

Flex : 주로 레이아웃을 잡을 때 쓰는 기능

레이아웃 : 글이나 그림 따위를 효과적으로 정리하고 배치 /

콘텐츠를 잘 정리하고 구조화하는 데 사용되는 중요한 요소

레이아웃 기반 사용 빈도

Table(1990년대~초반) 》 Position(2000년대 초반) 》 Float(2000년대 중반 ~ 후반)

》 Flexbox(2010년대 중반 ~ 현재) 》 Grid(현재 & 미래)

CSS Flexbox(플렉스 박스)는 화면의 공간 분배와 콘텐츠 정렬을 보다 간단하고 유연하게 만들기 위해 도입된 레이아웃 방식

Flex는 기존의 블록( block ) 레이아웃이나 인라인( inline ) 레이아웃과 달리,

레이아웃 기본 개념

-인라인<span> : 사용 가능한 만큼 영역을 사용하여 요소들이 수평으로 쌓임

-블록라벨<div> : 사용 가능한 최대 가로 너비를 사용하여 요소들이 수직으로 쌓임

가로·세로 방향에서 아이템들을 자유롭게 배치하고 정렬할 수 있도록 하는 기능

**\*\*Flex 기초 설명 진행**

---

## Flexbox를 사용할 때의 장점

### 1. 수직 정렬이 간단

기존에는 수직 정렬이 까다로워 다양한 트릭( line-height, table-cell, transform 등 )을 사용해야 했지만, Flexbox에서는 align-items: center;만 설정하면 손쉽게 수직 중앙 정렬이 가능

### 2. 반응형 레이아웃 구성 용이

flex-wrap, flex-shrink, flex-grow 등을 조합하여 다양한 화면 크기에 쉽게 대응

### 3. 동적인 레이아웃

아이템의 순서 변경(order), 줄바꿈(flex-wrap), 아이템 크기 할당(flex-basis) 등으로 레이아웃을 유연하게 구성할 수 있습니다.

**\*\*Flex 사용 장점 및 기존 방식 트릭과 효율성 비교 설명 진행**

## Flex 내용 정리 (실습 전 트릭 설명)

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

## Flexbox 》 contatiner(부모) , item(자식 요소)

```
/* 아래의 『display: flex;』 코드는 flex를 사용하기 위한 기본 전제 코드 */
.container{
  display: flex;
}
```

### 『Flexbox의 종류』

- display                    》 기본 전제 코드
- flex-direction            》 아이тем 배치 방향 설정 row, colmun, “ reverse
- flex-wrap                 》 줄 바꿈 허용 여부
- flex-flow                 》 flex-drection + flex-wrap 단축 속성
- justify-content            》 가로 정렬 방식
- align-items                》 세로 정렬 방식
- align-content             》 줄이 여러 개일 때 세로 정렬

### 『item의 종류』

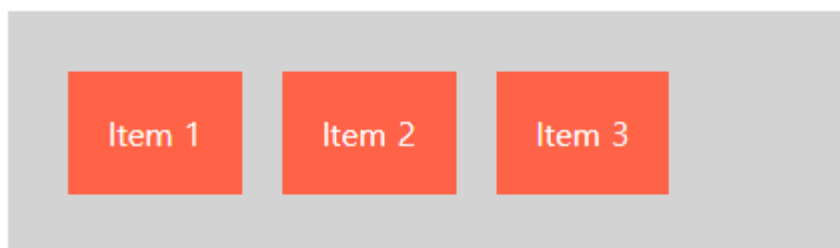
- order                     》 아이тем의 순서 변경 / <div> 순서를 변경
- flex-grow                 》 아이тем이 차지할 비율 설정
- flex-shrink                》 아이тем이 줄어드는 비율
- flex-basis                 》 아이тем의 기본 크기
- flex                        》 flew-grow + flew-shrink + flex-basis 축약형
- align-self                 》 개별 아이тем 정렬

**\*\*Flex 종류에 관한 이해**

## 『Flexbox의 종류』

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Flexbox 기본</title>
</head>
<style>
  .container {
    display: flex; /* Flexbox 활성화 */
    background-color: lightgray;
    padding: 20px;
  }
  .item {
    background-color: tomato;
    color: white;
    padding: 20px;
    margin: 10px;
  }
</style>
<body>
  <div class="container">
    <div class="item">Item 1</div>
    <div class="item">Item 2</div>
    <div class="item">Item 3</div>
  </div>
</body>
```

출력



## flex-direction (아이템 배치 방향 설정)

```
flex-direction: column; /* 가로(row) → 세로(column) 정렬 */
```

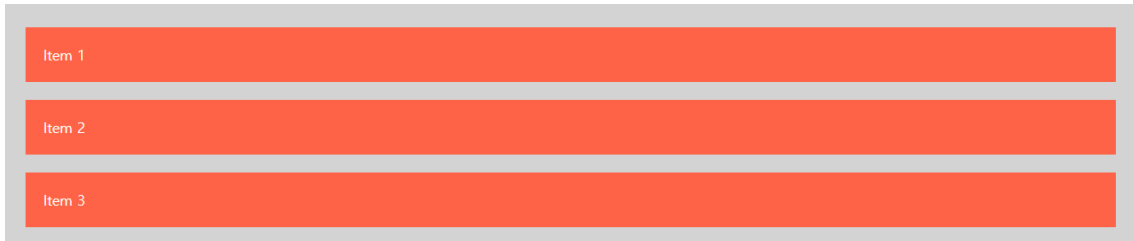
설명 : .container에 가로세로 정렬 column을 넣었을 시  
아이템이 위에서 아래로(column) 배치됨

row, row-reverse, column, column-reverse

그외!

flex-direction: row; 이면 기본값(왼쪽 → 오른쪽)으로 정렬됨

### 출력



---

## flex-wrap (줄 바꿈 허용 여부)

```
flex-wrap: wrap; /* 컨테이너 공간이 부족하면 줄 바꿈 */
```

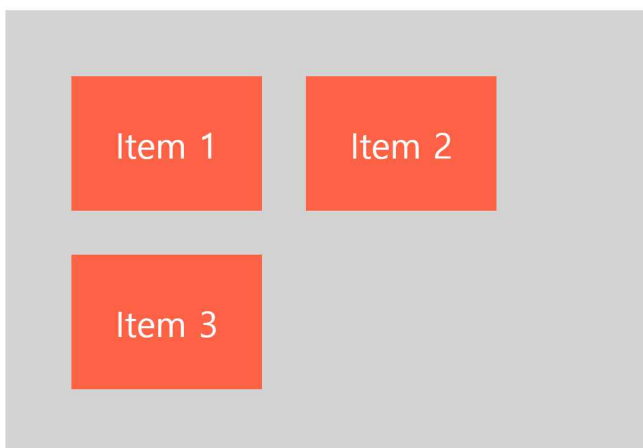
화면 크기가 작아질 때 아이템이 줄 바꿈이 가능한지 여부를 설정함.

wrap을 적용하면 화면 크기가 작아질 때 자동으로 줄 바꿈 됨.

그외!

nowrap이면 줄 바꿈 없이 한 줄에 계속 배치됨.

### 출력



## flex-flow (단축 속성)

```
flex-flow: column wrap; /* 세로(column) 정렬 + 줄 바꿈 허용 */
```

flex-direction + flex-wrap을 합친 속성.

아이템이 세로 정렬되면서 화면이 좁아지면 줄 바꿈 됨.

출력이 애매해서 그냥 flex-direction + flex-wrap 합친 속성이라고 이해해주세요!

출력 x

---

## justify-content (가로 정렬)

```
justify-content: center; /* 가로 중앙 정렬 */
```

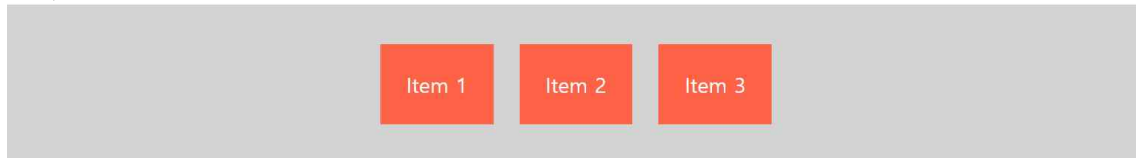
아이템이 컨테이너의 가운데 정렬됨.

flex-start → 왼쪽 정렬 / flex-end → 오른쪽 정렬.

그외!

flex-start, flex-end, center, space-between, space-around, space-evenly 사용 가능

출력



---

## align-items (세로 정렬)

```
align-items: center; /* 세로 중앙 정렬 */  
height: 200px; /* 높이를 주어야 효과가 보임 */
```

그 외!

flex-start, flex-end, center, baseline, stretch 사용 가능.

출력

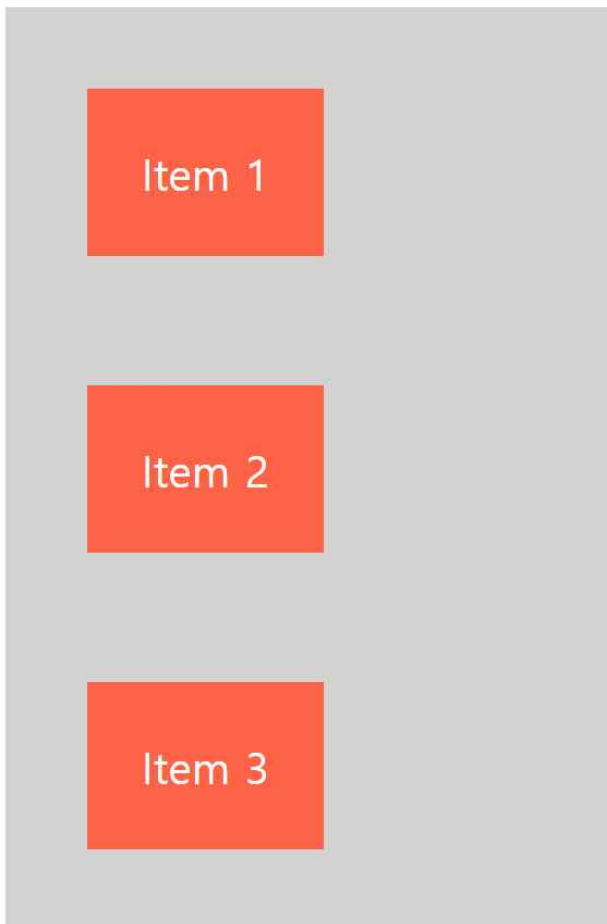


align-content (줄이 여러 개일 때 세로 정렬)

```
flex-wrap: wrap;  
align-content: space-between; /* 줄 간 간격 조절 */  
height: 300px;
```

여러 줄이 존재할 때 간격이 균등하게 배치됨

출력



## 『item의 종류』

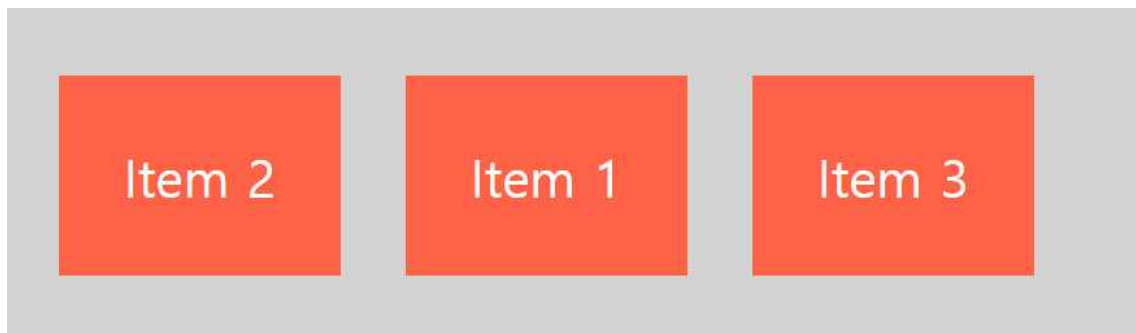
### order (아이템 순서 변경)

```
.item {
  background-color: tomato;
  color: white;
  padding: 20px;
  margin: 10px;
}
.item:nth-child(2){
  order: -1; /* 두 번째 아이템을 가장 앞으로 이동 */
}
```

숫자가 낮을수록 먼저 배치됨.

원래 두 번째 아이템이 맨 앞으로 이동.

출력



---

### flex-grow (아이템이 차지할 비율)

```
flex-grow: 1; /* 모든 아이템이 같은 크기 */
```

flex-grow: 1; 이면 공간을 균등 분배.

모든 아이템이 같은 너비로 배치됨.

출력

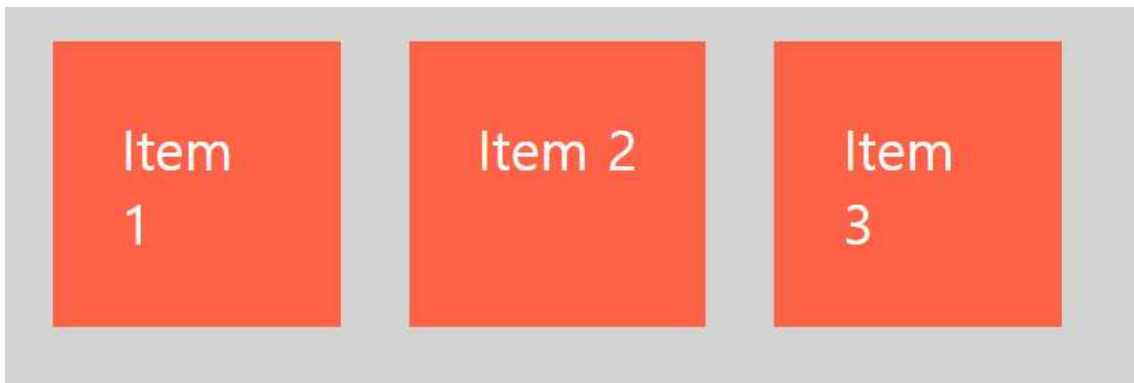


flex-shrink (아이템이 줄어드는 비율)

```
.item:nth-child(2){  
    flex-shrink:0; /* 두 번째 아이템은 줄어들지 않음 */  
}
```

다른 아이템들은 줄어드는데 두 번째 아이템만 그대로 유지됨.

출력



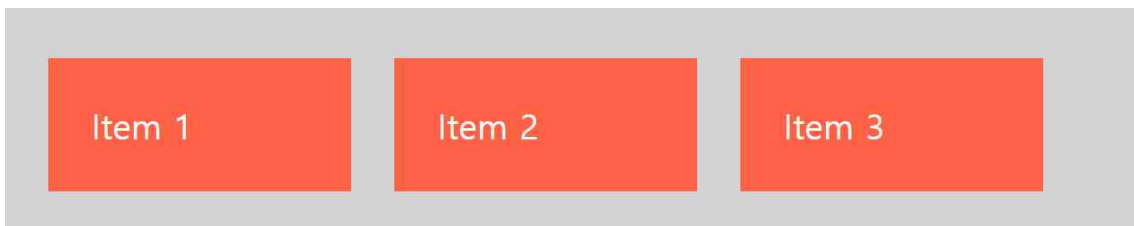
---

flex-basis (아이템 기본 크기)

```
flex-basis:100px; /* 기본 너비 100px */
```

숫자만 바꿔주면 아이템 기본 크기가 수치에 따라 바뀔  
설명할게 없어요..

출력





## flex (축약형)

```
flex:1 1 auto;/* grow 1, shrink 1, basis auto */
```

모든 아이템이 유동적으로 크기가 조정됨.

flex-grow + flex-shrink + flex-basis를 한 번에 설정.

### 출력

기존과 동일 묶음 x

---

## align-self (개별 아이템 정렬)

```
.item:nth-child(2){  
    align-self: flex-end;/* 두 번째 아이템만 아래 정렬 */  
}
```

align-items와 달리 특정 아이템만 정렬 가능.

두 번째 아이템만 아래쪽으로 정렬됨.

### 출력

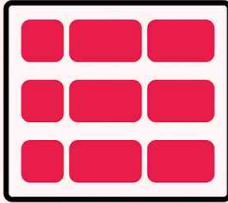


---

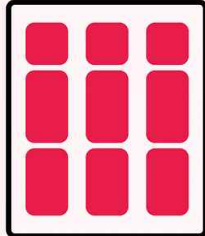
**\*\*Flex에 관련된 것 그림으로 설명 진행 1**

# CSS Flexbox

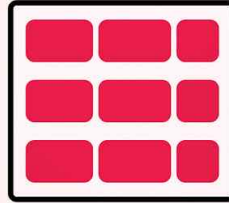
## flex-direction



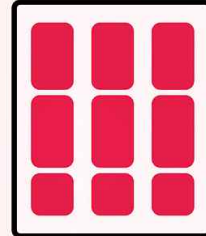
row



column

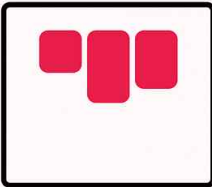


row-reverse

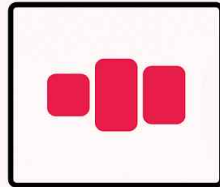


column-reverse

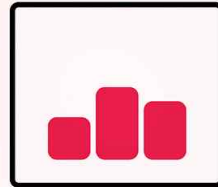
## align-items



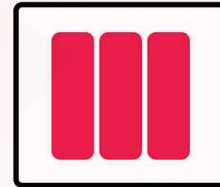
flex-start



center

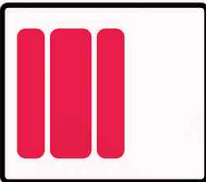


flex-end

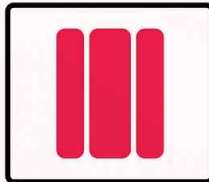


stretch

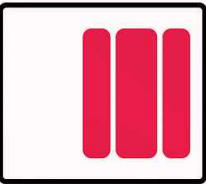
## justify-content



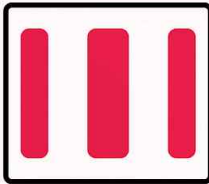
flex-start



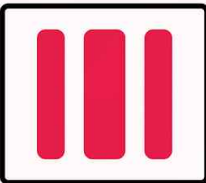
center



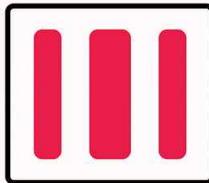
flex-end



space-between

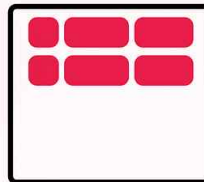


space-around

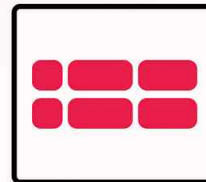


space-evenly

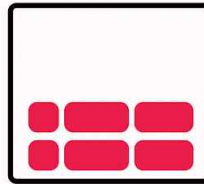
## align-content



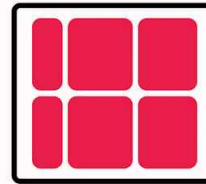
flex-start



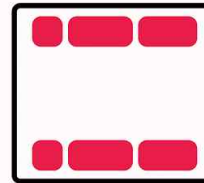
center



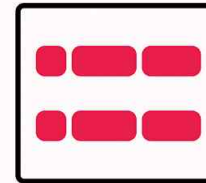
flex-end



stretch



space-between



space-around



\*\*Flex에 관련된 것 그림으로 설명 진행 2

<https://codepen.io/enxaneta/pen/adLPwv>

## 미디어쿼리 개념 & 활용 커리큘럼(Media Query)

### 1. 미디어쿼리(Media Query)의 개념 이해

- 미디어쿼리는 다양한 디바이스 크기에 맞게 스타일을 변경해주는 CSS 기술
- 반응형 웹(Responsive Web Design, RWD)의 핵심 요소
- @media 키워드를 사용하여 특정 조건에서만 CSS를 적용할 수 있음

#### 미디어쿼리 기본 문법

```
@media (조건){  
    /* 특정 조건을 만족할 때 실행할 스타일 */  
}
```

#### 예시

```
@media (max-width:300px){  
    .container {  
        background-color: lightblue;  
    }  
}
```

max-width: 300px

》 301px 이상일 때 기존 배경색

》 299px 이하일 때 background-color: lightblue;

※가장 쉽게 설명

미디어 쿼리는 웹 페이지 창을 조절할 때 그것에 맞게 “스타일 반응”을 해주는 기술

※※미디어쿼리의 간단한 이해

### 2. 다양한 화면 크기(모바일, 태블릿, 데스크톱)에 대응하는 반응형 웹을 구현

max-width: 화면이 작아질 때 스타일 변경 (주로 모바일용)

min-width: 화면이 커질 때 스타일 변경 (주로 데스크톱용)

html 참고

	콘텐츠	패딩	테두리	여백
크기 관련 프로퍼티	width height	padding-top padding-right padding-bottom padding-left	border-top-width border-right-width border-bottom-width border-left-width	margin-top margin-right margin-bottom margin-left
크기 관련 단축 프로퍼티	-	padding	border-width	margin
스타일 관련 프로퍼티	-	-	border-top-style border-right-style border-bottom-style border-left-style	-
스타일 관련 단축 프로퍼티	-	-	border-style	-
색 관련 프로퍼티	-	패딩의 색은 따로 없음. 태그의 배경색으로 칠해짐.	border-top-color border-right-color border-bottom-color border-left-color	여백은 투명, 부모 태그의 배경이 비춰보임
색 관련 단축 프로퍼티	-	-	border-color	-
전체 단축 프로퍼티	-	-	border	-