

CSS (Cascading Style Sheets)

By Mr.Siravut

LOGO

CSS คืออะไร?

- ❖ XHTML/HTML ทำหน้าที่ในการวางโครงสร้างเอกสาร
- ❖ CSS ทำหน้าที่ในการตกแต่งเอกสาร

ข้อดีของ CSS

- สามารถแยกการตกแต่งออกจากเนื้อหาของเอกสาร ทำให้สามารถนำเอกสารหลักไปใช้งานกับ Application อื่นได้
- สามารถแก้ไขการตกแต่งของทุกหน้าเอกสารได้โดยใช้ไฟล์ CSS ตัวเดียว
- โหลดได้เร็ว (โหลดเนื้อหา ก่อน)
- กำหนดการแสดงผลให้กับอุปกรณ์ต่าง ๆ กันได้ เช่น จอคอมพิวเตอร์ เครื่องพิมพ์ เครื่องพิมพ์อักษรเบรลล์ WebTV โทรศัพท์มือถือ ฯลฯ
- มีขีดความสามารถในการออกแบบที่หลากหลาย ซึ่งวิธีธรรมดาไม่สามารถทำได้

แนวคิดแบบ CSS

How Styles "Cascade"

1 Browser CSS default

2 External Style Sheet



3 Internal Style Sheet

```
7 <head>
8 </head>
9 <title>Style Sampler</title>
10 <style type="text/css">
11 H1 {font-family: Times;
12 color: green;
13 P {margin: 10px;
14 hr {background: darkred;
15 height: 20px;
16 td {padding: 1.5cm;
17 </style>
18 </head>
19 <body>
```

4 Inline Style

CSS #1

CSS #2

CSS #3

<HTML /> CODE



การเรียกใช้งาน CSS

1. External CSS (เอกสาร CSS ภายนอก)

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html;  
charset=tis-620" />
```

```
<link rel="stylesheet" href="/f2000/F2000/main.css"  
type="text/css" media="screen" />
```

```
<title>ข้อมูลการอัปเดตเว็บไซต์ของสมาชิก</title>
```

```
</head>
```

* ควรใช้วิธีนี้ มากกว่ารูปแบบการใช้งานอื่น ๆ

การเรียกใช้งาน CSS - ต่อ

2. Internal CSS (โดยใช้แท็ก style ตรงส่วน head ของเอกสาร html)

```
<head>
<title>About My Organization</title>
<link rel="stylesheet" type="text/css" href=" ../apcdcss.css" title="Default CSS" />
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<style type="text/css" media="all">
h2 {
  background-color: #0066cc;
  font-size: x-large;
  padding: 0 0.5em;
  color: #fff;
}
</style>
</head>
```

* ไม่ควรใช้วิธีนี้มากนัก

การเรียกใช้งาน CSS - ต่อ

3. Inline CSS (โดยใช้แอททริบิวต์ style)

```

```

* แนะนำไม่ให้ใช้วิธีนี้



รูปแบบการใช้งาน CSS

CSS SYNTAX (ไวยากรณ์)

CSS Syntax (ไวยากรณ์)

```
Selector    {  
    Property: Value;    /* This is a comment */  
}
```

หรือ

```
@media MediaName    {  
    Selector    {  
        Property: Value;  
    }  
}
```

```
/* This is a comment */
```

CSS Syntax - ต่อ

```
h1    {  
      background-color: blue; /* This is a comment */  
    }
```

หรือ

```
@media print    {  
    h1    {  
          text-align: center;  
    }  
}
```

```
/* This is a comment */
```

CSS Syntax - ต่อ

```
/* For mobile devices */  
  
@media handheld {  
    h1 {  
        background-color: blue;  
    }  
}  
  
/* For printers */  
  
@media print {  
    h1 {  
        text-align: center;  
    }  
}
```

CSS Length Units (หน่วยความยาว)

ตัวอย่างของหน่วยแบบเชิงสัมพัทธ์ (สามารถปรับขนาดได้)

- em (em, ความสูงของฟอนต์) เช่น 2em, 1.5em, 0.88em ฯลฯ
- ex (x-height, ความสูงของตัวอักษร "x") เช่น 1.5ex, 2ex, ฯลฯ
- px (pixels, สัมพันธ์กับค่าความละเอียดของ Canvas) เช่น 1px, 4px, ฯลฯ
- % (percent) เช่น 50%, 130%, ฯลฯ

ตัวอย่างของหน่วยแบบคงที่ (ไม่สามารถปรับขนาดได้)

- in (inches; 1in=2.54cm) เช่น 2in, 1.5in, ฯลฯ
- cm (centimeters; 1cm=10mm) เช่น 2cm, 1.11cm, ฯลฯ
- mm (millimeters) เช่น 50mm, 0.8mm, ฯลฯ
- pt (points; 1pt=1/72in) เช่น 12pt, 20pt, ฯลฯ
- pc (picas; 1pc=12pt) เช่น 1pc, 2pc, ฯลฯ

CSS Length Units - ต่อ

```
/* ตัวอย่างการใช้ Length Units */
```

```
h1 {  
    font-size: 2em;  
    margin: 1ex;  
    padding: 0.5cm;  
    width: 90%;  
    top: 12pt;  
    left: -1in;  
}
```

ค่าสี (Color Units)

1. 16 ค่าสีจาก Windows VGA palette: aqua, black, blue, fuchsia (แดงอมม่วง), gray, green, lime, maroon (น้ำตาลแดง), navy, olive, purple, red, silver, teal (เขียวขุ่นเปิด), white, และ yellow
2. ค่าสีแบบ RGB:
 - a) `#rrggbb` เช่น `#00cc00`
 - b) `#rgb` เช่น `#ec0` หมายถึง `#eecc00`
 - c) `rgb(x,x,x)` โดยที่ `x` คือจำนวนเต็มตั้งแต่ **0-255** เช่น `rgb(0,204,0)`
 - d) `rgb(y%,y%,y%)` โดยที่ `y` คือเปอร์เซ็นต์ตั้งแต่ **0%-100%** เช่น `rgb(0%,80%,0%)`

ค่าสี - ต่อ

```
/* ตัวอย่างการใช้ Length Units */  
/* ใช้ชื่อสีจาก Windows VGA Palette */  
h1          { background-color: yellow; }  
/* ใช้สี RGB แบบเลขฐาน 16 จำนวน 6 หลัก */  
p           { color: #0011bb; }  
/* ใช้สี RGB แบบเลขฐาน 16 จำนวน 3 หลัก */  
div#content { background-color: #01b; }  
/* ใช้สี RGB แบบตัวเลข 3 ชุด (0-255) */  
span.news   { color: rgb(34, 45, 255); }  
/* ใช้สี RGB แบบตัวเลข 3 ชุด (0% - 100%) */  
img         { background-color: rgb(2%, 30%, 100%); }
```

URL

```
/* CSS Document */
```

```
body { background: url(stripe.gif) }
```

```
body { background: url(http://www.htmlhelp.com/stripe.gif) }
```

```
body { background: url( stripe.gif ) }
```

```
body { background: url("stripe.gif") }
```

```
body { background: url(\"Ulalume\".png) }
```

```
@import url(mycss.css);
```

CSS Selector

CSS Selector กำหนด Element ที่จะทำงานด้วย เช่น

1. ทำงานกับแท็กใดแท็กหนึ่ง เช่น h1, p, ul, li, em, ฯลฯ
2. ทำงานกับแท็กที่ใช้แอททริบิวต์ class เท่ากับค่าใดค่าหนึ่ง
3. ทำงานกับแท็กที่ใช้แอททริบิวต์ id เท่ากับค่าใดค่าหนึ่ง
4. ทำงานกับแท็กที่ใช้แอททริบิวต์บางอย่าง
5. ทำงานกับแท็กที่ติดกับแท็กใดแท็กหนึ่ง
6. ทำงานกับแท็กที่เป็น Element ลูกของแท็กใดแท็กหนึ่ง
7. ทำงานกับแท็กโดยมีเงื่อนไขมากกว่า 1 เงื่อนไขข้างต้น ฯลฯ

CSS Selector - ต่อ

- ❖ Element Selector (ใช้แท็กเป็นตัวเลือก)
- ❖ ID Selector (ใช้ ID เป็นตัวเลือก)
- ❖ Class Selector (ใช้ Class เป็นตัวเลือก)
- ❖ Selector ของลิงค์ (a-anchor)
- ❖ Select แบบผสม

Element Selector (ใช้แท็กเป็นตัวเลือก)

```
/* CSS document */  
h1    {  
        text-align: center;  
        font-size: 2em;  
    }
```

```
<!-- HTML Document -->  
...<body>  
<h1>หัวข้อลำดับที่ 1</h1>  
</body>...
```

Element Selector - ต่อ

```
/* CSS document */  
  
p    {  
    text-align: justify;  
    padding: 0.5em;  
}
```

```
<!-- HTML Document -->  
...<body>  
<h1>หัวเรื่องลำดับที่ 1</h1>  
<p>ย่อหน้าที่หนึ่ง</p>  
</body>...
```

ID Selector

```
/* CSS document */  
#content{  
    background-color: rgb(0, 34, 50);  
    color: #00cc00;  
}
```

```
<!-- HTML Document -->  
...<div id="content">  
<h1>หัวข้อลำดับที่ 1</h1>  
<p>ย่อหน้าที่หนึ่ง</p>  
</div>...
```

ID Selector - ต่อ

```
/* CSS document */  
div#content { /* เขียนชื่อแท็กกำกับไว้ได้ */  
    background-color: rgb(0, 34, 50);  
    color: #00cc00;  
}
```

```
<!-- HTML Document -->  
...<div id="content">  
<h1>หัวข้อลำดับที่ 1</h1>  
<p>ย่อหน้าที่หนึ่ง</p>  
</div>...
```

* ในหน้าเว็บเพจเดียวกันจะใช้ ID ชื่อซ้ำกันไม่ได้ การตั้งชื่อ ID ต้องไม่มีช่องว่าง ไม่ขึ้นต้นด้วยตัวเลข

Class Selector - ต่อ

```
/* CSS document */  
  
.news {  
    background-color: rgb(0, 34, 50);  
    color: #00cc00;  
}
```

```
<!-- HTML Document -->  
...<div id="content">  
<h1>หัวเรื่องลำดับที่ 1</h1>  
<p class="news">ย่อหน้าที่หนึ่ง</p>  
</div>...
```

* ในหน้าเว็บเพจเดียวกันจะใช้ ID ชื่อซ้ำกันไม่ได้ การตั้งชื่อ ID ต้องไม่มีช่องว่าง ไม่ขึ้นต้นด้วยตัวเลข

Class Selector - ต่อ

```
/* CSS document */  
  
p.news {  
    background-color: rgb(0, 34, 50);  
    color: #00cc00;  
}
```

```
<!-- HTML Document -->  
...<div id="content">  
<h1>หัวเรื่องลำดับที่ 1</h1>  
<p class="news">ย่อหน้าที่หนึ่ง</p>  
</div>...
```

* ในหน้าเว็บเพจเดียวกันสามารถเรียกใช้คลาสเดียวกันได้มากกว่า 1 ครั้ง ชื่อคลาสต้องไม่มีช่องว่าง ไม่ขึ้นต้นด้วยตัวเลข

Selector ของลิงค์ (แท็ก a)

```
/* CSS document */  
  
a:link          { color: blue; }  
  
a:visited { color: gray; }  
  
a:hover        { color: red; }  
  
a:active { color: green; }
```

```
<!-- HTML Document -->  
...<div id="content">  
<h1>หัวข้อลำดับที่ 1</h1>  
<p>...<a href="http://www.firefox.com">ดาวน์โหลด  
โปรแกรม Firefox</a>...</p>  
</div>...
```

Selector แบบผสม

```
/* CSS document */  
p span { /* แท็ก span ที่อยู่ในแท็ก p */  
    text-decoration: underline;  
}
```

```
<!-- HTML Document -->  
...<div id="content">  
<h1>หัวข้อลำดับที่ 1</h1>  
<p>...<span>ข้อความ</span>...</p>  
<h2><span>หัวข้อลำดับที่ 2</span></h2>  
</div>...
```

Selector แบบผสม - ต่อ

```
/* CSS document */  
p span.underline {  
    /* แท็ก span ที่ใช้ class="underline" ที่อยู่ในแท็ก p */  
    text-decoration: underline;  
}
```

```
<!-- HTML Document -->  
...<div id="content">  
<h1>หัวข้อลำดับที่ 1</h1>  
<p>...<span class="underline">ข้อความ  
</span>...<span>ข้อความถัดไป</span>...</p>  
</div>...
```

Selector แบบผสม - ต่อ

```
/* CSS document */  
div#content ul li {  
    /* แท็ก li ที่อยู่ใน ul ที่อยู่ใน div ที่ใช้ id="content" */  
    list-style-type: upper-roman;  
}
```

```
<!-- HTML Document -->  
...<div id="content">  
...<ul>  
    <li>Item A</li>  
    <li>Item B</li>  
...</ul>...  
...</div>...
```

Selector แบบผสม - ต่อ

```
/* CSS document */  
div#mainmenu a:hover {  
/* แท็ก a (hover action) ที่อยู่ใน div ที่ใช้ id="mainmenu" */  
    font-weight: bold;  
}
```

```
<!-- HTML Document -->  
...<div id="mainmenu">  
...<ul>  
    <li><a href="www.somewhere.com">Item A</a></li>  
    <li><a href="www.something.com">Item B</a></li>  
</ul>...  
</div>...
```

Selector แบบผสม - ต่อ

```
/* CSS document */  
h1, h2, h3, h4 {  
    /* กำหนดการแสดงผลหลายแท็กพร้อมกัน */  
    font-weight: bold;  
}
```

```
<!-- HTML Document -->  
...<div id="mainmenu">  
  <h1>Heading 1</h1>  
  <h2>Heading 2</h2>  
  <h3>Heading 3</h3>  
  <h4>Heading 1</h4>  
</div>...
```

เมื่อใดต้องใช้ Selector แบบไหน

Element Selector (การใช้ชื่อแท็กเป็นตัวเลือก)

วิธีการนี้ เหมาะสำหรับการ "กำหนดการแสดงผลทั่ว ๆ ไปให้กับแท็กนั้น ๆ" หมายถึงว่า ไม่ว่าจะเรียกใช้แท็กนั้นเมื่อไร ก็จะแสดงผลในรูปแบบที่กำหนดไว้เสมอ ทุกหน้า ทุกเอกสาร

เมื่อใดต้องใช้ Selector แบบไหน - ต่อ

Element Selector (การใช้ชื่อแท็กเป็นตัวเลือก)

วิธีการนี้ เหมาะสำหรับการ "กำหนดการแสดงผลทั่ว ๆ ไปให้กับแท็กนั้น ๆ" หมายถึงว่า ไม่ว่าจะเรียกใช้แท็กนั้นเมื่อไร ก็จะแสดงผลในรูปแบบที่กำหนดไว้เสมอ ทุกหน้า ทุกเอกสาร เช่น ต้องการให้แท็ก `<h1>` อยู่ตรงกลางในทุก ๆ หน้าเอกสาร, ต้องการให้ตัวอักษรในแท็ก `<p>` ในทุกเอกสารใช้ตัวอักษรสีเทาเข้ม เป็นต้น

เมื่อใดต้องใช้ Selector แบบไหน - ต่อ

Class Selector (การใช้คลาสเป็นตัวเลือก)

เหมาะสำหรับการสร้างการแสดงผลบางอย่างให้กับ Element เป็นครั้งคราว โดยจะมีผลกับเฉพาะอิลิเมนต์ที่เรียกใช้คลาสนั้น หากไม่ได้เรียกใช้ก็จะแสดงผลตามค่าปกติที่กำหนดไว้ และสามารถเรียกใช้ได้มากกว่า 1 ครั้ง เช่น ต้องการขีดเส้นใต้ข้อความ, ต้องการให้ตัวหนังสือเป็นสีแดง เป็นต้น

เมื่อใดต้องใช้ Selector แบบไหน - ต่อ

ID Selector (การใช้ ID เป็นตัวเลือก)

วิธีการนี้เหมาะกับการวางเลย์เอาต์เอกสาร เพื่อให้ง่ายต่อการกำหนดการแสดงผลให้กับแต่ละส่วนโดยไม่กระทบกับส่วนอื่น ๆ มักใช้แท็ก div พร้อมกับกำหนดแอททริบิวต์ id ด้วยเพื่อตั้งชื่อให้กับส่วนนั้น ๆ สิ่งที่ต้องพึงระวังคือ อย่าใช้ ID ชื่อเดียวกันในเอกสารเดียวกัน เนื่องจากหากมีการเขียนสคริปต์เพื่อให้มีผลกับ ID แล้วจะเกิดข้อผิดพลาดได้ (สามารถเรียกใช้ ID เดิมได้เพียงครั้งเดียวเท่านั้น) ตัวอย่างการใช้งาน

CSS Layout (การแบ่งส่วนเอกสาร)

1

```
<div id="mainmenu">  
width: auto;  
margin: 0;  
padding: 1em;
```

2

```
<div id="localmenu">  
float: left;  
width: 14em;  
margin: 0;  
padding: 1em;
```

3

```
<div id="content">  
width: auto;  
margin-left: 16em;  
padding: 1em;
```

4

```
<div id="footer">  
width: auto;  
margin: 0;  
padding: 0;
```

CSS Layout – ต่อ

1

```
...  
<body>  
<!-- Main Menu -->  
<div id="mainmenu">
```

2

```
.....  
</div>  
<!-- Local Menu -->  
<div id="localmenu">
```

3

```
.....  
</div>  
<!-- Content -->  
<div id="content">
```

4

```
.....  
</div>  
<!-- Footer -->  
<div id="footer">  
.....  
</div>  
</body>  
</html>
```

CSS Layout+ ต่อ

```
html, body {
    margin: 0;
    padding: 0;
    font-size: medium;
    font-family: Sans-serif, Arial;
    background-color: #fff;
    color: #000;
}
```

```
/* Main menu */
```

```
div#mainmenu {
    width: auto;
    margin: 0;
    padding: 1em;
    background-color: blue;
}
```

```
/* Local menu */
```

```
div#localmenu {
    width: 14em;
    float: left;
    margin: 0;
    padding: 1em;
    background-color: green;
}
```

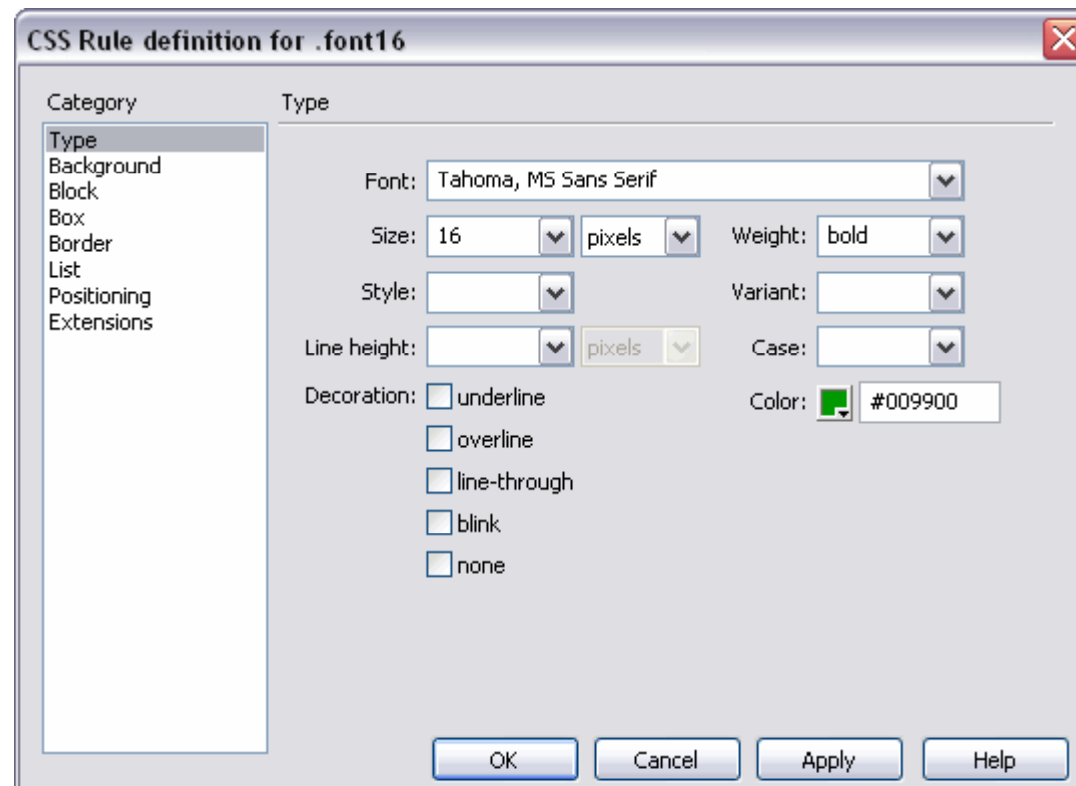
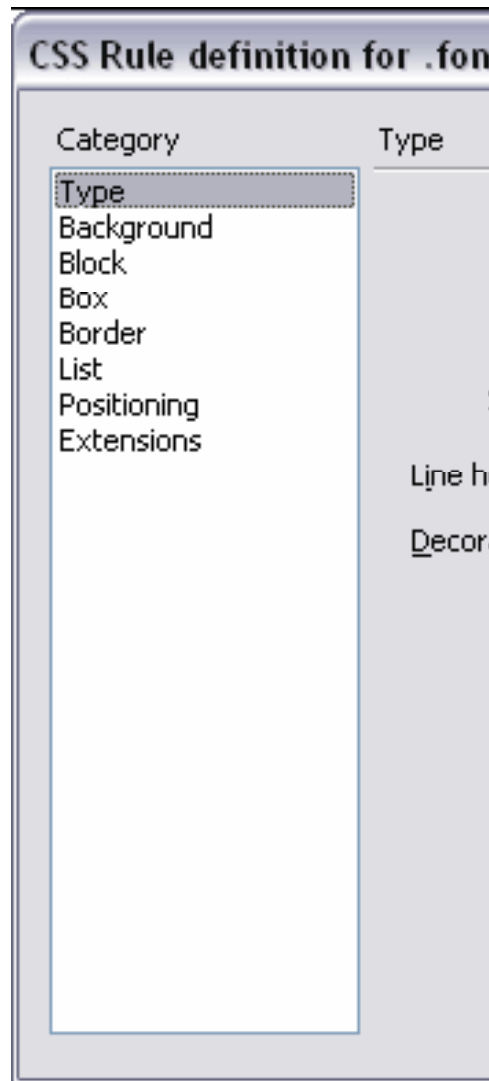
```
/* Content */
```


```
div#content {
    width: auto;
    margin: 0;
    padding: 1em;
    background-color: orange;
}
```

```
/* Footer */
```

```
div#footer {
    width: auto;
    margin: 0;
    padding: 1em;
    background-color: fuchsia;
}
```

CSS in Dreamweaver



- 
- ❖ **Type** ใช้ในการกำหนดค่าคุณสมบัติรูปแบบตัวอักษร หรือ **Font** เช่น การกำหนดสีให้ตัวอักษร ขนาดของตัวอักษร เป็นต้น
 - ❖ **Background** ใช้ในการกำหนดค่าคุณสมบัติของพื้นหลัง เช่น สีของพื้นหลัง การกำหนดค่า **repeat** ของพื้นหลัง เป็นต้น
 - ❖ **Block** ใช้ในการกำหนดค่า คุณสมบัติรูปแบบการแสดงผลของกลุ่มข้อความหรือกลุ่มของ **Object** ที่เรียก **CSS** นั้นใช้งาน
 - ❖ **Box** ใช้ในการกำหนดค่าคุณสมบัติความสูง-กว้างของกลุ่มข้อความ หรือ **Object** และใช้กำหนดระยะห่างของ **Object (Margin)**
 - ❖ **Border** ใช้ในการกำหนดค่า คุณสมบัติของเส้นกรอบ ลักษณะของเส้นกรอบที่ต้องการ
 - ❖ **List** ใช้ในการกำหนดค่าคุณสมบัติของรายการ เช่น การกำหนดรูปร่างหน้ารายการ เป็นต้น
 - ❖ **Positioning** ใช้ในการกำหนดค่าคุณสมบัติลักษณะตำแหน่งแสดงผลของ **Object** ที่เรียก **CSS** นี้
 - ❖ **Extension** ใช้ในการกำหนดค่าคุณสมบัติอื่น ๆ ที่เพิ่มเติมขึ้นมา เช่น การกำหนดค่า **Filter** ให้ทำเอฟเฟกต์กับรูปภาพ เป็นต้น ซึ่งคุณสมบัติของ **Option** นี้จะรองรับทาง **IE** เสียมากกว่า