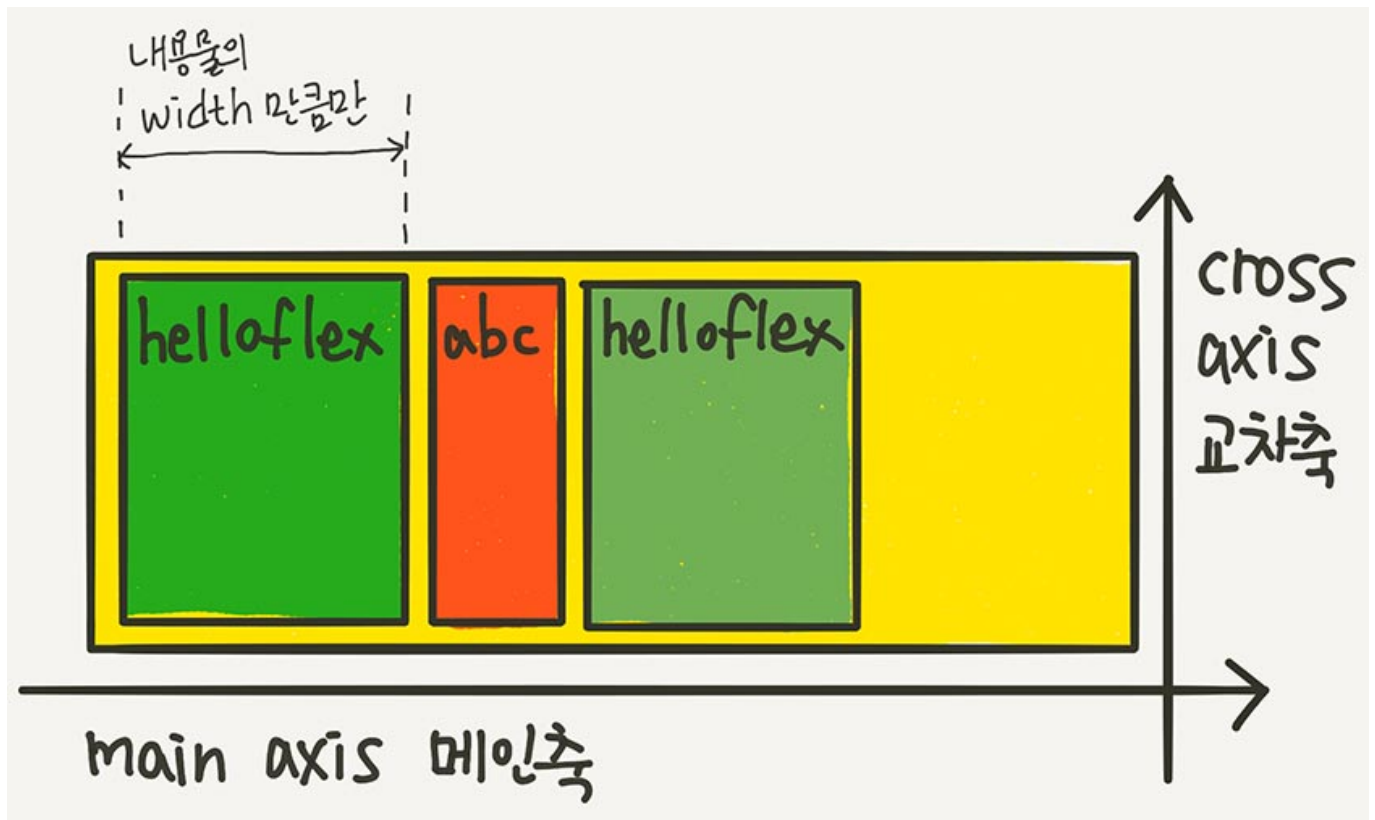


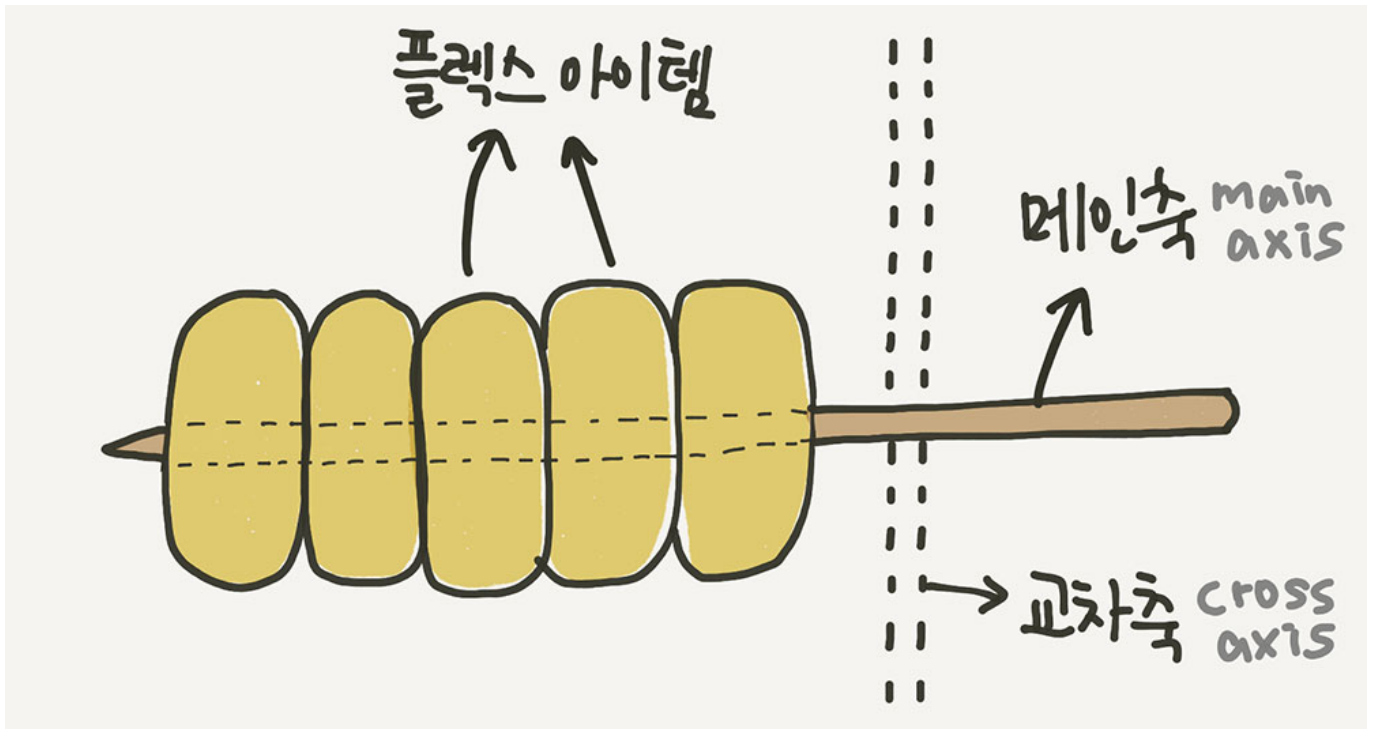
# Flex

flexbox는 float나 inline-block의 단점을 보완하여 레이아웃 배치 전용 기능으로 고안된 CSS의 새로운 display 특성으로 컨테이너에 적용하는 속성과 아이টে에 적용하는 속성으로 나누어 집니다. flexbox를 사용하면 레이아웃 구성에 많은 편의성이 보장되지만 IE는 지원하지 않기 때문에 IE를 고려해야 하는 프로젝트에서는 사용해서는 안됩니다.

## #01. 메인축(main-axis)과 교차축(cross-axis)

| 용어  | 설명              |
|-----|-----------------|
| 메인축 | 아이템들이 배치된 방향의 축 |
| 교차축 | 메인축과 수직인 축      |

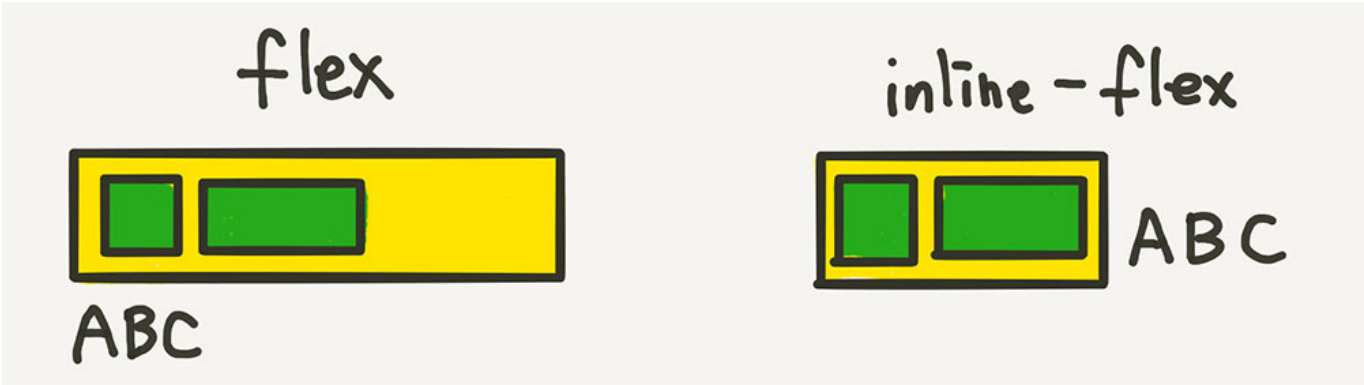
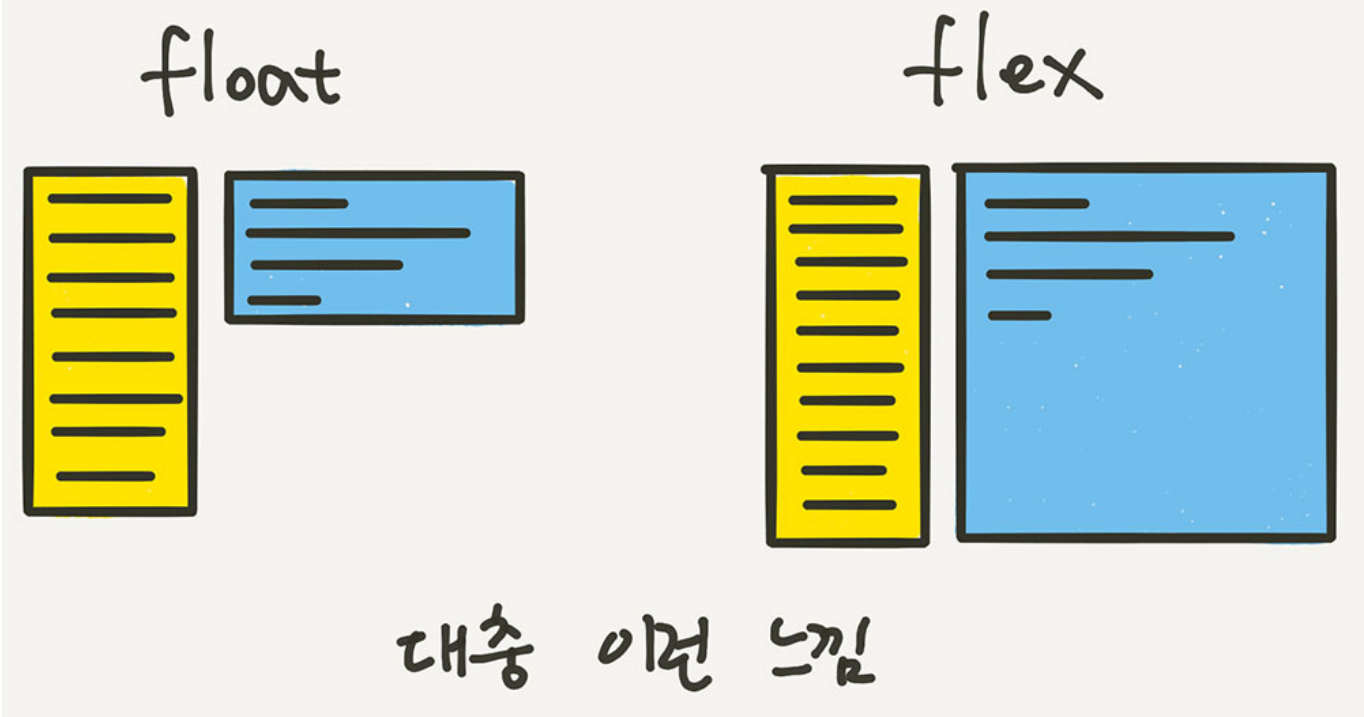




## #02. 컨테이너(부모)에 적용하는 속성

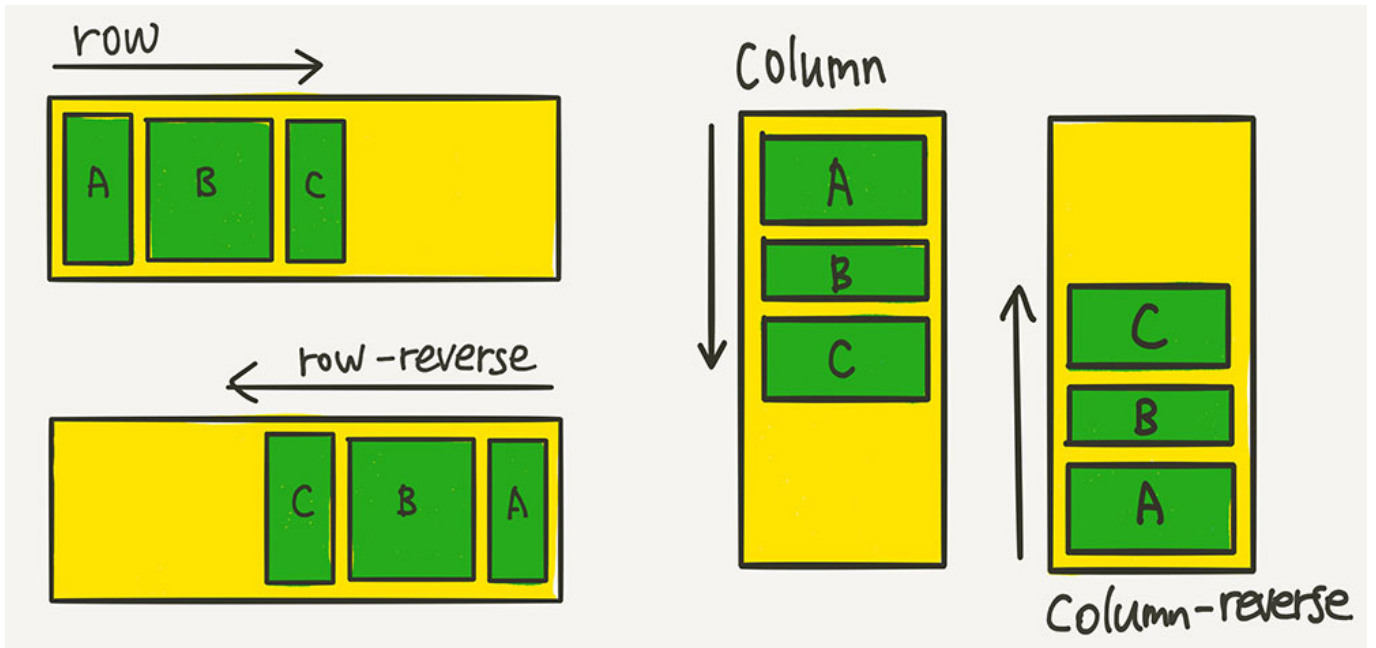
### 1) display 속성

| 값           | 설명  |
|-------------|---|
| flex        | <ul style="list-style-type: none"> <li>- 컨테이너 내의 아이템들을 가로로 배치함.</li> <li>- 컨테이너 자체의 넓이는 <b>block</b>의 특성을 유지함.</li> <li>- 아이템들은 자신이 가진 내용물의 width 만큼만 넓이를 차지함.</li> <li>- height는 컨테이너의 높이만큼 늘어남. (내용의 높이가 아닌 부모의 높이만큼 설정됨)</li> <li>- height가 알아서 늘어나는 특징은 컬럼 레이아웃을 만들 때 편리</li> <li>- 정렬 속성을 통해 height를 어떻게 처리할지도 조정할 수 있음.</li> <li>- 내용만큼의 높이를 설정하고 싶다면 컨테이너에게 높이를 지정하지 않아야 함.</li> </ul> |
| inline-flex | <ul style="list-style-type: none"> <li>- 컨테이너의 넓이가 아이템들(내용)만큼만 설정됨.</li> <li>- 그 외의 특성은 <b>flex</b>와 동일함</li> </ul>   |



2) 배치방향 설정 - flex-direction

| 값              | 설명                  |
|----------------|---------------------|
| row            | 가로배치, 왼쪽정렬, 순서배치    |
| row-reverse    | 가로배치, 오른쪽 정렬, 역순 배치 |
| column         | 세로배치, 상단정렬, 순서배치    |
| column-reverse | 세로배치, 하단정렬, 역순배치    |



### 3) 줄바꿈 속성 : flex-wrap

| 값            | 설명                                       |
|--------------|--|
| nowrap       | 줄바꿈을 하지 않고 아이템의 크기가 컨테이너에 맞게 강제로 축소된다.   |
| wrap         | 줄바꿈을 한다. float나 inline-block과 비슷하게 동작한다. |
| wrap-reverse | 줄바꿈을 하지만 아이템의 행을 역순으로 배치한다.              |

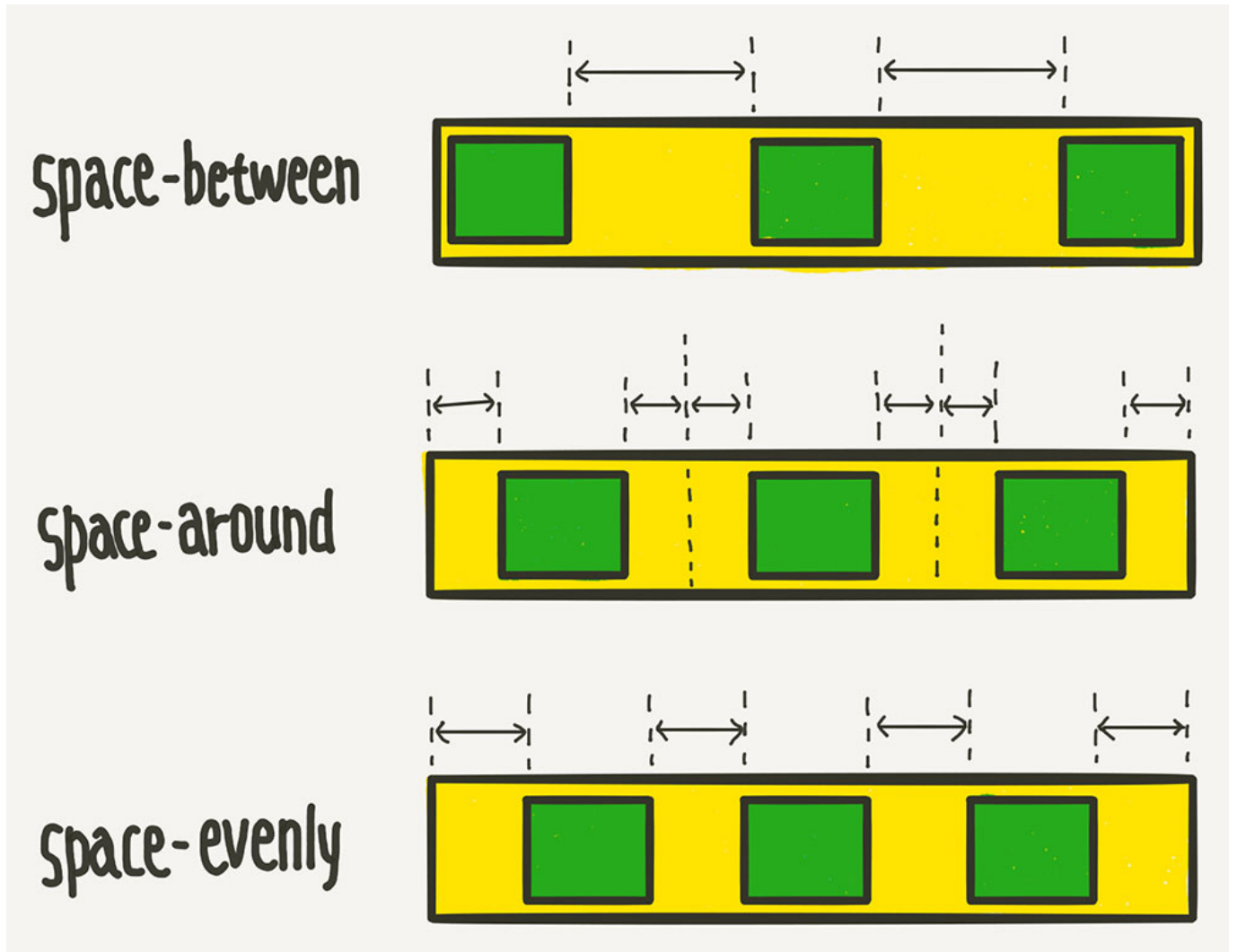
### 4) flex-flow

**flex-direction** 속성과 **flex-wrap**을 일괄지정하기 위한 단축 속성

flex-direction, flex-wrap의 순으로 한 칸 떼고 명시한다.

### 5) 좌우정렬 : justify-content

| 값             | 설명   |
|---------------|--|
| flex-start    | 아이템들을 시작점으로 정렬한다.<br>flex-direction이 row(가로 배치)일 때는 왼쪽, column(세로 배치)일 때는 위  |
| flex-end      | 아이템들을 끝점으로 정렬한다.<br>flex-direction이 row(가로 배치)일 때는 오른쪽, column(세로 배치)일 때는 아래 |
| center        | 아이템들을 가운데로 정렬한다.   |
| space-between | 아이템들의 "사이(between)"에 균일한 간격을 만들어 준다.   |
| space-around  | 아이템들의 "둘레(around)"에 균일한 간격을 만들어 준다.  |
| space-evenly  | 아이템들의 사이와 양 끝에 균일한 간격을 만들어 준다.<br>(주의! IE와 엣지(Edge)에서는 지원되지 않음.)             |



## 6) 수직축 방향 정렬 : align-items

| 값          | 설명   |
|------------|--|
| stretch    | 아이템들이 수직축 방향으로 끝까지 늘어난다.   |
| flex-start | 아이템들을 시작점으로 정렬<br>flex-direction이 row(가로 배치)일 때는 위, column(세로 배치)일 때는 왼쪽 |
| flex-end   | 아이템들을 끝으로 정렬<br>flex-direction이 row(가로 배치)일 때는 아래, column(세로 배치)일 때는 오른쪽 |
| center     | 아이템들을 가운데로 정렬  |
| baseline   | 아이템들을 텍스트 베이스라인 기준으로 정렬  |

## 7) 여러 행 정렬 : align-content

**flex-wrap: wrap;**이 설정된 상태에서, 아이템들의 행이 2줄 이상 되었을 때의 수직축 방향 정렬을 결정하는 속성.

| 값       | 설명                       |
|---------|--------------------------|
| stretch | 아이템들이 세로축 방향으로 끝까지 늘어난다. |

| 값             | 설명   |
|---------------|--|
| flex-start    | 아이템들을 시작점으로 정렬(위)  |
| flex-end      | 아이템들을 끝으로 정렬(아래)   |
| center        | 아이템들을 가운데로 정렬  |
| space-between | 아이템들의 "사이(between)"에 균일한 간격을 만들어 준다.                             |
| space-around  | 아이템들의 "둘레(around)"에 균일한 간격을 만들어 준다.                              |
| space-evenly  | 아이템들의 사이와 양 끝에 균일한 간격을 만들어 준다.<br>(주의! IE와 엣지(Edge)에서는 지원되지 않음.) |

## #02. 아이템에 적용하는 속성들.

### 1) 유연한 박스의 기본 영역 : flex-basis

flex-basis는 Flex 아이템의 기본 크기를 설정한다.(flex-direction이 row일 때는 너비, column일 때는 높이)

flex-basis의 값으로는 width, height 등에 사용하는 각종 단위의 수(px,%,rem)가 들어갈 수 있으며 기본값 **auto**는 해당 아이템의 width값을 사용한다.(width를 따로 설정하지 않으면 콘텐츠의 크기가 됨)

**content**는 콘텐츠의 크기로, width를 따로 설정하지 않은 경우와 같다.

### 2) 유연하게 늘리기 : flex-grow

flex-grow는 아이템이 flex-basis의 값보다 커질 수 있는지를 결정하는 속성이다.

flex-grow에는 숫자값이 들어가는데, 몇이든 일단 0보다 큰 값이 세팅이 되면 해당 아이템이 유연한(Flexible) 박스로 변하고 원래의 크기보다 커지며 빈 공간을 메우게 된다.

기본값이 0이기 때문에, 따로 적용하기 전까지는 아이템이 늘어나지 않는다.

### 3) 유연하게 줄이기 : flex-shrink

flex-shrink는 flex-grow와 쌍을 이루는 속성으로, 아이템이 flex-basis의 값보다 작아질 수 있는지를 결정한다.

flex-shrink에는 숫자값이 들어가는데, 몇이든 일단 0보다 큰 값이 세팅이 되면 해당 아이템이 유연한(Flexible) 박스로 변하고 flex-basis보다 작아진다.

기본값이 1이기 때문에 따로 세팅하지 않는다면 아이템이 flex-basis보다 작아질 수 있다.

### 4) 축약형 속성 : flex

flex-grow, flex-shrink, flex-basis를 한 번에 쓸 수 있는 축약형 속성.

이 세 속성들은 서로 관련이 깊기 때문에, 이 축약형을 쓰는 경우가 많다.

주의할 점은, flex: 1; 이런 식으로 flex-basis를 생략해서 쓰면 flex-basis의 값은 0이 된다.

## 예시

```
/* flex-grow: 1; flex-shrink: 1; flex-basis: 0%; */
flex: 1;

/* flex-grow: 1; flex-shrink: 1; flex-basis: auto; */
flex: 1 1 auto;

/* flex-grow: 1; flex-shrink: 1; flex-basis: 500px; */
flex: 1 500px;
```

## 5) 배치 순서 : order

각 아이템들의 시각적 나열 순서를 결정

숫자값이 들어가며, 작은 숫자일 수록 먼저 배치됩니다. “시각적” 순서일 뿐, HTML 자체의 구조를 바꾸는 것은 아니므로 접근성 측면에서 사용에 주의해야 한다.

시각 장애인분들이 사용하는 스크린 리더로 화면을 읽을 때, order를 이용해 순서를 바꾼 것은 의미가 없다.

## 6) 수직축으로 아이템 정렬 : align-self

align-items의 아이템 버전

align-items가 전체 아이템의 수직축 방향 정렬이라면, align-self는 해당 아이템의 수직축 방향 정렬이다.

기본값은 auto로, 기본적으로 align-items 설정을 상속 받는다.

align-self는 align-items보다 우선 적용된다.

auto외의 나머지 값들은 align-items와 동일하다.

| 값          | 설명   |
|------------|--|
| auto       | 컨테이너의 align-items 설정을 상속 받는다.  |
| stretch    | 아이템들이 수직축 방향으로 끝까지 늘어난다.   |
| flex-start | 아이템들을 시작점으로 정렬<br>flex-direction이 row(가로 배치)일 때는 위, column(세로 배치)일 때는 왼쪽 |
| flex-end   | 아이템들을 끝으로 정렬<br>flex-direction이 row(가로 배치)일 때는 아래, column(세로 배치)일 때는 오른쪽 |
| center     | 아이템들을 가운데로 정렬  |
| baseline   | 아이템들을 텍스트 베이스라인 기준으로 정렬  |

## 참고문헌

<https://studiomeal.com/archives/197>