

HTML & CSS

◆ 그리드 레이아웃

정수아

Contents

01 그리드 레이아웃

02 그리드 레이아웃 속성



01

그리드 레이아웃

그리드 레이아웃

❖ 그리드(grid)란?

- 격자, 바둑판을 의미

❖ 그리드 레이아웃

- 웹 사이트를 여러 개의 칼럼(column)으로 나눈 후 메뉴나 본문, 이미지 등의 웹 요소를 화면에 맞게 배치하는 것을 의미
- 특징
 - 시각적으로 안정된 레이아웃 유지
 - 업데이트가 편리
 - 자유로운 요소 배치

그리드 레이아웃

❖ 그리드 레이아웃





02

그리드 레이아웃 속성

그리드 레이아웃 속성

❖ display 속성

- 그리드를 적용한 요소의 바깥 부분을 그리드 컨테이너로 생성

값	설명
grid	컨테이너 안의 항목을 블록 레벨 요소로 배치
inline-grid	컨테이너 안의 항목을 인라인 레벨 요소로 배치

그리드 레이아웃 속성

❖ 칼럼, 줄 지정 속성

값	설명
grid-template-columns	컨테이너 안에 배치할 칼럼의 크기와 개수 지정
grid-template-rows	컨테이너 안에 배치할 줄의 크기와 개수 지정

❖ 예시

```
grid-template-columns: 200px 200px 200px;  
grid-template-rows: 100px;
```

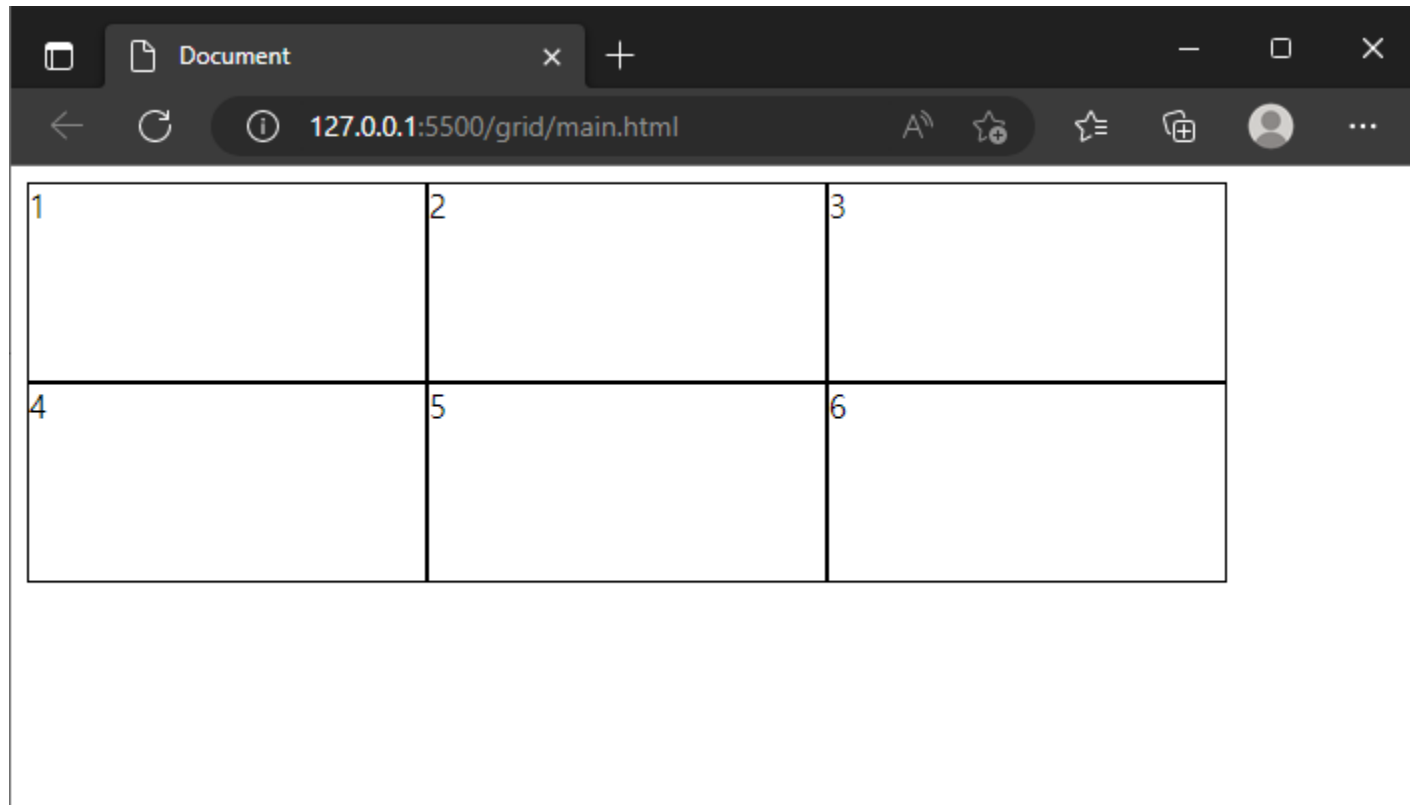

[실습] 그리드 레이아웃 만들기

```
#parent {  
    display: grid;  
    grid-template-columns: 200px 200px 200px;  
    grid-template-rows: 100px 100px;  
}  
  
.child {  
    border: 1px solid black;  
}
```

```
<div id="parent">  
    <div class="child">1</div>  
    <div class="child">2</div>  
    <div class="child">3</div>  
    <div class="child">4</div>  
    <div class="child">5</div>  
    <div class="child">6</div>  
</div>
```

[실습] 그리드 레이아웃 만들기

❖ 실행 결과



그리드 레이아웃 속성

❖ repeat() 함수

- 값이 반복될 때 줄여서 표현할 수 있는 함수
- repeat(반복횟수, 너비)
- 예시

```
grid-template-columns: 200px 200px 200px;
```



```
grid-template-columns: repeat(3, 200px);
```

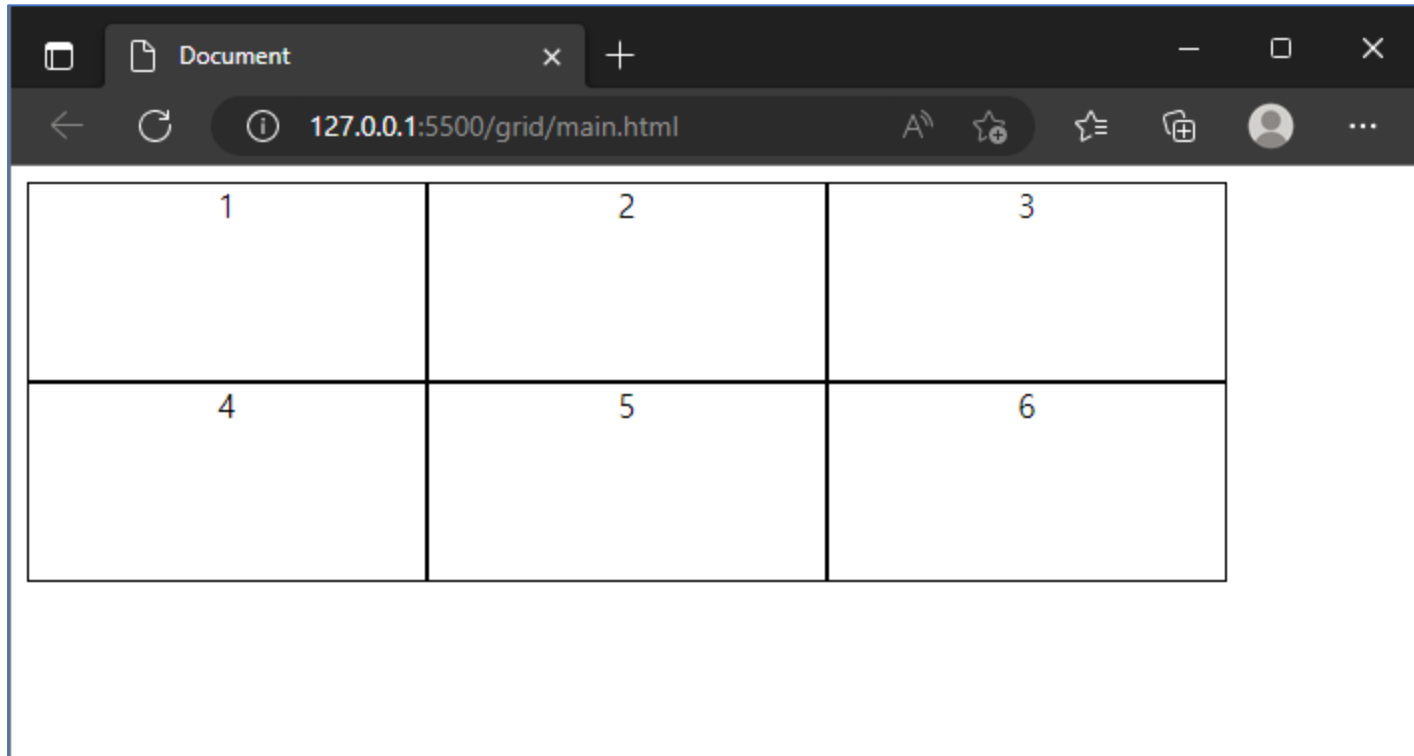
[실습] repeat() 함수

```
#parent {  
    display: grid;  
    grid-template-columns: repeat(3, 200px);  
    grid-template-rows: 100px 100px;  
}  
  
.child {  
    border: 1px solid black;  
    text-align: center;  
}
```

```
<div id="parent">  
    <div class="child">1</div>  
    <div class="child">2</div>  
    <div class="child">3</div>  
    <div class="child">4</div>  
    <div class="child">5</div>  
    <div class="child">6</div>  
</div>
```

[실습] repeat() 함수

❖ 실행 결과



그리드 레이아웃 속성

❖ fr(fraction) 단위

- 그리드 레이아웃에서 상대적인 크기를 지정하는 단위

- 예시

- 너비가 같은 칼럼 2개 배치

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

- 2:1:2로 줄 배치

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

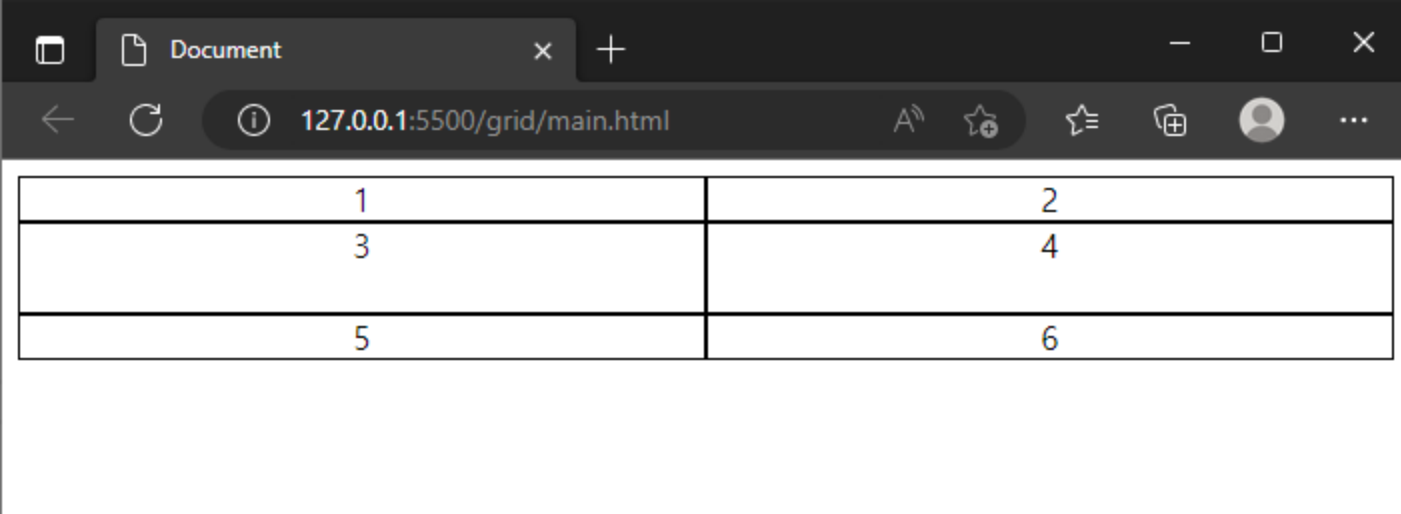
[실습] fr 단위

```
#parent {  
    display: grid;  
    grid-template-columns: 1fr 1fr;  
    grid-template-rows: 1fr 2fr 1fr;  
}  
  
.child {  
    border: 1px solid black;  
    text-align: center;  
}
```

```
<div id="parent">  
    <div class="child">1</div>  
    <div class="child">2</div>  
    <div class="child">3</div>  
    <div class="child">4</div>  
    <div class="child">5</div>  
    <div class="child">6</div>  
</div>
```

[실습] fr 단위

❖ 실행 결과



A screenshot of a web browser window. The address bar shows the URL `127.0.0.1:5500/grid/main.html`. The page content displays a table with 2 columns and 3 rows. The cells contain the numbers 1 through 6 in a sequential order from top-left to bottom-right.

1	2
3	4
5	6

그리드 레이아웃 속성

❖ minmax() 함수

- 최소값 최대값을 지정하여 줄 높이를 유연하게 지정하는 함수
- minmax(최소값, 최대값)

- 예시

- 줄의 최소 높이 100px, 최대 높이 auto

```
grid-template-rows: minmax(100px, auto);
```

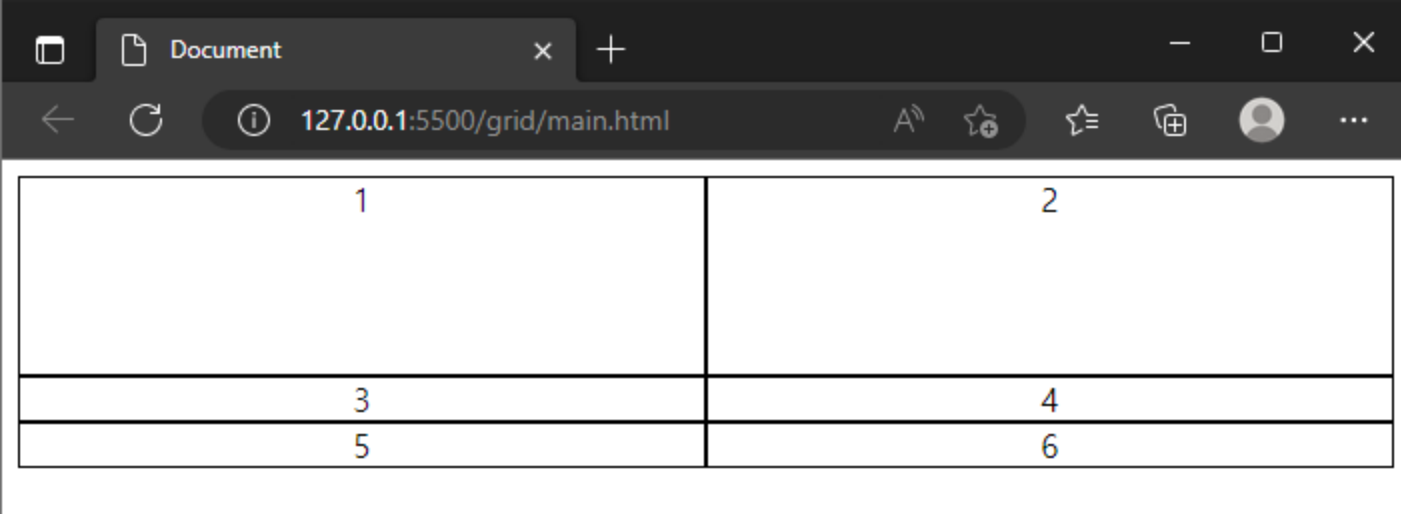
[실습] minmax() 함수

```
#parent {  
    display: grid;  
    grid-template-columns: 1fr 1fr;  
    grid-template-rows: minmax(100px, auto);  
}  
  
.child {  
    border: 1px solid black;  
    text-align: center;  
}
```

```
<div id="parent">  
    <div class="child">1</div>  
    <div class="child">2</div>  
    <div class="child">3</div>  
    <div class="child">4</div>  
    <div class="child">5</div>  
    <div class="child">6</div>  
</div>
```

[실습] minmax() 함수

❖ 실행 결과



A screenshot of a web browser window. The address bar shows the URL `127.0.0.1:5500/grid/main.html`. The browser displays a 3x2 grid of numbers. The numbers are arranged as follows:

1	2
3	4
5	6

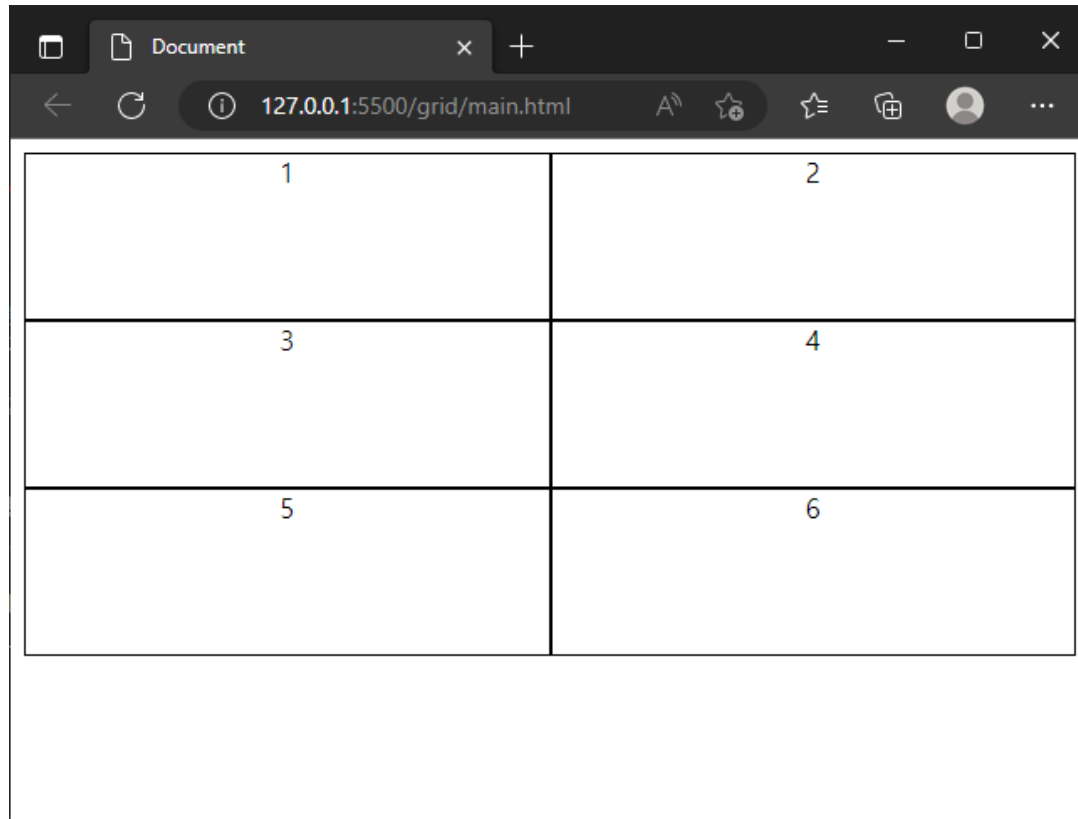
[실습] grid-auto-rows 속성

```
#parent {  
    display: grid;  
    grid-template-columns: 1fr 1fr;  
    grid-auto-rows: 100px;  
}  
  
.child {  
    border: 1px solid black;  
    text-align: center;  
}
```

```
<div id="parent">  
    <div class="child">1</div>  
    <div class="child">2</div>  
    <div class="child">3</div>  
    <div class="child">4</div>  
    <div class="child">5</div>  
    <div class="child">6</div>  
</div>
```

[실습] grid-auto-rows 속성

❖ 실행 결과



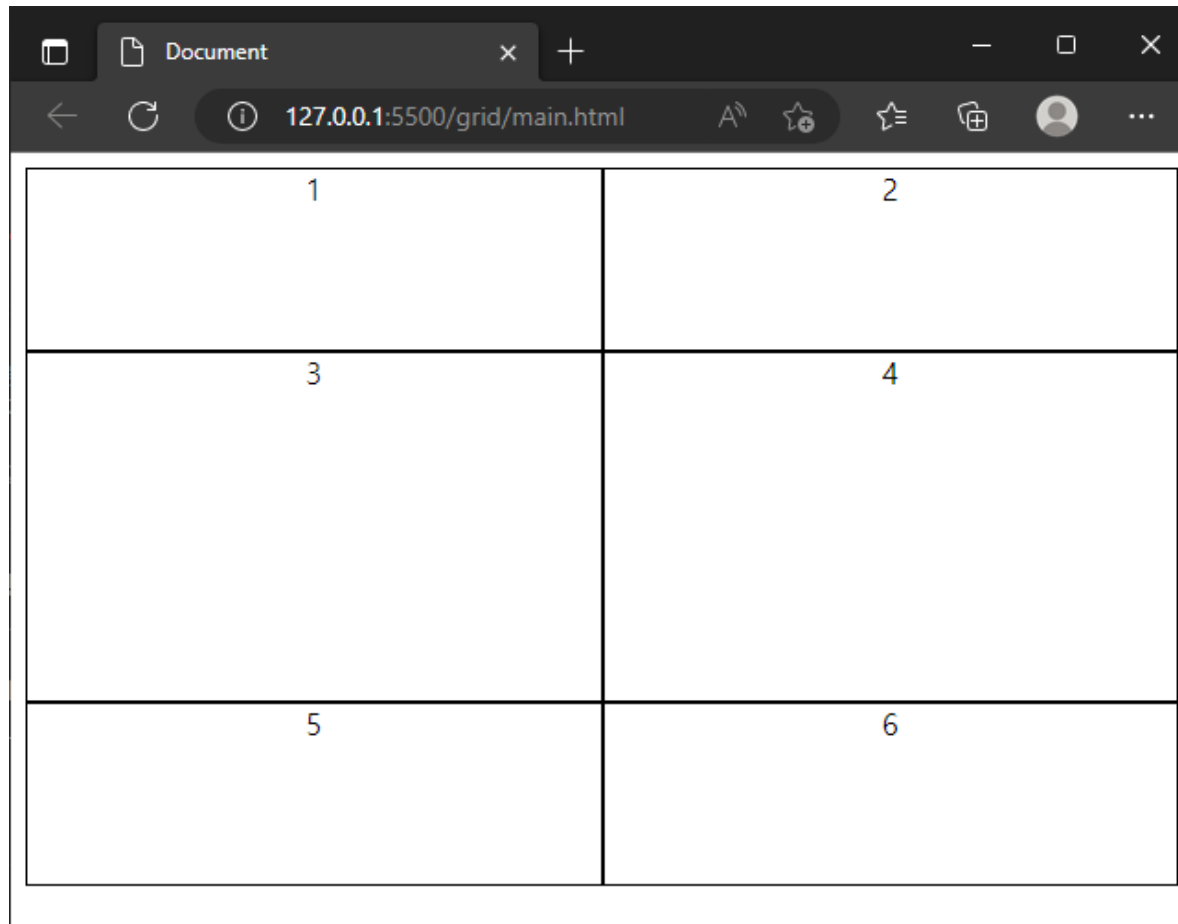
[실습] grid-auto-rows + minmax()

```
#parent {  
    display: grid;  
    grid-template-columns: 1fr 1fr;  
    grid-auto-rows: minmax(100px, auto);  
}  
  
.child {  
    border: 1px solid black;  
    text-align: center;  
}
```

```
<div id="parent">  
    <div class="child">1</div>  
    <div class="child">2</div>  
    <div class="child">3<br/><br/><br/><br/><br/><br/><br/></div>  
    <div class="child">4</div>  
    <div class="child">5</div>  
    <div class="child">6</div>  
</div>
```

[실습] grid-auto-rows + minmax()

❖ 실행 결과



그리드 레이아웃 속성

❖ auto-fill, auto-fit 값

- 칼럼 개수를 자동으로 조절 해주는 값
- auto-fit
 - 화면의 남은 공간 없이 꽉 채워서 칼럼을 표시
- auto-fill
 - 칼럼의 최소 너비만 표시하고 남은 공간은 그대로 둠

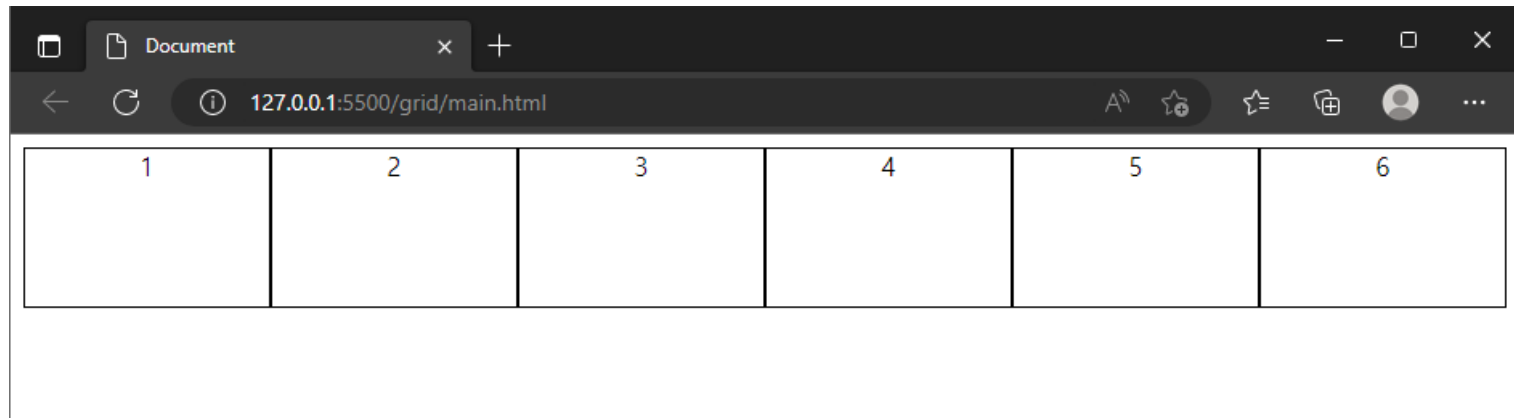
[실습] auto-fit

```
#parent {  
    display: grid;  
    grid-template-columns: repeat(auto-fit, minmax(100px, 1fr));  
    grid-template-rows: minmax(100px, auto);  
}  
  
.child {  
    border: 1px solid black;  
    text-align: center;  
}
```

```
<div id="parent">  
    <div class="child">1</div>  
    <div class="child">2</div>  
    <div class="child">3</div>  
    <div class="child">4</div>  
    <div class="child">5</div>  
    <div class="child">6</div>  
</div>
```

[실습] auto-fit

❖ 실행 결과



A screenshot of a web browser window. The address bar shows the URL `127.0.0.1:5500/grid/main.html`. The browser displays a table with 6 columns and 1 row. The columns are numbered 1 through 6. The table is styled with a dark header bar and a light body.

1	2	3	4	5	6
---	---	---	---	---	---

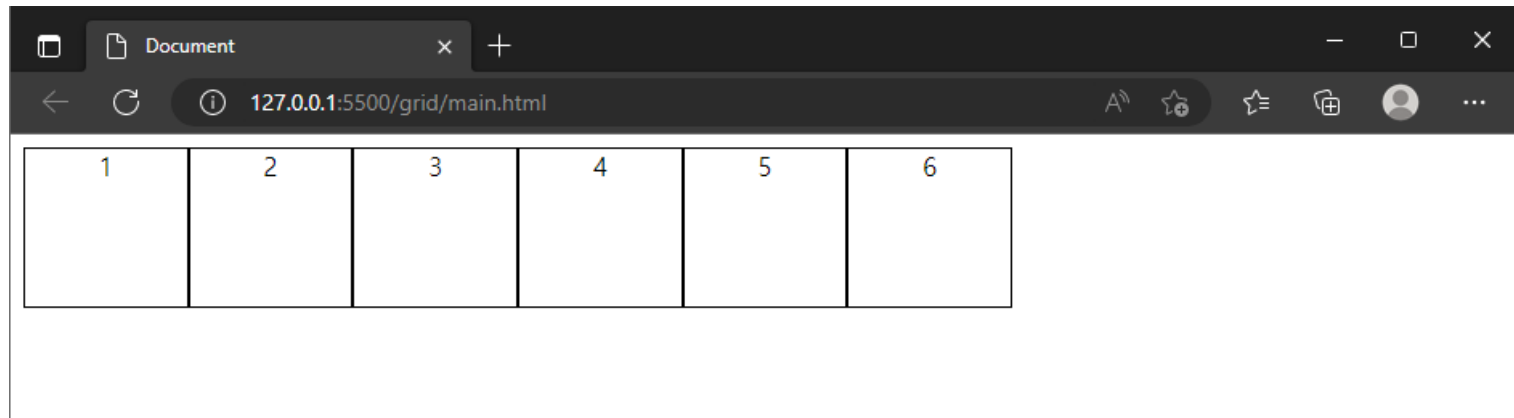
[실습] auto-fill

```
#parent {  
    display: grid;  
    grid-template-columns: repeat(auto-fill, minmax(100px, 1fr));  
    grid-template-rows: minmax(100px, auto);  
}  
  
.child {  
    border: 1px solid black;  
    text-align: center;  
}
```

```
<div id="parent">  
    <div class="child">1</div>  
    <div class="child">2</div>  
    <div class="child">3</div>  
    <div class="child">4</div>  
    <div class="child">5</div>  
    <div class="child">6</div>  
</div>
```

[실습] auto-fill

❖ 실행 결과



그리드 레이아웃 속성

❖ 그리드 항목의 간격을 지정하는 속성

값	설명
grid-column-gap	칼럼과 컬럼 사이의 간격을 지정
grid-row-gap	줄과 줄 사이의 간격을 지정
grid-gap	칼럼과 줄 사이의 간격을 한번에 지정

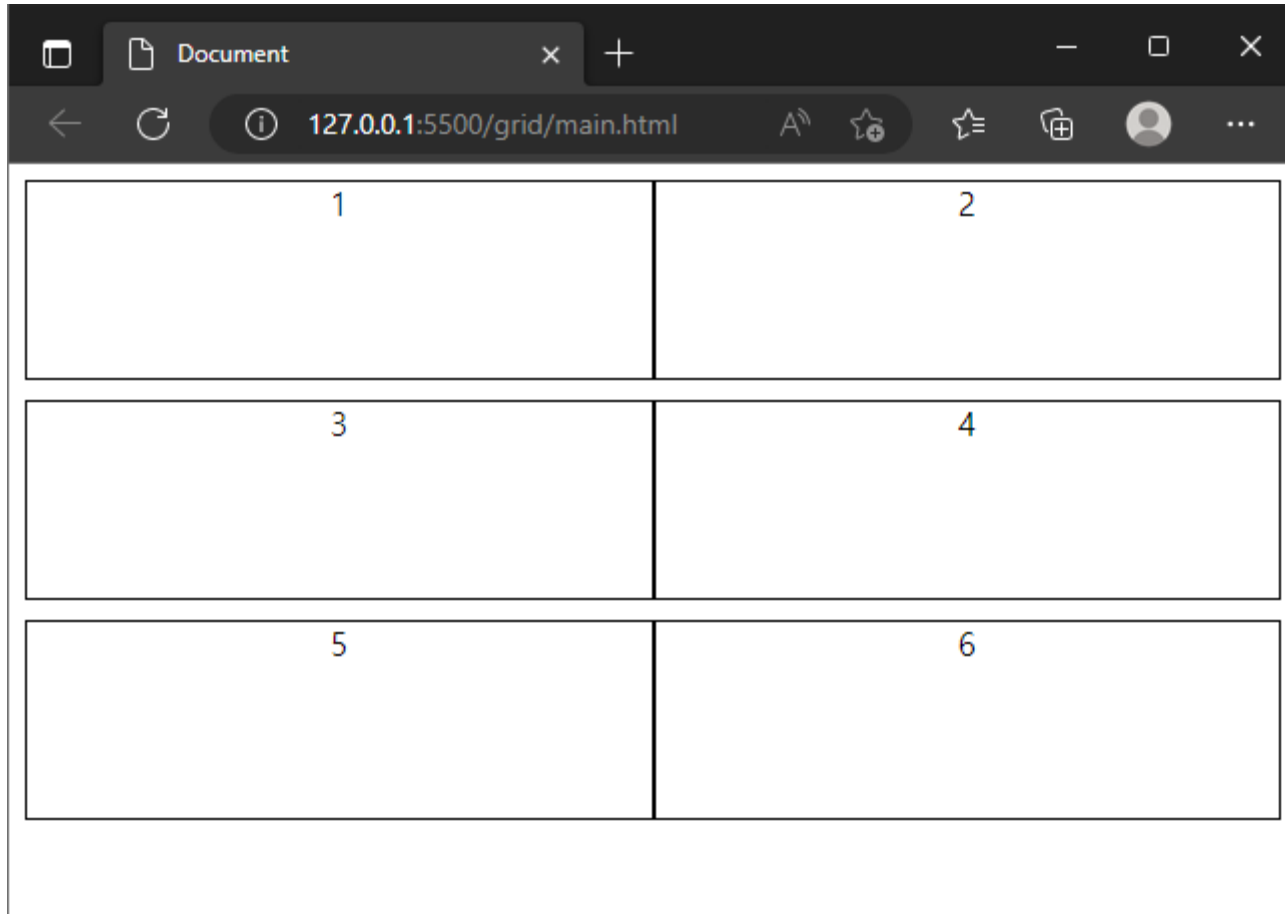
[실습] grid-row-gap

```
#parent {  
    display: grid;  
    grid-template-columns: 1fr 1fr;  
    grid-auto-rows: minmax(100px, auto);  
    grid-row-gap: 10px;  
}  
  
.child {  
    border: 1px solid black;  
    text-align: center;  
}
```

```
<div id="parent">  
    <div class="child">1</div>  
    <div class="child">2</div>  
    <div class="child">3</div>  
    <div class="child">4</div>  
    <div class="child">5</div>  
    <div class="child">6</div>  
</div>
```

[실습] grid-row-gap

❖ 실행 결과



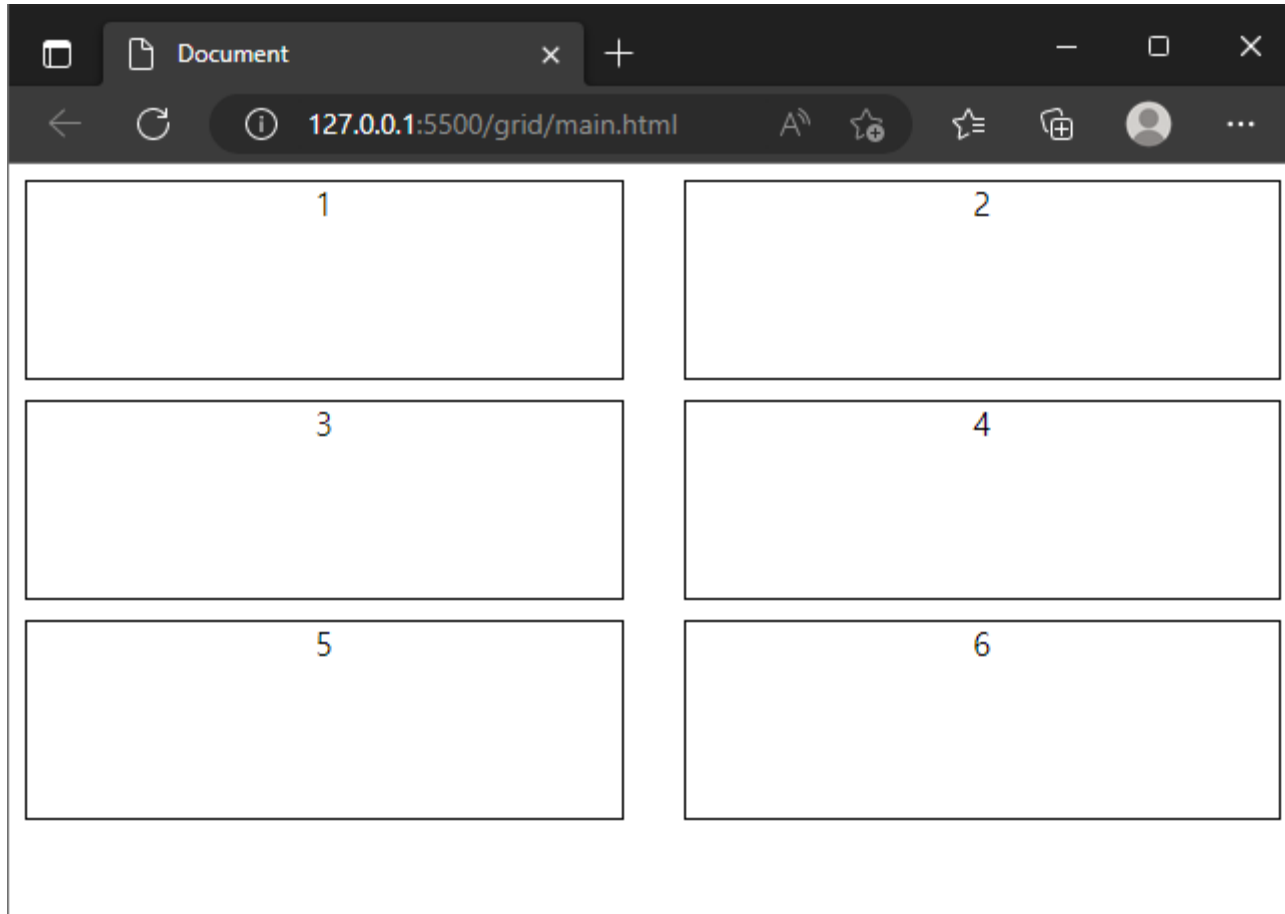
[실습] grid-column-gap

```
#parent {  
    display: grid;  
    grid-template-columns: 1fr 1fr;  
    grid-auto-rows: minmax(100px, auto);  
    grid-row-gap: 10px;  
    grid-column-gap: 30px;  
}  
  
.child {  
    border: 1px solid black;  
    text-align: center;  
}
```

```
<div id="parent">  
    <div class="child">1</div>  
    <div class="child">2</div>  
    <div class="child">3</div>  
    <div class="child">4</div>  
    <div class="child">5</div>  
    <div class="child">6</div>  
</div>
```


[실습] grid-column-gap

❖ 실행 결과



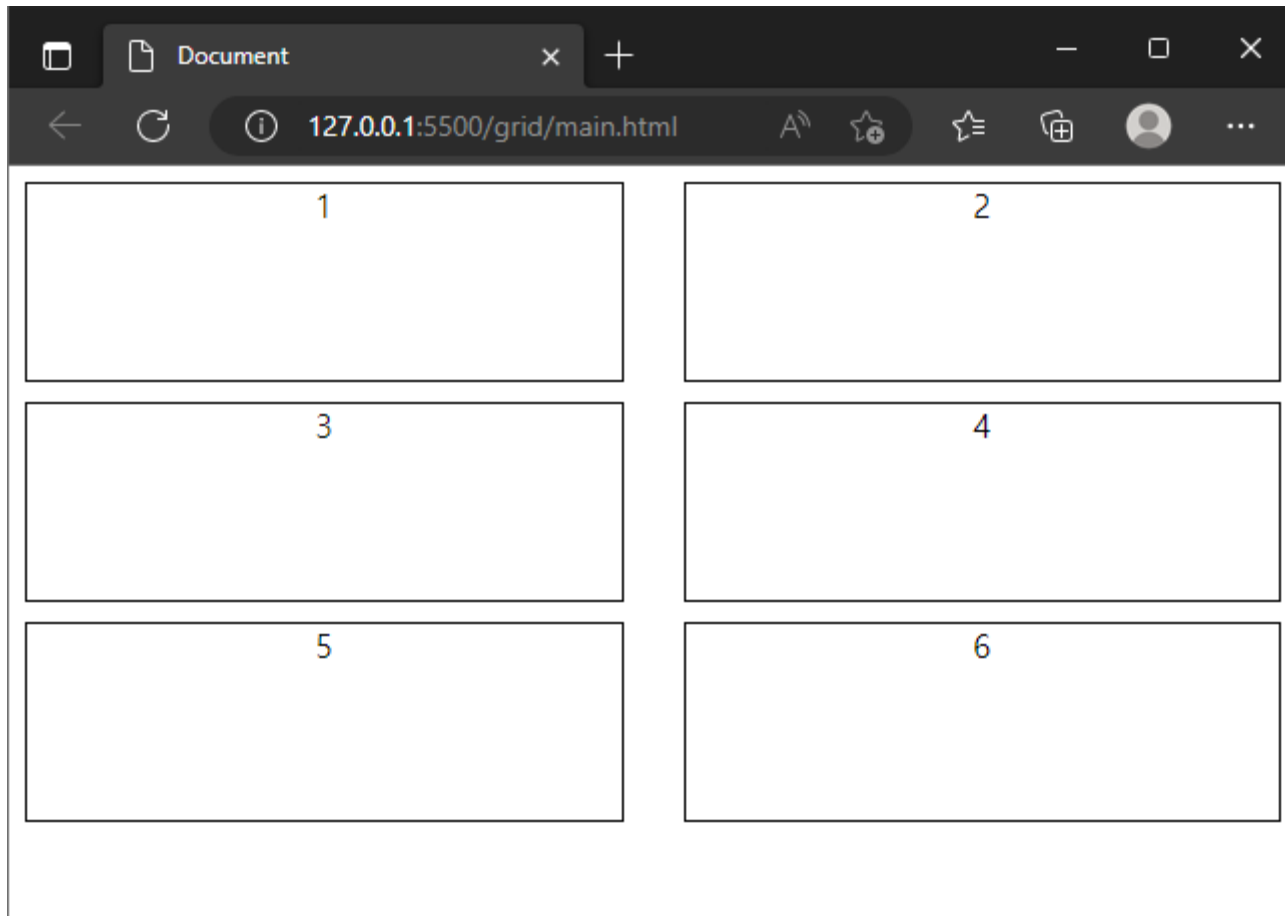
[실습] grid-gap

```
#parent {  
    display: grid;  
    grid-template-columns: 1fr 1fr;  
    grid-auto-rows: minmax(100px, auto);  
    grid-gap: 10px 30px;  
}  
  
.child {  
    border: 1px solid black;  
    text-align: center;  
}
```

```
<div id="parent">  
    <div class="child">1</div>  
    <div class="child">2</div>  
    <div class="child">3</div>  
    <div class="child">4</div>  
    <div class="child">5</div>  
    <div class="child">6</div>  
</div>
```

[실습] grid-gap

❖ 실행 결과



[실습] 상하좌우 동일한 간격 생성

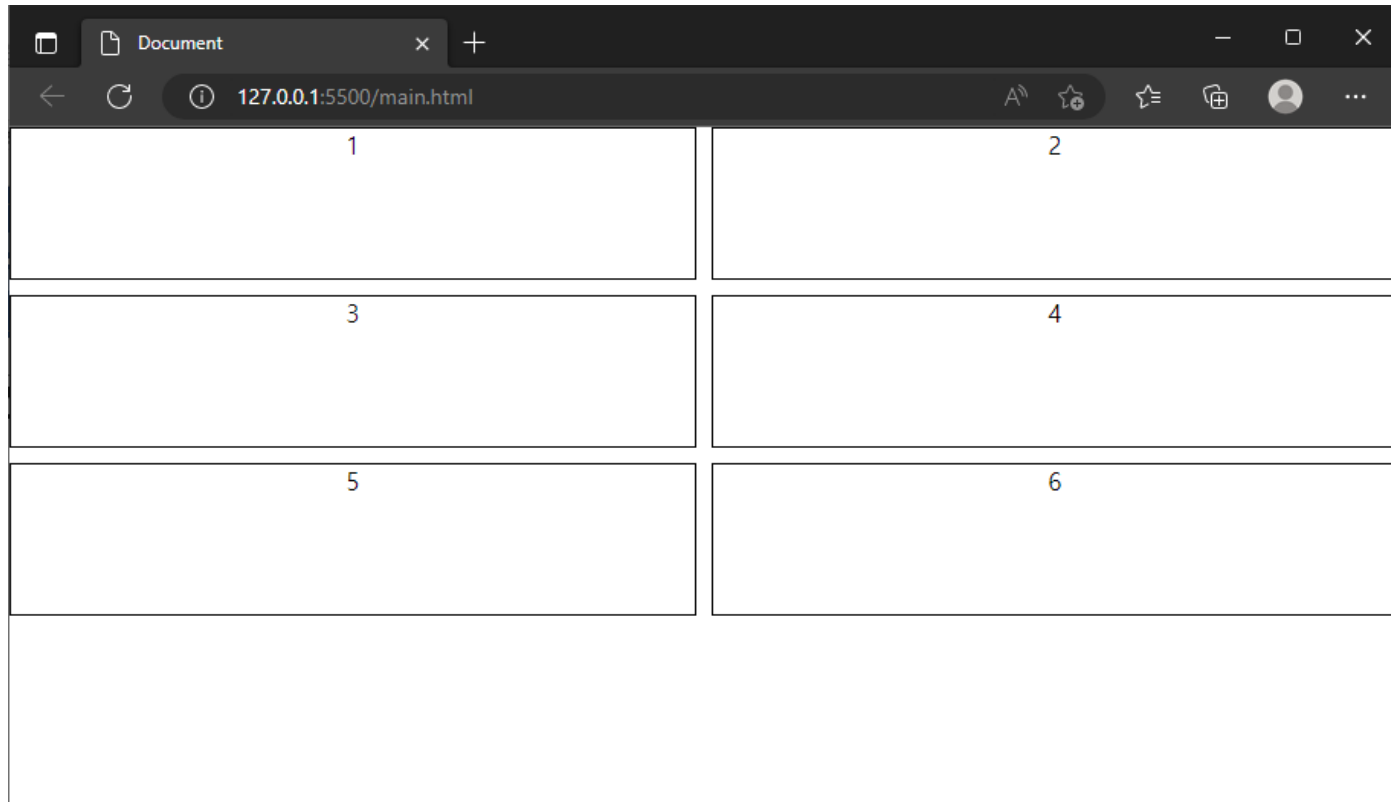
```
* {  
    margin: 0;  
    padding: 0;  
}  
  
#parent {  
    display: grid;  
    grid-template-columns: 1fr 1fr;  
    grid-auto-rows: minmax(100px, auto);  
    grid-gap: 10px;  
}
```

```
.child {  
    border: 1px solid black;  
    text-align: center;  
}
```

```
<div id="parent">  
    <div class="child">1</div>  
    <div class="child">2</div>  
    <div class="child">3</div>  
    <div class="child">4</div>  
    <div class="child">5</div>  
    <div class="child">6</div>  
</div>
```

[실습] 상하좌우 동일한 간격 생성

❖ 실행 결과



[실습] 상하좌우 동일한 간격 생성

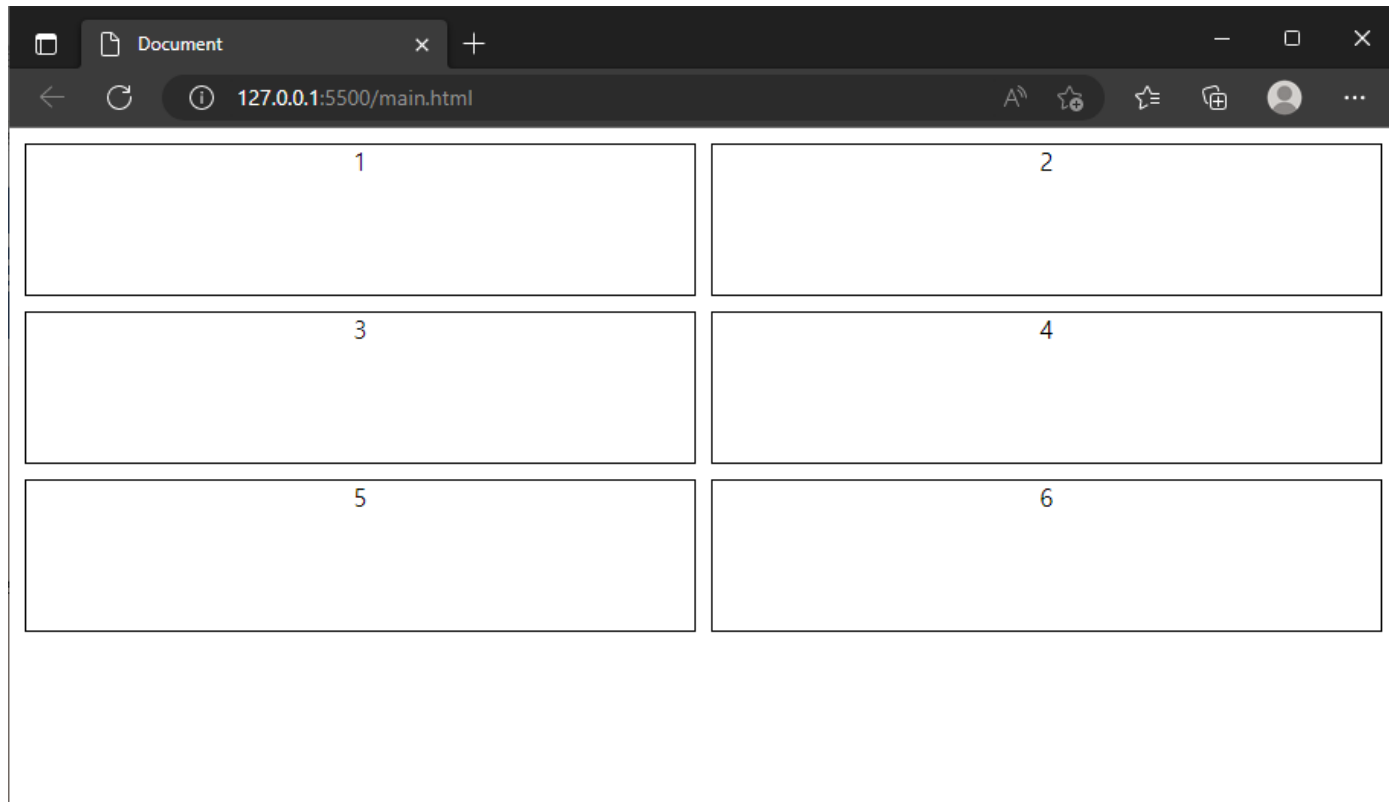
```
* {  
    margin: 0;  
    padding: 0;  
}  
  
#parent {  
    display: grid;  
    grid-template-columns: 1fr 1fr;  
    grid-auto-rows: minmax(100px, auto);  
    grid-gap: 10px;  
    padding : 10px;  
}
```

```
.child {  
    border: 1px solid black;  
    text-align: center;  
}
```

```
<div id="parent">  
    <div class="child">1</div>  
    <div class="child">2</div>  
    <div class="child">3</div>  
    <div class="child">4</div>  
    <div class="child">5</div>  
    <div class="child">6</div>  
</div>
```

[실습] 상하좌우 동일한 간격 생성

❖ 실행 결과



그리드 레이아웃 속성

❖ 그리드 라인을 이용한 배치

		1	2	3	4
		칼럼	칼럼	칼럼	
1	줄	1	2	3	
2	줄	4	5	6	
3	줄	7	8	9	
4					

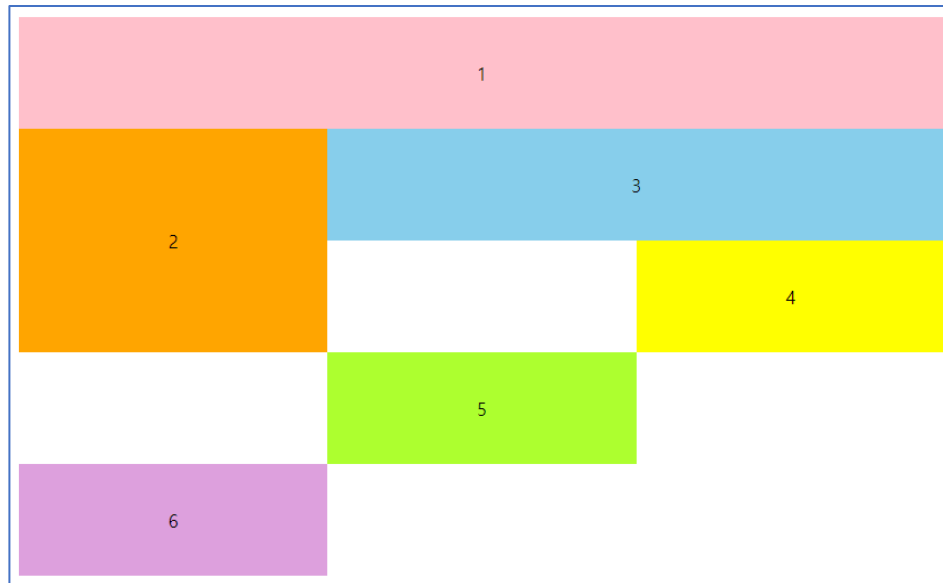
그리드 레이아웃 속성

❖ 그리드 라인을 이용한 배치 속성

종류	설명
grid-column-start	칼럼 시작 라인 번호 지정
grid-column-end	칼럼 마지막 라인 번호 지정
grid-column	칼럼 시작 번호와 끝 번호 사이에 /를 넣어 지정
grid-row-start	줄 시작의 라인 번호 지정
grid-row-end	줄 마지막 라인 번호를 지정
grid-row	줄 시작 번호와 끝 번호 사이에 /를 넣어 지정

[실습] 그리드 라인을 이용한 배치

```
<div id="parent">  
  <div class="child box1">1</div>  
  <div class="child box2">2</div>  
  <div class="child box3">3</div>  
  <div class="child box4">4</div>  
  <div class="child box5">5</div>  
  <div class="child box6">6</div>  
</div>
```



[실습] 그리드 라인을 이용한 배치

```
* {
  margin: 0;
  padding: 0;
}

#parent {
  padding: 10px;
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  grid-auto-rows: minmax(100px, auto);
}

.child {
  display: flex;
  justify-content: center;
  align-items: center;
}

.box1 {
  background: pink;
  grid-column-start: 1;
  grid-column-end: 4;
}
```

```
.box2 {
  background: orange;
  grid-row-start: 2;
  grid-row-end: 4;
}

.box3 {
  background: skyblue;
  grid-column: 2/4;
}

.box4 {
  background: yellow;
  grid-column-start: 3;
}

.box5 {
  background: greenyellow;
  grid-column-start: 2;
}

.box6 {
  background: plum;
  grid-row-start: 5;
}
```

그리드 레이아웃 속성

❖ grid-area 속성

- 1. 각 영역에 템플릿 이름을 지정

```
.box1 {  
    background: pink;  
    grid-area: box1;  
}  
  
.box2 {  
    background: orange;  
    grid-area: box2;  
}  
  
... 생략 ...
```

그리드 레이아웃 속성

❖ grid-area 속성

- 2. 컨테이너 영역의 grid-template-areas 속성
 - 각 영역을 어떻게 배치할지 지정

```
/* 한 줄에 들어갈 영역을 따옴표로 묶음 */
```

```
grid-template-areas:
```

```
    "box1 box1 box1"
```

```
    "box2 box3 box3"
```

```
/* 비워 둘 영역은 마침표(.)를 넣음 */
```

```
grid-template-areas:
```

```
    "box1 box1 box1"
```

```
    "box2 . box3"
```

[실습] grid-area 속성을 이용한 배치

```
<div id="parent">  
  <div class="child box1">1</div>  
  <div class="child box2">2</div>  
  <div class="child box3">3</div>  
  <div class="child box4">4</div>  
  <div class="child box5">5</div>  
  <div class="child box6">6</div>  
</div>
```



[실습] grid-area 속성을 이용한 배치

```
#parent {  
  ... 생략 ...  
  grid-template-areas:  
    "box1 box1 box1"  
    "box2 box3 box3"  
    "box2 . box4"  
    ". box5 ."  
    "box6 . .";  
}  
  
.child { ... 생략 ... }  
  
.box1 {  
  background: pink;  
  grid-area: box1;  
}  
  
.box2 {  
  background: orange;  
  grid-area: box2;  
}
```

```
.box3 {  
  background: skyblue;  
  grid-area: box3;  
}  
  
.box4 {  
  background: yellow;  
  grid-area: box4;  
}  
  
.box5 {  
  background: greenyellow;  
  grid-area: box5;  
}  
  
.box6 {  
  background: plum;  
  grid-area: box6;  
}
```

THANK 😊 YOU