ใบความรู้ที่ 2.2

หน่วยการเรียนรู้ที่ 2 เรื่อง การสร้างแบบฟอร์ม (Form) **ภาคเรียนที่ 1 ปีการศึกษา 2560**

เว็บเพจเป็นสื่อที่แตกต่างจากเอกสารธรรมดา เพราะเป็นสื่อสองทางที่ผู้ใช้สามารถสื่อสารกับเจ้าของเว็บเพจได้ ปัจจุบันมีการใช้แบบฟอร์มในเว็บเพื่อเก็บข้อมูล ตั้งแต่รายละเอียดเกี่ยวกับผู้ที่มาชมเว็บ การสั่งซื้อสินค้า ไปจนถึงการรวบรวมความคิดเห็นเกี่ยวกับเรื่องต่างๆ ที่กำลังอยู่ในความสนใจเช่น การเลือกตั้ง การลงคะแนนความคิดเห็น เป็นต้น

ช่องทางการรับข้อมูลต่างๆ ใน HTML

ฟอร์ม (Form)

โดยปกติก่อนที่จะสร้างช่องรับข้อมูลต่างๆ เราจะต้องสร้างฟอร์มก่อน ซึ่งฟอร์มจะทำหน้าที่เป็นที่รวมของช่องข้อมูลต่างๆ และท่าสำคัญฟอร์มยังทำหน้าที่กำหนดวิธีการส่งข้อมูล การระบุไฟล์ที่จะส่งข้อมูลไปประมวลผล

รูปแบบโค้ด HTML ของฟอร์มเป็นดังนี้

```
<form name='ชื่อฟอร์ม' method='วิธีการส่ง' action='ไฟล์ที่รับข้อมูลจากฟอร์ม'>
    ...
    ช่องรับข้อมูลต่างๆ
    ...
</form>
```

โดยที่

name='ชื่อฟอร์ม' ระบุชื่อของฟอร์ม ในส่วนนี้อาจจะไม่ระบุก็ได้(ในกรณีที่ได้ต้องการอ้างอิงตำแหน่งฟอร์ม)

method='วิธีส่งข้อมูล' กำหนดวิธีส่งค่าข้อมูลไปยังเซิร์ฟเวอร์ ซึ่งมีอยู่ 2 ประเภท คือ

- get : เป็นวิธีการส่งข้อมูลที่จะนำค่าของช่องรับข้อมูลรวมไปกับ URL
- post : เป็นวิธีส่งค่าข้อมูลที่ต้องการให้เป็นความลับ คือจะไม่เห็นค่าของช่องรับข้อมูล

action='ไฟล์ที่รับข้อมูลจากฟอร์ม' เป็นการระบุชื่อไฟล์ที่จะส่งไปเพื่อทำการประมวลผล โดยนำค่าจากช่องรับข้อมูลไปประมวลผลดังไฟล์นี้

คุณสมบัติของ <form>

action : ระบุชื่อไฟล์ที่ฟอร์มนี้จะส่งข้อมูลไปประมวลผล

method : ระบุวิธีการส่งข้อมูล (POST หรือ GET)

enctype : กำหนดประเภทของข้อมูล เช่น ถ้าเป็นการอัพโหลดไฟล์ก็จะกำหนด enctype เป็น multipart/form-data

ช่องทางรับข้อมูลแท็ก <input>

เป็นการสร้างช่องทางรับข้อมูลประเภท input โดยแท็กนี้ มีช่องทางรับข้อมูลหลายชนิด เช่น การสร้างกรอบป้อนข้อความบรรทัดเดียว (Text) ช่องป้อนรหัสผ่าน (Password) ปุ่มแบบวิทยุ (Radio) ช่องทำเครื่องหมาย (Checkbox) ปุ่มยืนยันข้อมูล (Submit) ปุ่มยกเลิกข้อมูล (Reset)

ช่องรับข้อความ(Text field)

เป็นช่องที่ให้ผู้ใช้กรอกข้อมูล หรือรายละเอียด เช่น เป็นช่องรับชื่อ-นามสกุล รับข้อมูลที่เป็นข้อความสั้นๆ รูปแบบโค้ด HTML ของ text ดังนี้

```
<form method='post' action='#'>
    <input type='text' name='ชื่อของ text field'>
</form>
```

โดยที่

type='text' ระบุว่าเป็นช่องรับข้อความ

name='ชื่อของ text field' ระบุชื่อของช่องรับข้อความ ให้ตั้งชื่อให้สื่อความหมายที่จะนำข้อมูลเข้า เช่น หาก text field รับนามสกุล ก็ให้ตั้งชื่อว่า surname เป็นต้น และเมื่อตั้งชื่อช่องรับข้อมูลแล้ว จะห้ามตั้งซ้ำกับช่องทางอื่นๆ กรณีที่มีหลายช่องรับข้อความ

คุณสมบัติของ <input type='text'>

name : เป็นชื่อของแท็ก <input> เพื่อใช้อ้างอิงตำแหน่งในตอนรับข้อมูล

size : เป็นจำนวนขนาด(ความยาว)ของช่องรับข้อความค่าเริ่มต้นกรณีที่ไม่กำหนดเท่ากับ 20 ตัวอักษร

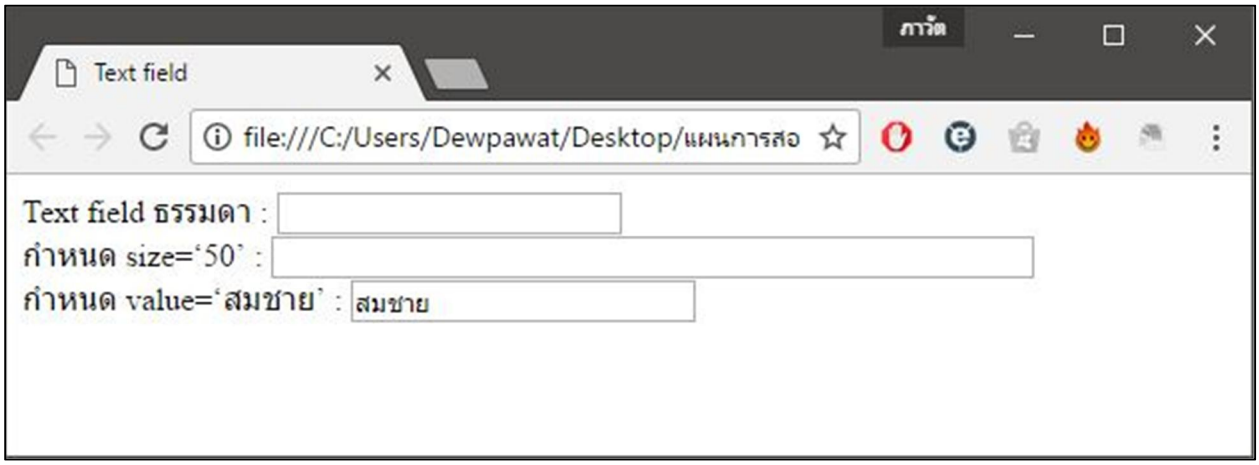
maxlength : เป็นจำนวนตัวอักษรที่สามารถกรอกเข้าไปได้สูงสุดในช่องรับ

value : ข้อความที่อยู่ระหว่างเครื่องหมายคำพูดจะปรากฏอยู่ในช่อรับข้อมูลให้โดยอัตโนมัติ หากเป็นการรับข้อมูลจากผู้ใช้มักจะไม่ระบุ

ตัวอย่างช่องรับข้อความ

```
<form method='post' action='#'>
        Text field ธรรมดา : <input type='text' name='name'><br>
        กำหนด size='50' : <input type='text' name='name' size='50'><br>
        กำหนด value='สมชาย' : <input type='text' name='name' value='สมชาย'>
</form>
```

ผลลัพธ์ที่ได้



ช่องรับข้อความเข้ารหัส (Password)

เป็นช่องรับข้อมูลที่ผู้ใช้ ใช้ในกรณีต้องการปิดบังข้อมูล เช่น การเข้า login ที่ต้องใช้ user และ password เพื่อป้องกันการเข้าถึงข้อมูลส่วนตัวเป็นต้น รูปแบบโค้ด HTML ของ password ดังนี้

```
<form method='post' action='#'>
        <input type='password' name='ชื่อของ password field'>
</form>
```

โดยที่

type='password' ระบุว่าเป็นช่องรับข้อความเข้ารหัส

name=‘ชื่อของ password field’ ระบุชื่อของช่องรับข้อความเข้ารหัส ให้ตั้งชื่อให้สื่อความหมายที่จะนำข้อมูลเข้า และเมื่อตั้งชื่อช่องรับข้อมูลแล้ว จะห้ามตั้งซ้ำกับช่องทางอื่นๆ กรณีที่มีหลายช่องรับข้อความ

คุณสมบัติของ <input type=‘password>

name : เป็นชื่อของแท็ก <input> เพื่อใช้อ้างอิงตำแหน่งในตอนรับข้อมูล

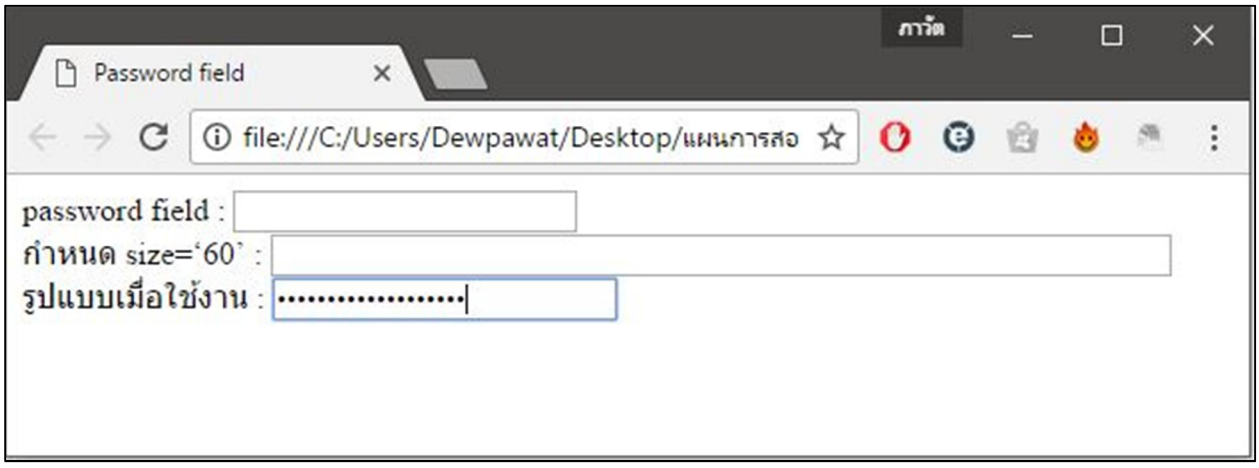
size : เป็นจำนวนขนาด(ความยาว)ของช่องรับข้อความค่าเริ่มต้นกรณีที่ไม่กำหนดเท่ากับ 20 ตัวอักษร

maxlength : เป็นจำนวนตัวอักษรที่สามารถกรอกเข้าไปได้สูงสุดในช่องรับ

ตัวอย่างช่องรับข้อความเข้ารหัส

```
<form method=‘post’ action=‘#’>
        password field : <input type=‘password’ name=‘pass’><br>
        กำหนด size=‘60’ : <input type=‘password’ name=‘pass’ size=‘60’><br>
        รูปแบบเมื่อใช้งาน : <input type=‘password’ name=‘pass’>
</form>
```

ผลลัพธ์ที่ได้



การสร้างปุ่ม (submit, reset, button)

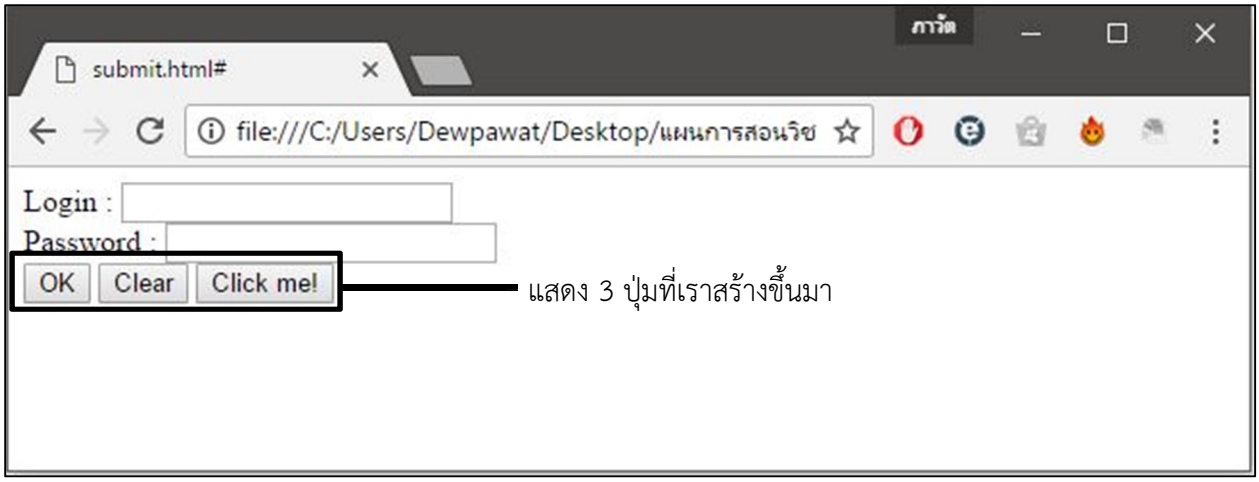
ในฟอร์มปกติจะต้องมีปุ่มให้กด ซึ่งปุ่มในฟอร์ม HTML จะมี 3 ลักษณะ คือ

1. ปุ่ม submit เมื่อกดปุ่มนี้ ข้อมูลในฟอร์มทั้งหมดจะไปตามที่เรากำหนดไว้ในคุณสมบัติ action ของฟอร์ม
2. ปุ่ม reset เมื่อกดปุ่มนี้ช่องทางนำเข้าข้อมูลทั้งหมดในฟอร์มจะกลับสู่ค่าเริ่มต้น ก่อนที่เราจะพิมพ์ข้อมูลใดๆ ลงฟอร์ม
3. ปุ่ม button เป็นปุ่มที่สร้างขึ้นมาเพื่อทำงานใดๆ โดยจะไปเรียกใช้ JavaScript ขึ้นมาทำงาน เช่น เมื่อคลิกปุ่มจะแสดงกล่องข้อความขึ้นมา ซึ่งจะใช้ในกรณีที่เราต้องการใช้ภาษาสคริปต์ เช่น JavaScript หรือ VBScript

รูปแบบโค้ด HTML ของปุ่มต่างๆ ดังนี้

```
<form method='post' action='#'>
        Login : <input type='text' name='user'><br>
        Password : <input type='password' name='pass'><br>
        < input type='submit' value='OK'>
        < input type='reset' value='Clear'>
        < input type='button' value='Click me!' onclick='alert("Hello, how are you ??")'>
</form>
```

ผลลัพธ์ที่ได้คือ

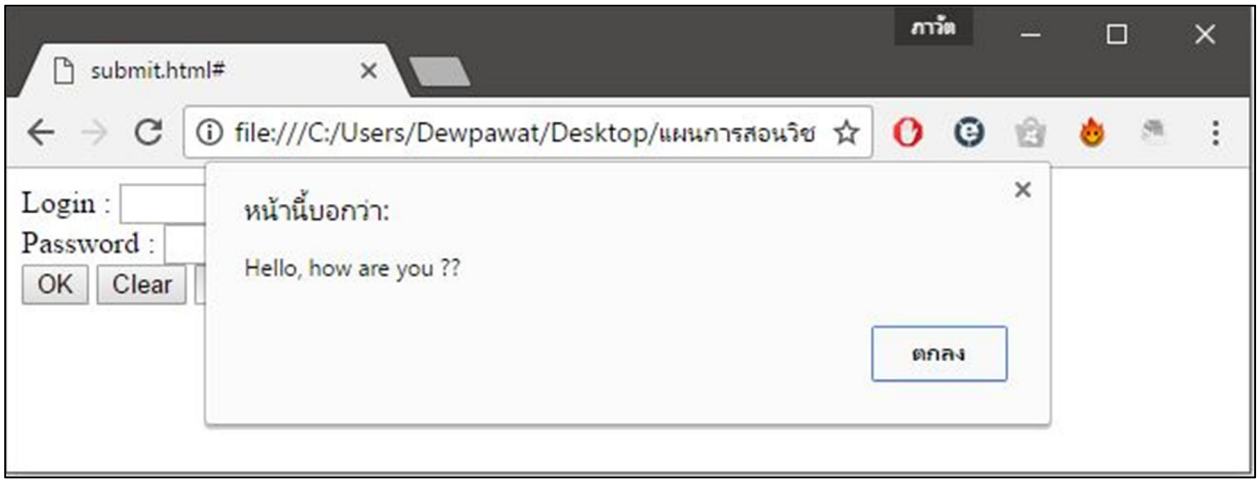


จากตัวอย่าง ถ้าเรากดปุ่ม OK ข้อมูลในฟอร์มของเราจะถูกส่งไปยังไฟล์ ที่เรากำหนดไว้ในคุณสมบัติ action ของแท็ก form จากตัวอย่างเราไม่ได้กดไว้ ก็จึงไม่มีการตอบสนองใดๆ การส่งข้อมูลไปที่ไฟล์อีกไฟล์นั้น จะต้องใช้ความรู้ในการเขียนโปรแกรมฝั่งเซิร์ฟเวอร์ เพื่อรับค่าจะฟอร์ม HTML แต่เนื้อหาในที่นี้ จะแนะนำเพียง HTML เท่านั้น

ถ้าเรากดปุ่ม Clear ข้อมูลที่กรอกลงอุปกรณ์ทั้งหมดจะกลับสู่ค่าเริ่มต้นก่อนที่จะมีการกรอกข้อมูล

ถ้าเรากดปุ่ม Click me! จะเรียกภาษาสคริปต์ให้ทำงาน จากตัวอย่างปุ่ม Click me! ถูกกดจะปรากฏหน้าต่างแสดงข้อความ “Hellow, How are you ??”

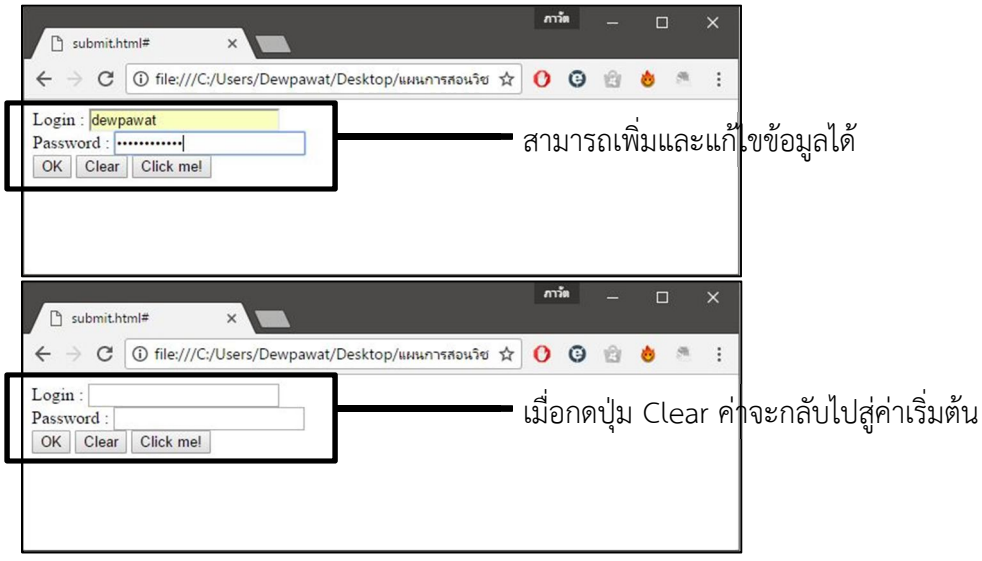
ตัวอย่างเมื่อกดปุ่ม Click me!



คุณสมบัติของ <input type=‘submit, reset, button’>

value : ค่าในคุณสมบัติ value จะใช้เป็นชื่อของปุ่มนั้นๆ ที่จะแสดงบนปุ่มเช่น ปุ่ม OK ปุ่ม Clear ในตัวอย่างที่ผ่านมา

name : ชื่อของปุ่มใช้ในการกำหนดตำแหน่งเพื่ออ้างอิงปุ่มนั้นๆ

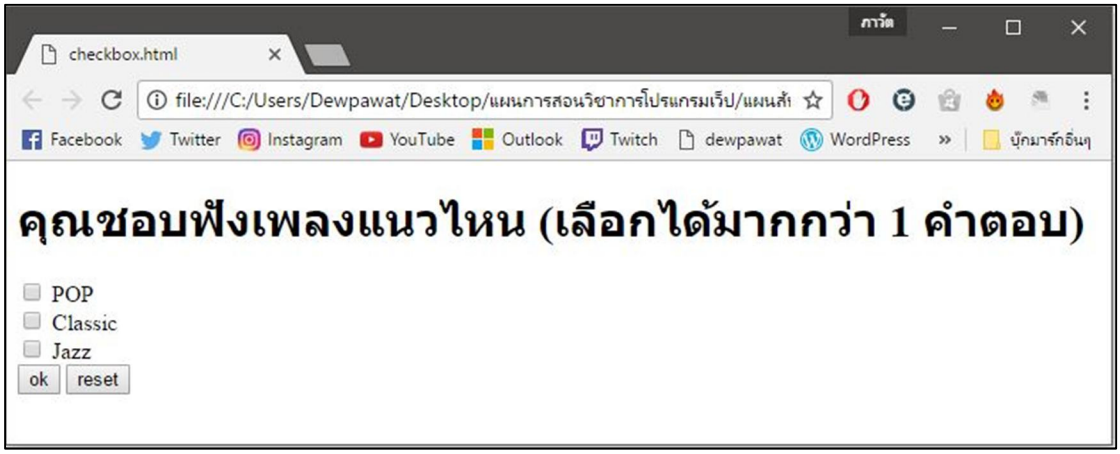


ตัวเลือกแบบให้เลือกหลายข้อ (checkbox)

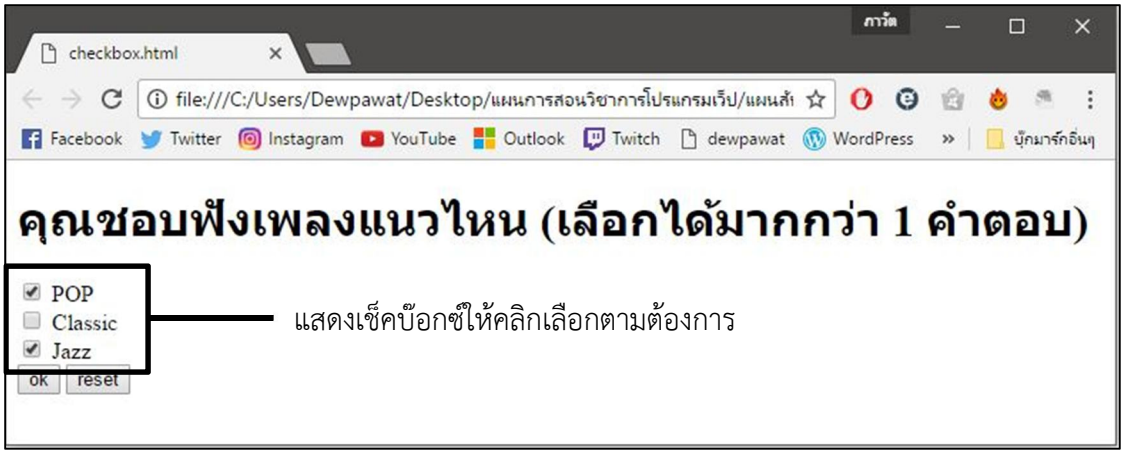
หากเราเคยทำแบบสอบถามทางอินเทอร์เน็ต ที่คำถามบอกว่าเราสามารถเลือกคำตอบได้มากกว่า 1 ข้อ คำถามแบบนี้จะใช้เช็คบ๊อกซ์ (Checkbox) ซึ่งจะมี 2 สถานะ คือ ใช่หรือไม่ เลือกหรือไม่เลือกเราจะสามารถใช้งานเช็คบ๊อกซ์ในกรณีที่ต้องการให้ผู้กรอกแบบฟอร์มเลือกได้หลายๆ ตัวเลือกพร้อมกันเช่น ประเภทกิจกรรมที่ชอบทำ ชื่อดาราคนโปรด เป็นต้น

```
<body>
        <h1>คุณชอบฟังเพลงแนวไหน (เลือกได้มากกว่า 1 คำตอบ)</h1>
        <form method='post' action='#'>
                <input type='checkbox' value='pop' name='song1'> POP <br>
                <input type='checkbox' value='classic' name='song2'> Classic <br>
                <input type='checkbox' value='jazz' name='song3'> Jazz <br>
                <input type='submit' value='ok'>
                <input type='reset' value='reset'>
        </form>
</body>
```

ผลลัพธ์ที่ได้



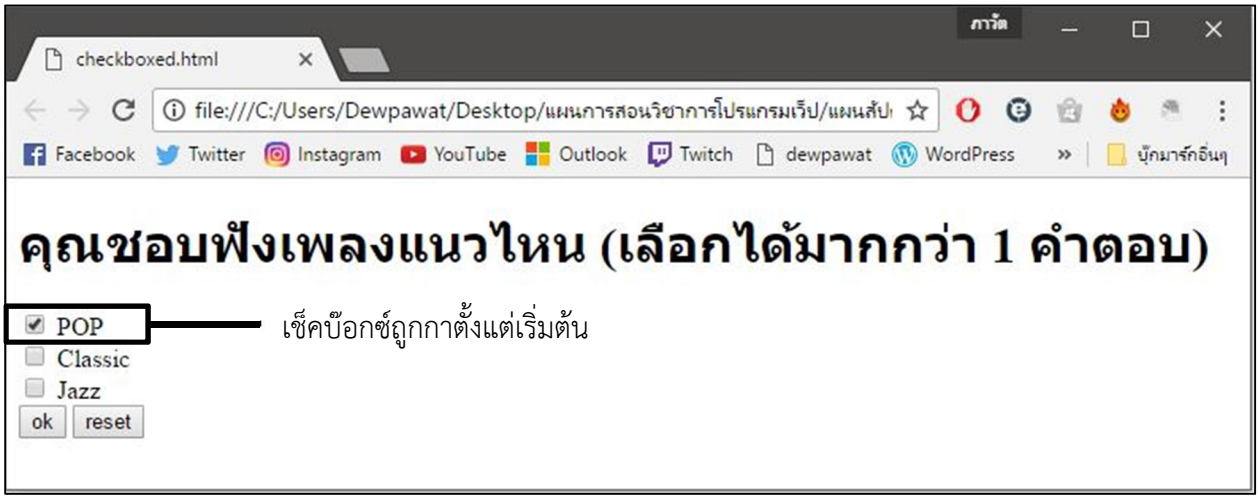
ตัวอย่างการเลือกแบบ Check box สามารถเลือกได้มากกว่า 1



เราสามารถกำหนดให้เช็คบ๊อกซ์ถูกกาตั้งแต่ค่าเริ่มต้นได้ โดยกำหนดคำว่า checked ที่ เช็คบ๊อกซ์นั้น ๆ

```
<body>
        <h1>คุณชอบฟังเพลงแนวไหน (เลือกได้มากกว่า 1 คำตอบ)</h1>
        <form method='post' action='#'>
                <input type='checkbox' value='pop' name='song1' checked> POP <br>
                <input type='checkbox' value='classic' name='song2'> Classic <br>
                <input type='checkbox' value='jazz' name='song3'> Jazz <br>
                <input type='submit' value='ok'>
                <input type='reset' value='reset'>
        </form>
</body>
```

ผลลัพธ์ที่ได้



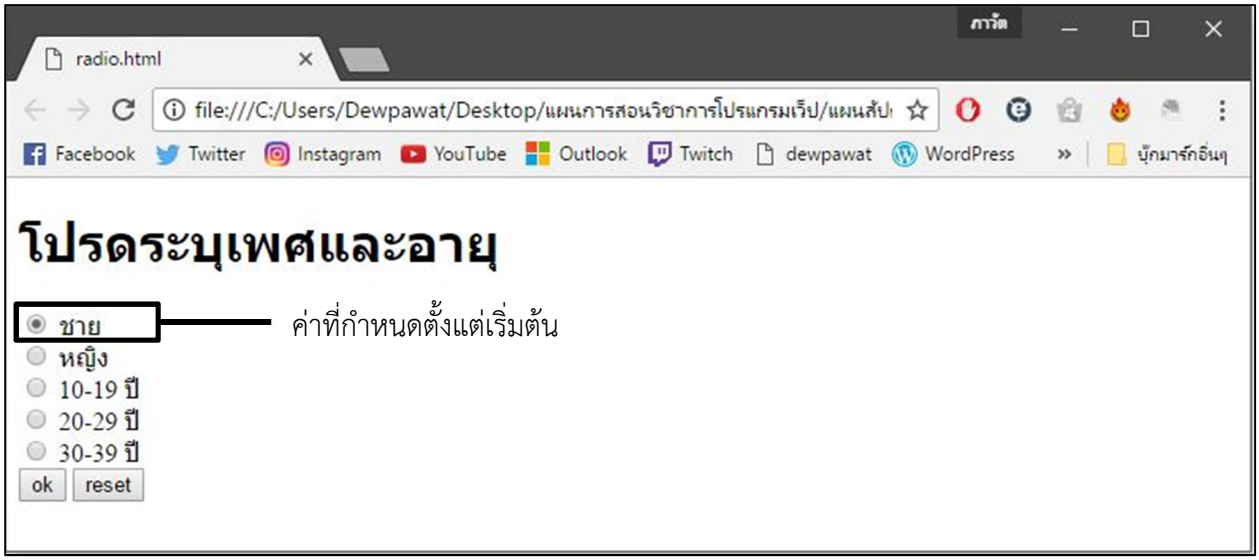
ตัวเลือกแบบให้เลือกข้อเดียว (Radio button)

จะตรงข้ามกับเช็คบ๊อกซ์ เพราะเรดิโอบัตทอน (radio button) จะให้ผู้ใช้เลือกเพียงข้อเดียว จากตัวเลือกทั้งหมด เช่น ถ้าเราเลือกข้อ ก. แล้วเปลี่ยนไปเลือกข้อ ข. ข้อ ก. ที่เคยเลือกไว้ก็จะเปลี่ยนสถานะไปเป็นข้อ ข. เป็นต้น การทำงานคล้ายกับสวิทช์ปุ่มพัดลม ตัวเลือกประเภทนี้เรามักจะเห็นในแบบทดสอบ แบบสอบถาม หรือการประเมินสิ่งใดสิ่งหนึ่ง

ตัวอย่างโค้ด HTML ของ radio button

```
<body>
        <form method='post' action='#'>
                <h1>โปรดระบุเพศและอายุ</h1>
                <input type='radio' value='male' name='gender' checked> ชาย <br>
                <input type='radio' value='female' name='gender'> หญิง <br>
                <input type='radio' value='19' name='age'> 10-19 ปี <br>
                <input type='radio' value='29' name='age'> 20-29 ปี <br>
                <input type='radio' value='39' name='age'> 30-39 ปี <br>
                <input type='submit' value='ok'>
                <input type='reset' value='reset'>
        </form>
</body>
```

ผลลัพธ์ที่ได้



จากตัวอย่างจะแบ่งการทำงานของเรดิโอบัตทอนเป็น 2 ชุดคือชุดแรกใช้เลือกเพศ และใช้เลือกอายุ ทั้ง 2 ชุดจะแยกกันทำงานออกจากกันโดนคุณสมบัติ name

เรดิโอบัตทอนที่มีค่าคุณสมบัติ name ชื่อเดียวกันจะถือเป็นกลุ่มเดียวกัน และจะเลือกได้เพียง 1 ตัวเลือกเสมอจากตัวเลือกทั้งหมดในกลุ่มเรดิโอนั้น ดังนั้นตัวอย่างชุดแรกคือเพศมี 2 ตัวเลือกเราสามารถเลือกได้แค่เพศเดียว และชุดที่อายุมี 3 ตัวเลือกเราก็สามารถเลือกได้เพียง 1 ตัวเลือกเท่านั้น

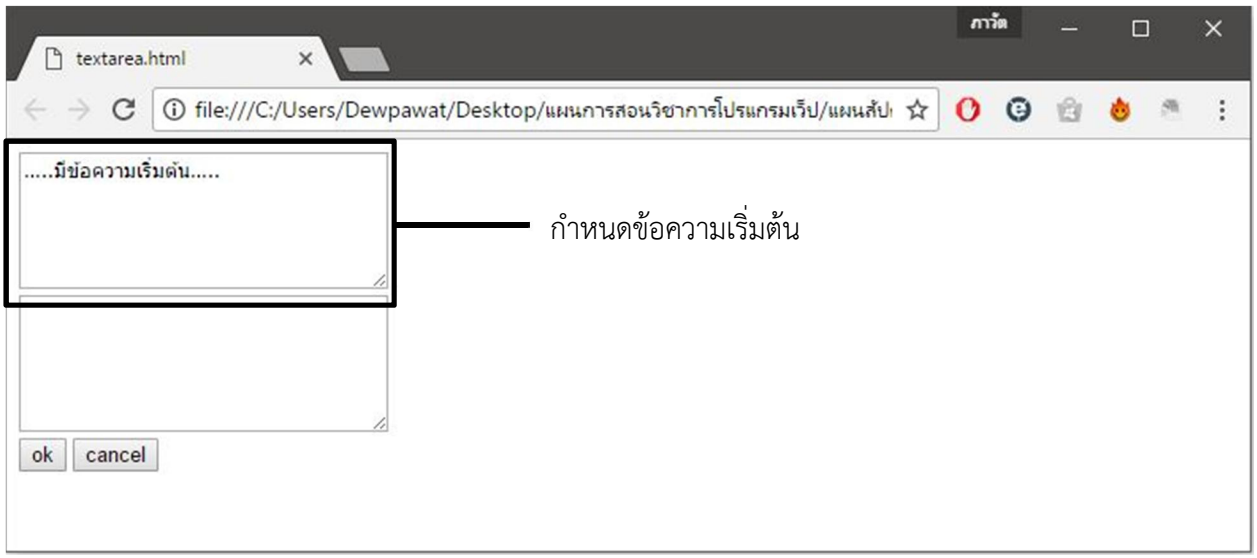
เช่นเดียวกันกับ เช็คบ๊อกซ์ เราสามารถใช้ checked เพื่อกำหนดให้ตัวเลือกเรดิโอตัวนั้นๆ ถูกเลือกตั้งแต่ต้นได้

ช่องรับข้อความแบบหลายบรรทัด <textarea>

ในกรณีที่มีข้อมูลที่ต้องป้อนในฟอร์มมากกว่า 1 บรรทัด เราสามารถใช้ Text Area ได้โดยมีรูปแบบโค้ด HTML ดังนี้

```
<body>
    <form method='post' action='#'>
        <textarea cols='30' rows='5' name='area1'>
        .....มีข้อความเริ่มต้น.....
        </textarea><br>
        <textarea cols='30' rows='5' name='area1'>
        </textarea><br>
        <input type='submit' value='ok'>
        <input type='reset' value='cancel'>
    </form>
</body>
```

ผลลัพธ์ที่ได้



คุณสมบัติของ <textarea>

name : เป็นชื่อของ แท็ก <textarea> เพื่อใช้อ้างอิงตอนรับข้อมูล

rows : จำนวนบรรทัดที่ต้องการ

cols : จำนวนความกว้างของกรอบรับข้อความ

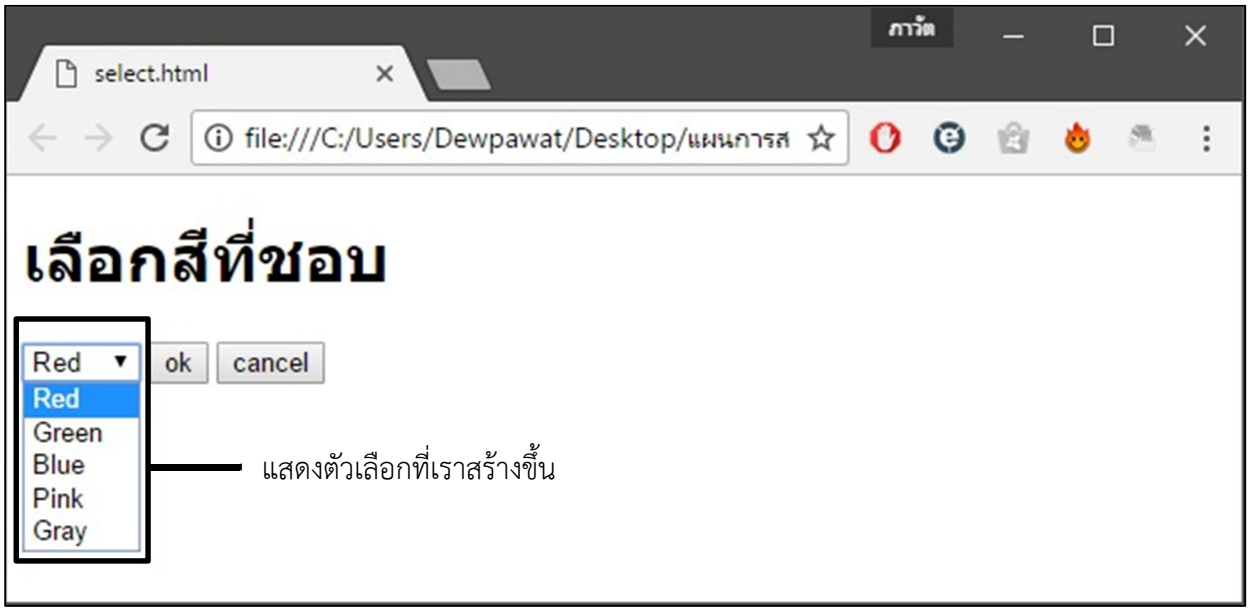
ตัวเลือกรายการ (Drop Down Menu)

ตัวเลือกรายการ (Drop Down Menu) หรือการใช้ Selection เป็นอุปกรณ์ที่ใช้แสดงข้อมูลให้ผู้เลือกเพียง 1 ข้อมูลจากข้อมูลทั้งหมด คล้ายเรดิโอบัตทอนแต่จะประหยัดพื้นที่การแสดงผลมากกว่า โดยมีรูปแบบโค้ด HTML ดังนี้

```
<body>
    <form method='post' action='#'>
        <h1>เลือกสีที่ชอบ</h1>
        <select name='color'>
            <option value='#ff0000'>Red</option>
            <option value='#00ff00'>Green</option>
            <option value='#0000ff'>Blue</option>
            <option value='#dd2222>Pink</option>
            <option value='#555555'>Gray</option>
        </select>
        <input type='submit' value='ok'>
        <input type='reset' value='cancel'>
    </form>
</body>
```

ตัวอย่างการใช้ select

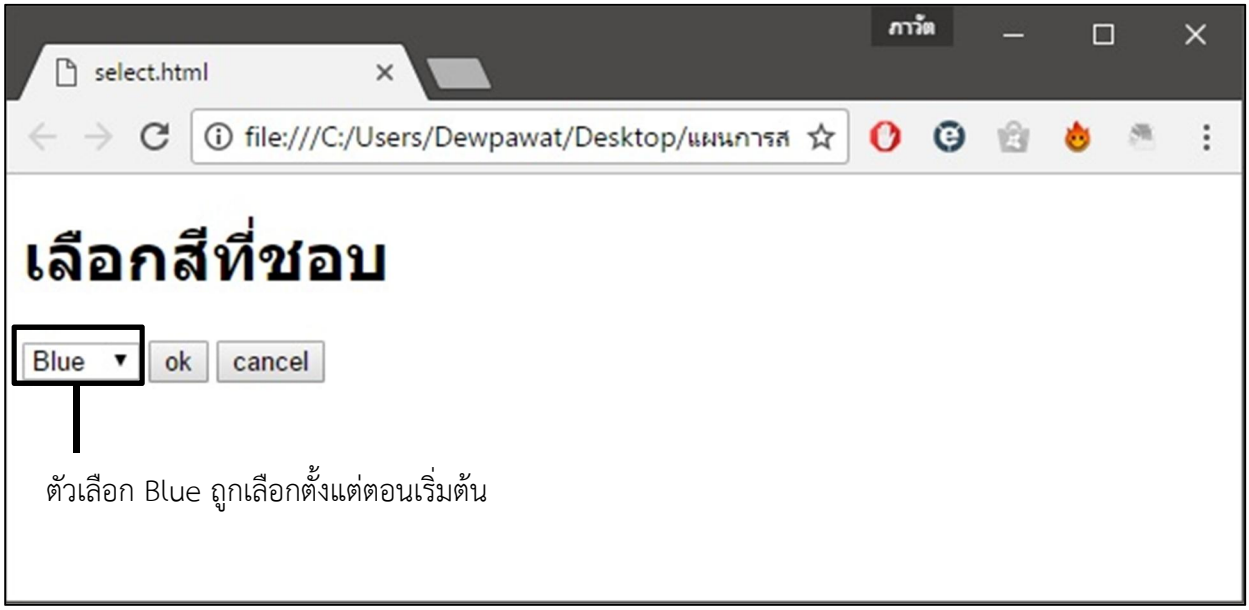
ผลลัพธ์ที่ได้



การกำหนดตัวเลือกเริ่มต้น เราสามารถกำหนดตัวเลือกที่ถูกเลือกเอาไว้ล่วงหน้าได้ โดยกำหนดแอททริบิวท์ selected

```
<body>
        <form method='post' action='#'>
                <h1>เลือกสีที่ชอบ</h1>
                <select name='color'>
                        <option value='#ff0000'>Red</option>
                        <option value='#00ff00'>Green</option>
                        <option value='#0000ff' selected>Blue</option>
                        <option value='#dd2222>Pink</option>
                        <option value='#555555'>Gray</option>
                </select>
                <input type='submit' value='ok'>
                <input type='reset' value='cancel'>
        </form>
</body>
```

ผลลัพธ์ที่ได้



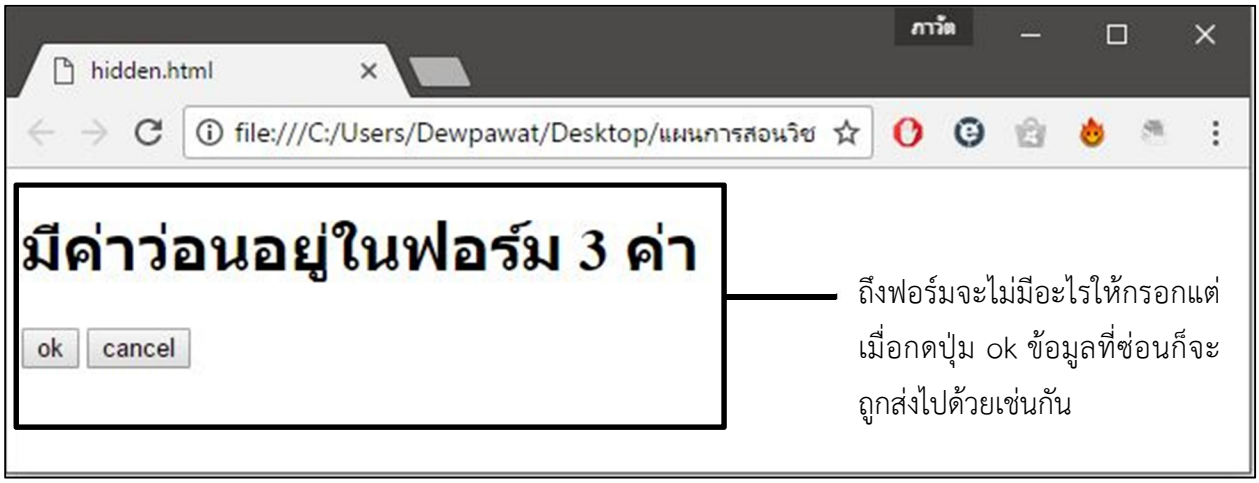
ช่องเก็บข้อมูลแบบซ่อน (hidden)

เป็นช่องเก็บข้อมูลที่เราใช้เก็บค่าใดๆ ที่เราต้องการส่งไปพร้อมกับข้อมูลที่กรอกลงในฟอร์มเมื่อเรากดปุ่ม submit เช่น ข้อมูลระบบคอมพิวเตอร์ วันเวลาที่กรอกข้อมูล เลขทะเบียนแบบฟอร์ม เป็นต้น

ข้อมูลเหล่านี้จะมีอยู่ในระบบปฏิบัติการอยู่แล้ว เราสามารถเรียกใช้ผ่านภาษาสคริปต์ฝั่งไคลเอ็นต์ (Client-Side Script) เช่น JavaScript หรือ VBScript ได้ และเราจะนำข้อมูลเหล่านี้มาเก็บในแท็ก hidden ซึ่งล้วนเป็นข้อมูลที่ไม่ได้รับจากผู้ใช้ แต่จะใช้ส่งต่อไปประมวลผล เมื่อกดปุ่ม submit ค่าในอุปกรณ์ hidden จะถูกส่งไปเซิร์ฟเวอร์พร้อมกับค่าในฟอร์มตัวอื่นๆ ตัวอย่างโค้ด HTML ดังนี้

```
<body>
        <form method='post' action='#'>
                <h1>มีค่าว่อนอยู่ในฟอร์ม 3 ค่า</h1>
                <input type='hidden' name='formID' value='FORM001'>
                <input type='hidden' name='userID' value='dewpawat'>
                <input type='hidden' name='keyID' value='X34B452CG45'>
                <input type='submit' value='ok'>
                <input type='reset' value='cancel'>
        </form>
</body>
```

ผลลัพธ์ที่ได้คือ

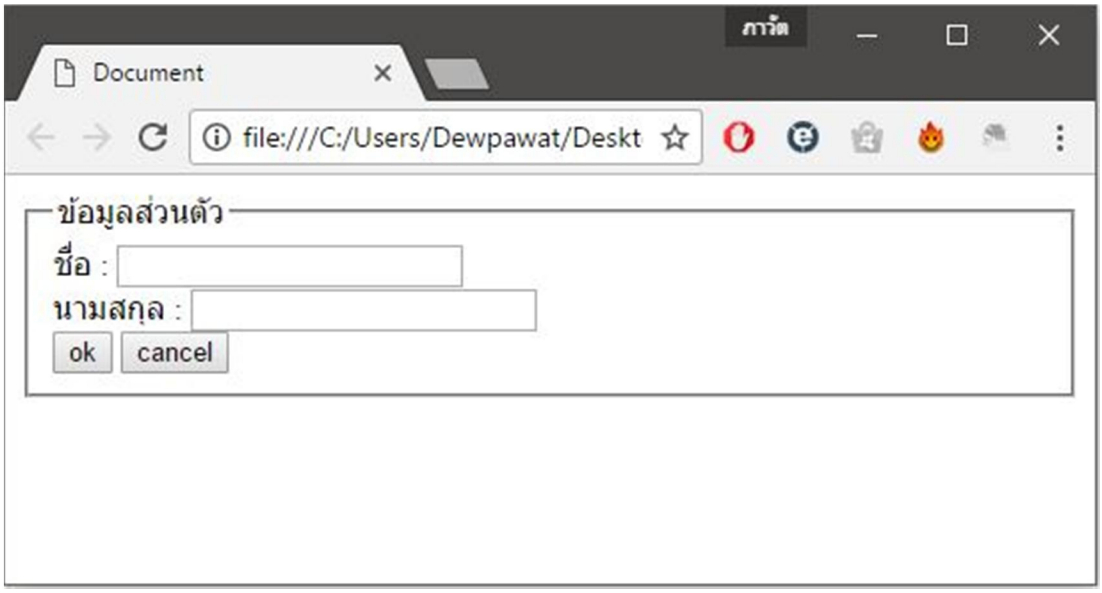


การจัดแบ่งแบบฟอร์ม <fieldset> และ <legend>

ถ้าหากเว็บของเรามีช่องทางรับข้อมูลหลายๆตัว การจัดกลุ่มช่องทางรับข้อมูลจะทำให้ดูดีและมีขอบเขตการกรอกข้อมูลมากยิ่งขึ้น การจัดแบ่งเราจะใช้แท็ก <fieldset> เป็นเส้นกำหนดขอบเขต และ <legend> ใช้ในการตั้งชื่อฟอร์มนั้น ๆ ตัวอย่างโค้ด HTML ดังนี้

```
<body>
        <form method='post' action='#'>
                <fieldset>
                        <legend>ข้อมูลส่วนตัว</legend>
                        ชื่อ : <input type='text' name='name'><br>
                        นามสกุล : <input type='text' name='surname'><br>
                        <input type='submit' value='ok'>
                        <input type='reset' value='cancel'>
                </fieldset>
        </form>
</body>
```

ผลลัพธ์ที่ได้คือ



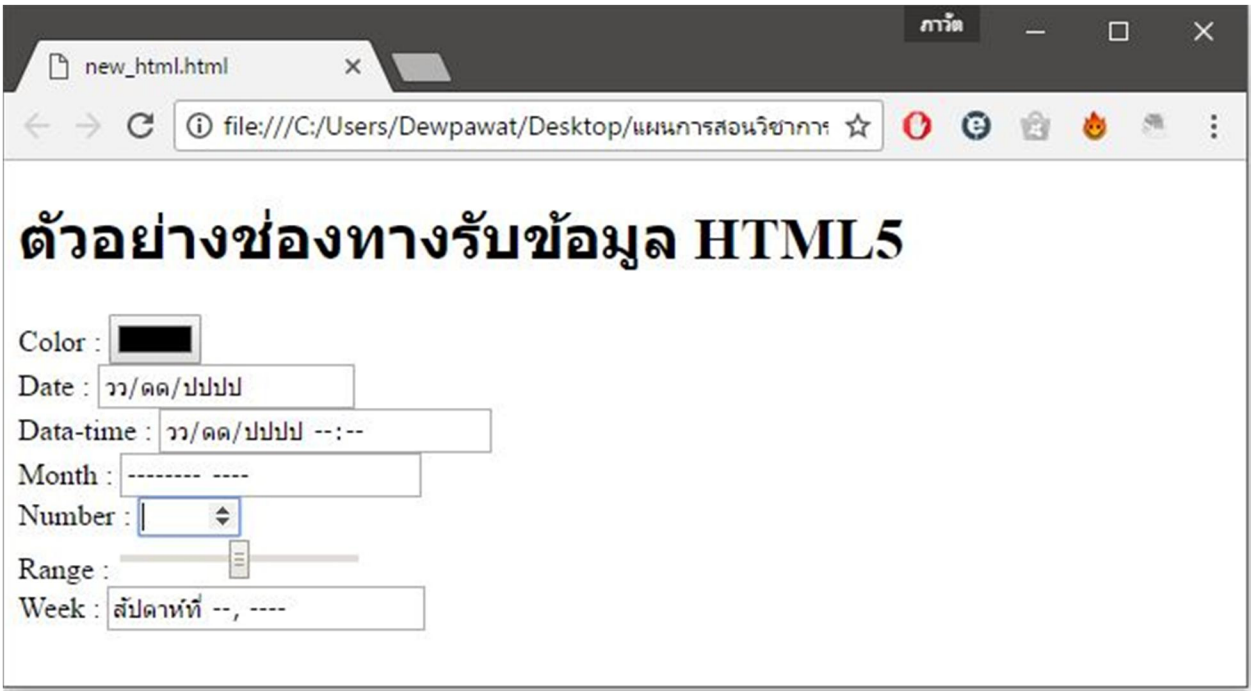
ช่องทางรับข้อมูลเพิ่มเติมของฟอร์มใน HTML5

type	ความหมาย
color	ช่องรับสีที่ต้องการ
date	ช่องรับวันที่จากปฏิทิน
datetime-local	ช่องรับวันที่และเวลา โดยไม่มีค่า Time zone
email	ช่องรับอีเมล์ โดยจะมีการตรวจสอบว่าข้อมูลที่ใส่ลงมาเป็น e-mail เท่านั้น
month	ช่องรับเดือนและปีจากปฏิทิน
number	ช่องรับเฉพาะตัวเลข
range	ช่องรับข้อมูลเป็นแถบใช้เมาส์ลากเอาข้อมูลที่ต้องการ
search	ช่องสำหรับข้อมูลค้นหา
tel	ช่องรับเบอร์โทรศัพท์
time	ช่องรับเวลา โดยมีปุ่มเพิ่มลดเวลา
url	ช่องรับ URL หรือ โดเมนเนม ซึ่งเก็บที่อยู่เว็บไซต์
week	ช่องรับสัปดาห์ เดือน และปีจากปฏิทิน

ตัวอย่างลักษณะช่องทางรับข้อมูลของ HTML

```
<!DOCTYPE html>
<html>
<body>
        <h1> ตัวอย่างช่องทางรับข้อมูล HTML5 </h1>
        <form method='post' action='#'>
                Color : <input type="color" name="favcolor"><br>
                Date : <input type="date" name="bday"><br>
                Data-time : <input type="datetime-local" name="bdaytime"><br>
                Month : <input type="month" name="bdaymonth"><br>
                Number : <input type="number" name="quantity" min="1" max="5"><br>
                Range : <input type="range" name="points" min="0" max="10"><br>
                Week : <input type="week" name="week_year">
        </form>
</body>
</html>
```

ผลลัพธ์ที่ได้คือ



จากตัวอย่างก็จะเป็นช่องทางรับข้อมูลที่เป็นฟีเจอร์ใหม่ของ HTML5 ซึ่งจากตัวอย่างก่อนที่จะใช้งาน HTML ต้องประกาศแท็ก <!DOCTYPE html> ด้านบนแท็ก <html> จากตัวอย่าง โดยนอกจะแท็กนำเข้าข้อมูลแล้ว HTML ก็มีคุณสมบัติใหม่ๆ ที่มาช่วยทำให้การนำเข้าข้อมูลมีความหลากหลายมากขึ้น

คุณสมบัติเพิ่มเติม HTML5

Attribute	ความหมาย
disabled	ปิดการรับข้อมูลของช่องรับข้อความ โดยช่องรับจะเปลี่ยนเป็นสีเทส
max (HTML5)	กำหนดค่าสูงสุดของช่องรับข้อมูล
maxlength	กำหนดจำนวนตัวอักษร
min (HTML5)	กำหนดค่าต่ำสุดของช่องรับข้อมูล
pattern (HTML5)	กำหนดชนิดข้อมูลให้เป็นไปตามเงื่อนไขที่เรากำหนด
readonly	ทำให้ช่องรับข้อมูลอ่านได้อย่างเดียว ไม่สามารถแก้ไขได้
required (HTML5)	ช่องสำหรับตรวจสอบไม่ให้เป็นค่าว่าง
size	กำหนดความยาวของช่องรับข้อมูล
step (HTML5)	ใช้ระบุช่วงข้อมูลของช่องรับตัวเลข เช่นรับทีละ 5,10,15 เป็นต้น
value	กำหนดค่าของช่องรับข้อมูลนั้น ๆ

หมายเหตุ (HTML5) หมายถึง คุณสมบัติที่ของ HTML5 ต้องประกาศ <!DOCTYPE html> ก่อนเรียกใช้งาน