

3-1 개념

grid속성



grid

2차원(가로+세로) 레이아웃을 위한 속성, 그리드 컨테이너를 만든다.

Flexbox는 가로 또는 세로 한 방향만 주로 정렬하는 데 반해,

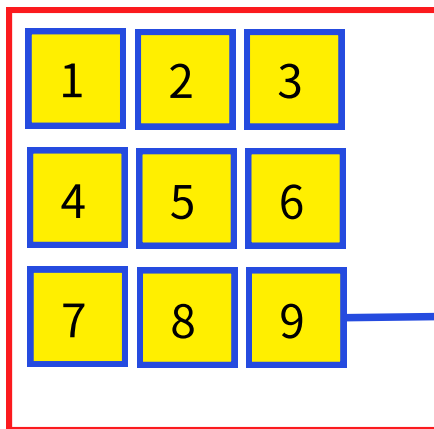
Grid는 가로+세로를 동시에 설계할 수 있어서 복잡한 레이아웃 만들 때 유리하다.

- Grid는 기본적으로 모든 아이템이 "정사각형 셀"에 들어가려고 한다.

- 라인(line) 기반 배치



구성

**grid-container**

- 블록레벨컨테이너
- `display:grid;` 사용

grid-item

- 별도로 자식에게 `display` 설정 필요 없음.

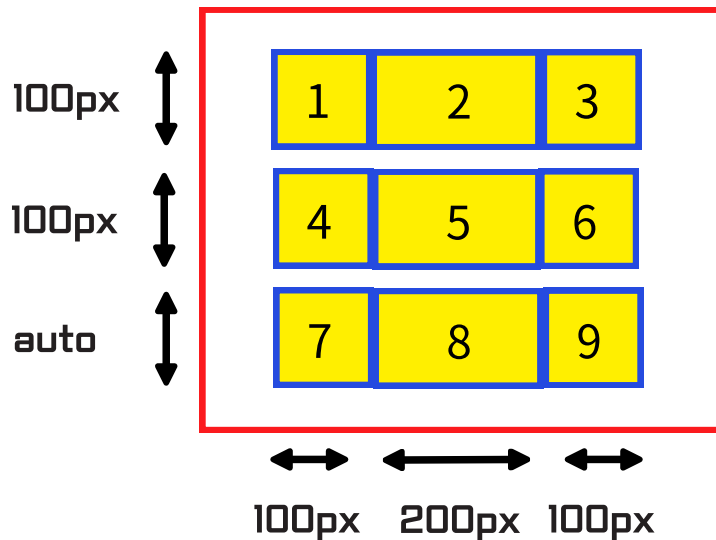
grid-container에 주는 속성

속성	내용
grid-template-rows grid-template-columns	행/열의 크기를 지정 (갯수가 아닌 크기)
gap (column-gap, row-gap)	아이템 간의 간격을 설정
grid-auto-rows grid-auto-columns	자동 생성된 행/열의 높이를 뭇로 할지 정하는 설정 기본값 auto
grid-auto-flow	아이템이 많아서 행/열 추가할 때 행이냐, 열이냐 정하는 설정. row(기본값, 행 추가), column(열 추가)
justify-content align-content	모든 item을 "묶어서" 어떻게 정렬할지 설정

💡 grid-template-columns, rows

[예제 CSS]

```
.container {
  display: grid;
  grid-template-rows : 100px 100px;
  // 세로 줄(행)을 2줄로 나누는데 각각 100px 높이로 설정
  grid-template-columns : 100px 200px 100px;
  // 가로 칸(열)을 3칸으로 나누는데 각각 100px, 200px, 100px 너비로 설정
}
```



새로운 행은 grid-template-rows에 정의되지 않은 행이라 height가 자동으로 auto로 잡힘
 grid-auto-rows: 100px;을 주면, 새로 생긴 3번째 줄도 깔끔하게 100px 높이로 맞춰줄 수 있다.
 ** fr은 남은 공간을 비율로 나누는 단위 -> 컨테이너 너비가 확정돼야 정확하다

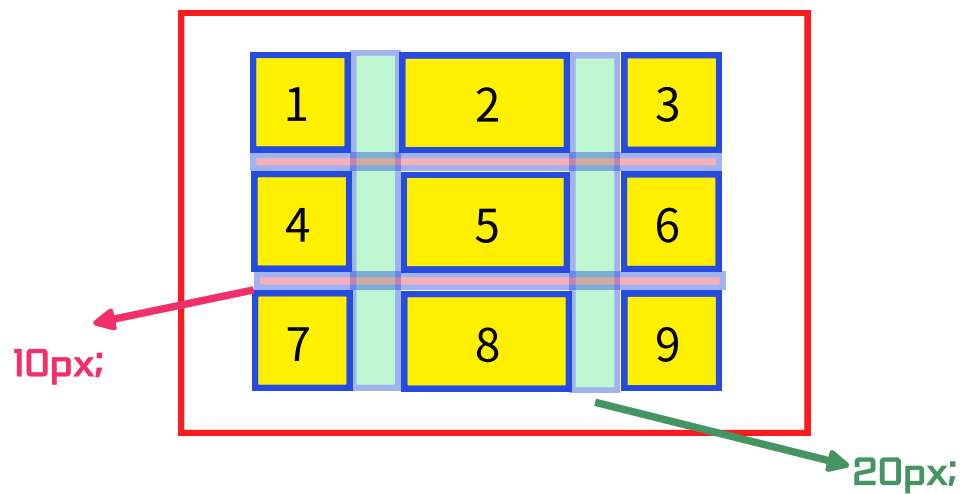
[예제 CSS]

```
.container {
  display: grid;
  grid-template-rows : repeat(4, 100px);
  // grid-template-rows : 100px 100px 100px 100px;
  grid-template-columns : repeat(4, 50px 100px);
  // grid-template-columns : 50px 100px를 4번 반;
}
```

[예제 CSS]

```
.container {
  display: grid;
  grid-template-rows : 1fr 2fr 1fr;
  // 총 4등분 후 1:2:1의 비율로 차지
}
```

```
.container {
  display: grid;
  grid-template-rows : 100px 100px;
  // 세로 줄(행)을 2줄로 나누는데 각각 100px 높이로 설정
  grid-template-columns : 100px 200px 100px;
  // 가로 칸(열)을 3칸으로 나누는데 각각 100px, 200px, 100px 너비로 설정
  gap : 10px 20px; //상하 좌우
}
```



grid-item 속성

grid-item에 주는 속성

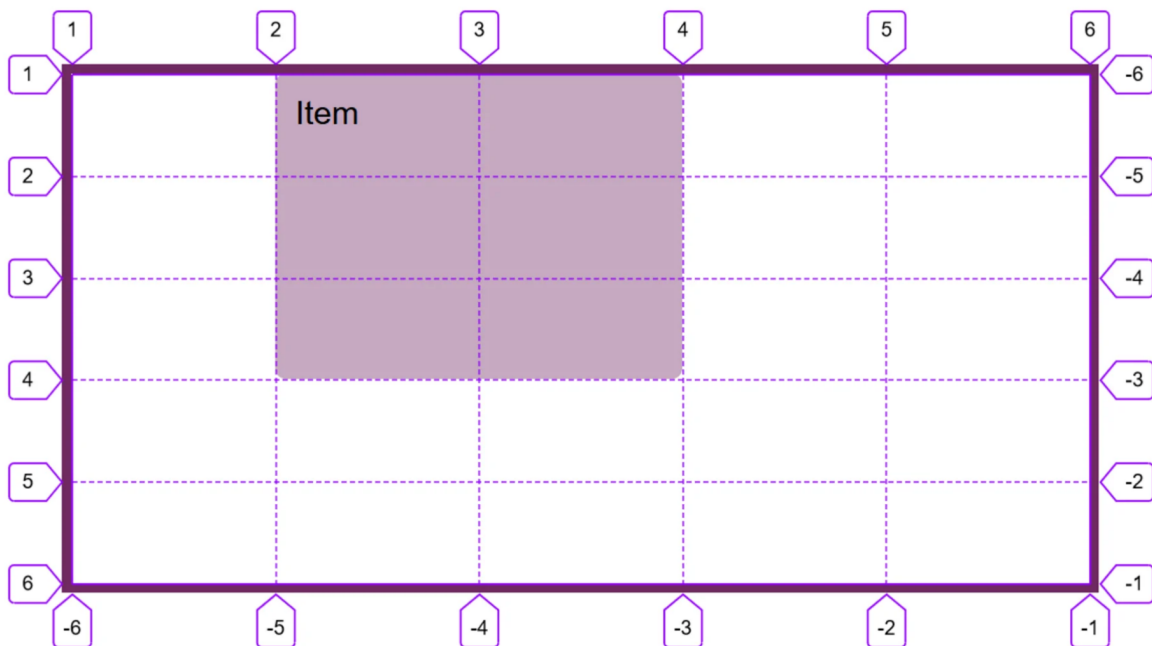
특정 아이템만 따로 위치나 크기를 조정하고 싶을 때

속성	내용
grid-column-start	시작할 열 번호 (선 번호). grid-column-start: 2;
grid-column-end	끝낼 열 번호 (선 번호)
grid-column	시작/끝을 한 줄로 축약. grid-column: 2 / 4;
grid-row-start	시작할 행 번호 (선 번호)
grid-row-end	끝낼 행 번호 (선 번호)
grid-row	시작/끝을 한 줄로 축약. grid-row: 1 / 3; ** grid-row: 2 / span 2; 로도 사용 가능 2번 시작으로 +2만큼의 공간 차지 (2행, 3행) ** grid-row: span 2 / 4; 2칸 차지하고 4번으로 끝난다. (2행, 3행)
grid-area	아이템의 위치를 한 번에 지정하는 축약형 속성 1. 아이템 이름 방식 각 아이템에 grid-area로 이름 설정 .container에서 grid-template-areas : "header header" "sidebar content"; //따옴표 : 첫번째 행, 따옴표 내부 띄워쓰기 : 열 2. 라인번호 방식 (축약형) grid-area : [grid-row-start]/[grid-column-start]/ [grid-row-end]/[grid-column-end]; // grid-area : 1 / 2 / 3 / 4;
justify-items align-items	각 item을 어떻게 정렬할지 설정 (셀 안에서)
place-self	justify-self + align-self 축약형 place-self: center start;

! 주요 개념

개념	내용
그리드 라인	칸을 나누는 경계선으로, 가로/세로 모두 존재 자동으로 번호가 매겨짐 (1부터 시작, 뒤에서 -1부터 시작)
그리드 셀	라인과 라인 사이의 하나의 칸
그리드 트랙	하나의 줄 (행 또는 열, row track, column track)
그리드 영역	여러 셀을 합쳐 만든 큰 영역 가로/세로에 대해 시작 라인, 끝 라인 지정 <pre> ** .item { grid-column : 1 / 3; //가로 2칸 병합 grid-row : 1 / 3; //세로 2칸 병합 } </pre>

! grid line



** flex 사용

상단에 메뉴를 가로로 나열 (ex. GNB), 카드 3개씩 한 줄로 쪽 나열, 버튼 안에 아이콘+텍스트를 수평 정렬

** grid 사용

2차원 배치하고 싶을 때, 반응형으로 칸을 정확히 쪼개고 싶은 경우,
메인페이지 섹션을 헤더/사이드바/콘텐츠로 나누기

큰 레이아웃 : grid -> 세부 정렬 : flex 조합 많이 사용됨