

## ■ Layout의 발전

### 1. Table

- 정보로써 사용되는 격자 형식의 요소를 요소 배치의 용도로 사용
- 배치된 요소를 변경하려는 경우 어려움

```
<table width="800" cellpadding="0" cellspacing="0" border="0" align="center">
<tr bgcolor="#8080FF" align="center">
  <td width="100">홈페이지 로고</td>
  <td width="100">메뉴1</td>
  <td width="100">메뉴2</td>
  <td width="100">메뉴3</td>
  <td width="100">메뉴4</td>
  <td width="100">메뉴5</td>
  <td width="200"><!-- 여백 --></td>
</tr>
<tr height="5">
  <td colspan="7"><!-- 여백 --></td>
</tr>
</table>
```

홈페이지 로고	메뉴1	메뉴2	메뉴3	메뉴4	메뉴5	
하위메뉴1	여기에 본문이 옵니다.					
하위메뉴2						
하위메뉴3						
하위메뉴4						
하위메뉴5						

```
<table width="800" cellpadding="0" cellspacing="0" border="0" align="center">
<tr>
  <td width="150" valign="top" bgcolor="#8080FF">
    <table width="130" cellpadding="5" cellspacing="0" border="0" align="center">
```

참고 : <http://www.homejjang.com/05/layout.php>

## ■ Layout의 발전

### 2. Position

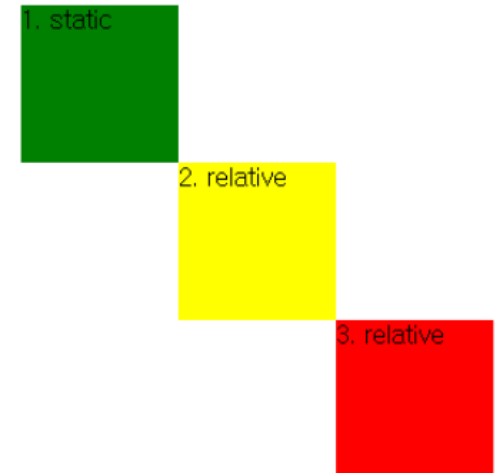
- static / relative / absolute / fixed
- 현재까지도 자주 사용되는 방식

```
#static
{
width: 100px;
height: 100px;
background-color: green;
position: static;
}
```

```
#relative2
{
width: 100px;
height: 100px;
background-color: yellow;
position: relative;
top: 0px;
left: 100px;
}
```

```
#relative3
{
width: 100px;
height: 100px;
background-color: red;
position: relative;
top: 0px;
left: 200px;
}
```

```
<div id="static">
1. static
</div>
<div id="relative2">
2. relative
</div>
<div id="relative3">
3. relative
</div>
```



참고 : <https://abooks.tistory.com/82>

## ■ Layout의 발전

### 3. Float

- left / right 등을 이용하여 글자와 이미지를 적절하게 배치
- 속성 사용이 까다로움

```
<style type="text/css">
```

```
img
```

```
{
```

```
float:left;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```

```

```
동해물과 백두산이 마르고 닳도록 하느님이 보우하사
```

```
(후렴) 무궁화 삼천리 화려강산 대한 사람, 대한으로 길이
```

```
남산 위에 저 소나무, 철갑을 두른 듯 바람서리 불변함은
```

```
(후렴) 무궁화 삼천리 화려강산 대한 사람, 대한으로 길이 보전하세
```

```
가을 하늘 공활한데 높고 구름 없이 밝은 달은 우리 가슴 일편단심일세,
```

```
(후렴) 무궁화 삼천리 화려강산 대한 사람, 대한으로 길이 보전하세
```

```
이 기상과 이 맘으로 충성을 다하여 괴로우나 즐거우나 나라 사랑하세,
```

```
(후렴) 무궁화 삼천리 화려강산 대한 사람, 대한으로 길이 보전하세
```

```
</body>
```

```
</html>
```



동해물과 백두산이 마르고 닳도록 하느님이  
보우하사 우리나라 만세. (후렴) 무궁화 삼천  
리 화려강산 대한 사람, 대한으로 길이 보전  
하세 남산 위에 저 소나무, 철갑을 두른 듯 바  
람서리 불변함은 우리 기상일세. (후렴) 무궁  
화 삼천리 화려강산 대한 사람, 대한으로 길  
이 보전하세 가을 하늘 공활한데 높고 구름  
없이 밝은 달은 우리 가슴 일편단심일세. (후  
렴) 무궁화 삼천리 화려강산 대한 사람, 대한  
으로 길이 보전하세 이 기상과 이 맘으로 충  
성을 다하여 괴로우나 즐거우나 나라 사랑하  
세. (후렴) 무궁화 삼천리 화려강산 대한 사

람, 대한으로 길이 보전하세

참고 : <https://aboooks.tistory.com/72>

## ■ Flex

- 유연한 반응형 레이아웃 구조를 쉽게 설계 할 수 있도록 구성
- Flex를 적용하기 위한 기본 구조

```
<parent>  
  <child></child>  
  <child></child>  
  <child></child>  
</parent>
```

```
<container>  
  <item></item>  
  <item></item>  
  <item></item>  
</container>
```

- 사용되는 속성

Container	Item
display	order
flex-direction	flex-grow
flex-wrap	flex-shrink
flex-flow	flex-basis
justify-content	flex
align-items	align-self
align-content	

## ■ Flex

### ● 기본 적용

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .flex-container {
      display: flex;
      background-color: lightgrey;
    }
    .flex-item {
      background-color: lightcyan;
      border: 1px solid red;
    }
  </style>
</head>
<body>
  <div class="flex-container">
    <div class="flex-item">1</div>
    <div class="flex-item">2</div>
    <div class="flex-item">3</div>
  </div>
</body>
</html>
```

Container	Item
<b>display</b>	order
flex-direction	flex-grow
flex-wrap	flex-shrink
flex-flow	flex-basis
justify-content	flex
align-items	align-self
align-content	



## flex-direction

### ● 방향 지정 (row/column/reverse)

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .flex-container {
      display: flex;
      background-color: lightgrey;
      flex-direction: column;
    }
    .flex-item {
      background-color: lightcyan;
      border: 1px solid red;
    }
  </style>
</head>
<body>
  <div class="flex-container">
    <div class="flex-item">1</div>
    <div class="flex-item">2</div>
    <div class="flex-item">3</div>
  </div>
</body>
</html>
```

Container	Item
display	order
<b>flex-direction</b>	flex-grow
flex-wrap	flex-shrink
flex-flow	flex-basis
justify-content	flex
align-items	align-self
align-content	

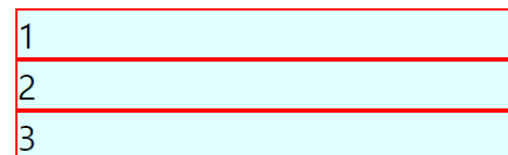
#### row (기본)



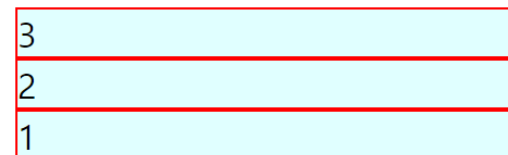
#### row-reverse



#### column



#### column-reverse



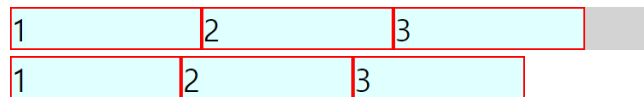
## ■ flex-wrap

### ● 요소 감싸기 (nowrap/wrap/reverse)

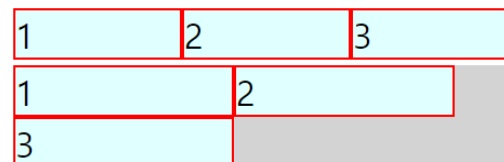
```
<!DOCTYPE html>
<html>
<head>
  <style>
    .flex-container {
      display: flex;
      background-color: lightgrey;
      flex-wrap: wrap;
    }
    .flex-item {
      width: 100px;
      background-color: lightcyan;
      border: 1px solid red;
    }
  </style>
</head>
<body>
  <div class="flex-container">
    <div class="flex-item">1</div>
    <div class="flex-item">2</div>
    <div class="flex-item">3</div>
  </div>
</body>
</html>
```

Container	Item
display	order
flex-direction	flex-grow
<b>flex-wrap</b>	flex-shrink
flex-flow	flex-basis
justify-content	flex
align-items	align-self
align-content	

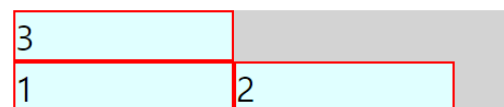
#### flex-wrap 속성 사용 전



#### wrap



#### wrap-reverse



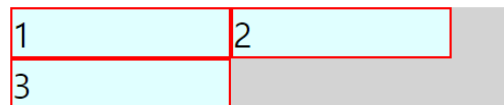
## flex-flow

### ● 방향 지정 + 요소 감싸기

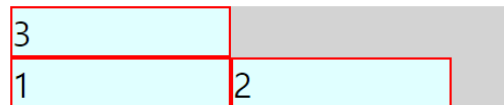
```
<!DOCTYPE html>
<html>
<head>
  <style>
    .flex-container {
      display: flex;
      background-color: lightgrey;
      flex-flow: row wrap;
    }
    .flex-item {
      width: 100px;
      background-color: lightcyan;
      border: 1px solid red;
    }
  </style>
</head>
<body>
  <div class="flex-container">
    <div class="flex-item">1</div>
    <div class="flex-item">2</div>
    <div class="flex-item">3</div>
  </div>
</body>
</html>
```

Container	Item
display	order
flex-direction	flex-grow
flex-wrap	flex-shrink
<b>flex-flow</b>	flex-basis
justify-content	flex
align-items	align-self
align-content	

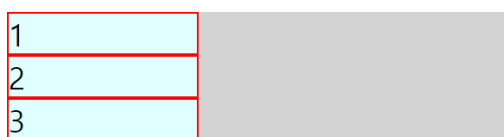
#### row + wrap



#### row + wrap-reverse



#### column + wrap



#### column + wrap-reverse



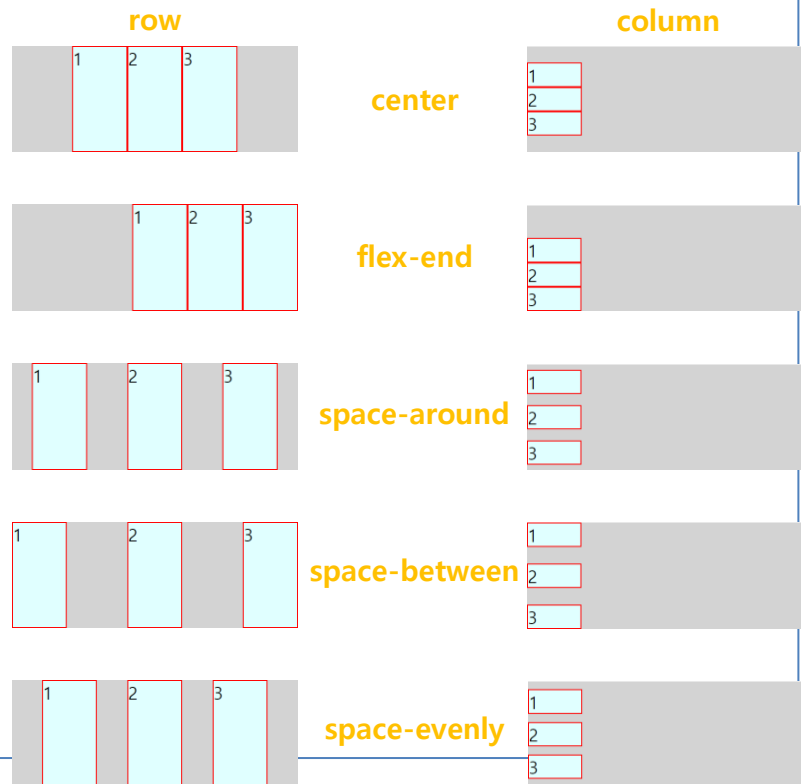


## ■ justify-content

### ● 요소 정렬 (center/end/around/between)

```
<head>
  <style>
    .flex-container {
      display: flex;
      height: 100px;
      background-color: lightgrey;
      flex-direction: row;
      justify-content: center;
    }
    .flex-item {
      width: 50px;
      background-color: lightcyan;
      border: 1px solid red;
    }
  </style>
</head>
<body>
  <div class="flex-container">
    <div class="flex-item">1</div>
    <div class="flex-item">2</div>
    <div class="flex-item">3</div>
  </div>
</body>
```

Container	Item
display	order
flex-direction	flex-grow
flex-wrap	flex-shrink
flex-flow	flex-basis
<b>justify-content</b>	flex
align-items	align-self
align-content	

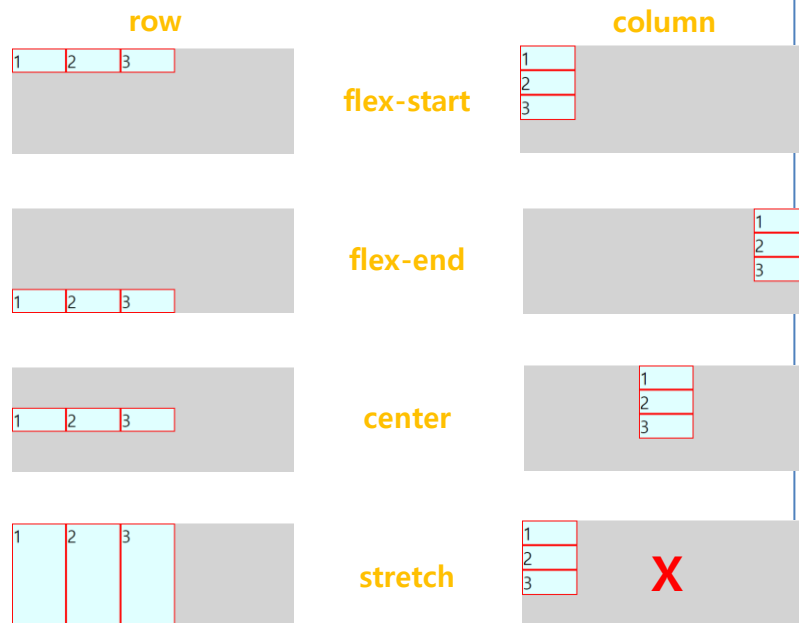


## ■ align-items

### ● 요소 정렬 (center/end/stretch)

```
<head>
  <style>
    .flex-container {
      display: flex;
      height: 100px;
      background-color: lightgrey;
      flex-direction: row;
      align-items: center;
    }
    .flex-item {
      width: 50px;
      background-color: lightcyan;
      border: 1px solid red;
    }
  </style>
</head>
<body>
  <div class="flex-container">
    <div class="flex-item">1</div>
    <div class="flex-item">2</div>
    <div class="flex-item">3</div>
  </div>
</body>
```

Container	Item
display	order
flex-direction	flex-grow
flex-wrap	flex-shrink
flex-flow	flex-basis
justify-content	flex
<b>align-items</b>	align-self
align-content	



## ■ align-content

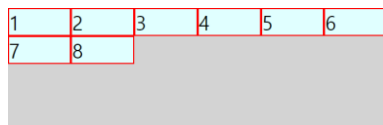
### ● 라인 정렬 (center/around/between)

```
<style>
  .flex-container {
    display: flex;
    height: 100px;
    background-color: lightgrey;
    flex-wrap: wrap;
    align-content: center;
  }
  .flex-item {
    width: 50px;
    background-color: lightcyan;
    border: 1px solid red;
  }
</style>

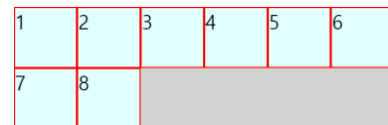
<div class="flex-container">
  <div class="flex-item">1</div>
  <div class="flex-item">2</div>
  <div class="flex-item">3</div>
  <div class="flex-item">4</div>
  <div class="flex-item">5</div>
  <div class="flex-item">6</div>
  <div class="flex-item">7</div>
  <div class="flex-item">8</div>
</div>
```

Container	Item
display	order
flex-direction	flex-grow
flex-wrap	flex-shrink
flex-flow	flex-basis
justify-content	flex
align-items	align-self
<b>align-content</b>	

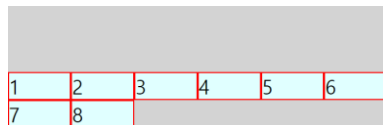
flex-start



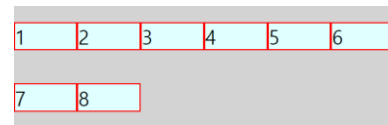
stretch



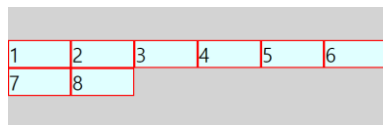
flex-end



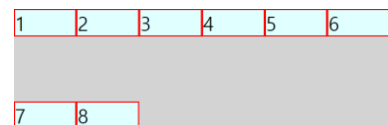
space-around



center



space-between



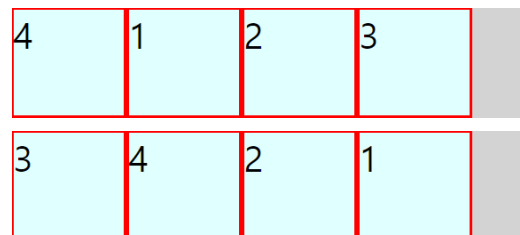
## ■ order

### ● 요소 순서

```
<head>
  <style>
    .flex-container {
      display: flex;
      height: 50px;
      background-color: lightgrey;
    }
    .flex-item {
      width: 50px;
      background-color: lightcyan;
      border: 1px solid red;
    }
    .flex-item:nth-child(1) { order: 2 }
    .flex-item:nth-child(2) { order: 3 }
    .flex-item:nth-child(3) { order: 4 }
    .flex-item:nth-child(4) { order: 1 }
  </style>
</head>
<body>
  <div class="flex-container">
    <div class="flex-item">1</div><div class="flex-item">2</div>
    <div class="flex-item">3</div><div class="flex-item">4</div>
  </div>
</body>
```

Container	Item
display	<b>order</b>
flex-direction	flex-grow
flex-wrap	flex-shrink
flex-flow	flex-basis
justify-content	flex
align-items	align-self
align-content	

지정하는 순서가 중복되면 → 방향으로 밀림

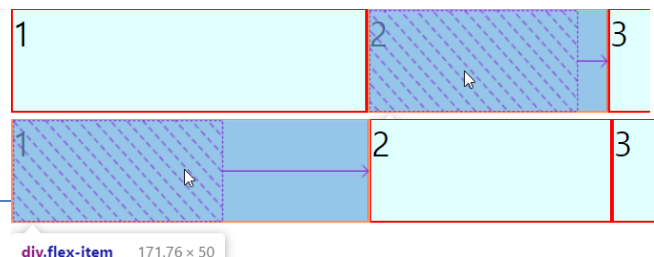
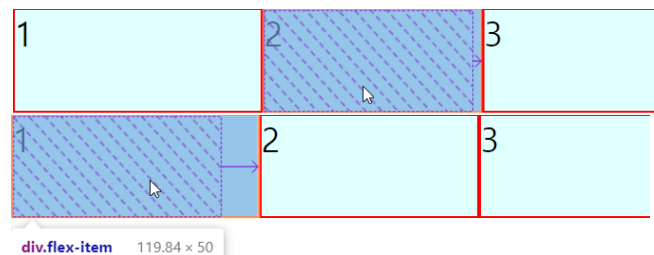


## flex-grow

### ● 요소 크기 증가 비중 (속도)

```
<head>
  <style>
    .flex-container {
      display: flex;
      height: 50px;
      background-color: lightgrey;
    }
    .flex-item {
      width: 50px;
      background-color: lightcyan;
      border: 1px solid red;
      flex-grow: 1;
    }
    .flex-item:nth-child(1) { flex-grow: 5 }
  </style>
</head>
<body>
  <div class="flex-container">
    <div class="flex-item">1</div>
    <div class="flex-item">2</div>
    <div class="flex-item">3</div>
  </div>
</body>
```

Container	Item
display	order
flex-direction	<b>flex-grow</b>
flex-wrap	flex-shrink
flex-flow	flex-basis
justify-content	flex
align-items	align-self
align-content	

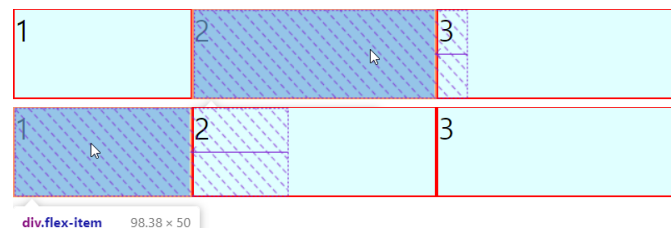


## flex-shrink

### ● 요소 크기 감소 비중 (속도)

```
<head>
  <style>
    .flex-container {
      display: flex;
      height: 50px;
      background-color: lightgrey;
    }
    .flex-item {
      width: 150px;
      background-color: lightcyan;
      border: 1px solid red;
      flex-shrink: 1;
    }
    .flex-item:nth-child(1) { flex-shrink: 3; }
  </style>
</head>
<body>
  <div class="flex-container">
    <div class="flex-item">1</div>
    <div class="flex-item">2</div>
    <div class="flex-item">3</div>
  </div>
</body>
```

Container	Item
display	order
flex-direction	flex-grow
flex-wrap	<b>flex-shrink</b>
flex-flow	flex-basis
justify-content	flex
align-items	align-self
align-content	

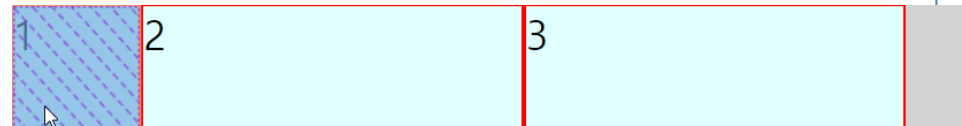


## flex-basis

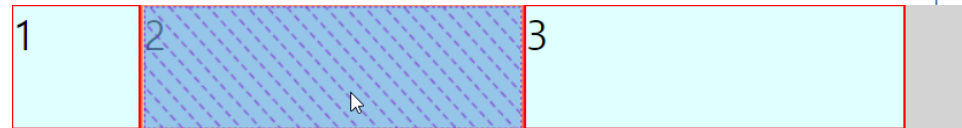
### ● 요소의 기본 크기 (center/around/between)

```
<head>
  <style>
    .flex-container {
      display: flex;
      height: 50px;
      background-color: lightgrey;
    }
    .flex-item {
      flex-basis: 150px;
      background-color: lightcyan;
      border: 1px solid red;
    }
    .flex-item:nth-child(1) {
      flex-basis: 50px;
    }
  </style>
</head>
<body>
  <div class="flex-container">
    <div class="flex-item">1</div>
    <div class="flex-item">2</div>
    <div class="flex-item">3</div>
  </div>
</body>
```

Container	Item
display	order
flex-direction	flex-grow
flex-wrap	flex-shrink
flex-flow	<b>flex-basis</b>
justify-content	flex
align-items	align-self
align-content	



div.flex-item 52 × 50



div.flex-item 152 × 50

## ■ align-self

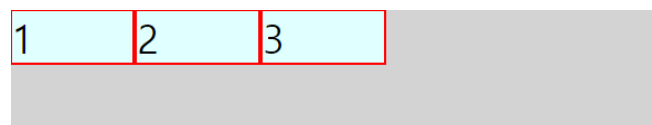
### ● 요소 정렬 (align-items 속성값 재정의)

```
<head>
  <style>
    .flex-container {
      display: flex;
      height: 50px;
      background-color: lightgrey;
      flex-direction: row;
      align-items: center;
    }
    .flex-item {
      width: 50px;
      background-color: lightcyan;
      border: 1px solid red;
      align-self: flex-end;
    }
  </style>
</head>
<body>
  <div class="flex-container">
    <div class="flex-item">1</div>
    <div class="flex-item">2</div>
    <div class="flex-item">3</div>
  </div>
</body>
```

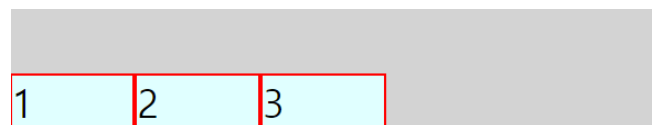
Container	Item
display	order
flex-direction	flex-grow
flex-wrap	flex-shrink
flex-flow	flex-basis
justify-content	flex
align-items	<b>align-self</b>
align-content	



flex-start



flex-end



stretch

