



การเขียนโปรแกรมบนเว็บ (Web Programming) 4122306



อาจารย์ ดร. บัณฑิต สงเนียม

บทที่ 3. แนะนำ CSS เบื้องต้น_Part_2



โดย ผู้ช่วยศาสตราจารย์ ดร. บัณฑิต สงเนียม

สาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏพระนคร

เว็บไซต์ <http://www.siam2dev.net>

Agenda

3.1 HTML เบื้องต้น

3.2 CSS เบื้องต้น

3.2.1 CSS คืออะไร

3.2.2 ประโยชน์ของ CSS

3.2.3 CSS Syntax

3.2.4 เวอร์ชันของ CSS

3.2.5 รูปแบบการเขียน CSS

3.3 การใช้งาน Visual Studio Code เบื้องต้น

CSS คืออะไร

CSS คือ ภาษาที่ใช้สำหรับตกแต่งเอกสาร HTML/XHTML ให้มีหน้าตา สี สัน ระยะห่าง พื้นหลัง เส้นขอบและอื่นๆ ตามที่ต้องการ CSS ย่อมาจาก Cascading Style Sheets มีลักษณะเป็นภาษาที่มีรูปแบบในการเขียน Syntax แบบเฉพาะและได้ถูกกำหนดมาตรฐานโดย W3C เป็นภาษาหนึ่งในการตกแต่งเว็บไซต์ ได้รับความนิยมอย่างแพร่หลาย

3.2 CSS เบื้องต้น

3.2.1 CSS คืออะไร

ในตัวอย่างข้างต้นเราใช้แท็ก `<h1>` เพื่อกำหนดหัวข้อเท่านั้น และแยกส่วนรูปแบบการแสดงผลให้เป็นหน้าที่ของ CSS และกำหนดให้แสดงผลสีฟ้า ดังนั้นจึงกล่าวได้ว่า CSS คือภาษาที่ใช้ในการกำหนดรูปแบบหรือสไตล์ (Style) สำหรับ HTML

การแยกการแสดงผลออกจากข้อมูลช่วยให้ง่ายต่อการเปลี่ยนแปลงรูปแบบในอนาคต เช่น ถ้าเราต้องการเปลี่ยนสีให้กับหัวข้อในแท็ก `<h1>` ในทุกๆเว็บเพจ เราเพียงเปลี่ยนที่ไฟล์ CSS เท่านั้น (ไฟล์ CSS สามารถใช้ได้กับทุกๆเว็บเพจที่ต้องการ) ปัจจุบัน CSS เป็นที่นิยมอย่างมากในการจัดรูปแบบให้กับเว็บเพจ ซึ่งก่อนการมาของ CSS นั้นเราใช้ตารางในการจัดเลย์เอาท์ และใช้พร็อพเพอร์ตี้ของ HTML ในการจัดรูปแบบ ซึ่งยุ่งยากมากหากเราต้องการเปลี่ยนรูปแบบของทุกเว็บเพจในเว็บไซต์ สำหรับสเปคและข้อมูลอย่างละเอียดของ CSS สามารถดูได้ที่

<http://www.w3.org/Style/CSS> ซึ่งกำกับและดูแลโดย World Wide Web Consortium (W3C)

HTML:

```
<h1>Welcome to my web site</h1>
```

CSS:

```
h1 { color: #0000FF; /* Blue */ }
```

CSS มีประโยชน์อย่างหลากหลาย ซึ่งได้แก่

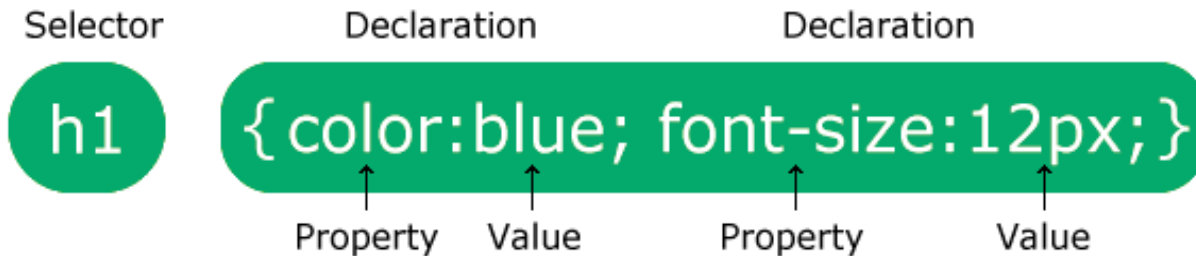
- 1. ช่วยให้เนื้อหาภายในเอกสาร HTML มีความเข้าใจได้ง่ายขึ้นและในการแก้ไขเอกสารก็สามารถทำได้ง่ายกว่าเดิม เพราะการใช้ CSS จะช่วยลดการใช้ภาษา HTML ลงได้ในระดับหนึ่ง และแยกแยะระหว่างเนื้อหากับรูปแบบในการแสดงผลได้อย่างชัดเจน**
- 2. ทำให้สามารถดาวน์โหลดไฟล์ได้เร็ว เนื่องจาก code ในเอกสาร HTML ลดลง จึงทำให้ไฟล์มีขนาดเล็กลง**
- 3. สามารถกำหนดรูปแบบการแสดงผลจากคำสั่ง style sheet ชุดเดียวกัน ให้มีการแสดงผลในเอกสารแบบเดียวกันทั้งหน้าหรือในทุกๆ หน้าได้ ช่วยลดเวลาในการปรับปรุงและทำให้การสร้างเอกสารบนเว็บมีความรวดเร็วยิ่งขึ้น นอกจากนี้ยังสามารถควบคุมการแสดงผล ให้คล้ายหรือเหมือนกันได้ในหลาย Web Browser**
- 4. ช่วยในการกำหนดการแสดงผลในรูปแบบที่มีความเหมาะสมกับสื่อต่าง ๆ ได้เป็นอย่างดี**
- 5. ทำให้เว็บไซต์มีความเป็นมาตรฐานมากขึ้นและมีความทันสมัย สามารถรองรับการใช้งานในอนาคตได้ดี**

ที่มา : <https://blog.sogoodweb.com/Article/Detail/79237/CSS-คืออะไร-มีประโยชน์-อย่างไร>

3.2 CSS เบื้องต้น

3.2.3 CSS Syntax

CSS Syntax



The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a CSS property name and a value, separated by a colon.

Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

3.2 CSS เบื้องต้น

3.2.4 เวอร์ชันของ CSS

เวอร์ชันของ CSS มีดังต่อไปนี้

- **CSS 1 CSS 1 หรือ CSS Level 1** คือเวอร์ชันแรกที่ถูกออกสู่สาธารณะและเป็น W3C Recommendation ซึ่งออกในเดือนธันวาคมปีค.ศ. 1996 CSS 1 นั้นมีฟีเจอร์เพียงไม่มาก
- **CSS 2 CSS 2** ออกในเดือนพฤษภาคมปีค.ศ. 1998 โดยเพิ่มเติมฟีเจอร์ให้กับ CSS 1 อย่างไรก็ตาม CSS 2 ไม่ประสบความสำเร็จมากนักในแง่ของการยอมรับและการรองรับจากเว็บเบราว์เซอร์ต่างๆ ทำให้ CSS 2.1 ต้องออกมาเพื่อแก้ไขปัญหาต่างๆในเวอร์ชัน 2 CSS 2.1 ได้เป็น Recommendation ในเดือนกรกฎาคมปีค.ศ. 2007
- **CSS 3 CSS 3** ยังอยู่ระหว่างการพัฒนา และการกำหนดสเปคใน CSS 3 นั้น จะถูกแบ่งออกเป็นส่วน (Module) ต่าง ๆ ซึ่งบางส่วนก็เสร็จสมบูรณ์แล้ว บางส่วนที่กำลังพัฒนาอยู่ แต่บางเว็บเบราว์เซอร์ก็เริ่มรองรับสเปคในบางส่วนบ้างแล้ว

<https://blog.sogoodweb.com/Article/Detail/79237/CSS-คืออะไร-มีประโยชน์-อย่างไร>

3.2 CSS เบื้องต้น

3.2.4 เวอร์ชันของ CSS

เวอร์ชันของ CSS มีดังต่อไปนี้

- **CSS 1 CSS 1 หรือ CSS Level 1** คือเวอร์ชันแรกที่ถูกออกสู่สาธารณะและเป็น W3C Recommendation ซึ่งออกในเดือนธันวาคมปีค.ศ. 1996 CSS 1 นั้นมีฟีเจอร์เพียงไม่มาก
- **CSS 2 CSS 2** ออกในเดือนพฤษภาคมปีค.ศ. 1998 โดยเพิ่มเติมฟีเจอร์ให้กับ CSS 1 อย่างไรก็ตาม CSS 2 ไม่ประสบความสำเร็จมากนักในแง่ของการยอมรับและการรองรับจากเว็บเบราว์เซอร์ต่างๆ ทำให้ CSS 2.1 ต้องออกมาเพื่อแก้ไขปัญหาต่างๆในเวอร์ชัน 2 CSS 2.1 ได้เป็น Recommendation ในเดือนกรกฎาคมปีค.ศ. 2007
- **CSS 3 CSS 3** ยังอยู่ระหว่างการพัฒนา และการกำหนดสเปคใน CSS 3 นั้น จะถูกแบ่งออกเป็นส่วน (Module) ต่าง ๆ ซึ่งบางส่วนก็เสร็จสมบูรณ์แล้ว บางส่วนที่กำลังพัฒนาอยู่ แต่บางเว็บเบราว์เซอร์ก็เริ่มรองรับสเปคในบางส่วนบ้างแล้ว

<https://blog.sogoodweb.com/Article/Detail/79237/CSS-คืออะไร-มีประโยชน์-อย่างไร>

เราสามารถเขียนโค้ด CSS ได้ 3 ที่หรือ 3 แบบหลัก ๆ ดังนี้

เราสามารถเขียนโค้ด CSS ได้ 3 ที่หรือ 3 แบบหลัก ๆ ดังนี้

- **Inline** แทรกในแท็ก HTML
- **Embedded** แทรกในส่วนหัวเอกสาร `<head></head>`
- **External** ภายนอกเอกสาร

รายละเอียดของแต่ละรูปแบบจะอธิบายในหัวข้อถัดไป

1. การเขียนโค้ด CSS แบบ Inline

แบบ Inline คือการเขียนโค้ด CSS ลงไปในแท็ก HTML ที่ต้องการให้เกิดการแสดงผลเลย และจะมีผลต่อแท็กนั้นๆเท่านั้น โดยมี Syntax ดังนี้

```
<tag style="property:value; property:value; . . .">
```

แอททริบิวต์ style ใช้ในการกำหนดค่าสไตล์หรือพเพอร์ตีต่างๆ ซึ่งจะอยู่ในรูปของ "property:value" หากมีมากกว่าหนึ่งชุดให้ใช้เครื่องหมาย Semicolon ";" คั่น ตัวอย่างเช่น

```
<h1 style="color:#0000FF">Where to Put CSS?</h1>?
```

ในตัวอย่างข้างต้นเรากำหนดสไตล์ให้กับแท็ก <h1> โดยให้ข้อความเป็นสีฟ้าด้วยสไตล์หรือพเพอร์ตี color การเขียนแบบ Inline ไม่เป็นที่นิยมนัก เนื่องจากต้องเขียนโค้ด CSS ลงไปในแท็ก HTML ดังนั้นจึงไม่สามารถนำสไตล์ไปประยุกต์ใช้กับแท็กอื่นๆได้ ลองนึกดูว่าถ้าต้องระบุค่าสีให้กับทุกๆแท็ก <h1> ในทุกๆเว็บเพจ คงจะลำบากน่าดูหากว่าต้องการเปลี่ยนสีในภายหลัง

```
<tag style="property:value; property:value; . . .">
```

3.2 CSS เบื้องต้น

3.2.5 รูปแบบการเขียน CSS

1. การเขียนโค้ด CSS แบบ Inline

แบบ Inline คือการเขียนโค้ด CSS ลงไปในแท็ก HTML ที่ต้องการให้เกิดการแสดงผลเลย และจะมีผลต่อแท็กนั้นๆเท่านั้น โดยมี Syntax ดังนี้

```
<tag style="property:value; property:value; . . .">
```

แอททริบิวต์ `style` ใช้ในการกำหนดค่าสไตล์หรือพเพอร์ตีต่างๆ ซึ่งจะอยู่ในรูปของ "property:value" หากมีมากกว่าหนึ่งชุดให้ใช้เครื่องหมาย Semicolon ";" คั่น ตัวอย่างเช่น

```
<h1 style="color:#0000FF">Where to Put CSS?</h1>?
```

ในตัวอย่างข้างต้นเรากำหนดสไตล์ให้กับแท็ก `<h1>` โดยให้ข้อความเป็นสีฟ้าด้วยสไตล์หรือพเพอร์ตี `color` การเขียนแบบ Inline ไม่เป็นที่นิยมนัก เนื่องจากต้องเขียนโค้ด CSS ลงไปในแท็ก HTML ดังนั้นจึงไม่สามารถนำสไตล์ไปประยุกต์ใช้กับแท็กอื่นๆได้ ลองนึกดูว่าถ้าต้องระบุค่าสีให้กับทุกๆแท็ก `<h1>` ในทุกๆเว็บเพจ คงจะลำบากน่าดูหากว่าต้องการเปลี่ยนสีในภายหลัง

```
<tag style="property:value; property:value; . . .">
```

ตัวอย่างการเขียนโค้ด CSS แบบ Inline

Test_css01.html

3.2.5 รูปแบบการเขียน CSS

1. การเขียนโค้ด CSS แบบ Inline

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1 style="color:blue; text-align:center;">This is a  
heading</h1>
```

```
<p style="color:red;">This is a paragraph.</p>
```

```
</body>
```

```
</html>
```

3.2.5 รูปแบบการเขียน CSS

1. การเขียนโค้ด CSS แบบ Inline

ตัวอย่างการเขียนโค้ด CSS แบบ Inline

Test_css01.html

```
1 <!DOCTYPE html>
2 <html>
3
4 <body>
5     <h1 style="color: blue; text-align:center;">This is a heading</h1>
6     <p style="color: red;">This is a paragraph.</p>
7 </body>
8
9 </html>
```

3.2 CSS เบื้องต้น

3.2.5 รูปแบบการเขียน CSS

2. การเขียนโค้ด CSS แบบ Embedded

แบบที่ 2 แบบ Embedded เป็นการเขียนโค้ด CSS ลงไปในไฟล์ HTML เช่นกัน แต่จะรวมโค้ด CSS ทั้งหมดไว้ภายใต้แท็ก `<head>` โดยมี Syntax ดังนี้

```
1 <head>
2   <title></title>
3   <style type="text/css">
4     selector {
5       property:value;
6       property:value;
7       . . .
8     }
9   </style>
10 </head>
```

3.2 CSS เบื้องต้น

3.2.5 รูปแบบการเขียน CSS

2. การเขียนโค้ด CSS แบบ Embedded

แบบที่ 2 แบบ Embedded เป็นการเขียนโค้ด CSS ลงไปในไฟล์ HTML เช่นกัน แต่จะรวมโค้ด CSS ทั้งหมดไว้ภายใต้แท็ก `<head>` โดยมี Syntax ดังนี้

```
1 <head>
2   <title></title>
3   <style type="text/css">
4     selector {
5       property:value;
6       property:value;
7       . . .
8     }
9   </style>
10 </head>
```

3.2 CSS เบื้องต้น

3.2.5 รูปแบบการเขียน CSS

3. การเขียนโค้ด CSS แบบ External CSS

การกำหนดสไตล์จะถูกกำหนดไว้ในไฟล์ .css แยกไว้ต่างหาก ถ้าต้องการเรียกใช้งาน ก็ให้ดึงมาโดยการใส่ Tag Link ดังตัวอย่าง

```
1 <head>
2   <title></title>
3   <style type="text/css">
4     h1 {
5       color: ■ #0000FF;
6       font-style: italic;
7     }
8     li {
9       list-style-type: upper-roman;
10    }
11  </style>
12 </head>
```



```
<!DOCTYPE html>
<html>
<head>
<style text/css>
body {
  background-color: linen;
}

h1 {
  color: maroon;
  margin-left: 40px;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

3.2.5 รูปแบบการเขียน CSS

2. การเขียนโค้ด CSS แบบ Embedded

Test_css02.html

เป็นการเขียน
CSS ไว้ในส่วน
<head>
</head>

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5     <style text/css>
6         body {
7             background-color: linen;
8         }
9
10        h1 {
11            color: maroon;
12            margin-left: 40px;
13        }
14    </style>
15 </head>
16
17 <body>
18     <h1>This is a heading</h1>
19     <p>This is a paragraph.</p>
20 </body>
21
22 </html>
```

This is a heading

This is a paragraph.

3. การเขียนโค้ด CSS แบบ External CSS

```
<!DOCTYPE html>  
<html>  
<head>  
<link rel="stylesheet" href="mystyle.css">  
</head>  
<body>  
  
<h1>This is a heading</h1>  
<p>This is a paragraph.</p>  
  
</body>  
</html>
```

03 แนะนำ CSS เบื้องต้น



3.2 CSS เบื้องต้น

3.2.5 รูปแบบการเขียน CSS

3. การเขียนโค้ด CSS แบบ External CSS

Test_css03.html

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <link rel="stylesheet" href="mystyle.css">
6 </head>
7
8 <body>
9   <h1>This is a heading</h1>
10  <p>This is a paragraph.</p>
11 </body>
12
13 </html>
```



3.2 CSS เบื้องต้น

3.2.5 รูปแบบการเขียน CSS

3. การเขียนโค้ด CSS แบบ External CSS

mystyle.css

```
<style text/css>
  body {
    background-color: linen;
  }

  h1 {
    color: maroon;
    margin-left: 40px;
  }
</style>
```

3.2 CSS เบื้องต้น

3.2.5 รูปแบบการเขียน CSS

3. การเขียนโค้ด CSS แบบ External CSS

mystyle.css

Multiple Style Sheets

- ```
h1 {
 color: navy;
}
```

- ```
h1 {  
  color: orange;  
}
```

ตัวอย่าง

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
h1 {
```

```
  color: orange;
```

```
}
```

```
</style>
```

```
<link rel="stylesheet" type="text/css" href="mystyle.css">
```

```
</head>
```

```
<body>
```

```
<h1>This is a heading</h1>
```

```
<p>The style of this document is a combination of an external stylesheet, and internal style</p>
```

```
</body>
```

```
</html>
```

3.2.5 รูปแบบการเขียน CSS

3. การเขียนโค้ด CSS แบบ External CSS

ตัวอย่าง

3.2.5 รูปแบบการเขียน CSS

3. การเขียนโค้ด CSS แบบ External CSS

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
<style>
h1 {
  color: orange;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>The style of this document is a combination of an external stylesheet, and internal
style</p>

</body>
</html>
```


3.2 CSS เบื้องต้น

CSS: Style

CSS Comments

เราสามารถเขียนโค้ด CSS ได้ 3 ที่หรือ 3 แบบหลัก ๆ ดังนี้

```
<!DOCTYPE html>
<html>
<head>
<style>
/* This is a single-line comment */
p {
  color: red;
}
</style>
</head>
<body>
  <p>Hello World!</p>
  <p>This paragraph is styled with CSS.</p>
  <p>CSS comments are not shown in the output.</p>
</body>
</html>
```

```
/* This is a single-line comment */
p {
  color: red;
}
```

3.2 CSS เบื้องต้น

CSS: Style

การเขียนโค้ด CSS แบบ Inline

```
body {  
  background-color: lightblue;  
}  
  
h1 {  
  color: white;  
  text-align: center;  
}  
  
p {  
  font-family: verdana;  
  font-size: 20px;  
}
```

<tag style="property:value; property:value; ...">

3.2 CSS เบื้องต้น

CSS: Style

CSS Text

CSS has a lot of properties for formatting text.

- Text Color
- Text Alignment
- Text Decoration
- Text Transformation
- Text Spacing
- Text Shadow

CSS Text

Text Color

Text Alignment

Text Decoration

Text Transformation

Text Spacing

Text Shadow

CSS has a lot of properties for formatting text.

The color property is used to set the color of the text. The color is specified by:

- a color name - like "red"
- a HEX value - like "#ff0000"
- an RGB value - like "rgb(255,0,0)"

```
body {  
  color: blue;  
}
```

```
h1 {  
  color: green;  
}
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h3 style="color:Tomato;">Hello World</h3>
```

```
<p style="color:DodgerBlue;">Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</p>
```

```
<p style="color:MediumSeaGreen;">Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.</p>
```

```
</body>
```

```
</html>
```

```
<h1 style="color:Tomato;">Hello World</h1>
```

```
<p style="color:DodgerBlue;">Lorem ipsum...</p>
```

```
<p style="color:MediumSeaGreen;">Ut wisi enim...</p>
```

03 แนะนำ CSS เบื้องต้น



3.2 CSS เบื้องต้น

CSS: Style

ตัวอย่าง CSS Text

CSS Text

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
body {
```

```
  color: blue;
```

```
}
```

```
h1 {
```

```
  color: green;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>This is heading 1</h1>
```

```
<p>This is an ordinary paragraph. Notice that this text is blue. The default text color for a page is defined in the body selector.</p>
```

```
<p>Another paragraph.</p>
```

```
</body>
```

```
</html>
```

This is heading 1

This is an ordinary paragraph. Notice that this text is blue. The default text color for a page is defined in the body selector.

Another paragraph.



3.2 CSS เบื้องต้น

CSS: Style

CSS Text Shadow

```
h1 {  
  text-shadow: 2px 2px;  
}
```

Text shadow effect!

```
h1 {  
  text-shadow: 2px 2px red;  
}
```

Text shadow effect!

3.2 CSS เบื้องต้น

CSS: Style

ตัวอย่าง CSS Text Shadow

CSS Text Shadow

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  color: white;
  text-shadow: 2px 2px 4px #000000;
}
</style>
</head>
<body>
<h1>Text-shadow effect!</h1>
</body>
</html>
```

Text-shadow effect!

3.2 CSS เบื้องต้น

CSS: Style

ตัวอย่าง CSS Text Shadow

CSS Text Shadow

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  text-shadow: 0 0 3px #FF0000, 0 0 5px #0000FF;
}
</style>
</head>
<body>

<h1>Text-shadow with red and blue neon glow!</h1>

</body>
</html>
```

Text-shadow with red and blue neon glow!

3.2 CSS เบื้องต้น

CSS: Style

Font-family

```
.p1 {  
  font-family: "Times New Roman", Times, serif;  
}  
  
.p2 {  
  font-family: Arial, Helvetica, sans-serif;  
}  
  
.p3 {  
  font-family: "Lucida Console", "Courier New", monospace;  
}
```

3.2 CSS เบื้องต้น

CSS: Style

Font-family

ตัวอย่าง Font-Family

```
<!DOCTYPE html>
<html>
<head>
<style>
.p1 {
  font-family: "Times New Roman", Times, serif;
}
.p2 {
  font-family: Arial, Helvetica, sans-serif;
}
.p3 {
  font-family: "Lucida Console", "Courier New", monospace;
}
</style>
</head>
<body>
  <h1>CSS font-family</h1>
  <p class="p1">This is a paragraph, shown in the Times New Roman font.</p>
  <p class="p2">This is a paragraph, shown in the Arial font.</p>
  <p class="p3">This is a paragraph, shown in the Lucida Console font.</p>
</body>
</html>
```

3.2 CSS เบื้องต้น

CSS: Style

Links

```
/* unvisited link */
a:link {
  color: red;
}

/* visited link */
a:visited {
  color: green;
}

/* mouse over link */
a:hover {
  color: hotpink;
}

/* selected link */
a:active {
  color: blue;
}
```

3.2 CSS เบื้องต้น

CSS: Style

1. เลือกองค์ประกอบ (Selectors) และกำหนดสไตล์ (Style)

```
css Copy code
/* การเลือกตามชื่อแท็ก HTML */
h1 {
  color: blue;
}

/* การเลือกตาม class */
.my-class {
  font-size: 16px;
}

/* การเลือกตาม ID */
#my-id {
  background-color: #eee;
}

/* การเลือกตามลำดับขององค์ประกอบ */
li:nth-child(even) {
  background-color: #f2f2f2;
}
```

3.2 CSS เบื้องต้น

CSS: Style

CSS: color

CSS: Font-family

```
/* การกำหนดสีของข้อความ */  
p {  
  color: #333;  
}  
  
/* การกำหนดสีพื้นหลัง */  
body {  
  background-color: #f0f0f0;  
}  
  
/* การกำหนดขนาดและประเภทของฟอนต์ */  
h2 {  
  font-size: 24px; font-family: 'Arial', sans-  
  serif;  
}
```

2. กำหนดสีและฟอนต์

```
css Copy code  
  
/* การกำหนดสีของข้อความ */  
p {  
  color: #333;  
}  
  
/* การกำหนดสีพื้นหลัง */  
body {  
  background-color: #f0f0f0;  
}  
  
/* การกำหนดขนาดและประเภทของฟอนต์ */  
h2 {  
  font-size: 24px;  
  font-family: 'Arial', sans-serif;  
}
```

CSS Background

The CSS background properties are used to add background effects for elements.

- Background Color
- Background image
- Background Repeat
- Background Attachment
- Background Shorthand

CSS Backgrounds

Background Color

Background Image

Background Repeat

Background Attachment

Background Shorthand

3.2 CSS เบื้องต้น

CSS Colors

CSS: Style

CSS Background Color

```
<!DOCTYPE html>
<html>
<body>
```

```
<h1 style="background-color: DodgerBlue;">Hello World</h1>
<p style="background-color: Tomato;">Lorem ipsum...</p>
```

```
<h1 style="background-color: DodgerBlue;">Hello World</h1>
```

```
<p style="background-color: Tomato;">
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Ut wisi enim ad minim veniam, quis nostrud exercitation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

```
</p>
```

```
</body>
```

```
</html>
```


3.2 CSS เบื้องต้น

CSS: Style

```
background-color: lightblue;
```

CSS Background Color

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: lightblue;
}
</style>
</head>
<body>

<h1>Hello World!</h1>

<p>This page has a light blue background color!</p>

</body>
</html>
```

3.2 CSS เบื้องต้น

CSS: Style

Other Elements

You can set the background color for any HTML elements:

CSS Background Color

```
h1 {  
  background-color: green;  
}  
  
div {  
  background-color: lightblue;  
}  
  
p {  
  background-color: yellow;  
}
```

3.2 CSS เบื้องต้น

CSS Colors

CSS: Style

CSS Color Names

```
<!DOCTYPE html>
<html>
<body>

<h1 style="background-color:Tomato;">Tomato</h1>
<h1 style="background-color:Orange;">Orange</h1>
<h1 style="background-color:DodgerBlue;">DodgerBlue</h1>
<h1 style="background-color:MediumSeaGreen;">MediumSeaGreen</h1>
<h1 style="background-color:Gray;">Gray</h1>
<h1 style="background-color:SlateBlue;">SlateBlue</h1>
<h1 style="background-color:Violet;">Violet</h1>
<h1 style="background-color:LightGray;">LightGray</h1>

</body>
</html>
```

```
/* This is a single-line comment */
p {
  color: red;
}
```

ตาราง (table)

```
table, th, td {  
  border: 1px solid;  
}
```

CSS : Table

The table above might seem small in some cases. If you need a table that should span the entire screen (full-width), add `width: 100%` to the `<table>` element:

Firstname	Lastname
Peter	Griffin
Lois	Griffin

Full-Width Table

```
table {  
  width: 100%;  
}
```

Collapse Table Borders

```
table {  
  border-collapse: collapse;  
}
```

Collapse Table Borders

The `border-collapse` property sets whether the table borders should be collapsed into a single border:

	Firstname	Lastname
Peter		Griffin
Lois		Griffin

CSS Table Size

CSS : Table

```
table {  
  width: 100%;  
}  
  
th {  
  height: 70px;  
}
```

Table Width and Height

The width and height of a table are defined by the `width` and `height` properties.

The example below sets the width of the table to 100%, and the height of the `<th>` elements to 70px:

Firstname	Lastname	Savings
Peter	Griffin	\$100
Lois	Griffin	\$150
Joe	Swanson	\$300

CSS Table Alignment

CSS : Table

```
td {  
  text-align:  
  center;  
}
```

Horizontal Alignment

The `text-align` property sets the horizontal alignment (like left, right, or center) of the content in `<th>` or `<td>`.

By default, the content of `<th>` elements are center-aligned and the content of `<td>` elements are left-aligned.

To center-align the content of `<td>` elements as well, use `text-align: center` :

Firstname	Lastname	Savings
Peter	Griffin	\$100
Lois	Griffin	\$150
Joe	Swanson	\$300

CSS Table Alignment

CSS : Table

```
th {  
  text-align: left;  
}
```

To left-align the content, force the alignment of <th> elements to be left-aligned, with the `text-align: left` property:

Firstname	Lastname	Savings
Peter	Griffin	\$100
Lois	Griffin	\$150
Joe	Swanson	\$300

Vertical Alignment

CSS : Table

```
td {  
  height: 50px;  
  vertical-align: bottom;  
}
```

Vertical Alignment

The `vertical-align` property sets the vertical alignment (like top, bottom, or middle) of the content in `<th>` or `<td>`.

By default, the vertical alignment of the content in a table is middle (for both `<th>` and `<td>` elements).

The following example sets the vertical text alignment to bottom for `<td>` elements:

Firstname	Lastname	Savings
Peter	Griffin	\$100
Lois	Griffin	\$150
Joe	Swanson	\$300

3.2 CSS เบื้องต้น

CSS: Style

- ใน HTML ตาราง (table) คือองค์ประกอบที่ใช้แสดงข้อมูลในลักษณะตาราง ตารางประกอบด้วยแถว (row) และคอลัมน์ (column) แต่ละแถวและคอลัมน์สามารถมีเซลล์ (cell) หนึ่งเซลล์หรือมากกว่า เซลล์เป็นพื้นที่ที่ใช้ในการแสดงข้อมูล
- แท็กพื้นฐานที่ใช้ในการสร้างตารางคือแท็ก `<table>` แท็กนี้ล้อมรอบตารางทั้งหมด แท็ก `<tr>` ใช้เพื่อสร้างแถวในตาราง แท็ก `<td>` ใช้เพื่อสร้างเซลล์ในตาราง
- ตัวอย่างโค้ด HTML ที่สร้างตาราง:

HTML

CSS : Table

```
<table>
  <tr>
    <td>ชื่อ</td>
    <td>อายุ</td>
  </tr>
  <tr>
    <td>John Doe</td>
    <td>30</td>
  </tr>
  <tr>
    <td>Jane Doe</td>
    <td>25</td>
  </tr>
</table>
```

The position Property

The position Property

The position property specifies the type of positioning method used for an element. There are five different position values:

- static
- relative
- fixed
- absolute
- sticky

The position Property

The position Property

```
position: static;
```

HTML elements are positioned static by default.

Static positioned elements are not affected by the top, bottom, left, and right properties.

An element with `position: static;` is not positioned in any special way; it is always positioned according to the normal flow of the page:

This `<div>` element has `position: static;`

```
div.static {  
  position: static;  
  border: 3px solid #73AD21;  
}
```

3.2 CSS เบื้องต้น

CSS: Style

The position Property

ตัวอย่าง

```
<!DOCTYPE html>
<html>
<head>
<style>
div.static {
position: static;
border: 3px solid #73AD21;
}
</style>
</head>
<body>
<h2>position: static;</h2>
<p>An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page:</p>
<div class="static">
This div element has position: static;
</div>
</body>
</html>
```

3.2 CSS เบื้องต้น

HTML : Table

```
<table>
<tr>
<td></td>
<td></td>
</tr>
</table>
```

แถว

3.2 CSS เบื้องต้น

HTML แท็ก <div>

- ใน HTML แท็ก <div> คือแท็กทั่วไปที่ใช้เพื่อกำหนดพื้นที่หรือส่วนประกอบในหน้าเว็บ แท็ก <div> ไม่ได้กำหนดการแสดงผลของเนื้อหาภายในแท็ก แต่จะทำหน้าที่เป็นกล่องหรือคอนเทนเนอร์สำหรับเนื้อหาเหล่านั้น
- แท็ก <div> มักใช้ร่วมกับ CSS เพื่อกำหนดสไตล์ของเนื้อหาภายในแท็ก ตัวอย่างเช่น เราสามารถเปลี่ยนสีพื้นหลังของแท็ก <div> เปลี่ยนสีตัวอักษร เปลี่ยนขนาดตัวอักษร หรือเปลี่ยนการจัดแนวของตัวอักษร
- ตัวอย่างโค้ด HTML ที่ใช้แท็ก <div> และ CSS เพื่อกำหนดสไตล์ของเนื้อหาภายในแท็ก:

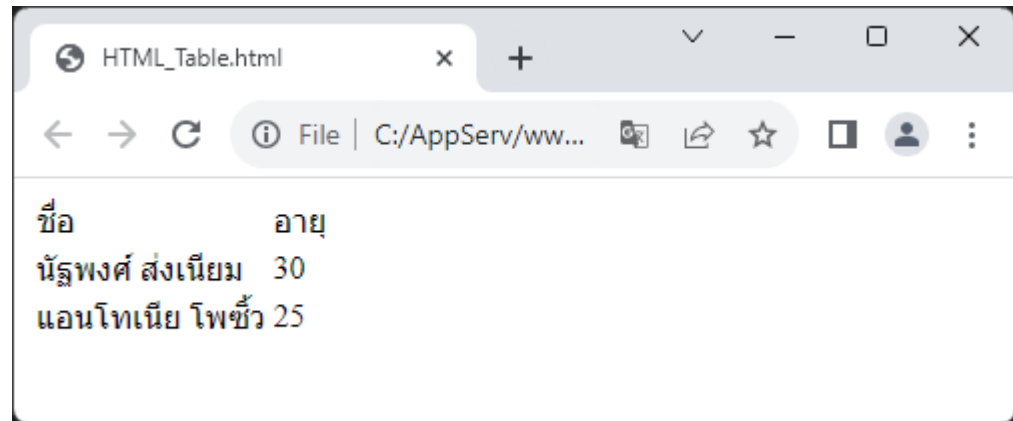
```
<div style="background-color: #ffffff; color: #000000; font-size: 16px; text-align: center;"> ข้อความภายในแท็ก <div>  
</div>
```


3.2 CSS เบื้องต้น

HTML : Table

ตัวอย่าง HTML : Table

```
<table>
  <tr>
    <td>ชื่อ</td>
    <td>อายุ</td>
  </tr>
  <tr>
    <td>นัฐพงศ์ ส่งเนียม</td>
    <td>30</td>
  </tr>
  <tr>
    <td>แอนโทเนีย โพธิ์</td>
    <td>25</td>
  </tr>
</table>
```



The screenshot shows a web browser window with the title "HTML_Table.html". The address bar shows the file path "C:/AppServ/ww...". The browser content displays a table with two columns: "ชื่อ" (Name) and "อายุ" (Age). The table contains three rows of data:

ชื่อ	อายุ
นัฐพงศ์ ส่งเนียม	30
แอนโทเนีย โพธิ์	25

แบบฝึกหัด : จงสร้างตารางที่มี 3 แถว 2 คอลัมน์

รายการ	ราคา
คอมพิวเตอร์	30000
เครื่องพิมพ์	2000

↑
2px , solid

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <style text/css>
5   body {
6     background-color: # linen;
7   }
8
9   h1 {
10    color: # maroon;
11    margin-left: 40px;
12  }
13
14  table,
15  th,
16  td {
17    border: 1px solid;
18    width: 60%;
19  }
20
21  th {
22    height: 70px;
23  }
24 </style>
25 </head>
```

```
27 <body>
28   <h1>This is a heading</h1>
29   <p>This is a paragraph.</p>
30   <table border="2px" align="center">
31     <tr>
32       <td>สินค้า</td>
33       <td>ราคา</td>
34     </tr>
35     <tr>
36       <td>คอมพิวเตอร์ </td>
37       <td>40000</td>
38     </tr>
39     <tr>
40       <td>เครื่องพิมพ์</td>
41       <td>2800</td>
42     </tr>
43   </table>
44 </body>
45
46 </html>
```

This is a heading

This is a paragraph.

สินค้า	ราคา
คอมพิวเตอร์	40000
เครื่องพิมพ์	2800

HTML : Table

จัดข้อความกึ่งกลางเซลล์

```
td {  
  text-align: center;  
}
```

```
th {  
  text-align: left;  
}
```

กำหนดความสูงและตำแหน่งแนวตั้ง

```
td {  
  height: 50px;  
  vertical-align: bottom;  
}
```

```
td {  
  height: 50px;  
  vertical-align: middle;  
}
```

สินค้า	ราคา
คอมพิวเตอร์	40000
เครื่องพิมพ์	2800

Table

```
<style =“text/css”>
table, td, th {
  border: 1px solid #ddd;
  text-align: left;
}

table {
  border-collapse: collapse;
  width: 100%;
}

th, td {
  padding: 15px;
}
</style>
```

HTML : Form

ใน HTML ฟอर्म (form) คือองค์ประกอบที่ใช้เพื่อรวบรวมข้อมูลจากผู้ใช้ ฟอर्मประกอบด้วยแท็ก input ที่ระบุประเภทของข้อมูลที่ผู้ใช้สามารถป้อนได้ เช่น ข้อความ ตัวเลข อีเมล หรือที่อยู่

แท็ก form มีองค์ประกอบที่สำคัญดังนี้

- ชื่อ (name): ชื่อของฟอर्म ใช้เพื่อระบุฟอर्मในโค้ด JavaScript หรือ PHP
- การกระทำ (action): URL ของหน้าเว็บที่ข้อมูลจะถูกส่งไป
- วิธีการ (method): วิธีที่ใช้ส่งข้อมูลไปยังหน้าเว็บที่ระบุโดย action อาจเป็น POST หรือ GET

`<form>`

.

form elements

.

`</form>`

Form : input

- `input {`
 `width: 100%;`
 `}`

HTML formFirst name: Last name: E-mail: Male Female

Form : method

```
<form method = "post" active = "" >
```

```
•  
<input>
```

```
<..>
```

```
•  
</form>
```

Form : method

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname"
value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

ตัวอย่าง

```
<!DOCTYPE html>
<html>
<body>

<h2>HTML Forms</h2>

<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>

<p>If you click the "Submit" button, the form-data will be sent to a page called
"/action_page.php".</p>

</body>
</html>
```

Form : Action

```
<form action="/action_page.php"  
target="_blank">
```

HTML formFirst name: Last name: E-mail:

- Male
 Female

```
<tr>
  <td>คอมพิวเตอร์ </td>
  <td>40000</td>
</tr>
<tr>
  <td>เครื่องพิมพ์</td>
  <td>2800</td>
</tr>
</table>
<form> <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname">
</form>
```

```
</body>
```

```
</html>
```

First name:

Last name:

FORM : Input

```
<button id="btnLogin"  
name="login">Login</button>
```

- ```
<button id="btnLogin" name="login"
type="submit">Login</button>
```

## FORM : Input

```
</code>
<form> <label for="fname">First name:</label>

 <input type="text" id="fname" name="fname">

 <label for="lname">Last name:</label>

 <input type="text" id="lname" name="lname">
 <button id="btnLogin" name="login" type="submit">Login</button>
 <button id="btnLogin" name="login" type="reset">Reset</button>
</form>
```

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <style>
5 input {
6 width: 100%;
7 }
8 </style>
9 </head>
10 <body>
11
12 <h2>A full-width input field</h2>
13
14 <form>
15 <label for="fname">First Name</label>
16 <input type="text" id="fname" name="fn
17 </form>
18
19 </body>
20 </html>
21
```

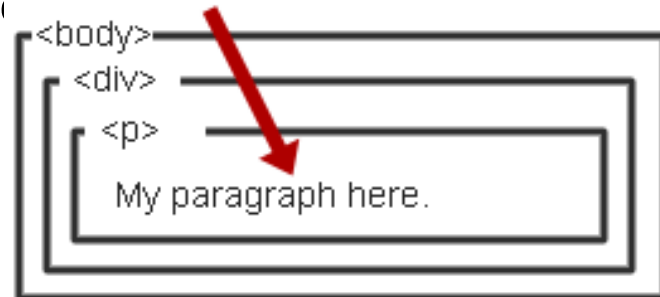


# Intro to CSS

- Covered in this lesson:
  - Overview
    - What is CSS?
    - Why to use CSS?
    - CSS for Skinning your Website
  - Structure
    - CSS Syntax Introduction
    - Three places CSS can be defined
    - CSS Syntax Specifics
    - Cascading Inheritance
  - Applied
    - CSS Hands-on

# What is CSS?

- Cascading Style Sheet
  - Stylesheet Language
    - Standards-based set of properties and attributes to define styles
  - To describe the presentation a document written in a 'markup language' like HTML or XML
    - Markup encoding: `<p>My paragraph here.</p>`
    - Defines the style of how things in `<p>` tags appear.
    - Font, color, size, margins, et
  - Cascading
    - Rules to determine how to
    - apply markup that contains
    - other markup



# Why CSS?

- Separate **Content** from **Form**
  - Content is the text and images, marked up to define regions of specific types
  - Form defines the “style” for the content

The old way:

```

My First Header

My information 1 goes here.

My Second Header

Different information goes here.

```

# Why CSS? Continued.

- Separate Content from Form

- Content

```
<p class="header">My First Header</p>
<p class="info">My Information 1 goes here</p>
<p class="header">My Second Header</p>
<p class="info">Different Information goes here</p>
```

(Specific markup properties like Class will be discussed later).

- Form or Style

```
.header { font-size:14px;}
.info { font-family: verdana;
 font-color: blue;
 font-size: 12px; }
```

# What does this separation get us?

- Separate **Content** from **Form**
  - Specify the style **once** for every instance of that class.
    - Example: Specify the font once for all text on the HTML page that you've identified as a "header".
  - The stylesheet can be a **separate file** which all HTML pages on your entire site can link to.
    - Only have to specify the style once for your **ENTIRE SITE**
  - Can change the style for your entire site by editing only **ONE FILE**.

# CSS Skinning

- “Skinning” - changing the look of a page or your site
  - Selecting an appearance by choosing which stylesheet to use.

```
<link rel="stylesheet"
type="text/css" href="skin1.css" />
```

+

```
<p class="info">My Information 1
goes here</p>
```

```
background-color: White;
font-family: Verdana;
```

```
font-color: Blue; }
```

Some information goes here.

# CSS Skinning 2

- “Skinning” - changing the look of a page or your site
  - Selecting an appearance by choosing which stylesheet to use.

```
<link rel="stylesheet" type="text/css" href="skin2.css" />
```

```
<p class="info">My Information 1 goes here</p>
```

+

```
skin1.css
```

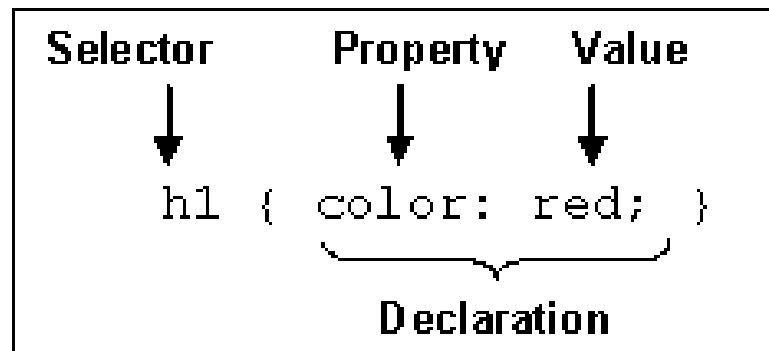
```
.info { background-color: Blue;
 font-family: Serif;
 font-color: White; }
```

=

Some information goes here.

# CSS Syntax

- 3 Elements to a CSS Statement
  - Selector
    - What HTML sections does it affect?
  - Property
    - What attribute of that HTML section will be affected?
  - Value
    - What change will be made to that attribute?





# Three CSS Definition Locations

- Inline: the “style” attribute

```
<p style="font-color:red;font-size:10px;">Content</p>
```

Note, the selector for inline CSS is the tag which contains the style attribute.

- Internal: the <style> markup tag

```
<html><head><style>
p { background-color: Red;
 font-family: serif;
 font-color: White; }
</style></head><body>
<p>Content</p>
</body></html>
```

- External: the .css stylesheet file

```
<link rel="stylesheet" type="text/css" href="mystylesheet.css" />
```

# CSS Syntax: Selectors

- There are many kinds of selectors and many ways to reference them:
  - Type, Class, ID, Pseudo, etc.
- HTML Type Tag – selected with the tag type

```
p { font-size: 10px;
 font-color: White; }
```

```
<p>Content</p>
```

- ~~The Class Attribute – precede the class with a period~~

```
.myinfo { font-size: 10px;
 font-color: White; }
```

```
<p class="myinfo">Content</p>
```

```
<div class="myinfo">Other content</div>
```

# Cascading Inheritance

- Nested elements inherit the properties from the its parent



- If you specify a style for the `<body>` tag it will affect all content in your HTML page.

```
body { font-family: Verdana;
 font-size: 14px; }
```

- If you want to override inherited settings, you need to specify a style in a more local element

```
body { font-family: Verdana;
 font-size: 1.1em; }
.littletext { font-size: 8px; }
```

```
<body>
This text is larger.
<p class="littletext">This text is
smaller.</p>
```