

IS216 การพัฒนาโปรแกรมประยุกต์โดยผู้ใช้
IS281 เครื่องมือส่งเสริมความสำเร็จของผู้ใช้
หัวข้อ

การตกแต่งเว็บด้วย CSS ขั้นต้น

ผศ.วันชัย ชันดี wanchai@tbs.tu.ac.th

คณะพาณิชยศาสตร์และการบัญชี มธ.
<http://www.bus.tu.ac.th/usr/wanchai/IS281>

หลักการเขียนเว็บที่ดี

2

วิธีบอกเนื้อหา (content) กับวิธีบอกลักษณะการเสนอ (style) ควรแยกออกจากกัน

- ต้องการนำเสนออะไร (เนื้อหา) บอกด้วย HTML, XML, XHTML
- นำเสนออย่างไร (รูปแบบ ลักษณะการแสดงผล) บอกด้วย CSS

เกร็ด
ความรู้

เดิมการจัดรูปแบบการนำเสนอต้องกำหนดไว้ในแท็กของ HTML หลังจากมี CSS
เราสามารถแยกการจัดรูปแบบการนำเสนอไปไว้ใน CSS ได้

มาตรฐาน HTML ที่ออกมาใหม่ก็พยายามตัดส่วนที่ใช้ในการจัดรูปแบบการนำเสนอออกไป

ข้อดีของการแยกเนื้อหาและลักษณะการแสดงผล ออกจากกัน

3

- รูปแบบ ลักษณะของการแสดงผลหรือการนำเสนอเป็นอิสระจากเนื้อหา
- ง่ายต่อการควบคุมกระบวนแบบ(style) ให้เป็นแบบเดียวกันทั้งเว็บไซต์
- ง่ายต่อการปรับเปลี่ยนแก้ไขรูปแบบในการนำเสนอ
- ทำให้เอกสารต้นฉบับสั้นและดูเรียบง่าย ไฟล์มีขนาดลดลง โหลดได้เร็วขึ้น
- ปรับรูปแบบในการนำเสนอให้เหมาะสมกับอุปกรณ์ต่างชนิดกันได้ง่าย

Cascading Style Sheets

4

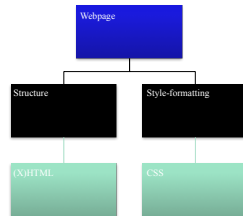
Cascading Style Sheets (CSS) เป็นภาษามาตรฐานสำหรับอธิบายลักษณะ
การนำเสนอเอกสารที่เขียนด้วยภาษา mark up เช่น HTML และ XML โดย
สามารถเขียนรวมไปกับเอกสาร HTML หรือแยกออกมาเป็นไฟล์ต่างหากก็ได้

สไตล์หลายแบบอาจนำมาต่อพ่วงกัน (cascading) เพื่อแสดงผลเป็นหนึ่งเดียวที่สัมพันธ์กัน

ทำไมจึงควรใช้สไตล์?

5

- เอกสารที่เขียนด้วย CSS -
 - มีความยืดหยุ่น แก้ไขยกชุดได้ง่าย
 - เป็นไปได้ที่จะทำให้เอกสารต้นฉบับสั่นกระชับ
 - ดูง่ายขึ้น (แต่ต้องเข้าใจ CSS ซึ่งต้องเรียนรู้เพิ่มเติม)
- เป็นเครื่องมือพื้นฐานในการจัดรูปแบบข้อความและหน้าเว็บ
- ง่ายในการจัดการเอกสารที่มีหน้าเป็นจำนวนมาก
- ประหยัดเวลาด้วยการเขียนคลาสแล้วนำไปใช้ในจุดต่าง ๆ ได้หลายที่
- เพิ่มโอกาสในการจัดรูปแบบต่าง ๆ ได้มากขึ้น



ประวัติความเป็นมาของ CSS

6

ปีพ.ศ. ๒๕๓๗ Hakon Wium Lie จาก MIT สร้าง CSS
ต่อมาเสนอเข้าองค์กร W3C เพื่อปรับปรุงให้เป็นมาตรฐาน และได้ประกาศใช้ร่วมกับ HTML 4.0 ครั้งแรกปี
พ.ศ. ๒๕๓๙

Version	ปี	Features
CSS 1	๒๕๓๙	Fonts, Colors, Alignment, Spacing
CSS 2	๒๕๔๑	Layout, positioning
CSS 2.1	๒๕๔๗	
CSS 3	๒๕๕๕ (ยังพัฒนาไม่เสร็จแต่ทยอยนำออกใช้)	Media Query, Effects, sizing, speech

หลักไวยากรณ์ CSS

7

CSS ประกอบด้วยกฎ (rule) ต่าง ๆ ที่บอกเบราว์เซอร์ว่าจะจัดรูปแบบการแสดงผล
ส่วนประกอบต่าง ๆ ของเอกสารอย่างไร

กฎแต่ละข้อประกอบด้วย

- *selector* - เป็นตัวเลือกว่ากฎนี้จะใช้กับแท็กใด
 - *declarations* - บ่งบอกคุณสมบัติต่าง ๆ ของรูปแบบในการแสดงผล
- แต่ละ declaration มีองค์ประกอบ 2 ส่วนคือ
- *property* - ชื่อสมบัติของรูปแบบ
 - *value* - ค่าที่จะกำหนดให้ property

หลักไวยากรณ์ CSS

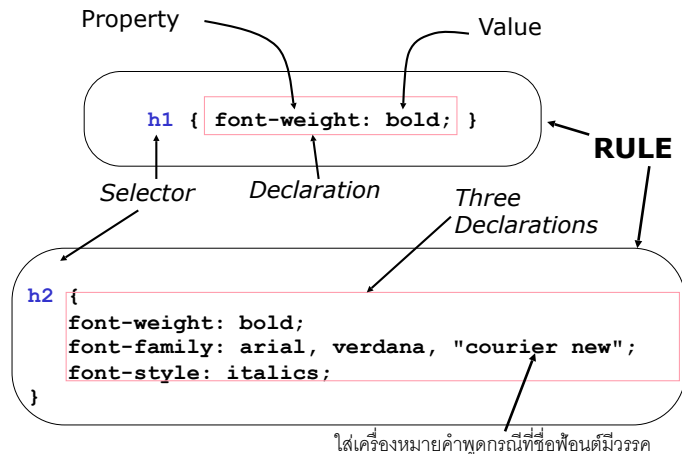
8

- ลักษณะการแสดงผล (style) และกฎ (rule) ซึ่งเปลี่ยนพฤติกรรมของแท็กในเอกสาร
- กฎแต่ละข้อจะมีตัวเลือก (selector) ซึ่งมักจะเป็นแท็ก HTML และ declaration ซึ่งประกอบด้วย property กับค่าของมันเป็นคู่ ๆ (property/value pair)

```
selector { property: value }
```



RULE = Selector + Declaration(s)
 Declaration = Property + Value(s)



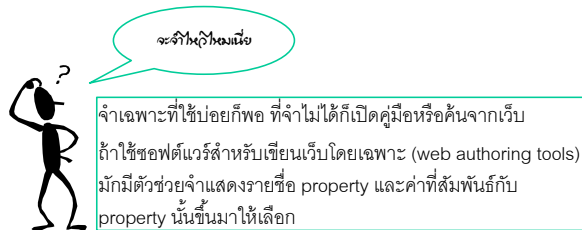
Comment

ใส่คำอธิบายหรือหมายเหตุไว้ในเครื่องหมาย /* ... */ เช่น

```
h1 { text-align: center;
    /* This is a comment */
    font-weight: bold;
  }
```

Property มีมากเหลือเกิน

- CSS1 มีประมาณ 50 properties
- CSS2 มีประมาณ 120 properties
- CSS3 แบ่งเป็น 26 modules มีคณะทำงานหลายคณะแยกกันกำหนดมาตรฐาน และแยกกันประกาศใช้



การระบุค่าให้ property

การบอกค่าของ property มีหลายแบบ เช่น

- คำเฉพาะ (keyword) เช่น **thin, thick, larger** ตามค่าที่มีให้เลือกของ property นั้น ๆ
 - ค่าตัวเลขจำนวนเต็มหรือเลขมีทศนิยมพร้อมด้วยหน่วยวัด เช่น 10mm
 - ค่าตัวอักษร ถ้ามี space ปนอยู่ด้วยต้องใส่ไว้ในเครื่องหมายคำพูด เช่น ชื่อฟอนต์ "Cordia New"
 - สัดส่วนเชิงเปรียบเทียบ เช่น 50%
 - ค่าที่บอกเป็นรูป Functional Notation เช่น ค่าสี เขียนเป็น rgb(255, 0, 0)
 - ค่า auto เป็นค่าที่คำนวณให้โดยอัตโนมัติโดยเบราว์เซอร์
- ทั้งนี้ ต้องกำหนดค่าให้เหมาะสมกับบริบทของ property ที่กำลังใช้

หน่วยการวัด (Units of Measure)

- ค่าข้อมูลที่จะใส่ให้ property บางตัวจำเป็นต้องบอกหน่วยด้วย
- หน่วยของ CSS มีดังนี้
 - หน่วยความยาว (length หรือ distance unit)
 - หน่วยสี (color unit)
 - หน่วยมุม (angle unit) มี deg, grad, rad, turn
 - หน่วยระยะเวลา (duration unit) มี s กับ ms
 - หน่วยความถี่ (frequency unit) มี Hz กับ kHz
 - หน่วยความละเอียด (resolution unit) มี dpi, dpcm, dppx
 - หน่วยร้อยละ (percentage unit)

หน่วยความยาว

- CSS มีหน่วยความยาวหลายชนิดให้เลือกใช้สำหรับบอกขนาดหรือบอกระยะห่างหรือบอกตำแหน่งขององค์ประกอบของหน้าเว็บ
- เราสามารถบอกขนาดหรือระยะห่างด้วยตัวเลขแล้วตามด้วยหน่วยที่ต้องการให้กับ property ที่ต้องการค่าความยาว
- หน่วยความยาวมีทั้งที่เป็น
 - หน่วยสัมบูรณ์ (absolute) ซึ่งบอกขนาดที่ตายตัว ได้แก่ cm, in, mm, pc, pt
 - หน่วยสัมพัทธ์ (relative) ซึ่งบอกขนาดเชิงเปรียบเทียบ ได้แก่ em, ex, px
- ค่าที่ใส่อาจมีเครื่องหมาย + หรือ - นำหน้า แล้วตามด้วยค่าตัวเลข แล้วปิดท้ายด้วยหน่วย ถ้าระบุค่าเป็นศูนย์ ไม่จำเป็นต้องใส่หน่วยก็ได้

หน่วยความยาว

ชื่อย่อ	ชื่อเต็ม	คำอธิบายเพิ่มเติม
cm	Centimeters	เซ็นติเมตร
em	Em space	วัดค่าตามความกว้างของอักษร M ตัวใหญ่สำหรับฟอนต์ที่กำลังใช้
ex	x-height	วัดค่าตามความสูงของอักษร x ตัวเล็กสำหรับฟอนต์ที่กำลังใช้
in	Inches	นิ้ว
mm	Millimeters	มิลลิเมตร
pc	Picas	วัดค่าเป็น pica ซึ่ง 1 pica มีค่าเท่ากับ 1/6 ของหนึ่งนิ้ว
pt	Points	วัดค่าเป็นพ้อยต์ ซึ่ง 1 pt มีค่าเท่ากับ 1/72 ของหนึ่งนิ้ว
px	Pixels	วัดค่าเป็นระยะทางเท่ากับจุดภาพ (pixel) ของอุปกรณ์แสดงผล
Q	quarter-millimetres	ความยาวมีค่าเท่ากับ 1/4 ของ มิลลิเมตร หรือเท่ากับ 1/40 ของ 1 เซนติเมตร

หน่วย em, ex

ความยาวของหน่วย em และ ex ขึ้นอยู่กับขนาดของ ตัว M และ x ของฟอนต์ปัจจุบันที่กำลังใช้อยู่



เหมาะสำหรับบอกระยะเชิงเปรียบเทียบกับขนาดของฟอนต์ปัจจุบันที่กำลังใช้อยู่ ซึ่งถ้ามีการเปลี่ยนขนาดของฟอนต์ ระยะซึ่งใช้หน่วยชนิดนี้จะเปลี่ยนตามไปด้วย

หน่วยร้อยละ (Percentage Unit)

17

- เป็นทางเลือกในการบอกหน่วยเชิงเปรียบเทียบ ซึ่งปรับค่าตามค่าอื่นตามค่าอื่นขึ้นกับบริบทที่ใช้
- สามารถใช้กับหน่วยต่าง ๆ ที่สามารถบอกค่าเป็นตัวเลขเช่นความยาว ความถี่ มุม ระยะเวลา
- เรากำหนดค่าเป็นร้อยละให้กับ property ด้วยการใส่ตัวเลขซึ่งแทนร้อยละแล้วตามด้วยเครื่องหมายเปอร์เซ็นต์ (%) เช่น padding-top: 5%

หน่วยสี (Color Units)

18

- ค่าสีอาจบอกด้วยชื่อสีเป็นคำหลัก (keyword) ตามมาตรฐาน HTML มี 16 สีดังนี้

maroon #800000	red #ff0000	orange #ffa500	yellow #ffff00	olive #808000
purple #800080	fuchsia #ff00ff	white #ffffff	lime #00ff00	green #008000
navy #000080	blue #0000ff	aqua #00ffff	teal #008080	
black #000000	silver #c0c0c0	gray #808080		

หมายเหตุ

บางแหล่งบอกว่าสีมาตรฐานมี 17 สีตามนี้ แต่ที่จริงแล้ว orange ไม่ได้เป็นสีมาตรฐาน แต่สามารถใช้ได้เพราะเบราว์เซอร์ทั่วไปรองรับสีมากกว่ามาตรฐานที่กำหนด ดูชื่อสีที่เบราว์เซอร์ส่วนใหญ่รองรับที่ http://w3schools.sinsixx.com/css/css_colornames.asp.htm

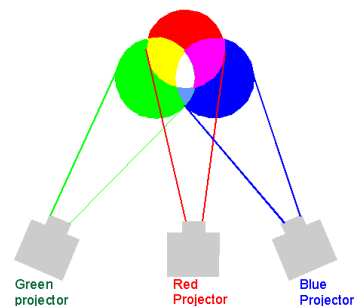
- หรือ บอกด้วยสัดส่วนของแม่สี (RGB) ผสมกันในรูปของเลขฐาน 16 แบบที่ใช้ในภาษา HTML หรือบอกในรูปแบบ Functional Notation

การผสมสีของแสง

19

แม่สี
(แบบ Additive)
ประกอบด้วย

แดง (Red)
เขียว (Green)
น้ำเงิน (Blue)



การบอกส่วนผสมของสี (แดง เขียว น้ำเงิน)

20

- กำหนดความเข้มของแม่สี ด้วยเลขฐานสิบหก
`#rrggbb` เช่น `#ff0000` ← ภาษา HTML ใช้แบบนี้แบบเดียว
- กำหนดความเข้มของแม่สี ด้วยเลขฐานสิบหกแบบย่อ
`#rgb` เช่น `#f00`
- กำหนดความเข้มของแม่สี ด้วยเลขฐานสิบ
`rgb(255, 0, 0)`
- กำหนดความเข้มของแม่สี เป็นร้อยละ
`rgb(100%, 0%, 0%)`
- กำหนดความเข้มของแม่สี และ **ความทึบแสง** ด้วยเลขฐานสิบ
`rgba(255, 0, 0, 0.5)` ← พารามิเตอร์ตัวที่ 4 เรียกว่า alpha เป็นระดับความทึบแสงมีค่า 0 ถึง 1
- กำหนดความเข้มของแม่สี เป็นร้อยละและ **ความทึบแสง**
`rgba(100%, 0%, 0%, 0.5)` ← 0 หมายถึงโปร่งแสงเต็มที่ 1 หมายถึงทึบแสงเต็มที่

ตัวอย่าง Property เกี่ยวกับฟอนต์ที่มีพบบ่อย

- font-weight
 - Normal | lighter | bold | bolder | 100 | 200 | ... | 900
- font-style
 - normal | italic | oblique
- font-family
 - ระบุชื่อฟอนต์ได้หลายชื่อ
- font-size
 - บอกขนาดโดยใช้หน่วยเป็น pt, pc, in, cm, mm, em, ex, px
 - บอกขนาดเชิงสัมพัทธ์เป็นร้อยละหรือ keyword % | xx-large | x-large | large | medium | small | x-small | xx-small | smaller | larger

Tip

ชื่อและค่าของ property มีความหมายอยู่ในตัว

ถ้ารู้ความหมายของคำภาษาอังกฤษก็เข้าใจและจำได้ไม่ยาก

ตัวอย่าง Property เกี่ยวกับข้อความที่มีพบบ่อย

- text-decoration
 - none | underline | overline | line-through | blink
- text-align
 - left | right | center | justify
- vertical-align การจัดข้อความในแนวดิ่ง
 - top | bottom | baseline | middle | sub | super | text-top | text-bottom | %
- text-indent หดบรรทัดแรกของพารากราฟ (ย่อหน้า)
 - +/- pt, pc, in, cm, mm | +/- em, ex, px, %
 - เช่น p { text-indent: -25px } ค่าติดลบหมายถึงยื่นออก 25 pixel
- line-height ระยะห่างระหว่างบรรทัด
 - normal | number | pt, pc, in, cm, mm | em, ex, px, %

ตัวอย่าง background property

- background-image
 - none | url (filename)
 - ตัวอย่างเช่น h1 { background-image: url (yellowband.gif); }
- background-repeat
 - repeat | repeat-x | repeat-y | no-repeat
 - ตัวอย่างเช่น body {
 - background-image: url (yellowband.gif);
 - background-repeat: repeat-x;
- background
 - ใช้บอกสมบัติหลายตัวเกี่ยวกับ background ข้างต้นรวมไว้ด้วยกันโดยไม่ต้องบอกชื่อ property ย่อย
 - ตัวอย่างเช่น p { background: url (textband.gif) repeat-x }

ตัวอย่าง Property เกี่ยวกับการจัดหน้า

```
{width: <ความกว้าง>}
{height: <ความสูง>}
{letter-space: <ระยะห่างระหว่างตัวอักษร หรือ normal>}
{line-height: < ระยะห่างระหว่างบรรทัด >}
{page-break-before: <กำหนดการขึ้นหน้าใหม่ก่อนแสดงผล auto, always, left, right>}
{page-break-after: <กำหนดการขึ้นหน้าใหม่หลังแสดงผล auto, always, left, right>}
{vertical-align: <align>}
{cursor: <รูปแบบ auto, crosshair, default, hand, move, e-resize,
ne-resize, nw-resize, n-resize, se-resize, sw-resize,
s-resize, w-resize, text, wait help>}
```

หมายเหตุ property เหล่านี้ต้องเลือกใช้กับ Element ที่เกี่ยวข้องและใช้อย่างเหมาะสม

การประยุกต์ใช้ CSS กับ HTML

เราสามารถนำ CSS มาใช้ร่วมกับ HTML ได้หลายวิธี

- ใช้แท็ก **<style>** แทรกลงใน head เพื่อบรรจุ CSS ลงในเอกสาร
- บอกใน CSS ว่าจะใช้สไตล์ที่เขียนกับแท็กใดบ้าง
- ใช้ attribute **class** หรือ **id** หรือ **style** ของแท็กต่าง ๆ เป็นตัวเชื่อมไปยังสไตล์ที่ต้องการ

HTML Global Attributes

มี attribute ของ html จำนวนหนึ่งที่เป็นตัวร่วมสามารถนำมาใช้กับแท็กต่าง ๆ โดยทั่วไปได้แทบทุกตัวเช่น

- **class** ใช้กำหนดชื่อกลุ่มหรือประเภทสำหรับอ้างอิงแท็กหลายตัวอยู่ในกลุ่มหรือ class เดียวกันได้
- **id** ใช้กำหนดชื่ออ้างอิงให้กับแท็ก ในหน้าเดียวกันชื่อจะต้องไม่ซ้ำกัน
- **style** ใช้กำหนด สไตล์ ให้กับแท็กด้วย CSS
- **title** ข้อมูลเพิ่มเติมสำหรับแท็ก อาจนำไปใช้เป็น tool tip หรืออื่น ๆ

3 ตัวแรกที่ถูกกล่าวถึงถูกนำไปใช้ร่วมกับ CSS เป็นหลัก

แท็กตัวช่วยสำหรับใช้กับ CSS โดยเฉพาะ

แท็กของ HTML สำหรับใช้งานทั่วไปสามารถใช้ร่วมกับ CSS ได้ แต่บางกรณีไม่เพียงพอ จึงมีการสร้างแท็กบางตัวที่สร้างขึ้นมาเพื่อใช้กับ CSS โดยเฉพาะได้แก่

- **<style>** ใช้แทรกกลุ่มของสไตล์ลงในส่วนหัวของ HTML
- **** ใช้แทรกในตำแหน่งที่ต้องการเปลี่ยนสไตล์
- **<div>** ใช้แบ่งส่วนของเอกสารหรือจัดกลุ่มข้อความหรือแท็กต่าง ๆ ให้อยู่ในกลุ่มเดียวกันที่ต้องการเปลี่ยนสไตล์

แท็ก

- เป็นแท็กตัวช่วยสำหรับระบุสไตล์ให้กับเนื้อหาย่อยในหน้าเว็บ
- เป็นแท็กสำหรับใช้ในระดับบรรทัด (in-line level element) เพื่อใช้จัดรูปแบบข้อความเฉพาะบางพื้นที่ซึ่งอยู่ใน element อื่น เช่นภายในพารากราฟ
- ไม่ทำให้ขึ้นบรรทัดใหม่ก่อนหรือหลัง element นี้

แท็ก <div>

- เป็นแท็กที่ทำงานเป็นบล็อก (block level element)
- เพื่อใช้จัดรูปแบบ แบ่งส่วน หรือแบ่งพื้นที่ของหน้าเว็บใน
- จะมีการตัดบรรทัดเพื่อขึ้นบรรทัดใหม่ทั้งก่อนและหลังแท็กนี้
- สามารถบรรจุแท็กต่าง ๆ ทั้งที่ทำงานเป็นบล็อกและไม่เป็นบล็อก

ชนิดของ Style Sheets

- Local Style Sheet (Inline style sheet)
เขียนแทรกไว้ในแท็กใดแท็กหนึ่ง มีผลกับแท็กตัวนั้นตัวเดียว
- Global Style Sheet (Embedded style sheet, Internal Style Sheet)
เขียนขึ้นมาไว้ใน head ของ HTML มีผลกับแท็กหลายตัวที่อยู่ในหน้าเดียวกัน
- External Style Sheet (Linking style sheet)
เขียนแยกต่างหากจาก HTML ไว้เป็นไฟล์อื่น สามารถนำไปใช้กับเว็บได้หลายหน้า

ตัวอย่าง local style sheet

```
<h2 style="font-size: 35pt;
color: lime;
line-height: 30pt;
font-weight: bold;
font-family: AngsanaUPC, CordiaUPC">
This is the new headline.
</h2>
```

ใช้ global attribute **style** ในการกำหนด CSS ให้กับแท็ก **h2** ตัวนี้เป็นการเฉพาะเพียงตัวเดียว ไม่มีผลกับแท็ก **h2** ตัวอื่น
ในที่นี้ **h2** ทำหน้าที่เป็น selector

[Web](#)

Local Style Sheet กับ tag <div> และ

```
<selector style="property: value;">
  ↓
  ↓
  ↓
<span style="color: #0000FF;"> ... </span>
<div style="color: #FF0000;"> ... </div>
```

แท็ก **** ใช้เปลี่ยนสมบัติของข้อความที่มันบรรจุให้มีสีได้ตามที่ระบุด้วย attribute **style** ข้อความที่ตามหลังตัวปิด **** จะไม่ขึ้นบรรทัดใหม่

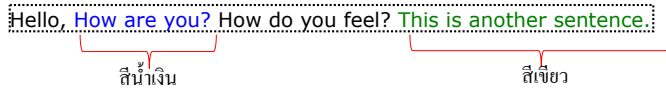
แท็ก **<div>** กำหนดขอบเขตของ section หลังจบ section ด้วยตัวปิด **</div>** จะขึ้นบล็อกใหม่

ตัวอย่าง local style sheet

```

Hello, <span style="color: blue;">
How are you? </span>How do you feel?
<span style="color: green;">
This is another sentence.</span>

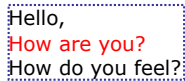
```



```

Hello, <div style="color: red;">
How are you?</div>How do you feel?

```



Global style sheet

```

<style>
  selector1 { property1: value1;
              property2:
value2; . . . }
  selector2 { property1: value1;
              property2:
value2; . . . }
</style>

```



```

<style>
  h4 {font-size: 17pt;
      font-family: AngsanaUPC,
CordiaUPC; }
  h2 {font-size: 15pt;
      color: blue}
</style>

```



ตัวอย่าง global style sheet

```

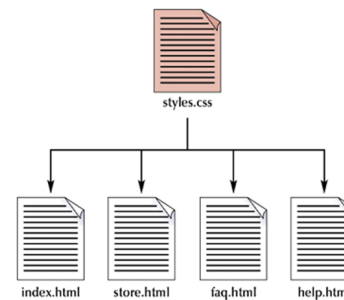
<html><head>
<style>
  h4 {font-size: 20pt;
      font-family: AngsanaUPC, CordiaUPC;
      color: maroon}
  h2 {font-size: 30pt;
      font-family: CordiaUPC;
      color: blue}
  p {font-size: 15pt;
      font-family: Arial Black;
      color:black}
</style>
</head>
<body>
<h2>Heading 2</h2>
<p>Normal text</p>
<h4>Heading 4</h4>
<p>normal text</p>
</body>
</html>

```

ใช้ tag <style> บรรจุไว้ใน ส่วนหัวของเอกสาร HTML

External Style Sheet

เขียน style sheet ครั้งเดียว ใช้กับหลายเอกสาร

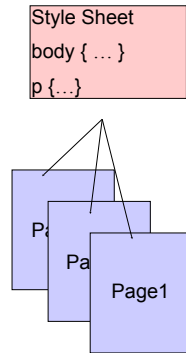


Style sheet 1 ไฟล์สามารถควบคุม
สี ลักษณะตัวอักษร และ ลักษณะการ
จัดวางให้กับเว็บเพจได้หลายหน้า

External Style Sheets

37

```
h4 { font-size: 20pt;
      font-family: AngsanaUPC,
                  CordiaUPC;
      color: maroon}
h2 { font-size: 30pt;
      font-family: CordiaUPC;
      color: blue}
p { font-size: 15pt;
     font-family: Arial Black;
     color: black
}
```



แยกเก็บเฉพาะ style sheet ไว้เป็นไฟล์สกุล .css

การใช้ External Stylesheet

38

1. เขียน Style Sheet ที่ต้องการแล้ว save เก็บเป็นไฟล์ให้มีนามสกุล .css ตัวอย่างเช่น style1.css
2. สร้างไฟล์ HTML ที่ต้องการนำไปใช้งาน
3. ในไฟล์ HTML ที่สร้างขึ้นให้ใส่ tag `<link>` บรรจุมุมุภายในแท็ก `<head>` ตัวอย่างรูปแบบ

```
<link rel="stylesheet" type="text/css" href="style1.css" />
```

ชื่อไฟล์ style sheet ที่สร้างขึ้น

CSS Selectors

39

การเลือกว่าจะนำ CSS ไปใช้ที่ใดบ้างทำได้หลายวิธีดังนี้

- เลือกแท็กทุกตัว (universal selector)
- เลือกด้วยชื่อแท็ก (HTML element name selector) (ตัวอย่างที่ผ่านมาใช้วิธีนี้)
- เลือกเป็นกลุ่มของแท็กที่ต้องการใช้สไตลเดียวกัน (class selector)
- เลือกตัวใดตัวหนึ่งด้วยชื่อที่ตั้งขึ้นมาเฉพาะ (id selector)
- เลือกตามค่าของ attribute (attribute selector)
- เลือกตามบริบท (contextual selector)
 - ตัวเลือกผู้สืบเชื้อสาย (descendant selector)
 - ตัวเลือกลูก (child selector)

ตัวเลือกครอบจักรวาล (Universal Selector)

40

เป็นการกำหนดให้สไตลซีทมีผลต่อทุกแท็กในเว็บเพจโดยใช้เครื่องหมาย * แทนตัวเลือก ตัวอย่างเช่น

```
* {
  margin:0;
  padding:10px;
}
```

เป็นการกำหนดให้ทุก element เว้นขอบนอก (margin) เป็น 0 และใส่บุใน (padding) หนา 10px

Element Name Selector

เป็นการเขียนสไตล์ชีทเพื่อใช้กับแท็กชื่อใดชื่อหนึ่งหรือหลายแท็ก ถ้าต้องการใช้การหลายแท็ก ให้เครื่องหมาย comma (,) คั่นชื่อแท็กแต่ละตัว

รูปแบบ

```
E1,E2,E3, ... En {
  property: value;
}
```

ตัวอย่าง

```
h1,h2,h3, p{
  color: blue;
}
```

E ย่อมาจาก Element หลายถึงแท็กรวมถึง content ของมัน

ตัวอย่างการใช้ Element Name Selector

```
<html>
<head>
<style>
  p, li{color: blue; font-style: italic; font-weight: bold}
</style>
<body>
<p>Select the following choices:</p>
<ol>
  <li>first choice</li>
  <li>second choice</li>
  <li>third choice</li>
</ol>
</body>
```

Class Selector

เป็นการกำหนด selector เพื่อเลือกใช้กับ*กลุ่ม*ของแท็กที่ต้องการให้มีสไตล์เดียวกันโดยไม่เจาะจงชื่อแท็ก

เช่นใน CSS

```
.cname{ font-style : italic;}
```

ใน HTML

```
<h3 class="cname">head 1</h3>
<p>ทดสอบ 1</p>
<h3>head 2</h3>
<p class="cname">ทดสอบ 2</p>
```

สไตล์ที่เขียนจะมีผลกับ <h3> ตัวแรก และ <p> ตัวที่สองซึ่งมีการระบุ ชื่อคลาสตรงกัน

ตัวอย่างการใช้ Class Selector

```
<style>
  .bold {font-weight: bold}
  .italic {font-style: italic}
</style>
```

นำสไตล์ไปใช้กับ tag ต่าง ๆ ด้วยการระบุชื่อของคลาส

```
<p class="bold">
  BOLD class</p>
<li class="italic">
  Statement ITALIC class</li>
```

Id Selector

เป็นการกำหนด selector เพื่อเลือกใช้กับแท็กตัวใดตัวหนึ่งเพียงตัวเดียวที่มีค่าของ attribute `id` ตรงตามที่กำหนด เช่น

ใน CSS

```
#iname{ font-weight :bold;}
```

ใน HTML

```
<p id="iname">ทดสอบ 1</p>
<p>ทดสอบ 2</p>
```

จะเลือกใช้กับ `<p>` ตัวแรกซึ่งมีค่าของ attribute `id` เป็น `iname` เท่านั้น

Descendant Selectors

เป็นการเลือกใช้สไตล์ให้กับแท็กที่อยู่ชั้นในลึกลงไปกี่ชั้นก็ได้ เช่น

```
div p {
  background-color:#ccbbaa;
}
```

เมื่อนำไปใช้กับ html ต่อไปนี้

```
<div>
  <p>I am a descendant</p>
  <form>
    <p>I am a descendant too</p>
  </form>
</div>
```

สไตล์ที่เขียนด้านบนจะมีผลกับแท็ก `<p>` ทุกตัวที่บรรจุอยู่ใน `<div>`

Child Selectors

เป็นการเลือกใช้สไตล์ให้กับแท็กลูก (child node) ที่อยู่ชั้นในลึกลงไปหนึ่งชั้น เช่น

```
div > p {
  background-color:#ccbbaa;
}
```

เมื่อนำไปใช้กับ html ต่อไปนี้

```
<div>
  <p>I am a child</p>
  <form>
    <p>I am a descendant but not a child</p>
  </form>
</div>
```

สไตล์ที่เขียนด้านบนจะมีผลกับแท็ก `<p>` ตัวแรกเท่านั้น

Attribute Selector

เลือกแท็กที่มีค่าของ attribute ตรงกับค่าที่กำหนด เช่น

```
img[alt=spacer] {
  padding:0px;
}
```

จะเลือกใช้กับแท็ก `` ที่มีการระบุ attribute `alt` เป็น `spacer`

```

```

หมายเหตุ selector ยังมีอีกหลายรูปแบบ ดูภาพรวมบางส่วนในสไลด์ถัดไป

Selectors & cascading

49

Pattern	Meaning
*	Universal selector: matches any element.
E	Type selector: matches any E element (i.e., an element of type E; e.g. H1 or P).
E F	Descendant selector: matches any F element that is a descendant of an E element.
E > F	Child selector: matches any F element that is a child of an element E.
E + F	Adjacent siblings selector: Matches any F element immediately preceded by an element E.
E[foo]	Attribute selector: matches any E element with the "foo" attribute set (whatever the value).
E[foo="warning"]	Attribute selector: matches any E element whose "foo" attribute value is exactly equal to "warning".
E[foo~="warning"]	Attribute selector: matches any E element whose "foo" attribute value is a list of space-separated values, one of which is exactly equal to "warning".
E[lang ="en"]	Attribute selector: matches any E element whose "lang" attribute has a hyphen-separated list of values beginning (from the left) with "en" (e.g. en-US).
DIV.warning	<i>HTML only.</i> The same as DIV[class~="warning"].
E#myid	ID selector: matches any E element ID equal to "myid".
E:lang(c)	Pseudo-class selector: matches element of type E if it is in (human) language c (the document language specifies how language is determined).
E:first-child	Pseudo-class selector: matches element E when E is the first child of its parent.
E:link, E:visited	Pseudo-class selector: matches element E if E is the source anchor of a hyperlink of which the target is not yet visited (:link) or already visited (:visited).
E:active, E:hover, E:focus	Dynamic Pseudo-class selector: matches E during certain user actions.
E:first-line, E:first-letter	Pseudo-element selector: matches the first formatted line or letter of element E.

Pseudo-classes

50

ใช้ในกรณีที่แท็กของ HTML สามารถแสดงผลได้มากกว่าหนึ่งพฤติกรรม เช่นกรณีของ hyperlinks

- CSS:

```
a:link {color: black}
a:active {color: red}
a:visited {color: yellow}
a:hover {color: blue}
```

- HTML

```
<a href="nextpage.html">Next page</a>
```

Style Sheet Equivalent for <body>

51

```
body {
    color: black;
    background-color: green;
    background-image: url(image.gif);
}
a:link {color: blue}
a:active {color: red}
a:visited {color: yellow}
```

เทียบเท่ากับ

```
<body text="black" link="blue"
alink="red" vlink="yellow"
bgcolor="green"
background="image.gif" >
```

ดูตัวอย่างที่ <http://www.bus.tu.ac.th/usr/wanchai/html/css/bodystyle.html>

การยกเลิกเส้นใต้ข้อความที่เป็นลิงค์

52

ใส่แท็ก <style> ต่อไปนี้ไว้ก่อน </head>

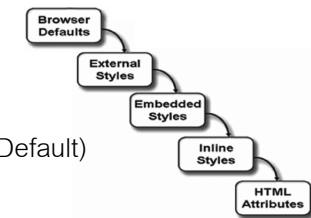
```
<style>
a:link {text-decoration: none}
a:visited {text-decoration: none}
a:active {text-decoration: none}
a:hover {text-decoration: none}
</style>
```

การรับทอด (inherit) สมบัติ

กรณีที่ใช้ style sheet หลายแบบผสมกัน และมีการกำหนดตัวเลือก (selector) ซ้ำกัน จะมีการนำ property จากทุก style sheet มาใช้ร่วมกัน แต่ถ้ามีการกำหนด property ซ้ำกันจะเลือกใช้ค่าของ property ตามลำดับการทำงาน

ลำดับการทำงานของ style sheet แต่ละแบบ

- กรณีที่ใช้ style sheet หลายแบบผสมกัน จะส่งต่อ property ตามลำดับการตกทอดเป็นชั้น ๆ เหมือนน้ำตก (cascade) ดังรูปด้านล่าง
- แต่ถ้ามีการระบุ property เดียวกันในหลายชั้น จะใช้ค่าของ property ตามลำดับก่อนหลัง (precedence) ดังนี้ (เลขลำดับน้อยมีสิทธิเหนือกว่าเลขลำดับมาก)
 1. Local Style Sheet
 2. Global Style Sheet
 3. External Style Sheet
 4. ค่า Style ที่กำหนดโดยเว็บเบราว์เซอร์ (Browser Default)



Tip

ยัดค่าที่อยู่ใกล้ตัวที่สุดเป็นลำดับ

ตัวอย่างการรับทอด

External Stylesheet เชฟเป็นไฟล์ชื่อ exstyle.css

```
/* external stylesheet exstyle.css */
h3{ color: red; text-align: left; font-size: 8pt }
```

HTML มี global stylesheet และยังมีกรลิงค์กับ External Stylesheet ซ้ำกัน

```
:
<link rel="stylesheet" type="text/css" href="exstyle.css">
<style>
  h3 {text-align: right; font-size: 20pt}
</style>
:
```

selector h3 มีการระบุ property **text-align** และ **font-size** ซ้ำกันใน 2 ลำดับชั้น (แสดงด้วยสีแดง) ผลที่ได้แก่ h3 จะมีค่าของสมบัติต่าง ๆ ดังนี้

```
color: red; text-align: right; font-size: 20pt
```

↑ เฉพาะ color เท่านั้นที่รับทอดมาจาก external stylesheet ส่วนตัวอื่นใช้ของ global stylesheet