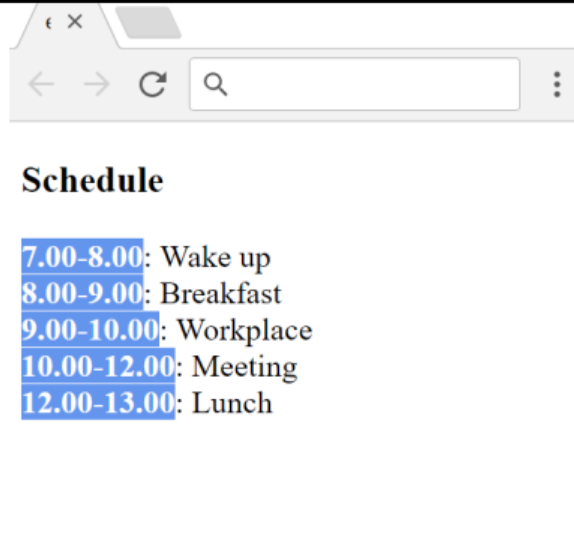
รูปแบบของซีเล็กเตอร์ชนิดสืบทอดให้อิลิเมนต์ถัดไปทั้งหมด (Preceded selector) มีดังนี้

```
ElementPreceded ~ ElementTarget {  
    Property : Value;  
}
```

# ชีล็กเตอร์ชนิดอิลิเมนต์ (Element selector)

ตัวอย่างการประยุกต์ใช้ชีล็กเตอร์ชนิดสืบทอดให้อิลิเมนต์ถัดไปทั้งหมด (Preceded selector)

```
<html>
<head>
<style type="text/css">
h3~b {
    background-color:
    cornflowerblue;
    color: white;
}
</style>
</head>
<body>
<h3>Schedule</h3>
<b>7.00-8.00</b>: Wake up<br>
<b>8.00-9.00</b>: Breakfast<br>
<b>9.00-10.00</b>: Workplace<br>
<b>10.00-12.00</b>: Meeting<br>
<b>12.00-13.00</b>: Lunch<br>
</body>
</html>
```



**Preceded selector**

**Target selector**

```
<h3>Schedule</h3>
<b>7.00-8.00</b>: Wake up<br>
<b>8.00-9.00</b>: Breakfast<br>
<b>9.00-10.00</b>: Workplace<br>
<b>10.00-12.00</b>: Meeting<br>
<b>12.00-13.00</b>: Lunch<br>
```



# ซีเล็กเตอร์ชนิดคลาส (Class selector)

---

ซีเล็กเตอร์ชนิดคลาส (Class selector) คือ การกำหนดชื่อตัวเลือกด้วยตนเองได้ซึ่งชื่อตัวเลือกนั้นๆ จะถูกเรียกว่า คลาส (Class) หมายถึง ชื่อหมวดลักษณะ โดยอิลิเมนต์ใดๆ สามารถเรียกใช้คุณลักษณะของคลาสได้บ่อยครั้งเท่าที่ต้องการ โดยการกำหนดคลาสจะใช้เครื่องหมาย จุด ( . ) นำหน้าชื่อคลาส

รูปแบบของซีเล็กเตอร์ชนิดคลาส (Class) มีดังนี้

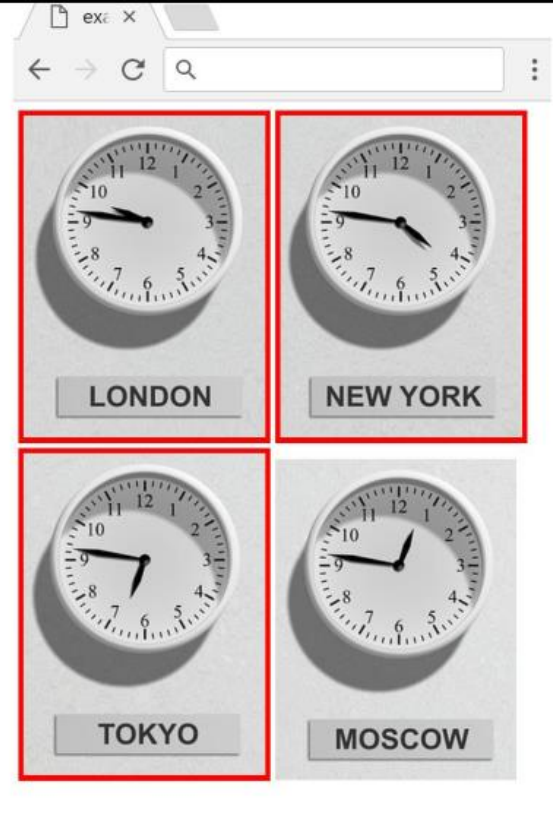
```
.ClassName {  
    Property : Value;  
}
```

รูปแบบการเรียกใช้คลาสของอิลิเมนต์ มีดังนี้

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

# ซีเล็กเตอร์ชนิดคลาส (Class selector)

ตัวอย่างการกำหนดซีเล็กเตอร์ชนิดคลาสและการเรียกใช้คลาสของอิลิเมนต์

<pre>&lt;html&gt; &lt;head&gt; &lt;style type="text/css"&gt; .clock {     border: 5px solid red; } &lt;/style&gt; &lt;/head&gt; &lt;body&gt; &lt;img class="clock"     src="london.jpg"&gt; &lt;img class="clock"     src="newyork.jpg"&gt; &lt;img class="clock"     src="tokyo.jpg"&gt; &lt;img src="moscow.jpg"&gt; &lt;/body&gt; &lt;/html&gt;</pre>	
--	---

```
<html>
  <head>
    <style>
      .clock {
        Border: 5px solid red;
      }
    </style>
  </head>
  <body>
    
    
    
    
  </body>
</html>
```

# ซีเล็คเตอร์ชนิดคลาส (Class selector)

ตัวอย่างการเรียกใช้คลาสมากกว่า 1 คลาส ของอีลิเมนต์

```
<html>
<head>
<style type="text/css">
.clock{
  height: 40%;
}
.leftframe{
  border-left: 5px solid red;
}
.topframe{
  border-top: 5px solid green;
}
.rightframe{
  border-right: 5px solid blue;
}
.bottomframe{
  border-bottom: 5px solid gray;
}
</style>
</head>
<body>




</body>
</html>
```

```
<html>
<head>
<style>
.clock{
  height: 40%;
}
.leftframe{
  border-left: 5px solid red;
}
.topframe{
  border-top: 5px solid green;
}
.rightframe{
  border-right: 5px solid blue;
}
.bottomframe{
  border-bottom: 5px solid gray;
}
</style>
</head>
<body>




</body>
</html>
```

# ซีเล็กเตอร์ชนิดเอกลักษณ์ (ID selector)

---

ซีเล็กเตอร์ชนิดเอกลักษณ์ (ID selector) หรือ เรียกว่า ซีเล็กเตอร์ชนิดไอดี โดยคำว่า ไอดี (ID) คือ ตัวย่อภาษาอังกฤษที่มาจากคำว่า “Identification” แปลว่า การระบุเอกลักษณ์เฉพาะ ซึ่งซีเล็กเตอร์ชนิดเอกลักษณ์ คือ การกำหนดชื่อเฉพาะที่เป็นเอกลักษณ์และไม่ซ้ำกับชื่อซีเล็กเตอร์อื่นๆ เมื่ออิลิเมนต์ใดๆ ก็ตามมีการกำหนดแอททริบิวต์ชื่อไอดี (Attribute ID) และมีชื่อตรงกับชื่อของซีเล็กเตอร์ชนิดเอกลักษณ์ตัวใด ก็จะได้แสดงสไตล์หรือลักษณะของซีเล็กเตอร์ชนิดเอกลักษณ์ตัวนั้นๆ ออกมา โดยการใช้ซีเล็กเตอร์ชนิดเอกลักษณ์นั้น เหมาะกับการระบุเพียง 1 ครั้งเท่านั้น เนื่องจากอิลิเมนต์ 1 อิลิเมนต์ควรมีรหัสเอกลักษณ์ (Attribute ID) เพียงชื่อเดียว เพื่อประโยชน์ในการใช้เฉพาะอิลิเมนต์ที่เจาะจงเป็นพิเศษ

รูปแบบของซีเล็กเตอร์ชนิดเอกลักษณ์ (ID selector) มีดังนี้

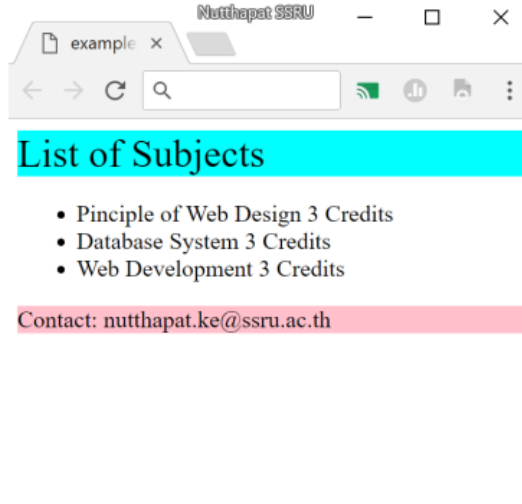
```
#IDname {  
    Property : Value;  
}
```

รูปแบบการเรียกใช้ซีเล็กเตอร์ชนิดเอกลักษณ์ (ID selector) ของอิลิเมนต์มีดังนี้

```
<p id="IDname"> ... </p>
```

# ชี้เลือกเตอร์ชนิดเอกลักษณ์ (ID selector)

ตัวอย่างการกำหนดและเรียกใช้ชี้เลือกเตอร์ชนิดเอกลักษณ์ (ID selector)

<pre>&lt;html&gt; &lt;head&gt; &lt;style type="text/css"&gt;   #header{     font-size: 20 pt;     background-color: aqua;   }   #footer{     background-color: pink;   } &lt;/style&gt; &lt;/head&gt; &lt;body&gt;</pre>	
--	--

<pre>&lt;div id="header"&gt;   List of Subjects &lt;/div&gt; &lt;div id="body"&gt;   &lt;ul&gt;     &lt;li&gt;Pinciple of Web       Design 3 Credits&lt;/li&gt;     &lt;li&gt;Database System       3 Credits&lt;/li&gt;     &lt;li&gt;Web Development       3 Credits&lt;/li&gt;   &lt;/ul&gt; &lt;/div&gt; &lt;div id="footer"&gt;   Contact:nutthapat.ke@ssru.ac.th &lt;/div&gt; &lt;/body&gt; &lt;/html&gt;</pre>	
---	--

# ซีเล็กเตอร์ชนิดเอกลักษณ์ (ID selector)

```
<html>
<head>
<style>
  #header{
    font-size: 20pt;
    background-color: aqua;
  }
  #footer{
    background-color: pink;
  }
</style>
</head>
<body>
  <div id="header">
    List of Subjects
  </div>
  <div id="body">
    <ul>
      <li>Pinciple of Web Design 3 Credits</li>
      <li>Database System 3 Credits</li>
      <li>Web Development 3 Credits</li>
    </ul>
  </div>
  <div id="footer">
    Contact: nutthapat.ke@ssru.ac.th
  </div>
</body>
</html>
```

A red dashed line forms a rectangular box around the HTML body content. The line starts at the opening tag of the first `<div id="header">`, goes down to the opening tag of the last `<div id="footer">`, and then goes back up to the closing tag of the first `<div id="header">`. This illustrates that the ID selector targets specific elements by their unique ID attribute.

# ซีเล็กเตอร์ชนิดแอททริบิวต์ (Attribute selector)

---

ซีเล็กเตอร์ชนิดแอททริบิวต์ (Attribute selector) โดยคำว่า “Attribute” แปลว่า คุณลักษณะ ซึ่งตัวเลือกชนิดแอททริบิวต์เป็นการเลือกปรับลักษณะอิลิเมนต์ที่มีการใช้แอททริบิวต์ตรงกับซีเล็กเตอร์ชนิดแอททริบิวต์ที่กำหนดไว้ในส่วนของสไตล์ชีท

รูปแบบของซีเล็กเตอร์ชนิดแอททริบิวต์ (Attribute selector) มีดังนี้

```
[attriburteName] {  
    Property : Value;  
}
```

# ซีเล็กเตอร์ชนิดแอททริบิวต์ (Attribute selector)

ตารางแสดงซีเล็กเตอร์ชนิดแอททริบิวต์

ซีเล็กเตอร์ชนิดแอททริบิวต์	ความหมาย	ตัวอย่าง
[attributename]	ซีเล็กเตอร์ชนิดแอททริบิวต์ที่เลือกอีลีเมนต์ที่มีการระบุแอททริบิวต์ภายในอีลีเมนต์	[width] [height] [src] [id] [name]
[attributename="val"]	ซีเล็กเตอร์ชนิดแอททริบิวต์ที่เลือกเฉพาะอีลีเมนต์ที่มีการใช้แอททริบิวต์และมีการกำหนดค่าที่ตรงกับค่าที่ระบุไว้ในซีเล็กเตอร์	[width="200px"] [src="logo.png"] [id="header"]
[attributename~="val"]	ซีเล็กเตอร์ชนิดแอททริบิวต์ที่เลือกเฉพาะอีลีเมนต์ที่มี	h2[title~="Thai"]



# ซีเล็กเตอร์ชนิดแอททริบิวต์ (Attribute selector)

ซีเล็กเตอร์ชนิดแอททริบิวต์	ความหมาย	ตัวอย่าง
	การใช้แอททริบิวต์ และมีการกำหนดค่า ส่วนใดส่วนหนึ่งที่ ตรงกับค่าที่ระบุไว้ใน ซีเล็กเตอร์โดยค่านั้น จะต้องเป็นค่าที่แยก ออกจากคำอื่น เท่านั้น	<code>&lt;h2 title="Thai" News"&gt; THAI NEWS&lt;/h2&gt;</code>

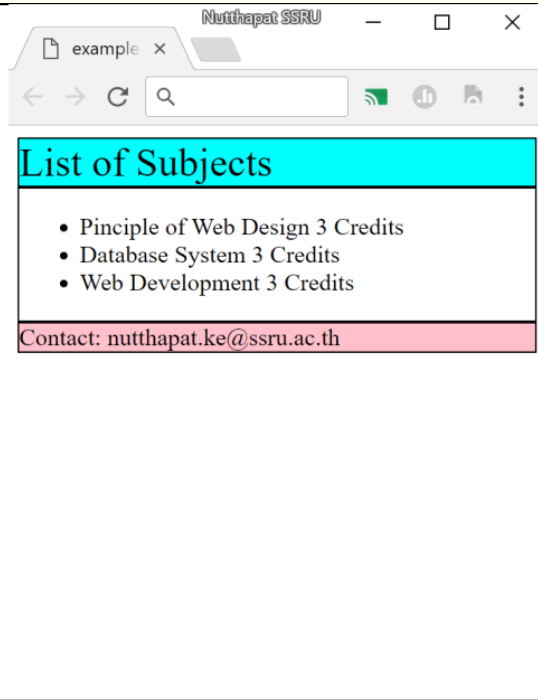
<code>[attributename="val"]</code>	ซีเล็กเตอร์ชนิด แอททริบิวต์ที่เลือก เฉพาะอีลิเมนต์ที่มี การใช้แอททริบิวต์ และมีการกำหนดค่า ในส่วนเริ่มต้นที่ตรง กับค่าที่ระบุไว้ในซี เล็กเตอร์โดยค่านั้น ไม่จำเป็นต้องเป็นคำ แยกจากคำอื่น	<code>h2[title="Thai"]</code>  <code>&lt;h2 title="ThaiNews"&gt; THAI NEWS&lt;/h2&gt;</code>
<code>[attributename^="val"]</code>	ซีเล็กเตอร์ชนิดแอ ตทริบิวต์ที่เลือก เฉพาะอีลิเมนต์ที่มี การใช้แอททริบิวต์ และมีการกำหนดค่า ในส่วนเริ่มต้นที่ตรง กับค่าที่ระบุไว้ในซี เล็กเตอร์	<code>a[href="http"]</code>  <code>&lt;a href="http://www.thai news.com"&gt;News&lt;/a&gt;</code>

# ซีเล็กเตอร์ชนิดแอตทริบิวต์ (Attribute selector)

ซีเล็กเตอร์ชนิดแอตทริบิวต์	ความหมาย	ตัวอย่าง
[attributename\$="val"]	ซีเล็กเตอร์ชนิดแอตทริบิวต์ที่เลือกเฉพาะอิลิเมนต์ที่มีการใช้แอตทริบิวต์และมีการกำหนดค่าจุดสิ้นสุดที่ตรงกับค่าที่ระบุไว้ในซีเล็กเตอร์	a[href\$="png"]
		<a href="http://www.thai news.com/logo.png">Logo</a>
[attributename*="val"]	ซีเล็กเตอร์ชนิดแอตทริบิวต์ที่เลือกเฉพาะอิลิเมนต์ที่มีการใช้แอตทริบิวต์และมีค่าส่วนหนึ่งส่วนใดตรงกับค่าที่ระบุไว้ในซีเล็กเตอร์	a[href*="image"]
		<a href="http://www.thai news.com/image.png">Logo</a>

# ซีเล็กเตอร์ชนิดแอททริบิวต์ (Attribute selector)

ตัวอย่างการใช้ซีเล็กเตอร์ชนิดแอททริบิวต์ (Attribute selector)

<pre>&lt;html&gt; &lt;head&gt; &lt;style type="text/css"&gt;   [id]{     border: 1px solid black;   }   #header{     font-size: 20pt;     background-color: aqua;   }   #footer{     background-color: pink;   } &lt;/style&gt; &lt;/head&gt; &lt;body&gt;   &lt;div id="header"&gt;     List of Subjects</pre>		<pre>&lt;/div&gt; &lt;div id="body"&gt;   &lt;ul&gt;     &lt;li&gt;Pinciple of Web..&lt;/li&gt;     &lt;li&gt;Database System..&lt;/li&gt;     &lt;li&gt;Web Development..&lt;/li&gt;   &lt;/ul&gt; &lt;/div&gt; &lt;div id="footer"&gt;   Contact:nutthapat.ke@ssru.ac.th &lt;/div&gt; &lt;/body&gt; &lt;/html&gt;</pre>
---	---	---

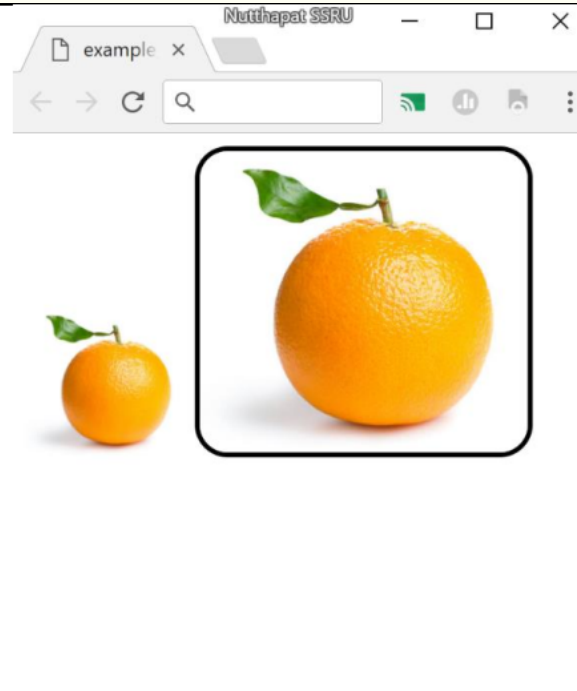
# ซีเล็กเตอร์ชนิดแอททริบิวต์ (Attribute selector)

```
<html>
<head>
<style>
  [id]{
    border: 1px solid black;
  }
  #header{
    font-size: 20pt;
    background-color: aqua;
  }
  #footer{
    background-color: pink;
  }
</style>
</head>
<body>
  <div id="header">
    List of Subjects
  </div>
  <div id="body">
    <ul>
      <li>Pinciple of Web Design 3 Credits</li>
      <li>Database System 3 Credits</li>
      <li>Web Development 3 Credits</li>
    </ul>
  </div>
  <div id="footer">
    Contact: nutthapat.ke@ssru.ac.th
  </div>
</body>
</html>
```

# ซีเล็กเตอร์ชนิดแอททริบิวต์ (Attribute selector)

ตัวอย่างการใช้ซีเล็กเตอร์ชนิดแอททริบิวต์ width ร่วมกับการระบุค่า 200

```
<html>
<head>
<style type="text/css">
  [width="200"]{
    border-radius: 20px;
    border: 3px solid black;
  }
</style>
</head>
<body>
  
  
</body>
</html>
```

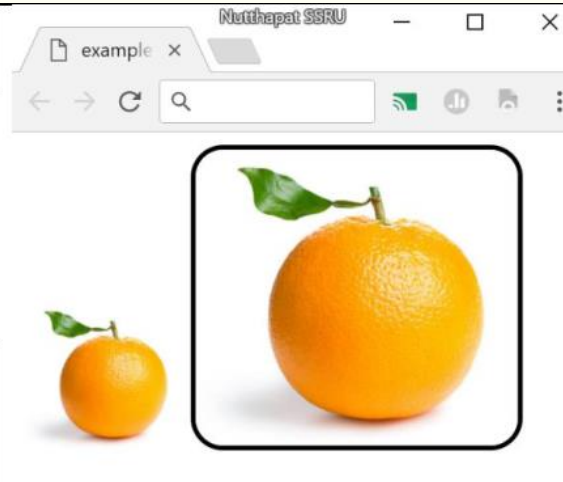


```
<html>
  <head>
    <style>
      [width="200"]{
        border-radius: 20px;
        border: 3px solid black;
      }
    </style>
  </head>
  <body>
    
    
  </body>
</html>
```

# ซีเล็กเตอร์ชนิดแอททริบิวต์ (Attribute selector)

ตัวอย่างการใช้ซีเล็กเตอร์ชนิดแอททริบิวต์ alt ร่วมกับการระบุค่า Small Orange

```
<html>
<head>
<style type="text/css">
  [alt="Small Orange"]{
    border-radius: 20px;
    border: 3px solid red;
  }
</style>
</head>
<body>
  
  
</body>
</html>
```



```
<html>
  <head>
    <style>
      [alt="Small Orange"]{
        border-radius: 20px;
        border: 3px solid red;
      }
    </style>
  </head>
  <body>
    
    
  </body>
</html>
```

# ซีเล็คเตอร์เทียม (Pseudo-Selector)

---

ซีเล็คเตอร์เทียม (Pseudo-Selector) คือ ส่วนเสริมของซีเล็คเตอร์ เพื่อช่วยให้นักออกแบบและพัฒนาเว็บสามารถกำหนดสไตล์แก้อิลิเมนต์ให้มีความเฉพาะเจาะจงมากยิ่งขึ้น โดยซีเล็คเตอร์เทียมมี 2 ลักษณะ ดังนี้

**1. ซีเล็คเตอร์เทียมประเภทคลาส (Pseudo-Class)** ใช้เมื่อต้องการกำหนดเงื่อนไขเฉพาะหรือสถานะเฉพาะให้แก่ซีเล็คเตอร์ชนิดคลาส โดยจะใช้สัญลักษณ์โคลอน “:” (Colon) นำหน้าเงื่อนไขหรือสถานะ นักออกแบบและพัฒนาเว็บนิยมนำซีเล็คเตอร์เทียมประเภทคลาสมาใช้ในการกำหนดสถานะของอิลิเมนต์เมื่อมีการนำเมาส์ไปชี้ (Mouse over) เช่น การกำหนดให้แก่สถานะของอิลิเมนต์เอ (<a>) เป็นต้น

รูปแบบการกำหนดซีเล็คเตอร์เทียมประเภทคลาส มีดังนี้

```
Selector : Pseudo-class {  
    Property : Value;  
}
```

# ซีเล็คเตอร์เทียม (Pseudo-Selector)

ตารางแสดงซีเล็คเตอร์เทียมประเภทคลาส

ซีเล็คเตอร์เทียมประเภท คลาส	ความหมาย	ตัวอย่าง
ซีเล็คเตอร์เทียมประเภทคลาสที่มีลักษณะพลวัตร (Dynamic pseudo-classes)		
:link	สไตล์จะทำงานก็ต่อเมื่ออีลิเมนต์นั้นๆ เป็นลิงค์	a:link { color: red; }
:visited	สไตล์จะทำงานก็ต่อเมื่อเคยมีการคลิกลิงค์ดังกล่าวแล้ว	a:visited { color: green; }
:hover	สไตล์จะทำงานก็ต่อเมื่อมีการนำเมาส์ไปวางไว้เหนือลิงค์ หรือ อีลิเมนต์อื่นๆ	a:hover { background-color: black; }
:active	สไตล์จะทำงานก็ต่อเมื่อลิงค์ถูกคลิก	a:active { color: blue; }



# ซีเล็คเตอร์เทียม (Pseudo-Selector)

ซีเล็คเตอร์เทียมประเภท คลาส	ความหมาย	ตัวอย่าง
:focus	สไตล์จะทำงานก็ต่อเมื่อ เคอร์เซอร์อยู่ที่อีลีเมนต์ หรือมีการคลิกที่อีลีเมนต์ เช่น การเลื่อนเคอร์เซอร์ไป ยังกล่องป้อนข้อความ เป็น ต้น	<pre>input:focus{   background-color:   blue; }</pre> <pre>&lt;body&gt; &lt;form&gt; Phone: &lt;input type="tel"&gt; &lt;br&gt; e-mail: &lt;input type="email" disabled="disabled"&gt; &lt;/form&gt; &lt;/body&gt;</pre>

ซีเล็คเตอร์เทียมประเภทคลาสที่ทำงานเมื่อถึงเป้าหมาย (The target pseudo-class)		
:target	สไตล์จะทำงานก็ต่อเมื่อถึง เป้าหมายจากการคลิกลิงค์	<pre>:target {   border:   2px solid green;   background-color:   yellow; }</pre> <pre>&lt;a href="#news1"&gt; Content 1&lt;/a&gt;&lt;br&gt; &lt;a href="#news2"&gt; Content 2&lt;/a&gt;  &lt;p id="news1"&gt; Content 1&lt;/p&gt; &lt;p id="news2"&gt; Content 2&lt;/p&gt;</pre>
ซีเล็คเตอร์เทียมประเภทคลาสที่ทำงานเกี่ยวกับภาษา (The language pseudo-class)		
:lang(LN)	สไตล์จะทำงานก็ต่อเมื่อมี อีลีเมนต์ที่ใช้แอตทริบิวต์ lang และมีค่าตรงกับ (LN) ที่ระบุไว้ในซีเล็คเตอร์เทียม	<pre>:lang(en){   font-style:   italic; }</pre> <pre>&lt;body&gt; &lt;p&gt;The man said   &lt;q lang="en"&gt;     ENGLISH.   &lt;/q&gt; &lt;/p&gt;</pre>

# ซีเล็คเตอร์เทียม (Pseudo-Selector)

ซีเล็คเตอร์เทียมประเภท คลาส	ความหมาย	ตัวอย่าง
		</body>
ซีเล็คเตอร์เทียมประเภทคลาสที่ทำงานเกี่ยวกับส่วนต่อประสานผู้ใช้ (The UI element states pseudo-classes)		
:enabled	สไตส์จะทำงานก็ต่อเมื่อสถานะของอีลิเมนต์ในฟอร์มมีค่าสามารถเข้าถึงหรือสามารถกรอกข้อมูลได้ (enabled)	<pre>input:enabled{   background-color:   red; } input:disabled{   background-color:   gray; }</pre> <pre>&lt;body&gt; &lt;form&gt; Phone: <input &gt;="" &lt;="" &lt;br&gt;="" <input="" body&gt;<="" disabled="disabled" e-mail:="" form&gt;="" pre="" type="email"/> </pre>

:disabled	สไตส์จะทำงานก็ต่อเมื่อสถานะของอีลิเมนต์ในฟอร์มมีค่าไม่สามารถเข้าถึงได้ (disabled)	<pre>input:enabled{   background-color:   red; } input:disabled{   background-color:   gray; }</pre> <pre>&lt;body&gt; &lt;form&gt; Phone: <input &gt;="" &lt;="" &lt;br&gt;="" <input="" disabled="disabled" e-mail:="" form&gt;<="" pre="" type="email"/> </pre>
-----------	---	--

# ซีเล็คเตอร์เทียม (Pseudo-Selector)

ซีเล็คเตอร์เทียมประเภท คลาส	ความหมาย	ตัวอย่าง
		</body>
:checked	สไตส์จะทำงานก็ต่อเมื่อสถานะของอีลีเมนต์กล่องเลือก (Check box) ในฟอร์มมีการถูกคลิกเลือก (checked)	<div> <input checked="" type="checkbox"/>input:checked+label{font-weight: bold;} </div> <div> &lt;body&gt; &lt;form&gt; &lt;input type="radio" id="m" name="gender" value="male"&gt; &lt;label for="m"&gt;male&lt;/label&gt; &lt;input type="radio" id="f" name="gender" value="female"&gt; &lt;label for="f"&gt;female&lt;/label&gt; &lt;/form&gt; &lt;/body&gt; </div>

ซีเล็คเตอร์เทียมประเภทคลาสที่ทำงานเกี่ยวกับโครงสร้างของเอชทีเอ็มแอล (Structural pseudo-classes)		
:root	สไตส์จะทำงานก็ต่อเมื่อพบอีลีเมนต์ราก (root) ซึ่งในโครงสร้างภาษาเอชทีเอ็มแอล รากของโครงสร้างคืออีลีเมนต์ <html></html> ดังนั้น เมื่อมีการใช้ซีเล็คเตอร์เทียมนี้จะมีผลต่อสไตส์ของอีลีเมนต์ทั้งหน้าเว็บเพจ	:root {background: red;}

# ซีเล็คเตอร์เทียม (Pseudo-Selector)

ซีเล็คเตอร์เทียมประเภท คลาส	ความหมาย	ตัวอย่าง
:nth-child(an+b)	สไตล์จะทำงานก็ต่อเมื่อพบลำดับที่ ( $n^{\text{th}}$ ) ของอีลิเมนต์ลูก (child) ภายในโครงสร้างเอกสารที่เอ็มแอล โดยสามารถกำหนดค่าได้ 3 ลักษณะ คือ 1) ตัวเลขจำนวนเต็ม (Integer) เช่น 1 2 3 -1 -2 -3 เป็นต้น 2) ข้อความ เช่น odd (คี่) และ even (คู่) 3) สูตร (Formula) โดยที่ตัวแปร a คือ ขนาดของวงรอบ ตัวแปร n คือ ค่าโดยปริยายซึ่งจะมีค่าตั้งแต่ 0 1 2 3 .. n ค่า n ขึ้นอยู่กับจำนวนลูกที่เป็นไปได้ทั้งหมด ตัวแปร b คือ ค่าทดแทนเพิ่มเติมเพื่อให้มีความยืดหยุ่นในการเลือกอีลิเมนต์ลูก	:nth-child(1) :nth-child(2n) :nth-child(odd) :nth-child(even) :nth-child(2n+1)

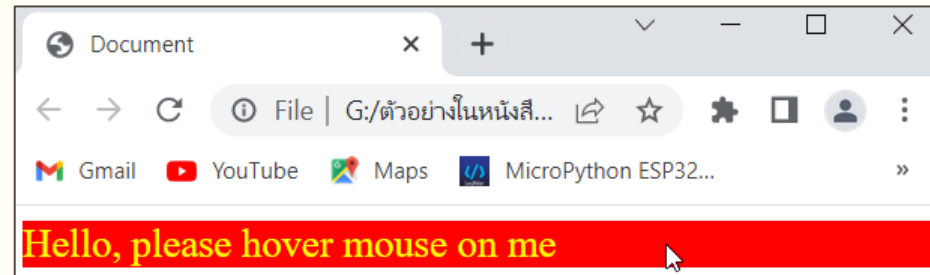
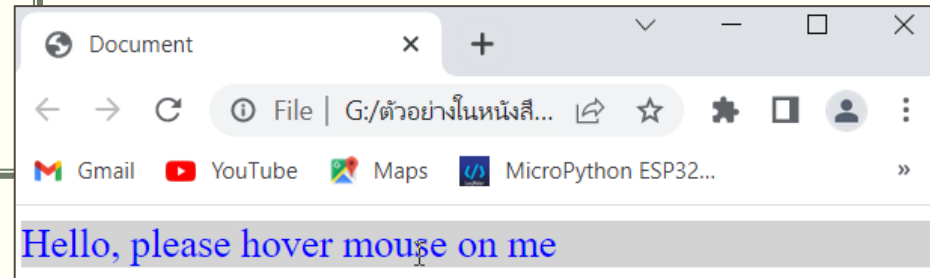
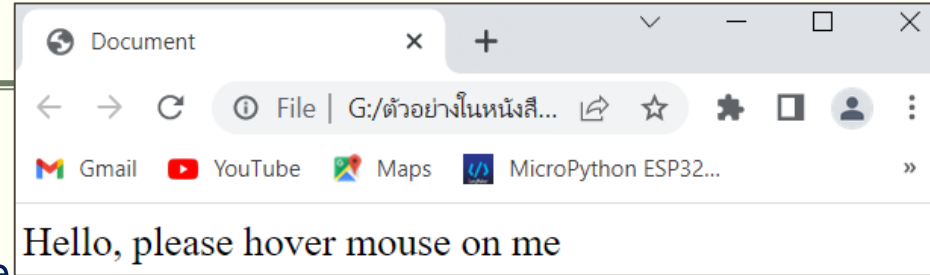
:nth-last-child(an+b)	เหมือนกับ :nth-child แต่จะเลือกอีลิเมนต์ลูกจากระดับต่ำสุดขึ้นมาในระดับสูงสุด	:nth-last-child(1) :nth-last-child(2n)
:first-child	คล้ายคลึงกับ	:nth-child(1)
	:nth-child(1)	
:last-child	คล้ายคลึงกับ :nth-last-child(1)	:nth-last-child(1)
:only-child	คล้ายคลึงกับการกำหนดซีเล็คเตอร์เทียมแบบร่วมกันดังนี้ :nth-child(1):nth-last-child(1)	:nth-child(1):nth-last-child(1)
ซีเล็คเตอร์เทียมประเภทคลาสเชิงนิเสธ (The negation pseudo-class)		
:not()	สไตล์จะข้ามการทำงาน	:not(p) :not(#header)

# ซีเล็คเตอร์เทียม (Pseudo-Selector)

ตัวอย่างการใช้ซีเล็คเตอร์เทียมประเภทคลาส แบบ :hover และ :active

```
<style>
  div { font-size: 16pt; }
  div:hover {
    background-color: lightgray;
    color: blue;
  }
  div:active {
    background-color: red;
    color: yellow;
  }
</style>
```

```
<body>
  <div>
    Hello, please hover mouse on me
  </div>
</body>
```

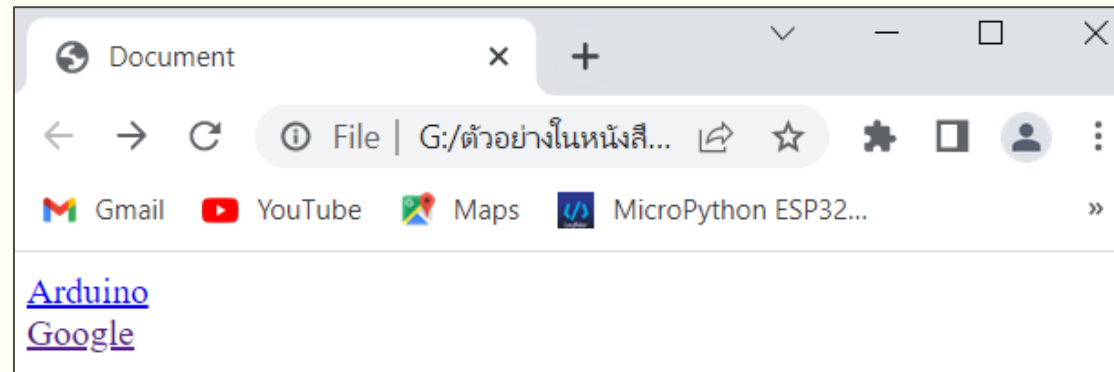


# ซีเล็คเตอร์เทียม (Pseudo-Selector)

ตัวอย่างการใช้ซีเล็คเตอร์เทียมประเภทคลาส แบบ :link และ :visited

```
<style>
  a:link { color: blue; }
  a:visited { color: purple; }
</style>
```

```
<body>
  <div>
    <a href="https://www.arduino.cc">Arduino</a><br>
    <a href="https://www.google.com">Google</a>
  </div>
</body>
```

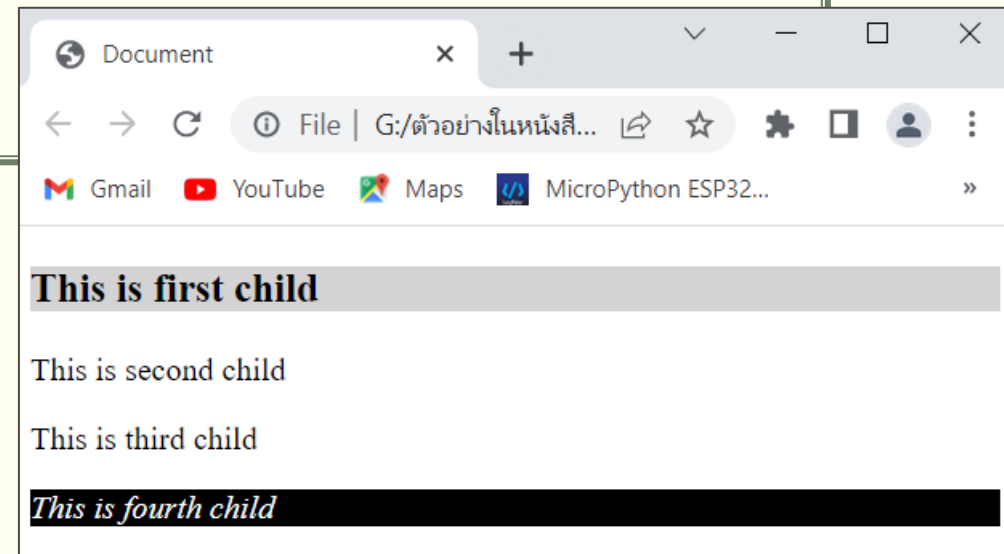


# ซีเล็คเตอร์เทียม (Pseudo-Selector)

ตัวอย่างการใช้ซีเล็คเตอร์เทียมประเภทคลาส แบบ :first-child และ :last-child

```
<style>
  p:first-child {
    background-color: lightgray;
    font-size: 20px;
    font-weight: bold;
  }
  p:last-child {
    background-color: black;
    color: white;
    font-style: italic;
  }
</style>
```

```
<body>
  <div>
    <p>This is first child</p>
    <p>This is second child</p>
    <p>This is third child</p>
    <p>This is fourth child</p>
  </div>
</body>
```

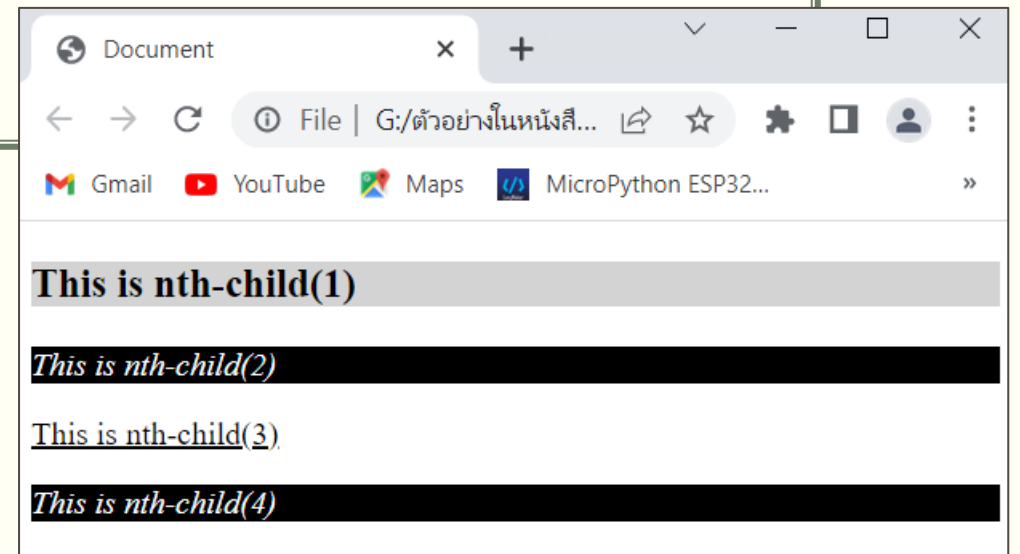


# ซีเล็คเตอร์เทียม (Pseudo-Selector)

ตัวอย่างการใช้ซีเล็คเตอร์เทียมประเภทคลาส แบบ :nth-child(n)

```
<style>
  p:nth-child(1) {
    background-color: lightgray;
    font-size: 20px;
    font-weight: bold;
  }
  p:nth-child(even) {
    background-color: black;
    color: white;
    font-style: italic;
  }
  p:nth-child(3n) { text-decoration: underline; }
</style>
```

```
<body>
  <div>
    <p>This is nth-child(1)</p>
    <p>This is nth-child(2)</p>
    <p>This is nth-child(3)</p>
    <p>This is nth-child(4)</p>
  </div>
</body>
```





# ซีเล็คเตอร์เทียม (Pseudo-Selector)

---

2. ซีเล็คเตอร์เทียมประเภทอิลิเมนต์ (Pseudo-Element) ใช้เมื่อต้องการกำหนดสไตล์เฉพาะบางส่วนให้แก่ข้อมูลภายในอิลิเมนต์หนึ่งๆ โดยจะใช้สัญลักษณ์โคลอนจำนวน 2 ตัว “::” นำหน้าคำสั่ง

รูปแบบการกำหนดซีเล็คเตอร์เทียมประเภทอิลิเมนต์ มีดังนี้

```
Selector :: Pseudo-Element {  
    Property : Value;  
}
```

# ซีเล็คเตอร์เทียม (Pseudo-Selector)

ตารางแสดงซีเล็คเตอร์เทียมประเภทอิลิเมนต์

ซีเล็คเตอร์เทียมประเภทอิลิเมนต์	ความหมาย	ตัวอย่าง
ซีเล็คเตอร์เทียมประเภทคลาสที่มีลักษณะพลวัตร (Dynamic pseudo-classes)		
::first-line	สไตล์จะทำงานเฉพาะข้อมูลในบรรทัดแรกของอิลิเมนต์	<code>p:first-line{ background: red; }</code>
::first-letter	สไตล์จะทำงานเฉพาะอักษรแรกของข้อมูลในอิลิเมนต์	<code>p:first-letter{ color: red; font-size: 200%; }</code>

# ซีเล็คเตอร์เทียม (Pseudo-Selector)

ซีเล็คเตอร์เทียมประเภท อีลีเมนต์	ความหมาย	ตัวอย่าง
::before	สไตส์จะแทรกเนื้อหา ข้อมูลในอีลีเมนต์	<pre>p::before{   content: "Hello!";   display: inline; }</pre>
		<pre>&lt;body&gt;   &lt;p&gt;My world.&lt;/p&gt; &lt;/body&gt;</pre>
::after	สไตส์จะแทรกเนื้อหา ข้อมูลในอีลีเมนต์	<pre>p::after{   content: "Bye!";   display: inline; }</pre>
		<pre>&lt;body&gt;   &lt;p&gt;My world.&lt;/p&gt; &lt;/body&gt;</pre>

# การเขียนคำอธิบายใน CSS

การเขียนคำอธิบายในส่วนของ CSS จะต้องใช้รูปแบบดังนี้

```
/* คำอธิบาย */
```

สิ่งที่อยู่ภายใน /\* ... \*/ จะไม่ถูกนำมาประมวลผล แม้ว่าจะเป็นโค้ดของ CSS ก็ตาม เช่น

```
<style>
  /* รูปแบบเมนู */
  div.mainmenu { color: blue; } /* เมนูหลัก */
  div.submenu { color: navy; } /* เมนูย่อย */
  /*
  #warning { color: red; }
  */
</style>
```

# ลำดับความสำคัญของสไตล์

---

เราสามารถจัดเรียงลำดับความสำคัญของการกำหนดสไตล์แต่ละแบบได้ดังนี้

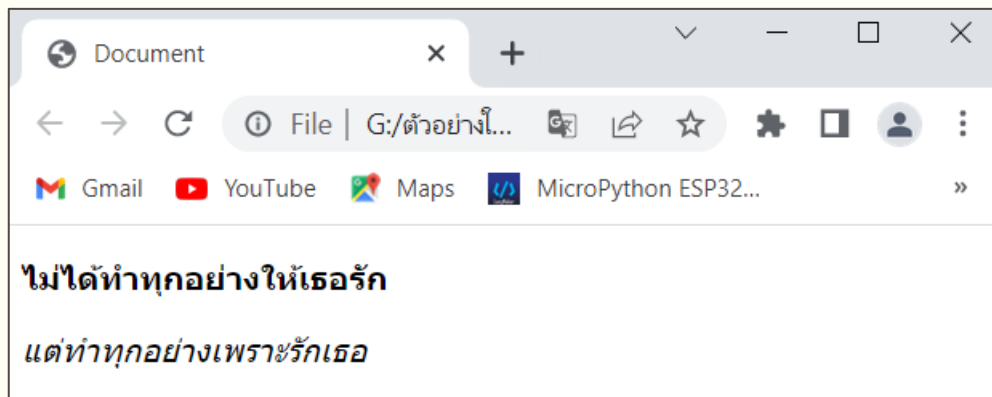
- 1) Inline Style
- 2) Internal Style
- 3) External Style

ลำดับดังกล่าวหมายความว่า Inline Style มีลำดับความสำคัญสูงสุด ตามด้วย Internal และ External ตามลำดับ ดังนั้น ถ้าเรากำหนดพร็อพเพอร์ตี้แบบซ้ำซ้อนกัน รูปแบบที่มีความสำคัญสูงกว่าจะถูกนำมาใช้ เช่น หากเราต้องการให้บรรทัดที่สองต่างจากบรรทัดอื่น ก็อาจเขียนพร็อพเพอร์ตี้แบบ Inline แทนที่แบบ Internal ดังโค้ดถัดไป

# ลำดับความสำคัญของสไตล์

```
<style>
  p { font-weight: bold; }
</style>
```

```
<body>
  <p>ไม่ได้ทำทุกอย่างให้เธอรัก</p>
  <p style="font-weight: normal; font-style: italic">
    แต่ทำทุกอย่างเพราะรักเธอ
  </p>
</body>
```



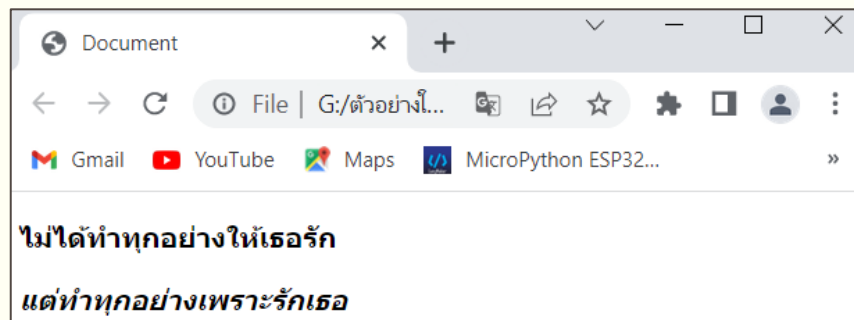
จากโค้ด บรรทัดที่ 2 เรากำหนดสไตล์แบบ Inline ซึ่งมีลำดับความสำคัญเหนือกว่าแบบ Internal ดังนั้นบรรทัดนี้จึงมีรูปแบบตามที่เรากำหนดแบบ Inline เราอาจนำไปใช้ในลักษณะอื่นๆ อีก เช่น หากใช้สไตล์แบบ External ถ้าที่เพจใดต้องการรูปแบบที่แตกต่างออกไป ก็สามารถ Override โดยใช้สไตล์แบบ Internal หรือ Inline ซึ่งมีลำดับความสำคัญเหนือกว่าแบบ External

# การกำหนดความสำคัญด้วย **!important**

พร็อพเพอร์ตี้ของสไตล์ที่มีลำดับความสำคัญสูงกว่า จะสามารถ Override พร็อพเพอร์ตี้ของสไตล์ที่มีลำดับความสำคัญต่ำกว่าได้ แต่อย่างไรก็ตาม ในบางพร็อพเพอร์ตี้อาจมีความสำคัญต่อเว็บเพจ ถ้าหากปล่อยให้มีการ Override การแสดงผลอาจผิดเพี้ยนไปได้ ดังนั้น จึงมีวิธีป้องกันไม่ให้พร็อพเพอร์ตี้เหล่านั้นถูก Override โดยการกำหนดคีย์เวิร์ด **!important** ตามหลังค่าของพร็อพเพอร์ตี้ที่ต้องการป้องกัน เช่น

```
<style>
  p { font-weight: bold !important; }
</style>
```

```
<body>
  <p>ไม่ได้ทำทุกอย่างให้เธอรัก</p>
  <p style="font-weight: normal; font-style: italic">
    แต่ทำทุกอย่างเพราะรักเธอ
  </p>
</body>
```



จากผลลัพธ์จะพบว่า แม้สไตล์แบบ Inline จะมีลำดับความสำคัญสูงกว่าแบบ Internal แต่พร็อพเพอร์ตี้ที่กำหนดคีย์เวิร์ด **!important** จะมีความสำคัญสูงสุด ดังนั้น แม้จะกำหนดค่าเป็นอย่างอื่นในภายหลัง ก็ไม่มีผลใดๆ

# การกำหนดสี (Color)

---

เนื่องจากการกำหนดสีนั้นถือเป็นเรื่องพื้นฐานที่สำคัญอย่างหนึ่งของ CSS เพราะมีพร็อพเพอร์ตี้หลายอย่างที่เกี่ยวข้องกับสี เช่น สีข้อความ พื้นหลัง เส้นขอบ เป็นต้น ซึ่งใน CSS มีทางเลือกในการกำหนดสีที่น่าสนใจดังต่อไปนี้

- **การระบุชื่อสี** โดยต้องเป็นชื่อในภาษาอังกฤษ ซึ่งเหมาะสำหรับการเลือกใช้สีพื้นฐานทั่วไป โดยตัวอย่างชื่อสีที่น่าสนใจ เช่น red, green, blue, black, white, yellow เป็นต้น สามารถดูรายชื่อสีเพิ่มเติมได้ที่

<https://htmlcolorcodes.com/color-names>

```
<style>
  div { color: navy; }
  span { background-color: black; }
</style>
```



# การกำหนดสี (Color)

- ฟังก์ชัน `rgb()` ซึ่งเป็นการผสมกันระหว่างแม่สีทั้งสามคือ แดง (Red) เขียว (Green) และน้ำเงิน (Blue) โดยใช้ฟังก์ชัน `rgb()` หรือ `rgba()` ดังรูปแบบต่อไปนี้

`rgb(red, green, blue)` หรือ `rgba(red, green, blue, alpha)`

- กำหนดเป็นตัวเลขระหว่าง 0-255 เช่น `rgb(0, 0, 255)` หรือ `rgb(100, 150, 200)`
- ค่า `alpha` เป็นระดับความโปร่งใส ซึ่งสามารถกำหนดเป็นตัวเลขระหว่าง 0.0 - 1.0 โดยค่า 0.0 หมายถึง โปร่งใส (Transparent) ส่วน 1.0 หมายถึง ไม่โปร่งใสหรือทึบ (Opaque) ทั้งนี้หากเราไม่ระบุค่า `alpha` ก็เปรียบเทียบกับเท่ากับการกำหนดค่าเป็น 1.0 นั่นเอง

```
<style>
  div { color: rgb(100, 200, 150); }
  span { background-color: rgba(0, 120, 220, 0.5); }
</style>
```

# การกำหนดสี (Color)

- **Hexadecimal** หรือเลขฐาน 16 จะประกอบด้วยอักขระ 6 ตัว ในรูปแบบต่อไปนี้

$\#H_1H_2H_3H_4H_5H_6$

- $H_x$  คืออักขระที่ใช้เขียนแทนเลขฐาน 16 ซึ่งอาจเป็นเลข 0-9 หรือ A-F หรือ a-f
- ต้องเขียนเครื่องหมาย # กำกับไว้ข้างหน้า เช่น #10AF70, #aabbcc
- ในกรณีที่อักขระแต่ละคู่ที่อยู่ติดกันมีค่าซ้ำกัน เช่น #336699, #CCDDEE, #77aa11 เราสามารถเขียนย่อให้เหลือเพียง 3 ตัว โดยคู่ที่ซ้ำกันอาจเขียนอักขระเพียงตัวเดียว เช่น #336699 อาจเขียนเป็น #369 เป็นต้น
- ตรวจสอบค่า Hex ของแต่ละสีได้ที่ <https://htmlcolorcodes.com>

```
<style>
    div { color: #808080; }
    span { background-color: #ff0; }      /* ffff00 */
    #title { color: #87ceeb; }
</style>
```

# การกำหนดหน่วยแบบต่างๆ (Unit)

ในการกำหนดค่าของพรีอปรเตอร์ที่เกี่ยวกับขนาด เช่น ความกว้าง ความยาว ขนาดของฟอนต์ นอกจากการกำหนดค่าที่เป็นตัวเลขแล้ว เราจำเป็นต้องระบุหน่วยของตัวเลขนั้นลงไปด้วย โดยหน่วยที่นิยมใช้กันส่วนใหญ่ใน CSS คือ

px	มาจากคำว่า pixel (พิกเซล) เป็นการกำหนดขนาดแบบคงที่
pt	มาจากคำว่า point โดยที่ $1\text{pt} = 1.33\text{px}$
em	ใช้ในการกำหนดขนาดหรือระยะ โดยเทียบกับขนาดล่าสุดของฟอนต์ที่กำหนดให้อิเลเมนต์นั้น หรืออิเลเมนต์ที่เป็น Parent ของมัน หรืออิเลเมนต์ที่อยู่ก่อนนี้ เช่น หากกำหนดขนาดของฟอนต์เป็น 16px แล้วให้อิเลเมนต์มีความสูงเป็น 1.5em แสดงว่าความสูงจะมีค่าเป็น 1.5 เท่าของขนาดฟอนต์ นั่นคือ $1.5 \times 16 = 24\text{px}$
rem	ใช้ในการกำหนดขนาดหรือระยะ โดยเทียบกับขนาดของ root element (หรือ html) ซึ่งโดยปกติ เบราวเซอร์จะตั้งค่าดีฟอลต์เอาไว้ที่ 16px นั่นแสดงว่า $1\text{rem} = 16\text{px}$ ถ้าเราระบุขนาดเป็น 2rem แสดงว่าขนาดที่แท้จริงคือ $2 \times 16 = 32\text{px}$ แต่หากเราตั้งค่าของ root element ใหม่ใน CSS เช่น <code>html { font-size: 14px }</code> แสดงว่า $1\text{rem} = 14\text{px}$ เป็นต้น

การกำหนดหน่วยต้องเขียนติดกับตัวเลขที่กำหนดขนาด เช่น 20px หรือ 2em เป็นต้น สำหรับในปัจจุบันเรานิยมใช้หน่วย rem กันมาก โดยเฉพาะอย่างยิ่งการเทียบขนาดใน Bootstrap

# การกำหนดหน่วยแบบต่างๆ (Unit)

unit.html

```
<style>
```

```
/* html { font-size: 16px; } */
```

```
p:nth-child(1) { font-size: 16pt; }
```

```
p:nth-child(2) { font-size: 16px; }
```

```
p:nth-child(3) { font-size: 1rem; }
```

```
p:nth-child(4) { font-size: 1.5rem; }
```

```
p:nth-child(5) { font-size: 2rem; }
```

```
</style>
```

```
<body>
```

```
<div>
```

```
<p>size: 16pt</p>
```

```
<p>size: 16px</p>
```

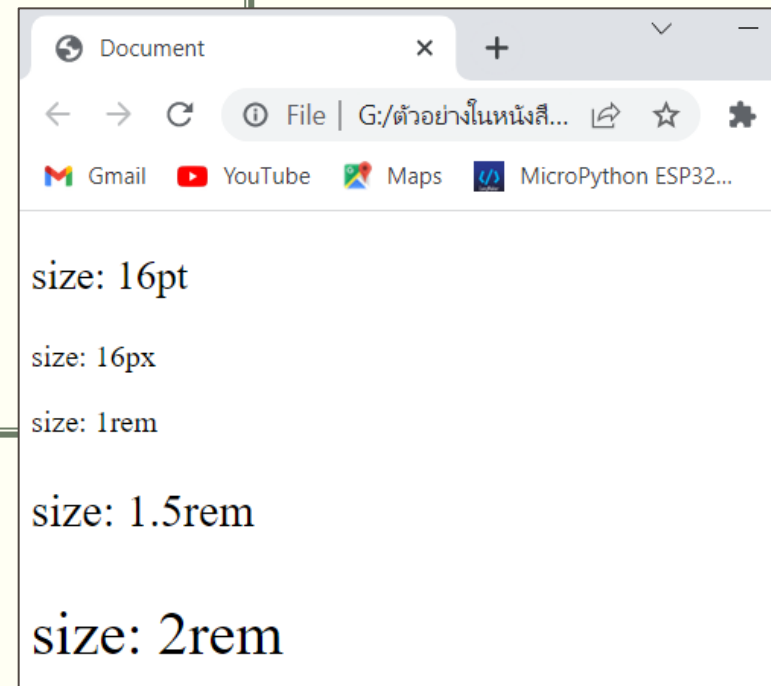
```
<p>size: 1rem</p>
```

```
<p>size: 1.5rem</p>
```

```
<p>size: 2rem</p>
```

```
</div>
```

```
</body>
```



# การกำหนดสไตล์เกี่ยวกับฟอนต์ (Font)

การจัดรูปแบบฟอนต์ด้วย CSS นั้น มีหลายพรีอเพอร์ติวี่ที่เกี่ยวข้องโดยตรง และพรีอเพอร์ติวี่แต่ละอันก็มีข้อกำหนดปลีกย่อยที่แตกต่างกันออกไป ซึ่งสามารถสรุปได้ดังนี้

font-family	สำหรับกำหนดชนิดฟอนต์ อาจจะระบุเพียงชนิดเดียวหรือหลายชนิดก็ได้ โดยค่าที่กำหนดได้คือ <ul style="list-style-type: none"><li>• ชนิดฟอนต์</li><li>• ชนิดฟอนต์ 1, ชนิดฟอนต์ 2, ชนิดฟอนต์ 3, ...</li></ul>
font-size	สำหรับกำหนดขนาดของฟอนต์ โดยทั่วไปเราจะระบุขนาดเป็นตัวเลขตามด้วยหน่วย (เช่น px) หรือไม่ก็ระบุเป็นค่าคงที่ก็ได้ นั่นคือให้เลือกอย่างใดอย่างหนึ่งดังต่อไปนี้ <ul style="list-style-type: none"><li>• &lt;ขนาด&gt;&lt;หน่วย&gt;</li><li>• xx-small, x-small, small, medium, large, x-large, xx-large แนวทางการนำไปใช้ เช่น div { font-size: x-large }</li></ul>
font-weight	ขนาดความหนาหรือน้ำหนักของฟอนต์ โดยค่าที่สำคัญคือ <ul style="list-style-type: none"><li>• lighter (ตัวบาง)</li><li>• bold (ตัวหนา)</li></ul>

# การกำหนดสไตล์เกี่ยวกับฟอนต์ (Font)

font-style	รูปแบบฟอนต์ที่เป็นตัวเอียง โดยค่าที่สำคัญคือ italic (ตัวเอียง)
font-variant	ลักษณะตัวพิมพ์ คือเปลี่ยนตัวพิมพ์เล็กให้เป็นตัวพิมพ์ใหญ่ที่มีขนาดเล็กลง โดยค่าที่สำคัญคือ <b>small-caps</b>
font	การกำหนดพร็อพเพอร์ตี้ของฟอนต์แบบรวม โดยระบุค่าอย่างใดอย่างหนึ่งดังนี้ <ul style="list-style-type: none"><li>• &lt;font-style&gt;&lt;font-variant&gt;&lt;font-weight&gt;&lt;font-size&gt;&lt;font-family&gt;</li><li>• &lt;font-size&gt;&lt;font-family&gt;</li></ul>

- ถ้าชนิด (ชื่อ) ฟอนต์ มีช่องว่างอยู่ด้วย ให้กำหนดไว้ในเครื่องหมาย ‘ ’ หรือ “ ”
- เราสามารถกำหนดฟอนต์ได้มากกว่า 1 ชนิด เพื่อกรณีที่เว็บเบราว์เซอร์ไม่รองรับฟอนต์ชนิดแรก ก็จะนำชนิดที่กำหนดไว้ในลำดับถัดไปมาใช้แทน โดยคั่นฟอนต์แต่ละชนิดด้วยเครื่องหมาย , เช่น

```
div { font-family: Tahoma; }  
span { font-family: 'Times New Roman'; }  
p { font-family: Tahoma, 'Cordia New', "Microsoft Sans Serif"; }
```

# การกำหนดสไตล์เกี่ยวกับฟอนต์ (Font)

---

- แต่ละค่าของพร็อพเพอร์ตี้ font ให้แยกโดยใช้ช่องว่าง ห้ามใช้เครื่องหมาย , ยกเว้น font-family กรณีที่ต้องการระบุฟอนต์หลายๆ ชนิด เช่น

```
div { font: italic normal bold 20px Tahoma; }  
span { font: normal small-caps bolder 18px Tahoma, 'Cordia New'; }
```

- ค่า 3 แรกของพร็อพเพอร์ตี้ font คือ font-style, font-variant และ font-weight อาจกำหนดเพียงบางอันหรือไม่มีเลยก็ได้ แต่พร็อพเพอร์ตี้ 2 อันหลังคือ font-size และ font-family ต้องมีครบทั้งสองอัน เช่น

```
div { font: italic bold 20px Tahoma; }  
span { font: bolder 20px Tahoma; }  
p { font: 20px Tahoma; }
```

# การกำหนดสไตล์เกี่ยวกับข้อความ (Text)

ลักษณะการจัดรูปแบบข้อความ เช่น การขีดเส้นแบบต่างๆ การจัดแนวซ้าย-ขวา การกำหนดระยะห่างและอื่นๆ อีกหลายแบบ โดยพรีอปรเตอร์ที่เกี่ยวข้องคือ

text-decoration	สำหรับการตกแต่งข้อความ โดยค่าที่กำหนดคือ <ul style="list-style-type: none"><li>• <b>Underline</b> ขีดเส้นใต้</li><li>• <b>line-through</b> ขีดเส้นใต้ทับข้อความ เหมือนกับแท็ก &lt;del&gt;</li></ul>
text-align	จัดแนวข้อความในแนวนอนว่าจะให้ชิดด้านใด โดยค่าที่กำหนดได้คือ <ul style="list-style-type: none"><li>• <b>left, right, center</b> ชิดด้านซ้าย, ขวา และกึ่งกลางตามลำดับ</li><li>• <b>justify</b> การกระจายข้อความให้ทุกบรรทัดมีความกว้างเท่ากัน คล้ายกับนิตยสารหรือหนังสือพิมพ์</li></ul>
vertical-align	จัดข้อความในแนวตั้ง โดยค่าที่มักนำไปใช้กันส่วนใหญ่คือ <ul style="list-style-type: none"><li>• <b>sub</b> ทำตัวห้อย เหมือนกับแท็ก &lt;sub&gt;</li><li>• <b>super</b> ทำตัวยก เหมือนกับแท็ก &lt;sup&gt;</li><li>• <b>middle</b> กรณีต้องการจัดข้อความในแนวตั้ง ต้องใช้ร่วมกับพรีอปรเตอร์ display: table-cell จึงจะได้ผล</li></ul>



# การกำหนดสไตล์เกี่ยวกับข้อความ (Text)

text-indent	กำหนดระยะเยื้องของย่อหน้าจากแนวขอบด้านซ้ายสำหรับบรรทัดแรกของย่อหน้าเท่านั้น ส่วนบรรทัดถัดไปจะเริ่มที่แนวขอบตามปกติ โดยระบุค่าในรูปแบบ <ul style="list-style-type: none"><li>• <code>&lt;ค่าระยะ&gt;&lt;หน่วย&gt;</code> เช่น 10px</li></ul>
text-transform	การกำหนดลักษณะตัวพิมพ์ของข้อความ โดยค่าที่กำหนดได้คือ <ul style="list-style-type: none"><li>• <code>lowercase</code> ตัวพิมพ์เล็กทั้งหมด เช่น hello world</li><li>• <code>uppercase</code> ตัวพิมพ์ใหญ่ทั้งหมด เช่น HELLO WORLD</li><li>• <code>capitalize</code> ตัวแรกของคำเป็นตัวพิมพ์ใหญ่ เช่น Hello World</li></ul>

text-format.html

```
<style>
p.justify {
    background-color: #cde;
    text-align: justify;
    width: 300px;
}
```

```
span.super {
    vertical-align: super;
}
span.sub {
    vertical-align: sub;
}
```

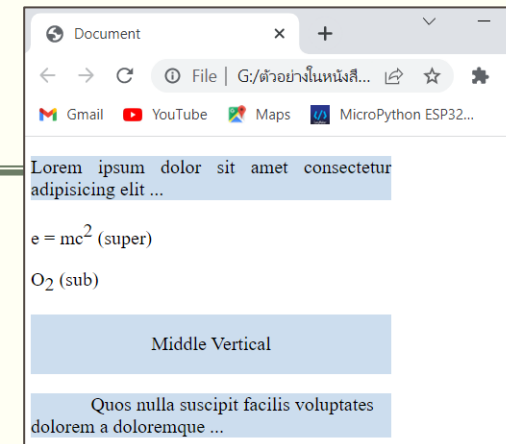
# การกำหนดสไตล์เกี่ยวกับข้อความ (Text)

```
div.middle-vert {  
    background-color: #cde;  
    width: 300px;  
    height: 50px;  
    text-align: center;  
    vertical-align: middle;  
    display: table-cell;  
}
```

```
p.indent {  
    background-color: #cde;  
    text-indent: 50px;  
    width: 300px;  
}
```

</style>

```
<body>  
    <p class="justify">  
        Lorem ipsum dolor sit amet consectetur adipisicing elit ...  
    </p>  
    <p>e = mc<span class="super">2</span> (super)</p>  
    <p>O<span class="sub">2</span> (sub)</p>  
    <div class="middle-vert">Middle Vertical</div>  
    <p class="indent">  
        Quos nulla suscipit facilis voluptates dolorem a doloremque ...  
    </p>  
</body>
```



# การกำหนดสไตล์ของเส้นขอบ (Border)

เส้นขอบ (Border) คือด้านทั้งสี่ที่ล้อมรอบอิลิเมนต์ โดยเส้นขอบแต่ละด้าน จะแทนด้วยคำว่า top, right, bottom และ left ซึ่งองค์ประกอบของเส้นขอบที่เราจะปรับแต่งคือ รูปแบบเส้น (style) ความหนา (width) และสี (color) โดยพร็อพเพอร์ตี้ที่เกี่ยวกับเส้นขอบมีดังนี้ (ให้ xxx แทนด้านใดด้านหนึ่ง คือ top, right, bottom และ left)

border-xxx-style	รูปแบบเส้นขอบ เช่น เส้นทึบ หรือเส้นประ แบบแยกแต่ละด้าน โดยค่าที่กำหนดได้คือ <b>solid, dotted, dashed, double, groove, inset, outset, ridge, none</b>
border-style	กำหนดรูปแบบเส้นขอบของทั้ง 4 ด้านในพร็อพเพอร์ตี้เดียวกัน
border-xxx-width	ขนาดของเส้นขอบแบบแยกแต่ละด้าน ค่าที่กำหนดได้คือ <ul style="list-style-type: none"><li>• <b>&lt;ขนาด&gt;&lt;หน่วย&gt;</b> เช่น 2px</li><li>• <b>thin, medium, thick</b> ซึ่งเป็นขนาดบาง ปานกลาง และหนา ตามลำดับ</li></ul>
border-width	ขนาดเส้นขอบของทั้ง 4 ด้าน
border-xxx-color	สีของเส้นขอบแบบแยกแต่ละด้าน โดยค่าที่กำหนดได้มีดังนี้ <ul style="list-style-type: none"><li>• <b>color-name</b> เช่น blue</li><li>• <b>#hex</b> เช่น #abcdef</li><li>• <b>rgb(red, green, blue)</b> เช่น rgb(100, 150, 250)</li></ul>

# การกำหนดสไตล์ของเส้นขอบ (Border)

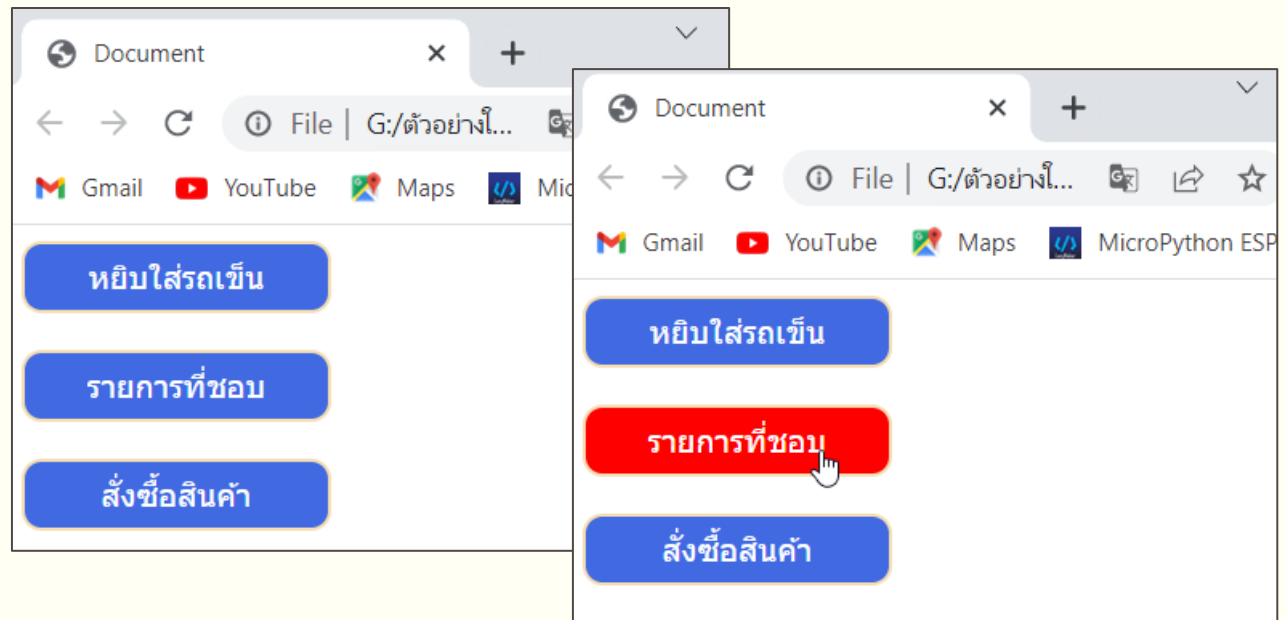
border-color	สีเส้นขอบของทั้ง 4 ด้าน
border-xxx	กำหนดลักษณะทั้ง 3 อย่างคือ style, width และ color แบบแยกแต่ละด้านในพร็อพเพอร์ตี้เดียวกัน ด้วยค่าต่อไปนี้ <ul style="list-style-type: none"><li>• <code>&lt;style&gt;&lt;width&gt;&lt;color&gt;</code> สามารถสลับกันได้</li><li>• <code>none</code> ใช้ในกรณีที่ไม่ต้องการให้มีเส้นขอบ</li></ul>
border	กำหนดลักษณะทั้ง 3 อย่างคือ style, width และ color ของทั้ง 4 ด้านในพร็อพเพอร์ตี้เดียวกัน โดยใช้หลักการเช่นเดียวกับ border-style
border-xxx-radius	กำหนดค่ารัศมี Rx และ Ry แบบแยกแต่ละมุม ด้วยค่าดังนี้ <ul style="list-style-type: none"><li>• <code>&lt;Rx&gt;&lt;Ry&gt;</code> ระบุหน่วยด้วย เช่น 10px 20px</li><li>• R กรณีที่ Rx = Ry อาจกำหนดเพียงค่าเดียว</li></ul>
border-radius	กำหนดค่ารัศมี Rx และ Ry สำหรับทั้ง 4 มุมในพร็อพเพอร์ตี้เดียวกัน

# การกำหนดสไตล์ของเส้นขอบ (Border)

border-radius.html

```
<style>
  a.btn {
    background-color: royalblue;
    color: white;
    width: 150px;
    height: 32px;
    border-radius: 10px;
    border: solid 2px wheat;
    text-align: center;
    vertical-align: middle;
    display: table-cell;
    text-decoration: none;
  }
  a:hover {
    background-color: red;
  }
</style>
```

```
<body>
  <a href="#" class="btn">หยิบใส่รถเข็น</a><br>
  <a href="#" class="btn">รายการที่ชอบ</a><br>
  <a href="#" class="btn">สั่งซื้อสินค้า</a>
</body>
```



# การกำหนดระยะ (Padding) (ระยะห่างภายใน)

---

ถ้าเราต้องการเว้นระยะห่างระหว่างแนวขอบกับเนื้อหาใน CSS ก็มีพร็อพเพอร์ตี้ padding ที่ใช้กำหนดระยะดังกล่าวนี้ คือ (ให้ xxx แทนด้านทั้ง 4 คือ top, right, bottom และ left)

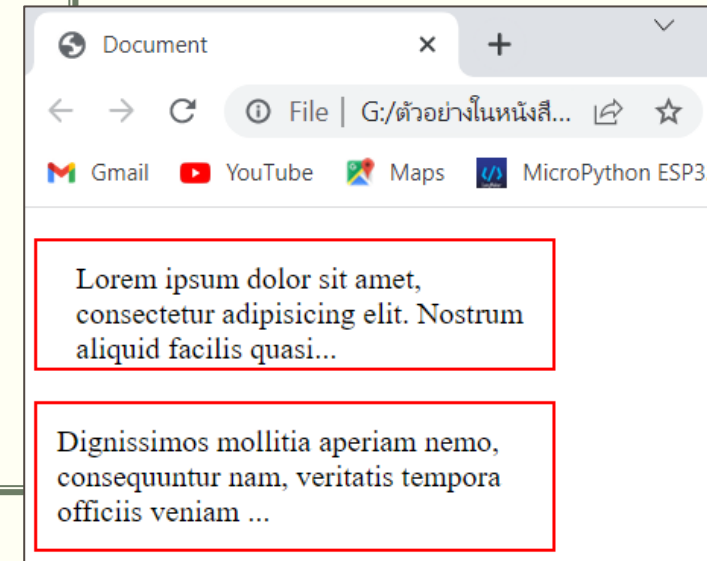
padding-xxx	กำหนดระยะห่างระหว่างแนวขอบแบบแยกแต่ละด้าน โดยค่าที่กำหนดได้คือ <ul style="list-style-type: none"><li>• &lt;ระยะ&gt;&lt;หน่วย&gt; เช่น padding-top: 5px</li></ul>
padding	กำหนดระยะห่างระหว่างแนวขอบกับเนื้อหาทั้ง 4 ด้านในพร็อพเพอร์ตี้เดียวกัน

# การกำหนดระยะ (Padding) (ระยะห่างภายใน)

padding.html

```
<style>
  p {
    border: solid 2px red;
    width: 250px;
  }
  #p1 {
    padding-top: 10px;
    padding-left: 20px;
  }
  #p2 { padding: 10px; }
</style>
```

```
<body>
  <p id="p1">
    Lorem ipsum dolor sit amet, consectetur
    adipisicing elit. Nostrum aliquid facilis quasi...
  </p>
  <p id="p2">
    Dignissimos mollitia aperiam nemo,
    consequuntur nam, veritatis tempora officiis
    veniam ...
  </p>
</body>
```



# การกำหนดระยะ (Margin) (ระยะห่างภายนอก)

หากเราจัดวางอีลีเมนต์ต่อเนื่องกัน ไม่ว่าจะเป็นในแนวนอนหรือแนวตั้ง อีลีเมนต์เหล่านี้อาจจะถูกละวางชิดกัน หรือมีระยะห่างที่ไม่สม่ำเสมอจนดูไม่สวย ถ้าเราต้องการเว้นระยะห่างระหว่างอีลีเมนต์ ใน CSS ก็มีกลุ่มพร็อพเพอร์ตี้ margin ที่ใช้กำหนดระยะดังกล่าวนี้ คือ (ให้ xxx แทนด้านใดด้านหนึ่ง คือ top, right, bottom และ left)

margin-xxx	กำหนดระยะห่างระหว่างอีลีเมนต์ แบบแยกแต่ละด้าน โดยค่าที่กำหนดได้คือ <ul style="list-style-type: none"><li>• &lt;ระยะ&gt;&lt;หน่วย&gt; เช่น margin-top: 5px</li><li>• auto นิยมใช้ในการจัดอีลีเมนต์ให้อยู่กึ่งกลาง Parent ของมัน</li></ul>
margin	กำหนดระยะห่างระหว่างอีลีเมนต์ทั้ง 4 ด้านในพร็อพเพอร์ตี้เดียวกัน



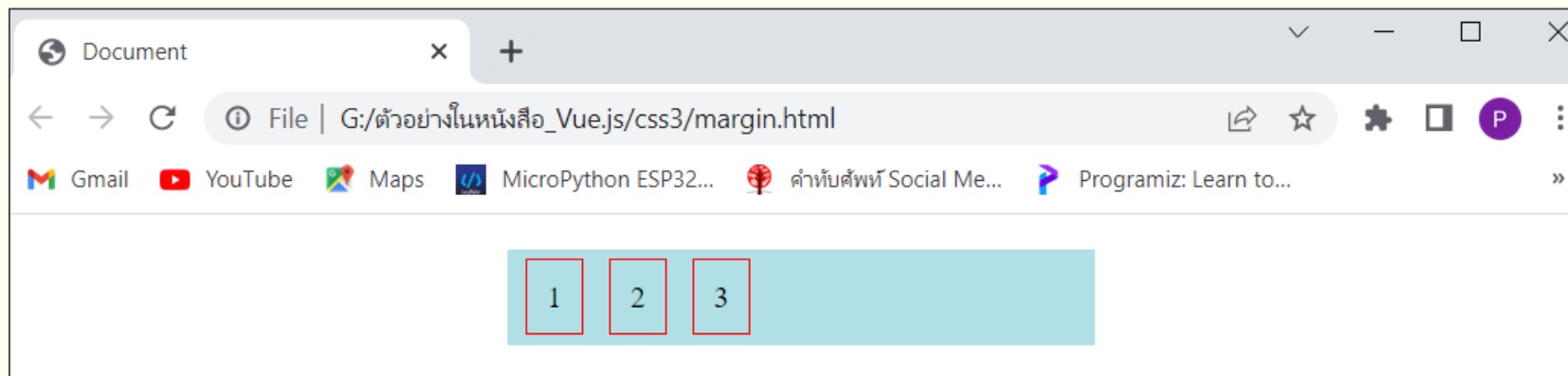
# การกำหนดระยะ (Margin) (ระยะห่างภายนอก)

margin.html

```
<style>
  body { padding: 10px; }
  div.container {
    width: 300px;
    margin: auto;
    padding: 5px;
    background-color: powderblue;
  }
```

```
div.item {
  padding: 10px;
  margin-left: 5px;
  margin-right: 5px;
  border: solid 1px red;
  display: inline-block;
}
</style>
```

```
<body>
  <div class="container">
    <div class="item">1</div>
    <div class="item">2</div>
    <div class="item">3</div>
  </div>
</body>
```



# การกำหนดสไตล์การแสดงผล (Display)

ตามปกติแล้ว การแสดงผลของแต่ละอิลิเมนต์จะแตกต่างกัน เช่น `<span>` จะเป็นแบบ Inline คือสามารถแสดงในบรรทัดเดียวกันได้ ขณะที่ `<div>` หรือ `<p>` จะเป็นแบบบล็อก แต่ใน CSS ยังมีพร็อพเพอร์ตี้ที่สามารถเปลี่ยนรูปแบบการแสดงผลเป็นลักษณะอื่นๆ ได้อีก โดยใช้พร็อพเพอร์ตี้ `display` ซึ่งค่าที่กำหนดได้มีดังนี้

display	<p>กำหนดลักษณะการแสดงผลของอิลิเมนต์ โดยค่าที่นิยมใช้กันเป็นส่วนใหญ่คือ</p> <ul style="list-style-type: none"><li>• <b>inline</b> แสดงอิลิเมนต์ภายในบรรทัดเดียวกันได้ เช่นเดียวกับ <code>&lt;span&gt;</code></li><li>• <b>block</b> แสดงอิลิเมนต์แบบขึ้นบรรทัดใหม่ เช่นเดียวกับ <code>&lt;div&gt;</code></li><li>• <b>inline-block</b> เป็นรูปแบบผสมระหว่าง inline กับ block โดยจะสร้างบล็อกของเนื้อหาให้อยู่ภายในบรรทัดนั้น</li><li>• <b>none</b> ซ่อนอิลิเมนต์ โดยไม่เกิดช่องว่างตรงตำแหน่งที่อิลิเมนต์ถูกซ่อน</li><li>• <b>table-cell</b> แสดงผลแบบเซลล์ตาราง (<code>&lt;td&gt;</code>) สามารถใช้ร่วมกับพร็อพเพอร์ตี้ <code>vertical-align</code> เพื่อจัดเนื้อหาในแนวตั้งได้</li></ul>
---------	---

# การกำหนดสไตล์การแสดงผล (Display)

display.html

```
<style>
```

```
span {  
  background: skyblue;  
  font: 12pt tahoma;  
  color: crimson;  
  margin: 5px;  
  display: block;  
  padding: 5px;  
}
```

```
p, div {  
  border: solid 2px tomato;  
  width: 150px;  
}  
p { display: inline; }  
div { display: inline-block; }  
</style>
```

```
<body>
```

`<span>`The display property`</span>`  
specifies the display behavior  
`<p>`(the type of rendering box)`</p>`.  
In HTML,  
`<div>`the default display property value is  
taken from the HTML specifications`</div>`  
or from default style sheet.

```
</body>
```

