

## บทที่ 6

### การจัดรูปแบบการแสดงผลเว็บเพจด้วย CSS

ในการพัฒนาเว็บไซต์นอกจากการคำนึงถึงเนื้อหาที่มีประโยชน์ตรงตามวัตถุประสงค์ในการพัฒนาเว็บไซต์แล้ว การออกแบบเว็บไซต์ให้มีความสวยงามโดดเด่นน่าใช้ก็เป็นอีกหนึ่งปัจจัยที่จะช่วยสร้างความประทับใจที่ทำให้ผู้ใช้กลับเข้ามาใช้งานเว็บไซต์อีกครั้งในอนาคต ซึ่งในการสร้างความประทับใจนั้นสามารถทำได้หลากหลายรูปแบบ เช่น การกำหนดให้แสดงผลเว็บเพจมีความเป็นหนึ่งเดียวกันทั้งเว็บไซต์ ทำให้ผู้ใช้งานรู้สึกเสมือนว่ายังทำงานอยู่ในหน้าเว็บเพจเดิมไม่ได้มีการย้ายหน้าเว็บเพจ ถึงแม้ว่าในความเป็นจริงแล้วมีการเปลี่ยนหน้าเว็บเพจในการใช้งานแล้วก็ตาม โดยในการออกแบบเว็บเพจให้มีความเป็นหนึ่งเดียวกันนั้นสามารถกระทำได้ด้วยการใช้ CSS หรือ Cascading Style Sheet ในการควบคุมการแสดงผลเว็บเพจ และยังทำให้ขนาดไฟล์ของเว็บเพจมีขนาดเล็กลง เนื่องจากในการควบคุมการแสดงผลด้วยการกำหนด Attribute ให้กับ TAG HTML นั้นจะทำให้ไฟล์มีขนาดใหญ่ เมื่อผู้ใช้เข้าใช้งานเว็บเพจจะใช้เวลาในการโหลดไฟล์แสดงผล แต่หากมีการใช้ CSS ในการควบคุมการแสดงผลเว็บเพจ จะช่วยทำให้ลดจำนวนคำสั่งที่ใช้ควบคุมการแสดงผล ขนาดไฟล์ของเว็บเพจจะมีขนาดเล็กเมื่อเทียบกับการควบคุมการแสดงผลโดยไม่มีการใช้ CSS และเวลาในการโหลดไฟล์แสดงผลก็จะน้อยลง

#### 1. ลักษณะทั่วไปของ CSS

CSS ย่อมาจากคำว่า Cascading Style Sheet เป็นภาษาที่ใช้สำหรับจัดรูปแบบการแสดงผลของเว็บไซต์ จากเดิมในการกำหนดรูปแบบการแสดงผลจะกำหนดโดยการใช้ TAG HTML ที่ควบคุมการแสดงผลโดยเฉพาะ หรือกำหนด Attribute ของ TAG HTML โดยตรง หากไฟล์เว็บเพจนั้นมี TAG HTML จำนวนมาก ก็จะมีคำสั่งที่ใช้ควบคุมการแสดงผลมากขึ้นตามจำนวนของ TAG HTML แต่ในการใช้งาน CSS นั้นจะการแยกคำสั่งที่ใช้ควบคุมการแสดงผลของ TAG HTML ออกจากเนื้อหาที่แสดงบนเว็บเพจโดยเฉพาะและสามารถครอบคลุมถึง TAG HTML ที่อยู่บนเว็บเพจได้หลาย TAG โดยไม่จำเป็นที่จะต้องมีการใช้ TAG HTML ที่ควบคุมการแสดงผล และไม่ต้องมีการกำหนด Attribute ของ TAG HTML อีกทั้งยังสามารถปรับปรุงรูปแบบการแสดงผลของเอกสารได้สะดวกรวดเร็วมากขึ้นและนอกจากนี้เว็บเพจยังมีรูปแบบการแสดงผลที่สอดคล้องกันทั้งเว็บไซต์ เพราะสามารถกำหนดรูปแบบการแสดงผลให้กับเว็บเพจหลายๆ เว็บเพจจากคำสั่ง CSS ในไฟล์

เดียวกันได้ สามารถกำหนดรูปแบบการแสดงผลให้รองรับการทำงานได้หลายเว็บเบราว์เซอร์และหลากหลายขนาดหน้าจอของอุปกรณ์แสดงผล ซึ่ง CSS ถูกกำหนดมาตรฐานการใช้งานโดยองค์กรกลางที่เรียกว่า World Wide Web Consortium หรือ W3C

ตัวอย่างของการใช้ TAG HTML เพื่อควบคุมการแสดงผล เช่นสร้างตารางให้มีเส้นขอบกว้าง 2PX สีของเส้นขอบเป็นสีแดง ระยะห่างของเนื้อหาจากเส้นขอบภายในตาราง 30PX และในแถวแรกคอลัมน์แรกของตารางให้แสดงข้อความมีรูปแบบตัวอักษรเป็น MS Sans Serif ตัวอักษรเป็นสีน้ำเงิน และตัวหนา สามารถเขียน TAG HTML ได้ดังรูปที่ 6.1

```
<TABLE BORDER="2" BORDERCOLOR="RED" CELLPADDING="30PX" >
<TR><TD>
  <FONT FACE="MS Sans Serif " COLOR="BLUE" >
    <B>ข้อความ</B>
  </FONT>
</TD></TR>
</TABLE>
```

### รูปที่ 6.1 ตัวอย่างการใช้ TAG HTML เพื่อควบคุมการแสดงผล

จากรูปที่ 6.1 ตัวอย่างการใช้ TAG HTML เพื่อควบคุมการแสดงผล จะประกอบด้วย TAG HTML ดังนี้

- TAG <TABLE> ที่กำหนด Attribute ดังนี้
  - BORDER="2" หมายถึงกำหนดให้ตารางมีเส้นขอบกว้าง 2PX
  - BORDERCOLOR="RED" หมายถึงกำหนดให้สีของเส้นขอบเป็นสีแดง
  - CELLPADDING="30PX" หมายถึงกำหนดให้มีระยะห่างของเนื้อหาจากเส้นขอบภายในตารางเท่ากับ 30PX
  - <TR><TD> หมายถึง TAG สร้างแถวและคอลัมน์ในตาราง
  - <FONT FACE="MS Sans Serif " COLOR="BLUE" ><B>ข้อความ</B></FONT> คือ TAG HTML กำหนดรูปแบบการแสดงผลข้อความ

จากตัวอย่างพบว่าจะต้องใช้คำสั่ง TAG HTML กำหนด Attribute เพื่อแสดงข้อความในตารางให้มีรูปแบบตามที่ต้องการ และใช้ TAG HTML เพื่อกำหนดรูปแบบตัวอักษรที่ต้องการ หากตารางมีจำนวนหลายคอลัมน์ก็จะต้องกำหนด TAG HTML จำนวน

มากขึ้นด้วย ก็จะทำให้ไฟล์เว็บเพจมีขนาดใหญ่ แต่หากถ้าใช้ CSS ในการกำหนดรูปแบบการแสดงผล จะมีการแยกคำสั่งที่ใช้ควบคุมการแสดงผลออกจากเนื้อหา ทำให้ลดคำสั่งที่ใช้ในการแสดงผลลง จากตัวอย่างข้างต้นสามารถใช้ CSS ควบคุมการแสดงผลโดยเริ่มต้นคำสั่ง CSS ด้วย TAG <STYLE> และจบคำสั่งด้วย TAG </STYLE> โดยข้างใน TAG <STYLE> จะเป็นคำสั่งที่ใช้ควบคุมการแสดงผลของ TAG HTML ที่อยู่ในเว็บเพจ โดยเฉพาะ ซึ่งสามารถกระทำได้ดังรูปที่ 6.2

```
<STYLE>
    TABLE,TD{
        BORDER:2PX SOLID RED;
    }
    TD{
        PADDING:30PX;
        FONT-FAMILY:"MS Sans Serif";
        COLOR:BLUE;
        FONT-WEIGHT:BOLD;}
</STYLE>
<TABLE><TR><TD>ข้อความ</TD></TR></TABLE>
```

### รูปที่ 6.2 ตัวอย่างการคำสั่ง CSS เพื่อควบคุมการแสดงผล

จากรูปที่ 6.2 ตัวอย่างการใช้คำสั่ง CSS เพื่อควบคุมการแสดงผล เริ่มต้นที่ TAG <STYLE> และใน TAG <STYLE> ประกอบด้วยคำสั่งควบคุมการแสดงผล โดยที่

- TABLE,TD คือ Selector ที่ใช้ควบคุม TAG <TABLE> และ TAG <TD> ภายในเครื่องหมาย { } คือคำสั่งที่ใช้ควบคุมการแสดงผล ประกอบด้วยคำสั่ง BORDER:2PX SOLID RED โดยที่ BORDER หมายถึงการกำหนด Property เส้นขอบ 2PX SOLID RED หมายถึงการกำหนด Value ของเส้นขอบกว้าง 2PX รูปแบบเส้นขอบทึบสีแดง

- TD คือ Selector ที่ใช้ควบคุม TAG <TD> ภายในเครื่องหมาย { } ประกอบด้วยคำสั่ง

- PADDING:30PX โดยที่ PADDING หมายถึงระยะห่างของเนื้อหาจากเส้นขอบภายในตาราง 30PX

- FONT-FAMILY:"MS Sans Serif" หมายถึงรูปแบบตัวอักษร MS Sans Serif

- COLOR:BLUE หมายถึง สีตัวอักษรเป็นสีน้ำเงิน

- FONT-WEIGHT:BOLD หมายถึงตัวอักษรเป็นตัวหนา

จากนั้นก็เขียน TAG `</STYLE>` ซึ่งเป็น TAG ปิดคำสั่ง CSS การกำหนดรูปแบบการแสดงผลสามารถครอบคลุมทุกเซลล์ที่อยู่ในตาราง ซึ่งจะแตกต่างจากการใช้ TAG HTML ที่จะต้องเพิ่ม TAG การแสดงผลควบคุมเนื้อหาทุกเนื้อหา

### 1.1 รูปแบบคำสั่ง CSS

รูปแบบคำสั่ง CSS ประกอบด้วย 2 ส่วน คือ

(1) Selector คือ ชื่อ TAG HTML เช่น BODY, TABLE, H1, P ฯลฯ เป็นต้น หรือค่าของ Attribute ที่กำหนดให้กับ TAG HTML เช่น ค่าของ Attribute Class หรือ ค่าของ Attribute ID ซึ่งจะเป็นส่วนที่อยู่ก่อนเครื่องหมายปีกกา มีไว้สำหรับเพื่อระบุถึงส่วนที่ต้องการควบคุมการแสดงผล

(2) Declaration คือ คุณสมบัติหรือรูปแบบที่ต้องการให้ส่วนที่ควบคุมซึ่งถูกระบุถึงใน Selector แสดงผล เป็นส่วนที่อยู่ภายในเครื่องหมายปีกกา ประกอบด้วย Property และ Value ซึ่งเป็นส่วนที่ใช้สำหรับกำหนดรูปแบบการแสดงผลโดยที่

(2.1) Property คือคุณสมบัติหรือรูปแบบที่กำหนดให้กับ Selector

(2.2) Value คือค่าที่กำหนดให้กับ Property ใช้เครื่องหมาย colon (:) คั่นระหว่าง Property และ Value จบคำสั่งใช้เครื่องหมาย Semi-colon(;)

รูปแบบการใช้งาน CSS ดังนี้

```
Selector {
    property1: value;
    property2: value;
}
```

ตัวอย่างการเขียน CSS เช่น

```
TD{
    PADDING:30PX;
    FONT-FAMILY:"MS Sans Serif";
    COLOR:BLUE;
    FONT-WEIGHT:BOLD;
}
```

จากตัวอย่างสามารถอธิบายได้ว่า

- Selector คือ TD ซึ่งเป็น TAG HTML แสดงคอลัมน์ในตาราง
- Property และ Value ประกอบด้วย Property PADDING มี Value เป็น 30PX, Property FONT-FAMILY มี Value เป็น MS Sans Serif, Property COLOR มี Value เป็น BLUE และ Property FONT-WEIGHT มี Value เป็น BOLD จบการกำหนดค่าให้กับแต่ละ Property ด้วยเครื่องหมาย Semi-colon (;)

## 1.2 การกำหนด Property ให้กับ Selector

(1) สามารถกำหนด Property ให้กับ Selector ได้มากกว่า 1 Property ด้วยการใส่เครื่องหมาย Semi-colon (;) คั่นระหว่างการกำหนดค่าของแต่ละ Property เช่น กำหนดให้พื้นหลังของเว็บเพจเป็นสีน้ำเงิน และให้เนื้อหาที่อยู่ในเว็บเพจแสดงผลห่างจากขอบซ้ายและห่างจากขอบขวาของเบราว์เซอร์อย่างละ 15% สามารถสร้างคำสั่ง CSS ได้ดังรูปที่ 6.3

```
<STYLE>
  BODY {
    BACKGROUND-COLOR: BLUE;
    MARGIN-LEFT: 15%; MARGIN-RIGHT: 15%; }
</STYLE>
```

**รูปที่ 6.3** ตัวอย่างคำสั่ง CSS กำหนด Property ให้กับแต่ละ Selector มากกว่า 1 Property

(2) ในแต่ละ Selector สามารถใช้คำสั่งการกำหนด Property ร่วมกันได้ โดยใช้เครื่องหมาย comma (,) คั่นระหว่าง Selector แต่ถ้าหากต้องการกำหนด Property เพียงบางส่วนเหมือนกันก็จะแยกส่วนที่ต่างกันออกไป ตัวอย่างเช่น กำหนดให้ TAG <H1> และ TAG <H2> มีรูปแบบตัวอักษรเป็นแบบตัวเอียง แต่ใน TAG <H1> ให้ตัวอักษรเป็นสีแดง สามารถสร้างคำสั่ง CSS ได้ดังรูปที่ 6.4

```
<style>
  H1, H2 {FONT-STYLE: ITALIC;}
  H1 {COLOR:RED; }
</style>
```

**รูปที่ 6.4** ตัวอย่างการใช้คำสั่ง CSS ที่กำหนด Property ร่วมกันของแต่ละ Selector

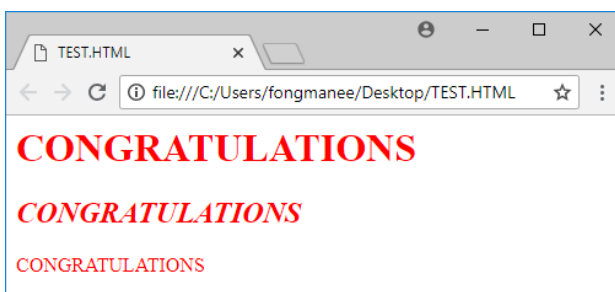
## 1.3 CSS มีคุณสมบัติการถ่ายทอด

จากโครงสร้างของภาษา HTML ประกอบด้วยโครงสร้าง 3 ส่วน คือ <HTML>...</HTML> และภายใน TAG <HTML> ก็จะประกอบด้วย TAG <HEAD> และ TAG <BODY> โดยเนื้อหาที่แสดงบนเบราว์เซอร์จะอยู่ภายใต้ TAG <BODY> ดังนั้น TAG HTML ใดก็ตามที่อยู่ภายใต้ TAG <BODY> ก็จะได้รับความสมบัติการถ่ายทอดจาก TAG <BODY> ด้วย เช่นกำหนดคุณสมบัติให้ TAG <BODY> มีสีตัวอักษรเป็นสีแดง ดังนั้น TAG อื่นๆ ที่อยู่ภายใต้ TAG <BODY> ก็จะมีสีตัวอักษรเป็นสีแดงด้วย ตัวอย่างเช่น สร้างคำสั่ง CSS กำหนด Selector เป็น H2 ซึ่งหมายถึง TAG HTML ที่เป็น TAG <H2> กำหนดรูปแบบตัวอักษรเป็นตัวเอียง สามารถสร้างคำสั่งได้ดังรูปที่ 6.5

```
<HTML><HEAD><STYLE>
  BODY {COLOR: RED; }
  H2 { FONT-STYLE: ITALIC;}
</STYLE></HEAD><BODY>
  <H1>CONGRATULATIONS</H1>   <H2>CONGRATULATIONS</H2>
  <DIV>CONGRATULATIONS</DIV>
</BODY></HTML>
```

รูปที่ 6.5 ตัวอย่างคำสั่ง CSS ที่มีคุณสมบัติการถ่ายทอด

จากรูปที่ 6.5 เมื่อแสดงผลผ่านทางเบราว์เซอร์ จะได้ผลลัพธ์ดังรูปที่ 6.6



รูปที่ 6.6 การแสดงผลบนเบราว์เซอร์จากไฟล์ตัวอย่างรูปที่ 6.5

จากรูปที่ 6.6 พบว่าตัวอักษรที่แสดงบนหน้าเว็บเพจมีสีตัวอักษรเป็นสีแดง ซึ่งได้รับความสมบัติการถ่ายทอดจากการกำหนดคำสั่งให้กับ TAG <BODY> มีตัวอักษรเป็นสีแดง

#### 1.4 การแทรกคำอธิบายในคำสั่ง CSS

การแทรกคำอธิบายในคำสั่ง CSS จะใช้เครื่องหมาย /\* เพื่อกำหนดจุดเริ่มต้นคำอธิบาย และใช้เครื่องหมาย \*/ เพื่อกำหนดจุดสิ้นสุดคำอธิบาย ตัวอย่างดังรูปที่ 6.7

```
<STYLE>
/* กำหนดสีตัวอักษรเป็นสีดำ ขนาด 14PX */
BODY{ COLOR:#000000;FONT-SIZE:14PX;}
</STYLE>
```

รูปที่ 6.7 ตัวอย่างการแทรกคำอธิบายในคำสั่ง CSS

#### 1.5 การกำหนดคำสั่ง CSS ให้กับเว็บเพจ

การกำหนดคำสั่ง CSS ให้กับเว็บเพจสามารถทำได้ 4 วิธีคือ

(1) แบบ In-line คือการกำหนด Attribute Style ให้กับ TAG HTML เพื่อใช้สำหรับควบคุมการแสดงผลเฉพาะ TAG HTML นั้น เช่นกำหนดให้ TAG <H1> มีสีตัวอักษรเป็นสีน้ำเงิน ขนาดตัวอักษร 20PX สามารถเขียนคำสั่ง TAG HTML ได้ดังนี้

```
<H1 STYLE="COLOR:BLUE;FONT-SIZE:20PX">
CONGRATULATIONS
</H1>
```

แบบ In-line นิยมใช้ควบคุม TAG HTML ที่ต้องการโดยตรงเท่านั้น ไม่นิยมใช้กับ TAG HTML ทั่วไป จะใช้เฉพาะในกรณีที่จำเป็น เพราะหากนำมาใช้กำหนดการแสดงผลของ TAG HTML โดยทั่วไปที่อยู่ในหน้าเว็บเพจ จะทำให้ไฟล์เว็บเพจมีขนาดใหญ่ เมื่อเปิดหน้าเว็บเพจจะใช้เวลาในการแสดงผลค่อนข้างนาน

(2) แบบใช้ TAG <STYLE> ที่อยู่ภายใต้ TAG <HEAD> เหมาะสมสำหรับใช้ควบคุมการแสดงผลเฉพาะหน้าเว็บเพจนั้น รูปแบบการใช้งานดังรูปที่ 6.8

```
<HTML><HEAD>
<STYLE>..... คำสั่ง CSS .....</STYLE>
</HEAD><BODY>
..... TAG HTML .....
</BODY></HTML>
```

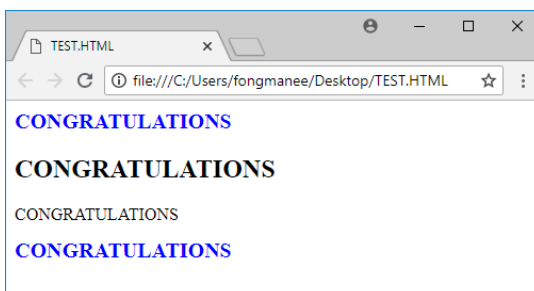
รูปที่ 6.8 ตัวอย่างรูปแบบการใช้งานคำสั่ง CSS แบบใช้ TAG <STYLE>

จากรูปที่ 6.8 ตัวอย่างรูปแบบการใช้งานคำสั่ง CSS แบบใช้ TAG <STYLE> หากต้องการกำหนดให้ TAG <H1> มีสีตัวอักษรเป็นสีน้ำเงิน และขนาดตัวอักษร 20PX สามารถเขียนคำสั่งได้รูปที่ 6.9

```
<HTML><HEAD>
<STYLE>
  H1 {
    COLOR:BLUE;
    FONT-SIZE:20PX;  }
</STYLE></HEAD>
<BODY>
  <H1>CONGRATULATIONS</H1>    <H2>CONGRATULATIONS</H2>
  <DIV>CONGRATULATIONS</DIV>  <H1>CONGRATULATIONS</H1>
</BODY></HTML>
```

รูปที่ 6.9 ตัวอย่างคำสั่ง CSS แบบใช้ TAG <STYLE>

จากรูปที่ 6.9 เมื่อแสดงผลผ่านทางเบราว์เซอร์ จะได้ผลลัพธ์ดังรูปที่ 6.10



รูปที่ 6.10 การแสดงผลบนเบราว์เซอร์จากไฟล์ตัวอย่างรูปที่ 6.9

จากรูปที่ 6.10 ข้อความในบรรทัดที่ 1 และบรรทัดที่ 4 คือผลจากการใช้ TAG <H1> ซึ่งควบคุมการแสดงผลด้วย CSS ในส่วนของ TAG <HEAD> จึงได้ตัวอักษรเป็นสีน้ำเงิน ขนาด 20PX ส่วนข้อความในบรรทัดที่ 2 เป็นข้อความในส่วนของ TAG <H2> และข้อความในบรรทัดที่ 3 เป็นข้อความในส่วนของ TAG <DIV> มีสีตัวอักษรเป็นสีดำ ซึ่งเป็นสีพื้นฐานของเบราว์เซอร์กำหนดไว้



(3) แบบใช้ TAG <LINK> ด้วยการสร้างไฟล์ CSS ไว้ต่างหาก จากนั้นเรียกใช้ไฟล์ CSS นั้น ด้วย TAG <LINK> ภายใน TAG <HEAD> วิธีการนี้เหมาะสำหรับการควบคุมการแสดงผลเว็บเพจหลายๆ เว็บเพจ และหากต้องการปรับแต่งคำสั่ง CSS สามารถแก้ไขได้ที่ไฟล์ CSS นั้นโดยตรง รูปแบบการใช้งานดังรูปที่ 6.11

```
<HTML><HEAD>
  <LINK REL = "STYLESHEET" TYPE="TEXT/CSS" HREF = "ชื่อไฟล์.CSS">
</HEAD>
  <BODY>..... TAG HTML .....</BODY>
</HTML>
```

**รูปที่ 6.11** ตัวอย่างรูปแบบการใช้งานคำสั่ง CSS แบบใช้ TAG <LINK>

จากรูปที่ 6.11 สร้างไฟล์ CSS ชื่อ STYLE.CSS ประกอบด้วยคำสั่ง CSS ที่กำหนดให้ TAG <H1> มีสีตัวอักษรเป็นสีน้ำเงิน ขนาดตัวอักษร 20PX สามารถเขียนคำสั่งได้ดังรูปที่ 6.12

```
H1{ COLOR:BLUE;
    FONT-SIZE:20PX; }
```

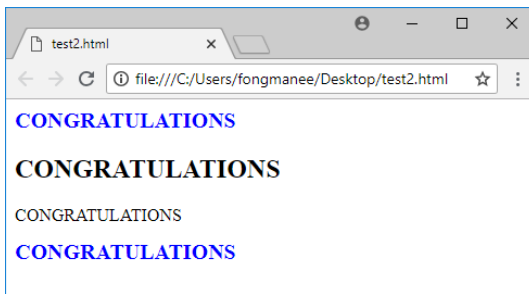
**รูปที่ 6.12** คำสั่ง CSS ในไฟล์ STYLE.CSS

เมื่อสร้างไฟล์ STYLE.CSS จากรูปที่ 6.12 เรียบร้อยแล้ว จากนั้นสร้างไฟล์เว็บเพจเพื่อเรียกใช้คำสั่ง CSS จากการ ใช้ TAG <LINK> ดังรูปที่ 6.13

```
<HTML><HEAD>
<LINK REL = "STYLESHEET" TYPE="TEXT/CSS" HREF = "STYLE.CSS">
</HEAD><BODY>
  <H1>CONGRATULATIONS</H1>      <H2>CONGRATULATIONS</H2>
  <DIV>CONGRATULATIONS</DIV>    <H1>CONGRATULATIONS</H1>
</BODY></HTML>
```

**รูปที่ 6.13** คำสั่ง HTML เรียกใช้คำสั่ง CSS จากไฟล์ STYLE.CSS แบบใช้ TAG <LINK>

จากรูปที่ 6.13 เมื่อแสดงผลบนเบราว์เซอร์จะได้ดังรูปที่ 6.14



รูปที่ 6.14 การแสดงผลบนเบราว์เซอร์จากไฟล์คำสั่งรูปที่ 6.13

จากรูปที่ 6.14 ข้อความในบรรทัดที่ 1 และบรรทัดที่ 4 คือผลจากการใช้ TAG <H1> ซึ่งมีการควบคุมการแสดงผลด้วยคำสั่ง CSS ในไฟล์ STYLE.CSS

(4) แบบใช้คำสั่ง @IMPORT ด้วยการสร้างไฟล์ CSS ไว้ต่างหาก จากนั้นเรียกใช้ไฟล์ CSS ใน TAG <STYLE> ภายใน TAG <HEAD> ด้วยคำสั่ง @IMPORT url("ชื่อไฟล์ CSS") วิธีการนี้เหมาะสำหรับการควบคุมการแสดงผลเว็บเพจด้วยคำสั่ง CSS จากไฟล์ CSS หลายไฟล์ รูปแบบการใช้งานดังรูปที่ 6.15

```
<HTML>
<HEAD>
<STYLE>
  @IMPORT URL("STYLE.CSS");
</STYLE>
</HEAD>
  <BODY>..... TAG HTML .....</BODY>
</HTML>
```

รูปที่ 6.15 ตัวอย่างการใช้งานคำสั่ง CSS แบบใช้คำสั่ง @IMPORT

จากรูปที่ 6.15 หากต้องการควบคุมการแสดงผลด้วยไฟล์ CSS หลายไฟล์ ก็จะเพิ่มคำสั่ง @IMPORT ตามไฟล์ CSS ที่ต้องการใช้งาน ตัวอย่างเช่นต้องการใช้งานไฟล์ CSS ชื่อ STYLE.CSS จากคำสั่งในรูปที่ 6.12 และไฟล์ CSS ชื่อ STYLE2.CSS ที่กำหนดให้ TAG <H2> มีสีตัวอักษรเป็นสีแดง ขนาดตัวอักษร 18PX ซึ่งสามารถเขียนคำสั่ง CSS ในไฟล์ STYLE2.CSS ดังรูปที่ 6.16

```
H2{
  COLOR:RED;
  FONT-SIZE:18PX;
}
```

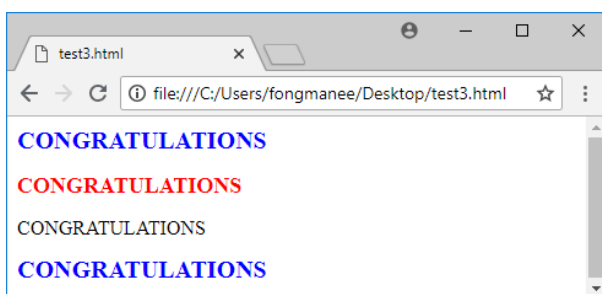
### รูปที่ 6.16 คำสั่ง CSS ในไฟล์ STYLE2.CSS

เมื่อสร้างไฟล์ STYLE2.CSS จากรูปที่ 6.16 เรียบร้อยแล้ว จากนั้นสร้างไฟล์เว็บเพจ เพื่อเรียกใช้คำสั่ง CSS จากไฟล์ STYLE.CSS และ STYLE2.CSS ด้วยคำสั่ง @IMPORT ดังรูปที่ 6.17

```
<HTML><HEAD>
<STYLE>
  @IMPORT URL("STYLE.CSS");
  @IMPORT URL("STYLE2.CSS");
</STYLE></HEAD><BODY>
  <H1>CONGRATULATIONS</H1>      <H2>CONGRATULATIONS</H2>
  <DIV>CONGRATULATIONS</DIV>    <H1>CONGRATULATIONS</H1>
</BODY></HTML>
```

### รูปที่ 6.17 คำสั่ง HTML ที่มีการใช้คำสั่ง CSS จากไฟล์ STYLE.CSS และ STYLE2.CSS แบบใช้คำสั่ง @IMPORT

จากรูปที่ 6.17 เมื่อแสดงผลบนเบราว์เซอร์จะได้ดังรูปที่ 6.18



### รูปที่ 6.18 การแสดงผลบนเบราว์เซอร์จากไฟล์คำสั่งรูปที่ 6.17

จากรูปที่ 6.18 ข้อความในบรรทัดที่ 1 และบรรทัดที่ 4 คือผลจากการใช้ TAG <H1> ที่ควบคุมการแสดงผลด้วยคำสั่ง CSS จากไฟล์ STYLE.CSS และข้อความในบรรทัดที่ 2 คือผลจากการใช้ TAG <H2> ที่ควบคุมการแสดงผลด้วยคำสั่ง CSS จากไฟล์ STYLE2.CSS ส่วนข้อความในบรรทัดที่ 3 จะมีสีตัวอักษรสีดำเนื่องจากเป็นค่าพื้นฐานของเบราว์เซอร์กำหนดไว้

### 1.6 ลำดับความสำคัญของคำสั่ง CSS

การกำหนดรูปแบบการแสดงผลด้วยคำสั่ง CSS ให้กับหน้าเว็บเพจนั้น สามารถกำหนดได้มากกว่าหนึ่งรูปแบบ โดยเบราว์เซอร์จะให้ความสำคัญแต่ละรูปแบบแตกต่างกัน ลำดับความสำคัญของคำสั่ง CSS แต่ละรูปแบบเรียงลำดับตามความสำคัญมากไปหาความสำคัญน้อยได้ดังนี้

- (1) การเรียกใช้แบบ In-line
- (2) การเรียกใช้ใน TAG <HEAD>
- (3) การเรียกใช้ด้วย TAG <LINK>
- (4) การเรียกใช้ด้วย คำสั่ง @IMPORT

โดยหากเว็บเพจนั้นมีคำสั่ง CSS ควบคุมการใช้งานมากกว่าหนึ่งรูปแบบ เบราวเซอร์จะแสดงผลตามรูปแบบคำสั่ง CSS ที่มีลำดับความสำคัญมากที่สุดบนหน้าเว็บเพจนั้น แต่ถ้าหากต้องการกำหนดความสำคัญให้กับคำสั่ง CSS ใดเป็นพิเศษ สามารถกำหนดด้วยการใช้คำสั่ง !IMPORTANT ต่อท้ายการกำหนดค่าให้กับ Property เช่นมีการเรียกใช้คำสั่ง CSS แบบใช้คำสั่ง @IMPORT ซึ่งมีความสำคัญน้อยกว่าการเรียกใช้แบบอื่น แต่หากต้องการให้เบราว์เซอร์แสดงผลตามคำสั่ง CSS ในไฟล์ที่ใช้คำสั่ง @IMPORT จึงเพิ่มคำสั่ง !IMPORTANT ไว้ต่อท้ายการกำหนดค่าให้กับ Property ตัวอย่างเช่น สร้างไฟล์ CSS ชื่อ STYLE1.CSS ประกอบด้วยคำสั่ง CSS ที่กำหนดให้ TAG <H1> มีสีตัวอักษรเป็นสีน้ำเงิน โดยเพิ่มคำสั่ง !IMPORTANT ต่อท้าย และกำหนดให้ขนาดตัวอักษรเท่ากับ 20PX สามารถเขียนคำสั่ง CSS ในไฟล์ STYLE1.CSS ได้ดังรูปที่ 6.19

```
H1{
COLOR:BLUE!IMPORTANT;
FONT-SIZE:20PX;
}
```

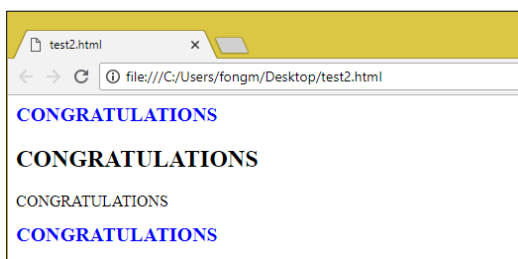
รูปที่ 6.19 คำสั่ง CSS ในไฟล์ STYLE1.CSS ที่มีการเพิ่มคำสั่ง !IMPORTANT ต่อท้าย

เมื่อสร้างไฟล์ STYLE1.CSS ดังรูปที่ 6.19 เรียบร้อยแล้ว จากนั้นสร้างไฟล์เว็บเพจ และกำหนดคำสั่ง CSS แบบ In-Line ให้กับ TAG <H1> ให้มีสีตัวอักษรเป็นสีชมพู และมีเรียกใช้ CSS แบบใช้คำสั่ง @IMPORT ดังรูปที่ 6.20

```
<HTML><HEAD>
<STYLE>
  @IMPORT URL("STYLE1.CSS");
</STYLE>
</HEAD>
<BODY>
  <H1 STYLE="COLOR:PINK;">CONGRATULATIONS</H1>
  <H2>CONGRATULATIONS</H2>
  <DIV>CONGRATULATIONS</DIV>
  <H1>CONGRATULATIONS</H1>
</BODY></HTML>
```

**รูปที่ 6.20** ตัวอย่างการใช้คำสั่ง CSS แบบต่างๆ

จากรูปที่ 6.20 เมื่อแสดงผลทางเบราว์เซอร์จะได้ดังรูปที่ 6.21



**รูปที่ 6.21** การแสดงผลบนเบราว์เซอร์จากไฟล์คำสั่งรูปที่ 6.20

จากรูปที่ 6.21 พบว่าข้อความในบรรทัดที่ 1 และบรรทัดที่ 4 คือผลจากการใช้ TAG <H1> ซึ่งมีการควบคุมการแสดงผลด้วย CSS ในไฟล์ STYLE1.CSS ที่มีการเพิ่มคำสั่ง !IMPORTANT ถึงแม้ว่าใน TAG HTML จะมีการกำหนด CSS แบบ In-Line แล้วก็ตาม

### 1.7 การกำหนดหน่วยในคำสั่ง CSS

ในการแสดงผลในหน้าเว็บเพจนั้น การกำหนดขนาดของสิ่งต่างๆ ที่แสดงผลนั้น หน่วยที่ใช้กำหนดขนาดถือว่าเป็นเรื่องสำคัญ เช่นการกำหนดขนาดความสูงและความกว้างของรูปภาพ ตาราง กล่องข้อความ หรือฟอร์มรับข้อมูล โดยการกำหนดหน่วยในคำสั่ง CSS ประกอบด้วยดังนี้

- PX ย่อมาจาก PIXEL เป็นการกำหนดขนาดโดยใช้จุดสี่
- EM ย่อมาจาก EM DASH เป็นการวัดความกว้างเทียบกับความกว้างของตัวอักษร M ที่เป็นตัวพิมพ์ใหญ่ ตามชนิดตัวอักษรที่ใช้งาน
- EX ย่อมาจาก X-height เป็นการวัดเทียบกับความสูงของตัว x ที่เป็นตัวพิมพ์เล็ก ตามชนิดของตัวอักษรที่ใช้งาน
- MM ย่อมาจาก Milliliter หน่วยเป็นมิลลิเมตร
- CM ย่อมาจาก Centimeter หน่วยเป็นเซนติเมตร
- IN ย่อมาจาก Inch หน่วยเป็นนิ้ว
- PT ย่อมาจาก Point

โดยที่หน่วยวัด PX, EM และ EX เป็นหน่วยวัดที่เทียบกับสิ่งอื่นไม่ใช่หน่วยตายตัว

### 1.8 การกำหนดสีใน CSS

การกำหนดสีด้วยคำสั่ง CSS จะใช้รูปแบบเช่นเดียวกับการกำหนดสีในภาษา HTML คือ สามารถกำหนดได้ 3 ลักษณะ ดังนี้

- 1) การระบุชื่อสีในภาษาอังกฤษ
- 2) ใช้ฟังก์ชัน rgb()
- 3) ระบุในรูปแบบเลขฐานสิบหกนำหน้าด้วยเครื่องหมาย #

## 2. ชนิดของ Selector ใน CSS

การกำหนดรูปแบบการแสดงผลด้วย CSS นั้น สามารถกำหนดชนิดของ Selector ที่ใช้ควบคุมการแสดงผลได้หลากหลายรูปแบบตาม TAG ของ HTML ดังนี้

- (1) Class Selectors
- (2) ID Selectors
- (3) The Universal Selector
- (4) Attribute Selectors

## 2.1 Class Selectors

การประกาศ Class Selectors จะเริ่มต้นด้วยเครื่องหมาย . แล้วตามด้วยชื่อของ Class ใน TAG HTML ที่อยู่ใน TAG <BODY> ของเว็บเพจที่ต้องการควบคุมการแสดงผล โดยชื่อ Class สามารถเป็นตัวอักษร ตัวเลข เครื่องหมาย -, \_ ได้เท่านั้น การใช้ Class Selectors ใน TAG HTML 1 TAG สามารถใช้ Class Selector ควบคุมการแสดงผลได้หลาย Class Selectors

ตัวอย่าง เช่นกำหนดคำสั่ง CSS ควบคุมการแสดงผล TAG HTML ที่มี Class ชื่อ FONT1 กำหนดให้มีตัวอักษรสีดำ และขนาดตัวอักษร 50PX และ Class ชื่อ FONT2 กำหนดให้มีตัวอักษรสีเขียว และรูปแบบตัวอักษรเอียง คำสั่งดังรูปที่ 6.22

```
<STYLE>
  .FONT1{ COLOR:BLACK; FONT-SIZE:50PX; }
  .FONT2{ COLOR:GREEN;FONT-STYLE: ITALIC; }
</STYLE>
```

### รูปที่ 6.22 ตัวอย่างการกำหนด CSS แบบ Class Selectors

จากรูปที่ 6.22 นำไปใช้งานกับคำสั่ง HTML ได้ดังคำสั่งดังรูปที่ 6.23

```
<HTML><HEAD>
<STYLE>
  .FONT1{ COLOR:BLACK; FONT-SIZE:50PX; }
  .FONT2{ COLOR:GREEN;FONT-STYLE: ITALIC; }
</STYLE>
</HEAD>
<BODY>
  <H1 CLASS="FONT1">CONGRATULATIONS</H1>
  <DIV>CONGRATULATIONS</DIV>
  <DIV CLASS="FONT1 FONT2">CONGRATULATIONS</DIV>
</BODY></HTML>
```

### รูปที่ 6.23 ตัวอย่างการใช้งานการกำหนด CSS แบบ Class Selectors

จากรูปที่ 6.23 เมื่อแสดงผลผ่านทางเบราว์เซอร์ จะได้ผลลัพธ์ดังรูปที่ 6.24



รูปที่ 6.24 การแสดงผลบนเบราว์เซอร์จากไฟล์คำสั่งในรูปที่ 6.23

จากรูปที่ 6.24 พบว่า

- ข้อความในบรรทัดที่ 1 คือผลจากการใช้ TAG <H1> ที่กำหนด Attribute Class เป็น FONT1 ดังนั้นเมื่อแสดงผลผ่านทางเบราว์เซอร์จะได้ตัวอักษรเป็นสีดำและขนาด 50PX
- ข้อความในบรรทัดที่ 2 คือผลจากการใช้ TAG <DIV> ที่ไม่ได้กำหนด Attribute Class ดังนั้นเมื่อแสดงผลผ่านทางเบราว์เซอร์จึงมีตัวอักษรสีดำขนาดปกติ ตามค่าพื้นฐานที่เบราว์เซอร์กำหนดไว้
- ข้อความในบรรทัดที่ 3 คือผลจากการใช้ TAG <DIV> ตัวที่สองที่กำหนด Attribute Class เป็น FONT1 และ FONT2 ดังนั้นเมื่อแสดงผลผ่านทางเบราว์เซอร์จึงมีคุณสมบัติที่กำหนดไว้ทั้ง Class FONT1 และ FONT2 ซึ่งก็คือสีตัวอักษรสีเขียว ขนาด 50PX และรูปแบบตัวอักษรเอียง

## 2.2 ID Selectors

การประกาศ ID Selectors จะเริ่มต้นด้วยเครื่องหมาย # แล้วตามด้วยชื่อของ ID ใน TAG HTML ที่อยู่ใน TAG <BODY> ของเว็บเพจที่ต้องการควบคุมการแสดงผล โดยที่ชื่อ ID สามารถเป็นตัวอักษร ตัวเลข เครื่องหมาย -, \_ ได้เท่านั้น การใช้ ID Selectors ใน TAG HTML 1 TAG สามารถใช้ ID Selectors ได้เพียง 1 เท่านั้น ซึ่งแตกต่างจากการใช้ Class Selectors ที่สามารถมีได้หลาย Class Selectors

ตัวอย่าง เช่นกำหนดคำสั่ง CSS ควบคุมการแสดงผล TAG HTML ที่มี ID ชื่อ SHOW1 ให้มีสีตัวอักษรเป็นสีน้ำเงิน ขนาดตัวอักษร 20PX คำสั่งดังรูปที่ 6.25

```
<STYLE>#SHOW1 { COLOR:BLUE;FONT-SIZE:20PX;}</STYLE>
```

รูปที่ 6.25 ตัวอย่างการกำหนด CSS แบบ ID Selectors

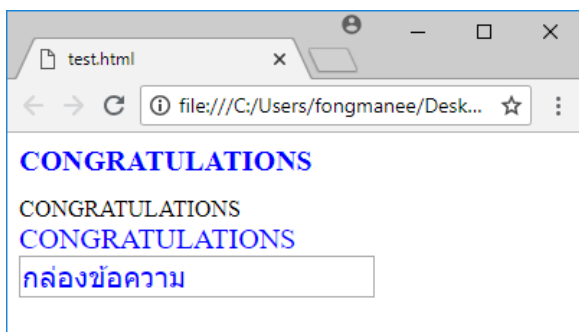


จากรูปที่ 6.25 กำหนดให้มีการเรียกใช้ Attribute ID SHOW1 ใน TAG <H1>, TAG <DIV> และ TAG <INPUT> คำสั่งรูปที่ 6.26

```
<HTML><HEAD>
<STYLE>
  #SHOW1{ COLOR:BLUE;FONT-SIZE:20PX;}
</STYLE></HEAD><BODY>
  <H1 ID="SHOW1">CONGRATULATIONS</H1>
  <DIV>CONGRATULATIONS</DIV>
  <DIV ID="SHOW1">CONGRATULATIONS</DIV>
  <INPUT TYPE="TEXT" ID="SHOW1">
</BODY></HTML>
```

**รูปที่ 6.26** ตัวอย่างการใช้งานการกำหนด CSS แบบ ID Selectors

จากรูปที่ 6.26 เมื่อแสดงผลผ่านทางเบราว์เซอร์ จะได้ผลลัพธ์ดังรูปที่ 6.27



**รูปที่ 6.27** การแสดงผลบนเบราว์เซอร์จากไฟล์คำสั่งในรูปที่ 6.26

จากรูปที่ 6.27 พบว่า

- ข้อความในบรรทัดที่ 1 คือผลจากการใช้ TAG <H1> ที่กำหนด Attribute ID เป็น SHOW1 ดังนั้นเมื่อแสดงผลผ่านทางเบราว์เซอร์จะมีตัวอักษรเป็นสีน้ำเงินและขนาด 20PX
- ข้อความในบรรทัดที่ 2 คือผลจากการใช้ TAG <DIV> ที่ไม่ได้กำหนด Attribute ID ดังนั้นเมื่อแสดงผลผ่านทางเบราว์เซอร์จึงมีตัวอักษรสีดำขนาดปกติ

- ข้อความในบรรทัดที่ 3 คือผลจากการใช้ TAG <DIV> ตัวที่สองที่กำหนด Attribute ID เป็น SHOW1 ดังนั้นเมื่อแสดงผลผ่านทางเบราว์เซอร์ได้มีสีตัวอักษรสีน้ำเงิน และขนาด 20PX

- กล่องข้อความในบรรทัดที่ 4 คือผลจากการใช้ TAG <INPUT> ที่กำหนด Attribute ID เป็น SHOW1 เมื่อป้อนข้อมูลในกล่องข้อความจึงมีสีตัวอักษรเป็นสีน้ำเงินและขนาด 20PX

### 2.3 The Universal Selectors

การประกาศ The Universal Selectors จะใช้เครื่องหมาย \* ซึ่งหมายถึงการควบคุมการแสดงผลของ TAG HTML ทุกตัวที่อยู่ภายในหน้าเว็บเพจ

ตัวอย่างเช่น กำหนดให้การแสดงผลวัตถุที่อยู่บนหน้าเว็บเพจมีเส้นขอบแบบทึบกว้าง 1PX สีดำ คำสั่งดังรูปที่ 6.28

```
<STYLE>
  * { BORDER: 1PX SOLID BLACK; }
</STYLE>
```

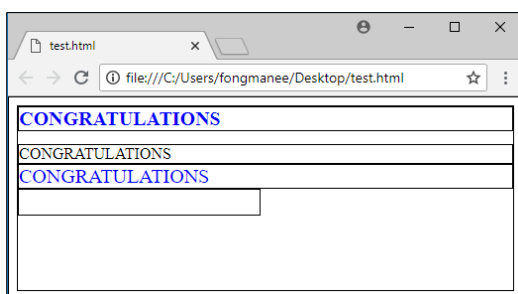
#### รูปที่ 6.28 ตัวอย่างการกำหนด CSS แบบ The Universal Selectors

จากรูปที่ 6.28 นำไปใช้งานกับคำสั่ง HTML ได้ดังคำสั่งดังรูปที่ 6.29

```
<HTML><HEAD><STYLE>
  * { BORDER: 1PX SOLID BLACK; }
  #SHOW1 { COLOR:BLUE;FONT-SIZE:20PX;}
</STYLE></HEAD>
<BODY>
  <H1 ID="SHOW1">CONGRATULATIONS</H1>
  <DIV>CONGRATULATIONS</DIV>
  <DIV ID="SHOW1">CONGRATULATIONS</DIV>
  <INPUT TYPE="TEXT" ID="SHOW1">
</BODY></HTML>
```

#### รูปที่ 6.29 ตัวอย่างการใช้งานการกำหนด CSS แบบ The Universal Selectors

จากรูปที่ 6.29 เมื่อแสดงผลผ่านทางเบราว์เซอร์ จะได้ผลลัพธ์ดังรูปที่ 6.30



**รูปที่ 6.30** การแสดงผลบนเบราว์เซอร์จากไฟล์ตัวอย่างรูปที่ 6.29

จากรูปที่ 6.30 การแสดงผลบนเบราว์เซอร์จากไฟล์คำสั่งในรูปที่ 6.29 พบว่าข้อความ หรือกล่องข้อความที่แสดงบนหน้าเว็บเพจไม่ว่าจะใช้ TAG HTML ใดก็ตามจะมีเส้นขอบแบบที่ขนาด 1PX สีดำ ซึ่งเป็นผลมาจากการกำหนด CSS ควบคุมแสดงผล แบบ The Universal Selectors

#### 2.4 Attribute Selectors

เป็นการควบคุมการแสดงผลด้วยการกำหนด Selector จาก Attribute ของ TAG HTML โดยมีรูปแบบการใช้งานดังนี้

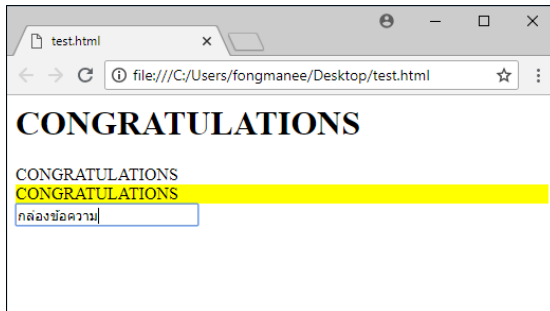
2.4.1 CSS [Attribute] Selectors คือการกำหนด Selectors โดยใช้ Attribute ที่กำหนดให้กับ TAG HTML รูปแบบการใช้งานคือ TAG HTML [Attribute]{คำสั่ง CSS;}

ตัวอย่างกำหนดสีพื้นหลังสีเหลืองให้กับ TAG HTML ที่กำหนด Attribute ID ดังรูปที่ 6.31

```
<HTML><HEAD><STYLE>
  DIV[ID] { BACKGROUND-COLOR: YELLOW;}
</STYLE></HEAD><BODY>
  <H1 ID="SHOW1">CONGRATULATIONS</H1>
  <DIV>CONGRATULATIONS</DIV>
  <DIV ID="SHOW1">CONGRATULATIONS</DIV>
  <INPUT TYPE="TEXT" ID="SHOW1">
</BODY></HTML>
```

**รูปที่ 6.31** ตัวอย่างการคำสั่ง CSS แบบกำหนด Selector จากการกำหนด [Attribute]

จากรูปที่ 6.31 เมื่อแสดงผลผ่านทางเบราว์เซอร์ จะได้ผลลัพธ์ดังรูปที่ 6.32



รูปที่ 6.32 การแสดงผลบนเบราว์เซอร์จากไฟล์ตัวอย่างรูปที่ 6.31

จากรูปที่ 6.32 พบว่าข้อความในบรรทัดที่ 1 และ บรรทัดที่ 2 มีพื้นหลังเป็นสีขาว เพราะเกิดจากการใช้ TAG <H1> และ TAG <DIV> ที่ไม่ได้อยู่ในเงื่อนไขของคำสั่ง CSS ที่กำหนดให้ TAG <DIV> ที่มีการกำหนด Attribute ID มีพื้นหลังเป็นสีเหลือง ส่วนข้อความในบรรทัดที่ 3 จะมีพื้นหลังเป็นสีเหลือง เป็นผลมาจากการใช้ TAG <DIV> ที่มีการกำหนด Attribute ID

2.4.2 CSS [Attribute="value"] คือการกำหนด Selectors โดยใช้ Attribute และค่าของ Attribute ที่กำหนดให้กับ TAG HTML รูปแบบการใช้งานคือ

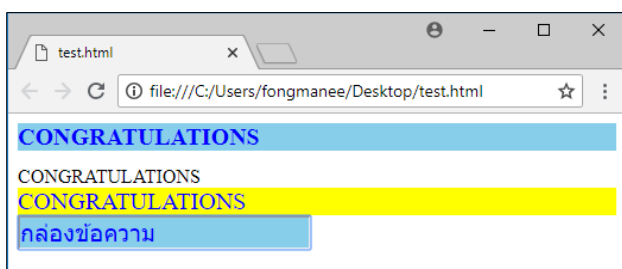
TAG HTML [Attribute="value"];{คำสั่ง CSS;}

ตัวอย่างเช่นกำหนดรูปแบบสีพื้นหลังสีน้ำเงินให้กับ TAG HTML ที่กำหนด Attribute ID ที่มีค่าเท่ากับ SHOW1 ดังรูปที่ 6.33

```
<HTML><HEAD><STYLE>
#SHOW1{ COLOR:BLUE;FONT-SIZE:20PX;}
DIV[ID] { BACKGROUND-COLOR: YELLOW;}
[ID="SHOW1"] {BACKGROUND-COLOR:SKYBLUE;}
</STYLE></HEAD><BODY>
  <H1 ID="SHOW1">CONGRATULATIONS </H1> <DIV>CONGRATULATIONS</DIV>
  <DIV ID="SHOW1">CONGRATULATIONS</DIV>
  <INPUT TYPE="TEXT" ID="SHOW1">
</BODY></HTML>
```

รูปที่ 6.33 ตัวอย่างคำสั่ง CSS แบบกำหนด Selectors จาก Attribute และค่าของ Attribute

จากรูปที่ 6.33 เมื่อแสดงผลผ่านทางเบราว์เซอร์ จะได้ผลลัพธ์ดังรูปที่ 6.34



รูปที่ 6.34 การแสดงผลบนเบราว์เซอร์จากไฟล์ตัวอย่างรูปที่ 6.33

จากรูปที่ 6.34 พบว่าข้อความในบรรทัดที่ 1 และกล่องข้อความในบรรทัดที่ 4 มีพื้นหลังเป็นสีฟ้า เพราะเกิดจากการใช้ TAG <H1> และ TAG <INPUT> ที่มีการกำหนด Attribute ID มีค่าเท่ากับ SHOW1 ส่วนข้อความในบรรทัดที่ 3 มีพื้นหลังเป็นสีเหลือง เกิดจากการใช้ TAG <DIV> ที่มีการกำหนด Attribute ID ซึ่งตรงกับเงื่อนไขของคำสั่ง CSS DIV[ID] ที่กำหนดให้มีพื้นหลังเป็นสีเหลืองซึ่งมีคุณสมบัติเจาะจงกว่า คำสั่ง [ID="SHOW1"] ซึ่งไม่ได้ระบุ Selector เจาะจง

### 3. การกำหนด Property ควบคุมการแสดงผล

#### 3.1 การกำหนด Property แสดงผลข้อความ

ในภาษา HTML Version 5 ได้มีการยกเลิกการใช้งาน TAG <FONT> ซึ่งเป็น TAG ที่ใช้สำหรับควบคุมการแสดงผลข้อความ โดยมีการใช้คำสั่ง CSS แทน อีกทั้งการใช้งาน TAG <FONT> มีความยุ่งยากเพราะต้องใช้ TAG <FONT> ทุกครั้งที่ต้องการควบคุมการแสดงผลข้อความ แต่หากใช้ CSS จะสะดวกรวดเร็วกว่า เนื่องจากสามารถควบคุมการแสดงผลได้ทั่วถึงทั้งเว็บเพจ ทำให้ไฟล์เว็บเพจมีขนาดเล็กลงและการเขียนคำสั่งไม่ยุ่งยาก การใช้งานคำสั่ง CSS ควบคุมการแสดงผลข้อความสามารถกำหนด Property ได้ดังนี้

ตารางที่ 6.1 การกำหนด Property ให้กับคำสั่ง CSS ควบคุมการแสดงผลข้อความ

| Property    | ค่าที่กำหนดให้       | ความหมาย   |
|-------------|----------------------|--|
| FONT-FAMILY | ชื่อชนิด<br>ตัวอักษร | กำหนดชนิดตัวอักษร หากชนิดตัวอักษรมีช่องว่างจะ<br>ระบุภายใต้เครื่องหมาย “” หรือ “” เช่น<br>FONT-FAMILY:"COURIER NEW"; |

ตารางที่ 6.1 (ต่อ)

| Property    | ค่าที่กำหนดให้   | ความหมาย  |
|-------------|--|---|
| FONT-SIZE   | ค่าตัวเลข/<br>MEDIUM/<br>XX-SMALL/<br>X-SMALL/<br>SMALL/<br>LARGE/<br>X-LARGE/<br>XX-LARGE/<br>SMALLER/<br>LARGER/<br>LENGTH/<br>INITIAL/<br>INHERIT | กำหนดขนาดตัวอักษร<br>- ค่าตัวเลขแล้วตามด้วยหน่วยที่ต้องการเช่น<br>FONT-SIZE:2EM;<br>- หากมีการกำหนดขนาดตัวอักษรปกติไว้แล้วต้องการ<br>แสดงตัวอักษรให้มีขนาดแตกต่างจากตัวอักษรปกติที่<br>กำหนดไว้สามารถระบุแบบสัมพัทธ์ได้คือ<br>- MEDIUM<br>- XX-SMALL<br>- X-SMALL<br>- SMALL<br>- LARGE<br>- X-LARGE<br>- XX-LARGE<br>- SMALLER<br>- LARGER<br>- LENGTH<br>- INITIAL คือค่าเริ่มต้นของเบราวเซอร์<br>- INHERIT คือสืบทอดคุณสมบัติจากองค์ประกอบหลัก |
| FONT-WEIGHT | NORMAL/<br>BOLD/<br>BOLDER/<br>LIGHTER/<br>100/200/300/<br>400/500/600/<br>700/800/900/<br>INITIAL/<br>INHERIT                                       | กำหนดความหนาของตัวอักษร<br>- NORMAL คือตัวอักษรปกติ<br>- BOLD คือตัวอักษรหนา<br>- หากมีการกำหนดตัวอักษรปกติไว้สามารถกำหนด<br>ความหนาของตัวอักษรเทียบกับตัวอักษรปกติ สามารถ<br>ระบุแบบสัมพัทธ์ได้คือ BOLDER คือตัวหนากว่าปกติ<br>หรือ LIGHTER คือมีความหนาน้อยกว่าตัวอักษรปกติ<br>- ตัวเลข 100 ถึง 900 คือระดับความหนาจากน้อยไป<br>มาก   |

ตารางที่ 6.1 (ต่อ)

| Property        | ค่าที่กำหนดให้  | ความหมาย   |
|-----------------|---|--|
| FONT-STYLE      | NORMAL/<br>ITALIC/<br>OBLIQUE/<br>INITIAL/<br>INHERIT | กำหนดรูปแบบตัวอักษร<br>- NORMAL คือตัวอักษรปกติไม่เอียง<br>- ITALIC คือตัวอักษรเอียง<br>- OBLIQUE คือตัวอักษรเอียง<br>- INITIAL คือค่าเริ่มต้นของเบราว์เซอร์<br>- INHERIT คือสืบทอดคุณสมบัติจาก TAG HTML ที่เป็นองค์ประกอบหลัก |
| COLOR           | สี  | กำหนดสีให้กับตัวอักษร  |
| TEXT-DECORATION | NONE/<br>UNDERLINE/<br>OVERLINE/<br>LINE-THROUGH      | กำหนดการขีดเส้นให้กับข้อความ โดยที่<br>- NONE หมายถึงไม่แสดงขีดเส้นใต้ข้อความ<br>- UNDERLINE หมายถึงแสดงขีดเส้นใต้ข้อความ<br>- OVERLINE หมายถึงแสดงขีดเส้นบนข้อความ<br>- LINE-THROUGH หมายถึงการขีดเส้นทับข้อความ              |
| TEXT-ALIGN      | LEFT/RIGHT/<br>CENTER/<br>JUSTIFY                     | กำหนดตำแหน่งการจัดวางข้อความโดยที่<br>- LEFT คือข้อความชิดซ้าย<br>- RIGHT คือข้อความชิดขวา<br>- CENTER คือข้อความอยู่ตรงกลาง<br>- JUSTIFY คือการกระจายข้อความให้ตรงกันทั้งแนวขอบซ้ายและขวา                                     |
| LINE-HEIGHT     | NORMAL/<br>ตัวเลข                                     | การกำหนดความสูงของบรรทัดที่แสดงผลข้อความ   |

จากตารางที่ 6.1 ตัวอย่างเช่นกำหนดให้ข้อความในหน้าเว็บเพจ มีชนิดตัวอักษรเป็น "Courier New" มีขนาดตัวอักษรเป็น 2EM มีความหนาของตัวอักษรเป็น 900 มีรูปแบบตัวอักษรเป็นตัวเอียง สีตัวอักษรเป็นสีแดง มีการขีดเส้นใต้ข้อความ กำหนดตำแหน่งการจัดวางข้อความอยู่ตรงกลางหน้าเว็บเพจ และกำหนดความสูงของบรรทัดที่แสดงผลข้อความเท่ากับ 2.5EM ดังรูปที่ 6.35

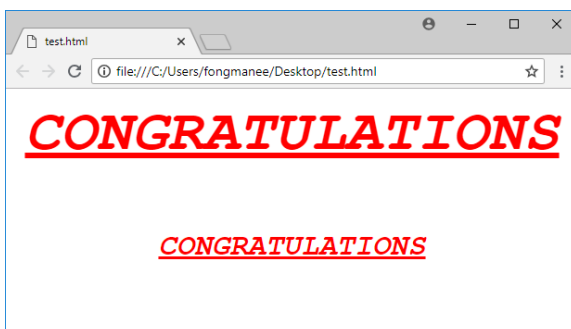
```

<HTML><HEAD><STYLE>
BODY{
  FONT-FAMILY:"COURIER NEW";
  FONT-SIZE:2EM;
  FONT-WEIGHT:900;
  FONT-STYLE:ITALIC;
  COLOR:RED;
  TEXT-DECORATION:UNDERLINE;
  TEXT-ALIGN:CENTER;
  LINE-HEIGHT:2.5EM;
}</STYLE></HEAD><BODY>
  <H1>CONGRATULATIONS </H1>
  <DIV>CONGRATULATIONS</DIV>
</BODY></HTML>

```

รูปที่ 6.35 ตัวอย่างการคำสั่ง CSS กำหนด Property ความคุมการแสดงผลข้อความ

จากรูปที่ 6.35 เมื่อแสดงผลผ่านทางเบราว์เซอร์ จะได้ผลลัพธ์ดังรูปที่ 6.36



รูปที่ 6.36 การแสดงผลบนเบราว์เซอร์จากไฟล์ตัวอย่างรูปที่ 6.35

จากรูปที่ 6.36 พบว่าข้อความในบรรทัดที่ 1 และบรรทัดที่ 2 มีรูปแบบการแสดงผลดังคำสั่ง CSS ที่กำหนดไว้ใน Selector BODY จากรูปที่ 6.35



### 3.2 การกำหนด Property ควบคุมการแสดงผลพื้นหลัง

พื้นหลังของเว็บเพจถือว่าเป็นสำคัญส่วนหนึ่งที่ช่วยทำให้เนื้อหามีความน่าสนใจมากยิ่งขึ้น สามารถใช้คำสั่ง CSS ควบคุมการแสดงผลพื้นหลังโดยการกำหนด Property ได้ดังตารางที่ 6.2

ตารางที่ 6.2 การกำหนด Property ให้กับคำสั่ง CSS ควบคุมการแสดงผลพื้นหลัง

| Property                | ค่าที่กำหนดให้  | คำอธิบาย   |
|-------------------------|---|--|
| BACKGROUND<br>-COLOR    | สี  | กำหนดสีพื้นหลัง  |
| BACKGROUND<br>-IMAGE    | URL("ชื่อไฟล์")   | กำหนดภาพพื้นหลัง โดยระบุชื่อชื่อไฟล์รูปภาพที่ต้องการ   |
| BACKGROUND<br>-REPEAT   | REPEAT/<br>NO-REPEAT/<br>REPEAT-X/<br>REPEAT-Y  | กำหนดการแสดงผลซ้ำภาพพื้นหลังโดยที่<br>- REPEAT คือการแสดงผลซ้ำทั้งแนวนอนและแนวตั้ง<br>- NO-REPEAT คือไม่แสดงผลซ้ำ<br>- REPEAT-X คือการแสดงผลซ้ำแนวนอน<br>- REPEAT-Y คือการแสดงผลซ้ำแนวตั้ง   |
| BACKGROUND<br>-POSITION | LEFT TOP/<br>LEFT CENTER/<br>LEFT BOTTOM/<br>RIGHT TOP/<br>RIGHT CENTER/<br>RIGHT BOTTOM/<br>CENTER TOP/<br>CENTER CENTER/<br>CENTER BOTTOM/<br>X% Y%/<br>X Y/<br>INITIAL | กำหนดตำแหน่งการจัดวางภาพพื้นหลังในแนวนอนและแนวตั้งของขอบเบราว์เซอร์ โดยที่<br>- LEFT TOP คือชิดซ้ายติดขอบบน<br>- LEFT CENTER คือชิดซ้ายตรงกลางแนวตั้ง<br>- LEFT BOTTOM คือชิดซ้ายติดขอบล่าง<br>- RIGHT TOP คือชิดขวาติดขอบบน<br>- RIGHT CENTER คือชิดขวาแนวตั้ง<br>- RIGHT BOTTOM คือชิดขวาติดขอบล่าง<br>- CENTER TOP คือตรงกลางแนวนอนติดขอบบน<br>- CENTER CENTER คือตรงกลางแนวนอนและแนวตั้ง |

ตารางที่ 6.2 (ต่อ)

| Property              | ค่าที่กำหนดให้   | คำอธิบาย  |
|-----------------------|------------------|---|
|                       |                  | <ul style="list-style-type: none"> <li>- CENTER BOTTOM คือตรงกลางแนวนอนและติดขอบล่าง</li> <li>- X% Y% คือการกำหนดระยะห่างเป็นแบบค่าร้อยละ โดย X% คือ ร้อยละที่ห่างจากขอบซ้ายไปขอบขวา และ Y% คือร้อยละที่ห่างจากขอบบนไปขอบล่าง</li> <li>- X Y คือ X คือการกำหนดระยะห่างเป็นค่าตัวเลข โดย X คือระยะที่ห่างจากขอบซ้ายไปขอบขวาและ Y คือระยะที่ห่างจากขอบบนไปขอบล่าง</li> <li>- INITIAL คือ ค่าเริ่มต้นของเบรราวเซอร์</li> </ul> |
| BACKGROUND-ATTACHMENT | SCROLL/<br>FIXED | <p>กำหนดการเลื่อนภาพพื้นหลัง</p> <ul style="list-style-type: none"> <li>- SCROLL คือให้ภาพพื้นหลังเลื่อนตามการเลื่อนเมาส์</li> <li>- FIXED คือภาพพื้นหลังไม่เลื่อนตามการเลื่อนเมาส์</li> </ul>  |

จากตารางที่ 6.2 การกำหนด Property ควบคุมการแสดงผลพื้นหลังสามารถใช้ได้กับพื้นหลังของเว็บเพจ พื้นหลังของข้อความ พื้นหลังของกล่องข้อความ พื้นหลังของตาราง เป็นต้น ตัวอย่างเช่น

- กำหนดพื้นหลังเว็บเพจเป็นสีเหลือง และมีรูปภาพชื่อ IMG.JPG เป็นพื้นหลัง

กำหนดให้ไม่แสดงซ้ำภาพพื้นหลัง และภาพพื้นหลังวางในตำแหน่งชิดขอบขวาและติดขอบล่าง

- กำหนดพื้นหลังข้อความใน TAG <H1> เป็นสีส้ม
  - กำหนดพื้นหลังข้อความใน TAG <DIV> เป็นสีฟ้า
  - กำหนดพื้นหลังตารางด้วย รูปภาพชื่อ IMG.JPG
  - กำหนดพื้นหลังกล่องข้อความใน TAG <INPUT> เป็นสีชมพู
  - กำหนดพื้นหลังให้ ID Selector ชื่อ CELL มีพื้นหลังเป็นสีขาว
  - กำหนดพื้นหลังให้ Class Selector ชื่อ BTN ให้มีพื้นหลังเป็นสีน้ำตาล
- สามารถสร้างคำสั่ง CSS ดังรูปที่ 6.37

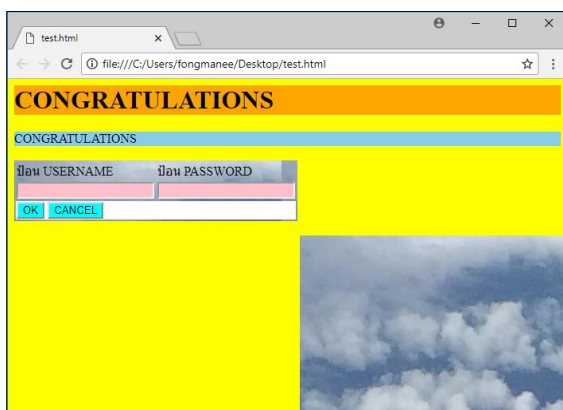
```

<HTML><HEAD><STYLE>
  BODY{BACKGROUND-COLOR:YELLOW;  BACKGROUND-IMAGE:URL("IMG.JPG");
  BACKGROUND-REPEAT:NO-REPEAT;  BACKGROUND-POSITION:RIGHT BOTTOM;}
  H1 {BACKGROUND-COLOR:ORANGE;}
  DIV {BACKGROUND-COLOR:SKYBLUE;}
  TABLE {BACKGROUND-IMAGE:URL("IMG.JPG");}
  INPUT {BACKGROUND-COLOR:PINK;}
  #CELL {BACKGROUND-COLOR:WHITE;}
  .BTN {BACKGROUND-COLOR:AQUA;}
</STYLE></HEAD><BODY>
  <H1>CONGRATULATIONS </H1>
  <DIV>CONGRATULATIONS</DIV><BR>
<TABLE>
<TR><TD>ป้อน USERNAME </TD><TD>ป้อน PASSWORD </TD></TR>
<TR><TD><INPUT TYPE="TEXT" NAME="USERNAME"></TD>
  <TD><INPUT TYPE="PASSWORD" NAME="PASS"></TD></TR>
<TR><TD ID="CELL" COLSPAN="2">
  <INPUT TYPE="SUBMIT" CLASS="BTN" VALUE="OK">
  <INPUT TYPE="RESET" CLASS="BTN" VALUE="CANCEL"></TD></TR>
</TABLE></BODY></HTML>

```

รูปที่ 6.36 ตัวอย่างการคำสั่ง CSS กำหนด Property ควบคุมการแสดงผลพื้นหลัง

จากรูปที่ 6.36 เมื่อแสดงผลผ่านทางเบราว์เซอร์ จะได้ผลลัพธ์ดังรูปที่ 6.37



รูปที่ 6.37 การแสดงผลบนเบราว์เซอร์จากไฟล์ตัวอย่างรูปที่ 6.36

จากรูปที่ 6.37 พบว่า

- พื้นหลังของเว็บเพจเป็นสีเหลือง และมีรูปภาพชื่อ IMG.JPG เป็นภาพพื้นหลังโดยวางไว้ในตำแหน่งชิดขอบและติดขอบล่างของเบราว์เซอร์ โดยไม่แสดงซ้ำภาพพื้นหลัง
- ข้อความในบรรทัดที่ 1 มีพื้นหลังเป็นสีส้ม คือผลจากการใช้ TAG <H1> ซึ่งควบคุมการแสดงผลด้วยคำสั่ง CSS ให้มีพื้นหลังเป็นสีส้ม
- ข้อความในบรรทัดที่ 2 มีพื้นหลังเป็นสีส้ม คือผลจากการใช้ TAG <DIV> ซึ่งควบคุมการแสดงผลด้วยคำสั่ง CSS ให้มีพื้นหลังเป็นสีฟ้า
- ข้อความในบรรทัดที่ 3 คือผลการใช้ TAG <TABLE> ซึ่งควบคุมการแสดงผลด้วยคำสั่ง CSS ให้มีพื้นหลังเป็นรูปภาพชื่อ IMG.JPG
- กล่องข้อความในบรรทัดที่ 4 คือผลการใช้ TAG <INPUT> ซึ่งควบคุมการแสดงผลด้วยคำสั่ง CSS ให้มีพื้นหลังเป็นสีชมพู
- ปุ่ม OK และ ปุ่ม CANCEL ในบรรทัดที่ 5 คือผลการใช้ CLASS Selector ชื่อ BTN ที่กำหนดให้มีพื้นหลังเป็นสีน้ำทะเล และพื้นที่ที่แสดงปุ่มมีพื้นหลังเป็นสีขาว คือผลการใช้ ID Selector

### 3.3 การกำหนด Property ควบคุมการแสดงผลเส้นขอบ

การแสดงผลเส้นขอบ สามารถกำหนด Property ควบคุมได้ดังตารางที่ 6.3

ตารางที่ 6.3 การกำหนด Property ให้กับคำสั่ง CSS ควบคุมการแสดงผลเส้นขอบ

| Property     | ค่าที่กำหนดให้                          | คำอธิบาย  |
|--------------|---|---|
| BORDER-COLOR | สี                                      | กำหนดสีเส้นขอบ  |
| BORDER-WIDTH | ค่าตัวเลข/<br>MEDIUM/<br>THIN/<br>THICK | ขนาดความกว้างของเส้นขอบ กำหนดเป็นค่าของตัวเลข หรือ<br>- MEDIUM คือความกว้างปานกลาง<br>- THIN คือความกว้างแบบบาง<br>- THICK คือความกว้างอย่างหนา |

ตารางที่ 6.3 (ต่อ)

| Property         | ค่าที่กำหนดให้ | คำอธิบาย   |
|------------------|----------------|--|
| BORDER-<br>STYLE | NONE/          | กำหนดลักษณะของเส้นขอบ โดยที่   |
|                  | HIDDEN/        | - NONE คือไม่มีเส้นขอบ   |
|                  | DOTTED/        | - HIDDEN คือซ่อนเส้นขอบ  |
|                  | DASHED/        | - DOTTED คือเส้นขอบแบบจุด  |
|                  | SOLID/         | - DASHED คือเส้นขอบแบบขีด  |
|                  | DOUBLE/        | - SOLID คือเส้นขอบแบบทึบ   |
|                  | GROOVE/        | - DOUBLE คือเส้นขอบแบบคู่  |
|                  | RIDGE/         | - GROOVE คือเส้นขอบแบบ 3 มิติแบบมีร่อง   |
|                  | INSET/         | - RIDGE คือเส้นขอบแบบนูนขึ้น   |
|                  | OUTSET/        | - INSET คือเส้นขอบด้านใน   |
|                  | INITIAL/       | - OUTSET คือเส้นขอบด้านนอก   |
|                  | INHERIT        | - INITIAL คือค่าเริ่มต้นของเบราว์เซอร์<br>- INHERIT คือสืบทอดคุณสมบัติจาก TAG HTML ที่เป็นองค์ประกอบหลัก |

จากตารางที่ 6.3 การกำหนด Property ควบคุมการแสดงผลเส้นขอบสามารถใช้ได้กับวัตถุต่างๆ ที่แสดงบนหน้าเว็บเพจ เช่น พื้นหลังของเว็บเพจ กล่องข้อความ ตาราง และรูปภาพ เป็นต้น ตัวอย่างเช่น กำหนดเส้นขอบให้กับวัตถุต่อไปนี้ในหน้าเว็บเพจ

- หน้าเว็บเพจ ลักษณะเส้นขอบเป็นแบบจุดสี่เหลี่ยม
  - ข้อความใน TAG <H1> ลักษณะเส้นขอบเป็นแบบขีด
  - ข้อความใน TAG <DIV> ลักษณะเส้นขอบเป็นแบบทึบ
  - รูปภาพ TAG <IMG> ลักษณะเส้นขอบเป็นแบบคู่สีน้ำเงิน
  - ตารางใน TAG <TABLE> ลักษณะเส้นขอบเป็นแบบ 3 มิติแบบมีร่องสีเหลือง
  - กล่องข้อความใน TAG <INPUT> ลักษณะเส้นขอบเป็นแบบนูนขึ้นสีน้ำเงิน
  - ID Selector ชื่อ CELL ลักษณะเส้นขอบด้านในสีชมพู
  - Class Selector ชื่อ BTN ลักษณะเส้นขอบด้านนอกสีเขียวมีความกว้าง 5PX
- สามารถสร้างคำสั่ง CSS ดังรูปที่ 6.38

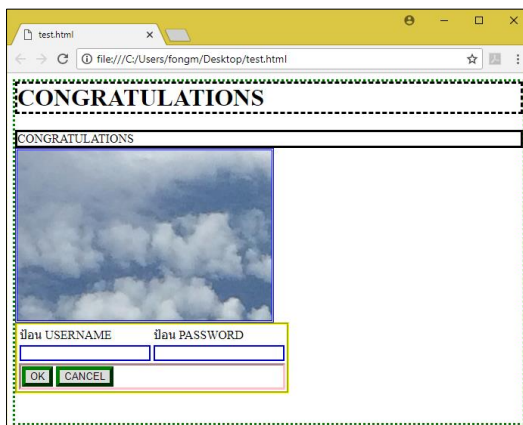
```

<HTML><HEAD><STYLE>
  BODY {BORDER-STYLE:DOTTED;BORDER-COLOR:GREEN;}
  H1 {BORDER-STYLE:DASHED;}
  DIV {BORDER-STYLE:SOLID; }
  IMG {BORDER-STYLE:DOUBLE;BORDER-COLOR:BLUE;}
  TABLE {BORDER-STYLE:GROOVE;BORDER-COLOR:YELLOW;}
  INPUT {BORDER-STYLE:RIDGE;BORDER-COLOR:BLUE;}
  #CELL {BORDER-STYLE:INSET;BORDER-COLOR:PINK;}
</STYLE></HEAD><BODY>
  <H1>CONGRATULATIONS </H1>   <DIV>CONGRATULATIONS</DIV>
  <IMG SRC="IMG.JPG">
<TABLE>
<TR><TD>ป้อน USERNAME </TD><TD>ป้อน PASSWORD </TD></TR>
<TR><TD><INPUT TYPE="TEXT" NAME="USERNAME"></TD>
  <TD><INPUT TYPE="PASSWORD" NAME="PASS"></TD></TR>
<TR><TD ID="CELL" COLSPAN="2">
  <INPUT TYPE="SUBMIT" CLASS="BTN" VALUE="OK">
  <INPUT TYPE="RESET" CLASS="BTN" VALUE="CANCEL"></TD></TR>
</TABLE></BODY></HTML>

```

รูปที่ 6.38 ตัวอย่างการคำสั่ง CSS กำหนด Property ควบคุมการแสดงผลเส้นขอบ

จากรูปที่ 6.38 เมื่อแสดงผลผ่านทางเบราว์เซอร์ จะได้ผลลัพธ์ดังรูปที่ 6.39



รูปที่ 6.39 การแสดงผลบนเบราว์เซอร์จากไฟล์ตัวอย่างรูปที่ 6.38

จากรูปที่ 6.39 พบว่า

- หน้าเว็บเพจมีเส้นขอบเป็นแบบจุดสี่เหลี่ยม
- ข้อความในบรรทัดที่ 1 มีเส้นขอบเป็นแบบขีด คือผลจากการใช้ TAG <H1> ซึ่งควบคุมการแสดงผลด้วยคำสั่ง CSS ให้มีเส้นขอบแบบขีด และที่เป็นสีดำเนื่องจากค่าเริ่มต้นของเบราว์เซอร์ถูกกำหนดให้สีพื้นฐานเป็นสีดำ
- ข้อความในบรรทัดที่ 2 มีเส้นขอบเป็นแบบทึบ คือผลจากการใช้ TAG <DIV> ซึ่งควบคุมการแสดงผลด้วยคำสั่ง CSS ให้มีเส้นขอบเป็นแบบทึบ
- รูปภาพในบรรทัดที่ 3 มีเส้นขอบเป็นแบบคู่สีน้ำเงิน คือผลจากการใช้ TAG <IMG> ซึ่งควบคุมการแสดงผลด้วยคำสั่ง CSS ให้มีเส้นขอบเป็นแบบคู่สีน้ำเงิน
- ข้อความในบรรทัดที่ 4 คือผลการใช้ TAG <TABLE> ซึ่งควบคุมการแสดงผลด้วยคำสั่ง CSS ให้มีเส้นขอบเป็นแบบ 3 มิติแบบมีร่องสีเหลือง
- กล่องข้อความในบรรทัดที่ 5 มีเส้นขอบเป็นแบบนูนขึ้นสีน้ำเงิน คือผลการใช้ TAG <INPUT> ซึ่งควบคุมการแสดงผลด้วยคำสั่ง CSS ให้มีเส้นขอบแบบนูนขึ้นสีน้ำเงิน
- ปุ่ม OK และ ปุ่ม CANCEL ในบรรทัดที่ 6 คือผลการใช้ CLASS Selector ชื่อ BTN ที่กำหนดให้มีเส้นขอบด้านนอกสี่เหลี่ยมกว้าง 5PX และพื้นที่ที่แสดงปุ่มมีเส้นขอบด้านในเป็นสีชมพู คือผลการใช้ ID Selector

### 3.4 การกำหนด Property ควบคุมระยะห่างการแสดงผล

การกำหนดระยะห่างของการแสดงผลในหน้าเว็บเพจนั้น สามารถกำหนดได้สองรูปแบบคือการกำหนดระยะห่างระหว่างวัตถุอื่นที่แสดงในหน้าเว็บเพจหรือระยะห่างจากขอบเบราว์เซอร์ จะใช้การกำหนด Property คือ MARGIN และการกำหนดระยะห่างระหว่างเนื้อหากับพื้นที่ที่แสดงผล จะใช้การกำหนด Property คือ PADDING ดังตารางที่ 6.34

ตารางที่ 6.4 การกำหนด Property ให้กับคำสั่ง CSS ควบคุมระยะห่างการแสดงผล

| Property   | ค่าที่กำหนดให้           | คำอธิบาย  |
|------------|--------------------------|---|
| MARGIN-TOP | ค่าตัวเลข/<br>AUTO/<br>% | ระยะห่างด้านบน สามารถกำหนดค่าเป็นค่าตัวเลข หรือ AUTO หมายถึงอัตโนมัติ หรือ % หมายถึงระยะห่างเป็นค่าร้อยละ |

ตารางที่ 6.4 (ต่อ)

| Property       | ค่าที่กำหนดให้             | คำอธิบาย   |
|----------------|----------------------------|--|
| MARGIN-RIGHT   | ค่าตัวเลข/<br>AUTO/<br>%   | ระยะห่างด้านขวาสามารถกำหนดค่าเป็นค่าตัวเลข หรือ AUTO หมายถึงอัตโนมัติ หรือ % หมายถึงระยะห่างเป็นค่าร้อยละ  |
| MARGIN-BOTTOM  | ค่าตัวเลข/<br>AUTO/<br>%/% | ระยะห่างด้านล่าง สามารถกำหนดค่าเป็นค่าตัวเลข หรือ AUTO หมายถึงอัตโนมัติ หรือ % หมายถึงระยะห่างเป็นค่าร้อยละ  |
| MARGIN-LEFT    | ค่าตัวเลข/<br>AUTO/<br>%   | ระยะห่างด้านซ้าย สามารถกำหนดค่าเป็นค่าตัวเลข หรือ AUTO หมายถึงอัตโนมัติ หรือ % หมายถึงระยะห่างเป็นค่าร้อยละ  |
| MARGIN         | ค่าตัวเลข/<br>AUTO/<br>%   | กำหนดระยะห่างแบบกลุ่ม คือระยะห่างด้านบน ด้านขวา ด้านล่างและด้านซ้าย สามารถกำหนดค่าเป็นค่าตัวเลข หรือ AUTO หมายถึงอัตโนมัติ หรือ % หมายถึงระยะห่างเป็นค่าร้อยละ |
| PADDING-TOP    | ค่าตัวเลข /<br>%           | ระยะจากขอบบน กำหนดค่าเป็นค่าตัวเลข หรือ % หมายถึงระยะห่างเป็นค่าร้อยละ   |
| PADDING-RIGHT  | ค่าตัวเลข /<br>%           | ระยะจากขอบขวา กำหนดค่าเป็นค่าตัวเลข หรือ % หมายถึงระยะห่างเป็นค่าร้อยละ  |
| PADDING-BOTTOM | ค่าตัวเลข /<br>%           | ระยะจากขอบล่าง กำหนดค่าเป็นค่าตัวเลข หรือ % หมายถึงระยะห่างเป็นค่าร้อยละ   |
| PADDING-LEFT   | ค่าตัวเลข /<br>%           | ระยะขอบซ้าย กำหนดค่าเป็นค่าตัวเลข หรือ % หมายถึงระยะห่างเป็นค่าร้อยละ  |
| PADDING        | ค่าตัวเลข /<br>%           | กำหนดระยะจากขอบแบบกลุ่มทั้งสี่ด้าน โดยเรียงลำดับคือขอบบน ขอบขวา ขอบล่างและขอบซ้าย กำหนดค่าเป็นค่าตัวเลข หรือ % หมายถึงระยะห่างเป็นค่าร้อยละ                    |

จากตารางที่ 6.4 การกำหนด Property ควบคุมระยะห่าง ตัวอย่างเช่น

- หน้าเว็บเพจมีเส้นขอบเป็นแบบจุดสี่เหลี่ยมและมีระยะห่างจากขอบบน 100PX



- ข้อความใน TAG <H1> มีลักษณะเส้นขอบเป็นแบบขีด ข้อความห่างจากขอบทั้งสี่ด้าน 100PX และกำหนดตำแหน่งการจัดวางข้อความให้อยู่ตรงกลาง
  - ข้อความใน TAG <DIV> ลักษณะเส้นขอบเป็นแบบทึบ และมีระยะห่างจากขอบเบรราวเซอร์ด้านซ้ายเป็นร้อยละ 10 จากความกว้างของขอบเบรราวเซอร์ที่แสดงผล
  - รูปภาพ TAG <IMG> ลักษณะเส้นขอบเป็นแบบคู่สีน้ำเงินและการแสดงผลรูปให้มีระยะห่างจากขอบทั้งสี่ด้านเป็น 10PX
  - ตารางใน TAG <TABLE> ลักษณะเส้นขอบเป็นแบบ 3 มิติแบบมีร่องสีเหลือง และห่างจากขอบเบรราวเซอร์ด้านซ้าย 10PX
  - คอลัมน์ตารางใน TAG <TD> ข้อความห่างจากขอบแต่ละด้าน 10PX
  - ก่อ่งข้อความและปุ่มใน TAG <INPUT> ลักษณะเส้นขอบเป็นแบบนูนขึ้นสีน้ำเงินและข้อความห่างจากขอบแต่ละด้าน 10PX
- สามารถสร้างคำสั่ง CSS ดังรูปที่ 6.40

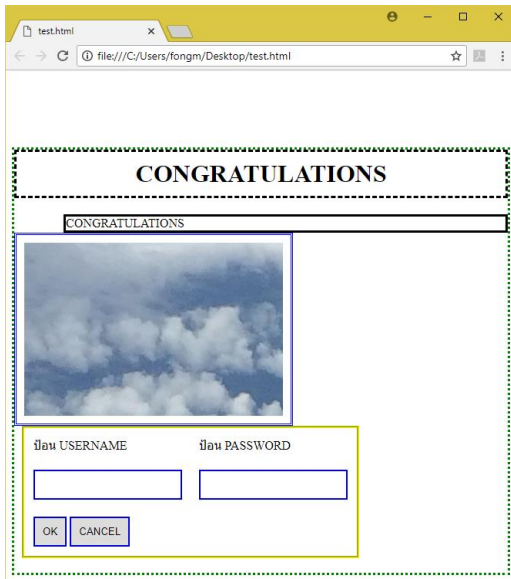
```

<HTML><HEAD><STYLE>
  BODY{BORDER-STYLE:DOTTED;BORDER-COLOR:GREEN;MARGIN-TOP:100PX;}
  H1 {BORDER-STYLE:DASHED;PADDING:10PX;TEXT-ALIGN:CENTER;}
  DIV {BORDER-STYLE:SOLID;MARGIN-LEFT:10%;}
  IMG {BORDER-STYLE:DOUBLE;BORDER-COLOR:BLUE;PADDING:10PX;}
  TABLE {BORDER-STYLE:GROOVE;BORDER-COLOR:YELLOW;MARGIN-LEFT:10PX;}
  TD {PADDING:10PX;}
  INPUT {BORDER-STYLE:RIDGE;BORDER-COLOR:BLUE;PADDING:10PX;}
</STYLE></HEAD><BODY>
  <H1>CONGRATULATIONS </H1> <DIV>CONGRATULATIONS</DIV> <IMG
  SRC="IMG.JPG">
  <TABLE> <TR><TD>ป้อน USERNAME </TD><TD>ป้อน PASSWORD </TD></TR>
    <TR><TD><INPUT TYPE="TEXT" NAME="USERNAME"></TD>
      <TD><INPUT TYPE="PASSWORD" NAME="PASS"></TD></TR>
  <TR><TD ID="CELL" COLSPAN="2">
    <INPUT TYPE="SUBMIT" CLASS="BTN" VALUE="OK">
    <INPUT TYPE="RESET" CLASS="BTN" VALUE="CANCEL"></TD></TR>
</TABLE></BODY></HTML>

```

รูปที่ 6.40 ตัวอย่างการคำสั่ง CSS กำหนด Property ควบคุมระยะห่างการแสดงผล

จากรูปที่ 6.40 เมื่อแสดงผลผ่านทางเบราว์เซอร์ จะได้ผลลัพธ์ดังรูปที่ 6.41



รูปที่ 6.41 การแสดงผลบนเบราว์เซอร์จากไฟล์ตัวอย่างรูปที่ 6.40

จากรูปที่ 6.41 พบว่า

- หน้าเว็บเพจมีเส้นขอบเป็นแบบจุดสี่เหลี่ยมและมีระยะห่างจากขอบบน 100PX
- ข้อความในบรรทัดที่ 1 มีเส้นขอบเป็นแบบขีดและมีระยะห่างจากขอบทั้งสี่ด้านเท่ากับ 10PX คือผลจากการใช้ TAG <H1>
- ข้อความในบรรทัดที่ 2 มีเส้นขอบเป็นแบบทึบและห่างจากขอบเบราว์เซอร์ด้านซ้ายร้อยละ 10 ของความกว้างเบราว์เซอร์ที่แสดงผล คือผลจากการใช้ TAG <DIV>
- รูปภาพในบรรทัดที่ 3 มีเส้นขอบเป็นแบบคู่สีน้ำเงินและรูปภาพห่างจากขอบทั้งสี่ด้าน 10PX คือผลจากการใช้ TAG <IMG>
- ข้อความในบรรทัดที่ 4 คือผลการใช้ TAG <TABLE> และ TAG <TD> ที่มีการกำหนดระยะห่างจากขอบเบราว์เซอร์ด้านซ้าย 10PX และข้อความข้างในคอลัมน์ห่างจากขอบแต่ละด้าน 10PX
- กล่องข้อความและปุ่มในบรรทัดที่ 5 และบรรทัดที่ 6 มีเส้นขอบเป็นแบบนูนขึ้นสีน้ำเงินและข้อความห่างจากขอบแต่ละด้าน 10PX คือผลการใช้ TAG <INPUT>

### 3.5 การกำหนด Property ควบคุมคุณสมบัติการแสดงผล

การกำหนดคุณสมบัติการแสดงผล สามารถกำหนด Property ดังตารางที่ 6.5

ตารางที่ 6.5 การกำหนด Property ให้กับคำสั่ง CSS ควบคุมคุณสมบัติการแสดงผล

| Property | ค่าที่กำหนดให้   | คำอธิบาย   |
|----------|--|--|
| DISPLAY  | INLINE/<br>BLOCK/<br>INLINE-BLOCK/<br>NONE/<br>INITIAL/<br>INHERIT | คุณสมบัติการแสดงผล สามารถกำหนดเป็น<br>INLINE หมายถึง แนวเดียวกับข้อความ<br>BLOCK หมายถึง แบบเป็นกล่อง<br>INLINE-BLOCK หมายถึง แบบกล่องและแนว<br>เดียวกับข้อความ<br>NONE หมายถึง ไม่แสดงผล<br>INITIAL หมายถึง ค่าเริ่มต้นของเบราว์เซอร์<br>INHERIT หมายถึง สืบทอดคุณสมบัติจาก<br>องค์ประกอบหลัก |

จากตารางที่ 6.5 ตัวอย่างกำหนดข้อความใน TAG <DIV> พื้นหลังเป็นสีเขียว ข้อความห่างจากขอบทั้งสี่ด้าน 20PX จัดวางข้อความอยู่ตรงกลาง ขนาดตัวอักษร 20PX และขอบบนห่างจากวัตถุอื่น 10PX กำหนดคุณสมบัติการแสดงผลให้กับ ข้อความที่ 1 และ 2 เป็นแบบ INLINE ข้อความที่ 3 และ 4 เป็นแบบ BLOCK ข้อความที่ 5 และ 6 เป็นแบบ INLINE-BLOCK สามารถสร้างคำสั่ง CSS ดังรูปที่ 6.42

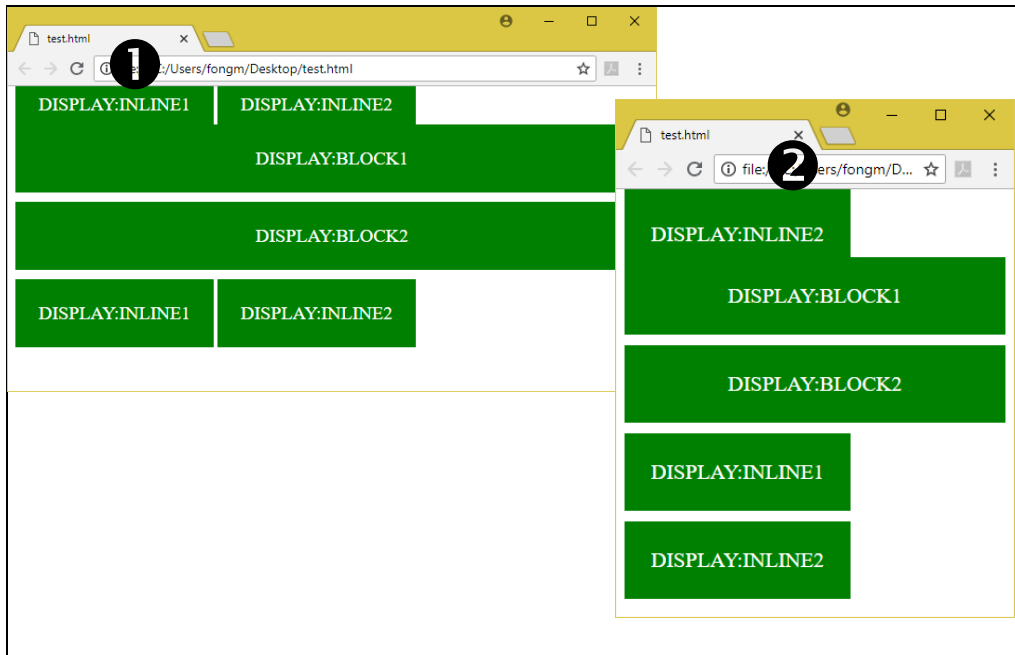
```

<HTML><HEAD><STYLE>DIV{COLOR:WHITE;BACKGROUND-COLOR:
GREEN;PADDING: 25PX;TEXT-ALIGN: CENTER;FONT-SIZE:20PX;MARGIN-TOP:10PX;}
.INLINE{DISPLAY:INLINE;} .BLOCK{DISPLAY:BLOCK;}
.INLINEBLOCK{DISPLAY:INLINE-BLOCK;}</STYLE></HEAD><BODY>
<DIV CLASS="INLINE">DISPLAY:INLINE1</DIV>
<DIV CLASS="INLINE">DISPLAY:INLINE2</DIV>
<DIV CLASS="BLOCK">DISPLAY:BLOCK1</DIV>
<DIV CLASS="BLOCK">DISPLAY:BLOCK2</DIV>
<DIV CLASS="INLINEBLOCK">DISPLAY:INLINE1</DIV>
<DIV CLASS="INLINEBLOCK">DISPLAY:INLINE2</DIV></BODY></HTML>

```

รูปที่ 6.42 ตัวอย่างการคำสั่ง CSS กำหนด Property ควบคุมคุณสมบัติการแสดงผล

จากรูปที่ 6.42 เมื่อแสดงผลผ่านทางเบราว์เซอร์ จะได้ผลลัพธ์ดังรูปที่ 6.43



รูปที่ 6.43 การแสดงผลบนเบราว์เซอร์จากไฟล์ตัวอย่างรูปที่ 6.42

จากรูปที่ 6.43 พบว่าหมายเลข ❶ ข้อความที่ 1 และ 2 จะแสดงผลอยู่ในบรรทัดเดียวกัน เมื่อแสดงผลข้อความที่ 3 และ 4 ปรากฏว่าจะมีระยะที่ติดกันกับข้อความที่ 1 และ 2 โดยข้อความที่ 3 และ 4 จะแสดงเป็นแบบกล่องคือจะแสดงเต็มบรรทัด เมื่อแสดงผลข้อความถัดไปจะขึ้นบรรทัดใหม่ ส่วนข้อความที่ 5 และ 6 จะมีระยะห่างจากข้อความข้างบนและแสดงในบรรทัดเดียวกัน ส่วนหมายเลข ❷ คือเมื่อปรับลดขนาดหน้าจอเบราว์เซอร์พบว่าข้อความที่ 1 จะหายไปเหลือแต่ข้อความที่ 2 เนื่องจากข้อความที่ 1 และ 2 แสดงในตำแหน่งเดียวกันจึงเห็นเพียงแต่ข้อความที่ 2 เท่านั้น ส่วนข้อความที่ 5 และ 6 จะแสดงแยกบรรทัดกัน ไม่มีการทับซ้อนบรรทัดกันเหมือนข้อความที่ 1 และ 2 ซึ่งเหมาะสำหรับการกำหนดคุณสมบัติการแสดงผลให้เหมาะสมกับทุกขนาดหน้าจอ

#### 4. CSS Pseudo-classes

คือ Class เทียมเป็น Class ที่มีชื่อเฉพาะประกอบด้วย Class ชื่อ active, focus, hover, link, visited, checked และ disabled ซึ่งจะเป็นคุณสมบัติเฉพาะของ TAG <A> ที่มีไว้สำหรับการเชื่อมโยงหน้าเว็บเพจ โดยมีรูปแบบการใช้งานคือ

Selector:pseudo-class {property:value}

Pseudo class ประกอบด้วย

:active คือสถานะที่รายการเชื่อมโยงกำลังถูกคลิกเลือก

:focus คือสถานะที่รายการเชื่อมโยงระบุกำลังถูกโฟกัส

:hover คือสถานะที่นำเมาส์ไปวางรายการเชื่อมโยง นอกจากนี้ ยังสามารถนำไปใช้ได้กับ TAG HTML อื่นๆ เช่น <H1>ถึง <H6> , <DIV>, <IMG>, <INPUT> เป็นต้น

:link คือรายการเชื่อมโยงก่อนการคลิกการเชื่อมโยง

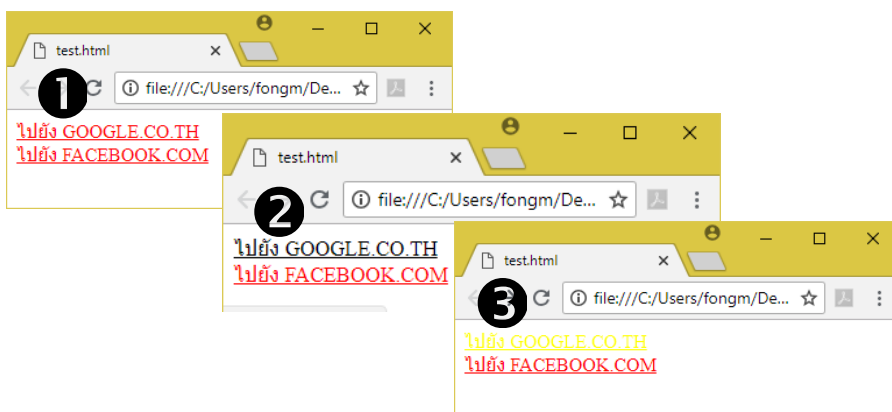
:visited คือรายการเชื่อมโยงหลังจากมีการคลิกการเชื่อมโยง

ตัวอย่างข้อความใน TAG <A> ก่อนคลิกเชื่อมโยงสีตัวอักษรเป็นสีแดง หลังคลิกเชื่อมโยงสีตัวอักษรเป็นสีเหลือง เมื่อนำเมาส์ไปวางบนข้อความให้สีตัวอักษรสีเขียว ขณะที่คลิกเลือกรายการให้สีตัวอักษรเป็นสีชมพู และขณะที่กำลังโฟกัสให้มีสีตัวอักษรเป็นสีดำ สามารถสร้างคำสั่ง CSS ดังรูปที่ 6.44

```
<HTML><HEAD><STYLE>
  A:LINK{COLOR:RED;};      A:VISITED{COLOR:YELLOW;};
  A:HOVER{COLOR:GREEN;};  A:ACTIVE{COLOR:PINK;};
</STYLE></HEAD><BODY>
  <A HREF="HTTP://WWW.GOOGLE.CO.TH">ไปยัง GOOGLE.CO.TH</A><BR>
  <A HREF="HTTP://WWW.FACEBOOK.COM">ไปยัง FACEBOOK.COM</A>
</BODY></HTML>
```

รูปที่ 6.44 ตัวอย่างการคำสั่ง CSS การใช้งาน Pseudo-classes

จากรูปที่ 6.44 เมื่อแสดงผลผ่านทางเบราว์เซอร์ จะได้ผลลัพธ์ดังรูปที่ 6.45



รูปที่ 6.45 การแสดงผลบนเบราว์เซอร์จากไฟล์ตัวอย่างรูปที่ 6.44

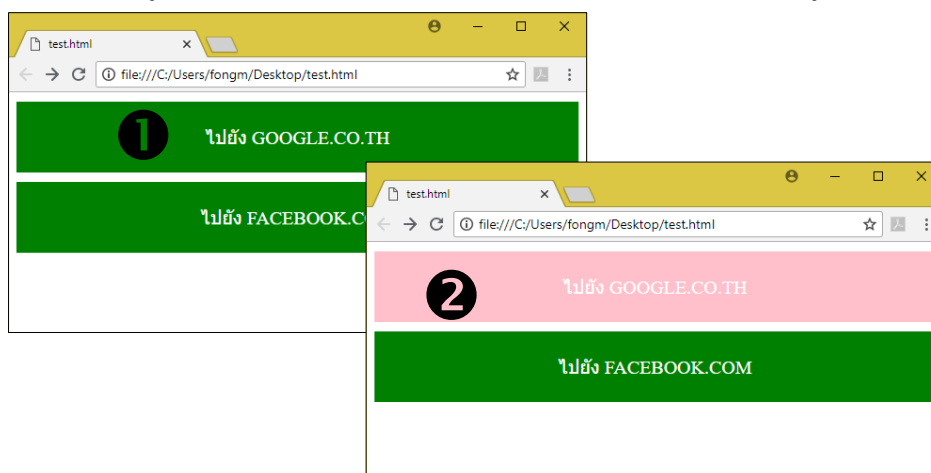
จากรูปที่ 6.45 พบว่า เมื่อแสดงผลก่อนการคลิกเชื่อมโยงข้อความในหมายเลข ❶ มีสีตัวอักษรสีแดง เมื่อนำเมาส์ไปวางขณะที่กำลังคลิกการเชื่อมโยงในหมายเลข ❷ จะมีสีเป็นสีดำเนื่องจากอยู่ในสถานะโฟกัส และเมื่อคลิกเชื่อมโยงเรียบร้อยแล้วข้อความจะเปลี่ยนเป็นสีเหลืองในหมายเลข ❸

นอกจากนี้ Class Hover ยังสามารถใช้กับ TAG HTML อื่นๆ ตัวอย่างสร้างปุ่มนำทางด้วยการใช้ TAG <A> เชื่อมโยงไปยังหน้าเว็บเพจอื่น ตัวหนังสือสีขาว ไม่แสดงขีดเส้นใต้ข้อความ ภายใน TAG <A> แสดง TAG <DIV> พื้นหลังเป็นสีเขียว ข้อความห่างจากขอบทั้งสี่ด้าน 20PX จัดวางข้อความอยู่ตรงกลาง ขนาดตัวอักษร 20PX และขอบบนห่างจากวัตถุอื่น 10PX เมื่อนำเมาส์ไปวางให้พื้นหลังเป็นสีชมพู สามารถสร้างคำสั่ง CSS ดังรูปที่ 6.46

```
<HTML><HEAD><STYLE>
  DIV{BACKGROUND-COLOR: GREEN;PADDING: 25PX;
  TEXT-ALIGN: CENTER;FONT-SIZE:20PX;MARGIN-TOP:10PX;}
  DIV:HOVER {BACKGROUND-COLOR: PINK;}
  A{TEXT-DECORATION:NONE;COLOR:WHITE;} </STYLE></HEAD><BODY>
<DIV><A HREF="HTTP://WWW.GOOGLE.CO.TH" TARGET="_BLANK">
  ไปยัง GOOGLE.CO.TH</A></DIV>
<DIV><A HREF="HTTP://WWW.FACEBOOK.COM" TARGET="_BLANK">
  ไปยัง FACEBOOK.COM</A></DIV></BODY></HTML>
```

รูปที่ 6.46 ตัวอย่างการคำสั่ง CSS สร้างปุ่มนำทาง

จากรูปที่ 6.46 เมื่อแสดงผลผ่านทางเบราว์เซอร์ จะได้ผลลัพธ์ดังรูปที่ 6.47



รูปที่ 6.47 การแสดงผลบนเบราว์เซอร์จากไฟล์ตัวอย่างรูปที่ 6.46

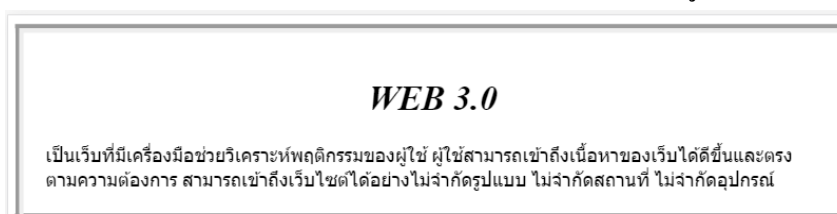
จากรูปที่ 6.47 พบว่า เมื่อแสดงผลก่อนการคลิกเชื่อมโยงข้อความในหมายเลข ❶ เมื่อนำเมาส์ไปวางพื้นหลังจะเปลี่ยนเป็นสีชมพูดังหมายเลข ❷ ซึ่งเป็นผลจากการใช้คำสั่ง CSS ที่ควบคุมการ HOVER

## 5. บทสรุป

การจัดรูปแบบการแสดงผลเว็บเพจด้วย CSS นั้นจะช่วยอำนวยความสะดวกในการจัดการรูปแบบการแสดงผลของหน้าเว็บเพจ เนื่องจากคำสั่ง CSS สามารถครอบคลุมหน้าเว็บเพจได้มากกว่าการกำหนด Attribute ให้กับ TAG HTML โดยตรง อีกทั้งยังทำให้ไฟล์เว็บเพจมีขนาดเล็กลง ประหยัดพื้นที่หน่วยความจำในการจัดเก็บไฟล์ และใช้เวลาในการดาวน์โหลดน้อย และยังสามารถรองรับการแสดงผลได้ทุกเบราว์เซอร์

## 6. แบบฝึกหัดท้ายบท

1. สร้างเว็บเพจเพื่อแสดงข้อความโดยใช้ TAG <DIV> โดยกำหนดให้ข้อความห่างจากขอบทั้งสี่ด้าน 25PX ขนาดตัวอักษร 20PX มีระยะห่างจากขอบบน 10PX ลักษณะเส้นขอบเป็นแบบ 3 มิติแบบมีร่องเส้นขอบกว้าง 10PX และด้านใน TAG <DIV> ประกอบด้วย TAG <H1> กำหนดรูปแบบตัวอักษรเป็นตัวเอียง จัดวางข้อความอยู่ตรงกลางหน้าเว็บเพจ เมื่อแสดงผลผ่านทางเบราว์เซอร์จะได้ผลลัพธ์ดังรูปที่ 6.48



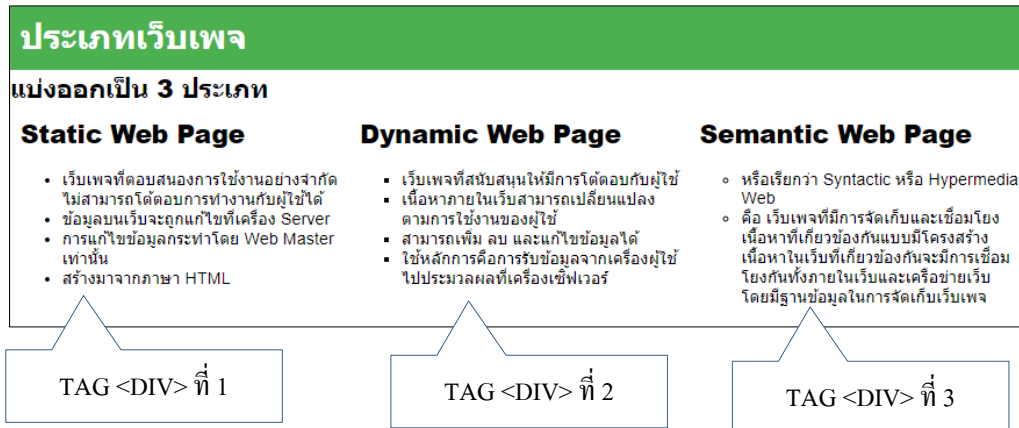
รูปที่ 6.48 การแสดงผลบนเบราว์เซอร์จากแบบฝึกหัดท้ายบทข้อที่ 1

2. สร้างเว็บเพจแสดงข้อความ

- ใช้ TAG <H1> เพื่อแสดงข้อความ ประเภทของเว็บเพจ โดยกำหนดให้มีพื้นหลังเป็นสีเขียว

- ใช้ TAG <H2> เพื่อแสดงข้อความ แบ่งออกเป็น 3 ประเภท

- ใช้ TAG <DIV> ข้างในประกอบด้วย TAG <H1> แสดงข้อความ Static Web Page และใช้ TAG <UL> เพื่อแสดงข้อความ จำนวน 3 <TAG> โดยกำหนดคุณสมบัติการแสดงผลเป็นแบบ BLOCK เมื่อแสดงผลผ่านทางเบราว์เซอร์ จะได้ผลลัพธ์ดังรูปที่ 6.49



รูปที่ 6.49 การแสดงผลบนเบราว์เซอร์จากแบบฝึกหัดท้ายบทข้อที่ 2

3. สร้างปุ่มนำทางเพื่อเชื่อมโยงหน้าเว็บเพจจากข้อที่ 1 และข้อที่ 2 กำหนดรูปแบบการแสดงผลด้วยคำสั่ง CSS ตามความเหมาะสม