



What is CSS ★



CSS หรือ กีชื่อเต็มๆ คือ "**Cascading Style Sheets**"

เป็นภาษาที่ใช้ผ่านมาลักษณะรูปแบบ ไล่นี้หลัง
หรือเพิ่มกรอบข้อความของหน้าเว็บ เพื่อเพิ่มความสวยงาม
ให้หน้าเว็บของเราระดับ CSS สามารถกำหนดรูปแบบร่วมกัน
กีเดียวได้ทำให้เวลาแก้ไขไม่ต้องค่อยแก้กีละส่วน



Example

```
● ● ●  
1 body {  
2   background-color: antiquewhite;  
3 }  
4  
5 h1 {  
6   color: blue;  
7   text-align: center;  
8 }  
9  
10 p {  
11   font-family: verdana;  
12   font-size: 20px;  
13 }
```

CSS Syntax

Selector

h1

Declaration

{ color:blue; font-size:12px; }

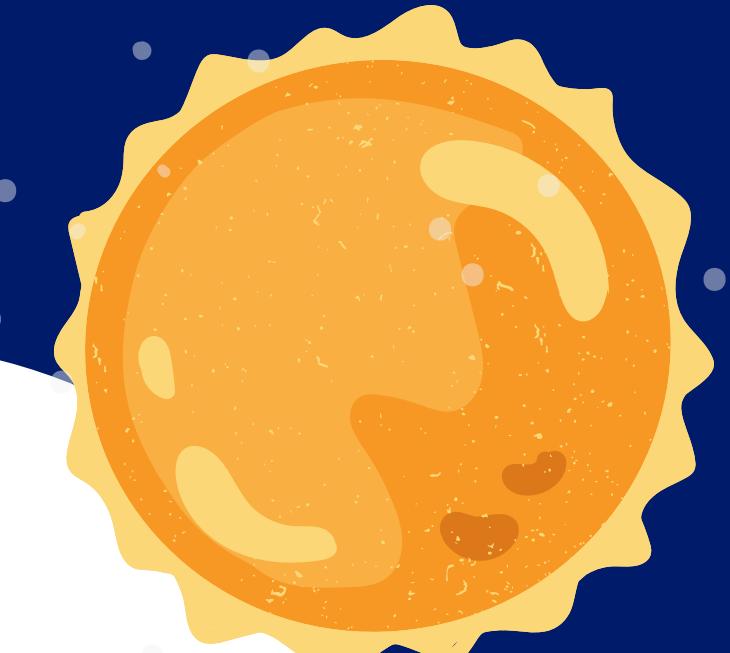
Property

Value

Declaration

Property

Value



Internal vs External vs Inline

```
● ● ●  
1  <!DOCTYPE html>  
2  <html lang="en">  
3  <head>  
4    <meta charset="UTF-8">  
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
6    <title>This is title</title>  
7  </head>  
8  <body>  
9    <h2 style="color: green;  
10      font-size: 18px;">  
11        Welcome To GFG  
12    </h2>  
13    <p style="color: red;  
14      font-size: 14px;">  
15      This is some text. style by inline CSS  
16    </p>  
17  </body>  
18  </html>
```

Inline

Internal vs External vs Inline

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5    <meta charset="UTF-8">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <title>This is title</title>
8
9  <style>
10   h1 {
11     color: blue;
12     font-size: 24px;
13     font-weight: bold;
14   }
15
16   p {
17     color: green;
18     font-size: 16px;
19   }
20 </style>
21 </head>
22
23 <body>
24   <h1>Hello World</h1>
25   <p>Lorem ipsum dolor
26   </p>
27 </body>
28 </html>
```

Internal

Internal vs External vs Inline

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>This is title</title>
8
9   <link rel="stylesheet" href="style.css">
10 </head>
11
12 <body>
13   <h1>Hello World</h1>
14   <p>Lorem ipsum dolor
15   </p>
16 </body>
17
18 </html>
```

```
1 body {
2   background-color: antiquewhite;
3 }
4
5 h1 {
6   color: blue;
7   text-align: center;
8 }
9
10 p {
11   font-family: verdana;
12   font-size: 20px;
13 }
```

External

CSS Selectors

ใช้สำหรับ tag ที่เราต้องการจะใส่ Style



Selector	Example	description
#id	#firstname	เลือกเฉพาะ tag ที่มี id = "firstname"
.class	.intro	เลือกเฉพาะ tag ที่มี class = "intro"
element.class	p.intro	เลือกเฉพาะ tag <p> ที่มี class = "intro"
*	*	เลือกทุก tag
element	p	เลือก tag <p> ทุกอัน
element,element,...	div,p	เลือกเฉพาะ tag <div> กับ <p> ทุกอัน



CSS Selectors



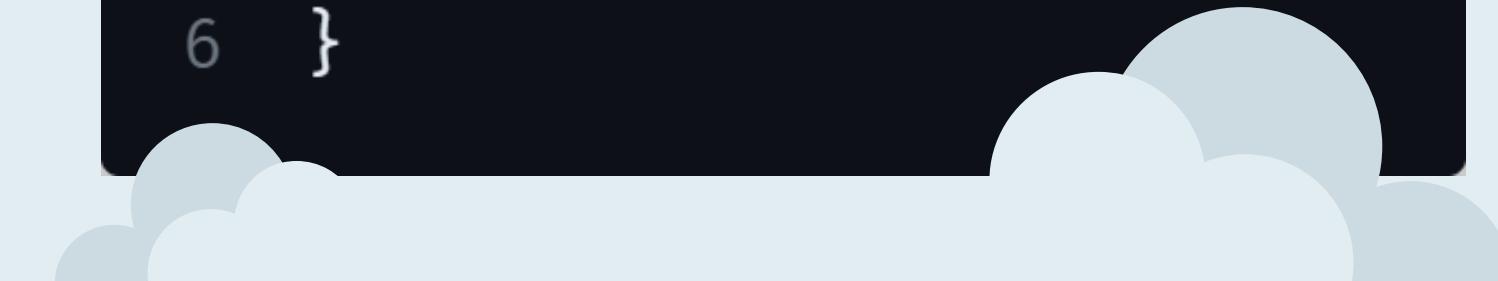
```
1 #firstname {  
2   width: 100%;  
3   padding: 12px 20px;  
4   margin: 8px 0;  
5   box-sizing: border-box;  
6 }  
7
```



#id



```
1 .intro {  
2   width: 100%;  
3   padding: 12px 20px;  
4   margin: 8px 0;  
5   box-sizing: border-box;  
6 }
```

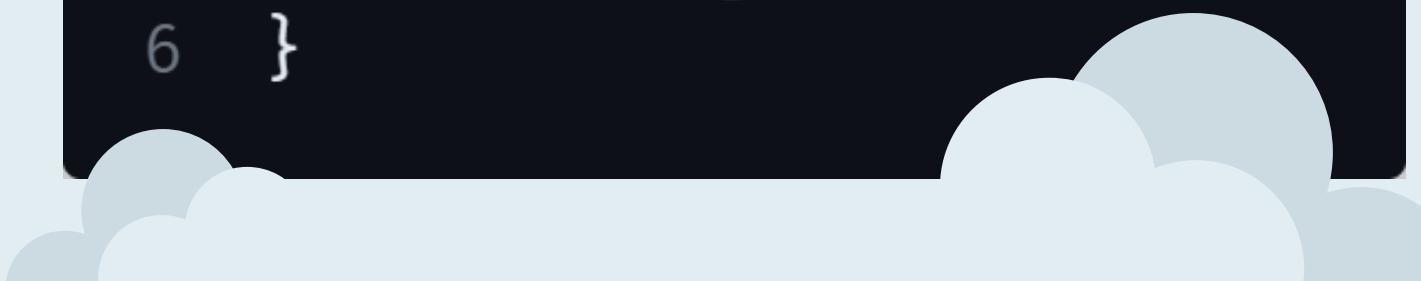


.class

CSS Selectors



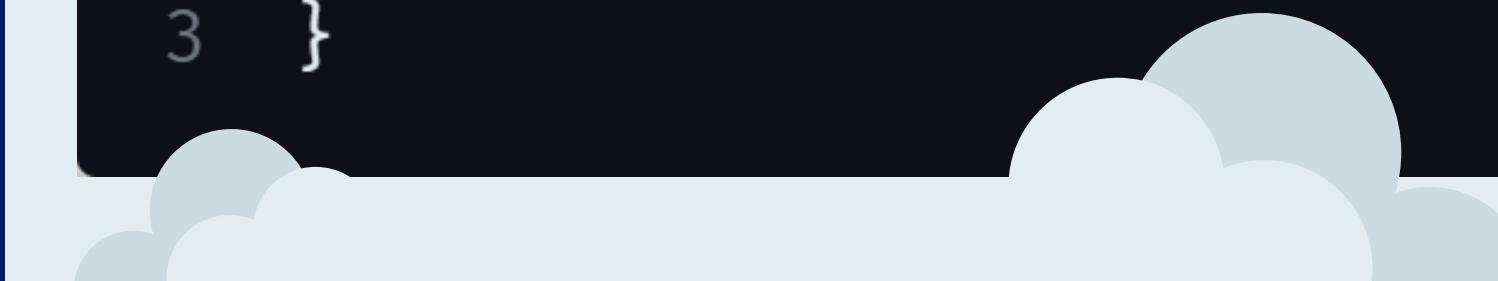
```
1 p.intro {  
2   width: 100%;  
3   padding: 12px 20px;  
4   margin: 8px 0;  
5   box-sizing: border-box;  
6 }
```



element.class



```
1 * {  
2   box-sizing: border-box;  
3 }
```



*

CSS Selectors



```
1 p {  
2   font-family: "Times New Roman", Times, serif;  
3   font-size: 20px;  
4 }
```

element



```
1 div,  
2 p {  
3   border: 1px solid black;  
4   background-color: lightgrey;  
5   padding: 20px;  
6 }
```

element, element, ...

Colors.

background-color - ໃຫ້ສໍາหารັບເປີດຕິບນີ້ background

color - ໃຫ້ສໍາหารັບເປີດຕິບນີ້ຕ້າວັກຈຽງ

```
● ● ●  
1 <!DOCTYPE html>  
2 <html lang="en">  
3  
4 <head>  
5   <meta charset="UTF-8">  
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">  
7   <title>This is title</title>  
8  
9   <style>  
10    p {  
11      color: red;  
12    }  
13  
14    * {  
15      background-color: #9aecf1;  
16    }  
17   </style>  
18 </head>  
19  
20 <body>  
21   <p>Lorem ipsum dolor  
22   </p>  
23 </body>  
24  
25 </html>
```

Lorem ipsum dolor

Units & Sizes

การกำหนดขนาดใน css จะมีหน่วยในที่ใช้บ่อย คือ px, em, rem

- **px (pixel)** - เป็นการกำหนดขนาดแบบตายตัว คือกำหนดไว้เท่าไรคือเท่านั้นเลย
- **em** - เป็น relative ของ em นั้น จะขึ้นอยู่กับ parent element ที่อยู่ใกล้กันสุด
- **rem** - คล้าย ๆ กับ em คือเป็น relative จะมีจุดต่างกันก็คือ rem (root em) ค่าของมันไม่ได้ขึ้นอยู่กับ element parent แต่จะขึ้นอยู่กับขนาดของ root element



Units & Sizes [px]

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5    <meta charset="UTF-8">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <title>This is title</title>
8
9  <style>
10   p {
11     font-size: 32px;
12   }
13  </style>
14 </head>
15
16 <body>
17   <p>P Tag</p>
18 </body>
19
20 </html>
```

P Tag

tag p จะมีขนาดเป็น 32 px

Units & Sizes [em]

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5    <meta charset="UTF-8">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <title>This is title</title>
8
9  <style>
10   .container {
11     font-size: 16px;
12   }
13
14   .container p {
15     font-size: 1em;
16   }
17
18   .container h2 {
19     font-size: 3em;
20   }
21
22   .container h3 {
23     font-size: 2em;
24   }
25 </style>
26 </head>
27
28 <body>
29   <div class="container">
30     <p>P Tag</p>
31     <h2>H2 Tag</h2>
32     <h3>H3 Tag</h3>
33   </div>
34 </body>
35
36 </html>
```

P Tag

H2 Tag

H3 Tag

จะเห็นว่าที่ parent element (container) จะกำหนด font-size เป็น 16px และ p h2 h3 อยู่ใน container เมื่อมีการใช้หน่วยเป็น em จะคำนวณขนาดของ p มีค่าเป็น 16px ($1\text{em} * 16\text{px}$) h2 เป็น 48px ($3\text{em} * 16\text{px}$) และ h3 เป็น 32px ($2\text{em} * 16\text{px}$)

Units & Sizes [rem]

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5    <meta charset="UTF-8">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <title>This is title</title>
8
9  <style>
10   p {
11     font-size: 1rem;
12   }
13
14   h2 {
15     font-size: 2rem;
16   }
17  </style>
18 </head>
19
20 <body>
21   <p>P Tag</p>
22   <h2>H2 Tag</h2>
23 </body>
24
25 </html>
```

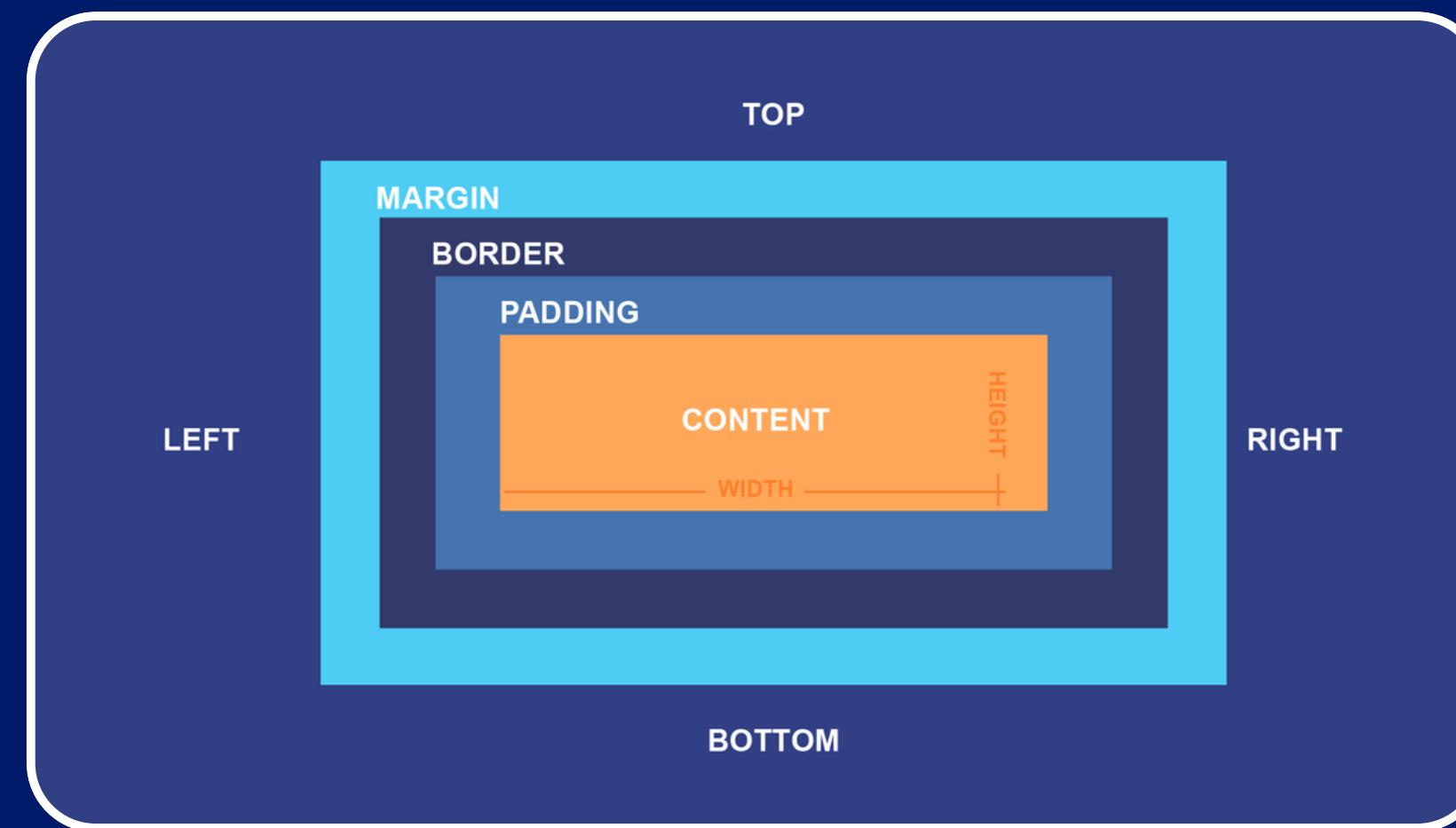
P Tag

H2 Tag

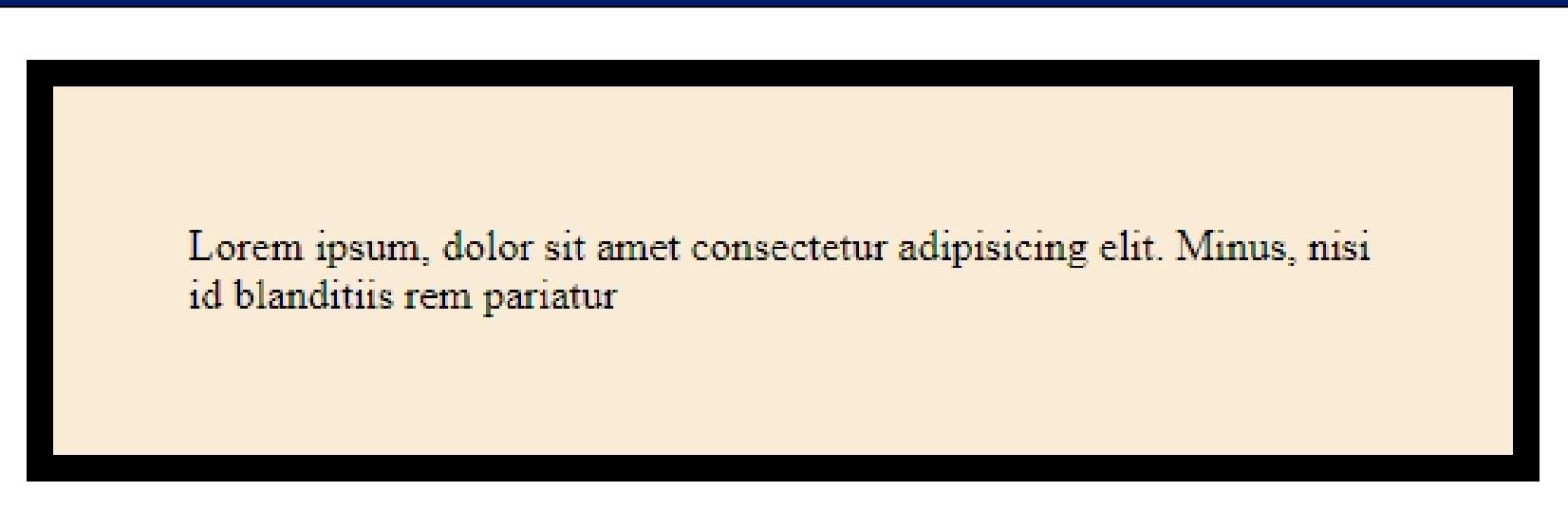
โดยปกติที่ root parent ค่า default ของ font-size
เป็น 16px โดย p มีขนาดเป็น 16px ($1\text{rem} * 16\text{px}$) และ
h2 เป็น 32px ($2\text{rem} * 16\text{px}$)

Box Model

- **Content** – นิ้นกีด้านในสุด ชี้่งเราสามารถกำหนดความกว้างและความสูงได้โดยใช้ width หรือ height
- **Padding** – ระยะห่างระหว่าง content และ border โดยสามารถกำหนดขนาดได้โดยใช้ padding
- **Border** – นิ้นกีคือลุ่มหรือบู๊ดของ content และ padding โดยอาจจะเรียกว่า เป็น “ขอบ” ของตัว element นั้นๆ ชี้่งสามารถกำหนดความหนา ความโค้งมนของขอบได้ โดยสามารถกำหนดขนาดได้โดยใช้ border
- **Margin** – นิ้นกีด้านนอกสุดและเป็นตัวกำหนดว่า element อันนี้ควรมีจะมีระยะห่างหรือติดกับ element อื่นๆ เป็นระยะเท่าไหร่ โดยสามารถกำหนดขนาดได้โดย margin



Box Model



```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Sticky Example</title>
8   <style>
9     p {
10       background-color: antiquewhite;
11       border: 10px solid black;
12       padding: 50px;
13       margin: 20px;
14     }
15   </style>
16 </head>
17
18 <body>
19   <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Minus, nisi id blanditiis rem pariatur</p>
20 </body>
21
22 </html>
23
```



Typography

font-family - กำหนดชนิดของ Font

font-weight - กำหนดความหนาของตัวอักษร

font-size - กำหนดขนาดของตัวอักษร

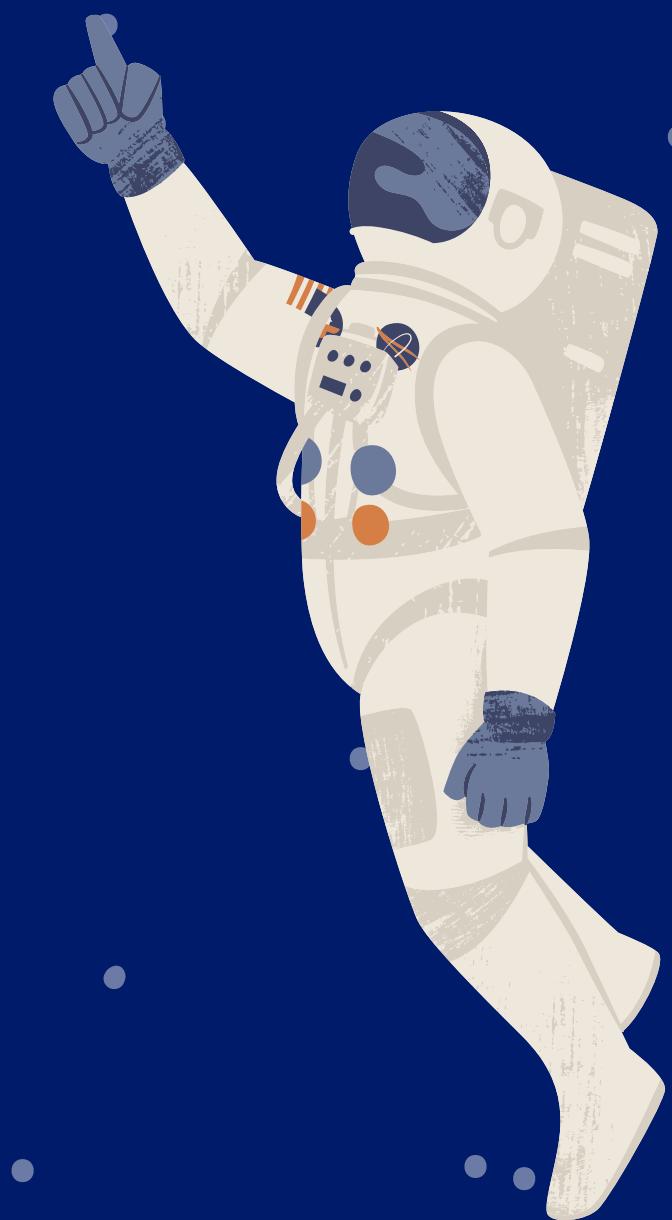
```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Sticky Example</title>
8   <style>
9     p {
10       font-family: monospace;
11       font-weight: 700;
12       font-size: 2rem;
13     }
14   </style>
15 </head>
16
17 <body>
18   <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Minus, nisi id blanditiis rem pariatur</p>
19 </body>
20
21 <html>
22
```

Lorem ipsum, dolor sit amet
consectetur adipisicing elit.
Minus, nisi id blanditiis rem
pariatur

Display

property ใน CSS กี่ช่วยในการแสดงผลของ CSS
ออกแบบในลักษณะต่างๆ

- "**inline**"
- "**block**"
- "**inline-block**"
- "**none**"



Display

inline คือ การแสดงผลของ Element ตัวนั้นให้วางตัวในแบบแนวอน ถ้าเรียกใช้ inline ข้อความหรือกล่องต่างๆ จะมีขนาดเป็นบล็อกทั้งหมดเดียวกันนั่นเอง โดยส่วนใหญ่ใช้กับข้อความเท่านั้น เนื่องจากไม่สามารถกำหนดค่า width และ height ได้

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>This is title</title>
8
9   <style>
10     p {
11       display: inline;
12     }
13   </style>
14 </head>
15
16 <body>
17   <p>This</p>
18   <p>in</p>
19   <p>line</p>
20 </body>
21
22 </html>
```

ก้อนใช้ inline

This
in
line

หลังใช้ inline

This in line

Display

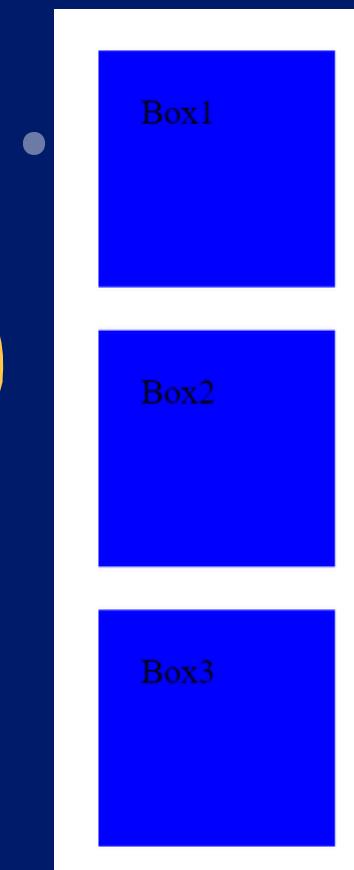
block គឺ ការរៀបចំលក្ខណៈរបស់ Element ទាំងបន្ទាត់ ជាមុន inline ដើម្បីថា block ត្រូវបានពិនិត្យដោយខ្លួន តាមរយៈការផ្តល់ព័ត៌មាន និងការរៀបចំការងារ ដែលត្រូវបានរៀបចំឡើង។

```
● ● ●  
1 <!DOCTYPE html>  
2 <html lang="en">  
3  
4 <head>  
5   <meta charset="UTF-8">  
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">  
7   <title>This is title</title>  
8  
9 <style>  
10   span {  
11     width: 70px;  
12     height: 70px;  
13     background-color: blue;  
14     padding: 20px;  
15     margin: 20px;  
16     display: block;  
17   }  
18 </style>  
19 </head>  
20  
21 <body>  
22   <span class="box1">Box1</span>  
23   <span class="box2">Box2</span>  
24   <span class="box3">Box3</span>  
25 </body>  
26  
27 </html>
```



កំណត់ថា បានប្រើប្រាស់ block

អតិថិជន ថា បានប្រើប្រាស់ block

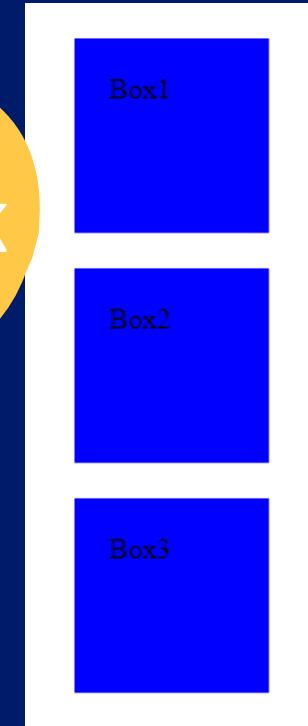


Display

inline-block คือ การจัดลักษณะของ Element ในรูปแบบรวม inline กับ block เข้าด้วยกัน โดยมีเงื่อนไขที่ว่าเป็นจัดลักษณะแบบ inline และสามารถกำหนด width กับ height ได้

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>This is title</title>
8
9 <style>
10  div {
11    width: 70px;
12    height: 70px;
13    background-color: blue;
14    padding: 20px;
15    margin: 20px;
16    display: inline-block;
17  }
18 </style>
19 </head>
20
21 <body>
22   <div class="box1">Box1</div>
23   <div class="box2">Box2</div>
24   <div class="box3">Box3</div>
25 </body>
26
27 </html>
```

ค่อนใช้ **inline-block**



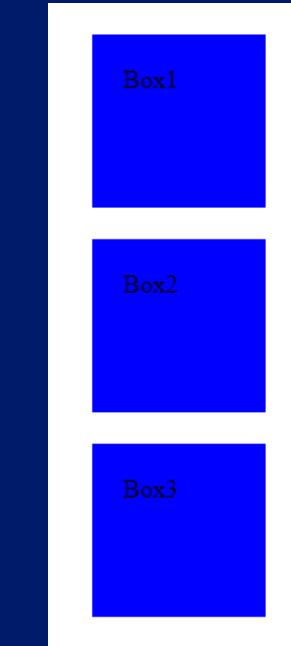
หลังใช้ **inline-block**



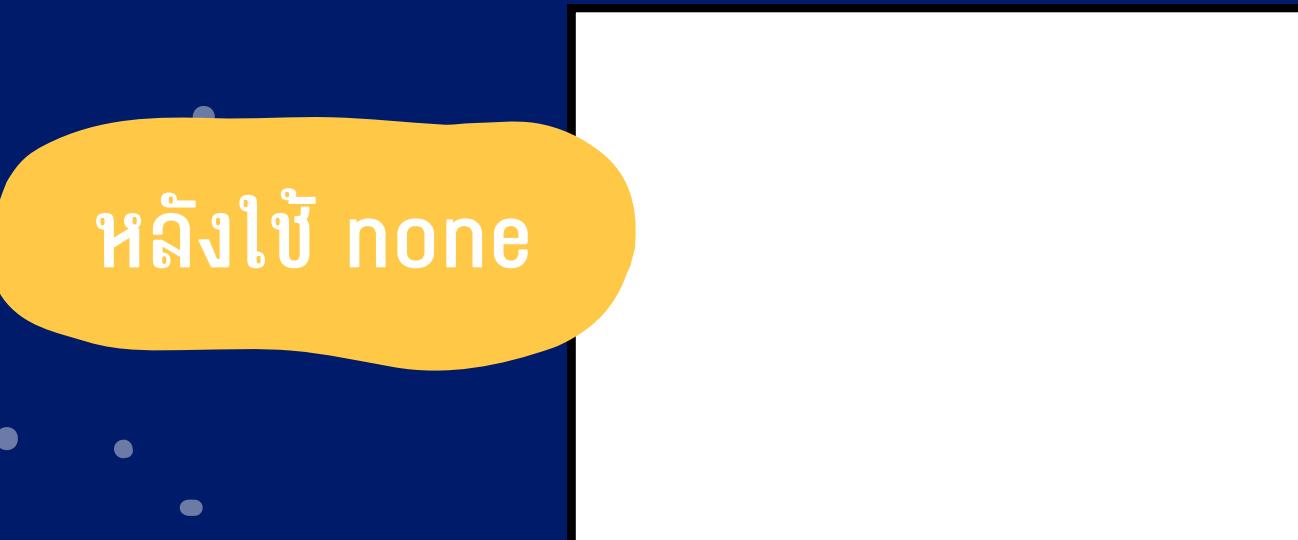
Display

none คือ การปิดการแสดงผลของ css ใน element นั้นออกไป

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Inline Block</title>
8   <style>
9     div {
10       display: none;
11       width: 70px;
12       height: 70px;
13       background-color: blue;
14       margin: 20px;
15       padding: 20px;
16     }
17   </style>
18 </head>
19
20 <body>
21   <div class="box1">Box1</div>
22   <div class="box2">Box2</div>
23   <div class="box3">Box3</div>
24 </body>
25
26 </html>
```



ก่อนใช้ none



หลังใช้ none

Position

ใช้ในการระบุตำแหน่งของ div ของ element

- static
- relative
- fixed
- absolute
- sticky



Position

Static - element ใน html จะมี default เป็น Static กำหนดค่า top bottom left right ไม่ส่งผลอะไร

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>This is title</title>
8
9  <style>
10     .static {
11         position: static;
12         border: 3px solid #73AD21;
13     }
14  </style>
15 </head>
16
17 <body>
18     <div class="static">
19         This div element has position: static;
20     </div>
21 </body>
22
23 </html>
```

Output

This div element has position: static;

Position

relative - element จะมีการวางแบบปกติ แต่การกำหนดค่า top bottom left right จะทำให้มีการวางตำแหน่งเปลี่ยนไปจากเดิม

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>This is title</title>
8
9 <style>
10   .relative {
11     position: relative;
12     left: 30px;
13     border: 3px solid #73AD21;
14   }
15 </style>
16 </head>
17
18 <body>
19   <div class="relative">
20     This div element has position: relative;
21   </div>
22 </body>
23
24 </html>
```

This div element has position: relative;

จะเห็นว่า element จะมีการเลื่อนจากทางซ้ายไป 30 px



Position

fixed - element จะอยู่ที่ตำแหน่งเดิมเสมอ แม้ว่าจะมีการเลื่อน page

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>This is title</title>
8
9   <style>
10     .fixed {
11       position: fixed;
12       bottom: 0;
13       right: 0;
14       border: 3px solid #73AD21;
15     }
16   </style>
17 </head>
18
19 <body>
20   <div class="fixed">
21     This div element has position: fixed;
22   </div>
23 </body>
24
25 </html>
```

This div element has position: fixed;



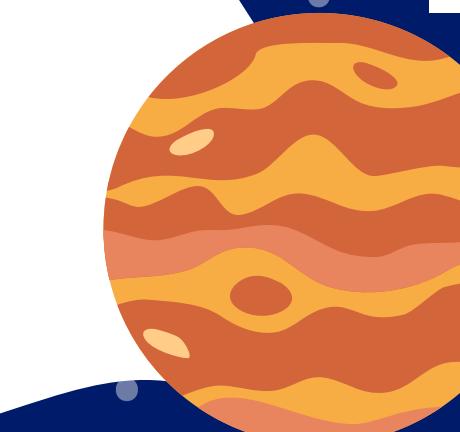
Position

absolute - จะมีการวางตำแหน่ง element โดยยึด Position ว่างอิงจาก parent element

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>This is title</title>
8
9 <style>
10   .relative {
11     position: relative;
12     width: 400px;
13     height: 200px;
14     border: 3px solid blue;
15   }
16
17   .absolute {
18     position: absolute;
19     top: 80px;
20     right: 0;
21     width: 200px;
22     height: 100px;
23     border: 3px solid red;
24   }
25 </style>
26 </head>
27
28 <body>
29   <div class="relative">
30     This div element has position: relative;
31     <div class="absolute">
32       This div element has position: absolute;
33     </div>
34   </div>
35 </body>
36
37 </html>
```

This div element has position: relative;

This div element has position:
absolute;



Position



sticky - เป็นการกำหนดตำแหน่งการวางของ element กี่ISM กันระหว่าง relative และ fixed โดยตอนแรกจะเป็น relative แต่พอ scroll ลง จะทำให้เปลี่ยนจาก relative เป็น fixed

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Sticky Example</title>
8   <style>
9     body {
10       height: 1500px;
11       font-family: Arial, sans-serif;
12       margin: 0;
13     }
14
15   .sticky {
16     position: sticky;
17     top: 0;
18     background-color: #cae8ca;
19     padding: 10px 0;
20     text-align: center;
21     font-weight: bold;
22     z-index: 100;
23   }
24   </style>
25 </head>
26
27 <body>
28   <div>
29     <h1>Hello Sticky</h1>
30   </div>
31   <div class="sticky">
32     <p>This is a sticky element</p>
33   </div>
34
35   <div style="padding: 20px; margin-top: 30px;">
36     <h2>Content Section</h2>
37     <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
38   </div>
39 </body>
40
41 </html>
```

ก่อน scroll

Hello Sticky

This is a sticky element

Content Section

Lore ipsum dolor sit amet, consectetur adipiscing elit.

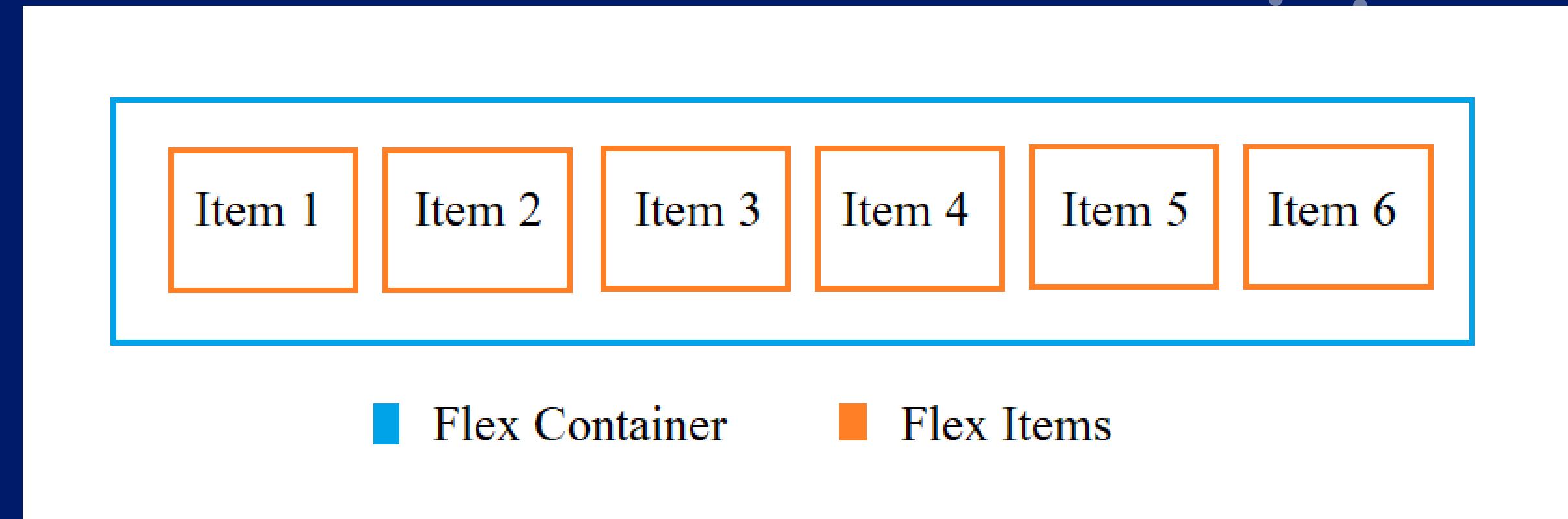
หลัง scroll

This is a sticky element

Flex Layout

Flex Layout จะมีส่วนประกอบด้วย 2 ส่วน ก็คือ **Flex Container** กับ **Flex Item**

- **Flex container** คือ html element ที่กำหนด property “display” ให้มีค่าเป็น “flex” หรือ “inline-flex”
- **Flex item** หมายถึง html elements ใดๆ ก็ตาม ที่เป็น child ของ flex container



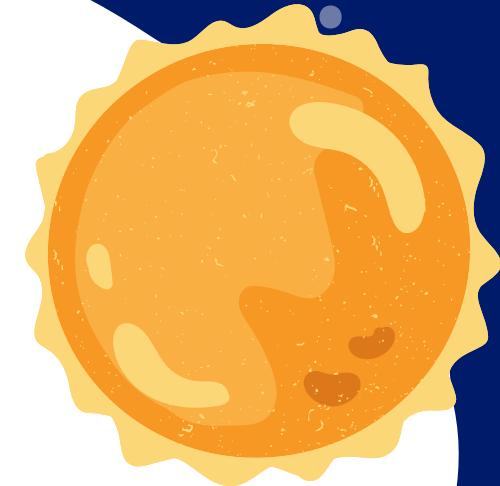
Flex Layout

flex-direction : เอกไว้สำหรับกำหนดการจัดเรียง Flex Item ภายใน Flex Container

- **row** : เรียงเป็นแนวแนวนอน จากขวาไปซ้าย
- **row-reverse** : เรียงเป็นแนวแนวนอน จากซ้ายไปขวา
- **column** : เรียงเป็นแนวตั้ง จากบนลงล่าง
- **column-reverse** : เรียงเป็นแนวตั้ง จากล่างขึ้นบน

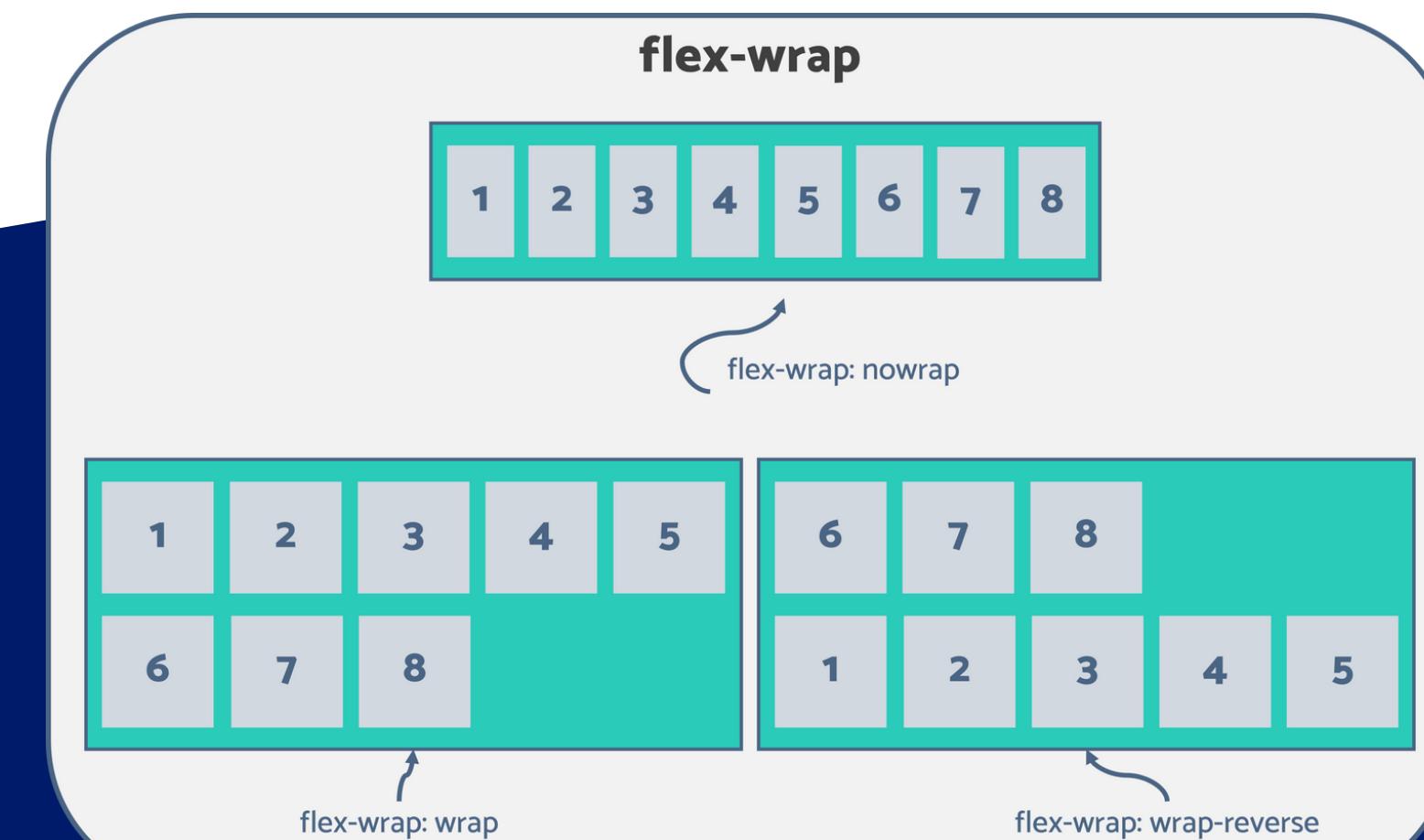


Flex Layout



flex-wrap : เป็นการกำหนดว่าจะให้ Flex Item ภายใน Flex Container

- **nowrap** : คือการไม่ให้ Flex Item มีการดันลงมาด้านล่าง และจะมีการลดขนาด Flex Item ให้มีขนาดพอดีกับ Flex Container หาก Flex Container เล็กกว่า Flex Item ก็จะหมด ก็จะทำให้ล้นออกไปทางด้านขวา
- **wrap** : การกำหนดให้ Flex Item มีการดันลงมาด้านล่างตามขนาดของ Flex Container โดยมีการจัดเรียง Flex Item จากบนลงล่าง
- **wrap-reverse** : การกำหนดให้ Flex Item มีการดันลงมาด้านล่างตามขนาดของ Flex Container โดยมีการจัดเรียง Flex Item จากล่างขึ้นบน

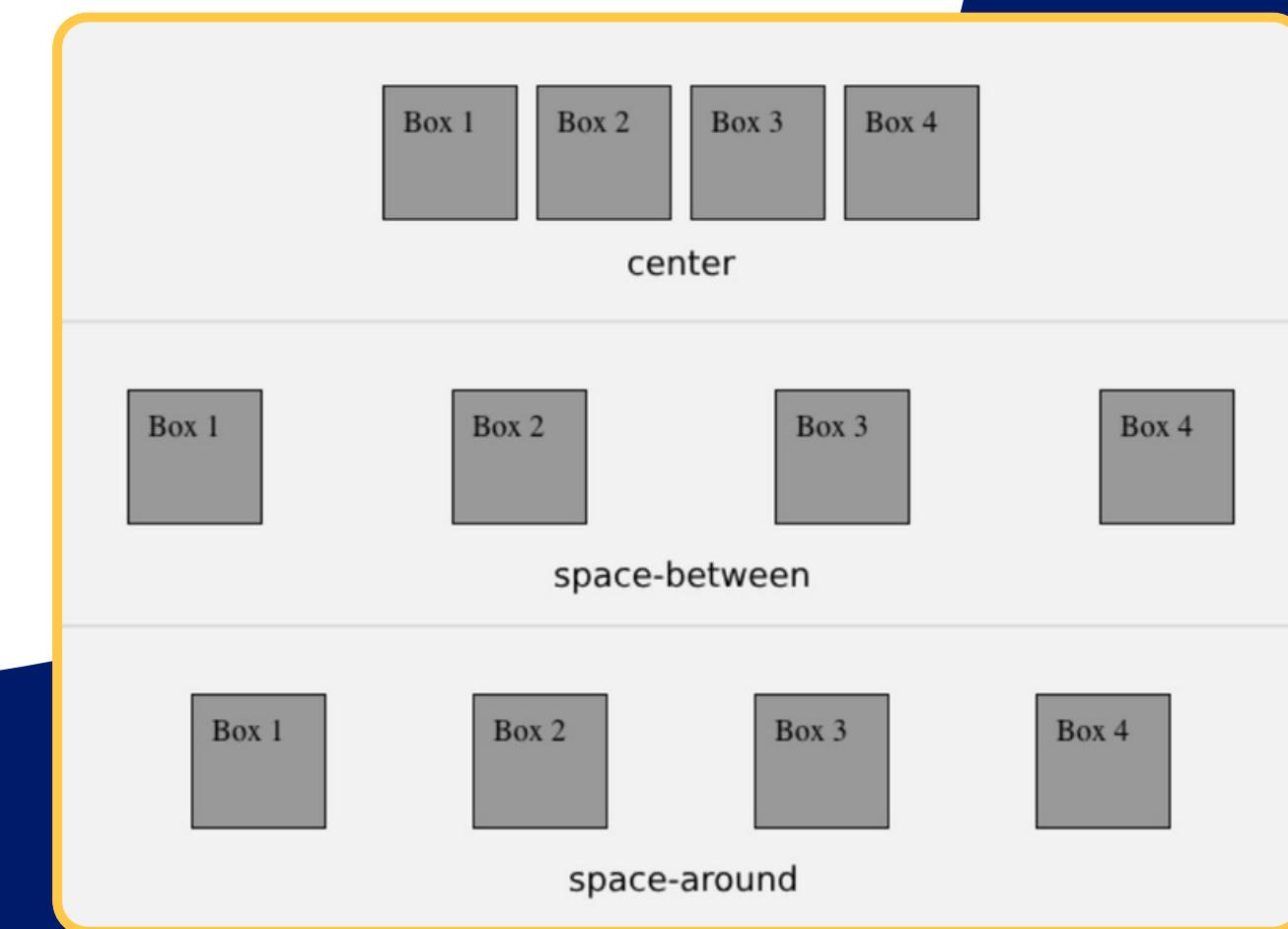
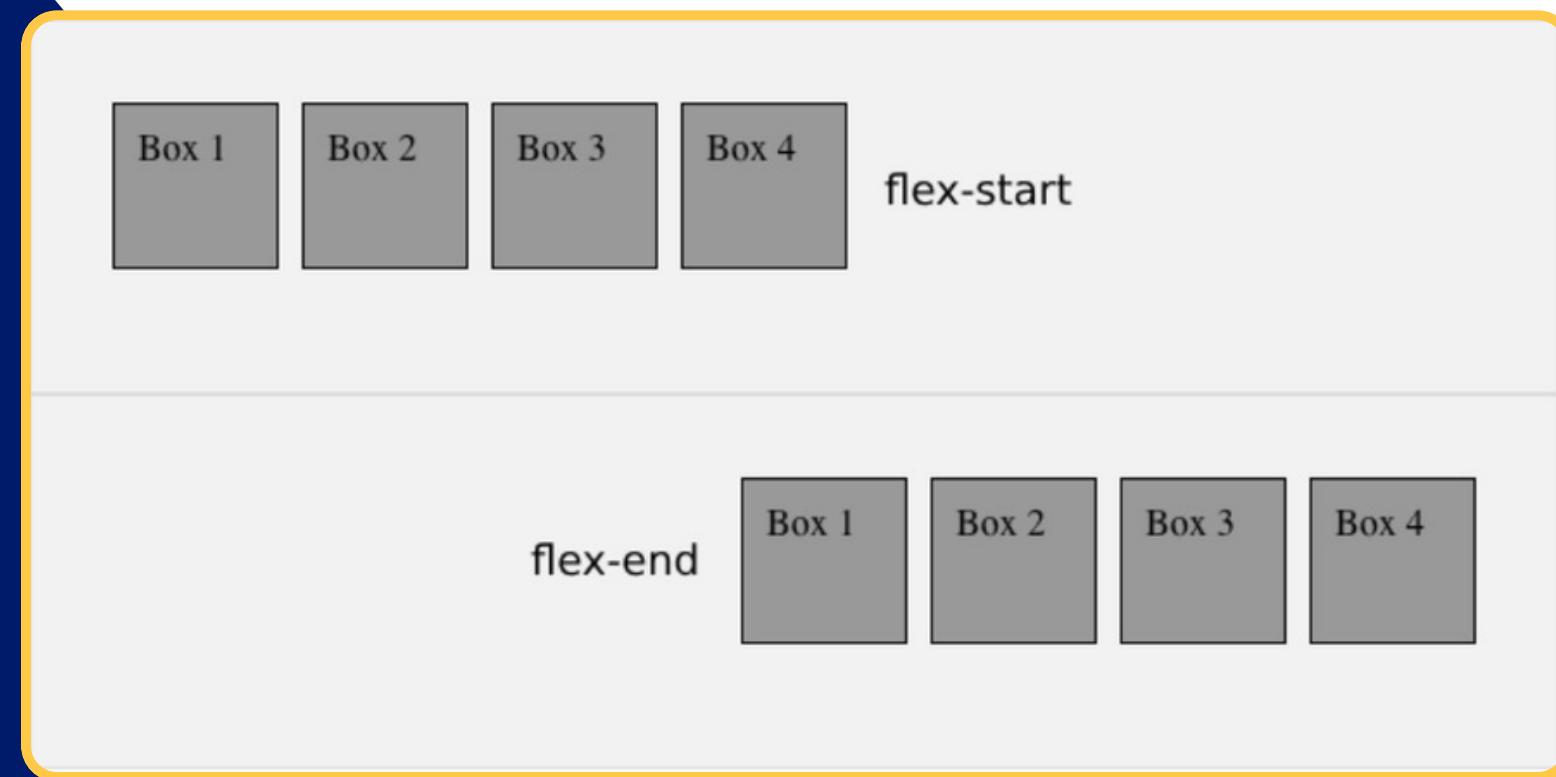


Flex Layout



justify-content : เป็นการกำหนดการจัดตำแหน่งของ Flex Item ภายใน Flex Container

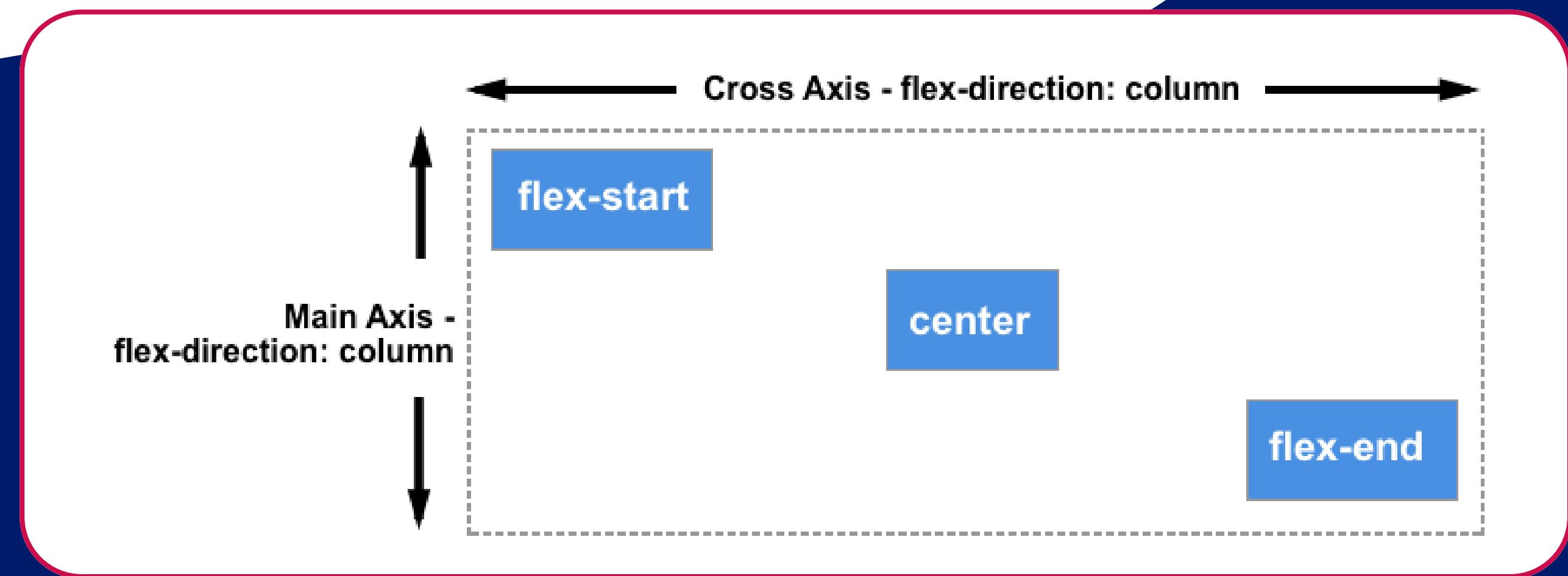
- **flex-start** : จัด Flex Item ให้ชิดซ้าย
- **flex-end** : จัด Flex Item ให้ชิดขวา
- **center** : จัด Flex Item ให้อยู่ตรงกลาง
- **space-between** : จัด Flex Item ให้อยู่ระหว่างกัน
- **space-around** : จัด Flex Item ให้มีช่องว่างระหว่าง ด้านหน้า ตรงกลาง ด้านหลัง ให้เท่ากัน



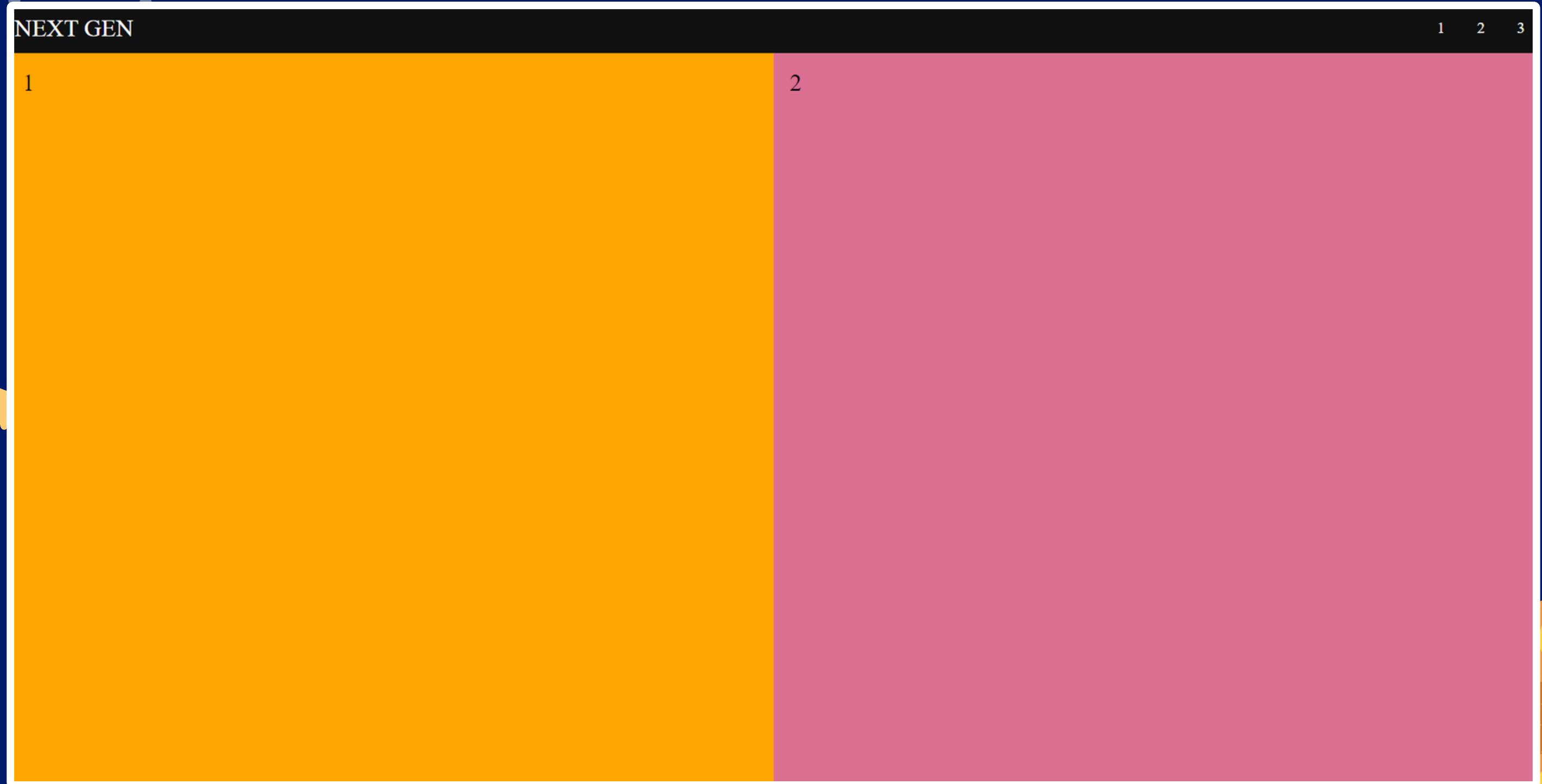
Flex Layout

align-items : เป็นการกำหนดการจัดตำแหน่งของ Flex Item ในแนวตั้ง ภายใน Flex Container

- **flex-start** : จัด Flex Item ให้อยู่บนสุดใน Flex Container
- **center** : จัด Flex Item ให้อยู่ตรงกลางใน Flex Container
- **flex-end** : จัด Flex Item ให้อยู่ล่างสุดใน Flex Container



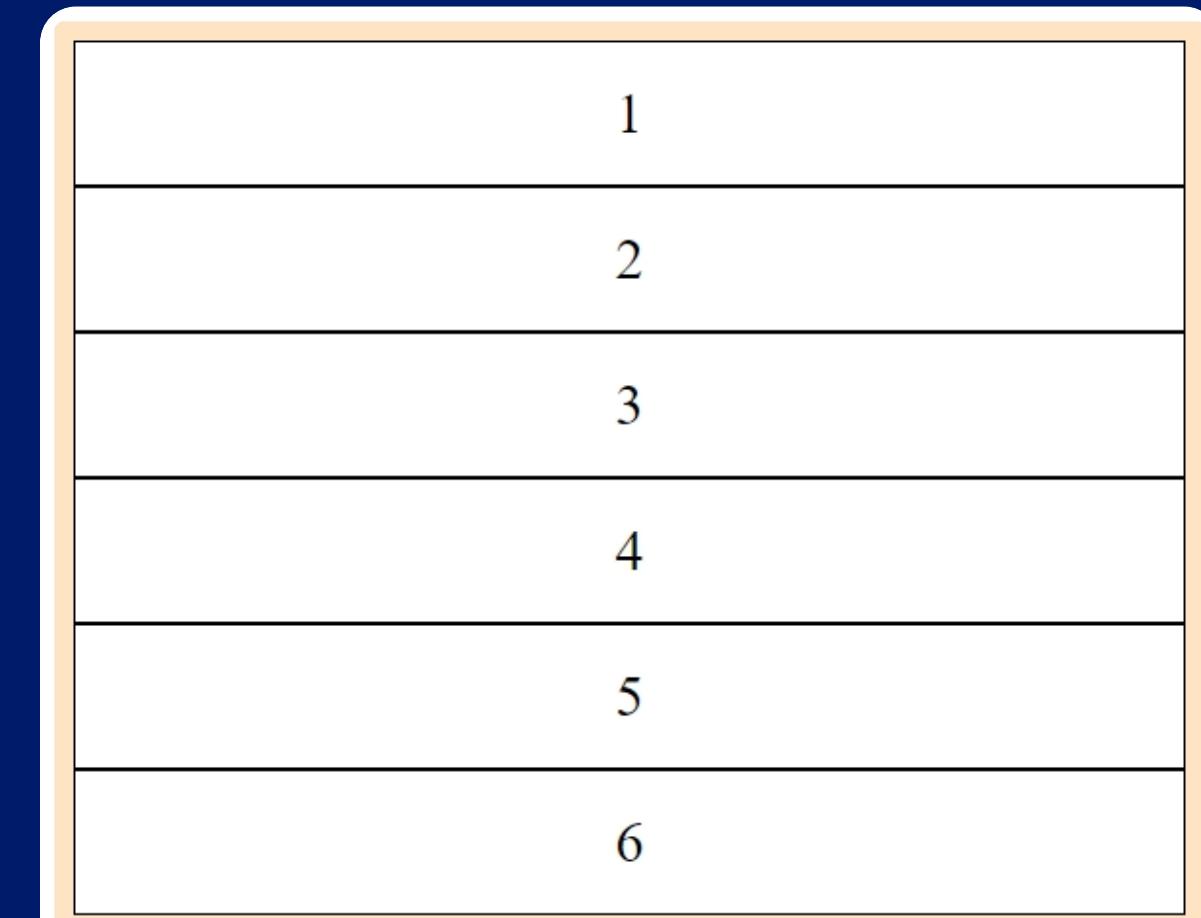
Flex Assignment



Grid Layout

Grid เป็นตัวช่วยในการจัด Layout ให้กับเราให้เปรียบเสมือนตาราง

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Sticky Example</title>
8   <style>
9     .grid-container {
10       display: grid;
11       background-color: bisque;
12       padding: 10px;
13     }
14
15   .grid-container>div {
16     background-color: #fff;
17     text-align: center;
18     padding: 20px 0;
19     font-size: 30px;
20     border: 1px solid black;
21   }
22 </style>
23 </head>
24
25 <body>
26   <div class="grid-container">
27     <div>1</div>
28     <div>2</div>
29     <div>3</div>
30     <div>4</div>
31     <div>5</div>
32     <div>6</div>
33   </div>
34 </body>
35
36 </html>
```



Grid Layout

grid-template-columns - เป็นการกำหนดว่าอยากให้มี column มีขนาดเท่าไหร่

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Sticky Example</title>
8   <style>
9     .grid-container {
10       display: grid;
11       grid-template-columns: 90px auto auto auto;
12       background-color: bisque;
13       padding: 10px;
14     }
15
16     .grid-container>div {
17       background-color: #fff;
18       text-align: center;
19       padding: 20px 0;
20       font-size: 30px;
21       border: 1px solid black;
22     }
23   </style>
24 </head>
25
26 <body>
27   <div class="grid-container">
28     <div>1</div>
29     <div>2</div>
30     <div>3</div>
31     <div>4</div>
32     <div>5</div>
33     <div>6</div>
34   </div>
35 </body>
36
37 </html>
```

1	2	3	4
5	6		

จากตัวอย่างจะกำหนดให้มี 4 col โดย col 1

กำหนดให้มีขนาด 90px
นอกนั้นกำหนดให้เป็น auto คือมีขนาดเท่ากัน

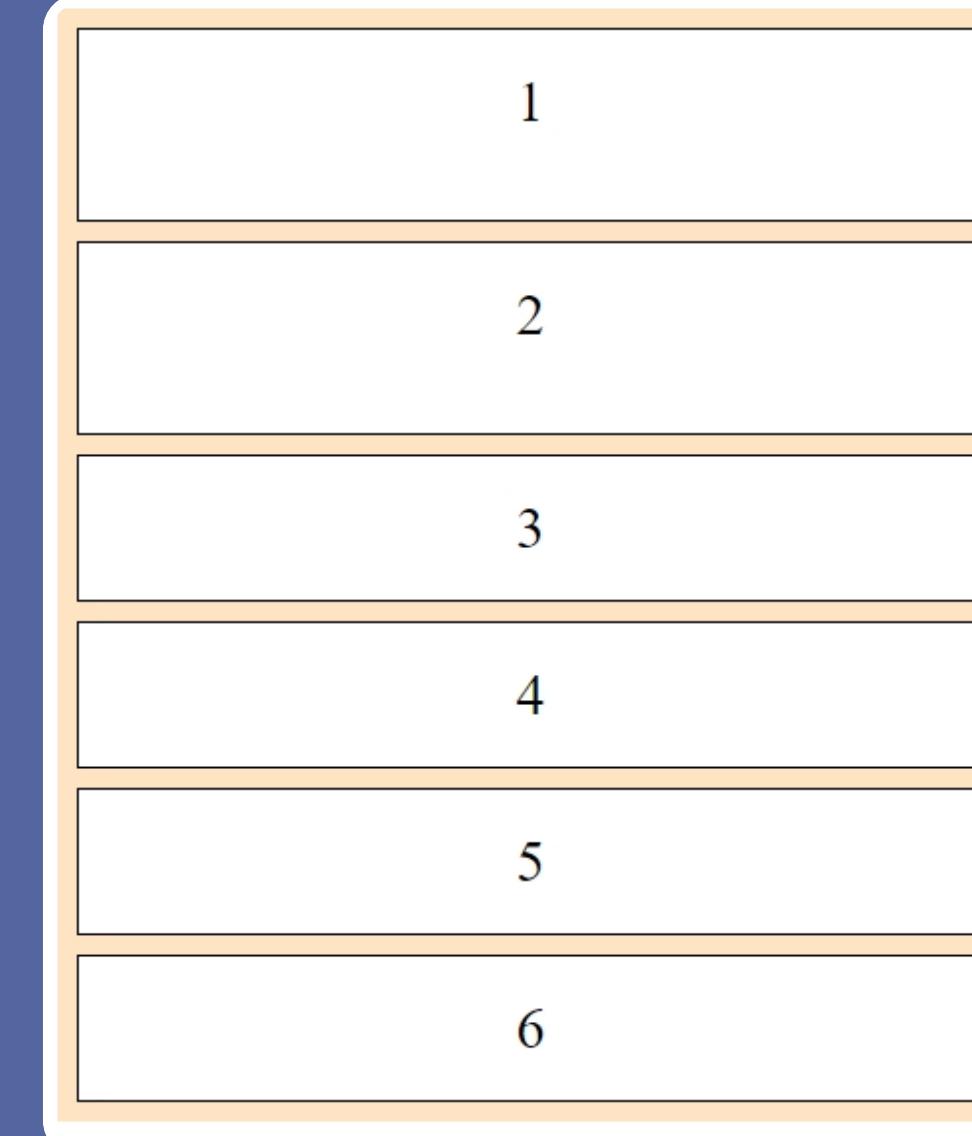


Grid Layout

grid-template-rows - เป็นการกำหนดว่าอย่างให้ row มีขนาดเท่าไหร่



```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Sticky Example</title>
8   <style>
9     .grid-container {
10       display: grid;
11       grid-template-rows: 100px 100px;
12       background-color: bisque;
13       padding: 10px;
14       gap: 10px;
15     }
16
17   .grid-container>div {
18     background-color: #fff;
19     text-align: center;
20     padding: 20px 0;
21     font-size: 30px;
22     border: 1px solid black;
23   }
24 </style>
25 </head>
26
27 <body>
28   <div class="grid-container">
29     <div>1</div>
30     <div>2</div>
31     <div>3</div>
32     <div>4</div>
33     <div>5</div>
34     <div>6</div>
35   </div>
36 </body>
37
38 </html>
```



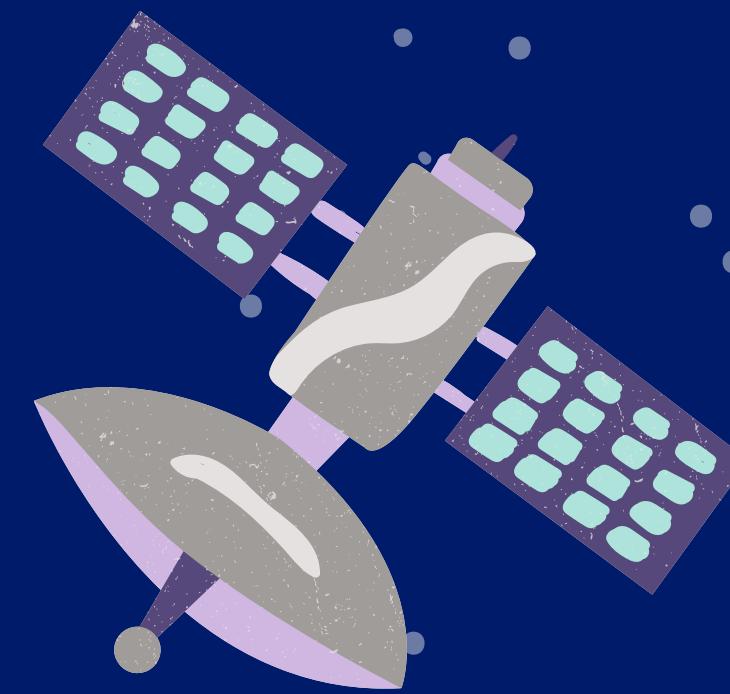
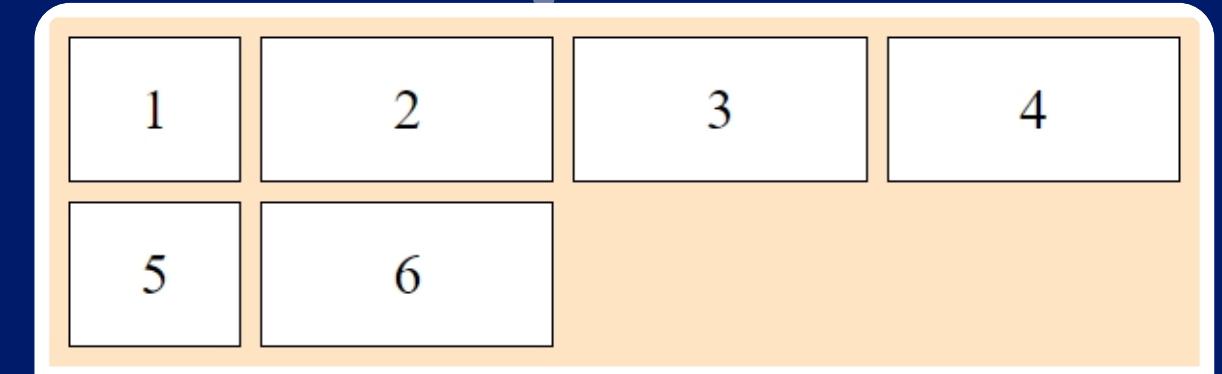
จากตัวอย่าง row1 กับ row2 มีขนาดเท่ากันคือ 100px
นอกจ้านั้นมีขนาดเป็น auto



Grid Layout

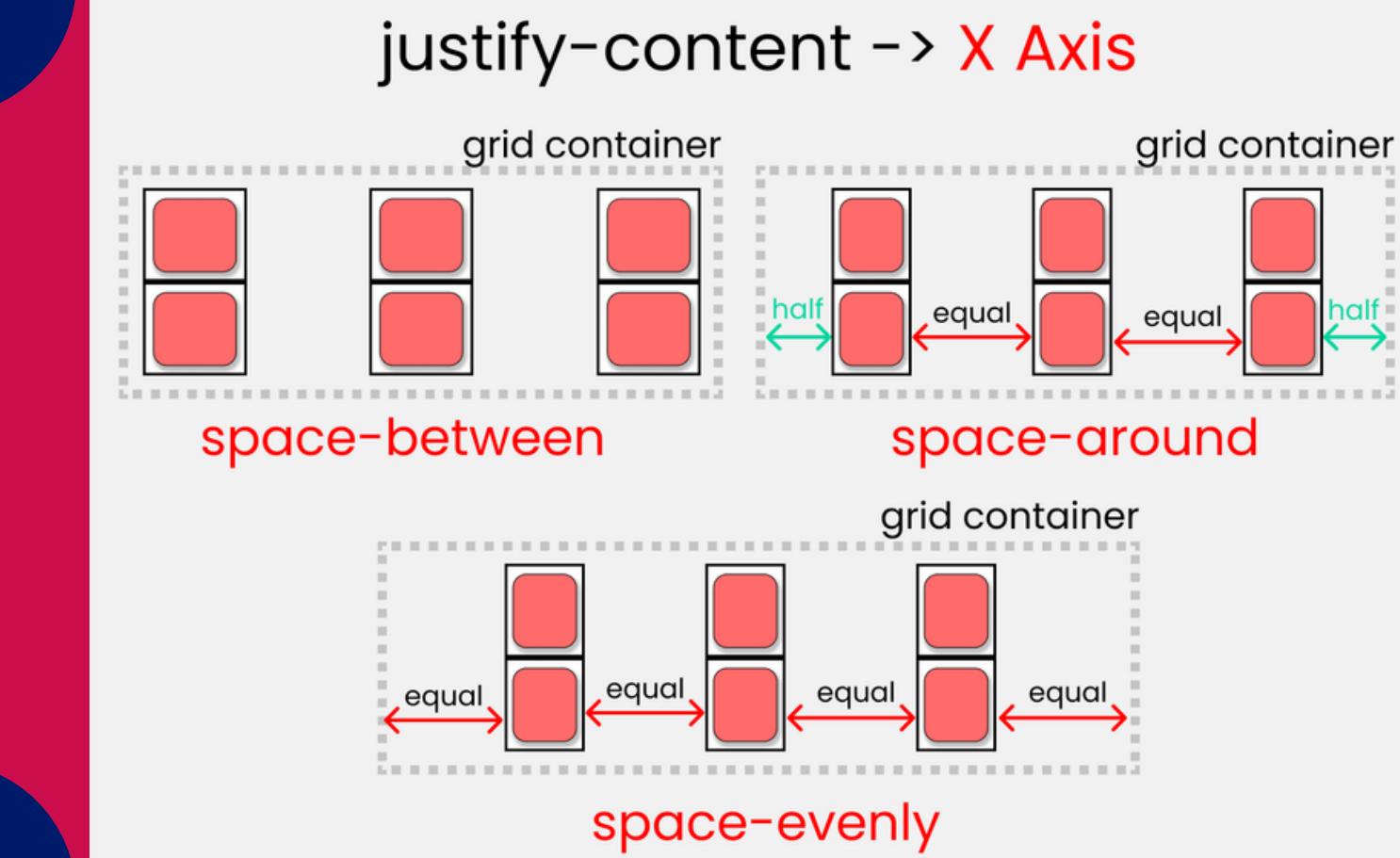
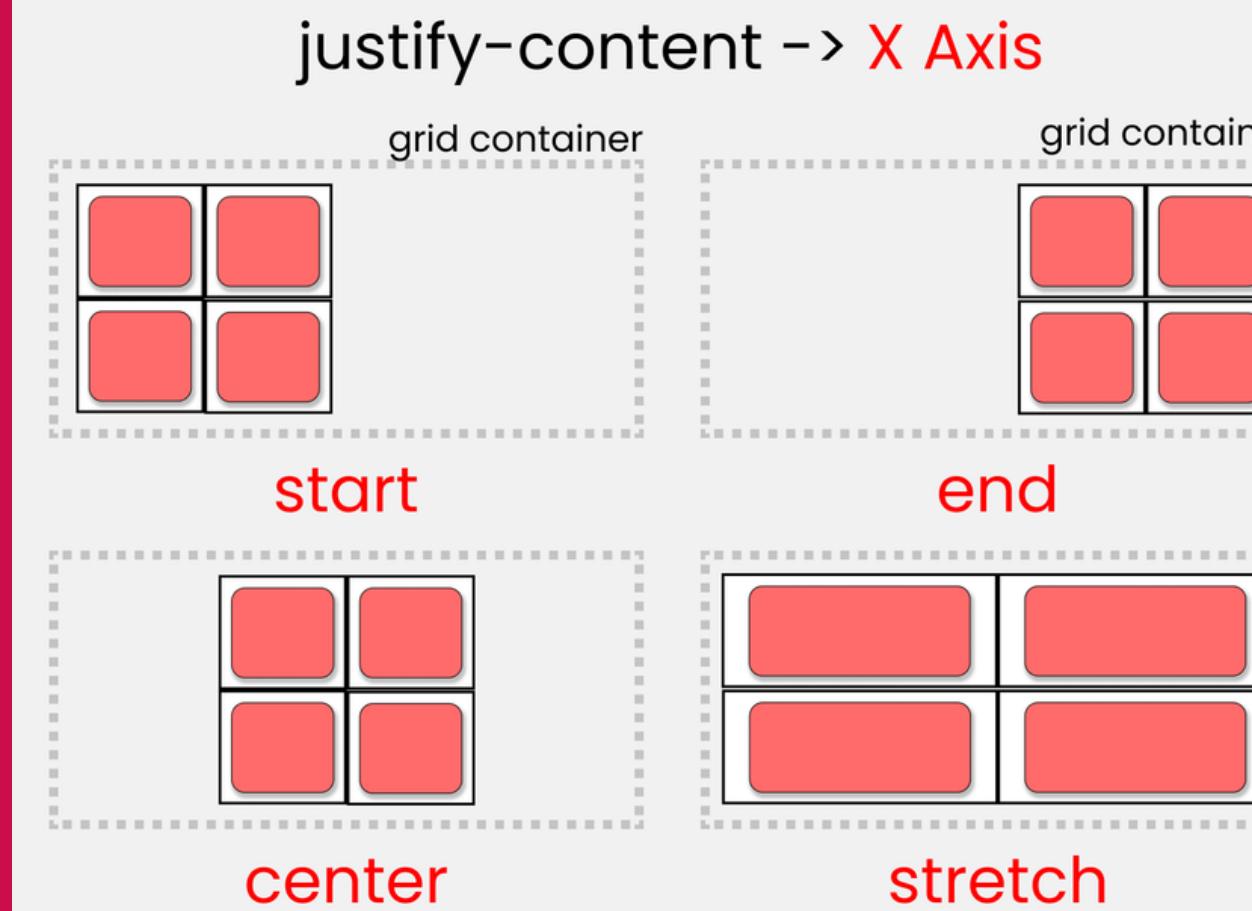
gap - กำหนดระยะห่างระหว่าง col กับ row

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Sticky Example</title>
8   <style>
9     .grid-container {
10       display: grid;
11       grid-template-columns: 90px auto auto auto;
12       background-color: bisque;
13       padding: 10px;
14       gap: 10px;
15     }
16
17     .grid-container>div {
18       background-color: #fff;
19       text-align: center;
20       padding: 20px 0;
21       font-size: 30px;
22       border: 1px solid black;
23     }
24   </style>
25 </head>
26
27 <body>
28   <div class="grid-container">
29     <div>1</div>
30     <div>2</div>
31     <div>3</div>
32     <div>4</div>
33     <div>5</div>
34     <div>6</div>
35   </div>
36 </body>
37
38 </html>
```



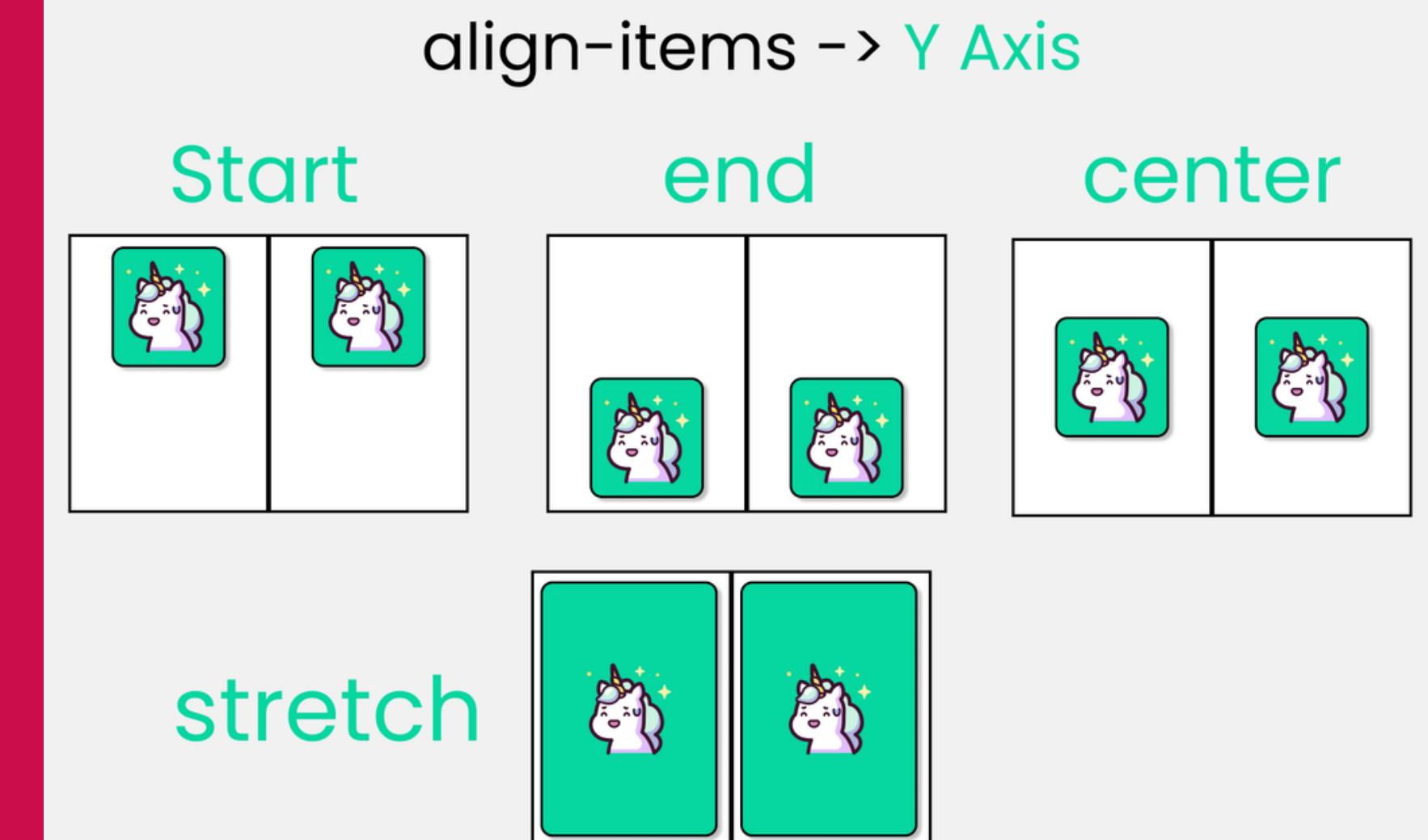
Grid Layout

justify-content - กำหนดรูปแบบการ align ของ cell ในแถวตามแกน x .

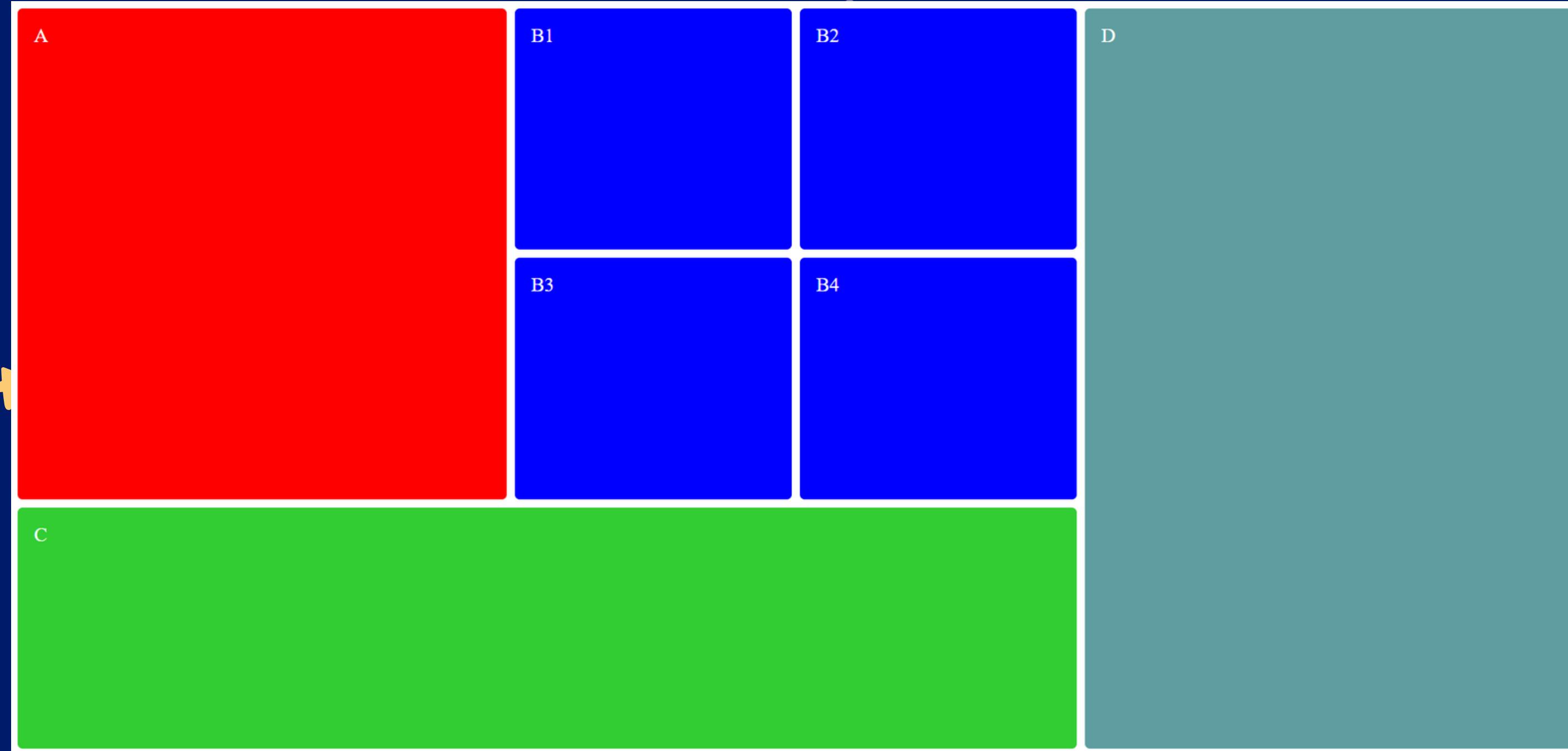


Grid Layout

align items = กำหนดรูปแบบการ align ของ cell ในตัวางตามแกน Y



Grid Assignment



Media Queries

Media Query ใช้ในการทำ responsive (การทำเว็บให้รองรับทุกหน้าจอ) โดยใช้การกำหนดเป็นเงื่อนไขว่า ถ้าหน้าจอขนาดเท่านี้ต้องการให้หน้าเว็บแสดงผลอย่างไร

```
1 @media screen and (max-width: xxxpx) and (min-width: xxxpx) {  
2     .class {  
3         property: value;  
4     }  
5 }
```

max-width: xxxpx คือ เมื่อความกว้างมากกว่า xxxpx
min-width: xxxpx คือ เมื่อความกว้างน้อยกว่า xxxpx

Media Queries



```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5    <meta charset="UTF-8">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <title>This is title</title>
8
9  <style>
10   @media screen and (max-width: 1000px) and (min-width: 500px) {
11     .example {
12       background-color: red;
13     }
14   }
15  </style>
16 </head>
17
18 <body>
19   <div class="example">example</div>
20 </body>
21
22 </html>
```

ผลลัพธ์ในขนาดหน้าจอระหว่าง 500 – 1000px

example

ผลลัพธ์นอกเหนือขนาดหน้าจอระหว่าง 500 – 1000px

example

Responsive Assignment

