

12. CSS 그리드 레이아웃



12-1 CSS 그리드 레이아웃 사용하기

12-2 그리드 라인과 템플릿 영역으로 배치하기

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)
- 행(row)과 열(column)로 화면을 구성하고, 필요할 경우 그리드 항목 사이에 간격을 둘 수 있다.



CSS 그리드 레이아웃 항목을 배치하는 속성

display 속성

배치 요소들을 감싸는 부모 요소를 그리드 컨테이너로 지정

종류	설명
grid	블록 레벨의 그리드 컨테이너를 만듭니다.
inline-grid	인라인 레벨의 그리드 컨테이너를 만듭니다.

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

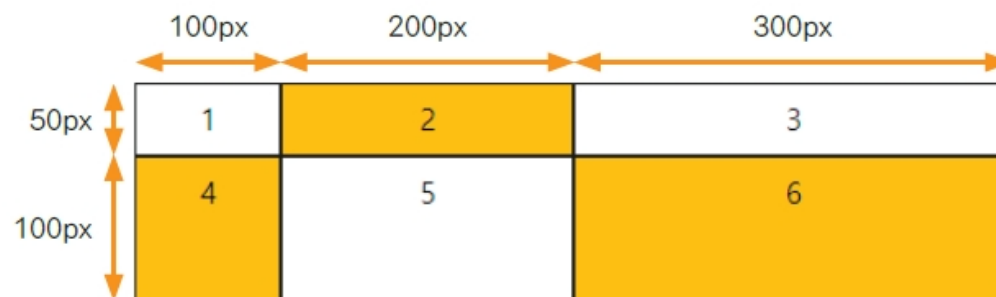
열과 행의 크기, 개수 지정

- grid-template-columns : 그리드 컨테이너 안의 열 개수와 너비값
- grid-template-rows: 그리드 컨테이너 안의 행 개수와 너비값

```
<style>
.container {
  display:grid;
  grid-template-columns: 100px 200px 300px;
  grid-template-rows: 50px 100px;
}
.items{ ... }
.items:nth-child(even){ ... }
</style>

.....

<div class="container">
  <div class="items">1 </div>
  .....
</div>
```



CSS 그리드 레이아웃 항목을 배치하는 속성

grid-auto-rows 속성

- 동적으로 추가되는 행의 높이 지정
- 대부분 동적인 웹 사이트는 행의 개수를 미리 정할 수 없는 경우가 많음
→ grid-template-rows보다 grid-auto-rows 속성을 많이 사용함

```
<style>
.container {
  display:grid;
  grid-template-columns: 100px 200px 300px;
  grid-auto-rows: 100px;
}
.items{ ... }
.items:nth-child(even){ ... }
</style>

.....

<div class="container">
  <div class="items"> 1 </div>

  .....
</div>
```



CSS 그리드 레이아웃 항목을 배치하는 속성

상대적인 크기를 지정하는 fr 단위

열과 행의 크기를 지정하는 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 최댓값을 지정하는 minmax() 함수

행의 높이를 px 값으로 고정하면 내용이 많을 경우 내용이 가려질 수 있음

→ minmax() 함수를 사용하면 행 높이를 최솟값과 최댓값으로 지정 가능

예) 행 높이는 최소 100px, 최대 auto

```
<style>
.container {
  display:grid;
  grid-template-columns: repeat(3, 1fr);
  grid-template-rows: minmax(100px, auto);
}
.....
</style>
```

Lorem ipsum dolor sit amet consectetur adipiscing elit. Amet, reprehenderit.Lorem ipsum dolor, sit amet consectetur adipiscing elit.	Lorem ipsum dolor, sit amet consectetur adipiscing elit.Lorem ipsum dolor, sit amet consectetur adipiscing elit	Lorem ipsum dolor sit amet.Lorem ipsum dolor, sit amet consectetur adipiscing elit
Lorem ipsum dolor sit.Lorem ipsum dolor, sit amet consectetur adipiscing elit	Lorem, ipsum dolor.	Lorem, ipsum dolor. Lorem ipsum dolor sit amet.

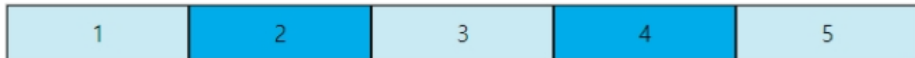
CSS 그리드 레이아웃 항목을 배치하는 속성

빈 공간을 채우는 auto-fill, auto-fit

열의 개수를 지정하는 대신 auto-fit이나 auto-fill을 지정하면 화면 너비에 따라 행을 채움.

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

auto-fit일 때



auto-fill일 때



```
<style>
.container {
  width:800px;
  border: 1px solid #ccc;
  display:grid;
  margin-bottom: 20px;
}
.container-1 {
  grid-template-columns:repeat(auto-fit, minmax(100px, 1fr));
}
.container-2 {
  grid-template-columns:repeat(auto-fill, minmax(100px, 1fr));
}
.....
</style>
```

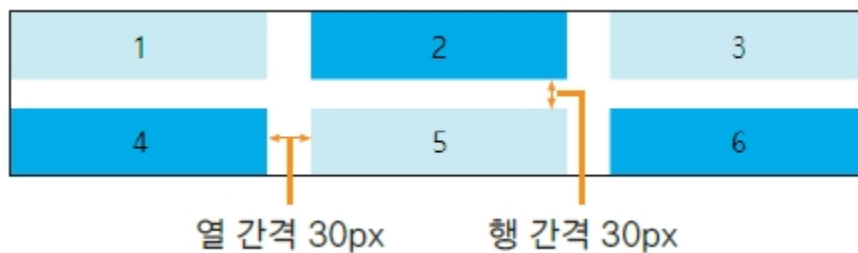
CSS 그리드 레이아웃 항목을 배치하는 속성

그리드 항목의 간격을 조절하는 gap 속성

값이 1개 : 열 간격과 행 간격을 똑같이 사용

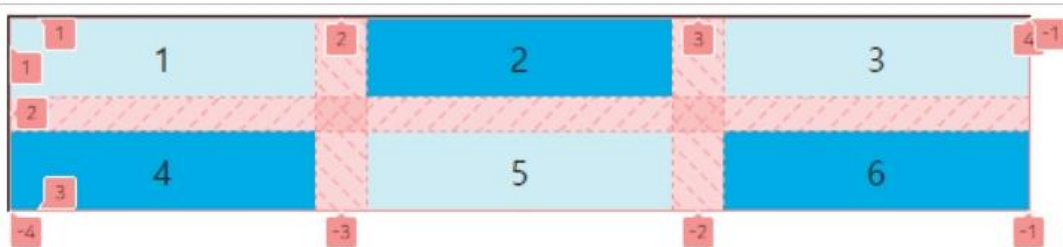
값이 2개 : 첫번째 값은 행 간격, 두번째 값은 열 간격으로 사용

```
<style>
.container {
  width: 600px;
  border: 2px solid #222;
  display: grid;
  grid-template-columns: repeat(3, 1fr); /* 열 3개 */
  gap: 20px 30px; /* 행 간격 20px, 열 간격 30px */
}
.....
</style>
```



그리드 라인을 사용해 배치하기

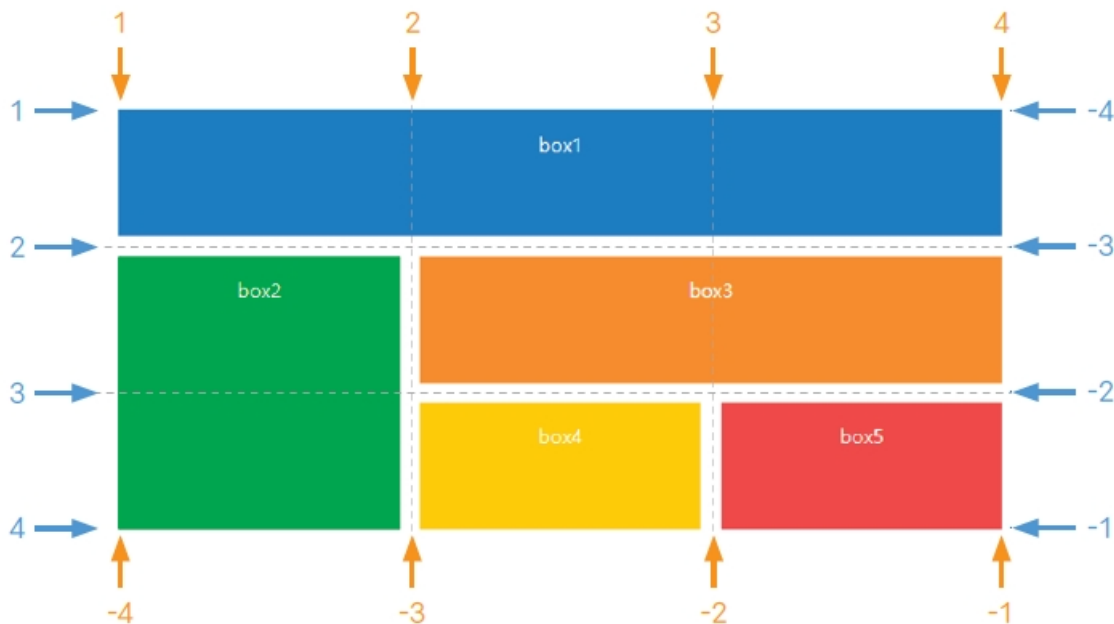
- CSS 그리드 레이아웃에는 눈에 보이지 않는 그리드 라인이 포함되어 있음
- 그리드 라인을 사용해 그리드 항목을 배치할 수 있음



```
<!DOCTYPE html>
<html lang="ko">
  <head> ... </head>
  <body>
    ... <div class="container">
      <div class="items">1</div>
      <div class="items">2</div>
      <div class="items">3</div>
      <div class="items">4</div>
      <div class="items">5</div>
      <div class="items">6</div>
    </div>
```

속성	설명	예시
grid-column-start	열의 시작 번호를 지정합니다.	grid-column-start: 1;
grid-column-end	열의 끝 번호를 지정합니다.	grid-column-end: 4
grid-column	열의 시작과 끝 번호를 함께 지정합니다. 시작과 끝 번호는 슬래시(/)로 구분합니다.	grid-column: 1 / 4;
grid-row-start	행의 시작 번호를 지정합니다.	grid-row-start: 2;
grid-row-end	행의 끝 번호를 지정합니다.	grid-row-end: 4;
grid-row	행의 시작과 끝 번호를 함께 지정합니다. 시작과 끝 번호는 슬래시(/)로 구분합니다.	grid-row: 2 / 4;

그리드 라인을 사용해 배치하기



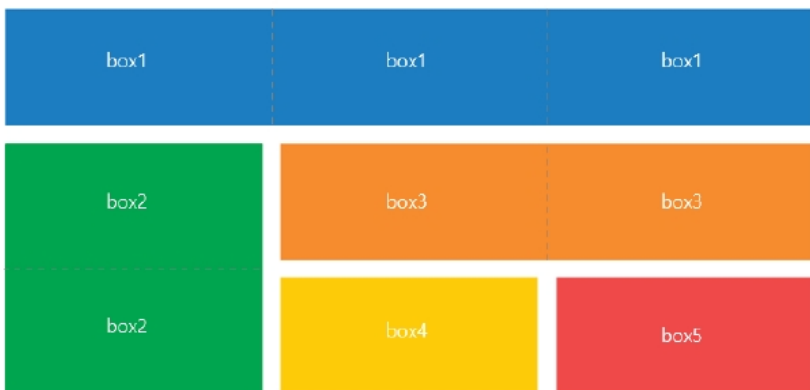
box1 영역	열은 처음부터 끝까지, 행은 1에서 2까지
box2 영역	열은 1에서 2까지, 행은 2에서 끝까지
box3 영역	열은 2에서 끝까지, 행은 2에서 3까지
box4 영역	열은 2에서 3까지, 행은 3에서 4까지
box5 영역	열은 3에서 끝까지, 행은 3에서 끝까지

```
<style>
.....
.box1 {
  background-color:#3689ff;
  grid-column: 1 / -1; /* grid-column:1 / 4; */
  grid-row-start: 1; /* grid-row: 1 / 2; */
}
.box2 {
  background-color:#00cf12;
  grid-column-start: 1; /* grid-column: 1 / 2; */
  grid-row: 2 / -1; /* grid-row: 2 / 4; */
}
.box3 {
  background-color:#ff9019;
  grid-column:2 / -1; /* grid-column: 2 / 4; */
  grid-row-start: 2; /* grid-row: 2 / 3; */
}
.box4 {
  background-color:#ffd000;
  grid-column-start:2; /* grid-column: 2 / 3; */
  grid-row-start:3; /* grid-row: 3 / 4; */
}
.box5 {
  background-color:#ff3f3f;
  grid-column: 3 / -1; /* grid-column: 3 / 4; */
  grid-row: 3 / -1; /* grid-row: 3 / 4; */
}
</style>
```

템플릿 영역을 사용해 배치하기

1) grid-area 속성을 사용해 템플릿 영역을 만들기

예) box1 ~ box4까지 네 개의 템플릿 영역을 만든 후 배치



```
<style>
.container{
  width:700px;
  display:grid;
  grid-template-rows:repeat(3, 100px);
  gap: 1rem;
}

.....

.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;
}
.box5 {
  background-color:#ff3f3f;
  grid-area:box5;
}
</style>
```

템플릿 영역을 사용해 배치하기

2) 템플릿 영역 배치하기

```
<style>
.container {
  width:700px;
  display:grid;
  grid-template-rows:repeat(3, 100px);
  grid-template-areas:
    "box1 box1 box1"
    "box2 box3 box3"
    "box2 box4 box5";
  gap: 1rem;
}
.....
</style>
```

