

## CSS Flex(Flexible Box)

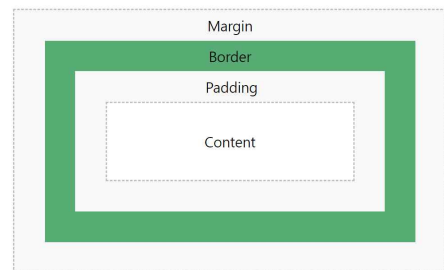
대부분 Site는 전체 layout이 column(수직) 구성이며 '위-아래'로 scroll 하여 사용한다. layout을 구성할 때 가장 많이 사용하는 element(Elements)들이 기본적으로 블록(Block) 개념으로 표시(Display)되며 이는 뷰(View)에 column(수직)(위에서 아래로)으로 쌓이기 때문에 column(수직) 구성은 상대적으로 쉽게 만들 수 있다.

그러나 row(수평)(왼쪽에서 오른쪽으로) 구성의 경우는 상황이 조금 다르다. 문제는 row(수평) 구조를 만드는 property이 명확하지 않았기 때문인데, 그래서 많은 경우 <table>나 float 혹은 inline-block 등의 도움을 받았다.

따라서 이러한 방법들은 다양한 문제(Clear, White space 등, 해결은 가능하지만..)를 가지고 있기 때문에 결국 row(수평) layout 구성의 차선택일 뿐이며, 이제 Flex(Flexible Box)라는 명확한 개념(property들)으로 layout을 쉽게 구성할 수 있다. 위에서 쉬운 row(수평) 구성을 얘기했지만 Flex는 쉬운 column(수직) 구성도 가능하다.

CSS에서 "Box Model"이라는 용어는 Design과 Layout 대해 이야기할 때 사용된다.

CSS Box Model은 기본적으로 모든 HTML element를 감싸는 box이다. margin, border, padding 및 실제 content로 구성된다. 아래 이미지는 Box Model을 보여준다.



- ☞ content: text와 image가 표시되는 box의 content
- ☞ padding: content 주변 영역을 지운다. padding이 transparent(투명) 하다
- ☞ border: padding과 content를 둘러싸는 border(테두리)
- ☞ margin: border(테두리) 밖의 영역을 지운다. margin이 transparent(투명) 하다

시작하기에 앞서 간단한 예제를 살펴본다.

float property을 이용한 row(수평) 구성의 경우 다음과 같이 style을 작성할 수 있다.

```
<div class="box"></div>
<div class="box"></div>
<div class="box"></div>
<div class="clear-element"></div>

.box {
  float: left;
}
.clear-element {
  clear: both; /* or left */
}
```

자세한 설명은 생략하고, 위 방법은 보기엔 단순하지만 box를 제외한 clear-element라는 이름(class)의 다음(next) element도 있어야 하기 때문에 실제 사용엔 매우 불편하며 명확하지 않은 방법으로써 많은 경우 아래 방식을 사용한다.

```
<div class="clearfix">
  <div class="box"></div>
  <div class="box"></div>
  <div class="box"></div>
</div>
```

```
/* IE 핵이나 기타 방식을 제외하고 가장 원리에 충실한 방법 */
.clearfix::after {
  content: "";
  clear: both;
  display: block;
}
.box {
  float: left;
}
```

예제를 보면 row(수평)이 될 element들에 직접 float를 적용하고 그 element들의 Container(parent element)에 미리 설정한 clearfix를 적용한다. 그러면 Flexible Box(이하 Flex)는 어떻게 작성할 수 있을까?

아주 간단하다.

```
<div class="box-container">
  <div class="box"></div>
  <div class="box"></div>
  <div class="box"></div>
</div>

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

Flex는 row(수평)이 될 element들의 Container(box-container)에 display: flex;를 적용한다. (세부 property이 필요하지 않은 경우도 많기 때문에 상당히 쉽고 빠르게 row(수평) element를 구성할 수 있다.)

## ☑ CSS3 Flexible Box

Flex는 element의 크기가 불분명하거나 동적인 경우에도 각 element를 정렬할 수 있는 효율적인 방법을 제공한다. 우선 Flex는 2개의 개념으로 나눈다. 첫 번째는 Container, 두 번째는 Items 이다.

위에서 살펴본 바와 같이 Container는 Items를 감싸는 parent element이며, 각 Item을 정렬하기 위해선 Container가 필수이다. 주의할 부분은 Container와 Items에 적용하는 property이 구분되어 있다는 것이다.

Container에는 **display**, **flex-flow**, **justify-content** 등의 property을 사용할 수 있으며, Items에는 **order**, **flex**, **align-self** 등의 property을 사용할 수 있다.



## ❖ Flex Container

Flex Container를 위한 property들은 다음과 같다. 주 축(main-axis)과 교차 축(cross-axis)의 개념은 뒤에서 살펴본다.

property	의미
display	Flex Container를 정의
flex-flow	flex-direction와 flex-wrap의 단축 property
flex-direction	Flex Items의 주 축(main-axis)을 설정
flex-wrap	Flex Items의 여러 줄 묶음(줄 바꿈) 설정
justify-content	주 축(main-axis)의 정렬 방법을 설정
align-content	교차 축(cross-axis)의 정렬 방법을 설정(2줄 이상)
align-items	교차 축(cross-axis)에서 Items의 정렬 방법을 설정(1줄)

## 🔗 display

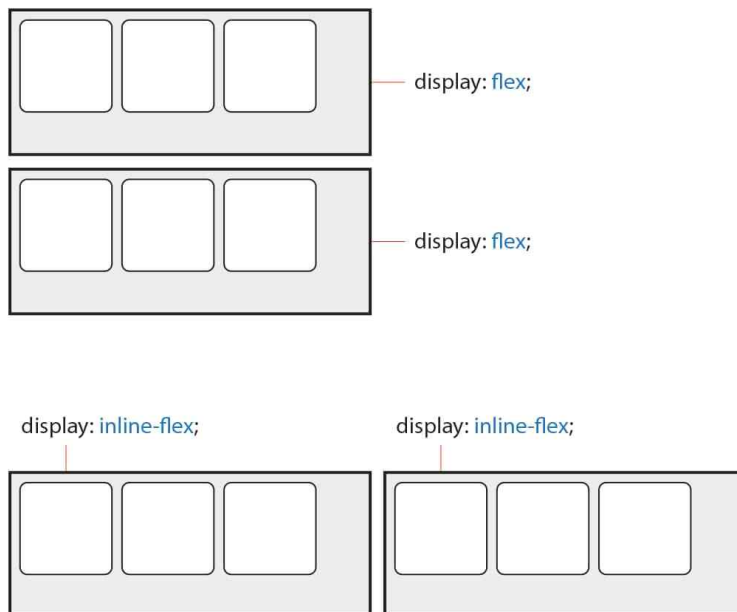
display property으로 Flex Container를 정의한다. 보통 element의 표시 방법을 display: block;, display: inline-block 혹은 display: none; 같이 사용하는 경우가 많다. 같은 element의 표시 방법으로 Block이나 Inline이 아닌 Flex(display: flex, display: inline-flex)로 정의한다.

값	의미	기본값
flex	Block 특성의 Flex Container를 정의	
inline-flex	Inline 특성의 Flex Container를 정의	

flex와 inline-flex는 차이는 단순하다.

- ✓ **display: flex;**로 지정된 Flex Container는 Block element와 같은 성향(column(수직) 쌓임)을 가지며,
- ✓ **display: inline-flex**로 지정된 Flex Container는 Inline(Inline Block) element와 같은 성향(row(수평) 쌓임)을 가진다.

여기서 말하는 column(수직)과 row(수평) 쌓임은 Items가 아니라 Container라는 것에 주의한다. 두 값의 차이는 내부에 Items에는 영향을 주지 않는다.



### 👉 flex-flow

이 property은 단축 property으로 Flex Items의 주 축(main-axis)을 설정하고 Items의 여러 줄 묶음(줄바꿈)도 설정한다.

**flex-flow: 주축 여러줄묶음;**

**.flex-container {**

```
flex-flow: row-reverse wrap;
}
```

값	의미	기본값
flex-direction	Items의 주 축(main-axis)을 설정	row
flex-wrap	Items의 여러 줄 묶음(줄 바꿈) 설정	nowrap

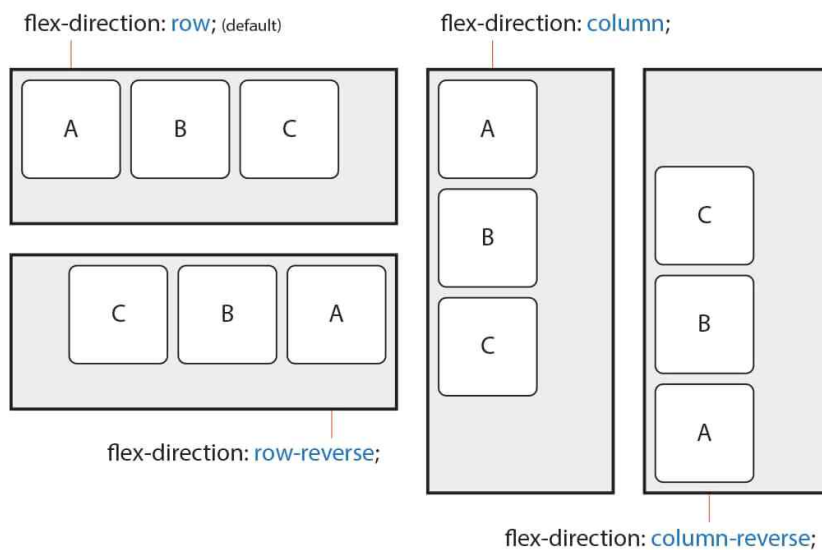
개별 property을 알아본다.

### ☞ flex-direction

Items의 주 축(main-axis)을 설정한다.

값	의미	기본값
row	Items를 row(수평)축(왼쪽에서 오른쪽으로)으로 표시	row
row-reverse	Items를 row의 반대 축으로 표시	
column	Items를 column(수직)축(위에서 아래로)으로 표시	
column-reverse	Items를 column의 반대 축으로 표시	

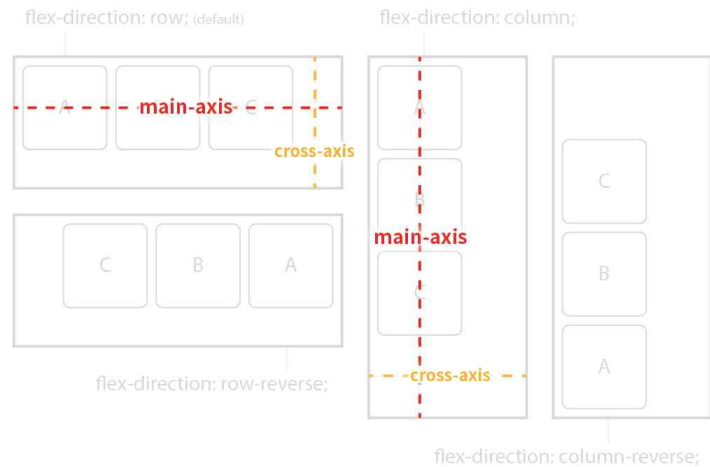
flex-direction: 주축;



### ★ 주 축(main-axis)과 교차 축(cross-axis)

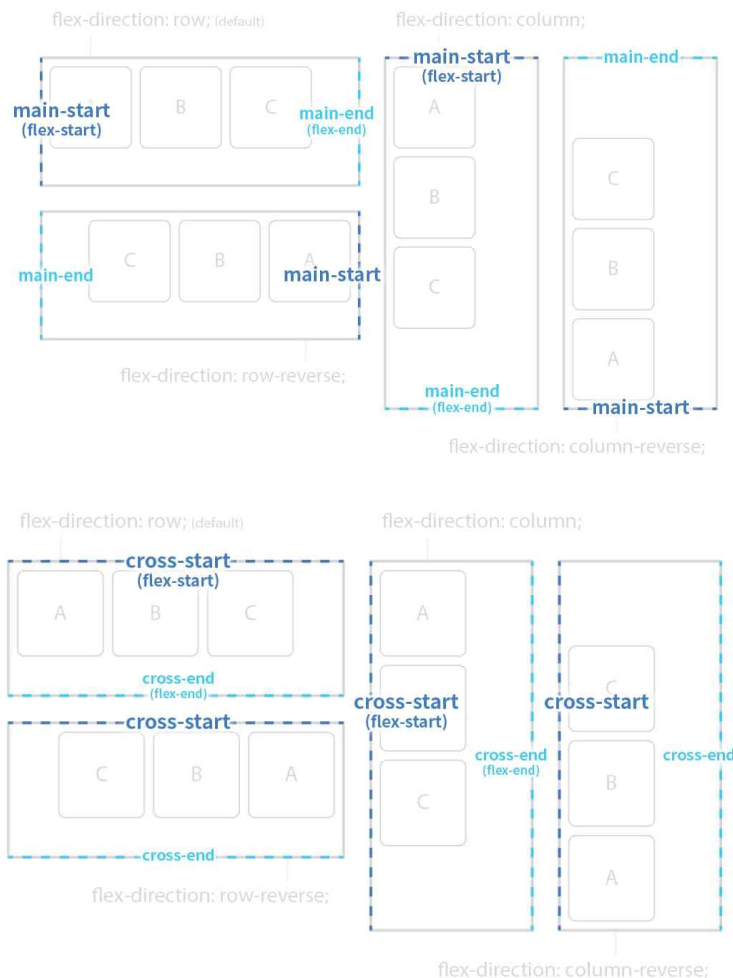
위에서 언급했었던 주 축(main-axis)과 교차 축(cross-axis)의 개념은 다음과 같다. 값 row는 Items를

row(수평)축으로 표시하므로 이때는 주 축이 row(수평)이며 교차 축은 column(수직)이 된다. 반대로 값 column은 Items를 column(수직)축으로 표시하므로 주 축은 column(수직)이며 교차 축은 row(수평)이 된다. 즉, 방향(row(수평), column(수직))에 따라 주 축과 교차 축이 달라진다.



### ★ 시작점(flex-start)과 끝점(flex-end)

시작점(flex-start)과 끝점(flex-end)이라는 개념도 있다. 이는 주 축이나 교차 축의 시작하는 지점과 끝나는 지점을 지칭한다. 역시 방향에 따라 시작점과 끝점이 달라진다.



뒤에서 언급할 property 중 값으로 flex-start와 flex-end를 사용하는데 이는 방향에 맞는 그 시작점과 끝점을 의미한다.

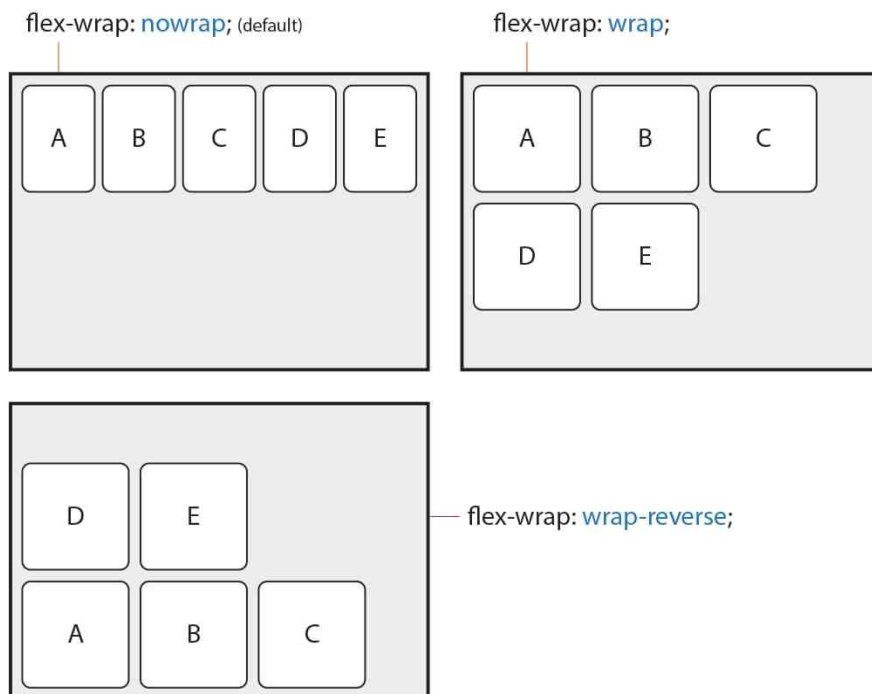
## ☞ flex-wrap

Items의 여러 줄 묶음(줄 바꿈)을 설정한다.

값	의미	기본값
nowrap	모든 Items를 여러 줄로 묶지 않음(한 줄에 표시)	nowrap
wrap	Items를 여러 줄로 묶음	
wrap-reverse	Items를 wrap의 역 방향으로 여러 줄로 묶음	

## flex-wrap: 여러줄묶음;

기본적으로 Items는 한 줄에서만 표시되고 줄 바꿈 되지 않는다. 이는 지정된 크기(주축에 따라 width 나 height)를 무시하고 한 줄 안에서만 가변한다. Items를 줄 바꿈 하려면 값으로 wrap을 사용해야 합니다.

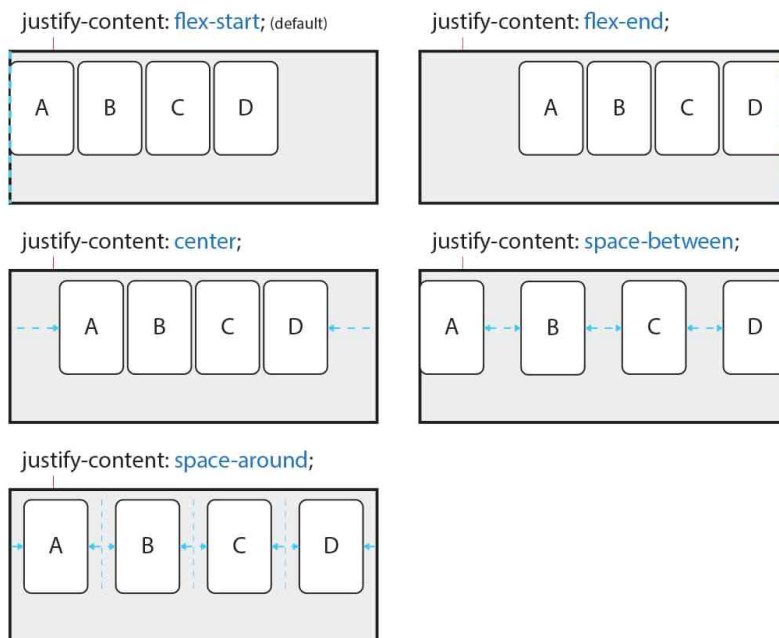


## ☞ justify-content

주 축(main-axis)의 정렬 방법을 설정한다.

값	의미	기본값
flex-start	Items를 시작점(flex-start)으로 정렬	flex-start
flex-end	Items를 끝점(flex-end)으로 정렬	
center	Items를 가운데 정렬	
space-between	시작 Item은 시작점에, 마지막 Item은 끝점에 정렬되고 나머지 Items는 사이에 고르게 정렬됨	
space-around	Items를 균등한 여백을 포함하여 정렬	

justify-content: 정렬방법;



## ☞ align-content

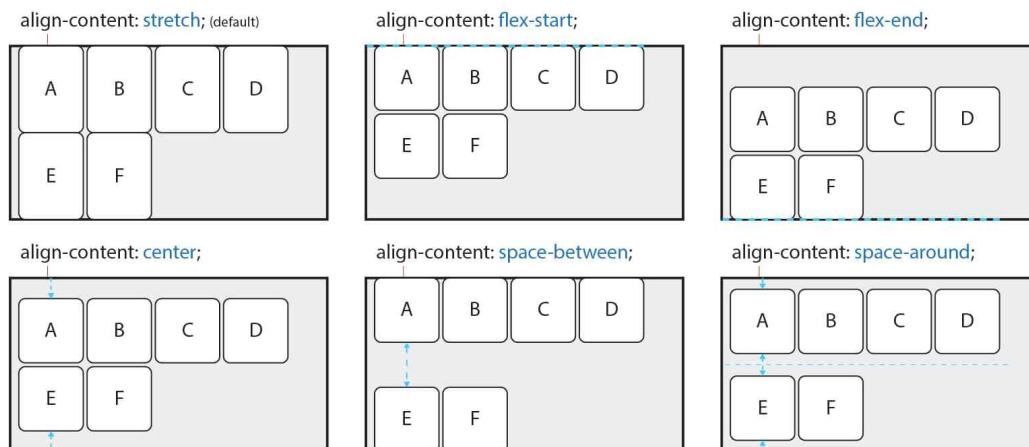
교차 축(cross-axis)의 정렬 방법을 설정한다. 주의할 점은 flex-wrap property를 통해 Items가 여러 줄(2줄 이상)이고 여백이 있을 경우만 사용할 수 있다. Items가 한 줄일 경우 align-items property를 사용한다.



값	의미	기본값
stretch	Container의 교차 축을 채우기 위해 Items를 늘림	stretch
flex-start	Items를 시작점(flex-start)으로 정렬	
flex-end	Items를 끝점(flex-end)으로 정렬	
center	Items를 가운데 정렬	
space-between	시작 Item은 시작점에, 마지막 Item은 끝점에 정렬되고 나머지 Items는 사이에 고르게 정렬됨	
space-around	Items를 균등한 여백을 포함하여 정렬	

### align-content: 정렬방법;

값 stretch는 교차 축에 해당하는 너비(property width 혹은 height)가 값이 auto(기본값)일 경우 교차 축을 채우기 위해 자동으로 늘어난다.

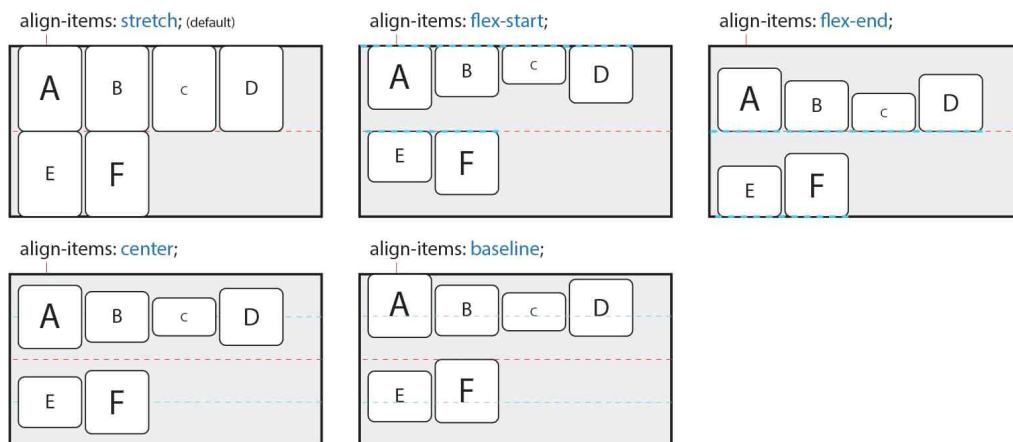


### ☞ align-items

교차 축(cross-axis)에서 Items의 정렬 방법을 설정한다. Items가 한 줄일 경우 많이 사용한다. 주의할 점은 Items가 flex-wrap을 통해 여러 줄(2줄 이상)일 경우에는 align-content property이 우선한다. 따라서 align-items를 사용하려면 align-content property을 기본값(stretch)으로 설정해야 한다.

값	의미	기본값
stretch	Container의 교차 축을 채우기 위해 Items를 늘림	stretch
flex-start	Items를 각 줄의 시작점(flex-start)으로 정렬	
flex-end	Items를 각 줄의 끝점(flex-end)으로 정렬	
center	Items를 가운데 정렬	
baseline	Items를 문자 기준선에 정렬	

align-items: 정렬방법;



## ❖ Flex Items

Flex Items를 위한 property들은 다음과 같다.

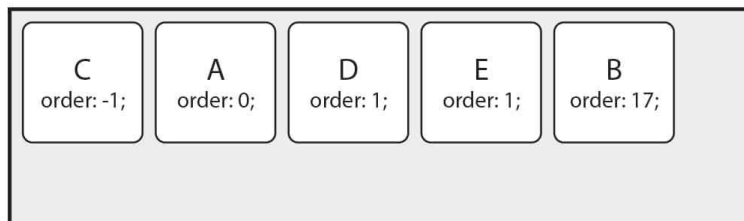
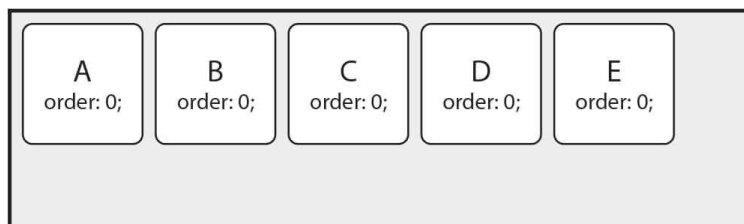
property	의미
order	Flex Item의 순서를 설정
flex	flex-grow, flex-shrink, flex-basis의 단축 property
flex-grow	Flex Item의 증가 너비 비율을 설정
flex-shrink	Flex Item의 감소 너비 비율을 설정
flex-basis	Flex Item의 (공간 배분 전) 기본 너비 설정
align-self	교차 축(cross-axis)에서 Item의 정렬 방법을 설정

## 👉 order

Item의 순서를 설정한다. Item에 숫자를 지정하고 숫자가 클수록 순서가 밀린다. 음수가 허용된다. HTML 구조와 상관없이 순서를 변경할 수 있기 때문에 유용하다.

값	의미	기본값
숫자	Item의 순서를 설정	0

order: 순서;



## 👉 flex

Item의 너비(증가, 감소, 기본)를 설정하는 단축 property이다.

값	의미	기본값
flex-grow	Item의 증가 너비 비율을 설정	0
flex-shrink	Item의 감소 너비 비율을 설정	1
flex-basis	Item의 (공간 배분 전) 기본 너비 설정	auto

flex: 증가너비 감소너비 기본너비;

```
.item {  
  flex: 1 1 20px; /* 증가너비 감소너비 기본너비 */  
  flex: 1 1; /* 증가너비 감소너비 */  
  flex: 1 20px; /* 증가너비 기본너비 (단위를 사용하면 flex-basis가 적용된다) */  
}
```

flex-grow를 제외한 개별 property은 생략할 수 있다. 만약 flex: 1;로 작성하면 flex-grow: 1;과 같다. 그러면 나머지 property은 생략했으니 기본값이 적용되어 flex-shrink: 1;, flex-basis: auto;가 된다. 즉 flex: 1; 혹은 flex: 1 1;은 flex: 1 1 auto;와 같다고 볼 수 있지만 그렇지 않다. flex-basis의 기본값은 auto 이지만 단축 property인 flex에서 그 값을 생략할 경우 0이 적용된다.

다시 정리하면 flex: 1; 혹은 flex: 1 1;은 flex: 1 1 0;이 된다는 것이다. 이 부분을 기억하지 않으면 엉뚱한 결과가 나올 수 있으니 주의한다.

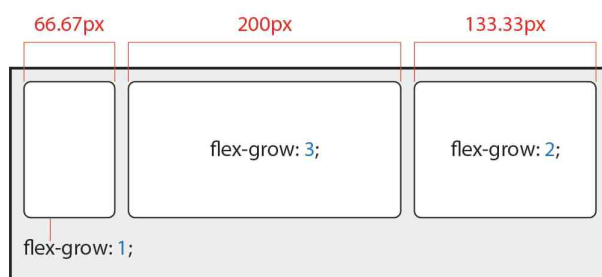
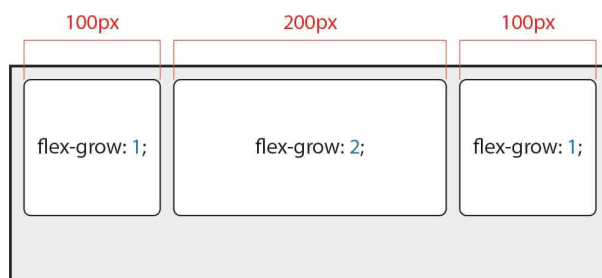
### ☞ flex-grow

Item의 증가 너비 비율을 설정한다. 숫자가 크면 더 많은 너비를 가진다. Item이 가변 너비가 아니거나, 값이 0일 경우 효과가 없다.

값	의미	기본값
숫자	Item의 증가 너비 비율을 설정	0

### flex-grow: 증가너비;

모든 Items의 총 증가 너비(flex-grow)에서 각 Item의 증가 너비의 비율 만큼 너비를 가질 수 있다. 예를 들어 Item이 3개이고 증가 너비가 각각 1, 2, 1이라면, 첫 번째 Item은 총 너비의 25%(1/4)을, 두 번째 Item은 총 너비의 50%(2/4)를, 세 번째 Item은 총 너비의 25%(1/4)을 가지게 된다.



## ☞ flex-shrink

Item이 감소하는 너비의 비율을 설정한다. 숫자가 크면 더 많은 너비가 감소한다.

Item이 가변 너비가 아니거나, 값이 0일 경우 효과가 없다.

값	의미	기본값
숫자	Item의 감소 너비 비율을 설정	1

## flex-shrink: 감소너비;

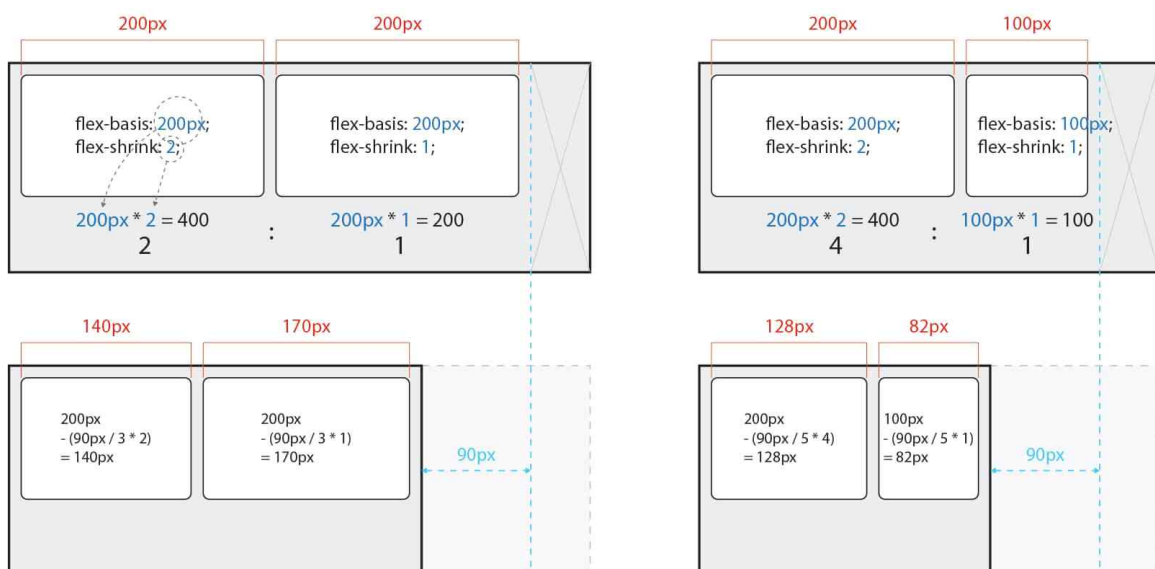
감소 너비(flex-shrink)는 element의 너비에 영향을 받기 때문에 계산하기 까다롭다. 영향을 받는다는 element의 너비는 width, height, flex-basis 등으로 너비가 지정된 경우를 의미한다. Container의 너비가 줄어 Items의 너비에 영향을 미칠 경우, 영향을 미치기 시작한 지점부터 줄어드는 거리 만큼 감소 너비 비율에 맞게 Item의 너비가 줄어든다.

예를 들어 Container의 너비가 줄어 Item의 너비에 영향을 미치기 시작한 지점부터 실제 줄어드는 거리가 90px일 때, element 너비가 같은 Item이 2개이고 지정된 감소 너비가 각각 2와 1이라면, 감소 너비는 2:1 비율이며, 첫 번째 Item은 90px의 2/3인 60px 만큼 너비가 감소하고, 두 번째 Item은 90px의 1/3인 30px 만큼 너비가 감소한다.

다른 예시로 Container의 너비가 줄어 Item의 너비에 영향을 미치기 시작한 지점부터 실제 줄어드는 거리가 90px일 때, element 너비가 다른 Item이 2개이고 element 너비는 각각 200과 100이고, 지정된 감소 너비가 각각 2와 1이라면,  $200 \times 2 = 400$ 과  $100 \times 1 = 100$  즉 감소 너비는 4:1 비율이며,

첫 번째 Item은 90px의 4/5인 72px 만큼 너비가 감소하고,

두 번째 Item은 90px의 1/5인 18px 만큼 너비가 감소한다.



계산이 까다롭기 때문에 활용도는 조금 떨어진다고 생각한다. 원리 정도만 이해하고 넘어간다.

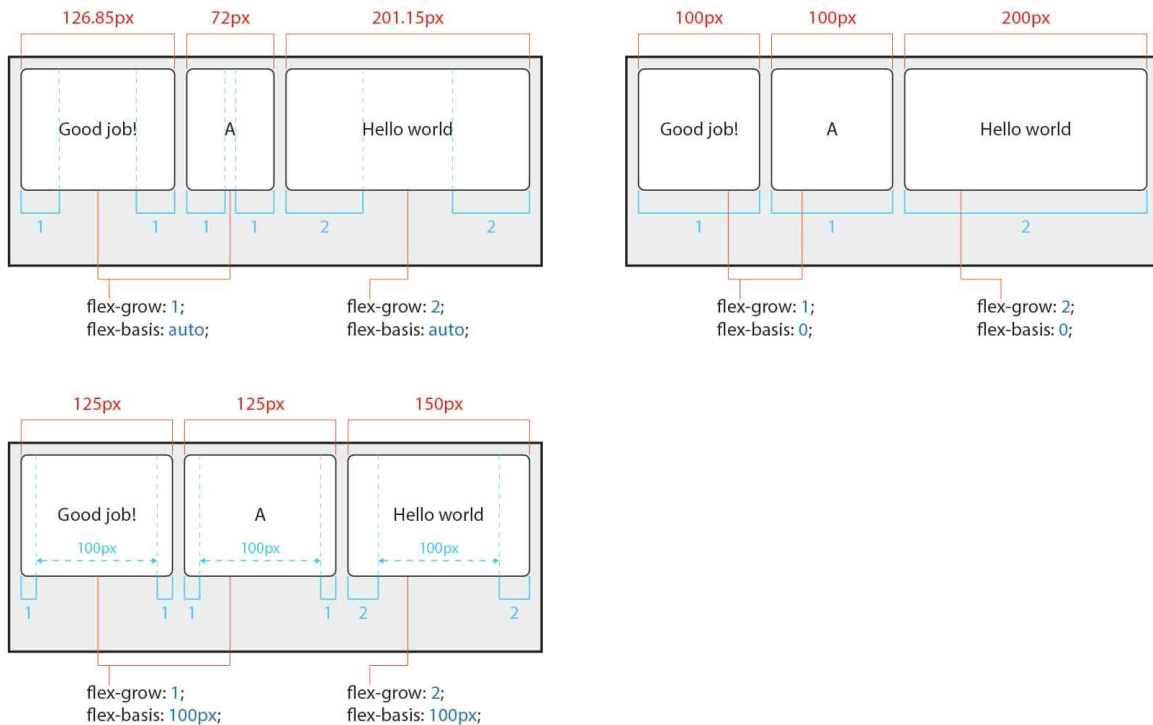
## ☞ flex-basis

Item의 (공간 배분 전) 기본 너비를 설정한다. 값이 auto일 경우 width, height 등의 property으로 Item의 너비를 설정할 수 있다. 그러나 단위 값이 주어질 경우 설정할 수 없다.

값	의미	기본값
auto	가변 Item과 같은 너비	auto
단위	px, em, cm 등 단위로 지정	

## flex-basis: 기본너비;

flex property에서 설명한 것 같이 단축 property 내에서 flex-basis를 생략하면 값이 0이 되는 것을 주의한다.

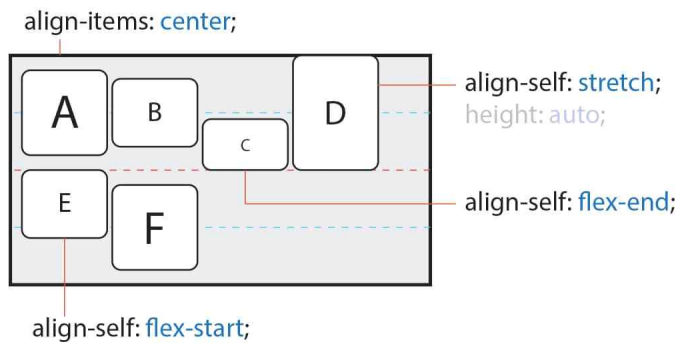


## ☞ align-self

교차 축(cross-axis)에서 개별 Item의 정렬 방법을 설정한다. align-items는 Container 내 모든 Items의 정렬 방법을 설정한다. 필요에 의해 일부 Item만 정렬 방법을 변경하려고 할 경우 align-self를 사용할 수 있다. 이 property은 align-items property보다 우선한다.

값	의미	기본값
auto	Container의 align-items property을 상속받음	auto
stretch	Container의 교차 축을 채우기 위해 Item을 늘림	
flex-start	Item을 각 줄의 시작점(flex-start)으로 정렬	
flex-end	Item을 각 줄의 끝점(flex-end)으로 정렬	
center	Item을 가운데 정렬	
baseline	Item을 문자 기준선에 정렬	

align-self: 정렬방법;



## ☑ 참고 자료(References)

<https://heropy.blog/2018/11/24/css-flexible-box/>