

CSS Training

By : Kosit Aromsava

Contact : nn.it@hotmail.com Tel : 080-992-6565

CSS ?

- ก่อนเริ่มศึกษา CSS ควรมีความรู้ความเข้าใจ HTML / XHTML
- CSS ย่อมาจาก Cascading Style Sheets
- ใช้สำหรับกำหนดรูปแบบการแสดงผลของหน้าเว็บ
- ใช้งานร่วมกับ HTML เพื่อเพิ่มประสิทธิภาพให้กับเว็บเพจมากกว่าเดิม เช่น การนำ CSS จัดรูปแบบ Layout และการใช้เทเบิล

ข้อดีของ CSS

1. สามารถใช้ควบคุมการแสดงผล ได้ที่เดียวหลาย ๆ หน้าเว็บเพจ
2. แก้ไขเพียงจุดเดียว ก็มีผลกับเอกสารทั้งหมดได้
3. สามารถจัดการเรียงลำดับได้อย่างแม่นยำและอิสระ
4. มีความยืดหยุ่นสูง ในการปรับแต่งแก้ไขในอนาคต
5. ง่ายในการเรียนรู้โค้ด
6. ภาษาที่ใช้เขียนง่าย ๆ ค่อนข้างตรงตัว ไม่ต้องแปลความให้ยุ่งยาก
7. ทำให้ขนาดไฟล์แต่ละหน้าเล็กลงกว่าเดิม เพราะใช้โค้ดน้อยกว่า
8. สามารถแยกไว้ต่างหากจากไฟล์ html

การเขียน CSS เพื่อการใช้งานร่วมกับ XHTML

- Inline Style Sheets
- Embed Style Sheets
- External Style Sheet

Inline Style Sheet

- เขียนไปที่บรรทัดคำสั่งของ TAG โดยตรง โดยผ่าน Attribute Style

ตัวอย่าง

```
<div style="font-size:24px;">Test Inline1</div>
<strong style="font-color:#ff00ff">Test
Inline2</strong>
```

Embed Style Sheet

- เป็นการเขียน CSS 放进 XHTML โดยเขียนไว้ภายใต้ Tag `<head>...</head>`

ตัวอย่าง

```
<html>
<head>
<title>Embed Style Sheet</title>
<style type="text/css">
    h1{
        font-size:24px;
        font-color:red;
    }
</style>
</head>
<body>
<h1>Test Embed Style Sheet</h1>
</body>
</html>
```

External Style Sheet

- เป็นการสร้างไฟล์ CSS แยกไว้ภายนอกกับ XHTML โดยมีนามสกุลเป็น .CSS เรียกใช้งานได้ 2 วิธีคือ

Linking Style Sheet

```
<link rel="stylesheet" href="file.css">
```

ลำดับความสำคัญ

1. Inline
2. Embed
3. External
4. Default Browser

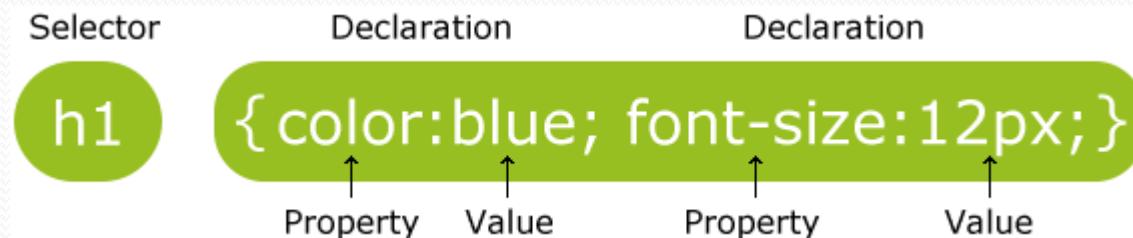
การเขียน Comment css

การ Comment เป็นการอธิบายโปรแกรมที่ได้พัฒนาไป comment ของ CSS จัดการโดยใช้เครื่องหมาย /*.....*/

ตัวอย่าง

```
.border{border:1px solid;}  
/*Class border จัดการขอบโดยมีความหนา 1 px และ solid*/  
#bg{background:#cccccc;}  
/*Id bg มีพื้นหลังเป็นสีเทา*/
```

ໄວຍາກຮົ່ນ CSS



ມີ 2 ส່ວນໃນການກຳຫນັດ CSS ຄື່ອ selector ກັບ Declaration

Selector ມີ 4 ປະເທດຄື່ອ Tag, Id, Class, Attribute ເປັນຕົວກຳຫນັດວ່າຈະ
ທຳການຈັດກຳຫນັດ element ຕົວໄໜ້

Declaration ປະກອບດ້ວຍ Property ກັບ value

ໄວຍາກຮົ່ນຂອງ CSS(ຕ່ອ)

Selector{property:value;}

ตัวอย่าง

.box500{width:500px;}

.box500{width:500px; height:500px;}

Selector ຄືອ ຂຶ້ອທີ່ໃຊ້ວ່າງອີງ

Property ຄືອ ຂຶ້ອຄຸນສມບັດ

Value ຄືອ ດຳເນັດໃຫ້ **property**

Selector มี 4 ประเภท

- Class Selector
- ID Selector
- Tag Selector
- Attribute Selector

Class selector

การกำหนด class css

.ClassName{properties:value;}

ตัวอย่าง

.center {text-align:center;}

กฎการตั้งชื่อ class

ชื่อ class นำหน้าด้วยเครื่องหมาย Dot(.)

ชื่อ class เป็นภาษา English

ชื่อ class ห้ามขึ้นต้นด้วยตัวเลข

ชื่อ class ห้ามเว้นวรรค

ชื่อ class เป็นแบบ Case sensitive คือ ตัวเล็กตัวใหญ่มีผลต่อการใช้งาน

การนำ class css ไปใช้งานในส่วนของ XHTML

<div class="center"></div>

ID Selector

กำหนด CSS ในรูปแบบ ID

```
#IdName{properties:value;}
```

ตัวอย่าง

```
#para1
{
    text-align:center;
    color:red;
}
```

การนำ id css ไปใช้งานกับ XHTML

```
<div id="para1"></div>
```

กฎการตั้งชื่อ ID

ชื่อ ID นำหน้าด้วยเครื่องหมาย #

ชื่อ ID เป็นภาษา English

ชื่อ ID ห้ามขึ้นต้นด้วยตัวเลข

ชื่อ ID ห้ามเว้นวรรค

ชื่อ ID เป็นแบบ Case sensitive คือ ตัวเล็กตัวใหญ่มีผลต่อการใช้งาน

Tag Selector

กำหนด css ในรูปแบบ Tag

TagName{properties:value;}

ตัวอย่าง

การนำ tag css ไปใช้งานในส่วนของ XHTML

P{font-size:24px;}

<p>Test Tag Selector</p>

เป็นการกำหนด Tag XHTML(ที่มีอยู่แล้วใน XHTML) เพื่อกำหนดเพิ่มเติม
หรือเป็นการกำหนดคุณสมบัติของ Tag XHTML นั้นใหม่ ตัวอย่าง เช่น

<p>,<input>,<select>,<table>,,<h1>

Attribute selector

ประกาศ CSS ในรูปแบบ Attribute

[name]{color:#cccccc}

ตัวอย่าง

การนำ Attribute css ไปใช้งานในส่วนของ XHTML

[name=inputname]{color:#cccccc}
<input type="text" name="inputname">

ประกาศ Attribute selector มีหลักการดังนี้

1. [attr] ประกาศ CSS ที่มีแอตทริบิวต์ชื่อ attr
2. [attr=value] ประกาศ CSS ที่มีแอตทริบิวต์ชื่อ attr และมีค่าเท่ากับ value
3. [attr!=value] ประกาศ CSS ที่มีแอตทริบิวต์ชื่อ attr และมีค่าไม่เท่ากับ value
4. [attr^=value] ประกาศ CSS ที่มีแอตทริบิวต์ชื่อ attr และมีค่าขึ้นต้นด้วยคำว่า value
5. [attr\$=value] ประกาศ CSS ที่มีแอตทริบิวต์ชื่อ attr และมีค่าลงท้ายด้วยคำว่า value
6. [attr*=value] ประกาศ CSS ที่มีแอตทริบิวต์ชื่อ attr และมีคำว่า value อญญาณใน

การนำ CSS ไปใช้งานกับ XHTML ในรูปแบบต่าง ๆ

Grouping เป็นการรวมกันของ Selector โดยที่มีค่าคุณสมบัติเช่นเดียวกัน เช่น

```
h1,h2{color:#cccccc}s
```

Universal เป็นการประกาศโดยใช้เครื่องหมาย * โดยมีความหมายถึง ทุกอิลิเมนต์

```
*{font-size:16px;}/*ทุกอิลิเมนต์กำหนดพ่อนต์ขนาด16px*/
```

Selector with DOM

DOM ย่อมาจาก Document Object Model

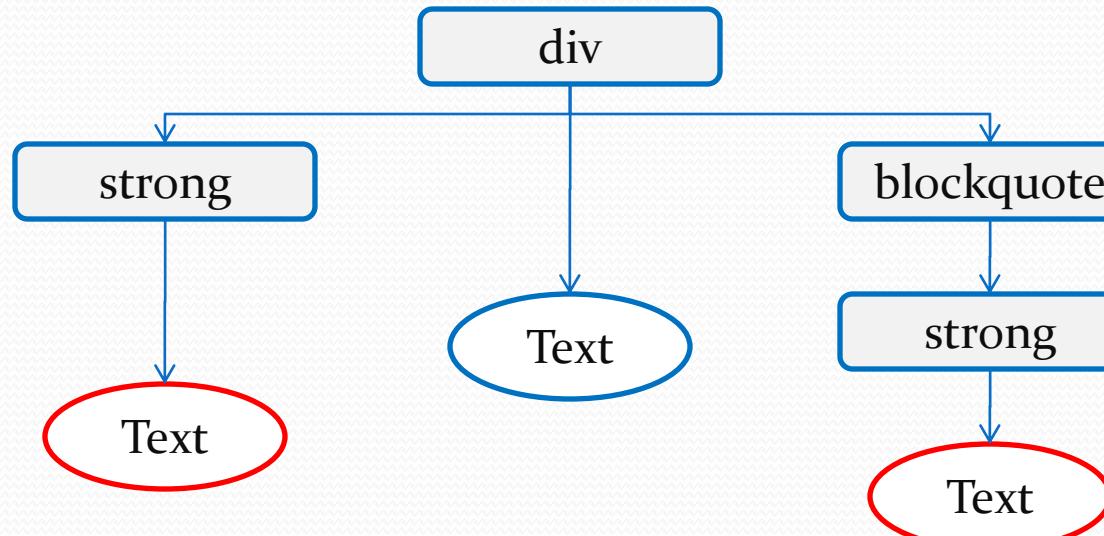
แนวคิดในการมองส่วนต่างๆ ของหน้าเว็บให้เป็น object เพื่อให้เราสามารถเรียกใช้ object เหล่านั้นได้

Descendant (ลูกหลาน)

เขียนโดยการเว้นช่องระหว่าง Selector

div strong{color:#cccccc;}

กรณีเขียนติดการเช่น div.red หมายถึง Tag div ที่มี class=red เช่น
`<div class="red"></div>`



Descendant (ลูกหลาน)

//Descendant

```
<style>
```

```
div strong{color:#cccccc;}
```

```
<style>
```

```
<div>
```

```
    <strong>Text</strong>
```

```
    <blockquote>
```

```
        <strong>Text</strong>
```

```
    </blockquote>
```

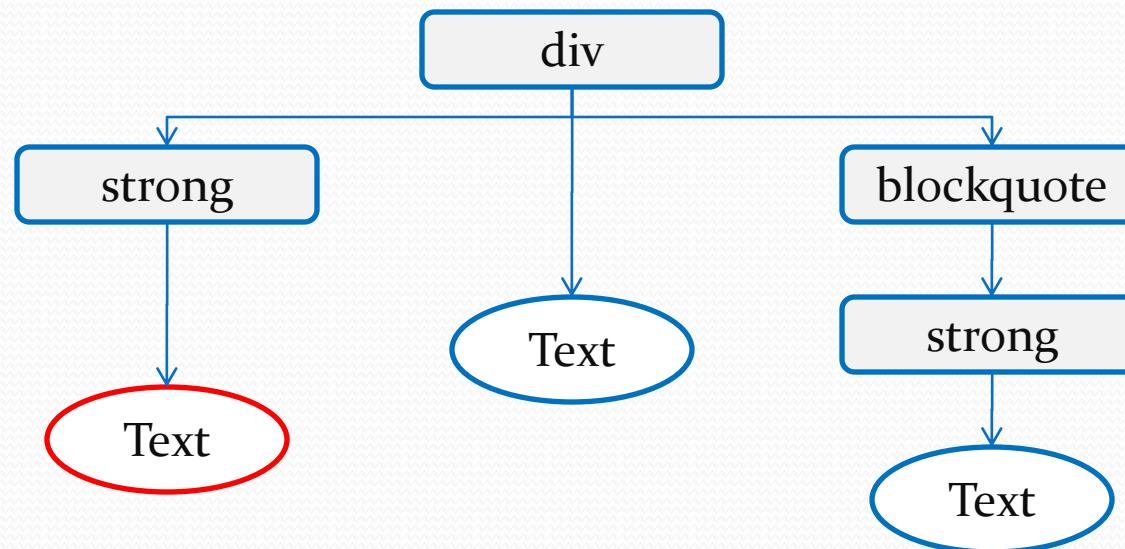
```
</div>
```

Child (ต้องเป็น element ลูก)

เป็นการเขียนโดยใช้เครื่องหมาย > เพื่อกำหนดถึง Element ลูก

ตัวอย่าง

```
div>strong{color:#005555;}
```



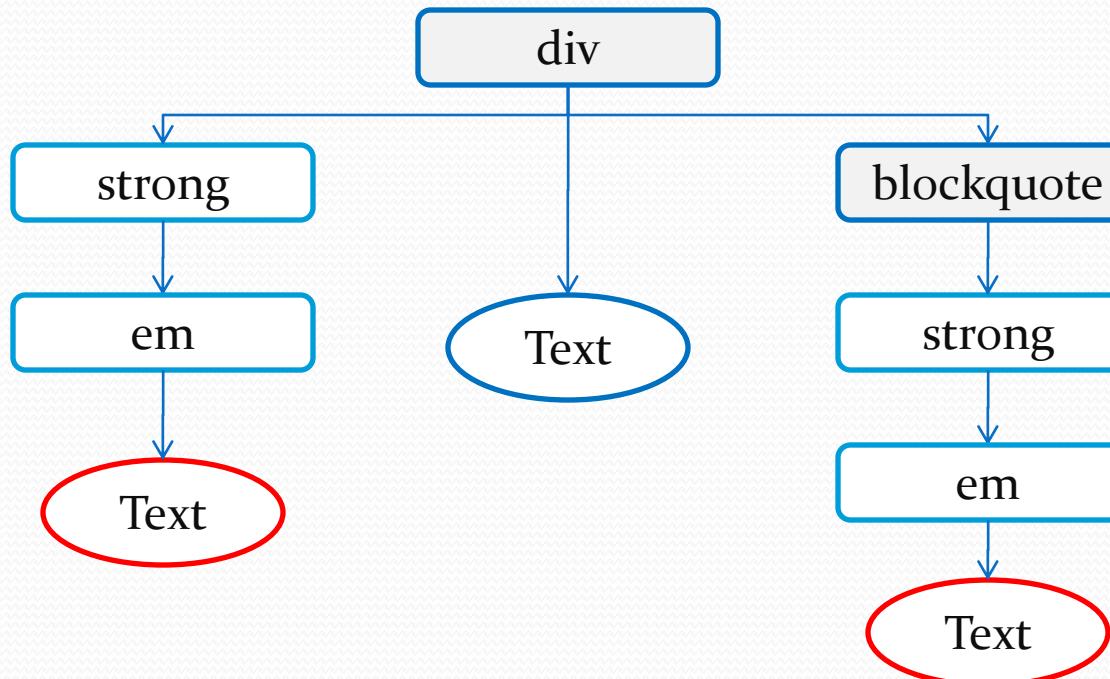
Child (ต้องเป็น element ลูก)

//Child

```
<style>
div>strong{color:#cccccc;}
</style>
<div>
    <strong>Text</strong>
    Text
    <blockquote>
        <strong>Text</strong>
    </blockquote>
</div>
```

Compound selector (การใช้งาน selector หลายรูปแบบร่วมกัน)

- Selector ที่ได้เรียนรู้ไป สามารถนำมาเขียนแบบผสมกันได้ เช่น
- `div strong>em{color: #00FF44;}`



Compound selector (การใช้งาน selector หลายรูปแบบร่วมกัน)

ลองเขียนโครงสร้าง XHTML มองในรูปแบบ
DOM(Document Object Model)

Compound selector (การใช้งาน selector หลายรูปแบบร่วมกัน)

//ACompound Selector

```
<style>
div strong>em{color:#cccccc;}
</style>
<div>
    <strong>
        <em>Text</em>
    </strong>
    Text
    <blockquote>
        <strong>
            <em>Text</em>
        </strong>
    </blockquote>
</div>
```

Pseudo-classes

เป็นคลาสที่ CSS กำหนดไว้ให้ใช้งานแล้วดังนี้

:first-child

:link

:visted

:hover

:active

:focus

การใช้งาน

Selector:pseudo-class{ ... }

ตัวอย่าง

a:link{text-decoration:none;}

Properties and Values

กลุ่มคุณสมบัติ

Font คุณสมบัติเกี่ยวกับตัวอักษร font-size,font-weight

Text คุณสมบัติเกี่ยวกับข้อความ text-align,text-decoration

Color and Background คุณสมบัติเกี่ยวกับสีและพื้นหลัง
color,background-color,background-image:url

Box คุณสมบัติเกี่ยวกับ Box Model margin,padding

Length ค่าเป็นตัวเลข pixels, pt, em

- px หรือ pixel
- em หรือ emphasize เทียบกับขนาดหน้าจอ
- % หรือ percentage
- pt หรือ point

Percentage ค่าเป็นตัวเลขมีเครื่องหมาย % กำกับ

Color ค่าสีกำหนดเป็นหัวสี หรือ key word ของสี #cccccc หรือ red,green

URL กำหนดค่าที่ต้องการอ้างถึงไฟล์อื่นๆ(url(url))

Short Hand Properties

การกำหนดคุณสมบัติโดยการเขียนแบบย่อ เป็นการเขียน CSS ที่อยู่ในกลุ่ม properties เดียวกันโดยกำหนดดังนี้

เขียน CSS แบบปกติ

```
.font{font:bold; font-size:12px; family:Tahoma;}
```

เขียนแบบย่อ

```
.font{font:bold 12px Tahoma;}
```

Important สำคัญที่สุด

ป้องกันการเขียนค่าทับโดยการกำหนด !important หลังค่า properties ที่ต้องการ
ตัวอย่าง

```
.font{ color:red!important; color:green;}
```

```
/*ไม่สามารถกำหนดค่าทับของเดิมได้*/
```

Firefox=red

Crome=red

IE=green

BOX Model



Block Element Properties

Display:block;แสดงในรูแบบของกล่องข้อมูล (ลงบรรทัดใหม่)

Display:inline;แสดงในบรรทัด ("ไม่ลงบรรทัดใหม่")

Width:value;ค่าความกว้าง

Height:value;ค่าความสูง

Margin:value;ค่าความห่างจากภายนอกของกล่องข้อมูล

Padding:value;ค่าความห่างจากภายในของกล่องข้อมูล

Border:value;ค่าความหนาของเส้นกรอบของกล่องข้อมูล

Float:value;กำหนดตำแหน่งในการแสดงผล

Clear:value;ลบค่า float ของกล่องข้อมูลที่อยู่ก่อนหน้า

Position:value;ค่าตำแหน่งการแสดงผล

Short Hand : margin,padding

ทุกด้านที่มีระยะเท่ากัน

Box{margin:20px;}

Top-Bottom,left-right

Box{margin:20px 10px;}

Top,left-right,bottom

Box{margin:20px 50px 10px;}

All

Box{margin:20px 50px 10px 30px;}

ตัวอย่างการใช้ **Float**

Float เป็นการจัดตำแหน่ง Laout ให้อยู่ในตำแหน่งที่ต้องการ float จะมีความสำคัญมากในการออกแบบ Laout

รูปแบบการใช้งาน

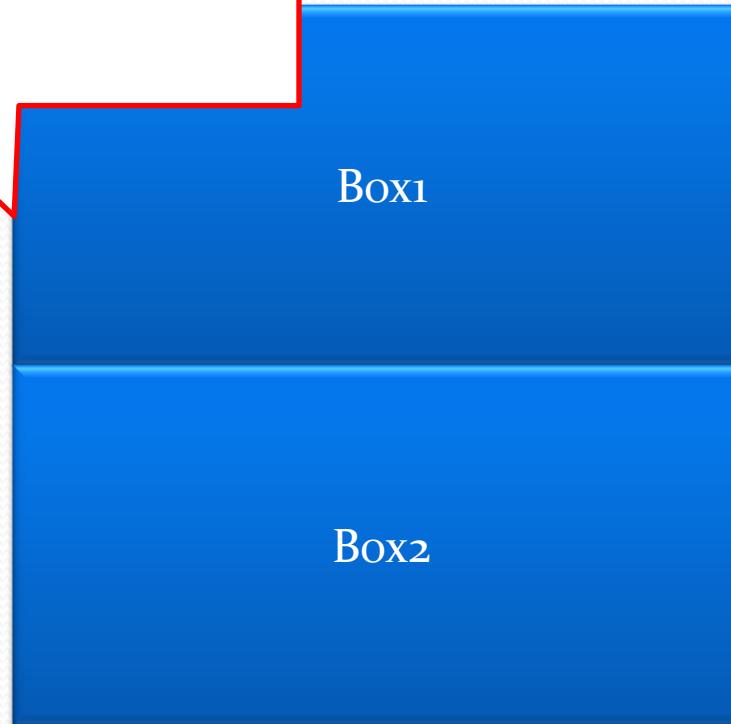
.box{float:left} จัดตำแหน่ง Laout อยู่ด้านซ้าย

.box{float:right} จัดตำแหน่ง Laout อยู่ด้านขวา

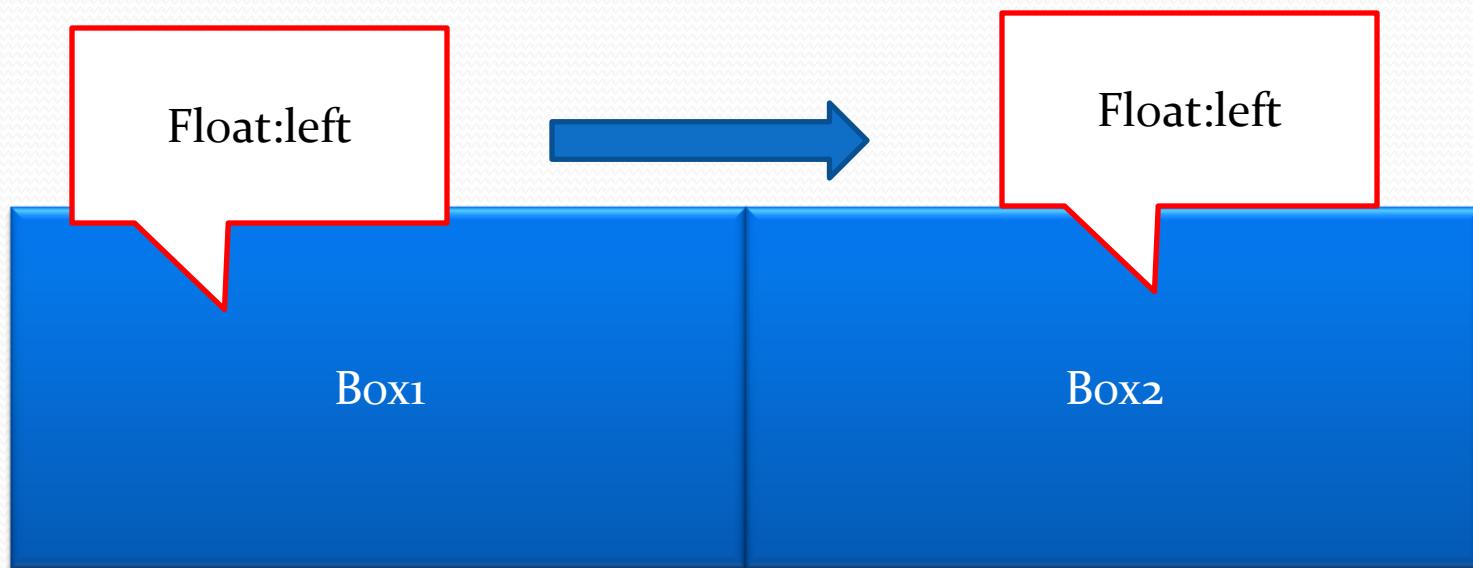
.box{clear:both} ลบค่า float ที่อยู่ตำแหน่งก่อนหน้า

ตัวอย่างไม่ใช้งาน float

ไม่มีการกำหนด float จะทำให้ Layout เรียงต่อกันลงไป
ข้างล่าง

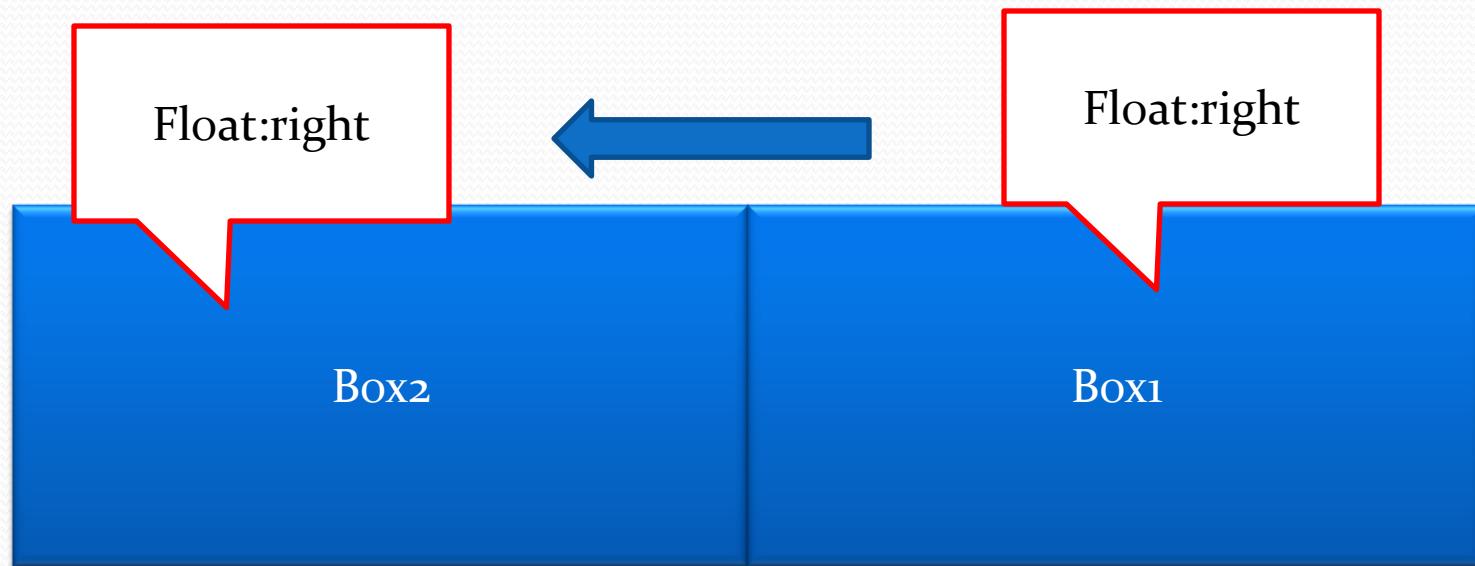


ตัวอย่างการใช้งาน float:left



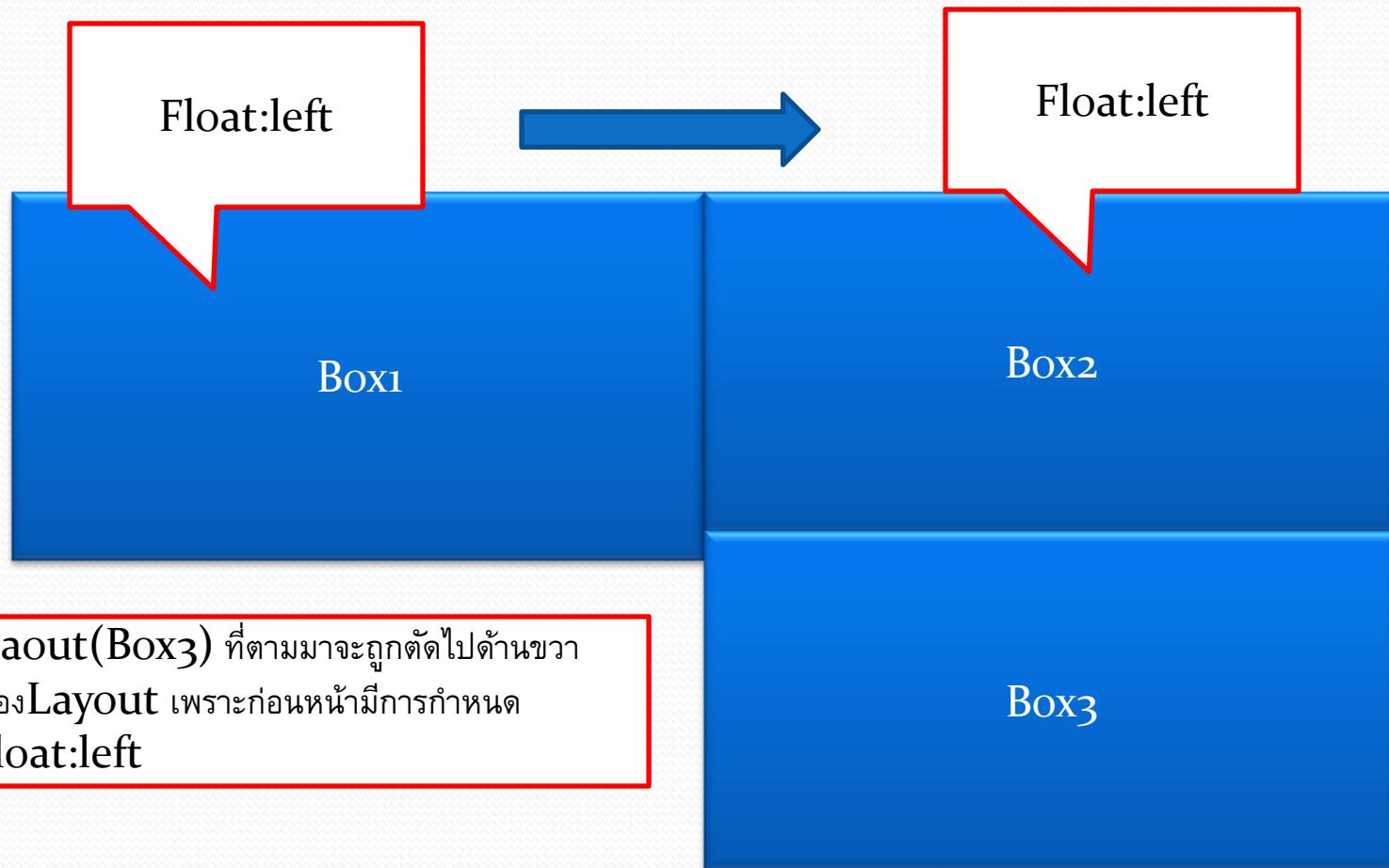
เมื่อกำหนด `float:left` จะทำให้ Layout เรียงต่อกันไปทางด้านขวา

ตัวอย่างการใช้งาน float:right

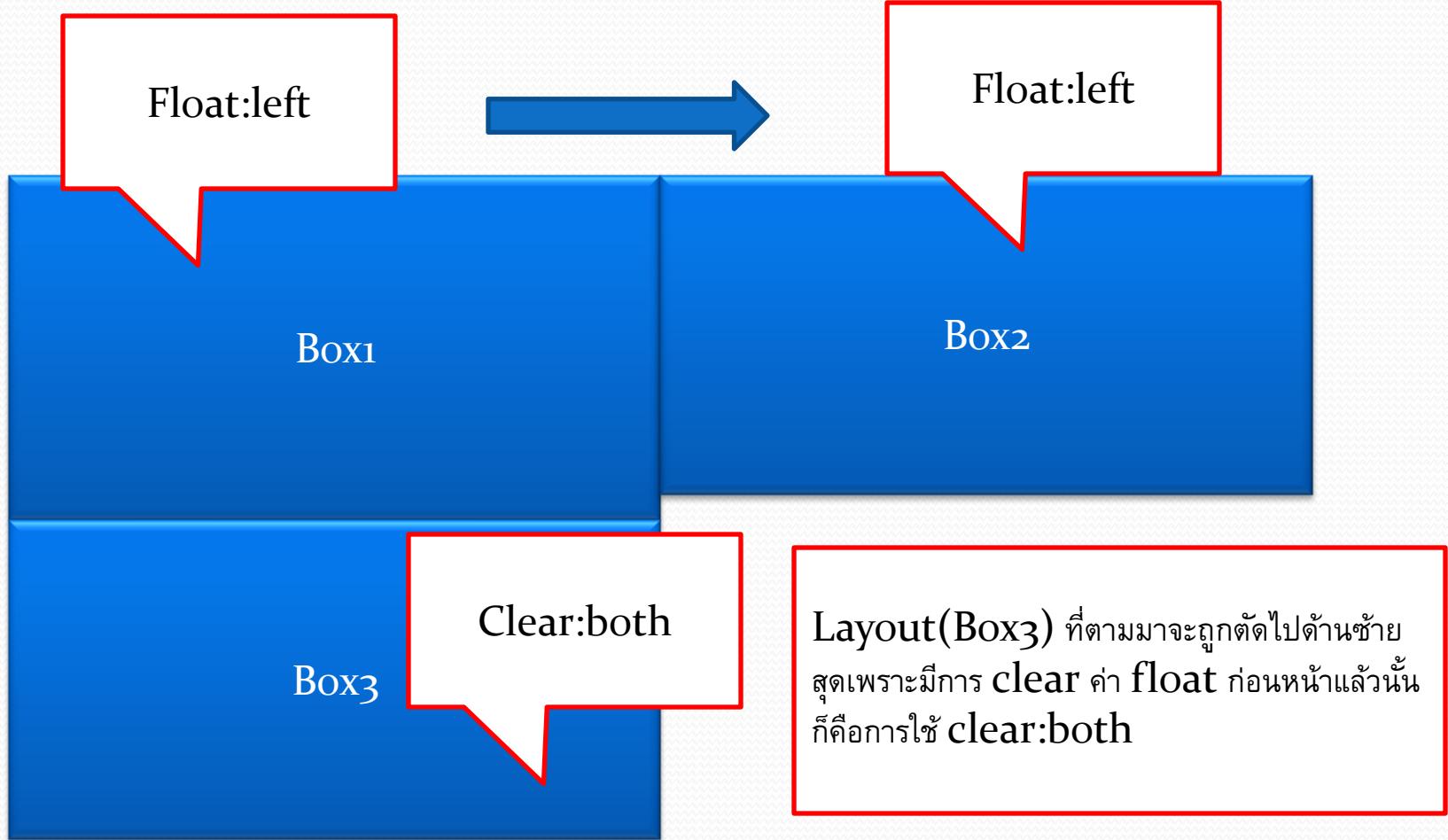


เมื่อกำหนด `float:right` จะทำให้ Layout เรียงต่อกันไปทางด้านซ้าย

ตัวอย่างไม่ใช้ clear:both



ตัวอย่างการใช้งาน clear:both



Position Properties

การกำหนดวิธีการแสดงผล และตำแหน่งการแสดง

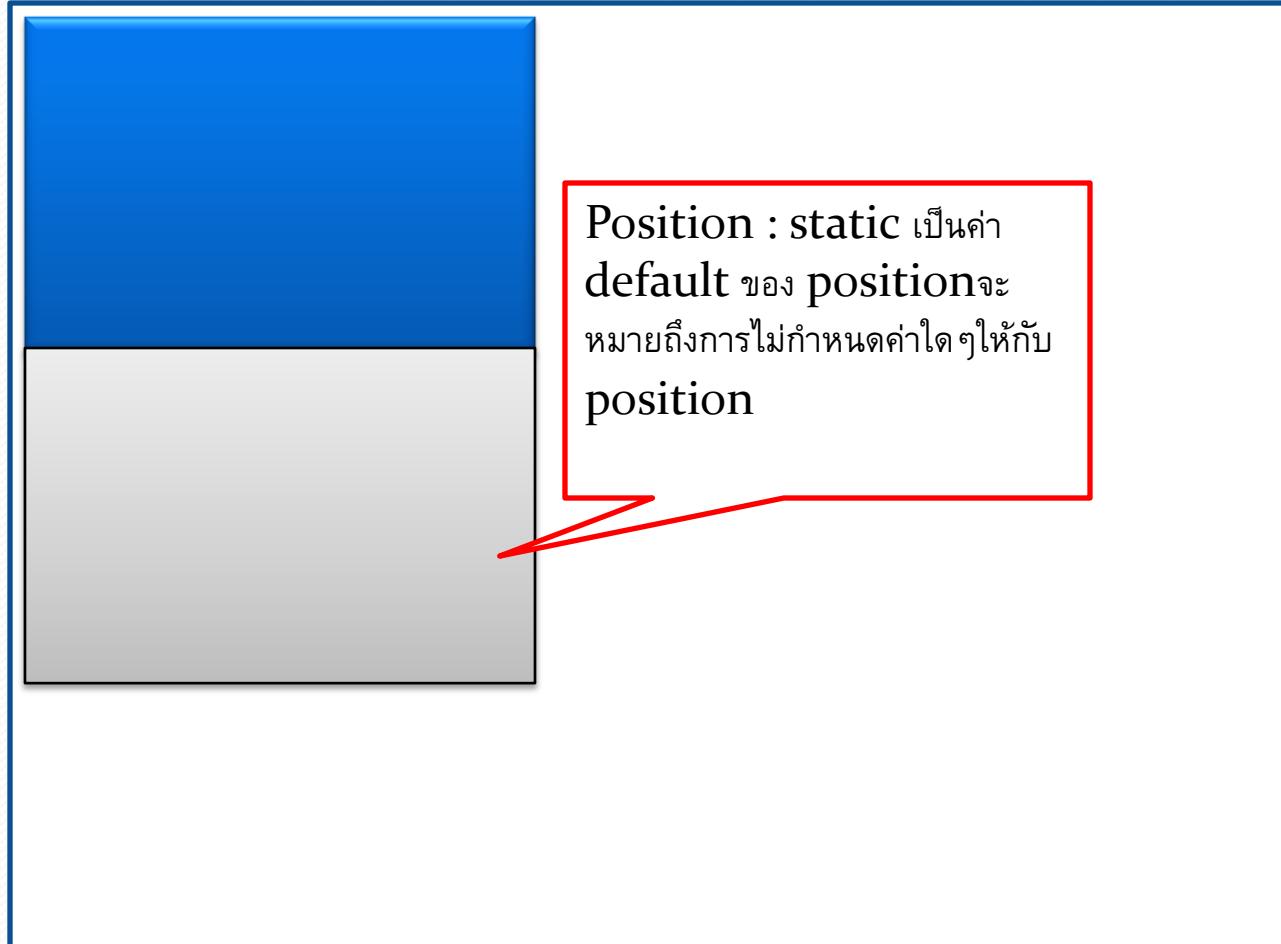
Position มีรูปแบบการแสดงผล 4 รูปแบบดังนี้

- **Absolute** ตำแหน่งที่แท้จริงโดยยึดจากขอบของ Browser
- **Relative** ตำแหน่งที่กำหนดจากความสัมพันธ์จาก Object ใกล้เคียง
- **Fixed** ให้ผลคล้ายกับ absolute แต่ยึดที่ขอบของ Browser จะปรากฏบนหน้าเว็บเพจเสมอ
- **Static(defalult)**ไม่มีการกำหนดตำแหน่ง ใช้ร่วมกับ top,left,bottom,right

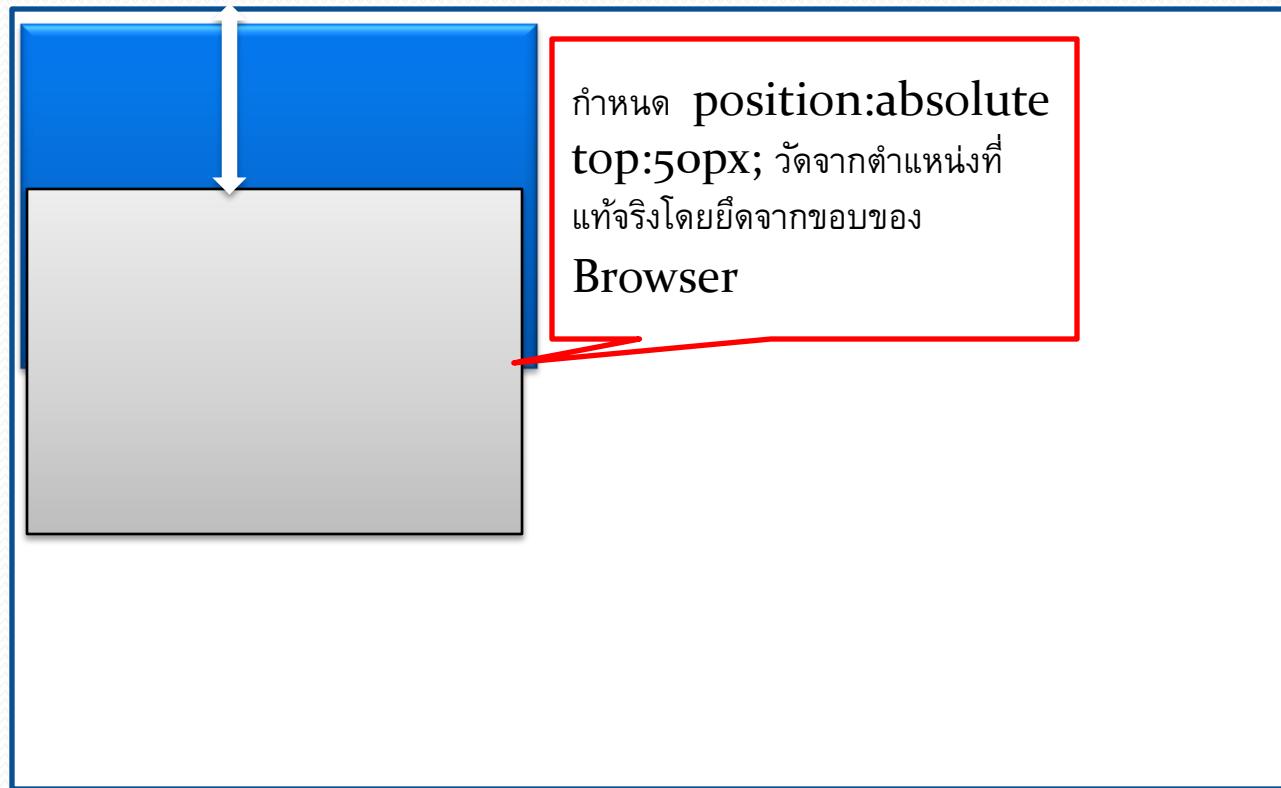
ตัวอย่าง

```
#box{position:relative; top:10px; left:50px;}
```

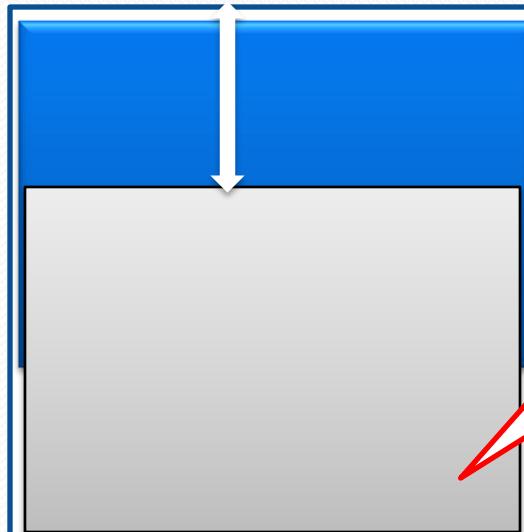
ตัวอย่างการแสดงผลในรูปแบบ static



ตัวอย่างการแสดงผลในรูปแบบ absolute

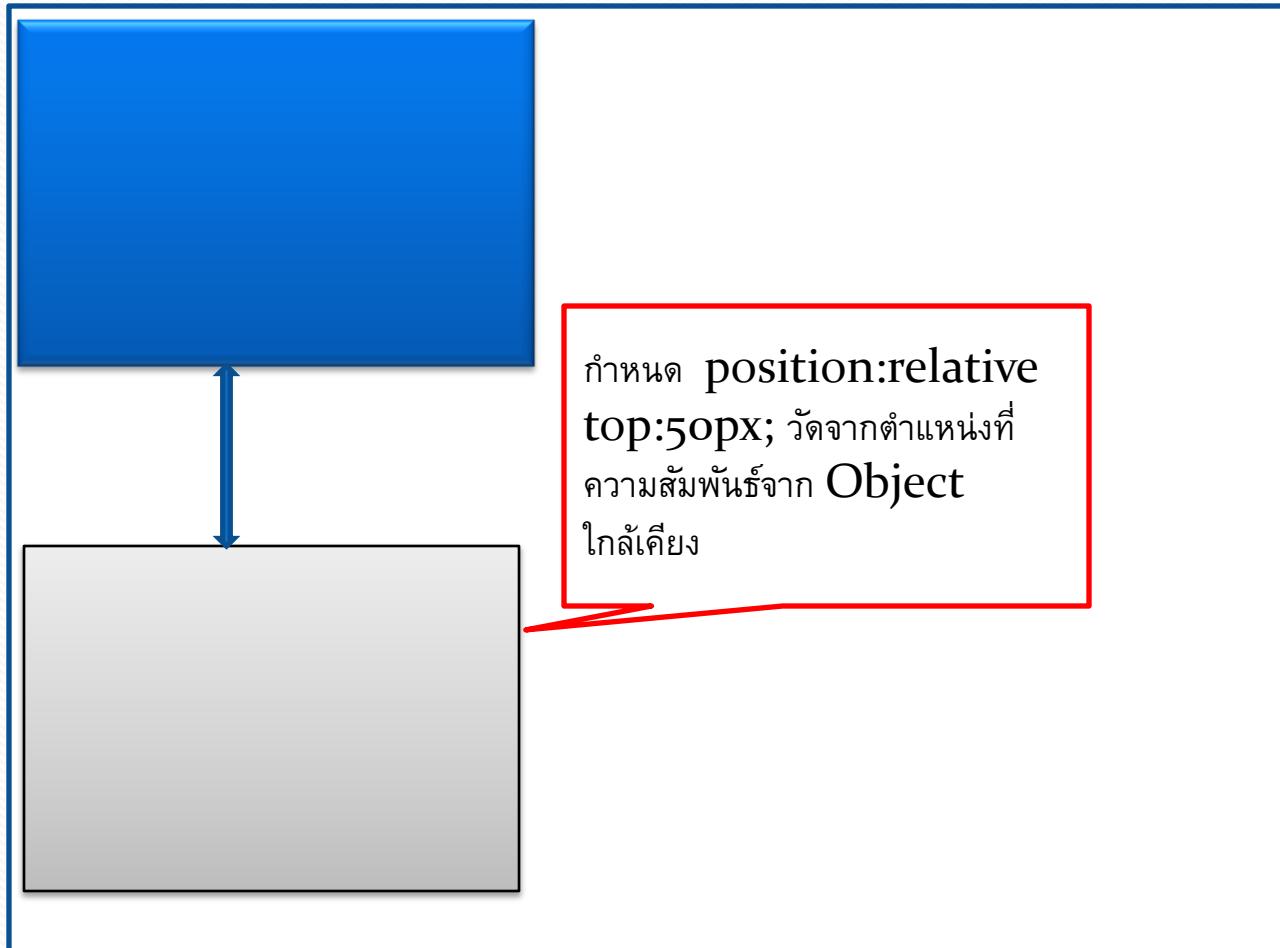


ตัวอย่างการแสดงผลในรูปแบบ fixed



กำหนด position:fixed
top:50px; วัดจากตำแหน่งที่
แท้จริงโดยยึดจากขอบของ
Browser แต่จะแสดง
ผลตำแหน่งเดิมตลอดถึงแม้เลื่อน
scroll bar(คล้ายกับ
absolute)

ตัวอย่างการแสดงผลในรูปแบบ relative



ตัวอย่าง Layout โดย CSS

Header

MainContent

Menu

Content

Footer



CSS Training End