

# CSS เบื้องต้น

Cascading Style Sheet (2020)



เอกสารแจกฟรี ห้ามจำหน่าย!!!!

# ติดตามผู้เชี่ยวชาญ ผ่านช่องทางยูทูป



[https://www.youtube.com/channel/UCQ1r\\_4x-P-fETLIU4pqf98w](https://www.youtube.com/channel/UCQ1r_4x-P-fETLIU4pqf98w)



<https://www.facebook.com/KongRuksiamTutorial/>

# หรือสแกน QR CODE



# Introduction

**CSS (Cascading Style Sheets)** เป็นเทคโนโลยีที่นำมาใช้จัดรูปแบบและควบคุมการแสดงผลของเว็บ เช่น การกำหนดขนาด สี แบบอักษร เส้นขอบ พื้นหลัง และตำแหน่งของข้อความ รูปภาพ เป็นต้น



# โครงสร้างของ CSS

มีส่วนประกอบอยู่ 2 ส่วน คือ Selector และ Declaration

- **Selector** คือแท็ก HTML, ID และ Class ที่ต้องการกำหนด
- **Declaration** ใช้สำหรับกำหนดค่าให้กับ Selector



# โครงสร้างของ CSS

Selector{  
Declaration

}

แท็กอ้างอิง{  
คุณสมบัติ

}



# โครงสร้างของ CSS

**Selector** คือแท็ก HTML, ID และ Class ที่ต้องการกำหนดคุณสมบัติ แบ่งออกเป็น 3 รูปแบบ

- **Element Selector** คือ การกำหนด style ให้แท็ก HTML โดยตรง
- **Class Selector** คือ การกำหนด style โดยใช้ชื่อ Class เป็น Selector โดยใช้เครื่องหมายจุด (.) สามารถมีค่าซ้ำกันได้
- **ID Selector** คือ การกำหนด style โดยระบุชื่อ ID เป็นรหัสเฉพาะของแท็กหลังเครื่องหมาย # โดย ID ไม่สามารถกำหนดค่าซ้ำกันได้

# โครงสร้างของ CSS

**Declaration** ใช้กำหนดค่าให้กับ selector ซึ่งมีส่วนประกอบอยู่ด้วยกัน 2 ส่วน คือ

- **Property** หรือ **Attribute** คือคุณสมบัติที่ต้องการกำหนดให้กับ Selector
- **Value** คือ ค่าที่กำหนดให้กับ Property หรือ Attribute





# โครงสร้างของ CSS

การกำหนดส่วนของ property กับ value จะแยกด้วยเครื่องหมาย colon (:) ถ้ามีมากกว่า 1 property จะต้องแยกด้วยเครื่องหมาย semi-colon (;) โดยมีรูปแบบดังนี้

```
selector { property : value; property : value ...}
```



```
<p>kongruksiam</p>
```

Tag Selector

```
p {  
  color: red;  
  text-align: center;  
  padding: 10px;  
}
```

Value

Property หรือ Attribute



```
<h2 class="title">kong</h2>  
<p class="title">ruksiam</p>
```

Class Selector

```
.title{  
  color: pink;  
  text-align:center;  
  padding: 10px;  
}
```

Value

Property หรือ Attribute



```
<h2 id="title">kong</h2>  
<p id="message">ruksiam</p>
```

Id Selector

```
#message{  
  color: green;  
  text-align:center;  
  padding: 10px;  
}
```

Value

Property หรือ Attribute



```
<div id="box">  
<h2 class="title">kong</h2>  
<p id="message">ruksiam</p>  
</div>
```

Union Selector

↓

```
.title , #message , #box{  
  color: green;  
  text-align:center;  
  font-size : 18px;  
}
```

↑  
Value

Property หรือ Attribute



```
<p class="error">kongruksiam</p>  
<p class="success">kongruksiam</p>
```

Tag.Class Selector



```
p.error{  
  color: red;  
}  
p.success{  
  color: green;  
}
```



# รูปแบบการประกาศใช้ CSS

## แบบ Internal

- **Inline** เป็นการแทรก attribute style ลงในแท็ก HTML
- **Embedded** การกำหนดรูปแบบ style sheet ภายในไฟล์เดียวกันกับไฟล์เว็บโดยการเขียนภายใต้ `<style> ... </style>`

## แบบ External

จะกำหนดไฟล์ style sheet แยกเป็นอีกไฟล์โดยมีนามสกุล .css แล้วค่อยเชื่อมต่อเข้ามาใช้ภายในเว็บเพจ เหมาะสำหรับการกำหนดให้หลายๆเว็บเพจใช้รูปแบบ style เดียวกันผ่านการเขียน style แค่ครั้งเดียว

# แบบ Inline

ใช้กำหนด style ให้กับอีลิเมนต์ของ HTML เพียงอันเดียวเป็นการเฉพาะ

```
<p style="color:green">kong ruksiam.</p>
```





# แบบ Internal / Embedded

กำหนด style เอาไว้ที่ส่วนหัวของเว็บเพจโดยกำหนดผ่าน  
<style>...</style> เพื่อกำหนด style ให้กับ element ในเว็บเพจนั้นๆ

```
<style>
p{
  color:red;
  text-align:center;
}
</style>
```



# แบบ External

จะกำหนดไฟล์ style sheet แยกเป็นอีกไฟล์โดยมีนามสกุล .css แล้วค่อยเชื่อมเข้ามาใช้ภายในเว็บเพจโดยใช้ <link>

**style.css**

```
p{  
  color:red;  
  text-align:center;  
}
```

**ไฟล์ html**

```
<head>  
<title></title>  
<link rel="stylesheet"href="style.css">  
</head>
```

# การเขียน Comment

ส่วนที่ใช้ในการอธิบายโค้ด ซึ่งจะช่วยให้สามารถเข้าใจ และสามารถแก้ไขโค้ดได้ในภายหลังได้

รูปแบบ

`/* ข้อความอธิบายโค้ด */`



# หน่วย (Unit) ใน CSS (Absolute) แบบตายตัว

px	pixel (พิกเซล) เป็นหน่วยที่นิยมใช้มากที่สุด โดยที่ $1\text{px} = 0.75\text{pt}$ สัมพันธ์กับรายละเอียดหน้าจอ
pt	point โดยที่ $1\text{pt} = 1/72\text{ inches}$ ใช้ในงานสิ่งพิมพ์
cm	เซนติเมตร
mm	มิลลิเมตร
in	inches ( $1\text{ in} = 96\text{px} = 2.54\text{cm}$ )
pc	picas ( $1\text{pc} = 12\text{pt}$ )

# หน่วย (Unit) ใน CSS (Relative) แบบอัตราส่วน

%	เป็นการกำหนดขนาดเป็นเปอร์เซ็นต์ มักใช้กับความกว้างหรือสูง
em	อ้างอิงขนาดกับ element parent ที่ใกล้ที่สุด ใช้ในการกำหนดขนาดจำนวนเท่าของขนาดปัจจุบันเช่น หากขนาดที่ใช้ตอนนี้คือ 10px <ul style="list-style-type: none"><li>- ถ้ากำหนดขนาดเป็น 2em จะหมายถึง 2 เท่าของขนาดปัจจุบัน คือ 20px</li><li>- ถ้ากำหนดขนาดเป็น 1.4em จะหมายถึง 1.4 เท่าของขนาดปัจจุบัน คือ 14px</li><li>- ถ้ากำหนดเป็นขนาดเดิมก็กำหนดเป็น 1em = 10px</li></ul>
rem	กำหนดขนาดโดยอ้างอิงกับ root element ปกติ font-size จะอยู่ที่ 16px
vw	1% หรือ 1/100 ของ viewport width <ul style="list-style-type: none"><li>- width ของ browser viewport เท่ากับ 750px ค่า 1vw = 7.5px</li></ul>
vh	1% หรือ 1/100 ของ viewport height <ul style="list-style-type: none"><li>- height ของ browser viewport เท่ากับ 900px ค่า 1vh = 9px</li></ul>
vmin , vmax	กำหนดค่าต่ำสุด และค่าสูงสุดของ viewport

# Universal Selector

กำหนด style sheet มีผลต่อทุกแท็กของเว็บเพจ โดยใช้เครื่องหมาย \*

```
*{  
  color:red;  
}
```



# การกำหนดชนิดฟอนต์ (Font)

**font-family** กำหนดชนิดของฟอนต์ในหน้าเว็บกำหนดได้ทั้งแบบฟอนต์เดียวหรือหลายฟอนต์ โดยกำหนดได้ดังนี้ คือ

**ฟอนต์แบบที่ 1;**

**ฟอนต์แบบที่ 1 , ฟอนต์แบบที่ 2, ชนิดฟอนต์;**

สามารถกำหนดฟอนต์ได้มากกว่า 1 ชนิดในกรณีBrowserไม่รองรับฟอนต์ชนิดแรก ก็จะนำฟอนต์ลำดับถัดไปมาใช้งานแทน

# การกำหนดขนาดข้อความ

**font-size** กำหนดขนาดข้อความหรือตัวอักษรในหน้าเว็บโดยกำหนด 2 รูปแบบ คือ

**แบบค่าคงที่** : xx-small, x-small, medium, large, x-large, xx-large

**แบบตัวเลข** : 10px 20px





# การกำหนดความหนาข้อความ

**font-weight** กำหนดได้ 2 รูปแบบ

**แบบค่าคงที่ :** lighter , bold (ตัวหนา) , bolder (ตัวหนากว่า)

**แบบตัวเลข :** 100, 200, ..., 800, 900

**font-style** กำหนดตัวเอียง โดยค่าที่กำหนดได้คือ italic , oblique (45 องศา)

# การกำหนดสี (Colors)

**color** กำหนดได้ 4 รูปแบบ

**ชื่อสี** : green , red , yellow , pink

**RGB** : *rgb(red, green, blue)* , *rgb(30%, 20%, 45%)*

**Hexadecimal** : #000 #FFF

**HSL** : (*hue, saturation, lightness*)



# การกำหนดลักษณะข้อความ

**Text-decoration** กำหนดได้ทั้ง 4 แบบ

**none** : (ค่าว่าง)

**underline** : (ขีดเส้นใต้)

**overline** : (ขีดเส้นเหนือข้อความ)

**line-through** : (ขีดเส้นทับข้อความ)



# การจัดแนวข้อความ

**Text-align** กำหนดได้ 4 รูปแบบ

- left, right, center (ชิดด้านซ้าย ขวา และกึ่งกลางตามลำดับ)
- justify (การกระจายข้อความให้ทุกบรรทัดมีความกว้างเท่ากัน)



# กำหนดความกว้างและความสูง

- **width** ใช้กำหนดความกว้างของวัตถุที่ต้องการ
  - auto คือ browser จะกำหนดให้เอง
  - length คือ ความกว้างแบบหน่วยวัด เช่น px cm เป็นต้น
  - % คือ ความกว้างแบบ % (แบบยืดหยุ่น)
- **height** ใช้กำหนดความสูงของวัตถุที่ต้องการ
  - auto คือ browser จะกำหนดให้เอง
  - length คือ ความสูงแบบหน่วยวัด เช่น px cm เป็นต้น
  - % คือ ความสูงแบบ % (แบบยืดหยุ่น)



# ความกว้างและความสูงด้วย max , min

- width-min ใช้กำหนดความกว้างต่ำสุดของวัตถุ
- width-max ใช้กำหนดความกว้างสูงสุดของวัตถุ
- height-min ใช้กำหนดความสูงต่ำสุดของวัตถุ
- height-max ใช้กำหนดความสูงสูงสุดของวัตถุ

## หน่วยที่รองรับ

- none ไม่กำหนด (default)
- length คือ กำหนดแบบหน่วยวัด เช่น px cm เป็นต้น
- % คือ กำหนดแบบ % (แบบยืดหยุ่น)



# การกำหนดเส้นขอบ

**Border** คือการกำหนดเส้นขอบ โดยมีรูปแบบดังนี้

`border : <รูปแบบเส้น> <ขนาด> <สี>`

รูปแบบเส้น ประกอบด้วย solid (เส้นทึบ) dotted (แบบจุด) dashed (เส้นปะ)

ขนาด คือ ความกว้างของเส้นกำหนดเป็นตัวเลขตามหน่วย

สี กำหนดรูปแบบเหมือน font เลย เช่น ชื่อสี , rgb , hex



# การกำหนดรูปแบบเส้นขอบ (style)

Border คือการกำหนดเส้นขอบ โดยมีรูปแบบดังนี้

`border-style : <รูปแบบเส้น>`

รูปแบบเส้น ประกอบด้วย solid (เส้นทึบ) dotted (แบบจุด) dashed (เส้นปะ)

`border-xxx-style : <รูปแบบเส้น>`

xxx คือ ทิศทางของรูปแบบเส้นขอบ





# การกำหนดสีเส้นขอบ (color)

Border คือการกำหนดเส้นขอบ โดยมีรูปแบบดังนี้

`border-color : <สี>`

สี ประกอบด้วย RGB , ชื่อสี , HEX

`border-xxx-color: <สี>`

xxx คือ ทิศทางของรูปแบบเส้นขอบ



# การกำหนดขนาดเส้นขอบ (width)

Border คือการกำหนดเส้นขอบ โดยมีรูปแบบดังนี้

```
border-width : <ขนาด>
```

```
border-xxx-width: <ขนาด>
```

xxx คือ ทิศทางของรูปแบบเส้นขอบ



# การกำหนดขนาดเส้นขอบ (width)

Border คือการกำหนดเส้นขอบ โดยมีรูปแบบดังนี้

`border-width : <ขนาด>`

แบบระบุค่าคงที่ :

medium - ขอบปานกลาง

thin - ขอบบาง

thick - ขอบหนา

initial - ค่าเริ่มต้น

inherit - อ้างอิงตาม parent element



# การกำหนดความโค้งเส้นขอบ (radius)

Border คือการกำหนดเส้นขอบ โดยมีรูปแบบดังนี้

`border-radius : <ค่าความโค้ง>`

`border-xxx-radius: <ค่าความโค้ง>`

xxx คือ ทิศทางของรูปแบบเส้นขอบ



# การกำหนดความโค้งเส้นขอบ

## เขียน 4 รูปแบบ

`border-radius: 10px 5px 15px 20px;`

- top-left = 10px
- top-right = 5px
- bottom-right = 15px
- bottom-left = 20px

## เขียน 3 รูปแบบ

`border-radius: 10px 5px 15px;`

- top-left = 10px
- top-right , bottom-left = 5px
- bottom-right = 15px

## เขียน 2 รูปแบบ

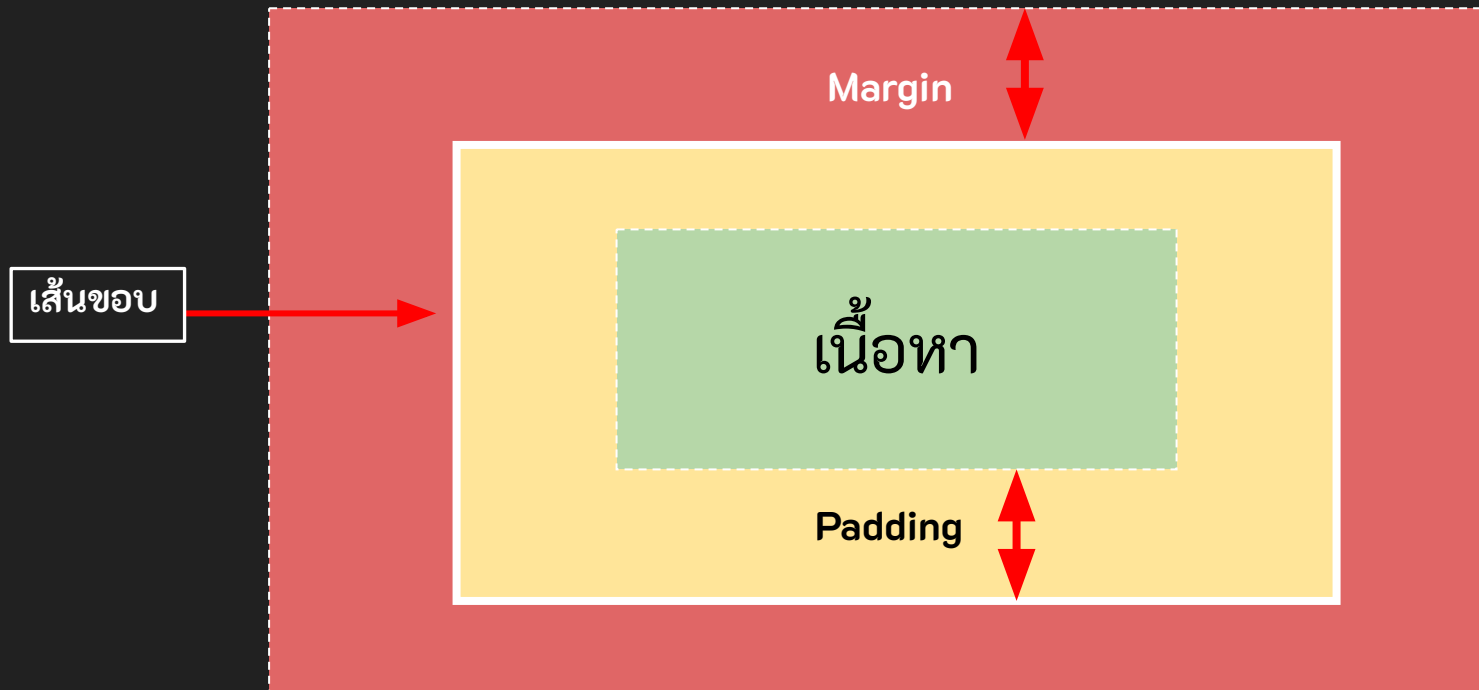
`border-radius : 10px 5px;`

- top-left , bottom-right = 10px
- top-right , bottom-left = 5px

`border-radius : 10px;`

ทุกทิศทางมีค่า = 10px

# Box Model



# การกำหนดพื้นที่รอบ Element

- **Margin** กำหนดระยะห่างจากเส้นขอบหรือพื้นที่ภายนอกของ Element เทียบกับวัตถุอื่นๆ (**ช่องว่างข้างนอก**)
- **Padding** กำหนดพื้นที่ระยะห่างภายในของ Element (**ช่องว่างข้างใน**)
- **Border** กำหนดเส้นขอบ

กำหนดได้ทั้งรูปแบบด้านบน (top) ด้านล่าง (bottom) ด้านขวา (right) และด้านซ้าย (left) ทั้ง margin และ padding เช่น margin-left , margin-right , padding-left, padding-bottom

หน่วยที่กำหนดได้ เช่น px, pt, cm อื่นๆ หรือแบบเปอร์เซ็นต์ (%) โดยมีค่า default เป็น 0

# การกำหนดพื้นที่รอบ Element

**Margin** กำหนดระยะห่างจากเส้นขอบหรือพื้นที่ภายนอกของ Element

margin: ค่าที่กำหนด ;

margin-left:ค่าที่กำหนด;

margin-right:ค่าที่กำหนด;

margin-top:ค่าที่กำหนด;

margin-bottom:ค่าที่กำหนด;

margin:auto;(ให้ browser จัดให้อย่างพอดีจะกำหนดกึ่งกลาง)





# การกำหนดพื้นที่รอบ Element

## เขียน 4 รูปแบบ

margin: 10px 5px 15px 20px;

- top margin = 10px
- right margin = 5px
- bottom margin = 15px
- left margin = 20px

## เขียน 3 รูปแบบ

margin: 10px 5px 15px;

- top margin = 10px
- right left = 5px
- bottom margin = 15px

## เขียน 2 รูปแบบ

margin: 10px 5px;

- top bottom margin = 10px
- right left = 5px

margin: 10px;

ทุกทิศทางมีค่า = 10px

**\*\* ลำดับ top, right, bottom, left**

# การกำหนดพื้นที่รอบ Element

**Padding** กำหนดพื้นที่ระยะห่างภายในของ Element

padding: ค่าที่กำหนด ;

padding-left:ค่าที่กำหนด;

padding-right:ค่าที่กำหนด;

padding-top:ค่าที่กำหนด;

padding-bottom:ค่าที่กำหนด;

padding:auto;(ให้ browser จัดให้เอง ปกติจะกำหนดกึ่งกลาง)



# การกำหนดพื้นที่รอบ Element

## เขียน 4 รูปแบบ

padding: 10px 5px 15px 20px;

- top padding = 10px
- right padding = 5px
- bottom padding = 15px
- left padding = 20px

## เขียน 3 รูปแบบ

padding: 10px 5px 15px;

- top padding = 10px
- right left = 5px
- bottom padding = 15px

## เขียน 2 รูปแบบ

padding: 10px 5px;

- top bottom padding = 10px
- right left = 5px

padding: 10px;

ทุกทิศทางมีค่า = 10px

**\*\* ลำดับ top, right, bottom, left**

# การกำหนดสีพื้นหลัง (background)

**background-color** ใช้เปลี่ยนสีพื้นหลังหรือกำหนดความโปร่งใสให้กับพื้นหลังและ Element ต่าง ๆ

**background-color : <สี> | transparent**

สี ประกอบด้วย RGB , ชื่อสี , HEX



# การกำหนดภาพพื้นหลัง (background)

**background-image** ใช้เปลี่ยนภาพพื้นหลัง

**background-image : url(รูปภาพ)**

**url('image.png')**

**url('http://www.unsplash.....png')**

# รูปแบบภาพพื้นหลัง (background-repeat)

```
background-image:url(xxx);
```

```
background-repeat : xxx;
```

repeat (ค่า default ทำให้ภาพเต็มพื้นที่ของอีลิเมนต์)

repeat-x (จัดเรียงเฉพาะในแนวแกน X)

repeat-y (จัดเรียงเฉพาะในแนวแกน Y)

no-repeat (ไม่จัดเรียงแนวแกน x,y แสดงภาพพื้นหลังภาพเดียว)



# กำหนดคุณสมบัติพื้นหลัง

โดยทั่วไปภาพพื้นหลังจะถูกเลื่อนตามการ scroll ของเว็บเบราว์เซอร์ ถ้าต้องการตรึงภาพพื้นหลังให้อยู่กับที่สามารถทำได้โดยใช้ **background-attachment**

- ❑ scroll (คือการเลื่อนพื้นหลังตาม scrollbar)
- ❑ fixed (คือการตรึงพื้นหลังไว้กับที่)



# กำหนดตำแหน่งพื้นหลัง

สามารถกำหนดได้ว่าจะเริ่มต้นวางพื้นหลังที่ตำแหน่งใด โดยใช้ **background-position** เช่น

`background-position : center right ;` (กลาง ขวา)

- ❑ left top หรือ left center หรือ left bottom
- ❑ right top หรือ right center หรือ right bottom
- ❑ center top หรือ center center หรือ center bottom





# กำหนดขนาดภาพ (Background-size)

## background-size : กำหนดค่า

- ❑ auto ค่าปกติภาพเท่าเดิม
- ❑ px หรือ %
- ❑ cover ขยายภาพทั้งความกว้างความสูงเต็มจอในอัตราส่วนเท่ากัน
- ❑ contain ขยายภาพทั้งความกว้างความสูงจนขอบ Browser



# กำหนดพื้นหลังแบบหลายคุณสมบัติพร้อมกัน

สามารถกำหนดโดยใช้ property ชื่อว่า background

```
background : white url(images/logo.jpg) no-repeat fixed center center;
```



## กำหนดตำแหน่งด้วย float

**float** กำหนดให้อิลิเมนต์สามารถลอยอยู่ด้านใดด้านหนึ่ง โดยค่าที่กำหนดได้ คือ

- left, right (ให้ลอยอยู่ทางด้านซ้ายและขวาตามลำดับ)
- inherit ลอยตาม parent element
- none ไม่ให้มีการลอย (default)

**clear** ไม่อนุญาตให้มีการลอยของอิลิเมนต์โดยค่าที่กำหนดได้คือ

- left, right (ไม่อนุญาตให้มีการลอยทางด้านซ้ายและขวาตามลำดับ)
- both (ไม่อนุญาตให้มีการลอยทั้งสองด้าน)

## กำหนด style ให้กับ Link

- `a : link` กำหนด style ให้กับ link ที่ยังไม่ถูกคลิก
- `a : hover` กำหนด style ให้กับ link เมื่อนำเมาส์ไปวาง
- `a : visited` กำหนด style ให้กับ link ที่ถูกคลิกแล้ว
- `a : active` กำหนด style ให้กับ link ขณะที่ถูกคลิก



# Display Inline , Block , Inline-Block

**display** คือการจัดรูปแบบการแสดงผลข้อมูล layout

- none ไม่มีการแสดงผล
- block แสดงผลแบบ block โดยการขึ้นบรรทัดใหม่ก่อน (เรียงในแนวตั้ง)
- inline แสดงผลแบบ inline โดยไม่มีการขึ้นบรรทัดใหม่ (เรียงในแนวนอน)
- inline-block แสดงผลแบบแนวนอนและขยายพื้นที่ด้านใน



# Visibility

visibility คือ การแสดงหรือซ่อนวัตถุโดยไม่กระทบ layout

- hidden ซ่อนวัตถุ
- visible แสดงวัตถุ



# การจัดตำแหน่งด้วย Position

## position ใช้ในการจัดตำแหน่งของวัตถุ

**static** - จัดวางแบบปกติเป็นค่า default

**relative** - ใช้ในการจัดวางและกำหนดตำแหน่งวัตถุโดยการนับจากจุดที่วัตถุนั้นๆอยู่

**absolute** - ใช้ในการจัดวางวัตถุ ให้ไปอยู่ตำแหน่งใดก็ได้แต่ต้องกำหนดตำแหน่งจากขอบของ element ที่บรรจุ วัตถุนั้นๆอีกที (ถ้าไม่มีอะไรครอบก็ถือว่า body เป็น element ที่ครอบวัตถุที่ระบุ absolute)

**fixed** - ใช้ในการจัดวางวัตถุให้อยู่ตำแหน่งเดิม

**sticky** - ให้วัตถุติดขอบเมื่อเลื่อนไปถึง



# ลำดับความสำคัญของ Style

- เรียงลำดับความสำคัญจากล่างขึ้นบน
- !important กำหนดให้มีความสำคัญที่สุด





# จำกัดการแสดงผลข้อมูลด้วย Overflow

## overflow

- visible แสดงข้อมูลทั้งหมด
- hidden ซ่อนข้อมูลที่เกิน
- scroll ให้มี scroll bar แสดงผลเมื่อมีข้อมูลเกิน
  - overflow:scroll
  - overflow-x : scroll
  - overflow-y : scroll
- auto ให้มี scroll bar แสดงผลออกมาอัตโนมัติ



# Cursor Style

auto	default	none	context-menu	help
pointer	progress	wait	cell	crosshair
text	vertical-text	alias	copy	move
no-drop	not-allowed	all-scroll	col-resize	row-resize
n-resize	s-resize	e-resize	w-resize	ns-resize

<https://i.stack.imgur.com/gJYJv.gif>



# กำหนดเงาให้วัตถุด้วย Box-shadow

**box-shadow : x y blur spread color**

- x คือ เงาแกน x (+ ขวา, - ซ้าย)
- y คือ เงาแกน y (+ ล่าง, - บน)
- blur คือ ขนาดความมัวของเงา
- spread คือ ขนาดของเงา
- color คือ สีของเงา (color name,rgb,...)



# กำหนดค่าความทึบ (Opacity)

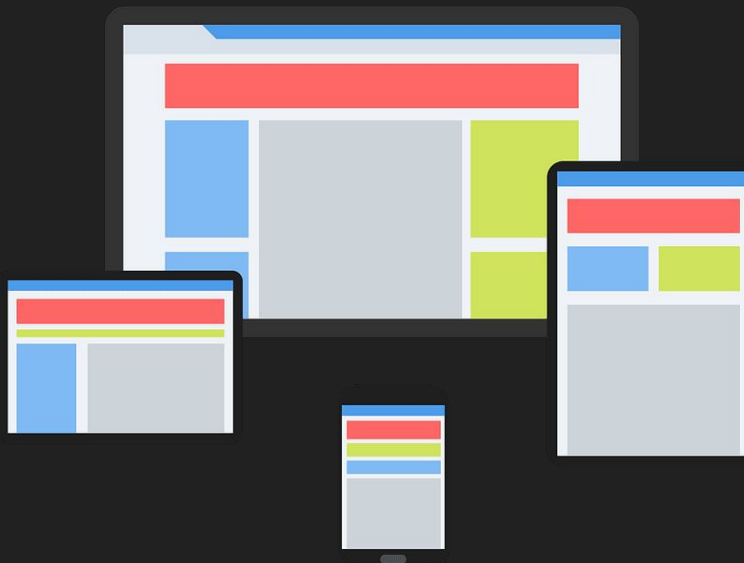
## opacity : value

- ค่าอยู่ระหว่าง 0.0 - 1.0
- ยิ่งค่าน้อย ยิ่งจาง
- ยิ่งค่ามาก ยิ่งทึบ



# Responsive Web Design

การออกแบบเว็บที่ตอบสนองและแสดงผลได้ดีบนอุปกรณ์ต่างๆหรือขนาดหน้าจอที่หลากหลาย เช่น จอคอมพิวเตอร์ หรือ สมาร์ทโฟน เป็นต้น



# Responsive Web Design (Media Query)

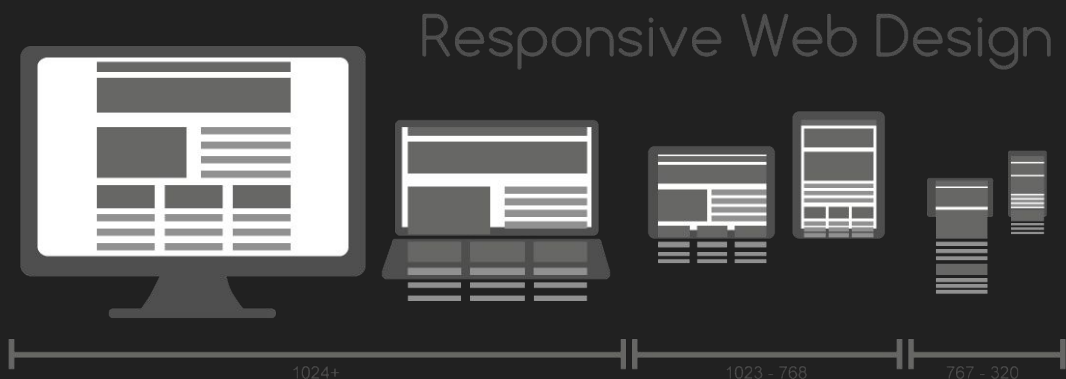
**Media Query** คือ รูปแบบการเขียน Style ให้แสดงผลตามขนาดหน้าจอที่แตกต่างกัน

```
@media ขนาดอุปกรณ์ {  
    ....style....
```

```
}
```

```
@media screen , printer {  
    ....style....
```

```
}
```



[https://www.youtube.com/channel/UCQ1r\\_4x-P-fETLIU4pqf98w](https://www.youtube.com/channel/UCQ1r_4x-P-fETLIU4pqf98w)



<https://www.facebook.com/KongRuksiamTutorial/>

# Responsive Web Design (Media Query)

## ความกว้างของขนาดอุปกรณ์

- 320px – 480px: Mobile devices
- 481px – 768px: iPads, Tablets
- 769px – 1024px: Laptops
- 1025px – 1200px: Desktops
- 1201px เป็นต้นไป – TV , Widescreen



# Viewport Units

ในอดีตเว็บถูกพัฒนาให้ทำงานบนจอคอมพิวเตอร์อย่างเดียว ถ้านำมาทำงานบน Smart Phone , Tablet ในปัจจุบันอาจจะไม่สามารถแสดงผลได้ตามที่ต้องการ เช่น ถ้าต้องการดูเนื้อหาในหน้าเว็บต้องทำการ zoom และ scroll หน้าจอเท่านั้น

ในปัจจุบันจึงได้มีการพัฒนาและเพิ่ม meta tag ชื่อว่า viewport เข้ามาในหน้าเว็บเพื่อบอกว่าให้เว็บแสดงผลบน Smart Phone , Tablet ได้ผ่านการอ้างอิงพื้นที่หรือสัดส่วนของ Web Browser ที่อยู่ในอุปกรณ์นั้นๆ ในการแสดงผลหน้าเว็บให้เหมาะสมกับอุปกรณ์ดังกล่าว





# องค์ประกอบของ Viewport

Viewport

(vh)

(vw)



[https://www.youtube.com/channel/UCQ1r\\_4x-P-fETLIU4pqf98w](https://www.youtube.com/channel/UCQ1r_4x-P-fETLIU4pqf98w)



<https://www.facebook.com/KongRuksiamTutorial/>

# Viewport Units

- **vw** - บอกสัดส่วนความกว้างของ viewport (%) เช่น เรามีจอกว้าง 100px ต้องการแสดงผล 10% ของความกว้างทั้งหมด จะมีค่า 10vw หรือพื้นที่ 10px หรือ 10/100 เป็นต้น ถ้าอยากกำหนดความกว้างของ viewport มีค่าเท่ากับ ความกว้างของ browser จะมีค่าเท่ากับ 100vw
- **vh** - บอกสัดส่วนความสูงของ viewport (%)
- **vmin** - ค่า % ต่ำสุดของ viewport
- **vmax** - ค่า % สูงสุดของ viewport



# รู้จักกับ Flexbox

Flexbox คือเครื่องมือใน CSS ที่ช่วยให้การจัดการ element ต่างๆในหน้าเว็บมีความง่ายและยืดหยุ่นมากยิ่งขึ้น โดยทั่วไปการจัดตำแหน่ง element ต่างๆต้องใช้ layout mode คือ block , inline , position และอื่นๆ

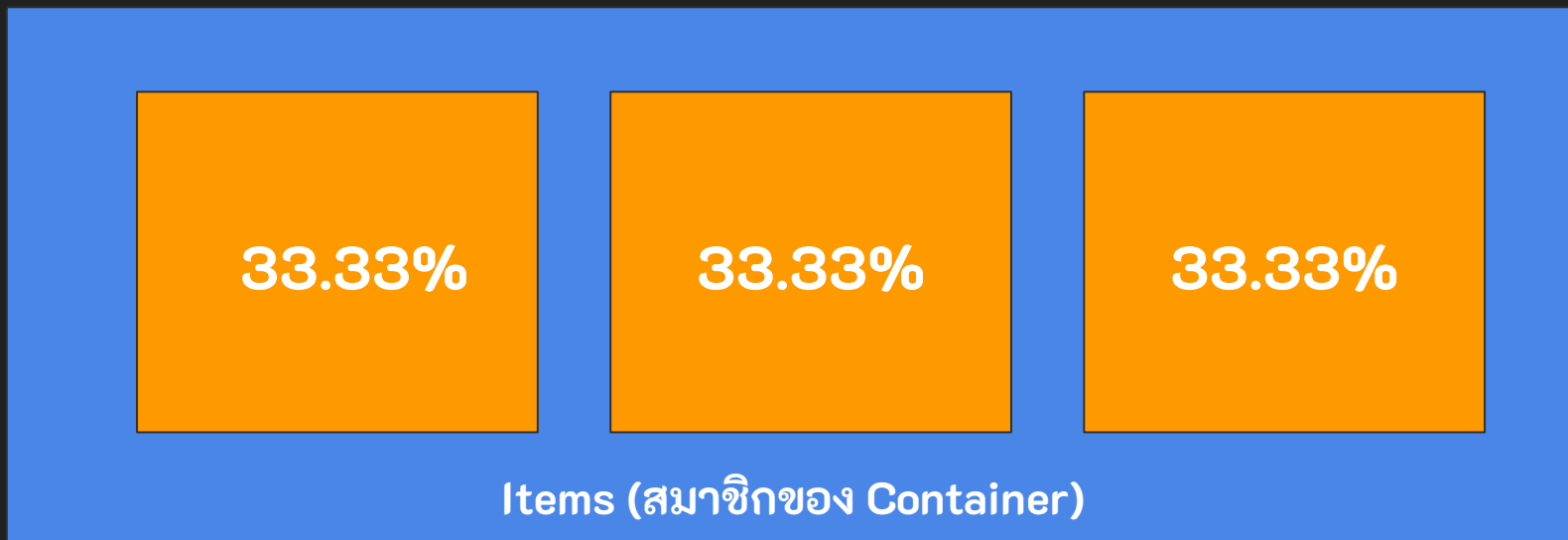
การพัฒนาเว็บในปัจจุบันมีความซับซ้อนมากยิ่งขึ้นทำให้การใช้ layout mode แบบเดิมไม่ตอบโจทย์เท่าที่ควร จึงได้มีการพัฒนา flexbox ขึ้นมาเพื่อจัดใช้ในการจัดการ element ให้มีความยืดหยุ่นสูง โดยมีคุณสมบัติดังนี้

- จัดเรียงตำแหน่งของ element ได้ง่ายขึ้น เรียงจากบนลงล่าง ซ้ายไปขวา อื่นๆ
- กำหนดขนาดให้พอดีกับพื้นที่ว่างแบบอัตโนมัติ (Sizing)



# องค์ประกอบของ Flexbox

Container (กล่องที่ครอบ Items)



# จัดเรียงได้หลากหลายแบบ

50%

50%

50%

50%

25%

25%

25%

25%

# Flexbox เบื้องต้น

```
.container{  
    display: flex;  
    box-sizing: border-box;  
}
```

- flex: ให้จัดวางในรูปแบบ flex
- border-box - กำหนดขนาดให้พอดีกับพื้นที่ว่างโดยคำนวณจากค่าจริงที่ผู้ใช้กำหนด (border + padding) เพื่อไม่ให้ item แสดงผลเพี้ยน (ระบุหรือไม่ระบุก็ได้)

# กำหนดทิศทางด้วย flex-direction

- row (ค่าเริ่มต้น) จัดวาง items ในแนวนอนทิศทางเดียวกับแกนหลัก
- column จัดวาง items ในแนวตั้งทิศทางเดียวกับแกนหลัก
- row-reverse จัดวาง items ในแนวนอนทิศทางตรงข้ามกับแกนหลัก
- column-reverse จัดวาง items ในแนวตั้งทิศทางตรงข้ามกับแกนหลัก



# กำหนดขนาดด้วย flex-wrap

## กรณีที่ขนาด items ใหญ่กว่าพื้นที่ container

- nowrap จัดวาง items ที่เกินพื้นที่ container ไปด้านขวามือ
- wrap จัดวาง items ที่เกินพื้นที่ container เรียงจากบนลงล่าง
- wrap-reverse จัดวาง items ที่เกินพื้นที่ container เรียงจากล่างขึ้นบน





# Flex – properties (Items)

- flex-shrink : ให้ item หดตัวจำนวนเท่าใดเมื่อเทียบกับ item อื่นๆ  
ค่าเริ่มต้นเป็น 1
- flex-grow : ให้ item ขยายจำนวนเท่าใดเมื่อเทียบกับ item อื่นๆ  
ค่าเริ่มต้นเป็น 0
- flex-basis : กำหนดค่าความยืดหยุ่นเริ่มต้น
- flex:1 ทำให้ item ที่อยู่แถวเดียวกันมีขนาดเท่ากัน



# Flex Justify (จัดวางตำแหน่ง item)

**\*\*เทียบกับแกนหลัก เช่น กำหนดแกนหลักเป็นแนวนอน**

## justify-content

- flex-start ซิดซ้าย container ทิศทางตามแนวนอน
- center กึ่งกลาง container ทิศทางตามแนวนอน
- flex-end ซิดขวา container ทิศทางตามแนวนอน
- space-around ระยะห่างซ้ายขวาและขนาด item เท่ากัน
- space-between ระยะห่างซ้ายขวาและขนาด item เท่ากัน (ติดมุม)

# Flex Alignment (จัดวางตำแหน่ง item)

**\*\*เทียบกับแกนตรงข้าม เช่น กำหนดแกนหลักเป็นแนวนอน**

**align-items (item ทุกตัว) และ align-self (Item ที่ต้องการ)**

- stretch - กำหนดขนาด item เท่ากับขนาด container
- flex-start ด้านบน container ทิศทางตามแนวนอน
- center กึ่งกลาง container ทิศทางตามแนวนอน
- flex-end ด้านล่าง container ทิศทางตามแนวนอน



# CSS Grid Layout

Flexbox ถูกออกแบบให้จัดการ layout แบบทิศทางเดียว คือ 1 มิติ เช่น เรียงลำดับจากบนลงล่าง ซ้ายไปขวา เป็นต้น



แต่ Grid Layout ถูกออกแบบมาเพื่อจัดการ layout แบบ 2 มิติ คือ มีทั้งแบบแนวนอนและแนวตั้งในเวลาเดียวกัน หรือมองง่าย ๆ ก็เป็นแบบตาราง



# CSS Grid Layout

## การใช้งาน

`display : grid;`

## กำหนดขนาดแถว (ความสูง) :

`grid-template-rows : ความสูงของแถวที่ 1 , 2 , 3;`

## กำหนดขนาดคอลัมน์ (ความกว้าง) :

`grid-template-columns : ความกว้างของคอลัมน์ที่ 1 , 2 , 3;`



# กำหนดคอลัมน์เริ่มต้น - สิ้นสุด

1	2	3	4



# กำหนดแถวเริ่มต้น - สิ้นสุด

1				
2				
3				
4				

## Gird Properties อื่นๆ

- กำหนดสัดส่วนพื้นที่ด้วย span
- กำหนดสัดส่วนพื้นที่ด้วยหน่วย fr
- กำหนดระยะห่างพื้นที่ด้วย gap
- กำหนดชื่อพื้นที่ด้วย grid-template-area





# Advance Selectors

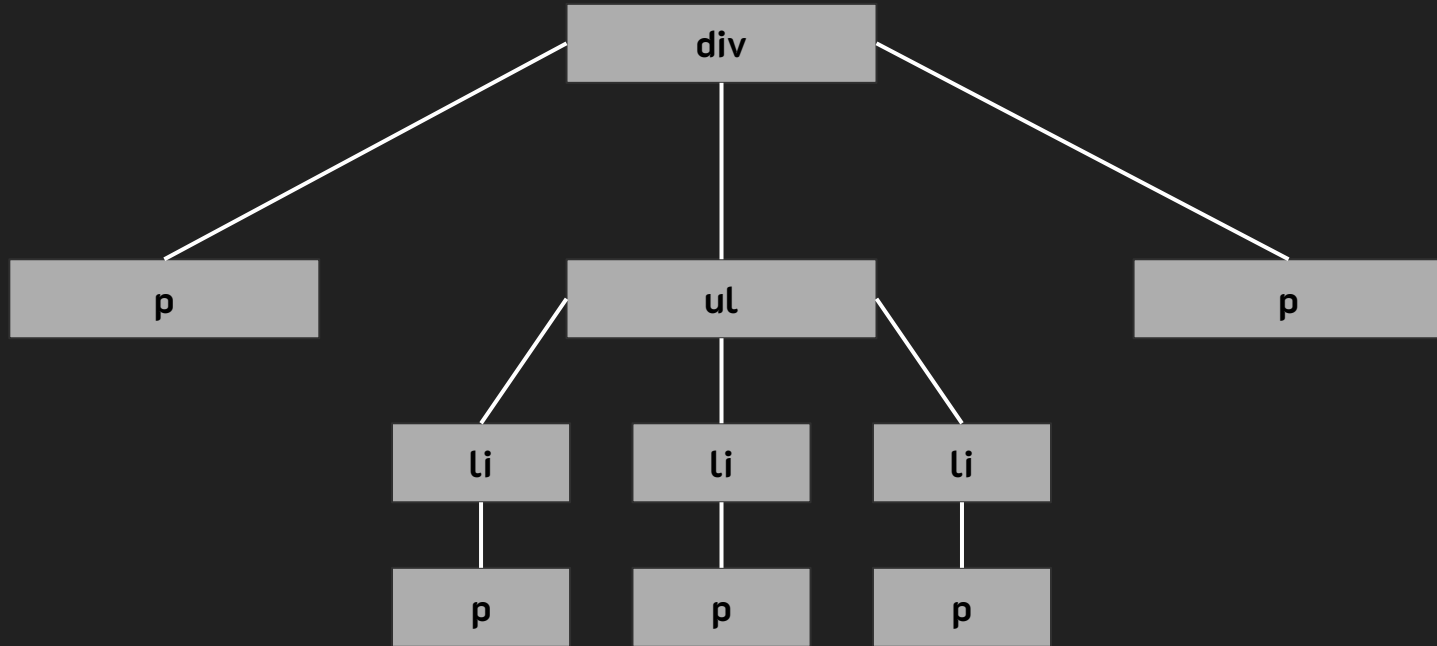


# Advance Selectors

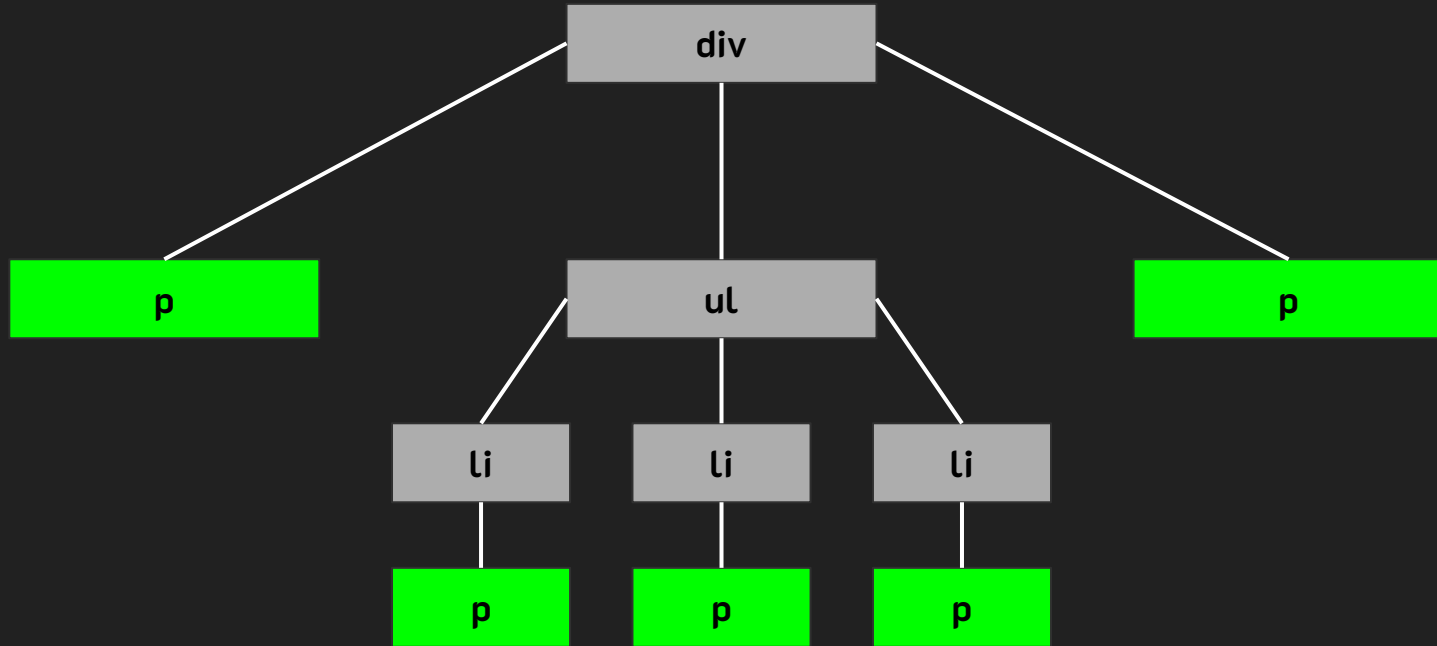
- Targeted Selectors
  - Descendant Selector
  - Child Selector
- Style By Attribute
- Special Attribute
- Pseudo Selectors
  - first-child
  - last-child
  - nth-child
- Before & After Pseudo



# Targeted Selectors

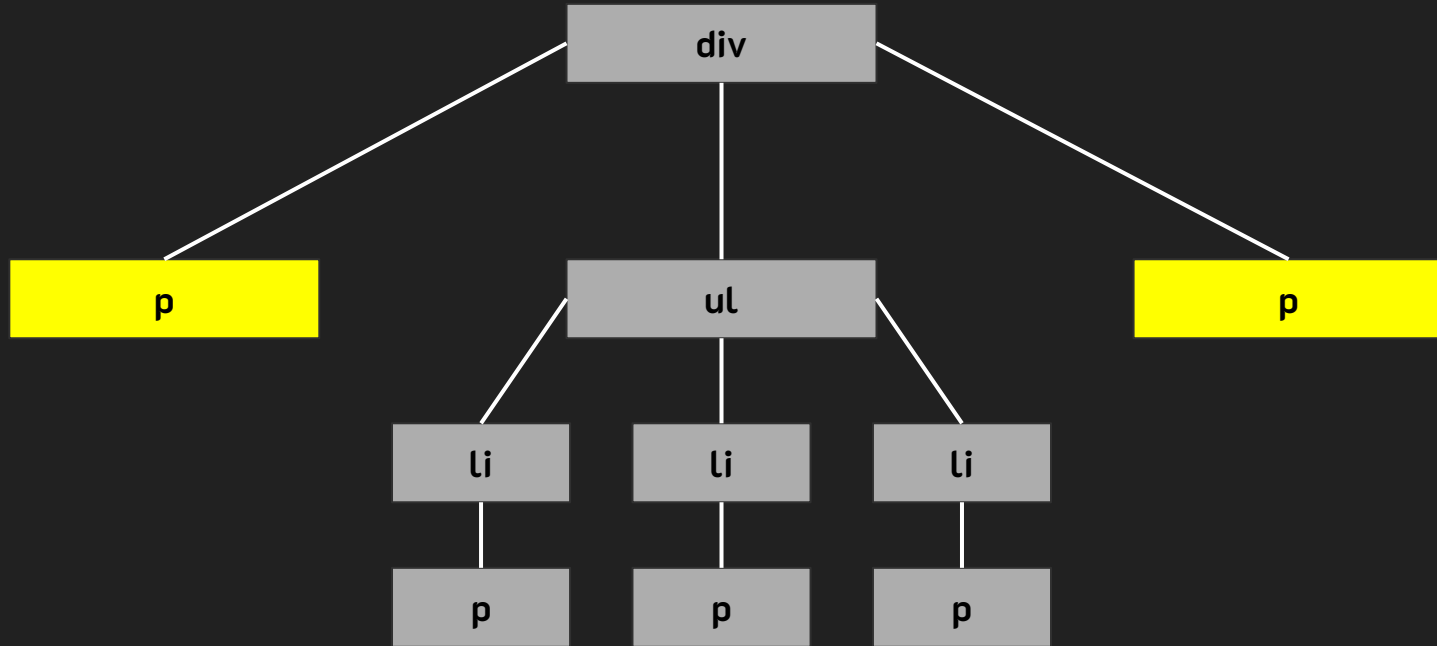


# Targeted Selectors



Descendant Selector

# Targeted Selectors



Child Selector

# กำหนดเงาให้ข้อความด้วย text-shadow

## text-shadow : x y blur color

- x คือ เงาแกน x (+ ขวา, - ซ้าย)
- y คือ เงาแกน y (+ ล่าง, - บน)
- blur คือ ขนาดความมัวของเงา
- color คือ สีของเงา (color name,rgb,...)



# CSS Variable

การสร้างตัวแปรใน css เพื่อช่วยอำนวยความสะดวกในการกำหนด style ให้แต่ละ element โดยลดขั้นตอนการทำงานที่ซ้ำซ้อนใน css ให้ได้มีความเป็นระเบียบและง่ายต่อการจัดการมากยิ่งขึ้น

การกำหนดตัวแปรสามารถกำหนดชื่อได้เอง (custom variable) โดยขึ้นต้นด้วยเครื่องหมาย - ตามด้วยชื่อตัวแปร ซึ่งตัวแปรส่วนใหญ่ใน css จะนำมาเก็บค่าที่ใช้เรียกทำงานซ้ำๆ เช่น สี เงา แอนิเมชัน เป็นต้น



# จัดการ Element ด้วย Transform

- `translate (x,y)` กำหนดตำแหน่ง element
- `scale(x,y)` ขยาย element
- `rotate (มุม deg)` กำหนดการหมุนของ element
- `skewX(มุม deg)` กำหนดการเอียงของ element แกน x
- `skewY (มุม deg)` กำหนดการเอียงของ element แกน y





# เปลี่ยนแปลง Element ด้วย Transition

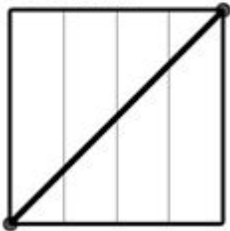
**transition** คือการเปลี่ยนค่าใน element จากค่าหนึ่งไปสู่อีกค่าหนึ่งในช่วงเวลาที่กำหนดประกอบด้วย

- transition-properties คือ รูปแบบคุณสมบัติที่การเปลี่ยนแปลงค่า
- transition-duration คือ ระยะเวลาในการเปลี่ยนแปลงค่า
- transition-timing-function คือ รูปแบบฟังก์ชันของการเปลี่ยนแปลงค่า (ease-in)
- transition-delay คือ ระยะเวลาที่จะเริ่มต้นเปลี่ยนแปลงค่า



# transition-timing-function

linear



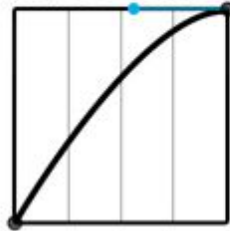
ease



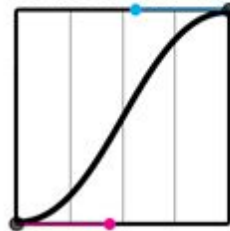
ease-in



ease-out



ease-in-out



# CSS Animation

- **animation-name** คือ ชื่อ animation
- **animation-duration** คือ ระยะเวลาของ animation จากเริ่มต้นไปสิ้นสุด
- **animation-timing-function** คือ รูปแบบการเล่น animation
- **animation-iteration-count** คือ จำนวนการเล่น animation (infinite คือ ไม่สิ้นสุด)
- **animation-direction** คือ ทิศทางการเล่น (เล่นจาก frame 1 ไป frame 10 หรือ แบบ ย้อนกลับก็ได้)
- **animation-delay** คือ ระยะเวลาที่จะเริ่มต้น

