

12. 반응형 웹과 미디어 쿼리

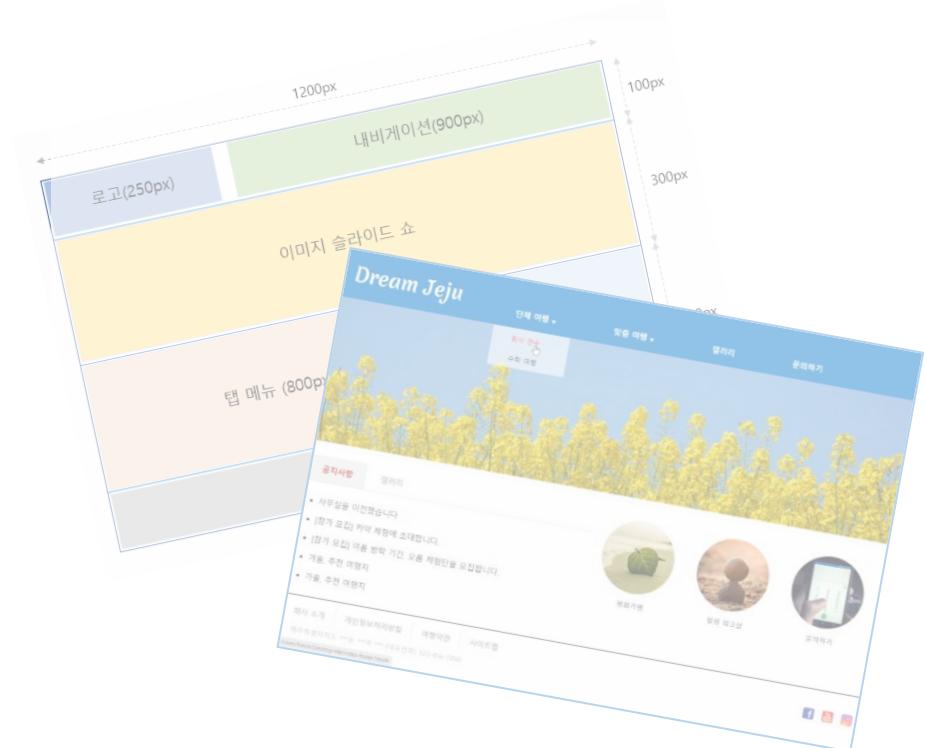
12-1 반응형 웹 알아보기

12-2 미디어 쿼리 알아보기

12-3 그리드 레이아웃 알아보기

12-4 플렉스 박스 레이아웃 알아보기

12-5 CSS 그리드 레이아웃 알아보기



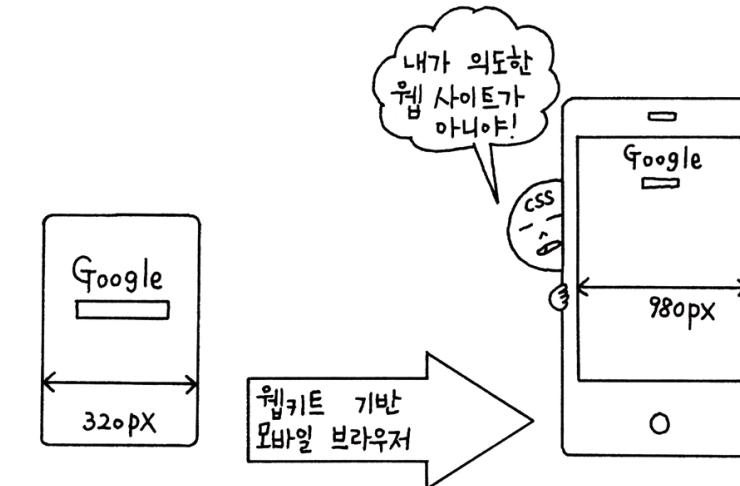
반응형 웹 알아보기

반응형 웹 디자인

- 웹 사이트의 내용을 그대로 유지하면서 다양한 화면 크기에 맞게 웹 사이트를 표시하는 방법
- 다양한 화면 크기의 모바일 기기들이 계속 쏟아져 나오는데 그 때마다 그 크기에 맞춘 사이트를 별도로 제작하는 것은 비효율적
→ 화면 크기에 '반응'해 화면 요소들을 자동으로 바꾸어 사이트를 구현하는 것이 바로 반응형 웹 디자인

모바일 기기를 위한 뷰포트(viewport)

- 뷰포트 : 실제 내용이 표시되는 영역
- PC 화면과 모바일 화면의 픽셀 표시 방법이 다르기 때문에
모바일 화면에서 의도한대로 표시되지 않음
→ 뷰포트를 지정하면 기기 화면에 맞춰 확대/축소해서 내용 표시



반응형 웹 알아보기

뷰포트 지정하기

기본형 <meta name="viewport" content="속성1=값1", "속성2=값2",">

종류	설명	사용 가능한 값	기본값
width	뷰포트 너비	device-width 또는 크기	브라우저 기본값
height	뷰포트 높이	device-height 또는 크기	브라우저 기본값
user-scalable	확대·축소 가능 여부	yes 또는 no (yes는 1로, device-width와 device-height값은 10으로 간주)	yes
initial-scale	초기 확대·축소 값	1~10	1

일반적인 사용법

뷰포트의 너비를 스마트폰 화면 너비에 맞추고 초기 화면 배율을 1로 지정

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

뷰포트 단위

- vw(viewport width): 1vw는 뷰포트 너비의 1%와 같습니다.
- vh(viewport height): 1vh는 뷰포트 높이의 1%와 같습니다.
- vmin(viewport minimum): 뷰포트의 너비와 높이 중에서 작은 값의 1%와 같습니다.
- vmax(viewport maximum): 뷰포트의 너비와 높이 중에서 큰 값의 1%와 같습니다.



Do it! 뷰포트 단위를 사용해 글자 크기 조절하기

예제 파일 12\vw.html

```
(... 생략 ...)  
<style>  
  h1 {  
    font-size: 5vw;  
    text-align: center;  
  }  
</style>  
....  
<h1>안녕하세요?</h1>  
(... 생략 ...)
```

안녕하세요?

안녕하세요?

미디어 쿼리 알아보기

미디어 쿼리(media queries) : 접속하는 장치(미디어)에 따라 특정한 CSS 스타일을 사용하는 방법

미디어 쿼리를 이용해 제작된 사이트들을 모아놓은
<http://mediaqueri.es> 참고

미디어 쿼리 구문

- @media 속성을 사용해 특정 미디어에서 어떤 CSS를 적용할 것인지 지정함
- <style> 태그와 </style> 태그 사이에 사용

기본형 @media [only | not] 미디어 유형 [and 조건] * [and 조건]
① ② ③

- ① only: 미디어 쿼리를 지원하지 않는 웹 브라우저에서는 미디어 쿼리를 무시하고 실행하지 않습니다.
- ② not: not 다음에 지정하는 미디어 유형을 제외합니다. 예를 들어 not tv라고 지정하면 TV를 제외한 미디어 유형에만 적용합니다.
- ③ and: 조건을 여러 개 연결해서 추가할 수 있습니다.

예) 미디어 유형이 screen이면서 최소 너비가 768px이고
최대 너비는 1439px일 경우에 적용할 CSS

```
@media screen and (min-width: 768px) and (max-width: 1439px) {  
    (... 생략 ...)  
}
```

미디어 유형의 종류

종류	설명
all	모든 미디어 유형에서 사용할 CSS를 정의합니다.
print	인쇄 장치에서 사용할 CSS를 정의합니다.
screen	컴퓨터 스크린에서 사용할 CSS를 정의합니다. 스마트폰의 스크린도 포함됩니다.
tv	음성과 영상이 동시에 출력되는 TV에서 사용할 CSS를 정의합니다.
aural	음성 합성 장치(주로 화면을 읽어 소리로 출력해 주는 장치)에서 사용할 CSS를 정의합니다.
braille	점자 표시 장치에서 사용할 CSS를 정의합니다.
handheld	패드(pad)처럼 손에 들고 다니는 장치를 위한 CSS를 정의합니다.
projection	프로젝터를 위한 CSS를 정의합니다.
tty	디스플레이 기능이 제한된 장치에 맞는 CSS를 정의합니다. 이런 장치에서는 픽셀(px) 단위를 사용할 수 없습니다.
embossed	점자 프린터에서 사용할 CSS를 정의합니다.

미디어 쿼리 알아보기

미디어 쿼리의 다양한 조건

웹 문서의 가로 너비와 세로 높이(뷰포트)

실제 웹 문서 내용이 나타나는 영역의 너비와 높이를 조건으로 사용

종류	설명
width, height	웹 페이지의 가로 너비, 세로 높이
min-width, min-height	웹 페이지의 최소 너비, 최소 높이
max-width, max-height	웹 페이지의 최대 너비, 최대 높이

단말기의 가로 너비와 세로 높이

대부분의 단말기 해상도와 실제 브라우저의 너비가 다르다는 점에 주의

종류	설명
device-width, device-height	단말기의 가로 너비, 세로 높이
min-device-width, min-device-height	단말기의 최소 너비, 최소 높이
max-device-width, max-device-height	단말기의 최대 너비, 최대 높이

예) 화면 너비가 1440px 이상일 때

```
@media screen and (min-width: 1440px) { /* 너비가 최소 1440px인 화면용 스타일 */
  (... 생략 ...)
}
```

화면 회전

스마트폰이나 태블릿에서 기기를 가로나 세로로 돌려보는지 확인

종류	설명
orientation: portrait	단말기의 세로 모드
orientation: landscape	단말기의 가로 모드

예) iPhoneX 를 가로로 돌릴 경우

```
@media screen and (min-device-width: 375px) and (min-device-height: 812px) {
  (... 생략 ...)
}
```

미디어 쿼리 알아보기

미디어 쿼리 중단점

- **중단점(breakpoint)** : 서로 다른 CSS를 적용할 화면 크기
- 대부분 기기의 화면 크기 기준.
- 모든 기기를 반영할 수 없기 때문에 스마트폰과 태블릿, 데스크톱 정도로 구분
- 모바일 퍼스트(mobile first) : 모바일 기기 레이아웃을 기본으로 작성 → 태블릿 & PC 레이아웃 작성

일반적인 사용 예

(미디어 쿼리 중단점은 개발자나 작업 조건에 따라 달라질 수 있다.)

- **스마트폰**: 모바일 페이지는 미디어 쿼리를 사용하지 않고 기본 CSS로 작성. 만일 스마트폰의 방향까지 고려해서 제작한다면 min-width의 세로와 가로를 각각 portrait 320px, landscape 480px로 지정.
- **태블릿**: 세로 크기가 768px 이상이면 태블릿으로 지정. 가로 크기는 데스크톱과 똑같이 1024px 이상으로 지정.
- **데스크톱**: 화면 크기가 1024px 이상이면 데스크톱으로 설정.

미디어 쿼리 알아보기

미디어 쿼리 적용하기

1. 외부 CSS 파일 연결

방법1) <link> 태그 사용하기

<head>와 </head> 사이에 삽입

기본형 <link rel="stylesheet" media="미디어 쿼리 조건" href="css 파일 경로">

예) <link rel="stylesheet" media="print" href="css/print.css">

방법2) @import 구문 사용하기

<style>와 </style> 사이에 삽입

기본형 @import url(css 파일 경로) 미디어 쿼리 조건

예) @import url("css/tablet.css") only screen and (min-width: 321px) and (max-width: 768px);

2. 웹 문서에서 직접 정의하기

방법1) <style> 태그 안에서 media 속성 사용하기

기본형 <style media="<조건>"> {
}
/<style>

예)

<style media="screen and (max-width: 320px)">
body {
background-color: orange;
}
</style>

방법2) @media문 사용하기

기본형 <style>
@media <조건> {
}
/<style>

예)

<style>
@media screen and (max-width: 320px) {
body {
background-color: orange;
}
}
</style>

그리드 레이아웃 알아보기

그리드 레이아웃이란

- 반응형 웹 디자인에서 웹 문서 요소를 배치하는 기준
- 웹 사이트 화면을 여러 개의 칼럼(column)으로 나눈 후 웹 요소를 배치
- 화면을 규칙적으로 배열하므로 레이아웃을 일관성 있게 유지할 수 있음

그리드 레이아웃의 특징

- 시각적으로 안정된 디자인
- 업데이트가 편한 웹 디자인 구성
- 요소를 자유롭게 배치

이지스퍼블리싱	이지스매체	로그인	회원가입	도서 찾기	도서 목록	FAQ												
회사소개 이지스퍼블리싱(주) 소개	회사소개 이지스퍼블리싱(주) 소개	도서	자료실	질문답변	동영상 강의													
이지스퍼블리싱(주) 소개																		
																		
<p>“사람을 구체적으로 도와주는 책”</p> <p>이지스퍼블리싱(주)에서는 ‘사람에게 구체적으로 도움이 되는 책’을 만난다는 데에 가치가 담겨 있습니다. 2010년 5월 출범한 이지스퍼블리싱(주)는 그에 두 명의 창립자를 출연합니다. IT 컨설팅 도서와 학습 판례 도서입니다. IT 교육과 사진 책을 출판하시는 이지스퍼블리싱(주)와 출판과 저네과 도서는 이지스퍼블리싱(주)로 출간하고 있습니다.</p>																		
<p>이지스퍼블리싱의 미션</p> <p>사람에게 구체적으로 도움을 주는 책 우리나라 학생과 서는 사람들에게 도움이 되고 싶습니다. 우리는 책을 출판하는 것에 절대할 것이라고. “이 책은 사람들에게 도움이 됩니다.”</p> <p>정보의 지름길을 만드는 책으로 원하는 곳으로 가도록 도와주는 책. 손에 꼽히는 벅书记을 만들 수 있도록 도와야 되는 책을 만들고 싶습니다.</p>																		
<p>조직판</p> <p>이지스퍼블리싱(주)에 참여하는 모든 사람에게 도움과 주고 싶습니다. 쉽게 참여해시 만드는 카페, 세미나 등을 주는 후원 활동 노미터 책을 만들고, 우리 책의 미래를 원하는 대안 스스로 만드는 터전, 기여한 바, 긍정한 문예가 이루어지고げる 문예를 함께 나누는 곳, 그곳에 바로 이지스퍼블리싱(주)입니다.</p>																		
<p>인재 채용</p> <p>이력서와 자기소개서 보낼 곳 : easy@easypublish.co.kr</p>																		
<p>관련 사이트 안내</p> <table border="1"><tr><td>이지스퍼블리싱 홈페이지</td><td>→</td></tr><tr><td>이지스에듀 홈페이지</td><td>→</td></tr><tr><td>페이스북 페이지</td><td>→</td></tr><tr><td>이지스퍼블리싱 Do 선스티디카페</td><td>→</td></tr><tr><td>이지스매체 카페</td><td>→</td></tr><tr><td>바빠 이지트·이지스 Benson 만나기</td><td>→</td></tr></table>							이지스퍼블리싱 홈페이지	→	이지스에듀 홈페이지	→	페이스북 페이지	→	이지스퍼블리싱 Do 선스티디카페	→	이지스매체 카페	→	바빠 이지트·이지스 Benson 만나기	→
이지스퍼블리싱 홈페이지	→																	
이지스에듀 홈페이지	→																	
페이스북 페이지	→																	
이지스퍼블리싱 Do 선스티디카페	→																	
이지스매체 카페	→																	
바빠 이지트·이지스 Benson 만나기	→																	

그리드 레이아웃 알아보기

그리드 레이아웃을 만드는 방법

1) 플렉스 박스 레이아웃(플렉서블 박스 레이아웃)

- 수평 방향이나 수직 방향 중 하나를 주축으로 정하고 박스를 배치
- 여유 공간이 생길 경우 너비나 높이를 적절하게 늘리거나 줄일 수 있음



2) CSS 그리드 레이아웃

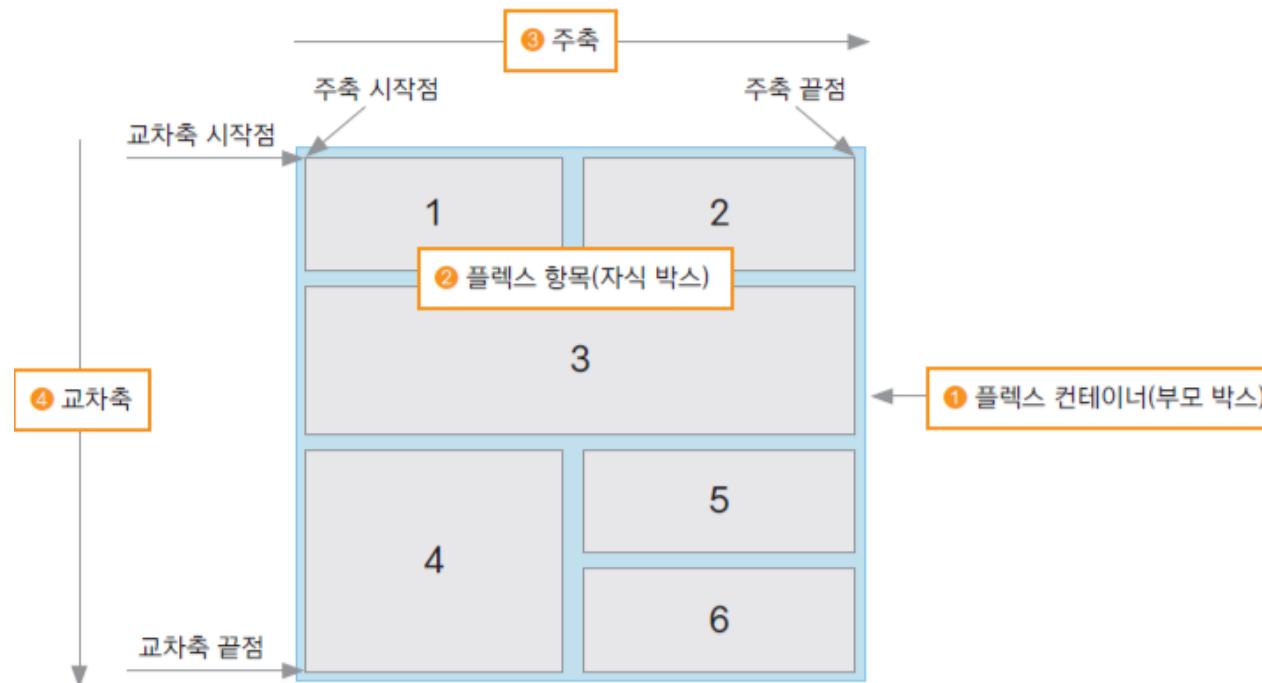
- 최근에 등장한 CSS 표준
- 수평 방향이나 수직 방향 어디로든 배치 가능
- 마치 레고 블록을 끼워 맞추듯 요소를 배치할 수 있음



플렉스 박스 레이아웃

플렉스 박스 레이아웃(flex box layout)

- 그리드 레이아웃을 기본으로, 플렉스 박스를 원하는 위치에 배치하는 것.
- 여유 공간에 따라 너비나 높이, 위치를 자유롭게 변형할 수 있음



① **플렉스 컨테이너(부모 박스)**: 플렉스 박스 레이아웃을 적용할 대상을 묶는 요소입니다.

② **플렉스 항목(자식 박스)**: 플렉스 박스 레이아웃을 적용할 대상으로 그림 12-18에서 1~6까지 작은 박스들이 모두 해당합니다.

③ **주축(main axis)**: 플렉스 컨테이너 안에서 플렉스 항목을 배치하는 기본 방향입니다. 기본적으로 왼쪽에서 오른쪽이며 수평 방향으로 배치합니다. 플렉스 항목의 배치가 시작되는 위치를 '주축 시작점', 끝나는 위치를 '주축 끝점'이라고 합니다.

④ **교차축(cross axis)**: 주축과 교차하는 방향을 말하며 기본적으로 위에서 아래로 배치합니다. 플렉스 항목의 배치가 시작되는 위치를 '교차축 시작점', 끝나는 위치를 '교차축 끝점'이라고 합니다.

플렉스 박스 레이아웃

Flex Container

- Flex Container를 위한 속성

속성	의미
display	Flex Container를 정의
flex-flow	flex-direction와 flex-wrap의 단축 속성
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)

플렉스 박스 레이아웃

플렉서블 박스 레이아웃 기본 속성

display 속성

배치 요소들을 감싸는 부모 요소를 플렉스 컨테이너로 지정

종류	설명
flex	컨테이너 안의 플렉스 항목을 블록 레벨 요소로 배치합니다.
inline-flex	컨테이너 안의 플렉스 항목을 인라인 레벨 요소로 배치합니다.

flex-direction 속성

플렉스 항목의 배치를 위해 주축과 방향 지정.

종류	설명
row	주축을 가로로 지정하고 왼쪽에서 오른쪽으로 배치합니다. 기본값입니다.
row-reverse	주축을 가로로 지정하고 반대로 오른쪽에서 왼쪽으로 배치합니다.
column	주축을 세로로 지정하고 위쪽에서 아래쪽으로 배치합니다.
column-reverse	주축을 세로로 지정하고 아래쪽에서 위쪽으로 배치합니다.

Do it! 플렉스 항목 배치하기

예제 파일 12\flex-1.html

(... 생략 ...)

<style>

```
.container { display: flex; } /* 플렉스 컨테이너 지정 */
```

.....

```
#opt1 { flex-direction: row; } /* 왼쪽에서 오른쪽으로 */
```

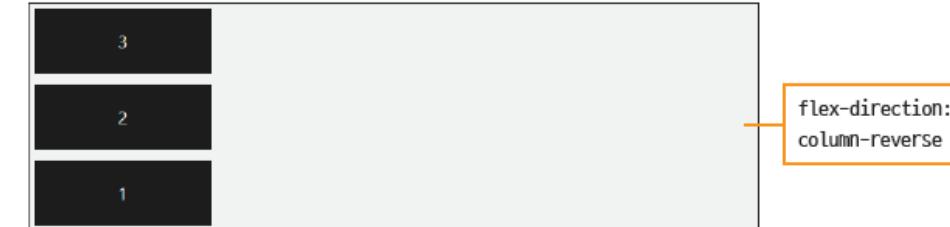
```
#opt2 { flex-direction: row-reverse; } /* 오른쪽에서 왼쪽으로 */
```

```
#opt3 { flex-direction: column; } /* 위에서 아래로 */
```

```
#opt4 { flex-direction: column-reverse; } /* 아래에서 위로 */
```

(... 생략 ...)

column:reverse



플렉스 박스 레이아웃

플렉서블 박스 레이아웃 기본 속성

flex-wrap 속성

플렉스 항목을 한 줄 또는 여러 줄로 배치

| 종류 | 설명 |
|--------------|------------------------------------|
| nowrap | 플렉스 항목을 한 줄에 표시합니다. 기본값입니다. |
| wrap | 플렉스 항목을 여러 줄에 표시합니다. |
| wrap-reverse | 플렉스 항목을 여러 줄에 표시하되, 시작점과 끝점이 바뀝니다. |

flex-flow 속성

- 플렉스 배치 방향과 여러 줄 배치를 한꺼번에 지정
- 기본 값은 flex-flow:row no-wrap



Do it! 플렉스 항목의 배치 방향과 줄바꿈 적용하기

예제 파일 12\flex-3.html

(... 생략 ...)

<style>

```
.....  
#opt1 { flex-flow: row wrap; } /* 왼쪽에서 오른쪽, 여러 줄 */  
#opt2 { flex-flow: row nowrap; } /* 왼쪽에서 오른쪽, 한 줄 */  
(... 생략 ...)
```



플렉스 박스 레이아웃

플렉스 항목 배치를 위한 속성

justify-content 속성

플렉스 항목을 주축 방향으로 배치할 때의 배치 기준

| 종류 | 설명 |
|---------------|--|
| flex-start | 주축의 시작점에 맞춰 배치합니다. |
| flex-end | 주축의 끝점에 맞춰 배치합니다. |
| center | 주축의 중앙에 맞춰 배치합니다. |
| space-between | 첫 번째 항목과 끝 항목을 주축의 시작점과 끝점에 배치한 후 나머지 항목은 그 사이에 같은 간격으로 배치합니다. |
| space-around | 모든 항목을 주축에 같은 간격으로 배치합니다. |



Do it! 주축에서 플렉스 항목 간의 간격 적용하기

예제 파일 12\flex-4.html

(... 생략 ...)

<style>

.....

```
#opt1 { justify-content: flex-start; }           /* 주축 시작점 기준 */  
#opt2 { justify-content: flex-end; }             /* 주축 끝점 기준 */  
#opt3 { justify-content: center; }               /* 주축 중앙 기준 */  
#opt4 { justify-content: space-between; }         /* 시작점과 끝점 배치한 후 같은 간격 */  
#opt5 { justify-content: space-around; }          /* 전체 항목 같은 간격 */
```

(... 생략 ...)

align-items:center



justify-content:
flex-start



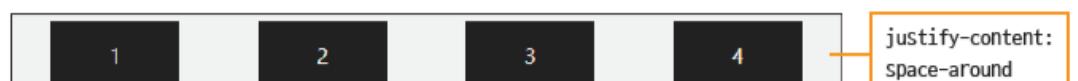
justify-content:
flex-end



justify-content:
center



justify-content:
space-between



justify-content:
space-around

플렉스 박스 레이아웃

플렉스 항목 배치를 위한 속성

align-items 속성, align-self 속성

교차축을 기준으로 하는 배치 방법 조절

교차축에서 특정 항목만 지정하고 싶다면 **align-self 속성** 사용

해당 항목 자체에 사용

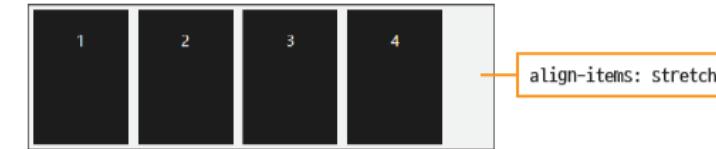
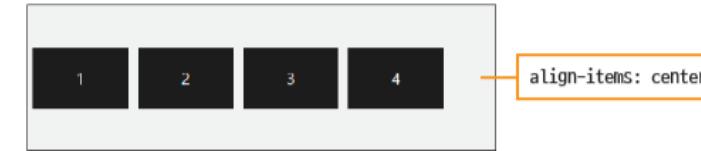
| 종류 | 설명 |
|------------|------------------------------|
| flex-start | 교차축의 시작점에 맞춰 배치합니다. |
| flex-end | 교차축의 끝점에 맞춰 배치합니다. |
| center | 교차축의 중앙에 배치합니다. |
| baseline | 교차축의 문자 기준선에 맞춰 배치합니다. |
| stretch | 플렉스 항목을 늘려 교차축에 가득 차게 배치합니다. |

Do it! 플렉스 박스에서 교차축 정렬 방법 지정하기

예제 파일 12\flex-5.html

```
(... 생략 ...)

<style>
    ...
    #opt1 { align-items: flex-start; }           /* 주축 시작점 기준으로 배치 */
    #opt2 { align-items: flex-end; }             /* 주축 끝점 기준으로 배치 */
    #opt3 { align-items: center; }               /* 주축 중앙 기준으로 배치 */
    #opt4 { align-items: baseline; }             /* 문자 기준선에 맞춰 배치 */
    #opt5 { align-items: stretch; }              /* 항목을 늘려 교차축에 가득 차게 배치 */
    ...
</style>
```



플렉스 박스 레이아웃

플렉스 항목 배치를 위한 속성

align-content 속성

플렉스 항목이 여러 줄로 표시될 때 교차 축 기준의 배치 방법 지정

| 종류 | 설명 |
|---------------|---|
| flex-start | 교차축의 시작점에 맞춰 배치합니다. |
| flex-end | 교차축의 끝점에 맞춰 배치합니다. |
| center | 교차축의 중앙에 맞춰 배치합니다. |
| space-between | 첫 번째 항목과 끝 항목을 교차축의 시작점과 끝점에 맞추고 나머지 항목은 그 사이에 같은 간격으로 배치합니다. |
| space-around | 모든 항목을 교차축에 같은 간격으로 배치합니다. |
| stretch | 플렉스 항목을 늘려서 교차축에 가득 차게 배치합니다. |



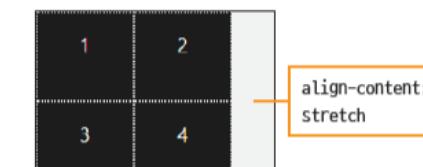
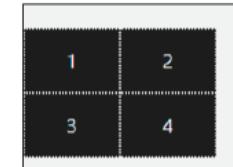
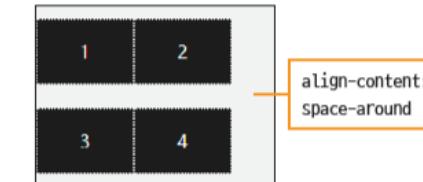
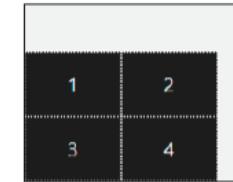
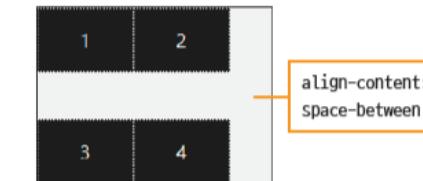
Do it! 여러 줄일 때 교차축에서 플렉스 항목 간의 간격 지정하기

예제 파일 12\flex-7.html

(... 생략 ...)

<style>

```
.container {  
    ...  
    display: flex; /* 플렉스 컨테이너 지정 */  
    flex-flow: row wrap; /* 왼쪽에서 오른쪽, 여러 줄 표시 */  
}  
#opt1 { align-content: flex-start; } /* 교차축 시작점 기준 */  
#opt2 { align-content: flex-end; } /* 교차축 끝점 기준 */  
#opt3 { align-content: center; } /* 교차축 중앙 기준 */  
#opt4 { align-content: space-between; } /* 시작점과 끝점에 배치한 뒤 중간 항목  
    은 같은 간격으로 배치 */  
#opt5 { align-content: space-around; } /* 전체 항목을 같은 간격으로 배치 */  
#opt6 { align-content: stretch; } /* 항목을 늘려 교차축에 가득 차게 배치 */  
(... 생략 ...)
```



플렉스 박스 레이아웃

Flex Items

- Flex Items를 위한 속성

| 속성 | 의미 |
|-------------|---|
| order | Flex Item의 순서를 설정 |
| flex | flex-grow, flex-shrink, flex-basis의 단축 속성 |
| 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 |

플렉스 박스 레이아웃

Item의 순서를 설정하기 위한 속성

order 속성

- 기본형

order: 순서;

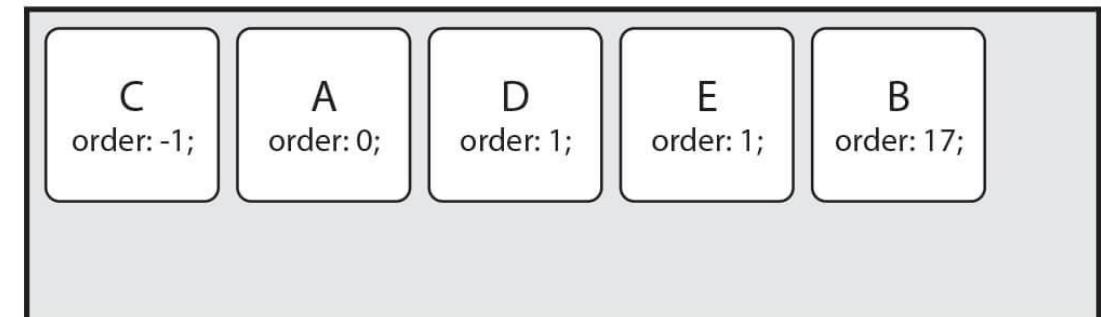
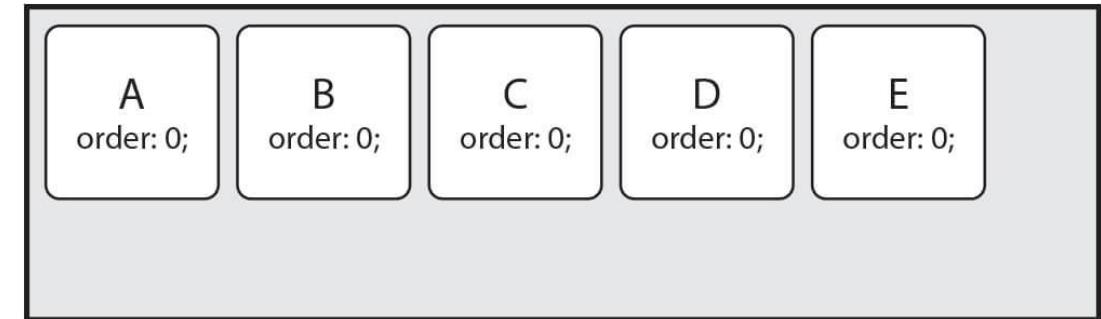
- 특징

Item에 숫자를 지정하고 숫자가 클수록 순서가 밀립니다.

음수가 허용됩니다.

HTML 구조와 상관없이 순서를 변경할 수 있기 때문에 유용합니다.

| 값 | 의미 | 기본값 |
|----|--------------|-----|
| 숫자 | Item의 순서를 설정 | 0 |



플렉스 박스 레이아웃

Item의 증가 너비 비율을 설정하기 위한 속성

flex-grow 속성

- 기본형

flex-grow: 증가너비;

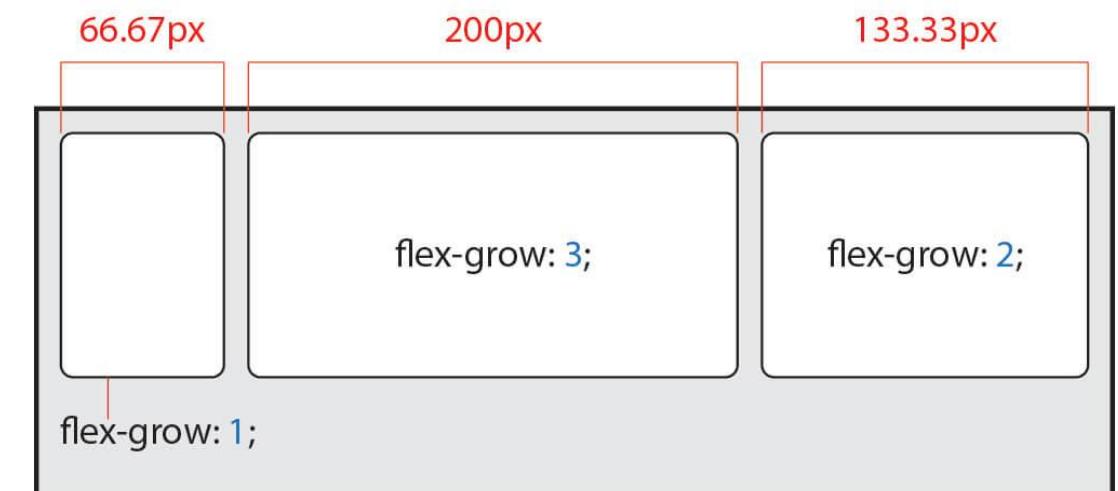
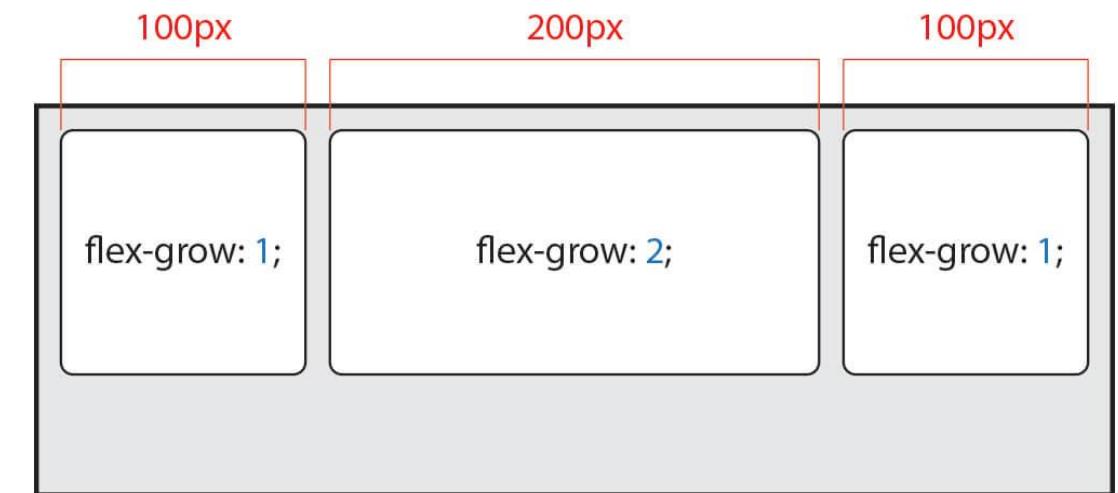
- 특징

Item의 증가 너비 비율을 설정합니다.

숫자가 크면 더 많은 너비를 가집니다.

Item이 가변 너비가 아니거나, 값이 0일 경우 효과가 없습니다.

| 값 | 의미 | 기본값 |
|----|--------------------|-----|
| 숫자 | Item의 증가 너비 비율을 설정 | 0 |



플렉스 박스 레이아웃

Item이 감소하는 너비의 비율을 설정하기 위한 속성

flex-shrink 속성

- 기본형

`flex-shrink: 감소너비;`

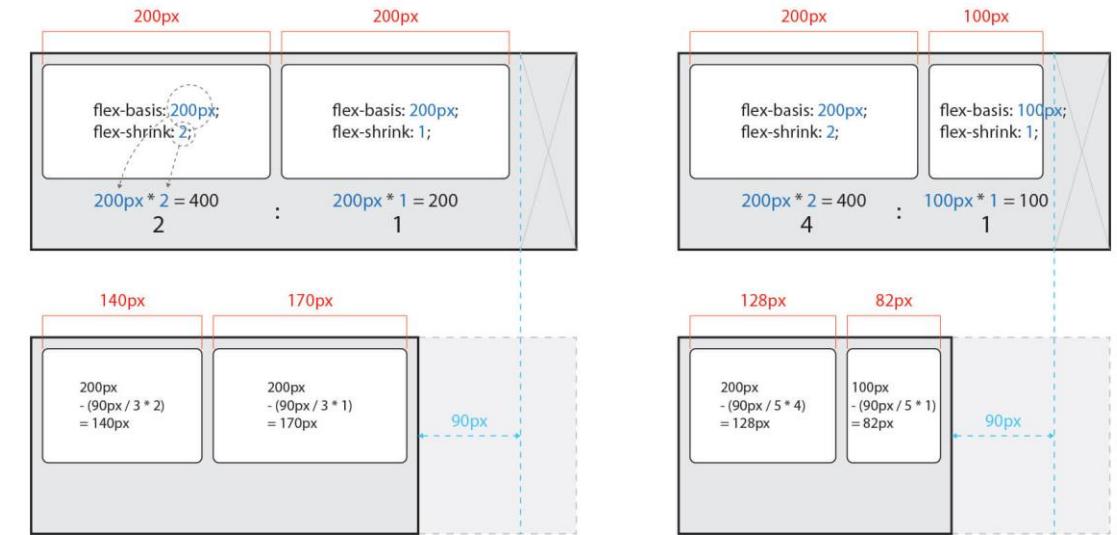
- 특징

Item이 감소하는 너비의 비율을 설정합니다.

숫자가 크면 더 많은 너비가 감소합니다.

Item이 가변 너비가 아니거나, 값이 0일 경우 효과가 없습니다.

| 값 | 의미 | 기본값 |
|----|--------------------|-----|
| 숫자 | Item의 감소 너비 비율을 설정 | 1 |



플렉스 박스 레이아웃

Item의 (공간 배분 전) 기본 너비를 설정하기 위한 속성

flex-basis 속성

- 기본형

flex-basis: 기본너비;

- 특징

flex-basis가 auto인 경우 width, height 속성이 우선한다

flex-basis가 auto가 아닌 경우, flex-basis 속성이 우선한다.

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



* 쉽게 기억하는 팁

- flex-basis는 기준 크기를 정하는 애
- auto는 다른 속성에 맡길게라는 뜻 → width가 있으면 그걸 따라감.
- 직접 값이 있으면(100px 등) → 내가 정한다, width는 무시.

플렉스 박스 레이아웃

Item의 (증가, 감소, 기본)를 설정하기 위한 속성

flex 속성

- 기본형

flex: 증가너비 감소너비 기본너비;

| 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 |

- 예시 구문

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

- flex-grow를 제외한 개별 속성은 가능
- flex: 1; => flex-grow: 1; / flex-shrink: 1; / flex-basis: 0;
=> Flex-basis의 기본값은 auto지만, 단축 속성인 flex에서 그 값을 생략할 경우 0 적용
- 결론 : flex: 1; 혹은 flex: 1 1; 은 flex: 1 1 0;이 됨

플렉스 박스 레이아웃

Item의 (증가, 감소, 기본)를 설정하기 위한 속성

flex 속성

- 기본형

flex: 증가너비 감소너비 기본너비;

(1) 값이 한 개일 때,

- 단위가 없으면 flex-grow 값이 된다.
- 단위가 있으면 flex-basis 값이 된다.

(2) 값이 두 개일 때,

- 첫번째 값은 단위가 없는 숫자여야 한다. 또한 첫번째 값은 flex-grow가 된다.
- 두번째 값은 단위가 없으면 flex-shrink, 단위가 있거나 auto면 flex-basis가 된다.

(3) 값이 세 개일 때,

- 첫번째 값은 flex-grow (단위 없어야 함)
- 두번째 값은 flex-shrink (단위 없어야 함)
- 세번째 값은 flex-basis 값이 된다. (단위 있거나 0 또는 auto여야 함)

플렉스 박스 레이아웃

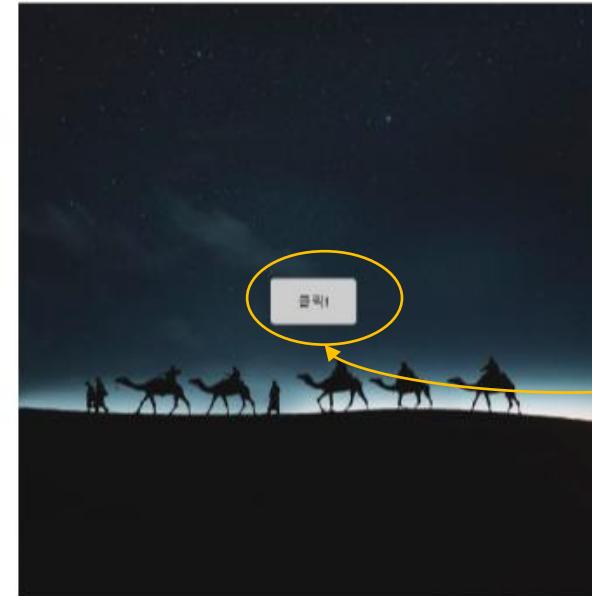
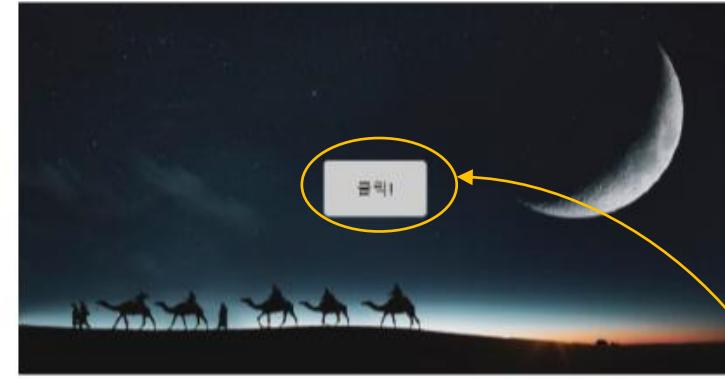
플렉스 박스 레이아웃을 사용해 화면 중앙에 배치하기

플렉스 박스 레이아웃을 가장 많이 사용하는 예

 **Do it!** 플렉스 박스 레이아웃을 사용해 화면 중앙에 배치하기 예제 파일 12\center.html

```
(... 생략 ...)

<style>
  * {
    margin: 0;
    box-sizing: border-box;
  }
  body {
    background: url('images/bg5.jpg') no-repeat left top fixed;
    background-size: cover;
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
  }
  button {
    .....
  }
  .....
<button>클릭!</button>
(... 생략 ...)
```

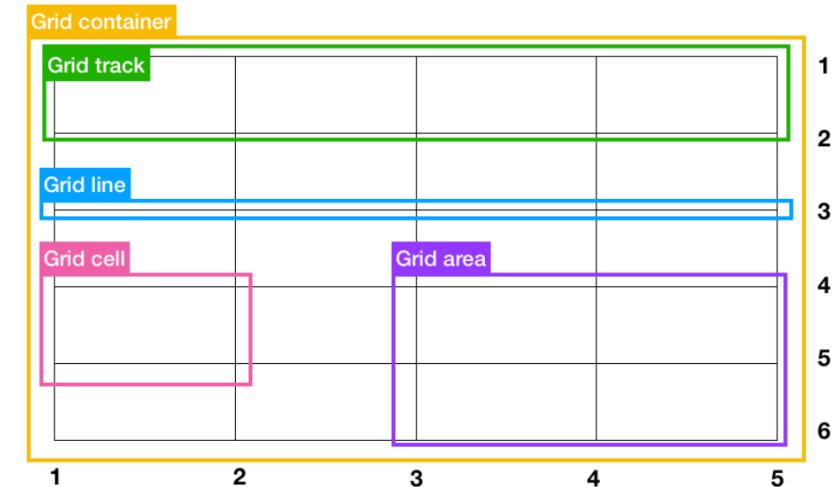


화면 크기에 상관없이
버튼은 항상 화면 중앙
에 표시됨

CSS 그리드 레이아웃

CSS 그리드 레이아웃

- 플렉스 박스 레이아웃은 주축/교차축 개념이 있지만 css 그리드 레이아웃은 양쪽 방향 모두 사용
(플렉스 그리드 레이아웃은 1차원, css 그리드 레이아웃은 2차원이라고도 함)
- 줄(row)과 칼럼(column)으로 화면을 구성하고, 줄 사이의 여백, 칼럼 사이의 여백을 조절.



- Grid container:** 전체 Grid layout을 감싸는 역할을 수행한다. `display: grid | inline-grid` 속성을 통해 지정할 수 있다.
- Grid item:** Grid container에 속해있는 하위 DOM 요소를 뜻한다.
- Grid track:** Grid layout에 존재하는 행 또는 열을 의미한다. Grid track의 개수는 명시적으로 지정할 수도, 암묵적으로 늘어나게 할 수도 있다.
- Grid line:** Grid track을 구분하는 선을 의미한다. 선의 번호는 위에서 아래로(↑), 왼쪽에서 오른쪽으로(→) 매겨진다. 그리드 라인의 번호는 **1부터 시작된다**는 점을 주의하자.
- Grid cell:** 그리드 레이아웃에서 가장 작은 단위 요소이며, 테이블의 셀과 유사하다.
- Grid area:** 다수의 Grid cell로 이루어진 영역을 뜻한다. Grid area는 항상 사각형의 모양을 가져야한다(ㄴ자, ㄱ자 형태 불가능).

CSS 그리드 레이아웃

CSS 그리드 레이아웃 항목을 배치하는 속성

display 속성

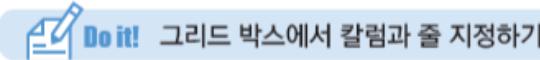
배치 요소들을 감싸는 부모 요소를 그리드 컨테이너로 지정

| 종류 | 설명 |
|-------------|-------------------------------|
| grid | 컨테이너 안의 항목을 블록 레벨 요소로 배치합니다. |
| inline-grid | 컨테이너 안의 항목을 인라인 레벨 요소로 배치합니다. |

grid-template-columns, grid-template-rows 속성

칼럼/줄의 크기와 개수 지정

- grid-template-columns : 그리드 컨테이너 안의 칼럼 개수와 너비값
- grid-template-rows: 그리드 컨테이너 안의 줄 개수와 너비값



그리드 박스에서 칼럼과 줄 지정하기

예제 파일 12\grid-1.html

(... 생략 ...)

<style>

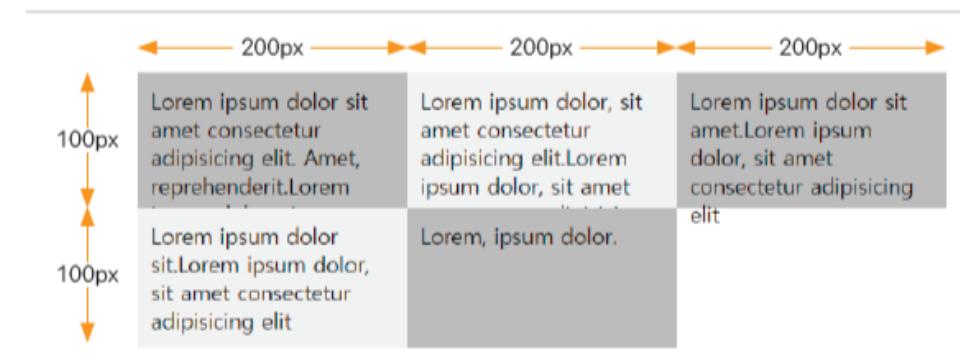
```
#wrapper {  
    display: grid; /* 그리드 컨테이너 지정 */  
    grid-template-columns: 200px 200px 200px; /* 너비가 200px인 칼럼 3개 */  
    grid-template-rows: 100px; /* 줄 높이 100px */  
}
```

.....

```
<div id="wrapper">  
    <div class="items">Lorem ipsum dolor sit.....</div>  
    <div class="items">Lorem .....</div>  
    <div class="items">Lorem .....</div>  
    <div class="items">Lorem .....</div>  
    <div class="items">Lorem .....</div>  
</div>
```

(... 생략 ...)

columns



CSS 그리드 레이아웃

상대적인 크기를 지정하는 fr 단위

칼럼/줄의 크기를 지정할 때 px 단위는 반응형 웹 디자인에 적합하지 않음 → 상대적인 크기를 지정하는 **fr(fraction)** 단위 사용

예) 너비의 비율이 2:1:2인 칼럼 3개를 배치한다면

```
grid-template-columns: 2fr 1fr 2fr;
```

값이 반복된다면 repeat() 함수

똑같은 값을 여러 번 반복한다면 내장 함수 **repeat()** 함수 사용

```
grid-template-columns: 1fr 1fr 1fr;
```

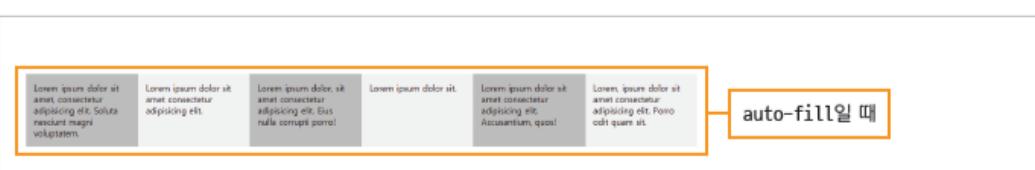


```
grid-template-columns: repeat(3, 1fr);
```

최솟값과 최댓값 지정하는 minmax() 함수

minmax(300px, auto)

minmax(300px, 1fr)



자동으로 칼럼 개수를 조절하는 auto-fit, auto-fit

칼럼 너비와 함께 auto-fit이나 auto-fill을 지정하면 화면 너비에 따라 칼럼 개수를 조절할 수 있음

- auto-fit : 남는 공간 없이 꽉 채우기
- auto-fill : 칼럼의 최소 너비까지만 표시하고 남는 공간은 그대로 둠

예) 너비 200px인 칼럼을 화면에 가득찰 때까지 배치

```
grid-template-columns: repeat(auto-fit, 200px);
```

Screenshot of a browser showing a grid with 6 items. A callout box labeled "auto-fit일 때" points to the right edge of the grid container, indicating that the grid has adjusted its width based on the available space.

```
<style>
  #wrapper1 {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
    margin-bottom: 20px;
  }
  #wrapper2 {
    display: grid;
    grid-template-columns: repeat(auto-fill, minmax(200px, 1fr));
  }

```

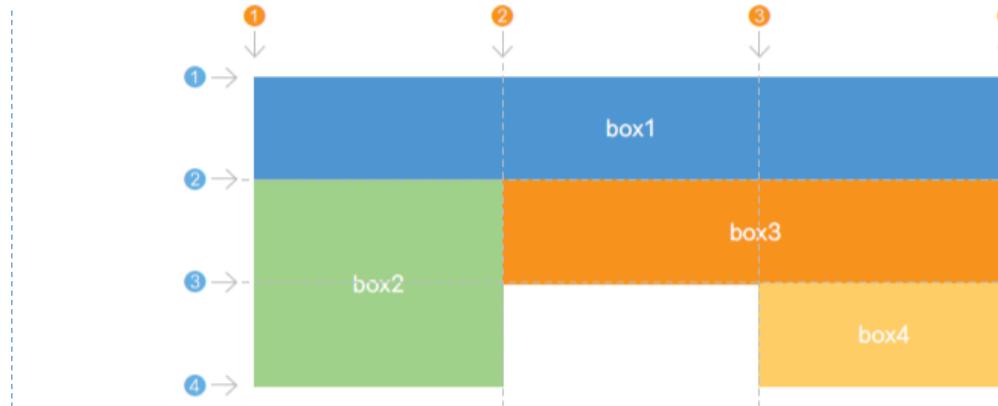
CSS 그리드 레이아웃

그리드 라인을 사용해 배치하기

- CSS 그리드 레이아웃에는 눈에 보이지 않는 그리드 라인이 포함되어 있음
- 그리드 라인을 사용해 그리드 항목을 배치할 수 있음



| 종류 | 설명 | 예시 |
|-------------------|---|----------------------|
| grid-column-start | 칼럼 시작의 라인 번호를 지정합니다. | grid-column-start: 1 |
| grid-column-end | 칼럼 마지막의 라인 번호를 지정합니다. | grid-column-end: 1 |
| grid-column | 칼럼 시작 번호와 칼럼 끝 번호 사이에 슬래시(/)를 넣어 사용합니다. | grid-column: 1/4 |
| grid-row-start | 줄 시작의 라인 번호를 지정합니다. | grid-row-start: 2 |
| grid-row-end | 줄 마지막의 라인 번호입니다. | grid-row-end: 4 |
| grid-row | 줄 시작 번호와 줄 끝 번호 사이에 슬래시(/)를 넣어 사용합니다. | grid-row: 2/4 |



```
<style>
  #wrapper {
    width: 700px;
    display: grid;
    grid-template-columns: repeat(3, 1fr);
    grid-template-rows: repeat(3, 100px);
  }
```

```
.box1 {
  background-color: #3689ff;
  grid-column: 1/4;
}

.box2 {
  background-color: #00cf12;
  grid-row: 2/4;
}

.box3 {
  background-color: #ff9019;
  grid-column: 2/4;
  grid-row-start: 2;
}

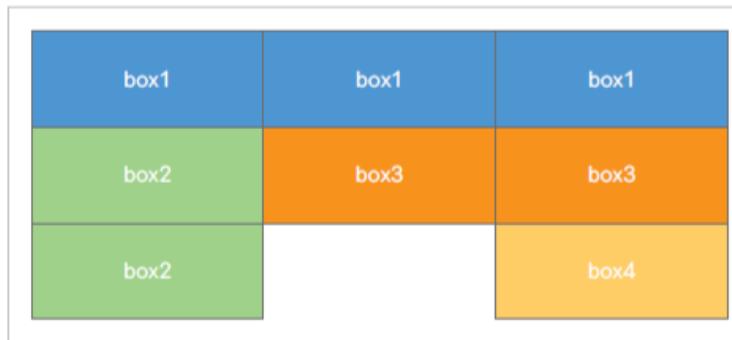
.box4 {
  background-color: #ffd000;
  grid-column-start: 3;
  grid-row-start: 3;
}
```

CSS 그리드 레이아웃

템플릿 영역을 만들어 배치하기

grid-area 속성을 사용해 템플릿 영역을 만든 후 배치

예) box1 ~ box4까지 네 개의 템플릿 영역을 만든 후 배치



1) 템플릿 영역 만들기

```
.box1 {  
    background-color: #3689ff;  
    grid-area: box1;  
}  
  
.box2 {  
    background-color: #00cf12;  
    grid-area: box2;  
}  
  
.box3 {  
    background-color: #ff9019;  
    grid-area: box3;  
}  
  
.box4 {  
    background-color: #ffd000;  
    grid-area: box4;  
}
```

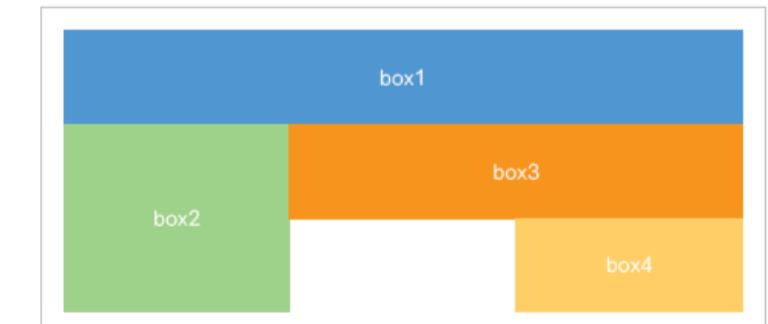
2) 템플릿 영역 배치하기

```
#wrapper {  
    width: 700px;  
    display: grid;  
    grid-template-columns: repeat(3, 1fr);  
    grid-template-rows: repeat(3, 100px);  
    grid-template-areas:  
        "box1 box1 box1"  
        "box2 box3 box3"  
        "box2 . box4";
```

한 줄에 들어갈 템플릿 영역을 큰따옴표(" ")로 묁습니다.

빈 영역은 마침표(.)를 넣습니다.

3) 결과 화면



CSS 그리드 레이아웃

grid-gap

- 기본형

grid-gap: 줄간격 열간격 ;

- 특징

일반적으로 요소 간 여백을 설정하기 위해 margin 사용

그리드 레이아웃에서는 grid-gap을 이용해 각 그리드 사이 여백을 설정

그리드 사이의 여백이기 때문에 컨테이너의 외곽은 여백 지정 안됨

대신 그리드 컨테이너에 padding 값 이용

grid-area

- 기본형

grid-gap: 시작열 / 끝열 / 시작행 / 끝행;

- 특징

grid-areas에서 지정한 영역의 이름으로 지정하면 해당 영역을 차지함



저자 고경희
발행일 2024-07-30
사양 688쪽 | 188*257mm
ISBN 979-11-6303-622-7 13000

<https://www.easyspub.co.kr/>