

## Cascading Style Sheets

### ความเป็นมา

ปัจจุบันสไตลชีต (CSS) เป็นสิ่งที่มีประโยชน์อย่างมากในการสร้างเว็บ โดยเฉพาะการควบคุมรูปแบบของหน้าเว็บ ให้มีลักษณะเดียวกันตลอดทั้งเว็บ อีกทั้งยังช่วยเพิ่มความสะดวกในการพัฒนาและปรับปรุงแก้ไขอีกด้วย จะเห็นได้ว่าการนำ Style Sheets มาใช้กันมากขึ้นและเริ่มแพร่หลาย

การนำ Style Sheet เป็นเพราะว่าเป็นการแก้ไขปัญหาไฟล์ HTML ที่เต็มไปด้วยคำสั่งในการตกแต่งรูปแบบเนื้อหา ทำให้ทาง World Wide Web Consortium (W3C) ก็ได้เป็นผู้กำหนดรายละเอียด CSS1 (Cascading Style Sheets, Level 1) เมื่อปี 1996 และ CSS2 (Cascading Style Sheets, Level 2) ในปี 1998 โดยสนับสนุนให้ใช้อย่างเต็มที่ (full recommendation) และให้ความสำคัญเทียบเท่ากับการใช้ภาษา HTML เลยทีเดียว

### คุณสมบัติของ Cascading Style Sheets

#### 1. หลากหลายรูปแบบ (Rich Styling)

Style Sheets ช่วยให้ผู้ออกแบบสามารถสร้างรูปแบบของเนื้อหาที่ต้องการได้อย่างเต็มที่ เช่นการกำหนดสีและสีพื้นขององค์ประกอบ สร้างเส้นกรอบรอบองค์ประกอบ รวมถึงการเพิ่มหรือลดพื้นที่โดยรอบของสิ่งต่างๆ

```
<H1>rich style content</H1>
```

นี่เป็นตัวอย่างถ้าเราต้องการให้หัวข้อนี้มีลักษณะ เอียง สีน้ำเงิน ใช้ตัวอักษรตามที่กำหนดขีดเส้นใต้ และมีพื้นหลังเป็นสีเหลือง ด้วยภาษา HTML คุณต้องนำข้อความทั้งหมดไปใส่ไว้ในตาราง พร้อมกำหนดรายละเอียดต่างไว้ระหว่างแท็ก <H1> กับ </H1> ด้วยคำสั่ง FONT และ U (underline) สำหรับ CSS คุณสามารถกำหนดรายละเอียดของแท็ก H1 ให้เป็นดังนี้

```
H1 {  
    color: blue;  
    font: italic 1em Times, serif;  
    text-decoration: underline;  
    background: yellow;  
}
```

ผลที่ได้ออกมาคือ

*rich style content*

ซึ่งจะเห็นได้ว่าทำให้เกิดรูปแบบตามที่เรต้องการ ยิ่งไปกว่านั้น เรายังสามารถสร้างลักษณะพิเศษเพิ่มเติมซึ่งไม่สามารถทำได้ในภาษา HTML เช่น เส้นขอบสีส้มขนาด 1 พิกเซล ใส่รูปเป็นพื้นหลังที่จะแสดงผลซ้ำเฉพาะในแนวนอน พร้อมกับสร้างพื้นที่ว่างโดยรอบข้อความอีก 5 พิกเซล และลดบริเวณ margin ทางด้านล่างลง ด้วยข้อกำหนดต่อไปนี้

```
H1 {  
    color: blue;
```

```
font: italic 1em Times, serif;
text-decoration: underline;
background: yellow url(titlebg.gif) repeat-x;
border: 1px solid orange;
margin-bottom: 0;
padding: 5px;
}
```

ผลที่ได้ออกมาคือ



จากตัวอย่างจะเห็นได้ที่เราสามารถตกแต่งตัวอักษรในลักษณะต่างๆ บนเว็บไซต์ของเราให้สวยได้ตามที่เราต้องการ และอิสระในการตกแต่ง

## 2. ใช้งานสะดวก (Ease of Use)

เป็นลักษณะสำคัญของ Style Sheets เป็นการเพิ่มความสะดวกในการพัฒนาเว็บได้อย่างมากก็คือ เราสามารถรวบรวมข้อกำหนดของรูปแบบต่างๆไว้ในที่เดียวกัน แทนที่จะกระจายอยู่ทั่วทั้งไฟล์ ตัวอย่างเช่น ถ้าเราต้องการให้หัวข้อย่อยทั้งหมดเป็นสีเขียว ในภาษา HTML ต้องใช้แท็กแท็ก FONT เข้าไปในทุกส่วนของหัวข้อดังนี้

```
<H2><FONT COLOR="green">Subheading in green</H2>
```

จะเห็นได้ว่ามีจำนวนหัวข้อย่อยเหล่านี้มากๆ ก็นับว่าเป็นงานที่ต้องใช้เวลาพอสมควร และถ้าวันใดวันหนึ่งคุณจำเป็นต้องเปลี่ยนจากสีเขียวเป็นสีอื่น คุณก็ต้องกลับไปแก้ไข Tag เหล่านี้ทั้งหมดในทุกๆไฟล์แน่นอนว่าเราสามารถใช่วิธีการ find and replace ได้ครบในที่ที่ไม่มีคำว่า "green" อยู่ในส่วนอื่นของไฟล์ ลองเปรียบเทียบกับเมื่อเรามีข้อกำหนดอยู่เพียงเท่านั้น

```
H2 {color: green;}
```

ซึ่งไม่เพียงแต่จะง่ายในการพิมพ์แล้ว ยังง่ายในการเปลี่ยนแปลงแก้ไขอีกด้วย สิ่งที่ต้องทำคือเปลี่ยนคำว่า green เป็นสีอื่นเพียงแห่งเดียวเท่านั้นสามารถเปลี่ยนแท็ก H2 ได้ทั้งเว็บไซต์ที่มีการใช้แท็กนี้

## 3. ใช้สไตล์ได้กับหลายๆหน้า (Multiple Uses)

สามารถสร้าง Style Sheets ใช้กับเว็บหลายๆหน้าได้อย่างไม่จำกัด ด้วยการเซฟไฟล์ที่กำหนดรูปแบบขององค์ประกอบต่างๆไว้ในไฟล์ที่มีนามสกุลเป็น .css แล้วจึง import ตัว Style Sheets เหล่านี้เข้าไปในทุกๆไฟล์ HTML ช่วยให้คุณสร้างเว็บไซต์ที่มีรูปแบบเหมือนกันตลอดทั้งเว็บได้อย่างง่ายดาย และเมื่อจำเป็นต้องเปลี่ยนแปลงลักษณะใดๆ ก็เพียงทำการแก้ไขที่ไฟล์นี้เพียงไฟล์เดียวเท่านั้น การแสดงผลของเว็บไซต์ทั้งหมดก็จะเปลี่ยนไปตามข้อกำหนดใหม่โดยอัตโนมัติ

## 4. คุณสมบัติในการลำดับความสำคัญ (Cascading)

Style Sheets ยอมให้มีการใช้สไตล์หลายๆแบบในไฟล์เดียวกัน ในกรณีที่เกิดความซ้ำซ้อนกัน ก็จะยึดตามลำดับความสำคัญของสไตล์เหล่านั้น เช่น สไตล์ที่กำหนดไว้ภายในเอกสารนั้น จะมีความสำคัญเหนือกว่า สไตล์ที่กำหนดไว้ในไฟล์อื่น คุณสมบัตินี้จะช่วยให้เราสามารถเปลี่ยนแปลงรูปแบบเฉพาะจุดในแต่ละไฟล์ตามต้องการได้ นอกจากนั้นในบราวเซอร์รุ่นใหม่ๆยังมีออบชั่นให้ผู้ใช้มีสิทธิสร้างและใช้สไตล์ของตัวเอง ในการแสดงผลของเว็บต่างๆได้อีกด้วย

## ขนาดไฟล์ที่เล็กลง (Compact File Size)

เป็นสิ่งที่ทำให้ผู้ชมพอใจมากนั่นก็คือ การแสดงผลที่รวดเร็ว อันเนื่องมาจากขนาดของไฟล์ที่ลดลง เพราะเราสามารถลดปริมาณการใช้แท็ก FONT และ table จำนวนมากออกไปได้ มีผลในการสร้างความประทับใจให้กับผู้ใช้บริการได้มากยิ่งขึ้น

## [การนำ Style Sheets มาใช้](#)

มี 3 วิธีในการนำ Style Sheets มาใช้ในการพัฒนาเว็บไซต์ โดยเราสามารถเลือกใช้วิธีใดวิธีหนึ่งหรือหลายวิธีในหน้าเดียวกันก็ได้ มีดังต่อไปนี้

### 1. ลิงค์ Style Sheets ที่อยู่ภายนอกด้วยคำสั่ง Link

วิธีนี้เราจะกำหนดสไตล์ไว้ไปอีกไฟล์หนึ่งที่มีนามสกุลเป็น .css จากนั้นจึงลิงค์เข้ามาสู่ไฟล์ html ด้วยคำสั่ง link ดังนั้นทุกไฟล์ html ที่มีการลิงค์กับไฟล์สไตล์นี้ก็จะแสดงผลของสไตล์ออกมาได้เหมือนกัน วิธีนี้นับเป็นวิธีที่น่าสนใจนำไปใช้มากที่สุด เนื่องจากเราสามารถกำหนดและแก้ไขได้ในไฟล์สไตล์เพียงไฟล์เดียวเท่านั้น ช่วยเพิ่มความสะอาดและประหยัดเวลาได้อย่างมาก

คำสั่ง link ที่จะใช้ลิงค์ถึงไฟล์สไตล์นั้นจะต้องอยู่ภายในแท็ก head ของ html เท่านั้น มิฉะนั้นจะใช้งานไม่ได้ ภายในไฟล์สไตล์นั้นจะประกอบด้วยข้อกำหนดของสไตล์เท่านั้น ไม่ต้องมีแท็กอื่นๆทั้งสิ้น

ค่าอื่นๆภายในแท็ก link ที่ควรรู้จักได้แก่ rel ที่หมายถึง relation ซึ่งในที่นี้ก็คือ style sheets ส่วนค่า type นั้นก็จำเป็นต้องกำหนดเป็น text/css ซึ่งจะบอกให้เบราว์เซอร์รู้ว่า style sheets ที่ใช้ก็คือ CSS style sheet เนื่องจากในอนาคตอาจมีสไตล์ประเภทอื่นเกิดขึ้นก็ได้ ต่อมาก็เป็นค่า href ซึ่งมีค่าเป็น url หรือที่อยู่ของไฟล์นั่นเอง ซึ่งอาจเป็นได้ทั้งแบบ absolute และ relative เช่น อาจจะใช้เป็น href= "http://www.mark myweb.com/style/mmw.css" ก็ได้

#### ตัวอย่างการใช้งาน

```
<html>
<head>
<title>CSS</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-874">
<link rel="stylesheet" type="text/css" href="style1.css" > </head> <body>
.....
</body>
</html>
```

### 2. กำหนดสไตล์ไว้ในส่วนของ Style element

เราสามารถกำหนดสไตล์ไว้ในไฟล์ html ได้เช่นกัน ด้วยการกำหนดสไตล์ไว้ในส่วนของ <style> ซึ่งมักจะเริ่มต้นด้วย <style type="text/css"> และปิดท้ายด้วย </style> เสมอ รูปแบบนี้อาจเรียกว่าเป็น document style sheet หรือ embedded style sheet ซึ่งเป็นสไตล์ที่จะนำไปใช้กับเนื้อหาทุกส่วนที่อยู่ภายในไฟล์นั้น

ข้อกำหนดสไตล์ที่อยู่ในส่วนของ <style> นั้นจะเป็นข้อกำหนดทั่วไป ดังในตัวอย่าง กำหนดให้ บริเวณขอบทุกด้านเป็น 0 ด้วยค่า margin และกำหนดลักษณะและสีของข้อความที่เป็น ลิงค์ สังกัดเครื่องหมาย <!-- และ --> ถูกใส่เข้ามาเพื่อป้องกันไม่ไห้บรรทัดของโค้ดที่ปรากฏใน <style> (ซึ่งเป็นแท็กที่ค่อนข้างใหม่) แสดงรายละเอียดออกมา

```
<style type="text/css">
<!--
body { background: lightsteelblue;}
a:link { color: #005CA2; text-decoration: none}
a:visited { color: #005CA2; text-decoration: none}
a:active { color: #0099FF; text-decoration: underline}
a:hover { color: #0099FF; text-decoration: underline}
-->
</style>
```

นอกจากกำหนดใน HTML แล้วเรายังสามารถ ลิงค์ไปยังไฟล์ style sheet ด้วยคำสั่ง @import

ในส่วนของ <style> นั้นก่อนที่จะถึงข้อกำหนดสไตล์ เรายังสามารถใช้คำสั่ง @import เพื่อลิงค์ไปยังไฟล์ภายนอกได้อีกด้วย ทำนองเดียวกับการใช้ link ที่ผ่านมา จะต่างกันก็ตรงที่ลำดับของการลิงค์ไปยังไฟล์ ซึ่งจะมีผลต่อระดับความสำคัญของสไตล์ เมื่อเกิดความซ้ำซ้อนกัน ข้อควรระวังในการใช้คำสั่งนี้ก็คือ คุณจำเป็นต้องใช้คำสั่ง @import ไว้ก่อนที่จะถึงข้อกำหนดสไตล์ที่อยู่ในบริเวณนี้

#### ตัวอย่างการใช้งาน

```
<style type="text/css">
<!-- @import url(style2.css);
body { margin: 0px 0px; padding: 0px 0px; background: lightsteelblue;}
a:link { color: #005CA2; text-decoration: none}
a:visited { color: #005CA2; text-decoration: none}
a:active { color: #0099FF; text-decoration: underline}
a:hover { color: #0099FF; text-decoration: underline}
-->
</style>
```

### 3. กำหนดสไตล์ในแต่ละแท็ก (Inline style)

ในบางครั้ง เราอาจต้องการกำหนดสไตล์เฉพาะในแต่ละแท็กโดยตรง โดยไม่ใช้การลิงค์ หรือกำหนดไว้ที่อื่นด้วยการเพิ่มส่วนของ "ข้อกำหนดของสไตล์" ลงในแท็กที่ต้องการ ดังตัวอย่าง

```
<p style="color: gray;">my </p>
```

จากตัวอย่างจะทำให้เนื้อหาภายในแท็ก <p> เปลี่ยนเป็นสีเทา วิธีนี้ถือว่าง่ายและรวดเร็ว เมื่อต้องการใช้สไตล์กับบางบริเวณในจำนวนไม่มาก แต่ถ้าต้องการใช้สไตล์เดียวกันกับหลายๆ บริเวณก็ควรจะใช้วิธีอื่นข้างต้น

สรุปว่าการนำ Style Sheet เข้ามาใช้กับเว็บไซต์ จะช่วยเพิ่มประสิทธิภาพในการพัฒนาเว็บไซต์ให้ดียิ่งขึ้น ทั้งในแง่ของรูปแบบหรือสไตล์ในการแสดงผลขององค์ประกอบต่างๆ และความสะดวกในการจัดการ แก๊ไข รูปแบบเหล่านั้น ส่วนคุณ sẽเลือกใช้ Style Sheet ด้วยวิธีการใด ก็ลองพิจารณาตามความเหมาะสมครับ

## ทดลองสร้าง Style Sheet ด้วยคุณเองกันเถอะ

ในการออกแบบเว็บไซต์ที่มีการกำหนด Layout ไว้ตายตัว บางครั้งขนาดของตัวอักษรอาจ จะสร้างปัญหาได้ เนื่องจากผู้ใช้สามารถเลือกขนาดตัวอักษรได้เองจากคำสั่ง View > Text Size ซึ่งจะมีอุปชั่นให้เลือกตั้งแต่ Largest, Larger, Medium, Smaller ไปจนถึง Smallest ได้ก็สามารถทำได้โดยอาศัยความสามารถของ style sheet ในการแก้ไขเปลี่ยนแปลงลักษณะของตัวอักษร ไม่ว่าจะ เป็นชนิดตัวอักษร ขนาด สี และคุณสมบัติอื่นๆ ได้อย่างสะดวกด้วยการแก้ไขเพียงที่ Style Sheet เพียงแห่งเดียว แล้วทุกๆ หน้าในเว็บไซต์ที่ลิงค์มายังไฟล์นี้ ก็จะใช้ลักษณะของสไตล์ใหม่ได้ทันที

### **ความคล้ายกันในการเขียนระหว่าง HTML กับ Style sheet**

การเขียนแบบ HTML tag = **<tag property="value">**

การเขียนแบบคำสั่ง CSS tag = **element {property: value;}** เราจะเห็นได้ว่ามีความแตกต่างกันน้อยมาก แตกต่างกันตรงที่ <> กับ {} ตรงนี้เอง ส่วน element นั้นก็เป็นสิ่งที่เราต้องการ กำหนด โดยจะเป็นการเลือกใช้ Id หรือ Class แล้วแต่จุดประสงค์ของการใช้งาน

#### **การใช้ Class**

ต้องประกาศกำหนดคุณสมบัติต่างๆ ให้กับคำสั่ง HTML แบบเฉพาะเจาะจงให้กับคำสั่งนั้นๆ โดยทำการประกาศชื่อ Class ในรูปแบบ Style Sheet นั้นๆ ไว้ก่อน แล้วจึงทำการเรียกใช้งานผ่านคำสั่ง HTML โดยการระบุชื่อ Class ที่ต้องการ เช่นการใช้ในการกำหนด <H1 class = "A"> ซึ่งอยู่ใน Class A ให้มีสีแดงก็ไม่ได้ทำให้ <H1> Tag ธรรมดาที่มีสีแดงไปด้วย

สำหรับการประกาศ Class จะประกาศภายใต้คำสั่ง <style>...</style> โดยจะใช้เครื่องหมาย # แล้วตามด้วยชื่อ Class นั้นๆ

#### **ตัวอย่างการใช้**

```
<style type="text/css">
```

```
<!--
```

```
.Class1 {font-size:12px; font-weight:bold; color:#008040;}
```

```
.Class2 {font-size:10px; font-size:x-small; color:#00f;}
```

```
-->
```

```
</style>
```

#### **เรียกใช้โดยเรียกในไฟล์ HTML**

```
<p Class ="Class1">ทดสอบการใช้ ID</p>
```

```
<p Class ="Class2">ทดสอบการใช้ ID</p>
```

#### **การใช้ ID**

ต้องประกาศกำหนดคุณสมบัติต่างๆ ให้กับคำสั่ง HTML แบบเฉพาะเจาะจงให้กับคำสั่งนั้นๆ แต่จะใช้ ID แทน Class โดยจะทำการประกาศชื่อ Id ไว้ก่อนแล้วจึงเรียกใช้งานผ่านทางคำสั่ง HTML โดยการระบุชื่อ ID ที่ต้องการ โดยการแสดงผลเหมือนกับการใช้ Class ในหน้าหนึ่งไม่ควร มี Tag ที่ระบุ Id เดียวกันมากกว่า 1 การใช้ ID มีประโยชน์มากเมื่อมีการใช้ร่วมกับ JavaScript ที่มีการ กำหนดให้มีการเคลื่อนย้ายตำแหน่งหรือซ่อนวัตถุ

สำหรับการประกาศ ID จะประกาศภายใต้คำสั่ง <style>...</style> โดยจะใช้เครื่องหมาย # แล้วตามด้วยชื่อ Id นั้นๆ

#### **ตัวอย่างการใช้**

```
<style type="text/css">
```

```
<!--
```

```
#ID1{font-size:12px; font-weight:bold; color:#008040;}
```

```

        #iD2{font-size:10px; font-size:x-small; color:#00f;}
-->
</style>
เรียกใช้โดยเรียกในไฟล์ HTML
<p id="ID1">ทดสอบการใช้ ID</p>
<p id="ID2">ทดสอบการใช้ ID</p>

```

### ทดลองเขียนกันได้แล้ว

#### การตกแต่งตัวอักษร

เริ่มด้วยการกำหนดคุณสมบัติตัวอักษรเริ่มด้วยการสร้างเอกสารที่เป็น HTML กันก่อนสักไฟล์หนึ่ง Save เป็นไฟล์เอกสารชื่อ Test.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

```

<html>
<head>
<title>Untitled</title>
</head>
<body>
<style type="text/css">
<!--
        p {font-style: italic;
        font-weight: bold;
        font-size: 14pt;
        font-family: arial, sans-serif;}
-->
</style>
</body>
</html>

```

จากตัวอย่างด้านบนนี้ นะครับ จะเห็นได้ว่า มีการกำหนด ให้ตัวอักษรที่อยู่ภายใต้ <p> tag จะมีคุณสมบัติดังนี้

**font-style: italic;** เป็นอักษรตัวเอียง  
**font-weight: bold;** เป็นอักษรตัวหนา  
**font-size: 14pt;** เป็นอักษรขนาด 14 px  
**font-family: arial, sans-serif;** เป็นอักษรใน Font ตระกูล arial, sans-serif;

(ปกติเราจะกำหนดไว้ 3 font โดยเผื่อไว้ในกรณีที่ไม่มี font นั้นในเครื่องก็จะทำการแสดงในรูปแบบของ font ถัดไป)

#### การตกแต่ง Button

สำหรับ Button นี้ก็คือ ปุ่มกด ต่างๆที่อยู่ในแบบฟอร์มของเรา ทั้ง ปุ่มกดธรรมดา (Button), ปุ่มส่งข้อมูล (Submit), ปุ่มยกเลิก (Reset) โดยเราสามารถที่จะตกแต่งให้ดูสวยงาม และกลมกลืนกับเว็บเพจของคุณได้ดังนี้

- กำหนดสีให้ตัวอักษรที่อยู่ภายใน Button โดยใช้คำสั่ง color
- กำหนดสีให้พื้นหลังของตัว Button โดยใช้คำสั่ง background-color
- กำหนดภาพพื้นหลังของตัว Button โดยใช้คำสั่ง background-image
- กำหนดรูปแบบกรอบของ Button โดยใช้คำสั่ง border , border-top , border-bottom , border-left , border-right

```

<input Type="button" Name="text1" Value="นี่คือปุ่มที่1" Style=" color:#FFFFFF;
,background-color:#000000; ,border:thin dotted red; ,cursor:hand; ">
<input Type="button" Name="text2" Value="นี่คือปุ่มที่2" Style="
color:#0000FF; ,background-color:#C0C0C0; ,border:thin groove #808080; ">

```

ผลที่ได้คือ



## **การทำเมนูด้วย CSS**

### 1.เขียนข้อกำหนดคุณสมบัติ

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>Untitled</title>
</head>
<body>
<style type="text/css">
  <!--
  #navigation{
    margin-left: 20px;
    background-color: #fff;
    border-bottom:1px solid #000;
    padding: 0;
  }
  #navigation a{
    text-decoration: none;
    color: #06f;
    margin: 0 0.4em;
    border-left: 0px ;
    padding: 0 0 0 0.5em;
    width: 500px;
  }
  #navigation a:link{
    color: blue;
  }
  #navigation a:visited{
    color: #930;
  }
  #navigation a:active{
    background-color: #fff;
  }
  </style>
</body>
</html>
```

```
#navigation a:hover{
    background-color: white;
    border: 1px dashed white;
}
-->
</style
<div id="navigation"> <a href="#main" accesskey="s">Skip Navigation</a>
| <a href="index.html">Room News</a>
| <a href="Activity.html">Activity</a>
| <a href="Accessibility.html">what's Accessibility</a>
| <a href="contact.html">contact</a>
| <a href="sitemap.html">Site Map</a> </div>
</body>
</html>
```

ผลของการเขียนเป็นดังนี้

[Skip Navigation](#) | [Room News](#) | [Activity](#) | [what's Accessibility](#) | [contact](#) | [Site Map](#)

**พอก่อนนะครับ ศึกษาเองพยายามถ้าไม่เข้าใจหาตัวอย่างดูนะครับ**