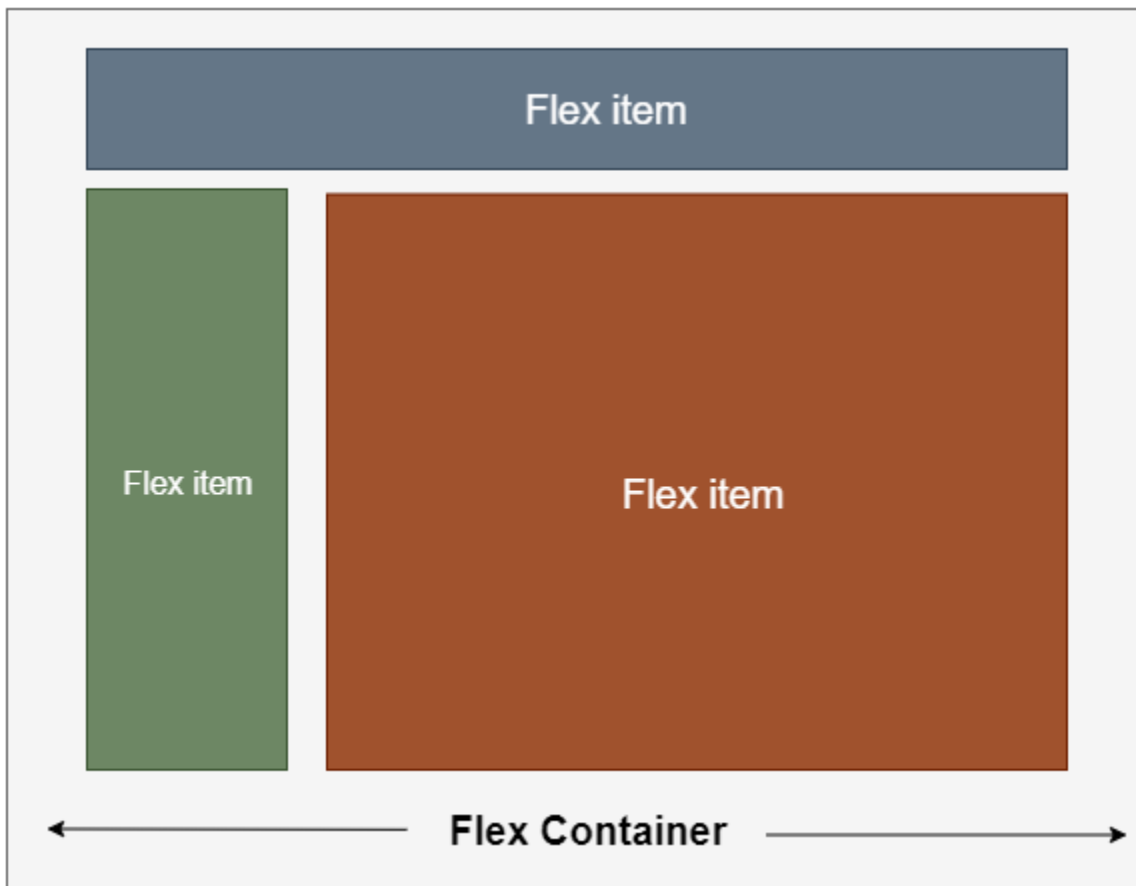


Bootstrap

1. Flexbox หรือ Flexible box เป็นเครื่องมือทาง CSS (Cascading Style Sheets) ที่ช่วยให้การจัดส่วนประกอบต่างบนหน้าเว็บไซต์ทำได้ง่ายขึ้น มีความยืดหยุ่นสูง โดยปกติ ถ้าจะใช้ layout mode กับส่วนไหน ต้องเขียน property “display” ให้กับส่วนนั้น แต่การใช้ flexbox จะมีความแตกต่างกันตรงที่ต้องกำหนด property “display” ไว้ที่ส่วนของ container แทน

flexbox ประกอบด้วย

1. Flex Container หรือ container คือส่วนที่กำหนด property “display” ให้เป็น “flex” และเป็นกล่องที่เก็บ flex items
2. Flex items คือส่วนต่างๆที่อยู่ภายใน flex container



สามารถจัดแต่ละ item ที่อยู่ใน container เดียวกันให้อยู่ในรูปแบบที่ต้องการได้โดยไม่ต้องเขียน CSS Class เอง เช่น การจัดเรียง item ในแนวนอน, แนวตั้ง , ตรงกลางหรือจะสลับตำแหน่งของ item การใช้ความสามารถของ flexbox ได้ จำเป็นต้องสร้าง flex container ขึ้นมาก่อน ตามตัวอย่างโค้ดต่อไปนี้

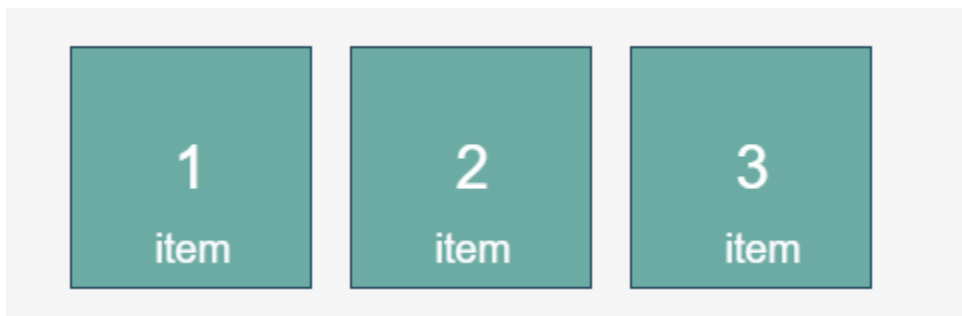
สร้าง flex container :

```
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>
```

กำหนด property “display: flex” :

```
.flex-container {
  display: flex;
}
```

จากโค้ดด้านบน ได้ทำการสร้าง flex container และกำหนดให้เป็น flex ซึ่ง <div> ที่อยู่ภายใน flex container ก็คือ flex items



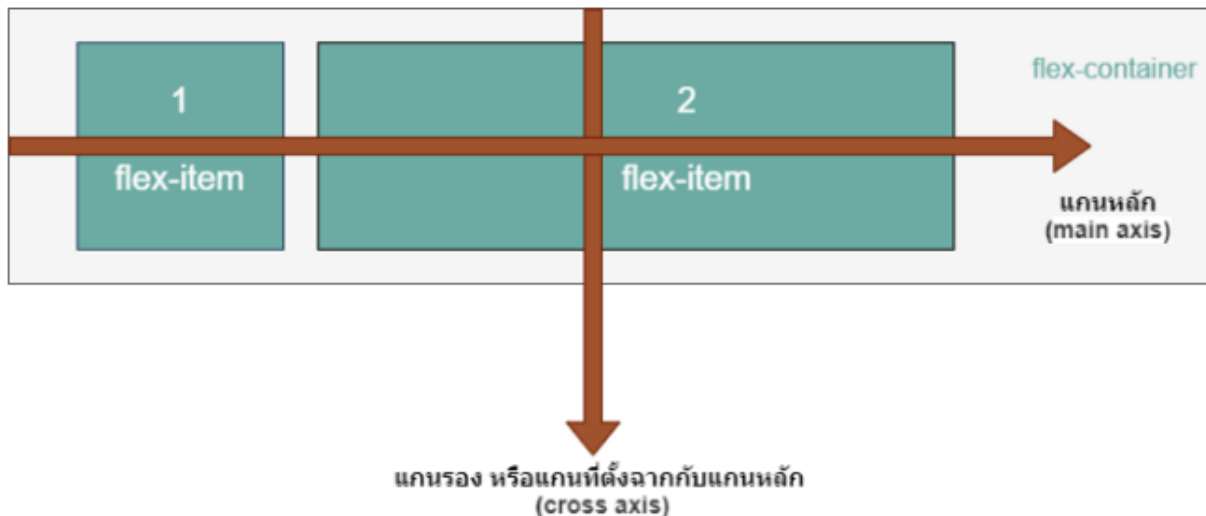
คุณสมบัติของ Flex Container

เมื่อสร้าง flex container แล้ว สามารถใช้คุณสมบัติต่างๆ(properties) เพื่อจัดการและสั่งงานที่ flex container ได้ ประกอบด้วย

- flex-direction
- flex-wrap
- justify-content
- align-items
- align-content

flex-direction

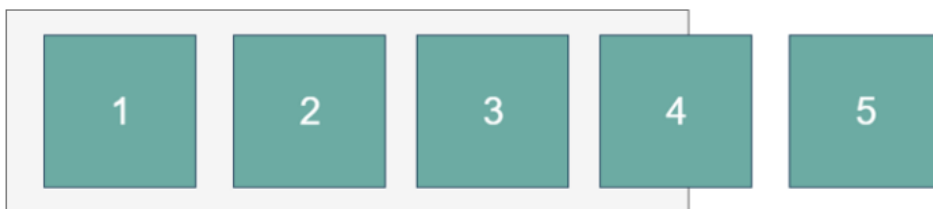
ใช้กำหนดทิศทางหลักของ flex items ที่อยู่ภายใน flex container



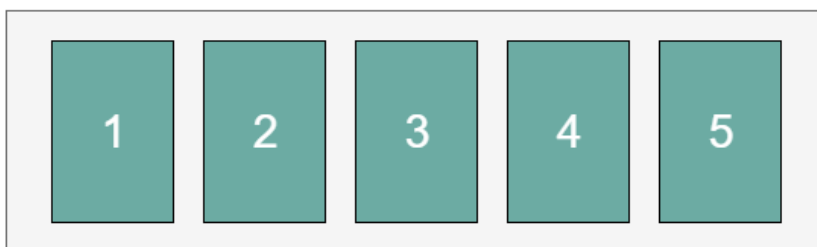
flex-wrap

กำหนดการวางตัวของ item เมื่อมีขนาดใหญ่มากกว่าขนาดของ container

```
.flex-container {  
  display: flex;  
  flex-wrap: nowrap;  
}
```



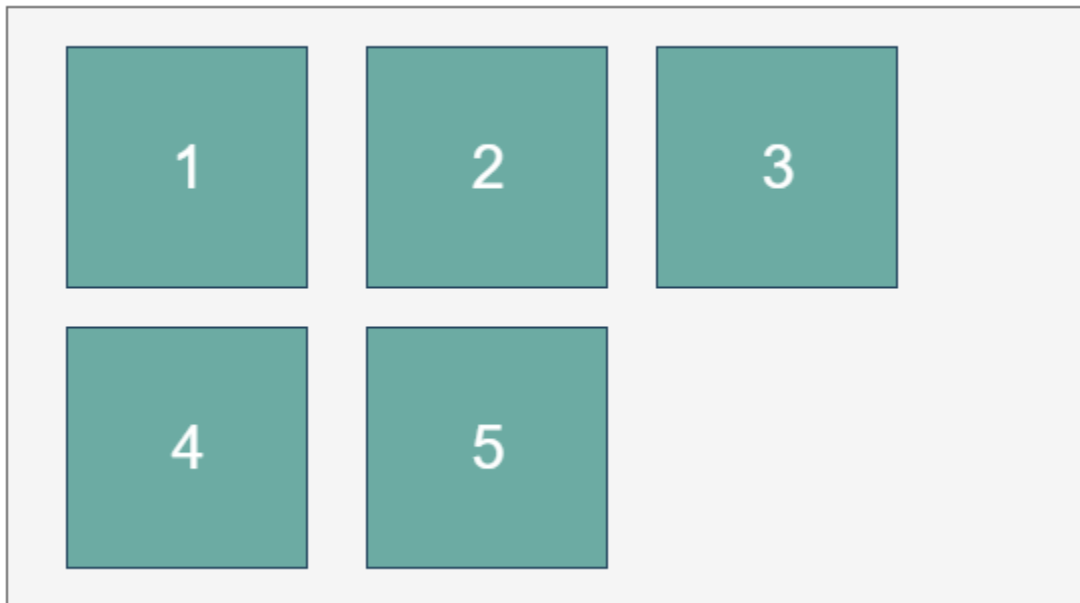
ซึ่งสามารถแก้ไขปัญหานี้ได้โดยลดขนาดของ items ด้วยการบีบให้เล็กลง เพื่อให้มีขนาดพอดีกับ container



flex-flow

เป็นการเขียนแบบ shorthand สำหรับการกำหนดค่าของ flex-direction และ flex-wrap

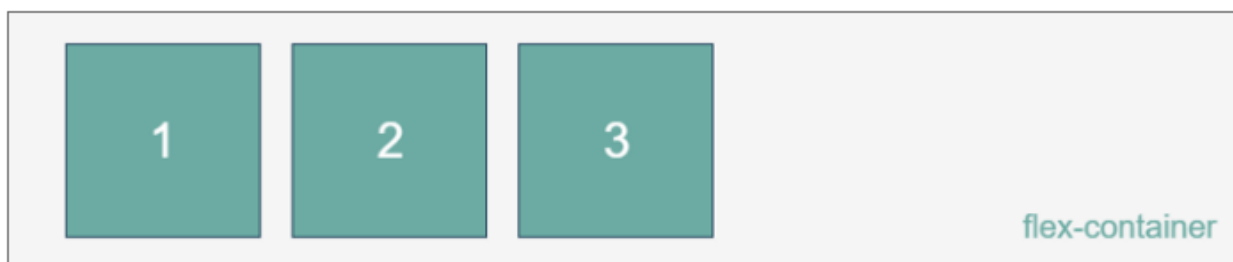
```
.flex-container {  
  display: flex;  
  flex-flow: row wrap;  
}
```



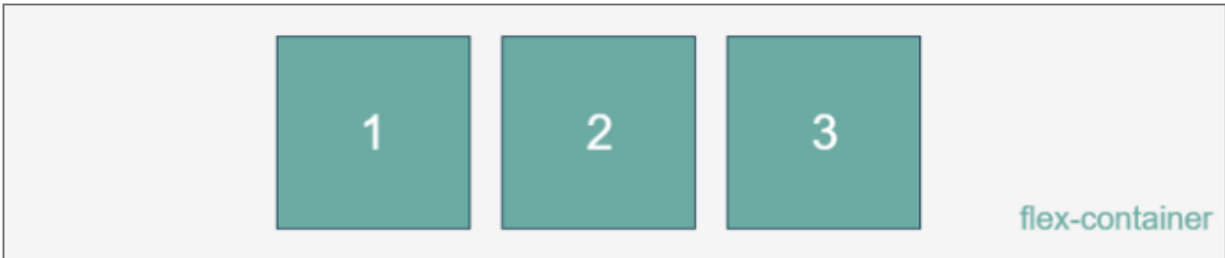
justify-content

กำหนดการจัดตำแหน่งของ items เมื่อเทียบกับทิศทางของแกนหลัก

```
.flex-container {  
  display: flex;  
  justify-content: flex-start;  
}
```



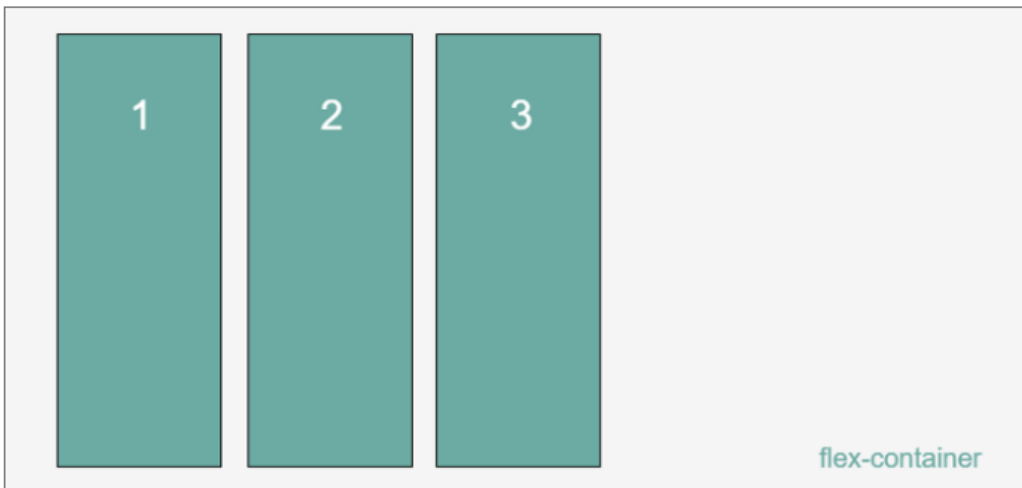
```
.flex-container {  
  display: flex;  
  justify-content: center;  
}
```



align-items

กำหนดการจัดตำแหน่งของ items เมื่อเทียบกับทิศตรงกันข้ามของแกนหลัก

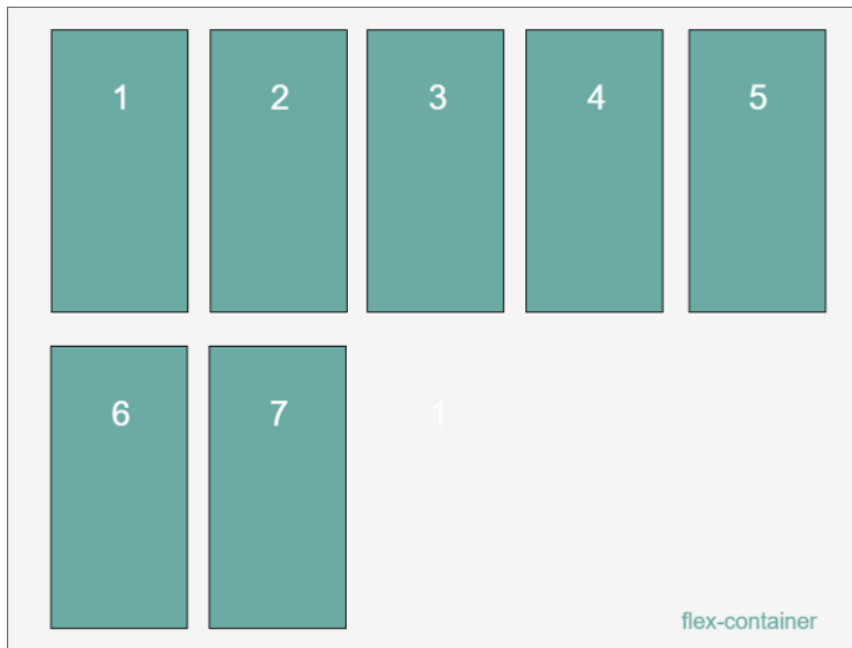
```
.flex-container {  
  display: flex;  
  align-items: stretch;  
}
```



align-content

ใช้สำหรับกำหนด items ใน container ซึ่งจะมีผลเมื่อมีการ wrap หรือการขึ้นบรรทัดใหม่เกิดขึ้น

```
.flex-container {  
  display: flex;  
  flex-wrap: wrap;  
  align-content: stretch;  
}
```



2. การกำหนดรูปแบบการแสดงผล เช่นการกำหนด border inline block

ค่าโครงการแสดงผล inline-block แต่ละค่าก็จะมีมีความแตกต่างกัน การกำหนดค่าของ display ก็จะมีดังนี้

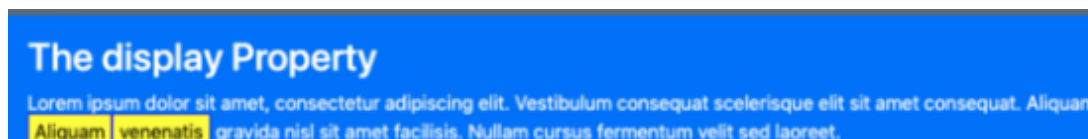
```
display: inline    //เรียงกันแนวนอน และมีพื้นที่จำกัด
display: inline-block  //เรียงกันแนวนอนและมีพื้นที่ด้านใน ตามที่เรากำหนด
display: block      //เรียงกันแนวตั้ง และมีพื้นที่จำกัด
```

The display

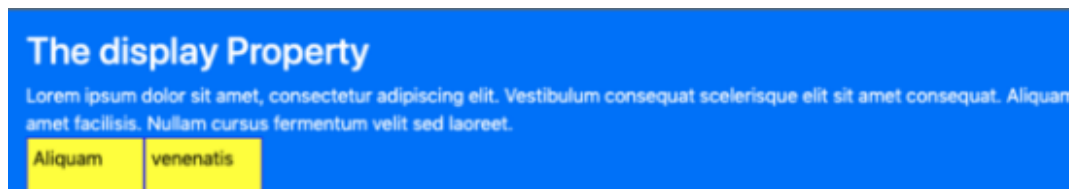
คือการจัดรูปแบบของ Box ให้เป็นแบบแนวตั้ง แนวนอน และมีพื้นที่ด้านใน และในตัวอย่างนี้จะยกมาทั้ง 3 ค่า และที่เด่นๆ เห็นจะเป็น inline-block คือเขาจะ นำ Box มาเรียงกันในบรรทัดเดียว และยังมีพื้นที่ด้านนอกภายใน Box ให้ด้วย

```
span.a{
  display: inline;    //เปลี่ยนค่าได้ตามต้องการ
  width: 100px;       //ความกว้างของขอบ
  height: 100px;      //ความสูงของขอบ
  color: black;        //สีตัวอักษร
  padding: 5px;        //ขยายขอบ
  border: 1px solid blue; //ความหนา รูปแบบ สีของเส้นขอบ
  background-color: yellow; //สีพื้นหลัง
}
```

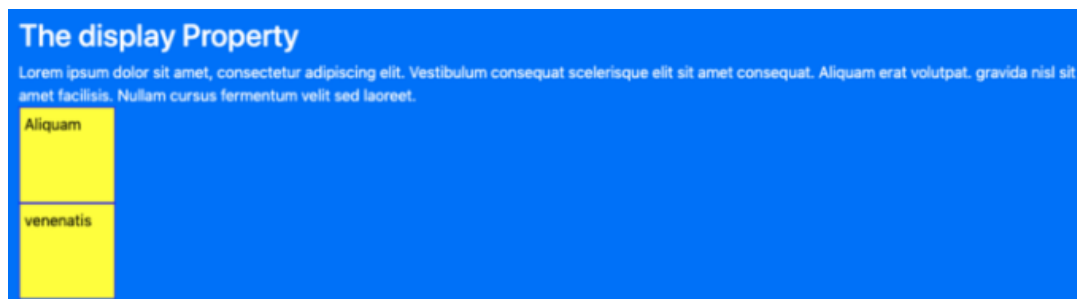
The display: inline Value



The display: inline-block Value



The display: block Value



3.Form

HTML Forms คือการสร้างฟอร์มใน html โดยฟอร์มที่ได้จะถูกใช้ในการส่งข้อมูลไปยังเซิร์ฟเวอร์
แท็กที่ใช้ในการสร้างฟอร์มคือ แท็ก <form>

ฟอร์มประกอบด้วยแท็ก input เช่น กรอบข้อความ (text fields), กล่องแบบเลือก (checkboxes), ปุ่มต่าง ๆ เป็นต้น โดยเราสามารถใส่รายการ, กรอบข้อความ(textarea), ฟิลด์เซต (fieldset), เลเจนด์(legend) และแท็ก label ซึ่งเราจะใช้แท็ก input ใช้ในการเก็บข้อมูลจากผู้ใช้งาน โดยแท็ก input สามารถสร้างได้หลายแบบ ขึ้นอยู่กับแอตทริบิวต์ type ดังนั้นแท็ก input สามารถเป็นกรอบข้อความ, กล่องตัวเลือก, กล่องรหัสผ่าน, ปุ่มส่งข้อมูล และอื่น ๆ

1. กรอบข้อความ (Text fields)

แท็ก input และแอตทริบิวต์ type="text" จะกำหนดการป้อนข้อความ 1 บรรทัด

```
<html>
<body>

<form>
ชื่อ: <input type="text" name="firstname"><br>
นามสกุล: <input type="text" name="lastname">
</form>

</body>
</html>
```

ผลลัพธ์ที่ได้

ชื่อ:
นามสกุล:

4. การทำ navigation

Navigation ใช้สำหรับจัดกลุ่มส่วนที่เป็นลิงค์ต่างๆ ไม่ว่าจะเป็นลิงค์ที่ไปยังไซต์อื่นๆ หรือลิงค์ที่ใช้เชื่อมโยงภายในหน้าเว็บเพจ (เช่นลิงค์ที่เมื่อคลิกแล้วเลื่อนไปแสดงยังส่วนต่างๆของหน้าเพจ) และที่สำคัญคือส่วนที่เป็นเมนูของเว็บไซต์

```
<body>
<header>
  <h1>Website Header</h1>
  <nav>
    <ul>
      <li><a href="home.html">Home</a></li>
      <li><a href="portfolio.html">Portfolio</a></li>
      <li><a href="about.html">Contact Us</a></li>
    </ul>
  </nav>
</header>
  

</body>
```