

CSS Flexbox

CSS Flexbox 레이아웃 모듈

Flexbox 레이아웃 모듈 이전에는 4 가지 레이아웃 모드가 있었습니다.

- 웹 페이지의 섹션에 대한 차단
- 텍스트 인라인
- 2 차원 테이블 데이터의 테이블
- 요소의 명시 적 위치를 위해 배치

플렉시블 박스 레이아웃 모듈은 플롯 또는 위치 지정 없이 유연한 반응 형 레이아웃 구조를 보다 쉽게 설계 할 수 있습니다.

브라우저 지원

flexbox 속성은 모든 최신 브라우저에서 지원됩니다.

29.0	11.0	22.0	10	48

Flexbox 요소

Flexbox 모델을 사용하려면 먼저 Flex 컨테이너를 정의해야 합니다.

3 개의 플렉스 아이템이있는 플렉스 컨테이너 :

```
<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  background-color: DodgerBlue;
```

```

}

.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  padding: 20px;
  font-size: 30px;
}
</style>
</head>
<body>
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>

<p>A Flexible Layout must have a parent element with the display property set to flex.</p>

<p>Direct child elements(s) of the flexible container automatically becomes flexible items.</p>

</body>
</html>

```

부모 요소 (컨테이너)

`display` 속성을 *flex* 로 설정하면 플렉스 컨테이너가 유연 해집니다.

```

.flex-container {
  display: flex;
}

```

플렉스 컨테이너 속성은 다음과 같습니다.

- [flex-direction](#)
- [flex-wrap](#)
- [flex-flow](#)
- [justify-content](#)
- [align-items](#)
- [align-content](#)

플렉스 방향 속성

이 `flex-direction` 속성은 컨테이너가 플렉스 항목을 쌓을 방향을 정의합니다.

열 값 (위에서 아래로) 수직 플렉스 항목 스택 :

```
<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  flex-direction: column;
}
.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  width: 100px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>
<h1>The flex-direction Property</h1>

<p>The "flex-direction: column;" stacks the flex items vertically (from top to bottom):</p>

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

</body>
</html>
```

컬럼 반전 값 (그러나 바닥에서 상단으로) 수직 플렉스 항목 스택 :

```
<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  flex-direction: column-reverse;
}
.flex-container > div {
  background-color: #f1f1f1;
}
```

```

margin: 10px;
width: 100px;
text-align: center;
line-height: 75px;
font-size: 30px;
}
</style>
</head>
<body>
<h1>The flex-direction Property</h1>

<p>The "flex-direction: column-reverse;" stacks the flex items vertically (but
from bottom to top):</p>

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

</body>
</html>

```

로우 값 (왼쪽에서 오른쪽으로) 수평 플렉스 항목 스택 :

```

<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  flex-direction: row;
  background-color: DodgerBlue;
}
.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  width: 100px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>
<h1>The flex-direction Property</h1>

<p>The "flex-direction: row;" stacks the flex items horizontally (from left to
right):</p>

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

```

```
</body>
</html>
```

행 반전 값 (단, 오른쪽에서 왼쪽으로) 수평 플렉스 항목 스택 :

```
<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  flex-direction: row-reverse;
  background-color: DodgerBlue;
}
.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  width: 100px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>
<h1>The flex-direction Property</h1>

<p>The "flex-direction: row-reverse;" stacks the flex items horizontally (but
from right to left):</p>

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

</body>
</html>
```

플렉스 랩 속성

이 `flex-wrap` 속성은 플렉스 항목을 감쌀 지 여부를 지정합니다.

아래 예제에는 12 가지 플렉스 항목이 있어 `flex-wrap` 속성을 보다 잘 보여줍니다.

랩 값이 필요한 경우 플렉스 항목을 포장하도록 지정합니다 :

```
<!DOCTYPE html>
```

```

<html>
<head>
<style>
.flex-container {
  display: flex;
  flex-wrap: wrap;
  background-color: DodgerBlue;
}
.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  width: 100px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>
<h1>The flex-wrap Property</h1>

<p>The "flex-wrap: wrap;" specifies that the flex items will wrap if
necessary:</p>

<div class="flex-container">
  <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>9</div>
  <div>10</div>
  <div>11</div>
  <div>12</div>
</div>

<p>Try resizing the browser window.</p>

</body>
</html>

```

*파라미터 nowrap*의 값은 플렉스 항목을 포장하지 않도록 지정합니다 (이 기본값) :

```

<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  flex-wrap: nowrap;
  background-color: DodgerBlue;
}

```

```

.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  width: 100px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>
<h1>The flex-wrap Property</h1>

<p>The "flex-wrap: nowrap;" specifies that the flex items will not wrap (this is
default):</p>

<div class="flex-container">
  <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>9</div>
  <div>10</div>
  <div>11</div>
  <div>12</div>
</div>

<p>Try resizing the browser window.</p>

</body>
</html>

```

랩 역방향 값은 필요한 경우가요 항목 역순 감싸도록 지정 :

```

<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  flex-wrap: wrap-reverse;
  background-color: DodgerBlue;
}
.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  width: 100px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}

```

```

</style>
</head>
<body>
<h1>The flex-wrap Property</h1>

<p>The "flex-wrap: wrap-reverse;" specifies that the flex items will wrap if
necessary, in reverse order:</p>

<div class="flex-container">
  <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>9</div>
  <div>10</div>
  <div>11</div>
  <div>12</div>
</div>

<p>Try resizing the browser window.</p>

</body>
</html>

```

플렉스 흐름 속성

이 **flex-flow** 속성은 **flex-direction** 및 **flex-wrap** 속성을 설정하기 위한 속기 속성입니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  flex-flow: row wrap;
  background-color: DodgerBlue;
}
.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  width: 100px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>

```



```

<body>
<h1>The flex-flow Property</h1>

<p>The flex-flow property is a shorthand property for the flex-direction and the
flex-wrap properties.</p>

<div class="flex-container">
  <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>9</div>
  <div>10</div>
  <div>11</div>
  <div>12</div>
</div>

<p>Try resizing the browser window.</p>

</body>
</html>

```

정당화 컨텐츠 속성

이 **justify-content** 속성은 플렉스 항목을 정렬하는 데 사용됩니다.

센터/값은 컨테이너의 중심에있는 플렉스 항목을 정렬 :

```

<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  justify-content: center;
  background-color: DodgerBlue;
}
.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  width: 100px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>

```

```

<h1>The justify-content Property</h1>

<p>The "justify-content: center;" aligns the flex items at the center of the container:</p>

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

</body>
</html>

```

플렉스 시작 값은 컨테이너의 시작 부분에 플렉스 항목을 (이 기본값) 정렬 :

```

<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  justify-content: flex-start;
  background-color: DodgerBlue;
}
.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  width: 100px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>
<h1>The justify-content Property</h1>

<p>The "justify-content: flex-start;" aligns the flex items at the beginning of the container (this is default):</p>

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

</body>
</html>

```

플렉스 엔드 값은 용기의 끝 부분에있는 플렉스 항목을 정렬 :

```

<!DOCTYPE html>

```

```

<html>
<head>
<style>
.flex-container {
  display: flex;
  justify-content: flex-end;
  background-color: DodgerBlue;
}
.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  width: 100px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>
<h1>The justify-content Property</h1>

<p>The "justify-content: flex-end;" aligns the flex items at the end of the
container:</p>

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

</body>
</html>

```

공간 주변 값 사이, 전 공간 플렉스 항목을 표시하고 선 후 :

```

<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  justify-content: space-around;
  background-color: DodgerBlue;
}
.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  width: 100px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>

```

```

<h1>The justify-content Property</h1>

<p>The "justify-content: space-around;" displays the flex items with space
before, between, and after the lines:</p>

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

</body>
</html>

```

공간 사이의 값은 선 사이의 공간 플렉스 항목을 표시합니다 :

```

<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  justify-content: space-between;
  background-color: DodgerBlue;
}
.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  width: 100px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>
<h1>The justify-content Property</h1>

<p>The "justify-content: space-between;" displays the flex items with space
between the lines:</p>

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

</body>
</html>

```

align-items 속성

이 `align-items` 속성은 플렉스 항목을 세로로 정렬하는 데 사용됩니다.

이 예제에서는 `align-items` 속성을 더 잘 보여주기 위해 200 픽셀 높이의 컨테이너를 사용합니다.

센터/값은 용기의 중간에 플렉스 항목을 정렬 :

```
<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  height: 200px;
  align-items: center;
  background-color: DodgerBlue;
}
.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  width: 100px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>
<h1>The align-items Property</h1>

<p> The "align-items: center;" aligns the flex items in the middle of the
container:</p>

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

</body>
</html>
```

플렉스 시작/값은 용기의 상단에있는 플렉스 항목을 정렬 :

```
<!DOCTYPE html>
<html>
<head>
```

```

<style>
.flex-container {
  display: flex;
  height: 200px;
  align-items: flex-start;
  background-color: DodgerBlue;
}
.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  width: 100px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>
<h1>The align-items Property</h1>

<p> The "align-items: flex-start;" aligns the flex items at the top of the
container:</p>

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

</body>
</html>

```

플렉스 최/종 값은 용기의 바닥에 플렉스 항목 정렬 :

```

<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  height: 200px;
  align-items: flex-end;
  background-color: DodgerBlue;
}
.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  width: 100px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>

```

```

<h1>The align-items Property</h1>

<p> The "align-items: flex-end;" aligns the flex items at the bottom of the container:</p>

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

</body>
</html>

```

스트레치 값 (이 기본값) 컨테이너를 채우기 위해 플렉스 항목을 뻗어 :

```

<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  height: 200px;
  align-items: stretch;
  background-color: DodgerBlue;
}
.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  width: 100px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>
<h1>The align-items Property</h1>

<p> The "align-items: stretch;" stretches the flex items to fill the container (this is default):</p>

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

</body>
</html>

```

기준선 값 은 기준선 과 같은 플렉스 항목을 정렬합니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  height: 200px;
  align-items: baseline;
  background-color: DodgerBlue;
}
.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  width: 100px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>
<h1>The align-items Property</h1>

<p> The "align-items: stretch;" stretches the flex items to fill the container
(this is default):</p>

<div class="flex-container">
  <div><h1>1</h1></div>
  <div><h6>2</h6></div>
  <div><h3>3</h3></div>
  <div><small>4</small></div>
</div>

</body>
</html>

```

참고 : 이 예제에서는 다른 글꼴 크기를 사용하여 텍스트 기준선에 따라 항목이 정렬됨을 보여줍니다.

정렬 내용 속성

이 **align-content** 속성은 플렉스 라인을 정렬하는 데 사용됩니다.

이 예제에서는 **align-content** 속성을 더 잘 보여주기 위해 **flex-wrap** 속성이 **wrap**으로 설정된 600 픽셀 높이의 컨테이너를 사용합니다.

공간 사이의 값은 그들 사이에 동일한 공간 플렉스 라인을 표시합니다 :


```

<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  height: 600px;
  flex-wrap: wrap;
  align-content: space-between;
  background-color: DodgerBlue;
}
.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  width: 100px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>
<h1>The align-content Property</h1>

<p>The "align-content: space-between;" displays the flex lines with equal space
between them:</p>

<div class="flex-container">
  <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>9</div>
  <div>10</div>
  <div>11</div>
  <div>12</div>
</div>

</body>
</html>

```

공간 주변 값 사이, 전 공간 플렉스 라인을 표시하고, 그 후 :

```

<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  height: 600px;
  flex-wrap: wrap;
  align-content: space-around;
}

```

```

    background-color: DodgerBlue;
}
.flex-container > div {
    background-color: #f1f1f1;
    margin: 10px;
    width: 100px;
    text-align: center;
    line-height: 75px;
    font-size: 30px;
}
</style>
</head>
<body>
<h1>The align-content Property</h1>

<p>The "align-content: space-around;" displays the flex lines with space before,
between, and after them:</p>

<div class="flex-container">
    <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>9</div>
    <div>10</div>
    <div>11</div>
    <div>12</div>
</div>

</body>
</html>

```

스트레치 값 (이 기본값) 남은 공간을 차지하기 위해 플렉스 라인을 뻗어 :

```

<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
    display: flex;
    height: 600px;
    flex-wrap: wrap;
    align-content: stretch;
    background-color: DodgerBlue;
}
.flex-container > div {
    background-color: #f1f1f1;
    margin: 10px;
    width: 100px;
    text-align: center;
    line-height: 75px;
    font-size: 30px;
}

```

```

}
</style>
</head>
<body>
<h1>The align-content Property</h1>

<p>The "align-content: stretch;" stretches the flex lines to take up the
remaining space (this is default):</p>

<div class="flex-container">
  <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>9</div>
  <div>10</div>
  <div>11</div>
  <div>12</div>
</div>

</body>
</html>

```

센터 값 표시는 용기의 중간에 굴곡 선을 표시 :

```

<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  height: 600px;
  flex-wrap: wrap;
  align-content: center;
  background-color: DodgerBlue;
}
.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  width: 100px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>
<h1>The align-content Property</h1>

<p>The "align-content: center;" displays the flex lines in the middle of the
container:</p>

```

```

<div class="flex-container">
  <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>9</div>
  <div>10</div>
  <div>11</div>
  <div>12</div>
</div>

</body>
</html>

```

플렉스 시작 값은 컨테이너의 시작 플렉스 라인을 표시합니다 :

```

<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  height: 600px;
  flex-wrap: wrap;
  align-content: flex-start;
  background-color: DodgerBlue;
}
.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  width: 100px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>
<h1>The align-content Property</h1>

<p>The "align-content: flex-start;" displays the flex lines at the start of the
container:</p>

<div class="flex-container">
  <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>9</div>
    <div>10</div>
    <div>11</div>
    <div>12</div>
  </div>

</body>
</html>

```

플렉스 엔드 값은 용기의 끝 부분에있는 플렉스 라인을 표시합니다 :

```

<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  height: 600px;
  flex-wrap: wrap;
  align-content: flex-end;
  background-color: DodgerBlue;
}
.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  width: 100px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>
<h1>The align-content Property</h1>

<p>The "align-content: flex-end;" displays the flex lines at the end of the
container:</p>

<div class="flex-container">
  <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>9</div>
  <div>10</div>
  <div>11</div>
  <div>12</div>
</div>

</body>
</html>

```

완벽한 센터링

다음 예제에서는 매우 일반적인 스타일 문제인 완벽한 센터링을 해결합니다.

해결 방법 : `justify-content` 및 `align-items` 속성을 *가운데*로 설정 하면 플렉스 항목이 완벽하게 가운데에 배치됩니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  height: 300px;
  justify-content: center;
  align-items: center;
  background-color: DodgerBlue;
}
.flex-container > div {
  background-color: #f1f1f1;
  color: white;
  width: 100px;
  height: 100px;
}
</style>
</head>
<body>
<h1>Perfect Centering</h1>

<p>A container with both the justify-content and the align-items properties set to center will align the item(s) in the center (in both axis).</p>

<div class="flex-container">
  <div></div>
</div>

</body>
</html>
```

자식 요소 (항목)

플렉스 컨테이너의 직접 하위 요소는 자동으로 가요 성 (**flex**) 항목이 됩니다.

위의 요소는 회색 플렉스 컨테이너 안에 있는 4 개의 파란색 플렉스 항목을 나타냅니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  background-color: #f1f1f1;
}
.flex-container > div {
  background-color: DodgerBlue;
  color: white;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>
<h1>Flexible Items</h1>

<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
</div>

<p> All direct children of a flexible container becomes flexible items.</p>

</body>
</html>

```

플렉스 아이템 속성은 다음과 같습니다.

- [order](#)
- [flex-grow](#)
- [flex-shrink](#)
- [flex-basis](#)
- [flex](#)
- [align-self](#)

주문 속성

이 **order** 속성은 flex 항목의 순서를 지정합니다.

코드의 첫 번째 플렉스 항목이 레이아웃에서 첫 번째 항목으로 표시 될 필요는 없습니다.

주문 값은 숫자 여야하며 기본값은 0 입니다.

주문 속성은 플렉스 항목의 순서를 변경할 수 있습니다 :

```
<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  align-items: stretch;
  background-color: #f1f1f1;
}
.flex-container > div {
  background-color: DodgerBlue;
  color: white;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>
<h1>The order Property</h1>
<p>Use the order property to sort the flex items as you like:</p>

<div class="flex-container">
  <div style="order: 3">1</div>
  <div style="order: 2">2</div>
  <div style="order: 4">3</div>
  <div style="order: 1">4</div>
</div>

</body>
</html>
```

플렉스 성장 속성

이 **flex-grow** 속성은 나머지 플렉스 항목에 비해 플렉스 항목이 얼마나 커질 지 지정합니다.

값은 숫자 여야하며 기본값은 0입니다.

세 번째 플렉스 항목을 다른 플렉스 항목보다 8 배 빠르게 자릅니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  align-items: stretch;
  background-color: #f1f1f1;
}
.flex-container > div {
  background-color: DodgerBlue;
  color: white;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>
<h1>The flex-grow Property</h1>
<p>Make the third flex item grow eight times faster than the other flex
items:</p>

<div class="flex-container">
  <div style="flex-grow: 1">1</div>
  <div style="flex-grow: 1">2</div>
  <div style="flex-grow: 8">3</div>
</div>

</body>
</html>
```

플렉스 수축 속성

이 **flex-shrink** 속성은 나머지 플렉스 항목에 비해 플렉스 항목이 축소되는 정도를 지정합니다.

값은 숫자 여야하며 기본값은 1입니다.

세 번째 플렉스 항목이 다른 플렉스 항목만큼 축소되지 않도록하십시오.

```
<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
```

```

    display: flex;
    align-items: stretch;
    background-color: #f1f1f1;
}
.flex-container > div {
    background-color: DodgerBlue;
    color: white;
    width: 100px;
    margin: 10px;
    text-align: center;
    line-height: 75px;
    font-size: 30px;
}
</style>
</head>
<body>
<h1>The flex-shrink Property</h1>
<p>Do not let the third flex item shrink as much as the other flex items:</p>

<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div style="flex-shrink: 0">3</div>
  <div>4</div>
  <div>5</div>
  <div>6</div>
  <div>7</div>
  <div>8</div>
  <div>9</div>
  <div>10</div>
</div>

</body>
</html>

```

플렉스 기반 속성

이 **flex-basis** 속성은 플렉스 항목의 초기 길이를 지정합니다.

세 번째 플렉스 항목의 초기 길이를 200 픽셀로 설정하십시오.

```

<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
    display: flex;
    align-items: stretch;
    background-color: #f1f1f1;
}
.flex-container > div {
    background-color: DodgerBlue;

```

```

    color: white;
    width: 100px;
    margin: 10px;
    text-align: center;
    line-height: 75px;
    font-size: 30px;
}
</style>
</head>
<body>
<h1>The flex-basis Property</h1>
<p>Set the initial length of the third flex item to 200 pixels:</p>

<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div style="flex-basis: 200px">3</div>
  <div>4</div>
</div>

</body>
</html>

```

플렉스 속성

flex 속성에 대한 약식 속성이다 **flex-grow**, **flex-shrink** 및 **flex-basis** 특성.

세 번째 플렉스 항목을 확장 가능하지 않고 (0) 축소 할 수없고 (0) 초기 길이 200 픽셀로 설정합니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  align-items: stretch;
  background-color: #f1f1f1;
}
.flex-container > div {
  background-color: DodgerBlue;
  color: white;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>

```

```

<h1>The flex Property</h1>
<p>Make the third flex item not growable (0), not shrinkable (0), and with an
initial length of 200 pixels:</p>

<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div style="flex: 0 0 200px">3</div>
  <div>4</div>
</div>

</body>
</html>

```

자기 정렬 속성

이 `align-self` 속성은 유연한 컨테이너 내에서 선택한 항목의 정렬을 지정합니다.

이 `align-self` 속성은 컨테이너 `align-items` 속성에 의해 설정된 기본 정렬보다 우선합니다.

이 예제에서는 200 픽셀 높이의 컨테이너를 사용하여 `align-self` 속성을 더 잘 보여줍니다.

컨테이너 가운데에 세 번째 플렉스 항목을 맞춥니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  height: 200px;
  background-color: #f1f1f1;
}
.flex-container > div {
  background-color: DodgerBlue;
  color: white;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>
<h1>The align-self Property</h1>
<p>The "align-self: center;" aligns the selected flex item in the middle of the

```

```

container:</p>

<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div style="align-self: center">3</div>
  <div>4</div>
</div>

<p>The align-self property overrides the align-items property of the
container.</p>

</body>
</html>

```

컨테이너 상단에 두 번째 플렉스 품목을, 컨테이너 하단에 세 번째 플렉스 품목을 맞춥니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  height: 200px;
  background-color: #f1f1f1;
}
.flex-container > div {
  background-color: DodgerBlue;
  color: white;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>
<h1>The align-self Property</h1>
<p>The "align-self: flex-start;" aligns the selected flex item at the top of the
container.</p>
<p>The "align-self: flex-end;" aligns the selected flex item at the bottom of the
container.</p>

<div class="flex-container">
  <div>1</div>
  <div style="align-self: flex-start">2</div>
  <div style="align-self: flex-end">3</div>
  <div>4</div>
</div>

<p>The align-self property overrides the align-items property of the
container.</p>

```

```
</body>
</html>
```

Flexbox를 사용한 반응형 이미지 갤러리

flexbox 를 사용하여 화면 크기에 따라 4 개, 2 개 또는 전체 너비 이미지로 변하는 반응 형 이미지 갤러리를 만듭니다.

```
<!DOCTYPE html>
<html>
<style>
* {
  box-sizing: border-box;
}

body {
  margin: 0;
  font-family: Arial;
}

.header {
  text-align: center;
  padding: 32px;
}

.row {
  display: flex;
  flex-wrap: wrap;
  padding: 0 4px;
}

/* Create four equal columns that sits next to each other */
.column {
  flex: 25%;
  max-width: 25%;
  padding: 0 4px;
}

.column img {
  margin-top: 8px;
  vertical-align: middle;
}

/* Responsive layout - makes a two column-layout instead of four columns */
@media (max-width: 800px) {
  .column {
    flex: 50%;
    max-width: 50%;
  }
}

/* Responsive layout - makes the two columns stack on top of each other instead
of next to each other */
```

```

@media (max-width: 600px) {
  .column {
    flex: 100%;
    max-width: 100%;
  }
}
</style>
<body>

<!-- Header -->
<div class="header">
  <h1>Responsive Image Grid</h1>
  <p>Resize the browser window to see the responsive effect.</p>
</div>

<!-- Photo Grid -->
<div class="row">
  <div class="column">
    
    
    
    
    
    
    
  </div>

  <div class="column">
    
    
    
    
    
    
  </div>

  <div class="column">
    
    
    
    
    
    
    
  </div>

  <div class="column">
    
    
    
    
    
    
  </div>
</div>

</body>
</html>

```

Flexbox를 사용하는 반응형 웹 사이트

flexbox 를 사용하여 유연한 탐색 막대와 유연한 콘텐츠가 포함 된 반응 형 웹 사이트를 만듭니다.

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
* {
  box-sizing: border-box;
}

/* Style the body */
body {
  font-family: Arial;
  margin: 0;
}

/* Header/logo Title */
.header {
  padding: 60px;
  text-align: center;
  background: #1abc9c;
  color: white;
}

/* Style the top navigation bar */
.navbar {
  display: flex;
  background-color: #333;
}

/* Style the navigation bar links */
.navbar a {
  color: white;
  padding: 14px 20px;
  text-decoration: none;
  text-align: center;
}

/* Change color on hover */
.navbar a:hover {
  background-color: #ddd;
  color: black;
}

/* Column container */
```



```

.row {
  display: flex;
  flex-wrap: wrap;
}

/* Create two unequal columns that sits next to each other */
/* Sidebar/left column */
.side {
  flex: 30%;
  background-color: #f1f1f1;
  padding: 20px;
}

/* Main column */
.main {
  flex: 70%;
  background-color: white;
  padding: 20px;
}

/* Fake image, just for this example */
.fakeimg {
  background-color: #aaa;
  width: 100%;
  padding: 20px;
}

/* Footer */
.footer {
  padding: 20px;
  text-align: center;
  background: #ddd;
}

/* Responsive layout - when the screen is less than 700px wide, make the two
columns stack on top of each other instead of next to each other */
@media screen and (max-width: 700px) {
  .row, .navbar {
    flex-direction: column;
  }
}
</style>
</head>
<body>

<!-- Note -->
<div style="background:yellow;padding:5px">
  <h4 style="text-align:center">Resize the browser window to see the responsive
effect.</h4>
</div>

<!-- Header -->
<div class="header">
  <h1>My Website</h1>
  <p>With a <b>flexible</b> layout.</p>
</div>

<!-- Navigation Bar -->

```

```

<div class="navbar">
  <a href="#">Link</a>
  <a href="#">Link</a>
  <a href="#">Link</a>
  <a href="#">Link</a>
</div>

<!-- The flexible grid (content) -->
<div class="row">
  <div class="side">
    <h2>About Me</h2>
    <h5>Photo of me:</h5>
    <div class="fakeimg" style="height:200px;">Image</div>
    <p>Some text about me in culpa qui officia deserunt mollit anim..</p>
    <h3>More Text</h3>
    <p>Lorem ipsum dolor sit ame.</p>
    <div class="fakeimg" style="height:60px;">Image</div><br>
    <div class="fakeimg" style="height:60px;">Image</div><br>
    <div class="fakeimg" style="height:60px;">Image</div>
  </div>
  <div class="main">
    <h2>TITLE HEADING</h2>
    <h5>Title description, Dec 7, 2017</h5>
    <div class="fakeimg" style="height:200px;">Image</div>
    <p>Some text..</p>
    <p>Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur
adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna
aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco.</p>
    <br>
    <h2>TITLE HEADING</h2>
    <h5>Title description, Sep 2, 2017</h5>
    <div class="fakeimg" style="height:200px;">Image</div>
    <p>Some text..</p>
    <p>Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur
adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna
aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco.</p>
  </div>
</div>

<!-- Footer -->
<div class="footer">
  <h2>Footer</h2>
</div>

</body>
</html>

```

CSS Flexbox 속성

다음 표는 flexbox 와 함께 사용되는 CSS 속성을 보여줍니다.

Property	Description
----------	-------------

<u>display</u>	Specifies the type of box used for an HTML element
<u>flex-direction</u>	Specifies the direction of the flexible items inside a flex container
<u>justify-content</u>	Horizontally aligns the flex items when the items do not use all available space on the main-axis
<u>align-items</u>	Vertically aligns the flex items when the items do not use all available space on the cross-axis
<u>flex-wrap</u>	Specifies whether the flex items should wrap or not, if there is not enough room for them on one flex line
<u>align-content</u>	Modifies the behavior of the flex-wrap property. It is similar to align-items, but instead of aligning flex items, it aligns flex lines
<u>flex-flow</u>	A shorthand property for flex-direction and flex-wrap
<u>order</u>	Specifies the order of a flexible item relative to the rest of the flex items inside the same container
<u>align-self</u>	Used on flex items. Overrides the container's align-items property
<u>flex</u>	A shorthand property for the flex-grow, flex-shrink, and the flex-basis properties