



M4. BASIC WEB DEVELOPMENT

Flex Layout

TABLE LAYOUT

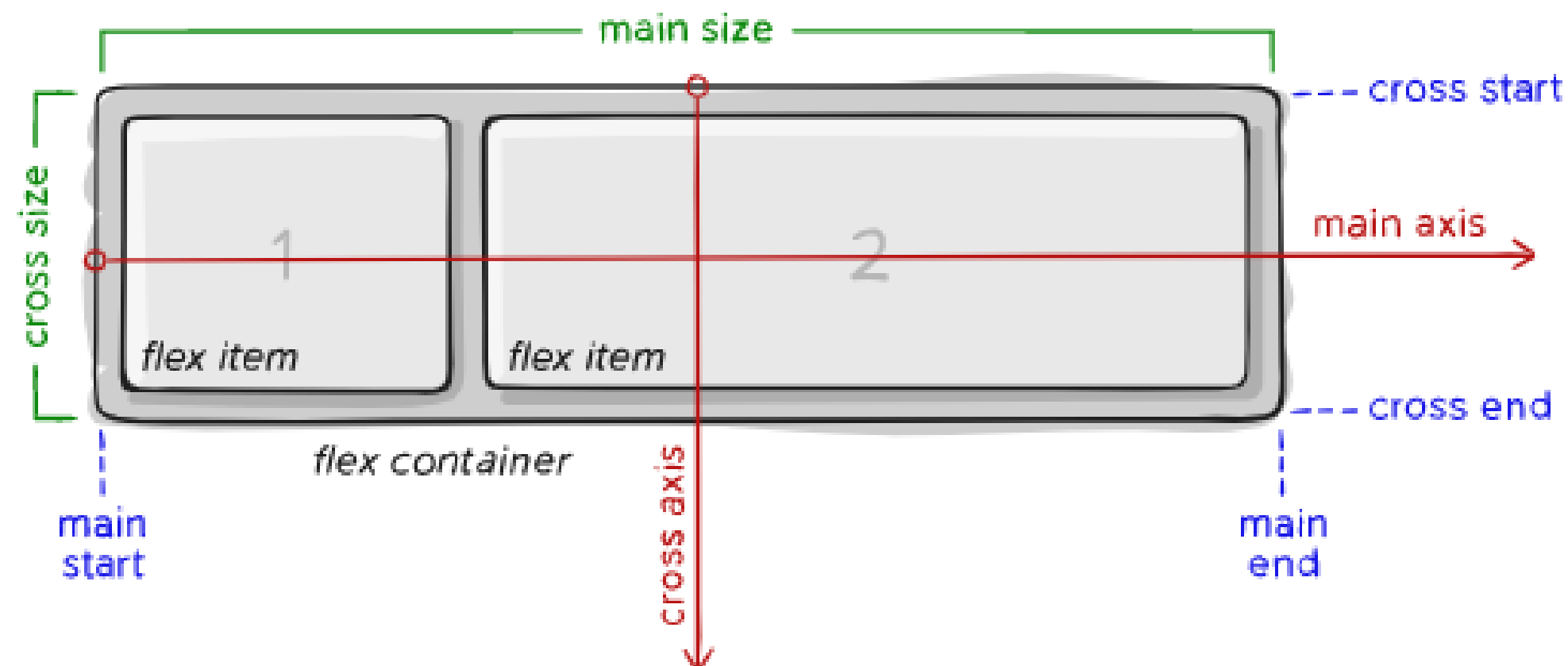
- ปรับตำแหน่งยาก
 - ต้องคำนวณพื้นที่จำนวน column ทุกส่วนให้เท่ากัน
- จัดขนาดลำบาก ต้องอ้างอิงกับ element ภายใน td
- จัดการกับ responsive ได้ไม่ดี
 - ขนาดของ cell จะต้องลด

FLEXBOX LAYOUT

- เริ่มโปรโมท ตอนตุลาคม 2017
- ใช้จัดการ container และจัดการขนาดความกว้างยาวของ container และ component
 - ถ้าไม่ระบุจะมีการจัดการโดยอัตโนมัติ

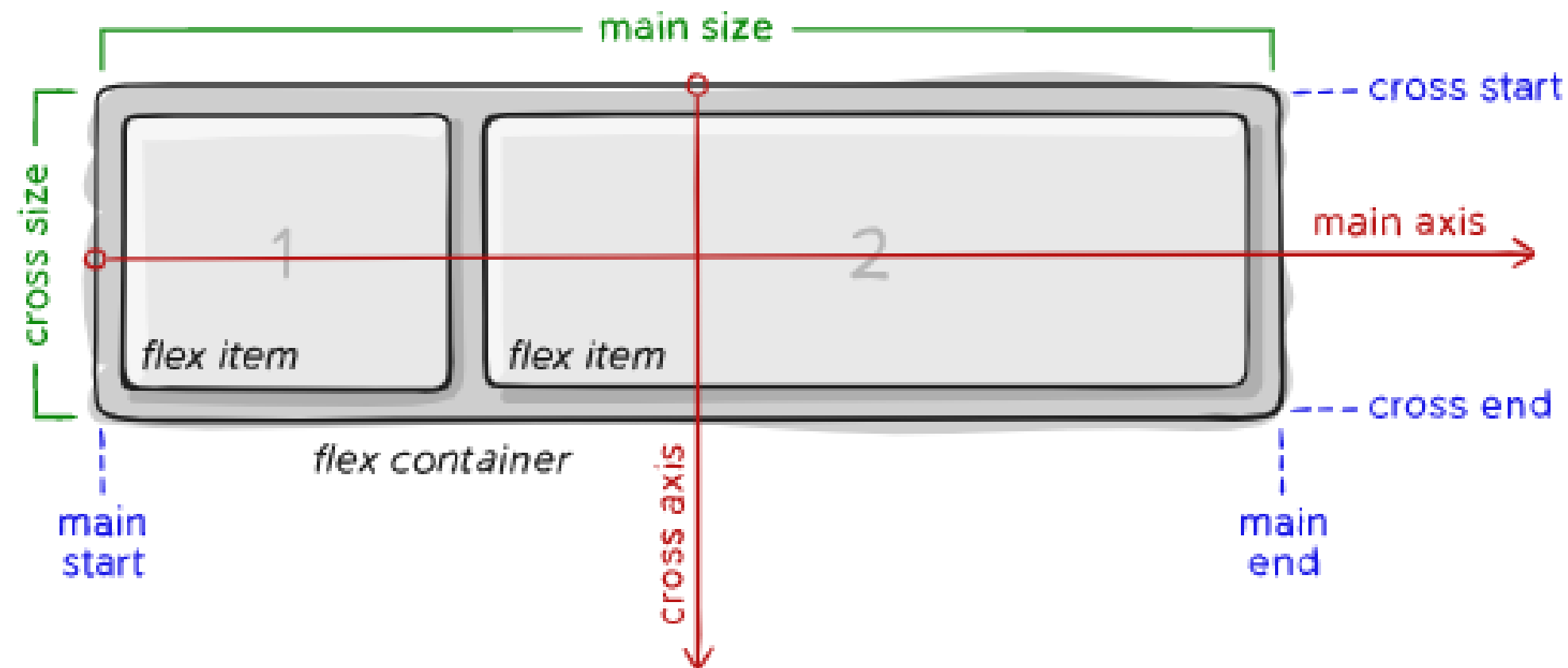
FLEXBOX LAYOUT

- Flexbox จะประกอบด้วย 2 components ที่ระบุรวมกัน



FLEXBOX LAYOUT

- Flexbox จะประกอบด้วย 2 components ที่ระบุรวมกัน



<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
https://www.w3schools.com/css/css3_flexbox.asp

FLEXBOX CONTAINER

- จำเป็นต้องสร้าง container มาก่อน
- ทดลองสร้าง flexstylesheet.css และ flex.html และทดลองพิมพ์ตามนี้

```
.flex-container {  
  display: flex;  
}  
  
.inline-flex-container {  
  display: inline-flex;  
}  
  
.bg-red {  
  background-color: red;  
}  
  
.bg-blue {  
  background-color: skyblue;  
}
```

```
<body>  
  <div class="flex-container bg-blue">  
    <div>1</div>  
  </div>  
  <div class="inline-flex-container bg-red">  
    <div>1</div>  
  </div>  
</body>
```

FLEXBOX CONTAINER

- จำเป็นต้องสร้าง container มาก่อน
- ทดลองสร้าง flexstylesheet.css และ flex.html และทดลองพิมพ์ตามนี้

```
.flex-container {  
  display: flex;  
}  
  
.inline-flex-container {  
  display: inline-flex;  
}  
  
.bg-red {  
  background-color: red;  
}  
  
.bg-blue {  
  background-color: skyblue;  
}
```

```
<body>  
  <div class="flex-container bg-blue">  
    <div>1</div>  
  </div>  
  <div class="inline-flex-container bg-red">  
    <div>1</div>  
  </div>  
</body>
```

1
1



1

FLEXBOX CONTAINER

- สร้าง style ให้กับ div ที่อยู่ใต้ class

```
.flex-container {  
  display: flex;  
}  
  
.inline-flex-container {  
  display: inline-flex;  
}  
  
.bg-red {  
  background-color: red;  
}  
  
.bg-blue {  
  background-color: skyblue;  
}
```

```
.flex-container>div {  
  background-color: darkgray;  
  margin: 10px;  
  padding: 20px;  
  font-size: 30px;  
}
```

```
<body>  
  <div class="flex-container bg-blue">  
    <div>1</div>  
  </div>  
  <div class="inline-flex-container bg-red">  
    <div>1</div>  
  </div>  
</body>
```


FLEXBOX CONTAINER

- สร้าง style ให้กับ div ที่อยู่ใต้ class

```
.flex-container {  
  display: flex;  
}  
  
.inline-flex-container {  
  display: inline-flex;  
}  
  
.bg-red {  
  background-color: red;  
}  
  
.bg-blue {  
  background-color: skyblue;  
}
```

```
.flex-container>div {  
  background-color: darkgray;  
  margin: 10px;  
  padding: 20px;  
  font-size: 30px;  
}
```

```
<body>  
  <div class="flex-container bg-blue">  
    <div>1</div>  
  </div>  
  <div class="inline-flex-container bg-red">  
    <div>1</div>  
  </div>  
</body>
```



FLEXBOX CONTAINER

- สร้าง style ให้กับ div ที่อยู่ใต้ class อีกคลาสหนึ่ง

```
.flex-container {  
  display: flex;  
}  
  
.inline-flex-container {  
  display: inline-flex;  
}  
  
.bg-red {  
  background-color: red;  
}  
  
.bg-blue {  
  background-color: skyblue;  
}
```

```
.flex-container>div {  
  background-color: darkgray;  
  margin: 10px;  
  padding: 20px;  
  font-size: 30px;  
}  
  
.inline-flex-container>div {  
  background-color: lightgray;  
  margin: 20px;  
  padding: 10px;  
  font-size: 25px;  
}
```

```
<body>  
  <div class="flex-container bg-blue">  
    <div>1</div>  
  </div>  
  <div class="inline-flex-container bg-red">  
    <div>1</div>  
  </div>  
</body>
```

FLEXBOX CONTAINER

- สร้าง style ให้กับ div ที่อยู่ใต้ class อีกคลาสหนึ่ง

```
.flex-container {  
  display: flex;  
}  
  
.inline-flex-container {  
  display: inline-flex;  
}  
  
.bg-red {  
  background-color: red;  
}  
  
.bg-blue {  
  background-color: skyblue;  
}
```

```
.flex-container>div {  
  background-color: darkgray;  
  margin: 10px;  
  padding: 20px;  
  font-size: 30px;  
}  
  
.inline-flex-container>div {  
  background-color: lightgray;  
  margin: 20px;  
  padding: 10px;  
  font-size: 25px;  
}
```

```
<body>  
  <div class="flex-container bg-blue">  
    <div>1</div>  
  </div>  
  <div class="inline-flex-container bg-red">  
    <div>1</div>  
  </div>  
</body>
```



FLEXBOX CONTAINER

- ทดลอง เพิ่มจำนวน

```
.flex-container {  
  display: flex;  
}  
  
.inline-flex-container {  
  display: inline-flex;  
}  
  
.bg-red {  
  background-color: red;  
}  
  
.bg-blue {  
  background-color: skyblue;  
}
```

```
.flex-container>div {  
  background-color: darkgray;  
  margin: 10px;  
  padding: 20px;  
  font-size: 30px;  
}  
  
.inline-flex-container>div {  
  background-color: lightgray;  
  margin: 20px;  
  padding: 10px;  
  font-size: 25px;  
}
```

```
<body>  
  <div class="flex-container bg-blue">  
    <div>1</div>  
    <div>2</div>  
    <div>3</div>  
    <div>4</div>  
    <div>5</div>  
    <div>6</div>  
    <div>7</div>  
    <div>8</div>  
  </div>  
  <div class="inline-flex-container bg-red">  
    <div>1</div>  
    <div>2</div>  
    <div>3</div>  
    <div>4</div>  
    <div>5</div>  
    <div>6</div>  
    <div>7</div>  
    <div>8</div>  
  </div>  
</body>
```

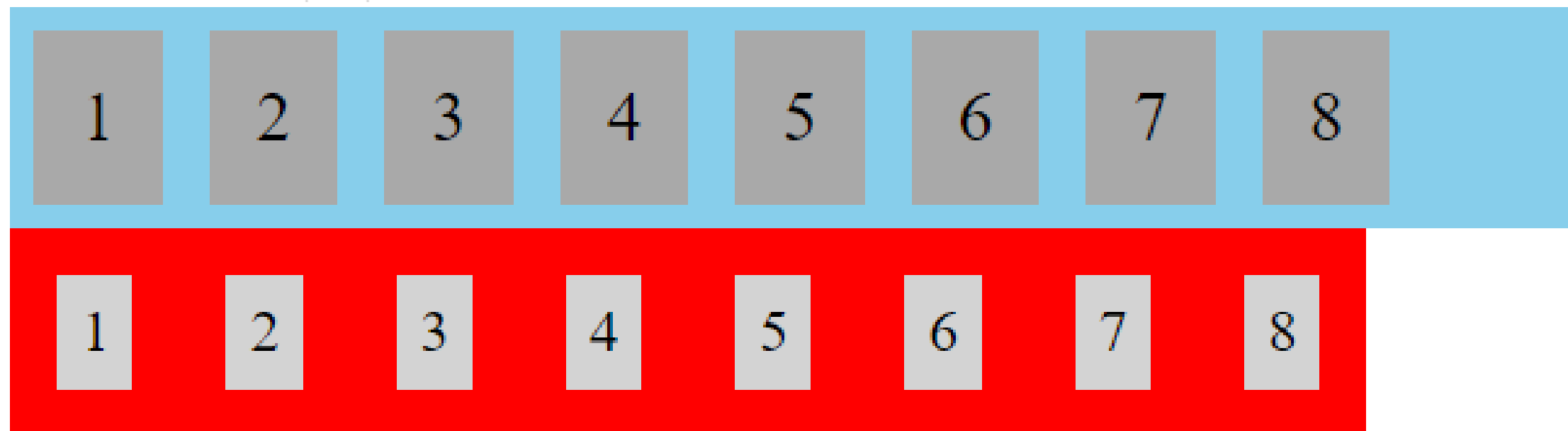
FLEXBOX CONTAINER

- ทดลอง เพิ่มจำนวน

```
.flex-container {  
  display: flex;  
}  
  
.inline-flex-container {  
  display: inline-flex;  
}  
  
.bg-red {  
  background-color: red;  
}  
  
.bg-blue {  
  background-color: skyblue;  
}
```

```
.flex-container>div {  
  background-color: darkgray;  
  margin: 10px;  
  padding: 20px;  
  font-size: 30px;  
}  
  
.inline-flex-container>div {  
  background-color: lightgray;  
  margin: 20px;  
  padding: 10px;  
  font-size: 25px;  
}
```

```
<body>  
  <div class="flex-container bg-blue">  
    <div>1</div>  
    <div>2</div>  
    <div>3</div>  
    <div>4</div>  
    <div>5</div>  
    <div>6</div>  
    <div>7</div>  
    <div>8</div>  
  </div>  
  <div class="inline-flex-container bg-red">  
    <div>1</div>  
    <div>2</div>  
    <div>3</div>  
  </div>  
</body>
```



FLEXBOX CONTAINER

- Flex direction

```
.flex-container {  
  display: flex;  
  direction: row;  
}  
  
.inline-flex-container {  
  display: inline-flex;  
  direction: row;  
}  
  
.bg-red {  
  background-color: red;  
}  
  
.bg-blue {  
  background-color: skyblue;  
}  
  
.flex-container>div {  
  background-color: darkgray;  
  margin: 10px;  
  padding: 20px;  
  font-size: 30px;  
}  
  
.inline-flex-container>div {  
  background-color: lightgray;  
  margin: 20px;  
  padding: 10px;  
  font-size: 25px;  
}
```

```
<body>  
  <div class="flex-container bg-blue">  
    <div>1</div>  
    <div>2</div>  
    <div>3</div>  
    <div>4</div>  
    <div>5</div>  
    <div>6</div>  
    <div>7</div>  
    <div>8</div>  
  </div>  
  <div class="inline-flex-container bg-red">  
    <div>1</div>  
    <div>2</div>  
    <div>3</div>  
    <div>4</div>  
    <div>5</div>  
    <div>6</div>  
    <div>7</div>  
    <div>8</div>  
  </div>  
</body>
```

FLEXBOX CONTAINER

- Flex direction

```
.flex-container {  
  display: flex;  
  flex-direction: row-reverse;  
}
```

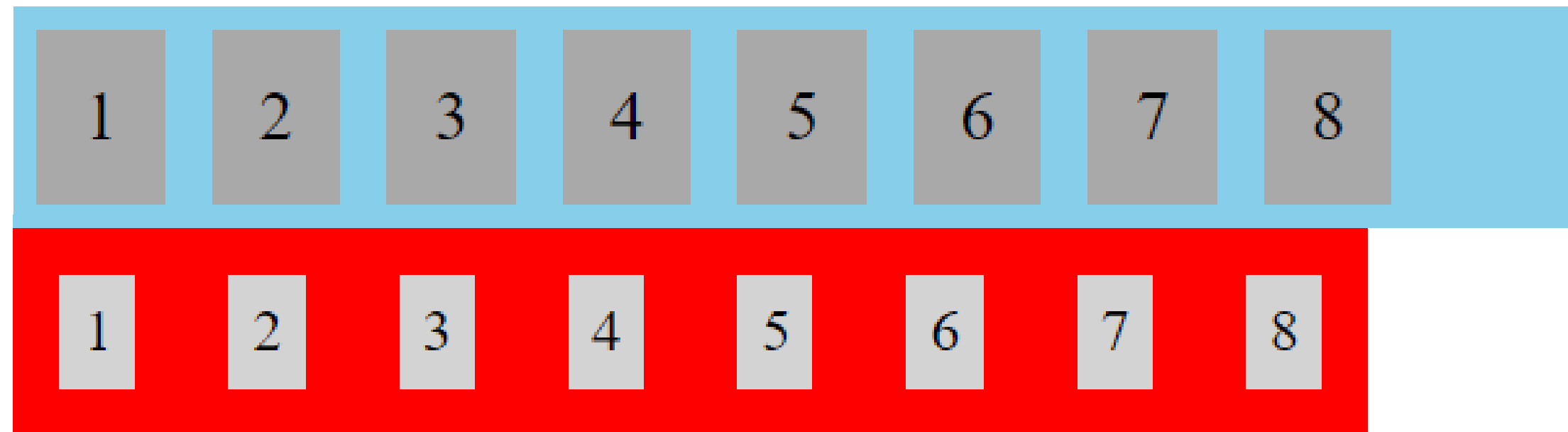
```
.inline-flex-container {  
  display: inline-flex;  
  flex-direction: row-reverse;  
}
```

```
.bg-red {  
  background-color: red;  
}
```

```
.bg-blue {  
  background-color: skyblue;  
}
```

```
.flex-container>div {  
  background-color: darkgray;  
  margin: 10px;  
  padding: 20px;  
  font-size: 30px;  
}
```

```
<body>  
  <div class="flex-container bg-blue">  
    <div>1</div>  
    <div>2</div>  
    <div>3</div>  
    <div>4</div>  
    <div>5</div>  
    <div>6</div>  
    <div>7</div>  
    <div>8</div>  
  </div>
```



FLEXBOX CONTAINER

- Flex direction
- ทดลองเปลี่ยนค่า direction เป็น

row-reverse

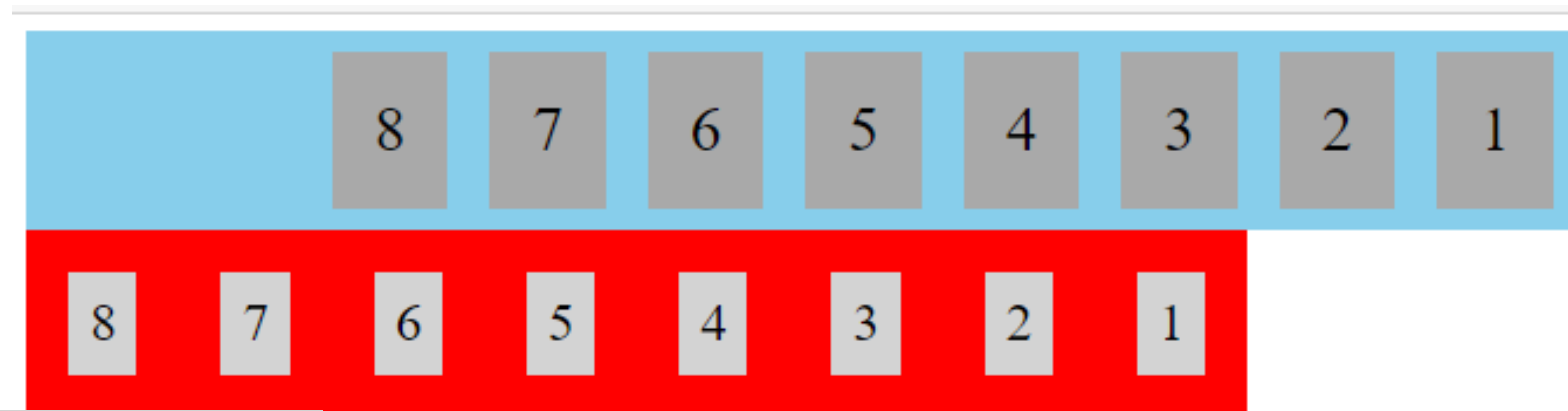
column

column-reverse

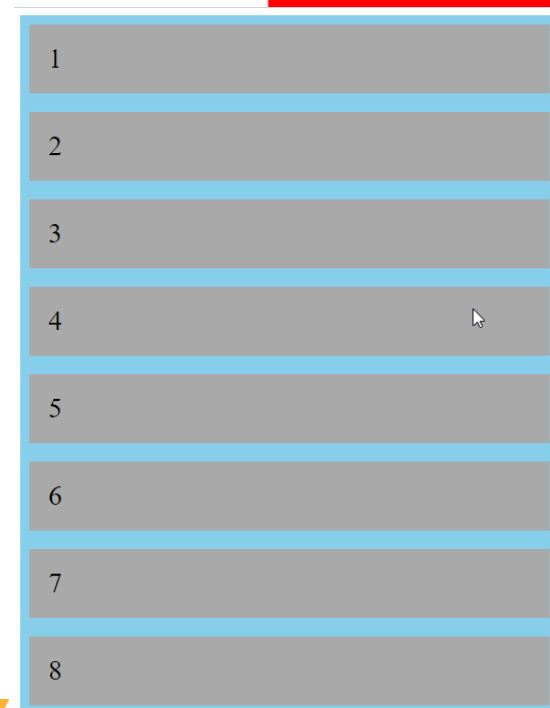
FLEXBOX CONTAINER

- Flex direction
- ทดลองเปลี่ยนค่า direction เป็น

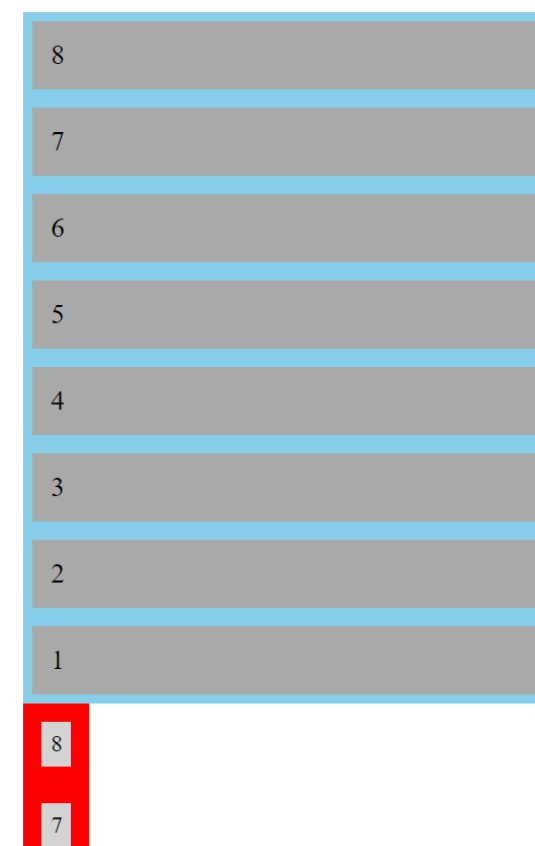
row-reverse



column



column-reverse



FLEXBOX CONTAINER

- Flex wrap จะทำอะไร ถ้าขนาดของ item มันยาวเกินไป
- แก้ไข css ให้กำหนดความยาวของ element ภายใน

```
.flex-container {  
  display: flex;  
  flex-direction: row;  
}  
  
.inline-flex-container {  
  display: inline-flex;  
  flex-direction: row;  
}  
  
.bg-red {  
  background-color: red;  
}  
  
.bg-blue {  
  background-color: skyblue;  
}  
  
.flex-container>div {  
  background-color: darkgray;  
  margin: 10px;  
  padding: 20px;  
  font-size: 30px;  
  width: 70px  
}  
  
.inline-flex-container>div {  
  background-color: lightgray;  
  margin: 20px;  
  padding: 10px;  
  font-size: 25px;  
  width: 70px  
}
```

- ปรับ flex-direction กลับเป็น row
- ทดลองย่อ ขยาย brosw

FLEXBOX CONTAINER

- เพิ่ม flex wrap ลงไป

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

- ทดลองย่อขยาย หน้าจออีกครั้ง
- ลองเปลี่ยนค่าเป็น wrap-reverse

FLEXBOX CONTAINER

- ถ้าต้องการใช้ ทั้ง direction และ wrap ในบันทึกเดียว สามารถใช้คำว่า flex-flow ได้

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

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

FLEXBOX CONTAINER

- flex justify-content
- ทดลอง ลด ปริมาณ item ก่อน

```
<body>
  <div class="flex-container bg-blue">
    <div>1</div>
    <div>2</div>
    <div>3</div>
  </div>
  <div class="inline-flex-container bg-red">
    <div>1</div>
    <div>2</div>
    <div>3</div>
  </div>
</body>
```

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

.inline-flex-container {
  display: inline-flex;
  flex-flow: row wrap;
  justify-content: flex-start;
}
```

FLEXBOX CONTAINER

- flex justify-content
- ทดลอง justify content ค่าอื่น เพื่อผลลัพธ์

```
<body>
  <div class="flex-container bg-blue">
    <div>1</div>
    <div>2</div>
    <div>3</div>
  </div>
  <div class="inline-flex-container bg-red">
    <div>1</div>
    <div>2</div>
    <div>3</div>
  </div>
</body>
```

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

.inline-flex-container {
  display: inline-flex;
  flex-flow: row wrap;
  justify-content: flex-start;
}
```

FLEXBOX CONTAINER

- ว่างๆ ทดลองส่วน align-items, align-content
- แต่ว่าทั้งสองอันนี้จะมีประโยชน์ถ้า item มีขนาดไม่เท่ากัน และ container มีขนาดใหญ่

FLEXBOX ITEM

- ระบุสถานะในการจัดการ item
- css ที่ระบุจะใช้ได้ก็เมื่อ component นี้อยู่ภายใน element ที่มี stylesheet display: flex, inline flex

FLEXBOX ITEM

- ทดลองเพิ่ม style order

```
.d1 {  
  order: 1;  
}  
  
.d2 {  
  order: 3;  
}  
  
.d3 {  
  order: 2;  
}
```

```
<body>  
  <div class="flex-container bg-blue">  
    <div class="d1">1</div>  
    <div class="d2">2</div>  
    <div class="d3">3</div>  
  
  </div>  
  <div class="inline-flex-container bg-red">  
    <div class="d1">1</div>  
    <div class="d2">2</div>  
    <div class="d3">3</div>  
  </div>  
</body>
```

FLEXBOX ITEM

- ทดลองเพิ่ม style order

```
.d1 {  
  order: 1;  
}  
  
.d2 {  
  order: 3;  
}  
  
.d3 {  
  order: 2;  
}
```

```
<body>  
  <div class="flex-container bg-blue">  
    <div class="d1">1</div>  
    <div class="d2">2</div>  
    <div class="d3">3</div>  
  
  </div>  
  <div class="inline-flex-container bg-red">  
    <div class="d1">1</div>  
    <div class="d2">2</div>  
    <div class="d3">3</div>  
  </div>  
</body>
```



FLEXBOX ITEM

- flex-grow ขยายเป็นสัดส่วน

```
.d1 {  
  order: 1;  
}  
  
.d2 {  
  order: 3;  
  flex-grow: 3;  
}  
  
.d3 {  
  order: 2;  
}
```

FLEXBOX ITEM

- flex-grow ขยายเป็นสัดส่วน

```
.d1 {  
  order: 1;  
}  
  
.d2 {  
  order: 3;  
}  
  
.d3 {  
  order: 2;  
  flex-grow: 1;  
}
```

FLEXBOX ITEM

- flex-grow ขยายเป็นสัดส่วน

```
.d1 {  
  order: 1;  
}  
  
.d2 {  
  order: 3;  
}  
  
.d3 {  
  order: 2;  
  flex-grow: 1;  
}
```



FLEXBOX ITEM

- flex-grow ขยายเป็นสัดส่วน

```
.d1 {  
  order: 1;  
}  
  
.d2 {  
  order: 3;  
  flex-grow: 1;  
}  
  
.d3 {  
  order: 2;  
  flex-grow: 1;  
}
```

- ลองปรับค่าแล้วดูผล

FLEXBOX ITEM

- เพิ่ม flex-shrink การย่อขนาดของ element ถ้าจำเป็น เมื่อเทียบกับ element อื่น
- ใช้ได้กับ flex flow no-wrap

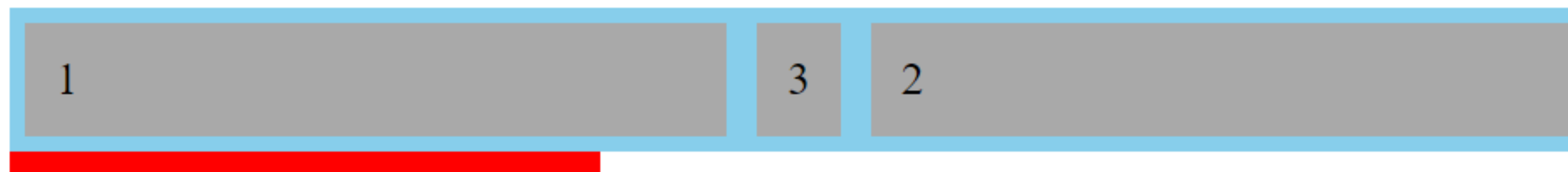
```
.flex-container {  
  display: flex;  
  flex-flow: row no-wrap;  
}
```

```
.d1 {  
  order: 1;  
}
```

```
.d2 {  
  order: 3;  
}
```

```
.d3 {  
  order: 2;  
  flex-shrink: 4;  
}
```

```
.flex-container>div {  
  background-color: darkgray;  
  margin: 10px;  
  padding: 20px;  
  font-size: 30px;  
  width: 600px;  
}
```



FLEXBOX ITEM

- flex-basis ระบุขนาดเริ่มต้นของแต่ละ item

```
.d1 {  
  order: 1;  
  flex-basis: 200px;  
}  
  
.d2 {  
  order: 3;  
  flex-basis: 100px;  
}  
  
.d3 {  
  order: 2;  
  flex-basis: 100px;  
}
```


FLEXBOX ITEM

- flex เป็นการย่อ flex-grow flex-shrink และ flex-basis

flex: 0 0 200px

not growable (0)

not shrinkable (0)

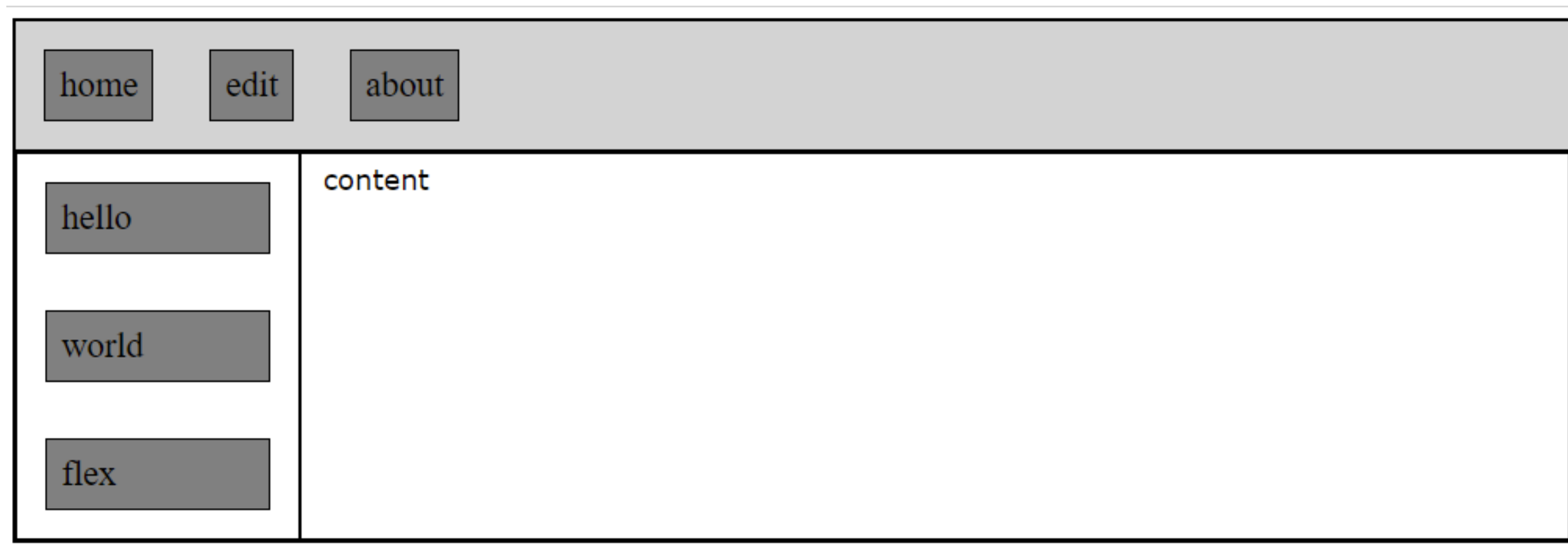
initial length 200px

FLEXBOX ITEM

- align-self
- ใช้กำหนด align ของ item เอง
- override ที่กำหนดโดย container

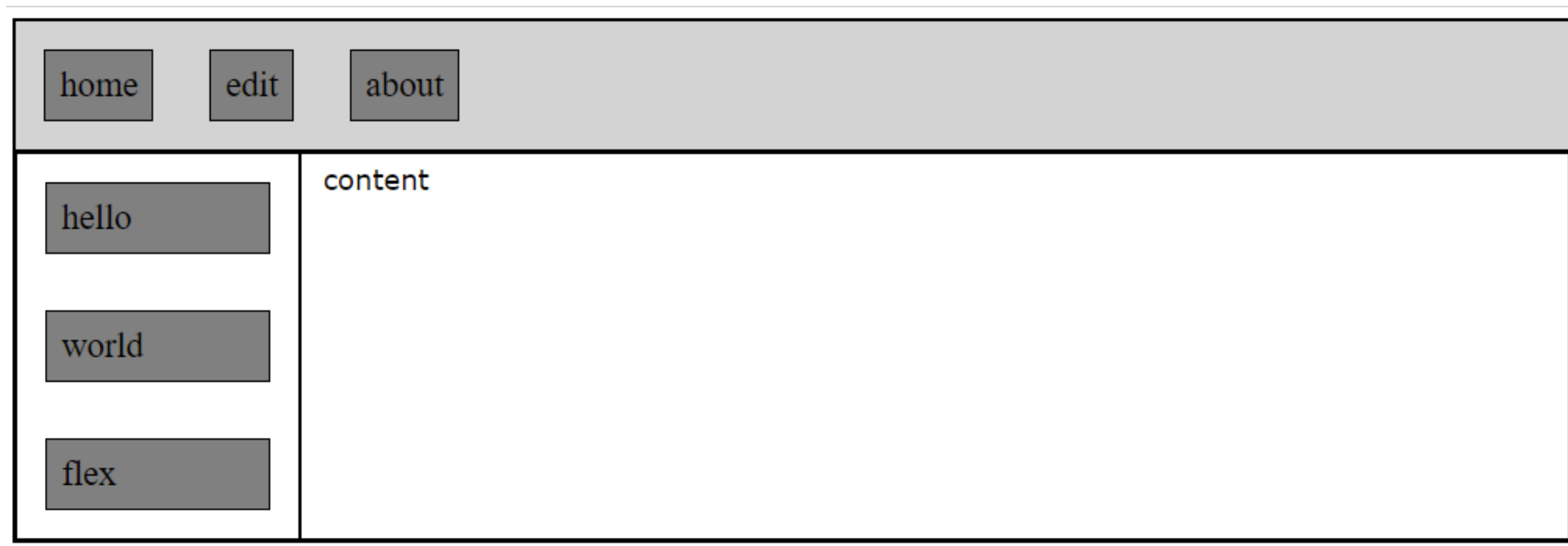
NEST FLEX COMPONENT

- สามารถนำ item มาซ้อน item ได้
- เพื่อความยืดหยุ่นในการออกแบบ
- ถ้าหน้าจอ เมนูแบบนี้ จะออกแบบเป็นชั้นๆ อย่างไร



NEST FLEX COMPONENT

- Act 1 ทดลองเขียน code เพื่อให้ได้หน้าตาแบบนี้





NEST FLEX COMPONENT

- Act 2 ลองปรับหน้าต่างของ เว็บร้านขายดอกไม้เพื่อให้ได้หน้าต่างแบบข้อเมื่อที่