

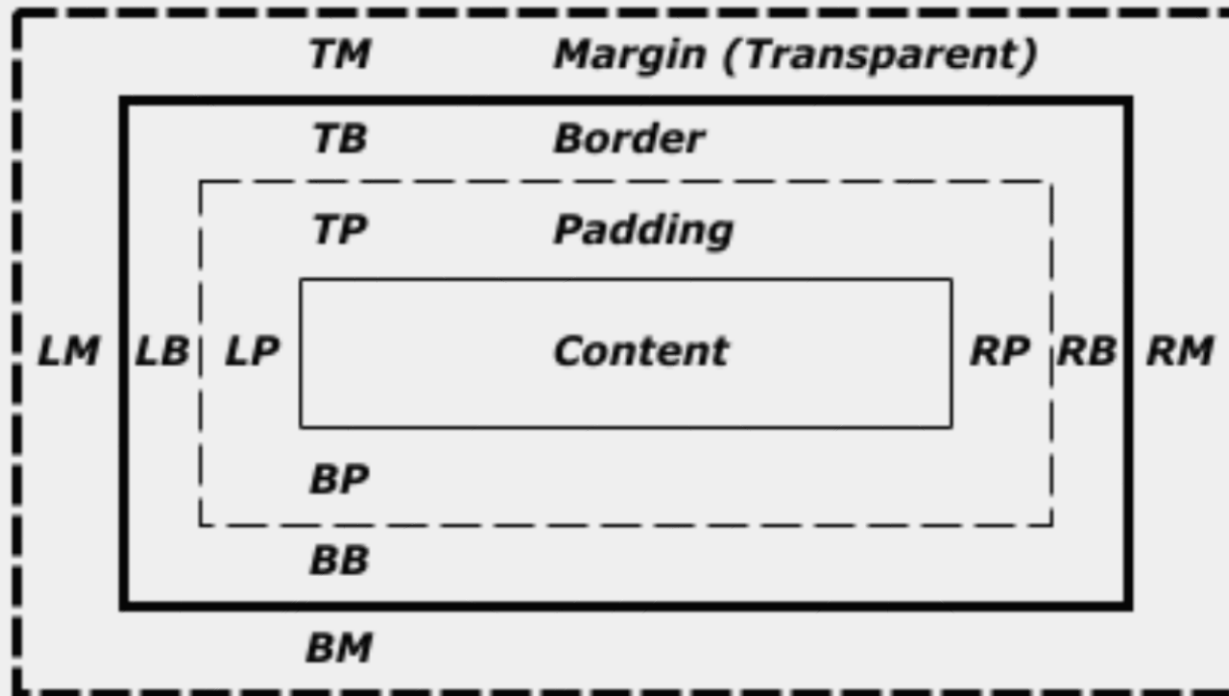
Written by Thapanapong Rukkanchanunt

CSS Element Positioning

สิ่งที่สำคัญที่สุดใน CSS

- สิ่งที่เราได้เรียนไปในคาบก่อนเป็นการแนะนำการใช้ CSS เบื้องต้น ซึ่งเป็นเพียงแค่การแบ่งการจัดรูปแบบออกจากโครงสร้าง
- สิ่งที่ทำให้ CSS ทรงพลังคือความสามารถในการจัดตำแหน่งของ HTML ภายในหน้า
- แต่ละแท็กในภาษา HTML เปรียบเสมือนกล่องเล็กๆ ที่เก็บข้อมูลข้อความหรือรูปภาพ ซึ่งกล่องดังกล่าวจะมีการจัดรูปภายในอีกมากมาย

กล่อง CSS



- Margin edge
- Border edge
- - - Padding edge
- Content edge

คุณสมบัติ Display

- คุณสมบัติ Display ใช้กำหนดลักษณะของกล่องที่ล้อมรอบแท็ก HTML
- display: block ทำให้แท็กเป็นกล่องที่กินพื้นที่ตามแนวกว้างทั้งหมด
- display: inline-block มีลักษณะเหมือน block ปกติแต่อนุญาตให้แท็กอื่นมาอยู่ข้างได้
- display: inline อนุญาตให้แท็กอื่นมาอยู่ข้างได้ และกินพื้นที่เท่าที่จำเป็น
- display: none ทำให้แท็กหายไปจากหน้าจอ

ทบทวน <div>

ในไฟล์ HTML

```
<div></div>
```

```
<div></div>
```

```
<div></div>
```

```
<div></div>
```

ในไฟล์ CSS

```
div {
```

```
    height: 50px;
```

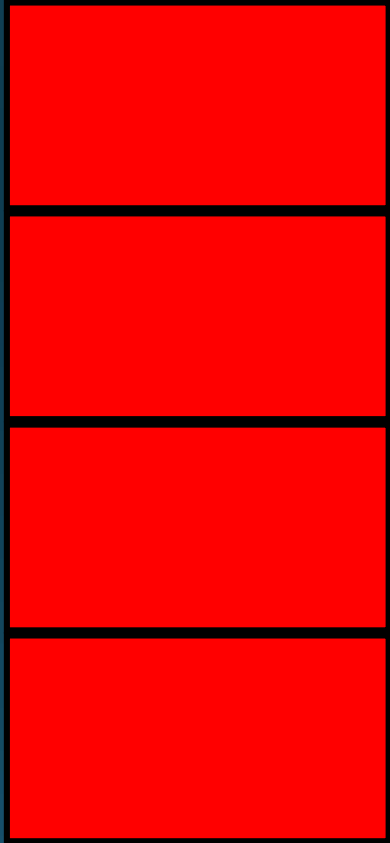
```
    width: 100px;
```

```
    border: 2px solid black;
```

```
    background-color: red;
```

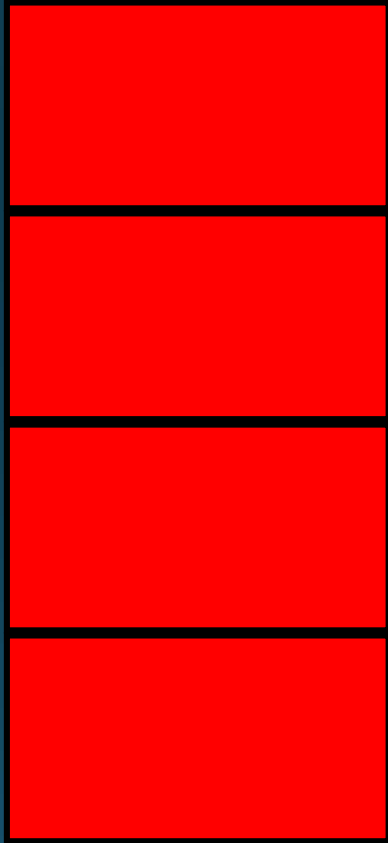
```
}
```

ผลลัพธ์ที่ได้



```
div {  
    height: 50px;  
    width: 100px;  
    border: 2px solid black;  
    background-color: red;  
}
```

Display: block



```
div {  
    height: 50px;  
    width: 100px;  
    border: 2px solid black;  
    background-color: red;  
    display: block;  
}
```

Display: inline-block



ถ้าสิ้นหน้าจอก็จะขึ้นบรรทัดใหม่

```
div {  
    height: 50px;  
    width: 100px;  
    border: 2px solid black;  
    background-color: red;  
    display: inline-block;  
}
```


Display: inline



```
div {  
    height: 50px;  
    width: 100px;  
    border: 2px solid black;  
    background-color: red;  
    display: inline;  
}
```

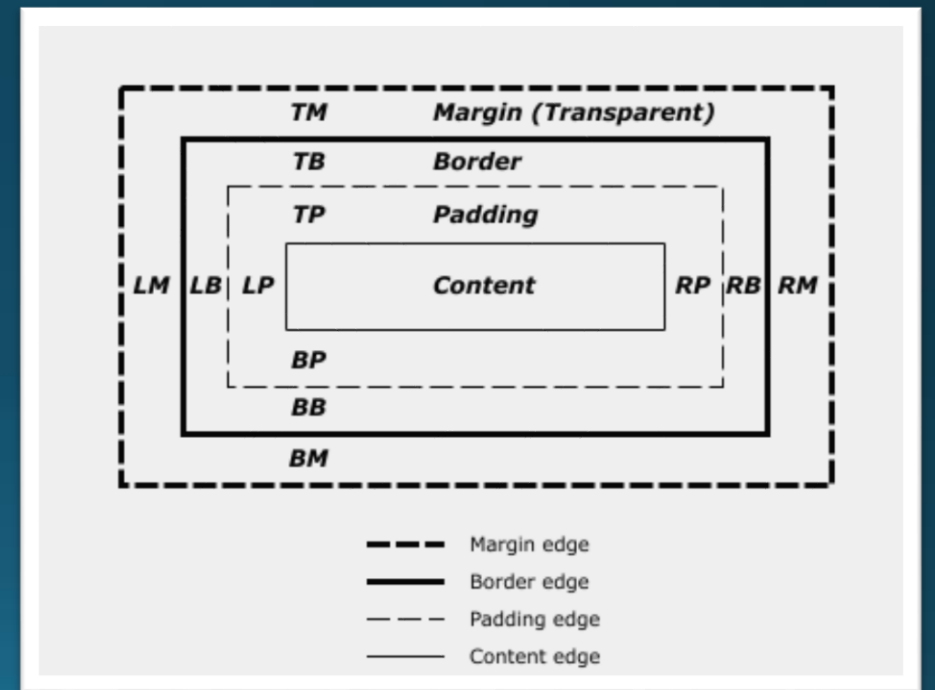
Display: none

```
div {  
    height: 50px;  
    width: 100px;  
    border: 2px solid black;  
    background-color: red;  
    display: none;  
}
```

การกำหนดระยะขอบ

ในกล่อง CSS เราจะมีระยะขอบระหว่างชั้นย่อย ๆ ซึ่งได้แก่

1. Margin คือช่องว่างระหว่างกล่อง
2. Border คือเส้นขอบของกล่อง
3. Padding คือช่องว่างระหว่างเนื้อหาและเส้นขอบ
4. Content คือเนื้อหาของกล่อง



Margin

- การกำหนดคุณลักษณะ Margin ของแท็กเป็นการขยับแท็กเมื่อเทียบกับแท็กอื่น
- ถ้าหากเรากำหนดให้แท็กจัดกึ่งกลางแบบอัตโนมัติ เราจะกำหนด Margin เป็น

```
margin: auto;
```

- ถ้าต้องกำหนดระยะของแต่ละด้าน เราจะกำหนดคุณลักษณะดังนี้

```
margin-top: 100px;          /* Margin ด้านบน */
```

```
margin-right: 100px;       /* Margin ด้านขวา */
```

```
margin-bottom: 100px;     /* Margin ด้านล่าง */
```

```
margin-left: 100px;       /* Margin ด้านซ้าย */
```

All Margin At Once

- ถ้าหากต้องการกำหนดทุกด้านในบรรทัดเดียว สามารถทำได้โดยการกำหนดเรียงลำดับ

```
margin: 10px 20px 30px 40px;
```

```
/* เรียงจาก บน ขวา ล่าง ซ้าย */
```

Padding

- Padding คือช่องว่างระหว่างเส้นขอบและเนื้อหา สามารถกำหนดคุณลักษณะนี้ได้เช่นเดียวกับ Margin นั่นคือ

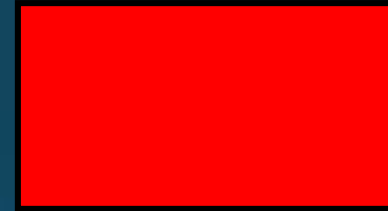
padding-top, padding-right, padding-bottom, padding-left

หรือกำหนดทุกด้านพร้อมกันแบบ

padding: บน ขวา ล่าง ซ้าย;

ตัวอย่าง Padding (1)

```
div {  
    height: 50px;  
    width: 100px;  
    border: 2px solid black;  
    background-color: red;  
}
```



ตัวอย่าง Padding (2)

```
div {  
    height: 50px;  
    width: 100px;  
    border: 2px solid black;  
    background-color: red;  
    padding-left: 40px;  
}
```



ค่าติดลบ

- เราสามารถกำหนดระยะ Margin เป็นเลขจำนวนเต็มลบได้ โดยจะถือว่าเป็นการขยับแท็กไปในทิศทางตรงข้าม



Float

- เราทราบการกำหนดระยะของแต่ละแท็กแล้ว ลำดับถัดไปคือการบอกแต่ละแท็กว่าควรจะอยู่ที่ไหนในหน้าเว็บของเรา
- Float เป็นคุณลักษณะที่ใช้บอกแท็ก HTML ว่าจะต้องไปอยู่ตำแหน่งใดของหน้าเว็บ แต่จะต้องสอดคล้องกับแท็กที่อยู่ในตำแหน่งดังกล่าวอยู่แล้ว (นั่นคือแท็กจะไม่ซ้อนทับกัน)
- float: left; จะวางแท็กไว้ทางด้านซ้าย ในขณะที่ float: right; วางไว้ด้านขวา

ตัวอย่างการใช้ Float (1)

```
.left {
```

```
    float: left;
```

```
    height: 300px;
```

```
    width: 150px;
```

```
}
```

```
.normal {
```

```
    height: 50px;
```

```
}
```

```
.right {
```

```
    float: right;
```

```
    height: 300px;
```

```
    width: 150px;
```

```
}
```

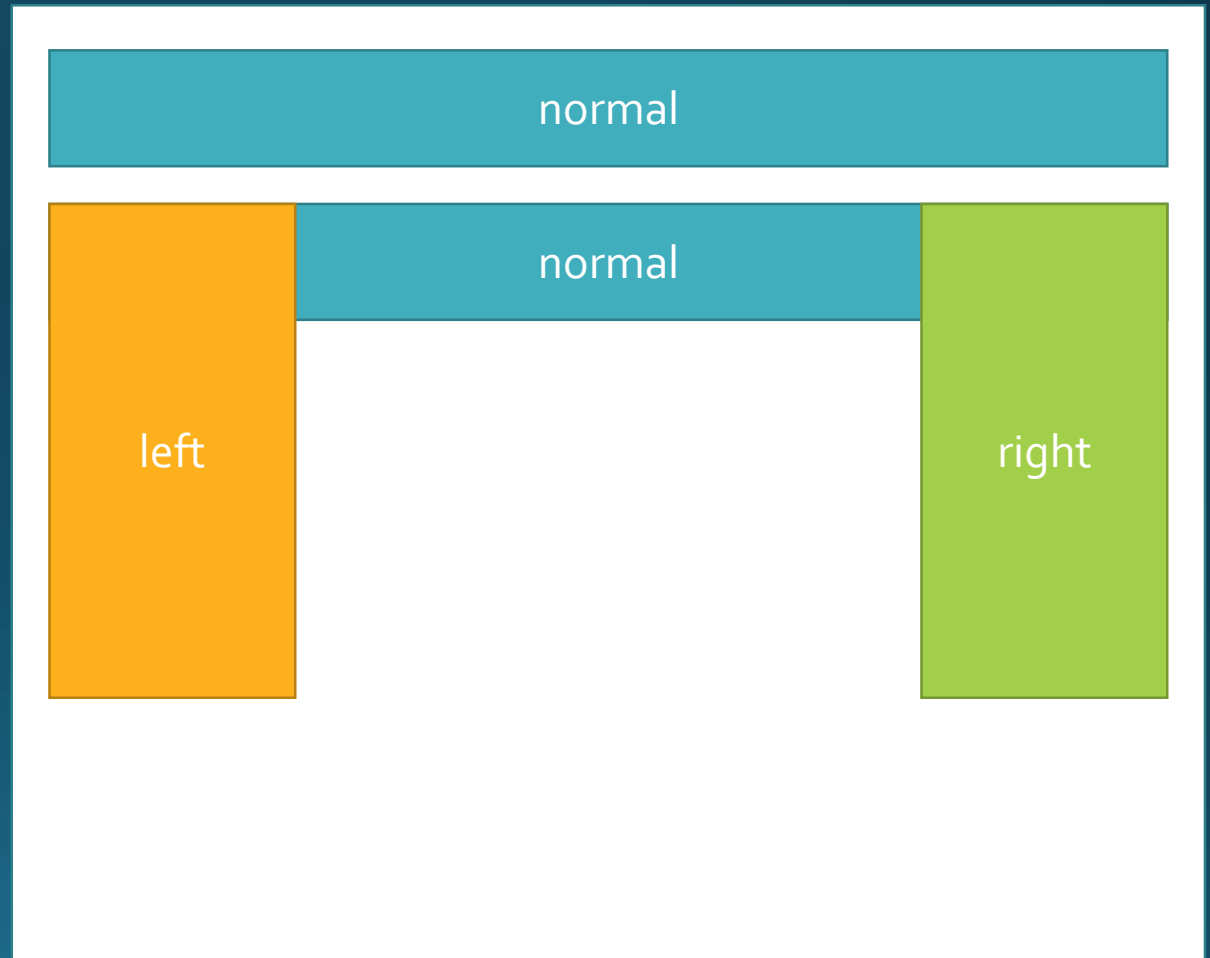
ตัวอย่างการใช้ Float (2)

```
<div class="normal"></div>
```

```
<div class="left"></div>
```

```
<div class="right"></div>
```

```
<div class="normal"></div>
```



การใช้ Clear

- จากตัวอย่างที่ผ่านมา เราจะพบว่าถ้าเราใช้แท็กที่มีการกำหนดคุณลักษณะ Float ร่วมกับแท็กที่ไม่มีการกำหนดคุณลักษณะ Float เราจะได้แท็กที่ซ้อนทับกัน
- ในการแก้ปัญหาดังกล่าว เราจำเป็นต้องกำหนดคุณลักษณะ Clear ให้กับแท็กที่ไม่ได้ใช้ Float โดยมีรายละเอียดดังนี้

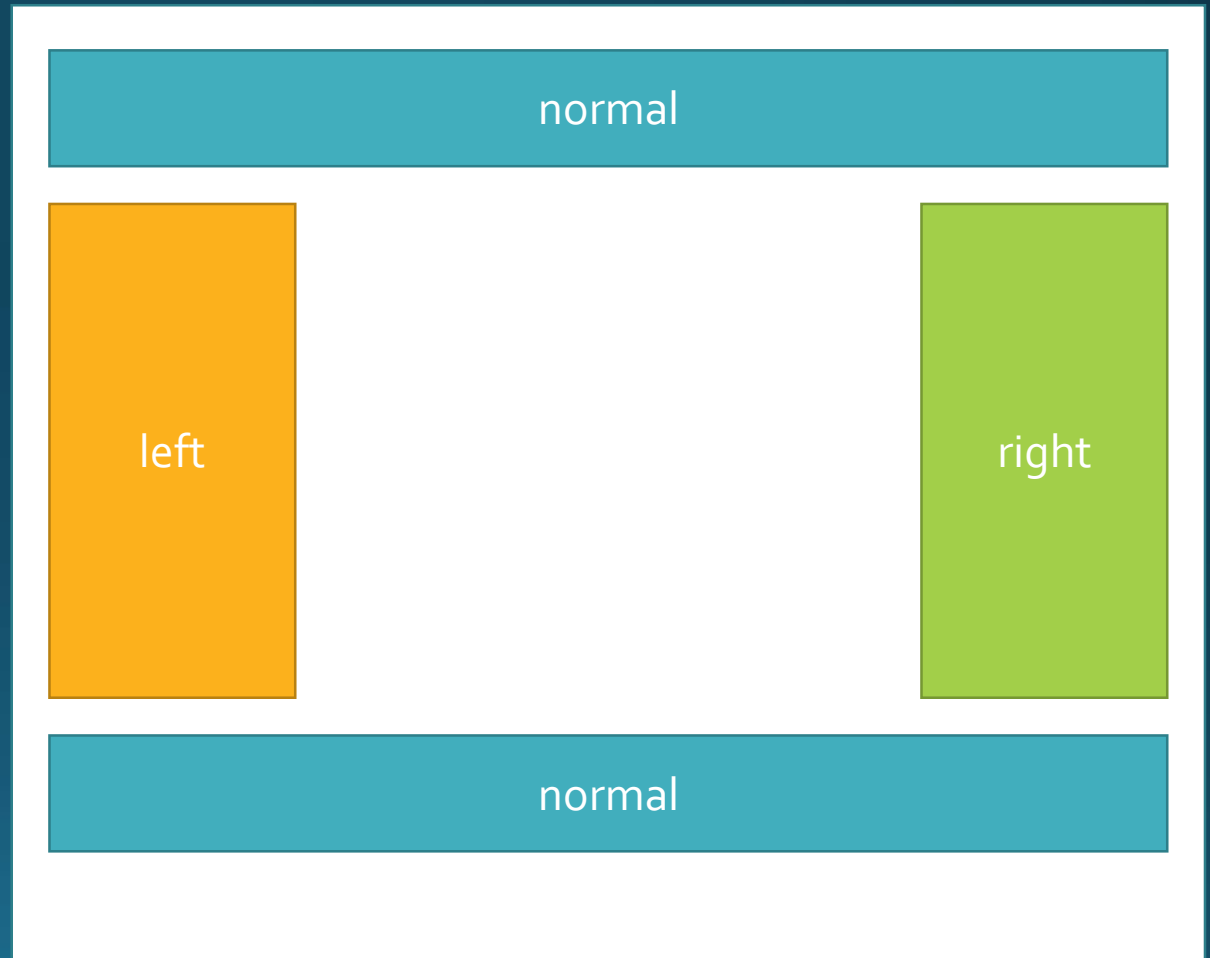
clear: left; /* ถ้าทับกับแท็กที่มี float: left; จะขยับแท็กนี้ไว้ด้านล่าง */

clear: right; /* ถ้าทับกับแท็กที่มี float: right; จะขยับแท็กนี้ไว้ด้านล่าง */

clear: both; /* เป็นทั้งสองแบบรวมกัน */

ตัวอย่างการใช้ Clear

```
.normal {  
    height: 50px;  
    clear: both;  
}
```



ประเภทของตำแหน่ง

- เราสามารถทำให้แท็กอยู่กับที่แม้เราจะเลื่อนหน้าจอไปมาโดยการกำหนด

position: fixed;

- นอกจากนี้แล้ว position ยังมีค่าที่เป็นไปได้อีกสามค่าคือ static, absolute และ relative
- Static เป็นค่าปกติ ให้ผลลัพธ์เช่นเดียวกับไม่กำหนดค่า position
- Absolute เป็นการกำหนดตำแหน่งเมื่อเทียบกับแท็กที่ครอบมันอยู่และต้องไม่มีค่า position เป็น static
- Relative มีลักษณะคล้ายกับ Static เพียงแต่อนุญาตให้กำหนดคุณลักษณะอื่นเพิ่มเติม เช่น top, left, right, และ bottom

ตัวอย่างการใช้งาน position (1)

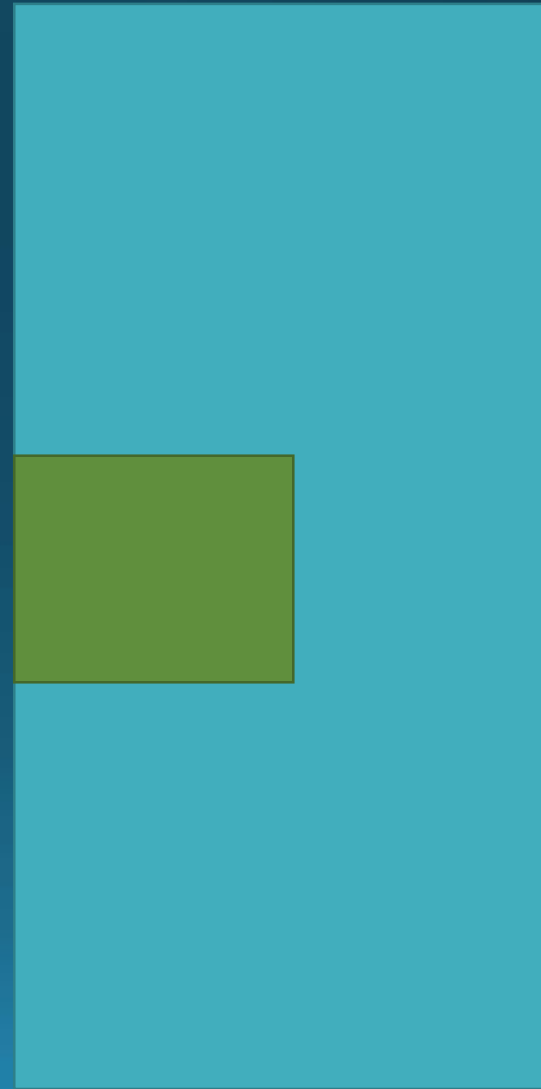
```
.outer {  
    height: 1500px;  
    width: 150px;  
    background-color: blue;  
}
```

```
.inner {  
    height: 75px;  
    width: 75px;  
    background-color: green;  
    position: fixed;  
    margin-top: 200px;  
}
```


ตัวอย่างการใช้งาน position (2)

```
<div class="outer">  
  <div class="inner"></div>  
</div>
```

ลองเลื่อนหน้าจอขึ้นลง



ทบทวนเนื้อหาวันนี้

- รู้จักกับกล่อง CSS
- กำหนดการจัดวางระหว่างแท็กด้วยคุณลักษณะ Display
- กำหนดระยะขอบต่างๆด้วยคุณลักษณะ Margin และ Padding
- กำหนดตำแหน่งแท็กด้วยคุณลักษณะ Float
- กำหนดประเภทของตำแหน่งด้วยคุณลักษณะ Position