

CSS

CSS คืออะไร

CSS ย่อมาจาก Cascading Style Sheets คือ ภาษาที่ใช้สำหรับตกแต่งเอกสาร HTML/XHTML ให้มีหน้าตา สีสัน ระยะห่าง พื้นหลัง เส้นขอบและอื่นๆ ตามที่ต้องการ มีลักษณะเป็นภาษาที่มีรูปแบบในการเขียน Syntax แบบเฉพาะ ถูกกำหนดมาตรฐานโดย W3C เป็นภาษาหนึ่งในการตกแต่งเว็บไซต์ ที่ได้รับความนิยมอย่างแพร่หลาย

ประโยชน์ของ CSS

CSS มีประโยชน์อย่างหลากหลาย ดังนี้

1. ช่วยให้เอกสาร HTML มีความเข้าใจได้ง่ายขึ้น และการแก้ไขเอกสารก็สามารถทำได้ง่ายขึ้น เพราะการใช้ CSS จะช่วยลดการใช้ภาษา HTML และแยก
ระหว่างเนื้อหา กับรูปแบบในการแสดงผลออกจากกัน
2. ทำให้สามารถดาวน์โหลดไฟล์ได้เร็ว เนื่องจากโค้ดในเอกสาร HTML ลดลง ทำให้ไฟล์มีขนาดเล็กลง
3. สามารถกำหนดรูปแบบการแสดงผลจากคำสั่ง style sheet ชุดเดียว ให้มีการแสดงผลในเอกสารแบบเดียวกันทั้งหน้าหรือในทุกๆ หน้าได้ ช่วยลดเวลาในการ
ปรับปรุงและทำให้การสร้างเอกสารบนเว็บมีความรวดเร็วมากขึ้น

4. ช่วยในการกำหนดรูปแบบการแสดงผลให้มีความเหมาะสมกับสื่อต่างๆ ได้ดียิ่งขึ้น
5. ทำให้เว็บไซต์มีความสวยงาม เป็นมาตรฐานมากขึ้น และมีความทันสมัย สามารถรองรับการใช้งานในอนาคตได้ดี

โครงสร้างพื้นฐานของ CSS

โครงสร้างของ CSS จะมี 2 ส่วนหลักๆ ดังนี้

Selector: ใช้ระบุแท็ก หรือเอลิเมนต์บนเว็บเพจที่ต้องการปรับแต่ง เช่น <p>, <div> หรือจะระบุตาม id หรือ class ที่ต้องการก็ได้

Declaration: คือส่วนที่ใช้เขียนโค้ดเพื่อปรับแต่งแก้ไขสไตล์ (Style) เช่น เปลี่ยนสีตัวอักษร เปลี่ยนสีพื้นหลัง กำหนดระยะห่าง ระยะขอบต่างๆ เป็นต้น โดยโค้ดในส่วนนี้จะเขียนอยู่ในเครื่องหมาย { } ประกอบด้วยคุณสมบัติ (Property) และค่า (Value) ที่ต้องการปรับแต่ง เช่น หากต้องการกำหนดสีของตัวอักษรเป็นสีน้ำเงินจะเขียนโค้ดเป็น color: blue; ดังตัวอย่างต่อไปนี้

HTML Code	Output
<pre><!DOCTYPE html> <html> <head> <meta charset="utf-8"> <title>CSS</title> <style> h2 { color: blue; } </style> </head> <body> <h2>ข้อความที่กำหนดสีด้วย CSS</h2> </body> </html></pre>	<p>ข้อความที่กำหนดสีด้วย CSS</p>

จากตัวอย่างนี้ Selector คือ h2, Property คือ color และ value คือ blue

การคอมเมนต์ใน CSS

คอมเมนต์ คือ การเขียนข้อความเพื่ออธิบายโค้ด โดยที่ browser จะไม่แสดงข้อความ หรือโค้ดที่คอมเมนต์ไว้ออกมาที่หน้าเว็บเพจ การคอมเมนต์ใน CSS จะเขียนอยู่ในเครื่องหมาย /* .../ ดังตัวอย่างต่อไปนี้

HTML Code	Output
<pre><!DOCTYPE html> <html> <head> <meta charset="utf-8"> <title>CSS</title> <style> /* กำหนดสีของ h2 เป็นสีน้ำเงิน */ h2 { color: blue; } </style> </head> <body> <h2>ข้อความที่กำหนดสีด้วย CSS</h2> </body> </html></pre>	<p>ข้อความที่กำหนดสีด้วย CSS</p>

กรณีต้องการคอมเมนต์หลายๆ บรรทัด ก็สามารถทำได้ เช่น

```
/*
CSS สำหรับเว็บ winwin.co.th
Create by MIS-BOY, 2023-03-02
*/
```

การใช้ CSS กับ HTML

การใช้ CSS กับ HTML สามารถทำได้ 3 วิธี ดังนี้

วิธีที่ 1: Internal Style เป็นการเขียนโค้ด CSS แทรกในไฟล์ HTML โดยใช้แท็ก <style> ใส่ไว้ในส่วนของ <head> ดังตัวอย่างก่อนหน้า

วิธีที่ 2: External Style เป็นการแยกโค้ด CSS เป็นไฟล์นามสกุล .css แล้วใส่ link เข้ามาในเอกสาร HTML ดังตัวอย่างนี้

HTML Code	Output
<pre><!DOCTYPE html> <html> <head> <meta charset="utf-8"> <title>CSS</title> <link type="text/css" rel="stylesheet" href="style.css"> </head> <body> <h2>External Style CSS</h2> </body> </html></pre>	<p>External Style CSS</p>
style.css	
<pre>h2 { font-size: 16pt; color: green; text-align: center; }</pre>	

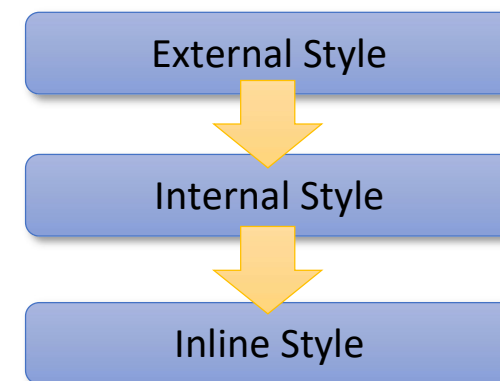
วิธีที่ 3: Inline Style เป็นการเขียนโค้ด CSS ให้กับแท็กที่ต้องการกำหนดรูปแบบ ตัวอย่างเช่น

HTML Code	Output
<pre><!DOCTYPE html> <html> <head> <meta charset="utf-8"> <title>CSS</title> </head> <body> <p style="color:red; font-size:20pt;">Inline Style CSS</p> </body> </html></pre>	Inline Style CSS

ลำดับการใช้สไตล์

การเรียกใช้สไตล์ มีลำดับที่แน่นอน คือ จะใช้โค้ด CSS จาก External Style ก่อน จากนั้นจะเรียกใช้ Internal Style และสุดท้ายถึงจะเรียกใช้ Inline Style ดังรูป

หากมีการเรียกใช้สไตล์ซ้ำกันหลายแบบ Browser จะใช้สไตล์ที่มีการเรียกใช้หลังสุด เช่น หากโค้ดในไฟล์ css (External Style) กำหนดให้แท็ก <p> เป็นตัวอักษรสีเหลือง แต่โค้ดในแท็ก <style> (Internal Style) กำหนดให้แท็ก <p> เป็นตัวอักษรสีแดง ตัวอักษรที่แสดงใน Browser จะเป็นสีแดง แต่ถ้ามีการกำหนดแอตทริบิวต์ style ในแท็ก <p> (Inline style) ให้ตัวอักษรเป็นสีน้ำเงิน ตัวอักษรในแท็ก <p> ที่สีแดงจะเป็นสีน้ำเงิน



การกำหนดค่าคุณสมบัติใน CSS

ค่าคุณสมบัติที่จะกำหนดให้แบแท็ก HTML แต่ละตัว มีความหลากหลาย ดังนั้น จึงทำให้มีหน่วยนับค่าคุณสมบัติที่แตกต่างกันไป โดยการกำหนดค่าคุณสมบัติแบ่งออกได้ 3 รูปแบบ ดังนี้

ค่าคุณสมบัติเพื่อระบุขนาด

การกำหนดขนาดให้คุณสมบัติ (Property) ใดๆ ที่เกี่ยวกับตัวอักษร ความกว้าง ความสูง ระยะห่างระหว่างบรรทัด ระยะห่างจากขอบ ฯลฯ ค่าเหล่านี้มีหน่วยวัดที่สามารถใช้กำหนดค่าให้กับคุณสมบัติได้ ดังตารางต่อไปนี้

หน่วยวัด	คำอธิบาย
%	ใช้กำหนดขนาดความกว้าง หรือความสูงเป็นเปอร์เซ็นต์ เช่น กำหนดให้แท็ก มีความกว้าง 100% เป็นต้น
in	นิ้ว
cm	เซนติเมตร
mm	มิลลิเมตร
em	ใช้กำหนดขนาดตัวอักษร โดย 1 em มีค่าเท่ากับขนาดปกติของฟอนต์นั้น หากกำหนดให้ขนาดตัวอักษรมีขนาด 2 em ก็หมายความว่า มีขนาดเป็น 2 เท่าของฟอนต์ปกติ
ex	ใช้กำหนดขนาดตัวอักษร โดย 1 ex มีค่าเท่ากับครึ่งหนึ่งของฟอนต์ปกติ
pt	Point โดยที่ 1 point เท่ากับ 1/72 นิ้ว
pc	Pica โดยที่ 1 pc เท่ากับ 12 pt
px	Pixel

ค่าคุณสมบัติเพื่อระบุฟอนต์

ใน CSS เราสามารถกำหนดฟอนต์และรูปแบบการแสดงผลตัวอักษรที่ต้องการได้ โดยกำหนดผ่านคุณสมบัติดังนี้

คุณสมบัติ	คำอธิบาย
font-family	กำหนดฟอนต์ที่ใช้แสดงผลภายในแท็ก HTML โดยกำหนดได้ทั้งชื่อฟอนต์ หรือกลุ่มของฟอนต์ เช่น <code>P { font-family: Tahoma, Geneva, sans-serif; }</code>
font-style	กำหนดรูปแบบการแสดงผลของตัวอักษร โดยสามารถกำหนดได้ 3 รูปแบบ <ol style="list-style-type: none">1. normal: ตัวอักษรปกติ2. italic: ตัวเอียง3. oblique: เหมือนกับ italic แต่ใช้สำหรับฟอนต์ในกลุ่ม Sans-Serif เท่านั้น
Font-size	กำหนดขนาดตัวอักษร เช่น <code>P { font-size: 16pt; }</code>
Font-weight	กำหนดน้ำหนักหรือความหนาของฟอนต์ โดยสามารถกำหนดเป็นจำนวนตั้งแต่ 100, 200, ... 900 หรือจะกำหนดเป็น Keyword ก็ได้ เช่น <code>P { font-weight: 700; }</code> หรือ <code>P { font-weight: bold; }</code>

ค่าคุณสมบัติเพื่อระบุสี

แท็กของ HTML บางแท็กจะมีคุณสมบัติ color หรือ background-color สำหรับระบุสีของตัวอักษรหรือพื้นหลังได้ การระบุสี จะระบุได้ 3 วิธี ดังนี้

1. **ระบุเป็นชื่อสี** เช่น `body { background-color: yellow; }` โดยจะมีทั้งหมด 17 สีที่ Browser ทุกตัวรองรับ ได้แก่ aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white และ yellow
2. **ระบุเป็นรหัสสี** การระบุความเข้มของสีแดง เขียว น้ำเงิน แทนด้วยตัวเลขฐาน 16 ตั้งแต่ 00 ถึง FF ผสมกันจะได้สีทั้งหมด 16 ล้านสี เช่น `p { color: #2255BB; }` โดยที่ 22 คือความเข้มของสีแดง, 55 คือความเข้มของสีเขียว และ BB คือความเข้มของสีน้ำเงิน
3. **ระบุเป็นรหัสสีด้วยคำสั่ง rgb** เป็นการระบุสีโดยการผสมสีแดง เขียว และน้ำเงินเช่นกัน แต่จะใช้เลขฐาน 10 ตั้งแต่ 0 ถึง 255 ผสมกัน เช่น `p { color: rgb(30, 60, 200); }`


การใช้ Selector เบื้องต้น

Selector ใช้สำหรับการระบุหรือเลือก HTML Element ที่ต้องการกำหนดคุณสมบัติ Selector ทั่วไปจะมีดังนี้

1. **Tag Selector** เป็น Selector ที่อ้างอิงตามชื่อแท็ก (Tag) ของ HTML เช่น <p>, <div>, <h1>, <table> เป็นต้น การเขียน Selector แบบนี้จะระบุชื่อ HTML แท็กที่ต้องการกำหนดคุณสมบัติในโค้ด CSS ได้เลย ดังตัวอย่างนี้

HTML Code	Output
<pre><!DOCTYPE html> <html> <head> <meta charset="utf-8"> <style> p { color: #ff0000; font-family: Tahoma; font-size: 20px; text-align: center; margin-top: 40px; } </style> </head> <body> <p>Tag Selector</p> <p>Tag Selector</p> </body> </html></pre>	<p>Tag Selector</p> <p>Tag Selector</p>

2. ID Selector เป็น Selector ที่อ้างอิงตามแอททริบิวต์ ID ของ HTML แท็ก การเขียน Selector แบบนี้จะระบุ ID ของ HTML ที่ต้องการกำหนดคุณสมบัติในโค้ด CSS โดยใช้เครื่องหมาย # หน้า ID ที่ต้องการ ดังตัวอย่างนี้

HTML Code	Output
<pre><!DOCTYPE html> <html> <head> <meta charset="utf-8"> <style> #title { color: #3333DD; font-size: 2em; font-weight: bold; margin-top: 40px; } </style> </head> <body> <p id="title">ID Selector</p> <p>ID Selector</p> </body> </html></pre>	

จากตัวอย่างนี้ ใน <body> มีแท็ก <p> อยู่ 2 ตัว ตัวแรกกำหนด ID เป็น “title” แต่ตัวที่ 2 ไม่ได้กำหนด ID ทำให้ข้อความในแท็ก <p> ตัวที่สอง ไม่ได้รับคุณสมบัติของสไตล์ที่กำหนด


3. **Class Selector** เป็น selector ที่อ้างอิงตาม Class ของ HTML แท็ก การเขียน Selector แบบนี้จะได้ . แล้วตามด้วยชื่อคลาสที่ต้องการกำหนดคุณสมบัติดังตัวอย่างต่อไปนี้

HTML Code	Output
<pre><!DOCTYPE html> <html> <head> <meta charset="utf-8"> <style> .text1 { color: red; background-color: black; padding: 10px; } .text2{ color: blue; background-color: silver; padding: 20px; } </style> </head> <body> <p>Class Selector</p> <p class="text1">Class Selector</p> <p class="text2">Class Selector</p> </body> </html></pre>	<p>Class Selector</p> <div><p>Class Selector</p></div> <div><p>Class Selector</p></div>

4. Group Selector เป็นการเลือก Element หลายๆ ตัว ที่ต้องการกำหนดคุณสมบัติเหมือนกัน การเขียน Selector แบบนี้จะใช้เครื่องหมาย , คั่นระหว่าง Select แต่ละตัว ดังตัวอย่างต่อไปนี้

HTML Code	Output
<pre><!DOCTYPE html> <html> <head> <meta charset="utf-8"> <style> p, #title, .text1 { color: blue; font-size: 1.5em; background-color: pink; margin: 15px; } </style> </head> <body> Group Selector
 <h3>Group Selector</h3> <p>Group Selector</p> <div id="title">Group Selector</div> Group Selector </body> </html></pre>	<p>Group Selector</p> <p>Group Selector</p> <p>Group Selector</p> <p>Group Selector</p> <p>Group Selector</p>

5. Universal Selector เป็นการกำหนดคุณสมบัติให้กับ Element ทุกตัวในเว็บ การเขียน Selector แบบนี้จะใช้เครื่องหมาย * ดังตัวอย่างต่อไปนี้

HTML Code	Output
<pre><!DOCTYPE html> <html> <head> <style> * { text-align: center; color: blue; } </style> </head> <body> <h2>Universal Selector</h2> <p id="p1">รายชื่อโครงการอบรม</p> <p>Full Stack Programmer รุ่นที่ 3</p> <p>iOS Application รุ่นที่ 3</p> <p>Android Application รุ่นที่ 2</p> </body> </html></pre>	<p style="text-align: center;">Universal Selector</p> <p style="text-align: center;">รายชื่อโครงการอบรม</p> <p style="text-align: center;">Full Stack Programmer รุ่นที่ 3</p> <p style="text-align: center;">iOS Application รุ่นที่ 3</p> <p style="text-align: center;">Android Application รุ่นที่ 2</p> <div style="text-align: center;"></div>

จากตัวอย่างนี้ จะเห็นว่า ข้อความในแท็ก <h2> และในแท็ก <p> จะเป็นสีน้ำเงิน และจัดอยู่กึ่งกลางทุกตัว

6. **Specific Selector** เป็นการเลือก Element ที่เฉพาะเจาะจงว่าเป็น Element ตัวไหน เช่น ในเพจมีการใช้แท็ก <p> หลายตัว การกำหนดสไตล์แบบ Tag Selector จะมีผลกับแท็ก <p> ทั้งหมดในเพจ กรณีที่เราต้องการให้แท็ก <p> บางตัวมีคุณสมบัติที่แตกต่าง จะใช้ Specific Selector ช่วย ดังตัวอย่างต่อไปนี้

HTML Code	Output
<pre><!DOCTYPE html> <html> <head> <style> p { text-align: center; color: blue; } p#p1{ color: red; font-size: 1.5em; } </style> </head> <body> <h2>Specific Selector</h2> <p id="p1">รายชื่อโครงการอบรม</p> <p>Full Stack Programmer รุ่นที่ 3</p> <p>iOS Application รุ่นที่ 3</p> <p>Android Application รุ่นที่ 2</p> </body> </html></pre>	<p>Specific Selector</p> <p>รายชื่อโครงการอบรม</p> <p>Full Stack Programmer รุ่นที่ 3</p> <p>iOS Application รุ่นที่ 3</p> <p>Android Application รุ่นที่ 2</p> 

จากตัวอย่างนี้เป็นการกำหนดให้แท็ก <p> ที่ใช้ ID เป็น “p1” ให้มีตัวอักษรสีแดง ขนาดใหญ่กว่าปกติ 0.5 เท่า นอกจากการกำหนดตาม ID แล้ว เรายังสามารถกำหนดตาม Class ก็ได้เหมือนกัน เช่น p.title { color: green } จะเป็นการกำหนดให้แท็ก <p> ที่ใช้ Class “title” มีตัวอักษรสีเขียว

7. Descendent Selector เป็นการเลือก Element โดยอ้างอิงจากโครงสร้างของการจัดเรียง Element บนเว็บเพจ ดังตัวอย่างต่อไปนี้

HTML Code	Output
<pre><!DOCTYPE html> <html> <head> <style> div p { color: red; background-color: silver; padding: 10px; } </style> </head> <body> <p>Descendent Selector</p> <div> <p>Descendent Selector</p> </div> <div> <p>Descendent Selector</p> </div> </body> </html></pre>	<p>Descendent Selector</p> <div>Descendent Selector</div> <div>Descendent Selector</div>

จากตัวอย่างนี้ แท็ก <p> ตัวแรก เป็นสีดำปกติ เพราะ ไม่ได้อยู่ในแท็ก <div> แท็ก <p> ตัวที่ 2 และ 3 จะได้รับคุณสมบัติตามสไตล์ div p ที่กำหนด เพราะ อยู่ในแท็ก <div> ถึงแม้แท็ก <p> ตัวที่ 3 จะมีแท็ก ก็ยังอยู่ก็ตาม

8. **Child Selector** เป็นการเลือก Element คล้ายกับ Descendent Selector คือมีการอ้างอิงจากโครงสร้างของการจัดเรียง Element บนเว็บเพจ แต่ Child Select จะเป็นการเลือกเฉพาะ Child Element ในลำดับที่ติดกับ Parent Element เท่านั้น การเขียน Selector แบบนี้จะใช้เครื่องหมาย > ดังตัวอย่างต่อไปนี้

HTML Code	Output
<pre><!DOCTYPE html> <html> <head> <style> div>p { color: red; background-color: silver; padding: 10px; } </style> </head> <body> <p>Child Selector</p> <div> <p>Child Selector</p> </div> <div> <p>Child Selector</p> </div> </body> </html></pre>	<p>Child Selector</p> <div>Child Selector</div> <p>Child Selector</p>

จากตัวอย่างจะเห็นได้ว่า โครงสร้างการจัดเรียง Element บนเว็บเพจ คล้ายกับตัวอย่างของ Descendent Selector แต่ใช้ Selector แบบ Child Selector แทน ส่งผลให้แท็ก <p> ตัวที่ 3 ไม่ได้รับคุณสมบัติของสไตล์ที่กำหนด เพราะแท็ก <p> ไม่ได้อยู่ลำดับที่ติดกับแท็ก <div>

การจัดตำแหน่งข้อความแต่ละย่อหน้าด้วย CSS

หากต้องการจัดตำแหน่งข้อความแต่ละย่อหน้า สามารถใช้คุณสมบัติ text-align ของ CSS จัดได้ ซึ่งจะมีค่าที่สามารถกำหนดได้ 4 อย่าง คือ left, center, right และ justify ดังตัวอย่างต่อไปนี้

HTML Code	Output
<pre><!DOCTYPE html> <html> <head> <title>การจัดตำแหน่งข้อความในแต่ละย่อหน้าด้วย CSS</title> <style> .ta-left { text-align: left; } .ta-center { text-align: center; } .ta-right { text-align: right; } .ta-justify { text-align: justify; } </style> </head> <body> <p class="ta-left">บริษัท วินวิน อินเตอร์แอคทีฟ จำกัด</p> <p class="ta-center">บริษัท วินวิน อินเตอร์แอคทีฟ จำกัด</p> <p class="ta-right">บริษัท วินวิน อินเตอร์แอคทีฟ จำกัด</p> <p class="ta-justify">บริษัท วินวิน อินเตอร์แอคทีฟ จำกัด เปิดโอกาส ให้กับทุกคนที่สนใจร่วมงานด้านเกมออนไลน์ในตำแหน่งต่างๆ จำนวนมาก เหมาะสำหรับผู้ที่ต้องการความก้าวหน้าในตำแหน่งงาน และต้องการทำงานในบริษัทเกมออนไลน์ที่มั่นคง ขอเชิญผู้มีความสามารถทุกท่านมาร่วมเป็นส่วนหนึ่งกับเรา</p> </body> </html></pre>	<p>บริษัท วินวิน อินเตอร์แอคทีฟ จำกัด</p> <p>บริษัท วินวิน อินเตอร์แอคทีฟ จำกัด</p> <p>บริษัท วินวิน อินเตอร์แอคทีฟ จำกัด</p> <p>บริษัท วินวิน อินเตอร์แอคทีฟ จำกัด เปิดโอกาส ให้กับทุกคนที่สนใจร่วมงานด้านเกมออนไลน์ในตำแหน่งต่างๆ จำนวนมาก เหมาะสำหรับผู้ที่ต้องการความก้าวหน้าในตำแหน่งงาน และต้องการทำงานในบริษัทเกมออนไลน์ที่มั่นคง ขอเชิญผู้มีความสามารถทุกท่านมาร่วมเป็นส่วนหนึ่งกับเรา</p>


การกำหนดระยะเยื้องของย่อหน้า

เพื่อให้ข้อความในเพจดูง่ายขึ้น ในบรรทัดแรกของแต่ละย่อหน้า มักจะมีการเยื้องไปทางด้านขวาเล็กน้อย เรียกว่า indent ลักษณะนี้สามารถใช้คุณสมบัติ text-indent ช่วยเพิ่มระยะเยื้องของแต่ละย่อหน้าได้ ดังตัวอย่างต่อไปนี้

HTML Code	Output
<pre><html> <head> <title>การกำหนดระยะเยื้องของย่อหน้า</title> <style> p { text-indent: 2em; } </style> </head> <body> <h3>X-SHOT</h3> <p>X-SHOT เป็นเกมแนว FPS (First Person Shooting) รูปแบบ Fantasy ที่ทำให้ผู้เล่นที่ได้ลองสามารถเลือกได้ว่า อยากที่จะเป็นแบบไหน อยากเป็นทหาร ดัน ดุเดือด หรือว่า อยากเป็นเด็ก Hip Hop สุดป๊อบ ก็ได้</p> <p>เกมของ Xshot มีความแตกต่าง ๆ ไม่เหมือนกับ FPS อื่น ๆ ซึ่งโดดเด่นมากในเรื่องของระบบเสียง ที่เป็นเกมแรกในโลกที่ใช้ระบบเสียง Dolby Surround เพื่อให้ตัวเกมดูสมจริงสมจังและผู้เล่นสามารถใช้ไมค์ทำการสนทนากับผู้เล่นคนอื่น ๆ ได้ซึ่งจะทำให้ผู้เล่นที่เล่นอยู่นั้นอินไปกับเกมได้</p> </body> </html></pre>	<p>X-SHOT</p> <p>X-SHOT เป็นเกมแนว FPS (First Person Shooting) รูปแบบ Fantasy ที่ทำให้ผู้เล่นที่ได้ลองสามารถเลือกได้ว่า อยากที่จะเป็นแบบไหน อยากเป็นทหาร ดัน ดุเดือด หรือว่า อยากเป็นเด็ก Hip Hop สุดป๊อบ ก็ได้</p> <p>เกมของ Xshot มีความแตกต่าง ๆ ไม่เหมือนกับ FPS อื่น ๆ ซึ่งโดดเด่นมากในเรื่องของระบบเสียง ที่เป็นเกมแรกในโลกที่ใช้ระบบเสียง Dolby Surround เพื่อให้ตัวเกมดูสมจริงสมจังและผู้เล่นสามารถใช้ไมค์ทำการสนทนากับผู้เล่นคนอื่น ๆ ได้ซึ่งจะทำให้ผู้เล่นที่เล่นอยู่นั้นอินไปกับเกมได้</p>

การใส่ภาพพื้นหลังด้วย CSS

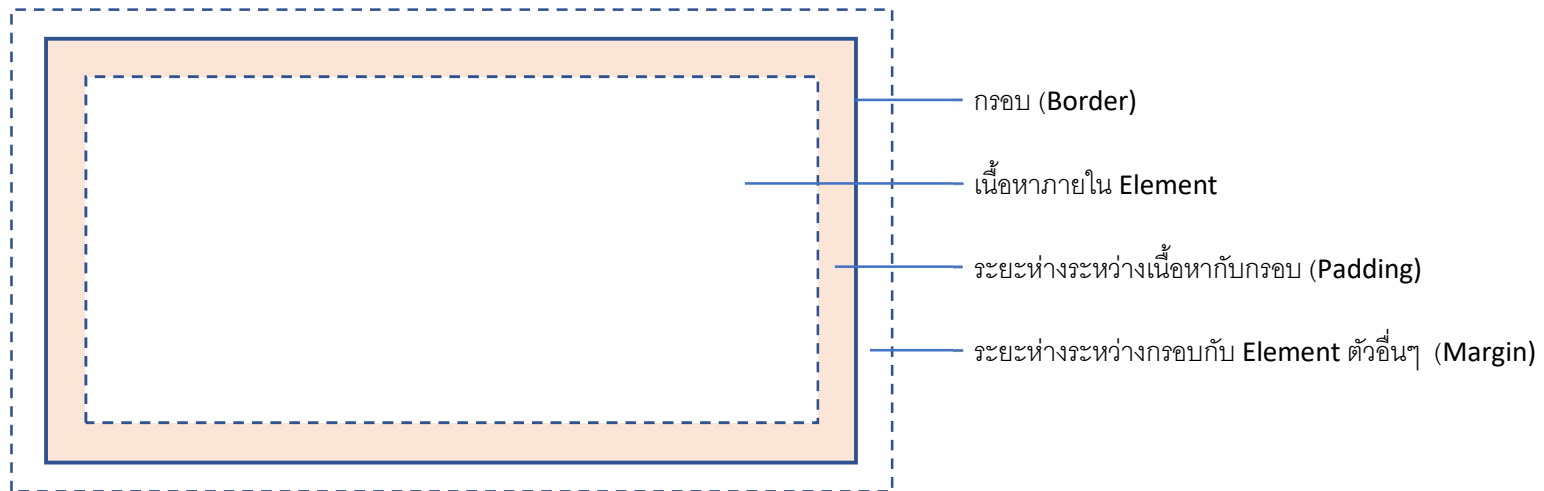
หากต้องการใส่ภาพพื้นหลังให้กับเว็บเพจ สามารถกำหนดได้โดยใช้คุณสมบัติ background ของแท็ก body ดังตัวอย่างต่อไปนี้

HTML Code	Output
<pre><!DOCTYPE html> <html> <head> <meta charset="utf-8"> <style> body { background: url(images/bg1.jpg) no-repeat; } p { text-indent: 3em; } * { color: #ffffff; } </style> </head> <body> <h2>โครงการอบรม</h2> <hr> <p>เริ่มต้นปีนี้ด้วยโครงการอบรมดีๆ ถึง 3 โครงการ อันได้แก่ Full Stack Programmer, iOS Application และ Android Application อบรมฟรี มีเงิน สนับสนุนให้ และที่สำคัญมีงานรองรับ ไม่จำเป็นต้องมีพื้นฐาน เพียงแค่สนใจ ในสายงานนี้ก็สมัครเข้ามาได้ บริษัท วินวิน อินเทอร์เน็ต จำกัด และบริษัท ในเครือ เปิดโอกาสให้ทุกท่านอีกเช่นเคย</p> </body> </html></pre>	


การใช้คุณสมบัติ background จะต้องกำหนด url ของภาพที่จะใช้เป็นภาพพื้นหลัง กำหนดการเรียงภาพซ้ำในแนวนอน (repeat-x) แนวตั้ง (repeat-y) กรณีกำหนดแค่ repeat จะเป็นการเรียงภาพซ้ำทั้งแนวนอน และแนวตั้ง หากไม่ต้องการให้เรียงภาพซ้ำ ให้กำหนดเป็น no-repeat เหมือนตัวอย่าง

Margin และ Padding

Margin คือ ระยะห่างระหว่างกรอบ (Border) ของ Element กับ Element ตัวอื่นๆ ส่วน **Padding** คือระยะห่างระหว่างเนื้อหาที่อยู่ภายใน Element กับกรอบของ Element ดังภาพตัวอย่าง



เพื่อให้เข้าใจได้ชัดเจนขึ้น ให้ดูจากตัวอย่างการใช้งานต่อไปนี้

HTML Code	Output
<pre><!DOCTYPE html> <html> <head> <title>Margin และ Padding</title> <style> body { background-color: #ffb; } div { margin: 30px; padding: 15px; background-color: #99f; } span { background-color: #fff; display: block; } </style> </head> <body> <div>WinWin Interactive Co.,Ltd.</div> <div>Winner Online Co.,Ltd.</div> <hr> </body> </html></pre>	

จากตัวอย่างนี้ กำหนดให้พื้นหลังของเว็บเป็นสีเหลือง กำหนดให้แท็ก <div> มี Margin 30 pixels มี Padding 15 pixels และมีสีพื้นหลังเป็นสีม่วง ในแท็ก <div> จะใส่แท็ก ที่มีสีพื้นเป็นสีขาว จะเห็นว่า ระยะห่างระหว่างแท็ก กับกรอบของแท็ก <div> จะมีขนาด 15 pixels และระยะห่างระหว่างกรอบของแท็ก <div> แต่ละตัวจะมีระยะห่างกัน 30 pixels รวมทั้งระยะห่างจากกรอบของ browser ด้วย

หากต้องการกำหนดระยะ Margin หรือ Padding แต่ละด้านไม่เท่ากัน สามารถทำได้โดยการกำหนดจำนวนระยะห่างจาก ด้านบน ด้านขวา ด้านล่าง และ ด้านซ้ายตามลำดับ เช่น `div { margin: 25px 10px 30px 15px; }`

หากต้องการกำหนดระยะ Margin หรือ Padding ที่ด้านบนและด้านล่างเท่ากัน และด้านซ้ายและด้านขวาเท่ากัน สามารถกำหนดระยะห่างแค่ 2 ตัว โดยที่ตัวแรกจะเป็นระยะห่างด้านบน และด้านล่าง ส่วนตัวที่ 2 จะเป็นระยะห่างของด้านซ้ายและด้านขวา เช่น `div { padding: 20px 15px; }`

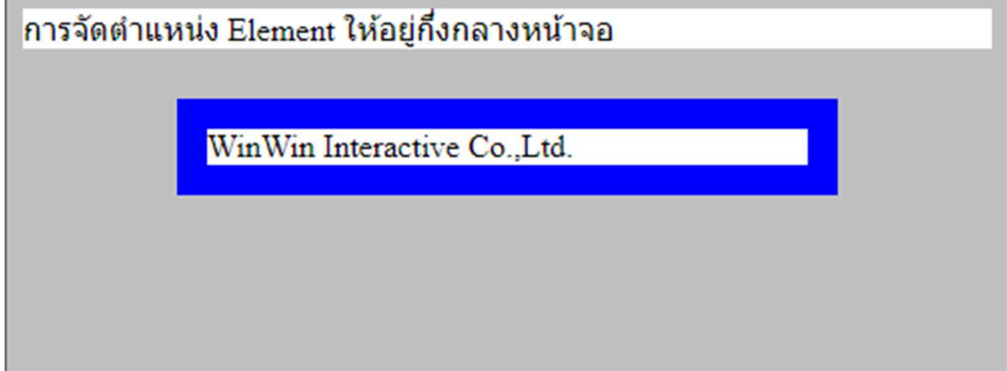
กรณีที่ต้องการกำหนดระยะ Margin หรือ Padding ที่ด้านบนกับด้านล่างไม่เท่ากัน แต่ด้านซ้ายกับด้านขวาเท่ากัน สามารถกำหนดระยะห่างแค่ 3 ตัว โดยที่ตัวแรกจะเป็นระยะห่างด้านบน ตัวที่ 2 จะเป็นระยะห่างด้านซ้ายและด้านขวา ตัวที่ 3 จะเป็นระยะห่างด้านล่าง เช่น `div { margin: 20px 15px 25px; }`

กรณีที่ต้องการกำหนดระยะ Margin หรือ Padding เฉพาะด้านใดด้านหนึ่ง จะใช้คุณสมบัติต่อไปนี้

คุณสมบัติ	คำอธิบาย
margin-top	กำหนดระยะห่างระหว่างกรอบของ Element ด้านบน
margin-right	กำหนดระยะห่างระหว่างกรอบของ Element ด้านขวา
margin-bottom	กำหนดระยะห่างระหว่างกรอบของ Element ด้านล่าง
margin-left	กำหนดระยะห่างระหว่างกรอบของ Element ด้านซ้าย
padding-top	ระยะห่างระหว่างเนื้อหาที่อยู่ภายใน Element ด้านบน
padding-right	ระยะห่างระหว่างเนื้อหาที่อยู่ภายใน Element ด้านขวา
padding-bottom	ระยะห่างระหว่างเนื้อหาที่อยู่ภายใน Element ด้านล่าง
padding-left	ระยะห่างระหว่างเนื้อหาที่อยู่ภายใน Element ด้านซ้าย

การจัดตำแหน่ง Element ให้อยู่กึ่งกลางหน้าจอ

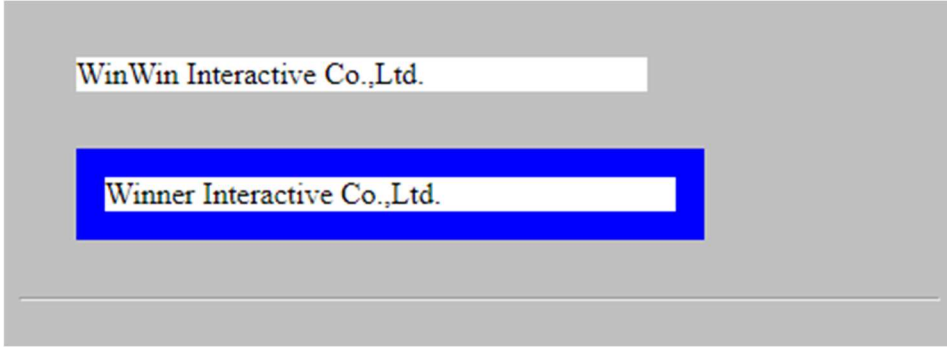
สามารถใช้ margin กำหนดได้ โดยกำหนดค่าให้เป็น auto ดังตัวอย่างนี้

HTML Code	Output
<pre><!DOCTYPE html> <html> <head> <title>Margin และ Padding</title> <style> body { background-color: silver; } div { width: 300px; margin: 25px auto; padding: 15px; background-color: blue; } span { background-color: #fff; display: block; } </style> </head> <body> การจัดตำแหน่ง Element ให้อยู่กึ่งกลางหน้าจอ <div>WinWin Interactive Co.,Ltd.</div> </body> </html></pre>	

จากตัวอย่างนี้ กำหนดให้แท็ก <div> มี margin: 25px auto; ก็คือให้มีระยะห่างจากขอบด้านบนและด้านล่าง 25 pixelsและให้อยู่กึ่งกลางหน้าจอ

ข้อควรระวังในการใช้ padding


การใช้ padding จะทำให้ขนาดของพื้นที่แสดงผลเพิ่มขึ้น เช่น ถ้าต้องการ Element ที่กว้าง 300 pixels และมีการใส่ padding เข้าไปด้วย ขนาดของ Element ที่ได้ ก็จะมากกว่า 300 pixels ดังตัวอย่างต่อไปนี้

HTML Code	Output
<pre><!DOCTYPE html> <html> <head> <title>Margin และ Padding</title> <style> body { background-color: silver; } div { width: 300px; margin: 30px; background-color: blue; } span { background-color: #fff; display: block; } .test { padding: 15px; } </style> </head> <body> <div>WinWin Interactive Co.,Ltd.</div> <div class="test">Winner Interactive Co.,Ltd.</div> <hr> </body> </html></pre>	

จากตัวอย่างนี้ มีการกำหนดความกว้างของแท็ก <div> ไว้ที่ 300 pixels และที่แท็ก <div> ตัวที่ 2 มีการใช้คลาส test ที่กำหนด padding ทุกด้าน 15 pixels ทำให้ขนาดของแท็ก <div> ตัวที่ 2 มีขนาดใหญ่กว่าตัวแรก

การจัดตำแหน่งด้วยคุณสมบัติ Float

Float ใช้สำหรับจัดตำแหน่ง Element ให้ลอยอยู่ในตำแหน่งที่ต้องการได้ เช่นอยู่ด้านซ้าย หรือด้านขวา ดังตัวอย่างต่อไปนี้

HTML Code	Output
<pre><!DOCTYPE html> <html> <head> <style> body { padding: 0 15px 15px; } p { text-align: justify; } img { float: right; margin: 5px 0 5px 15px; } </style> </head> <body> <h2>มะม่วง</h2> <hr> <p> มะม่วง (ชื่อวิทยาศาสตร์: Mangifera Indica) เป็นไม้ยืนต้นในสกุล Mangifera ซึ่งเป็นไม้ผลเมืองร้อนในวงศ์ Anacardiaceae (กลุ่มเดียวกับถั่วพิสตาชิโอและมะม่วงหิมพานต์) เชื่อว่าเป็นพืชที่มีถิ่นกำเนิดในภูมิภาคอินเดีย บังกลาเทศ และพม่าตะวันตกเฉียงเหนือ ซึ่งเห็นได้จากความหลากหลายทางพันธุกรรมและร่องรอยฟอสซิลจำนวนมาก นับย้อนไปได้ถึง 25 - 30 ล้านปีก่อน มะม่วงมีความแตกต่างประมาณ 49 สายพันธุ์กระจายอยู่ตามประเทศในเขตร้อนตั้งแต่อินเดียไปจนถึงฟิลิปปินส์ จากนั้นจึงแพร่หลายไปทั่วโลก เป็นไม้พุ่มขนาดกลาง ใบโต ยาว ปลายแหลม ขอบใบเรียบ ใบอ่อนสีแดง ออกดอกเป็นช่อตามปลายกิ่ง ดอกขนาดเล็ก สีขาว ผลอ่อนสีเขียว ผลแก่สีเหลือง เมล็ดแบน เปลือกหุ้มเมล็ดแข็ง </p> </body> </html></pre>	<div><h2>มะม่วง</h2><hr/></div> <div><p>มะม่วง (ชื่อวิทยาศาสตร์: Mangifera Indica) เป็นไม้ยืนต้นในสกุล Mangifera ซึ่งเป็นไม้ผลเมืองร้อนในวงศ์ Anacardiaceae (กลุ่มเดียวกับถั่วพิสตาชิโอและมะม่วงหิมพานต์) เชื่อว่าเป็นพืชที่มีถิ่นกำเนิดในภูมิภาคอินเดีย บังกลาเทศ และพม่าตะวันตกเฉียงเหนือ ซึ่งเห็นได้จากความหลากหลายทางพันธุกรรมและร่องรอยฟอสซิลจำนวนมาก นับย้อนไปได้ถึง 25 - 30 ล้านปีก่อน มะม่วงมีความแตกต่างประมาณ 49 สายพันธุ์กระจายอยู่ตามประเทศในเขตร้อนตั้งแต่อินเดียไปจนถึงฟิลิปปินส์ จากนั้นจึงแพร่หลายไปทั่วโลก เป็นไม้พุ่มขนาดกลาง ใบโต ยาว ปลายแหลม ขอบใบเรียบ ใบอ่อนสีแดง ออกดอกเป็นช่อตามปลายกิ่ง ดอกขนาดเล็ก สีขาว ผลอ่อนสีเขียว ผลแก่สีเหลือง เมล็ดแบน เปลือกหุ้มเมล็ดแข็ง</p></div> <div></div>