

12 반응형 웹과 미디어 쿼리

2-1 반응형 웹 알아보기

■ 반응형 웹 디자인

- 웹 사이트의 내용을 그대로 유지하면서 다양한 화면 크기에 맞게 웹 사이트를 표시하는 방법
 - 다양한 화면 크기의 모바일 기기들에 맞춘 사이트를 별도로 제작하는 것은 비효율적
 - 반응형 웹 디자인이란 화면 크기에 반응해 화면 요소들을 자동으로 바꾸어 사이트를 구현하는 것을 말함
 - 웹 요소를 화면 크기에 맞게 재배 치하고 각 요소의 표시 방법만 바꾸어 구현해 줌

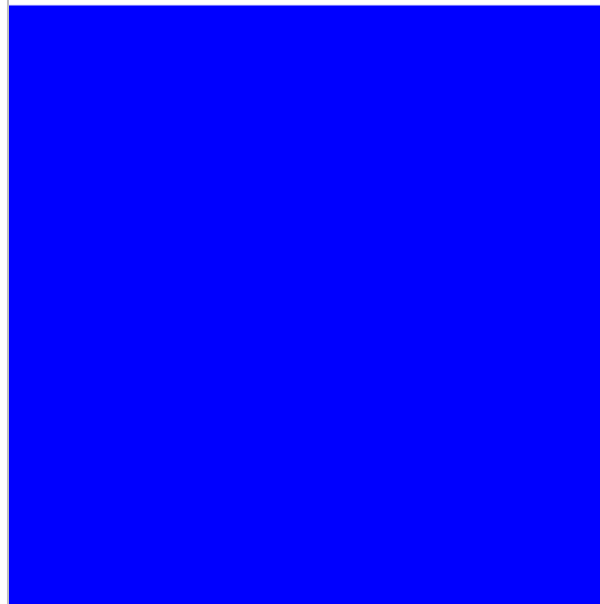
■ 모바일 기기를 위한 뷰포트(viewport)

- 뷰포트란 실제 내용이 표시되는 영역을 말함
- PC 화면과 모바일 화면의 픽셀 표시 방법이 다름
 - PC에서 제작한 문서는 모바일 화면에서 의도한대로 표시되지 않음
 - 뷰포트를 지정하면 기기 화면에 맞춰 확대/축소해서 내용 표시

```
chapter12 > <> non-viewport.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <style>
6          body { margin: 0px; padding: 0px; }
7          div {
8              background-color: blue;
9              width:300px; height:300px;
10         }
11     </style>
12 </head>
13 <body>
14     <h1>뷰포트 테스트</h1>
15     <hr>
16     <p>Lorem ipsum dolor, sit amet consectetur
17         adipisicing elit. Qui voluptatibus aliquid
18         corrupti iste quae sequi esse quo accusantium
19         cumque repudiandae earum deleniti doloribus
20         vel, maiores illo tenetur fuga velit placeat.
21     </p>
22     <div></div>
23 </body>
24 </html>
```

뷰포트 테스트

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Qui voluptatibus aliquid corrupti iste quae sequi esse quo accusantium cumque repudiandae earum deleniti doloribus vel, maiores illo tenetur fuga velit placeat.

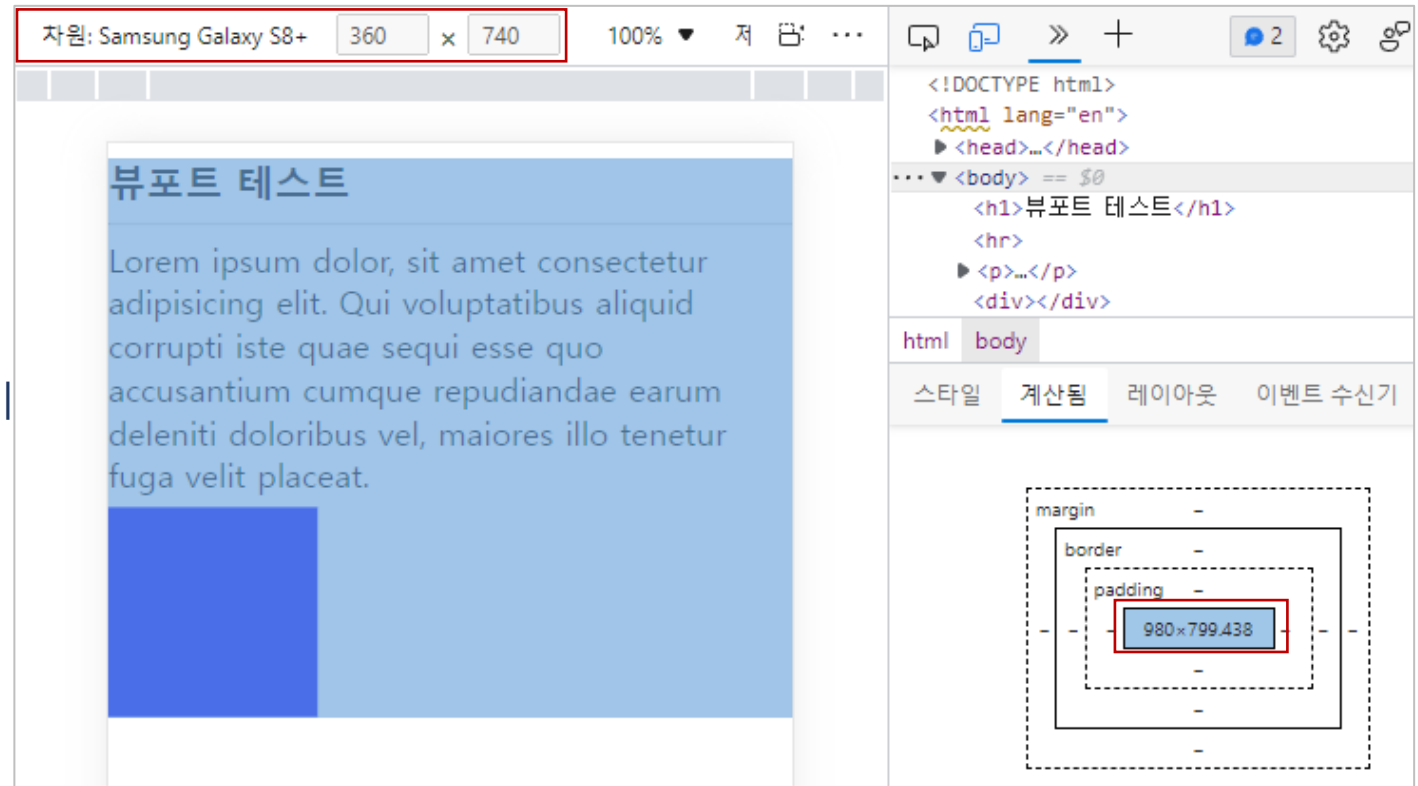


■ 모바일 폰의 기본 뷰포트 너비

- 웹킷(webkit)를 기반으로 한 모바일 브라우저의 기본 뷰포트 너비는 980px로 인식됨
 - 모바일 기기에 관계없이(물리적인 너비에 관계 없이) 980px 로 인식됨

- 뷰포트 해상도의 예

- 개발자 도구를 실행
- 모바일 기기의 해상도 : 360*720
- 뷰포트의 너비 해상도는 980으로 인식됨
- <div>의 크기를 300px로 지정했지만 1/3 정도의 크기 나타남



■ 뷰포트 지정하기

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

- content 속성을 이용해 너비와 크기 등을 지정할 수 있음

```
<meta name="viewport" content="width=device-width user-scalable=no initial-scale=1">
```

- width=device-width

width 속성으로 뷰포트의 너비를 픽셀로 지정함

width 속성의 값으로 device-width를 지정하면 뷰포트의 너비와 모바일 기기의 너비를 같게 인식함

모바일 기기의 너비에 따라 뷰포트의 너비도 증가함

- user-scalable=no

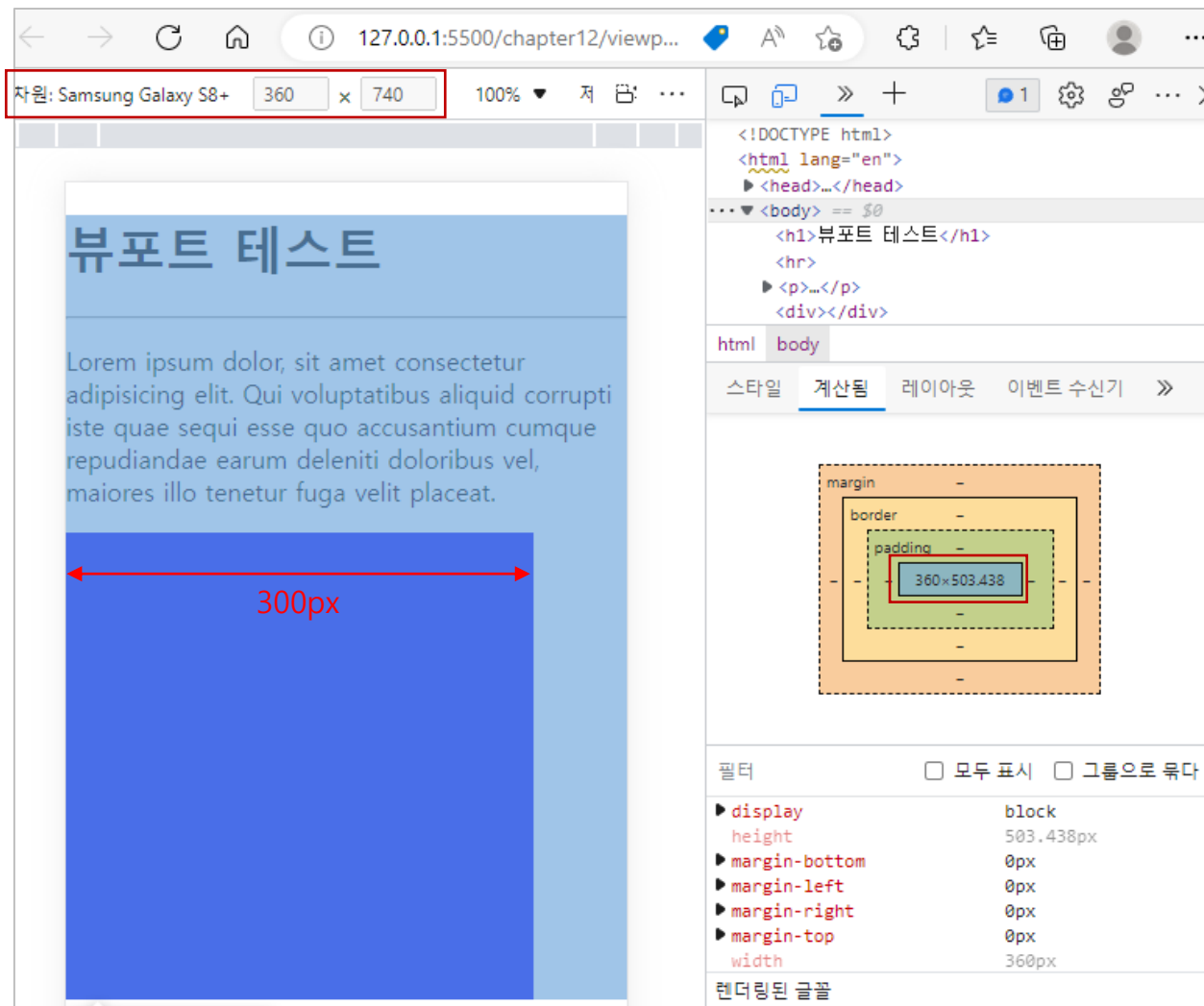
사용자가 인위적으로 확대나 축소 가능 여부를 지정 (yes: 확대 축소 가능, no: 확대 축소 불가능) (기본값 yes)

- initial-scale=1

콘텐츠의 초기 확대.축소 값을 지정 (기본값은 1)

1~10까지 지정 가능

```
chapter12 > <> viewport.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width initial-scale=1">
6   <style>
7     body { margin: 0px; padding: 0px; }
8     div {
9       background-color: blue;
10      width: 300px; height: 300px;
11    }
12  </style>
13 </head>
14 <body>
15   <h1>뷰포트 테스트</h1>
16   <hr>
17   <p>Lorem ipsum dolor, sit amet consectetur
18     adipiscing elit. Qui voluptatibus aliquid
19     corrupti iste quae sequi esse quo accusantium
20     cumque repudiandae earum deleniti doloribus
21     vel, maiores illo tenetur fuga velit placeat.
22   </p>
23   <div></div>
24 </body>
25 </html>
```



- 모바일 기기 변경

The screenshot shows a web browser at the address `127.0.0.1:5500/chapter12/viewp...`. The developer tools are open, and the 'body' element is selected. The dimensions of the body are `768 x 461.44`. A red double-headed arrow indicates a width of `300px`. The right sidebar shows the HTML structure and a box model diagram.

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body> == $0
    <h1>뷰포트 테스트</h1>
    <hr>
    <p>...</p>
    <div></div>
```

html body

스타일 계산된 레이아웃 이벤트 수신기 >>

margin -
border -
padding -
768x461.438

■ 뷰포트에서 사용하는 단위

- CSS에서는 주로 픽셀이나 %를 단위로 사용함
 - 픽셀은 모바일 기기의 해상도에 따라 크기가 다르기 때문에 사용하기 어려움
 - 주로 모바일 기기의 해상도에 따른 비율로 크기를 나타내는 vw(viewport width)와 vh(viewport height)를 사용

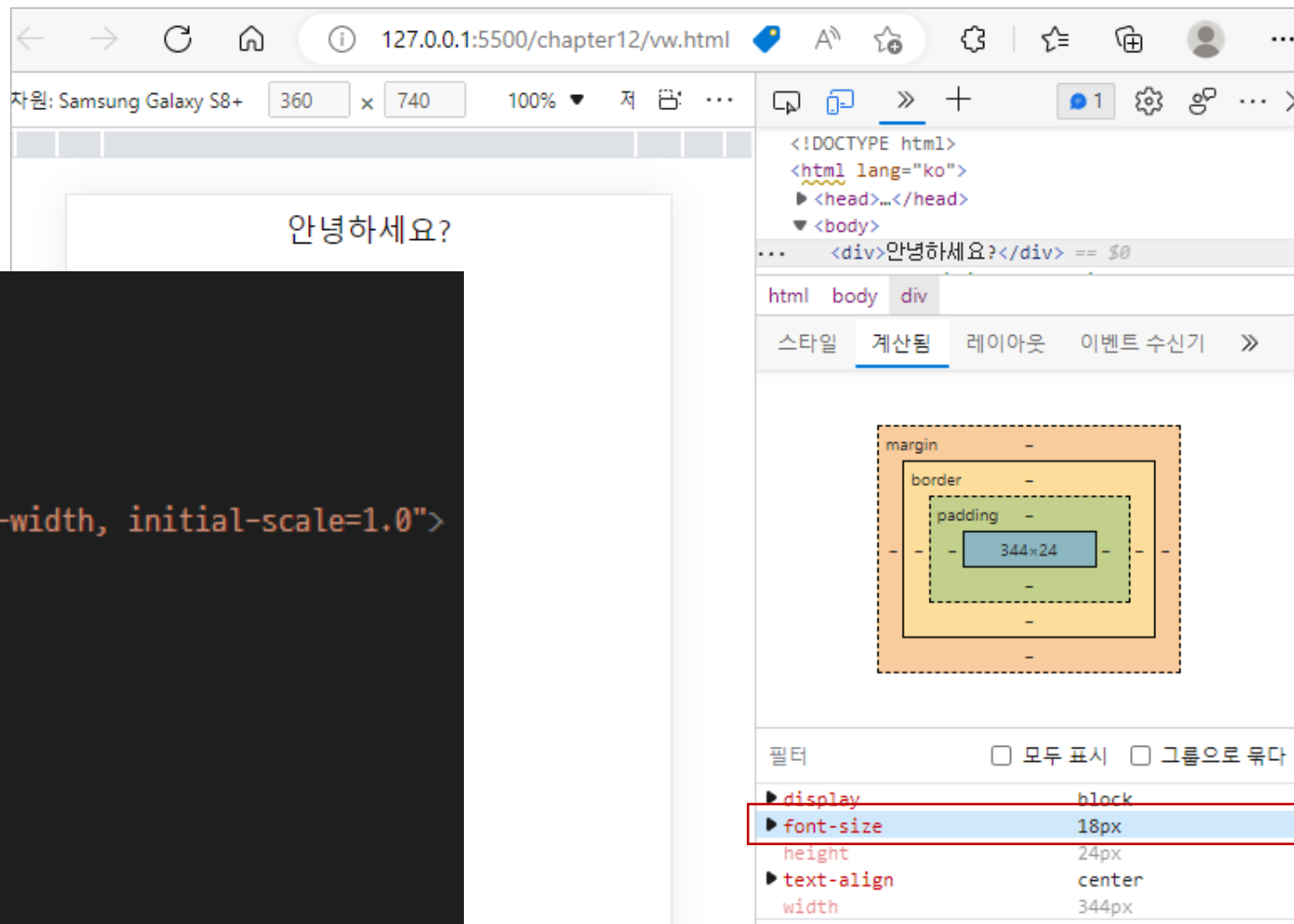
단위	설명
vw (viewport width)	뷰포트 너비(픽셀)의 1%와 같음 - 해상도가 1000 X 800 픽셀인 모바일 기기에서 1vw는 10픽셀 임
vh (viewport height)	뷰포트 높이(픽셀)의 1%와 같음 - 해상도가 1000 X 800 픽셀인 모바일 기기에서 1vh는 8픽셀 임
vmin (viewport minimum)	뷰포트의 너비와 높이 중 작은 값(픽셀)의 1%와 같음 - 해상도가 1000 X 800 픽셀인 모바일 기기에서 20vmin는 160픽셀 임
vmax (viewport maximum)	뷰포트의 너비와 높이 중 큰 값(픽셀)의 1%와 같음 - 해상도가 1000 X 800 픽셀인 모바일 기기에서 20vmax는 200픽셀 임

- 주로 vw와 vh가 사용됨

chapter12 > <> vw.html > ...

```

1  <!DOCTYPE html>
2  <html lang="ko">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <style>
7      div {
8        font-size: 5vw;
9        text-align: center;
10     }
11   </style>
12 </head>
13 <body>
14   <div>안녕하세요?</div>
15 </body>
16 </html>
    
```



12-2 미디어 쿼리 알아보기

■ 미디어 쿼리(media query)

- 사이트에 접속하는 디바이스에 따라 서로 다른 CSS 스타일이 적용되도록 하는 기법
 - 접속하는 디바이스의 화면 크기에 따라 서로 다른 레이아웃을 적용

```
@media [ only | not ] 미디어 유형 [ and (조건) ] [ and (조건) ] { 실행문 }
```

단위	설명
only	미디어 쿼리를 지원하는 미디어 유형만 미디어 쿼리 구문을 해석해 실행 - 미디어 쿼리를 지원하지 않는 미디어 유형은 미디어 쿼리를 무시하고 실행하지 않음
not	not 다음에 오는 미디어 유형을 제외하고 다른 모든 미디어 유형에서 해석해 실행
미디어 유형	미디어 쿼리가 적용되는 디바이스
and (조건)	실행문을 수행하기 위한 조건을 지정 (적용되는 조건들을 and로 결합해 표현)
실행문	조건이 참일 때 실행되는 문장 (CSS 구문)

- [예]

```
@media screen and (min-width:768px) and (max-width:1439px)
```

미디어 유형이 screen 이고 이면서 최소 너비가 768px이고 최대 너비는 1439px일 경우

■ 미디어 유형의 종류

미디어 유형	설명
all	모든 미디어 유형에서 사용할 CSS를 정의
print	인쇄장치에서 사용할 CSS를 정의
screen	컴퓨터 스크린에서 사용할 CSS를 정의 (스마트폰 스크린 포함)
tv	TV에서 사용할 CSS를 정의
aural	화면을 음성으로 읽어주는 장치에서 사용할 CSS를 정의
braille	점자 표시 장치에서 사용할 CSS를 정의
handheld	패드와 같은 장치에서 사용할 CSS를 정의
projection	프로젝터에서 사용할 CSS를 정의
tty	디스플레이 기능이 제한된 장치에서 사용할 CSS를 정의
embossed	점자 프린터에서 사용할 CSS를 정의

■ 뷰포트의 너비와 높이

종류	설명
width, height	뷰포트의 너비와 높이
min-width, min-height	뷰포트의 최소 너비와 최소 높이
max-width, max-height	뷰포트의 최대 너비와 최대 높이

- 뷰포트의 너비와 높이를 지정
 - 뷰포트의 크기는 body 태그의 콘텐츠를 표시하는 너비와 높이 실제 스크린의 크기와는 다름
 - 반응형 웹에서 일반적인 조건으로 사용됨
- 사용 예


```
@media screen and (min-width:768px) and (max-width:1024px)
```

 - 뷰포트의 너비가 768 픽셀보다 크고 1024 픽셀보다 작은 경우

■ 디바이스(스크린)의 너비와 높이

종류	설명
device-width, device-height	스크린의 너비와 높이
min-device-width, min-device-height	스크린의 최소 너비와 최소 높이
max-device-width, max-device-height	스크린의 최대 너비와 최대 높이

- 스크린의 너비와 높이를 지정
 - 대부분 단말기의 해상도와 브라우저의 너비는 다름
 - 일반적으로 뷰포트보다는 큼
- 사용 예


```
@media screen and (device-width:320px) and (device-height:480px)
```

 - 스크린 너비가 320px이고 높이가 480px인 경우

■ 화면 회전 속성

- orientation 속성을 사용해 화면을 가로나 세로로 전환 가능

종류	설명
orientation: portrait	세로 모드 스크린 지정
orientation: landscape	가로 모드 스크린 지정

■ 미디어 쿼리 중단점

- 서로 다른 CSS를 적용하는 분기점
 - 대부분 디바이스의 화면 크기를 기준으로 함
 - 모든 기기를 반영할 수 없기 때문에 스마트폰과 태블릿, 데스크톱 정도로 구분
- 모바일 우선(mobile first)
 - 모바일을 먼저 고려해 미디어 쿼리를 작성하는 방법
 - min-width를 사용
 - 모바일을 기본 CSS로 작성하고 태블릿, 데스크톱 순으로 확장
화면이 큰 기기의 CSS가 작은 기기의 CSS를 overwrite함
- 데스크톱 우선(desktop first)
 - 데스크톱을 기본 CSS로 작성하고 태블릿, 모바일 순으로 작성
 - max-width를 사용

■ 미디어 쿼리 중단점

- 서로 다른 CSS를 적용하는 분기점

- 대부분 디바이스의 화면 크기를 기준으로 함

모든 기기를 반영할 수 없으므로 스마트폰과 태블릿, 데스크톱 정도로 구분

- 모바일 우선(mobile first)

- 모바일을 먼저 고려해 미디어 쿼리를 작성하는 방법
- min-width를 사용
- 모바일을 기본 CSS로 작성하고 태블릿, 데스크톱 순으로 확장

```
.item { font-size: 0.75rem; }  
  
@media screen and (min-width:768px) { .item { font-size: 0.85rem; } }  
  
@media screen and (min-width:1025px) { .item { font-size: 1rem; } }
```

- 데스크톱 우선(desktop first)

- 데스크톱을 기본 CSS로 작성하고 태블릿, 모바일 순으로 작성
- max-width를 사용
- 데스크톱 반응형 웹에서 주로 사용

```
@media screen and (max-width:768px) { .item { font-size: 0.75rem; } }  
  
@media screen and (max-width:1025px) { .item { font-size: 0.85rem; } }  
  
.item { font-size: 1rem; }
```

- 해상도 범위 지정 방식

- 기본 CSS를 사용하지 않고 min-width 와 max-width를 모두 사용해 디바이스 별로 CSS를 적용하는 방식

```
@media screen and (min-width:767px) { .item { font-size: 0.75rem; } }  
@media screen and (min-width:768px) and (max-width:1024px) {  
    .item { font-size: 0.85rem; }  
}  
  
@media screen and (max-width:1025px) { .item { font-size: 1rem; } }
```


■ 웹 문서에 미디어 쿼리를 직접 지정

- <style> 태그에 미디어 쿼리를 직접 지정하는 방식
 - media 속성을 사용해 지정
 - 하나의 조건만 지정 가능

```
<style media="<조건>"> {  
    <스타일 규칙>  
}  
</style>
```

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

- <style>과 </style> 사이에 미디어 쿼리를 지정하는 방식
 - @media를 사용해 지정
 - 하나 이상의 조건에 따른 스타일 지정 가능

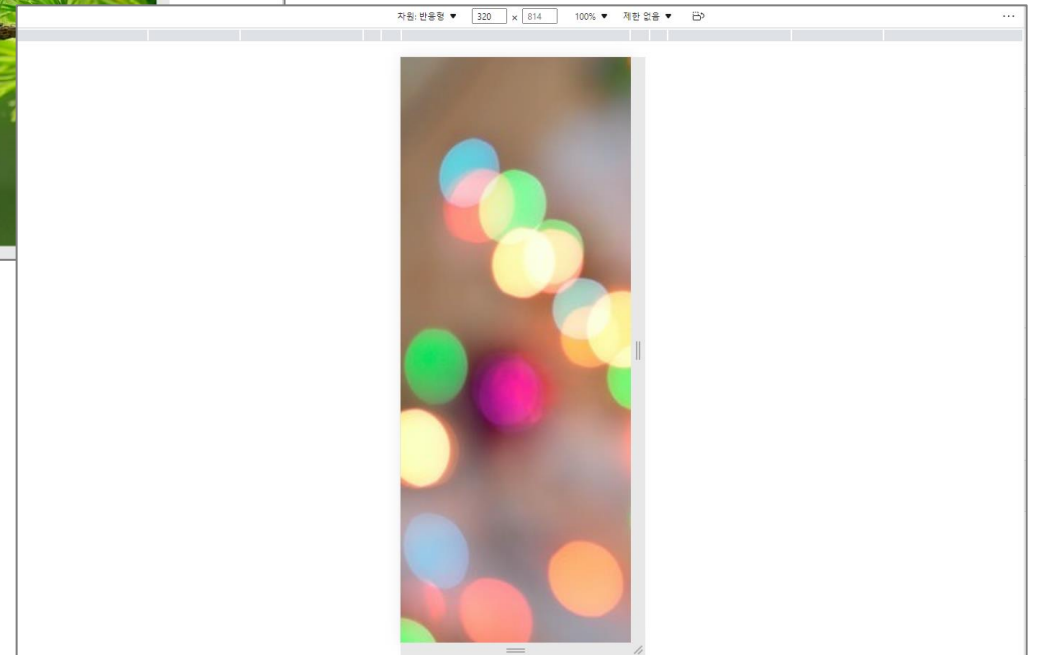
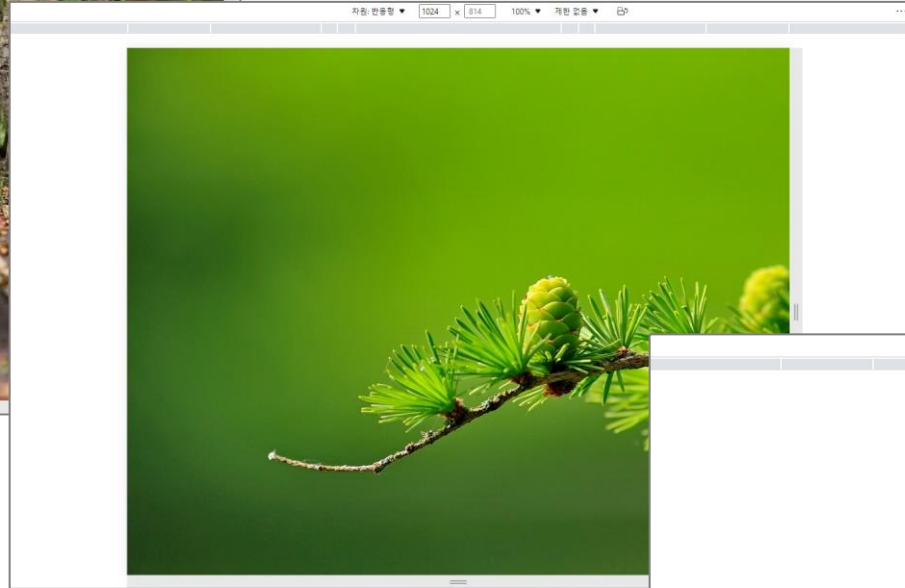
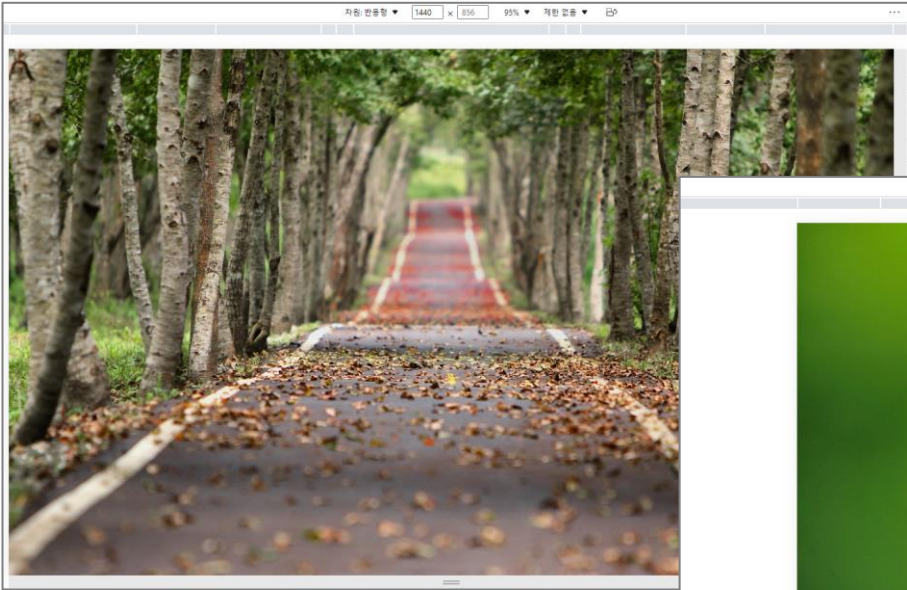
```
<style>  
  @media <조건> {  
    <스타일 규칙>  
  }  
</style>
```

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

chapter12 > <> mq-embed.html > ...

```
1  <!DOCTYPE html>
2  <html lang="ko">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>미디어 쿼리 사용하기</title>
7    <style>
8      /* 기본 배경 이미지 지정 */
9      body {
10        background: url(images/bg0.jpg) no-repeat fixed;
11        background-size: cover;
12      }
13      /* 가로가 1024px 이하면 bg1.jpg 지정 */
14      @media screen and (max-width:1024px) {
15        body {
16          background: url(images/bg1.jpg) no-repeat fixed;
17          background-size: cover;
18        }
19      }
20      /* 가로가 768px 이하면 bg2.jpg 지정 */
21      @media screen and (max-width:768px) {
22        body {
23          background: url(images/bg2.jpg) no-repeat fixed;
24          background-size:cover;
25        }
26      }
```

```
27      /* 가로가 320px 이하면 bg3.jpg 지정 */
28      @media screen and (max-width:320px) {
29        body {
30          background: url(images/bg3.jpg) no-repeat fixed;
31          background-size: cover;
32        }
33      }
34    </style>
35  </head>
36  <body>
37
38  </body>
39  </html>
```



■ <link> 태그를 사용하는 방법

```
<link rel="stylesheet" media="미디어 쿼리 조건" href="CSS파일">
```

- 주로 사용되는 방법임

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

■ @import 문을 사용하는 방법

```
@import url(CSS파일) 미디어 쿼리 조건
```

```
@import url(css/tablet.css) only screen and (min-width:311px) and (max-width:768px)
```

chapter12 > css > # mq.css > ...

```
1  body {
2      background: url(../images/bg0.jpg) no-repeat fixed;
3      background-size: cover;
4  }
5  /* 가로가 1024px 이하면 bg1.jpg 지정 */
6  @media screen and (max-width:1024px) {
7      body {
8          background: url(../images/bg1.jpg) no-repeat fixed;
9          background-size: cover;
10     }
11 }
12 /* 가로가 768px 이하면 bg2.jpg 지정 */
13 @media screen and (max-width:768px) {
14     body {
15         background: url(../images/bg2.jpg) no-repeat fixed;
16         background-size: cover;
17     }
18 }
19 /* 가로가 320px 이하면 bg3.jpg 지정 */
20 @media screen and (max-width:320px) {
21     body {
22         background: url(../images/bg3.jpg) no-repeat fixed;
23         background-size: cover;
24     }
25 }
```

chapter12 > <> mq-link1.html > ...

```
1  <!DOCTYPE html>
2  <html lang="ko">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <link rel="stylesheet" media="screen" href="css/mq.css">
7  </head>
8  <body>
9
10 </body>
11 </html>
```

chapter12 > css > # mq-default.css > ...

```
1 body {  
2     background: url(../images/bg0.jpg) no-repeat fixed;  
3     background-size: cover;  
4 }
```

chapter12 > css > # mq-1024.css > ...

```
1 body {  
2     background: url(../images/bg1.jpg) no-repeat fixed;  
3     background-size: cover;  
4 }
```

chapter12 > css > # mq-768.css > ...

```
1 body {  
2     background: url(../images/bg2.jpg) no-repeat fixed;  
3     background-size: cover;  
4 }
```

chapter12 > css > # mq-320.css > ...

```
1 body {  
2     background: url(../images/bg3.jpg) no-repeat fixed;  
3     background-size: cover;  
4 }
```

chapter12 > <> mq-link2.html > ...

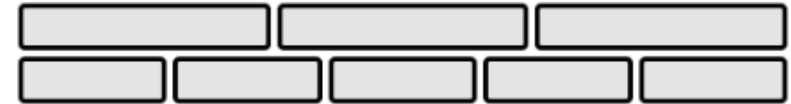
```
1 <!DOCTYPE html>  
2 <html lang="ko">  
3 <head>  
4     <meta charset="UTF-8">  
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">  
6     <link rel="stylesheet" href="css/mq-default.css">  
7     <link rel="stylesheet" media="screen and (max-width:1024px)" href="css/mq-1024.css">  
8     <link rel="stylesheet" media="screen and (max-width:768px)" href="css/mq-768.css">  
9     <link rel="stylesheet" media="screen and (max-width:320px)" href="css/mq-320.css">  
10 </head>  
11 <body>  
12  
13 </body>  
14 </html>
```

12-3 그리드 레이아웃 알아보기

■ 그리드 레이아웃(grid layout)

- 반응형 웹 디자인에서 웹 문서 요소를 배치하는 기준
 - 웹 사이트 화면을 여러 개의 칼럼(column)으로 나눈 후 요소를 배치
 - 규칙적으로 배열하므로 레이아웃을 일관성 있게 유지할 수 있음
- 그리드 레이아웃의 특징
 - 시각적으로 안정된 디자인
 - 업데이트가 편한 웹 디자인 구성
레이아웃을 먼저 작성하고 콘텐츠를 배치
 - 요소를 자유롭게 배치
하나의 라인에 하나 이상의 요소를 배치
 - 플렉스 박스 레이아웃과 CSS 그리드 레이아웃이 있음

- 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



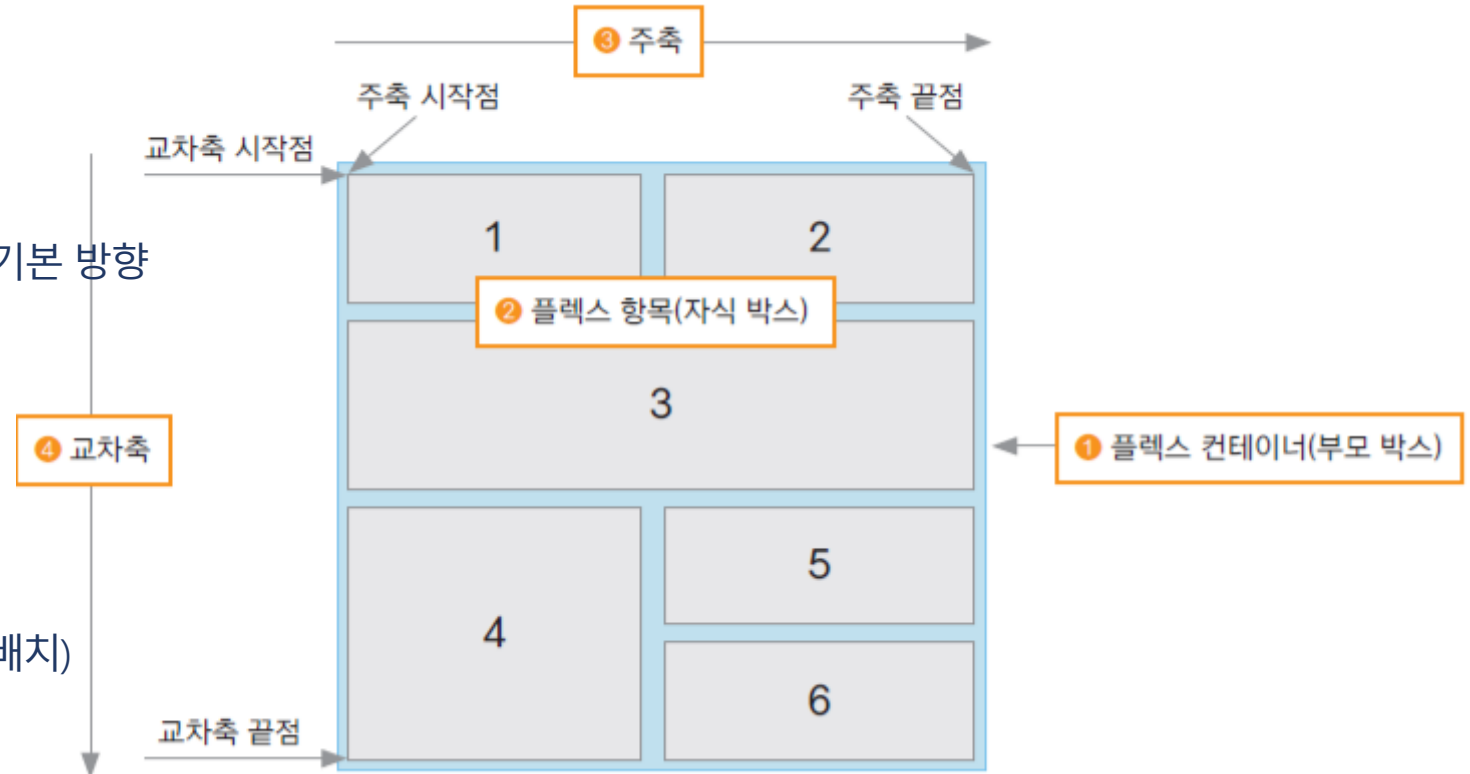
- CSS 그리드 레이아웃
 - 최근에 등장한 CSS 표준
 - 수평 방향이나 수직 방향 어디로든 배치 가능
 - 마치 레고 블록을 끼워 맞추듯 요소를 배치할 수 있음



12-4 플렉스 박스 레이아웃 알아보기

■ 플렉스 박스 레이아웃의 용어

- 플렉스 컨테이너(부모 박스)
 - 플렉스 박스 레이아웃을 적용할 대상을 묶는 요소
- 플렉스 항목(자식 박스)
 - 플렉스 박스 레이아웃을 적용할 대상
- 주축
 - 플렉스 박스 컨테이너서 플렉스 항목을 배치하는 기본 방향
 - 기본적으로 왼쪽에서 오른쪽으로 수평방향 배치
 - 주축 시작점 : 배치가 시작되는 위치
 - 주축 끝점 : 배치가 끝나는 위치
- 교차축
 - 주축과 교차하는 방향(기본적으로 위로 아래로 배치)
 - 교차축 시작점 : 배치가 시작되는 위치
 - 교차축 끝점 : 배치가 끝나는 위치



■ display 속성

display : flex | inline-flex

- 플렉스 항목들을 감싸는 플렉스 컨테이너의 정렬 방식을 지정
 - 플렉스 컨테이너가 하나 이상일 때 블록 레벨 요소로 배치할 지 아니면 인라인 블록 레벨 요소로 배치할지를 지정
 - 컨테이너 항목에는 영향을 주지 않음

종류	설명
flex	플렉스 컨테이너를 블록 요소로 지정
inline-flex	플렉스 컨테이너를 인라인 블록 요소로 지정

chapter12 > <> display.html > ...

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <style>
7          .container {
8              width:200px;
9              display:flex;
10             background-color:■ #eee;
11             border:1px solid □ #222;
12             margin-bottom: 5px;
13         }
14         .box {
15             margin: 5px; width:50px; height:50px;
16             background-color:□ #222;
17         }
18         p { color:■ #fff; text-align: center;}
19     </style>
20 </head>
21 <body>
22     <div class="container" id="opt1">
23         <div class="box"><p>1</p></div>
24         <div class="box"><p>2</p></div>
25         <div class="box"><p>3</p></div>
26     </div>

```

```

27     <div class="container" id="opt2">
28         <div class="box"><p>1</p></div>
29         <div class="box"><p>2</p></div>
30         <div class="box"><p>3</p></div>
31     </div>
32     <div class="container" id="opt3">
33         <div class="box"><p>1</p></div>
34         <div class="box"><p>2</p></div>
35         <div class="box"><p>3</p></div>
36     </div>
37 </body>
38 </html>

```

```
display:flex;
```



```
display:inline-flex;
```



■ flex-direction 속성

flex-direction : row | row-reverse | column | column-reverse

- 플렉스 컨테이너 안에서 플렉스 항목의 배치를 위해 주축과 방향 지정

종류	설명
row	주축을 가로로 지정하고, 플렉스 항목을 왼쪽에서 오른쪽으로 배치 (기본값)
row-reverse	주축을 가로로 지정하고, 플렉스 항목을 오른쪽에서 왼쪽으로 배치
column	주축을 세로로 지정하고, 플렉스 항목을 위쪽에서 아래쪽으로 배치
column-reverse	주축을 세로로 지정하고, 플렉스 항목을 아래쪽에서 위쪽으로 배치

chapter12 > <> flex-direction.html > ...

```

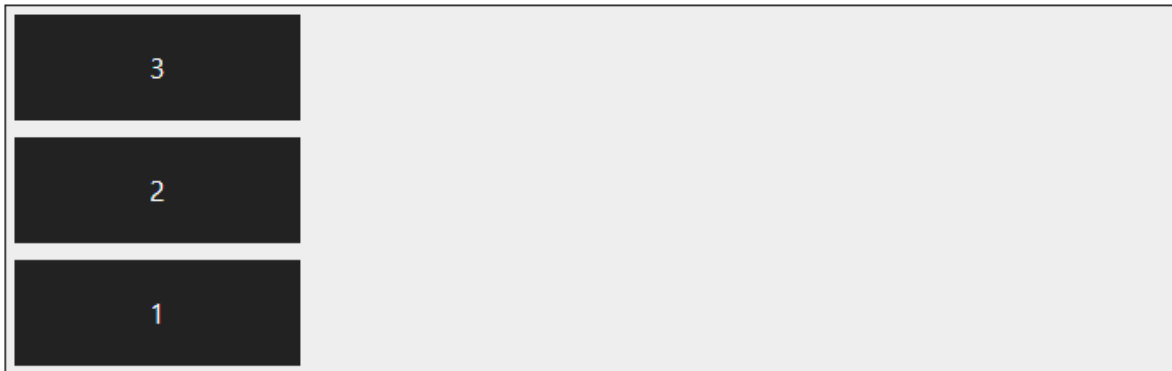
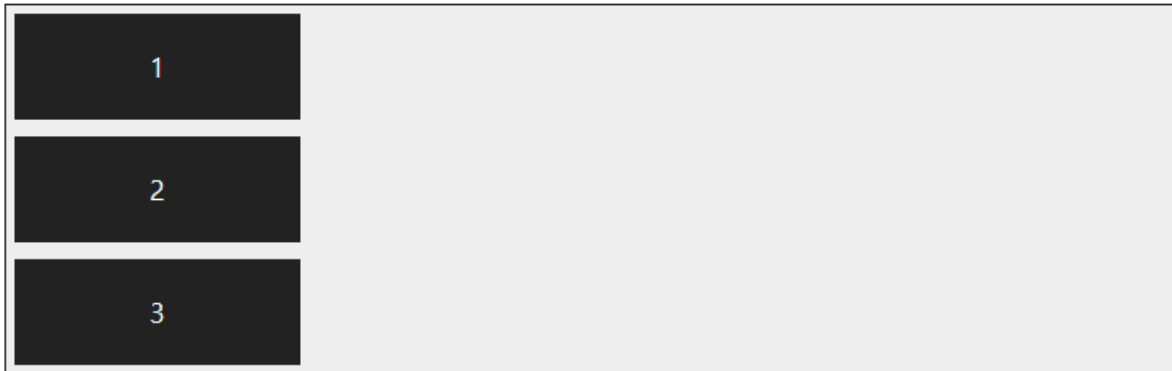
1  <!DOCTYPE html>
2  <html lang="ko">
3  <head>
4    <meta charset="UTF-8">
5    <title>플렉스 박스 레이아웃</title>
6    <meta name="viewport" content="width=device-width,initial-scale=1">
7    <style>
8      .container {
9        width:700px;
10       display:flex; /* 플렉스 컨테이너 지정 */
11       background-color:■ #eee;
12       border:1px solid □ #222;
13       margin-bottom:30px;
14     }
15     .box {
16       padding:5px 45px;
17       margin:5px;
18       width:80px;
19       background-color:□ #222;
20     }
21     #opt1{ flex-direction: row; }           /*왼쪽에서 오른쪽으로*/
22     #opt2{ flex-direction: row-reverse; }   /*오른쪽에서 왼쪽으로*/
23     #opt3{ flex-direction: column; }        /*위에서 아래로*/
24     #opt4{ flex-direction: column-reverse; } /*아래에서 위로*/
25     p { color:■ #fff; text-align: center; }
26   </style>
27 </head>

```

```

28 <body>
29   <div class="container" id="opt1">
30     <div class="box"><p>1</p></div>
31     <div class="box"><p>2</p></div>
32     <div class="box"><p>3</p></div>
33   </div>
34   <div class="container" id="opt2">
35     <div class="box"><p>1</p></div>
36     <div class="box"><p>2</p></div>
37     <div class="box"><p>3</p></div>
38   </div>
39   <div class="container" id="opt3">
40     <div class="box"><p>1</p></div>
41     <div class="box"><p>2</p></div>
42     <div class="box"><p>3</p></div>
43   </div>
44   <div class="container" id="opt4">
45     <div class="box"><p>1</p></div>
46     <div class="box"><p>2</p></div>
47     <div class="box"><p>3</p></div>
48   </div>
49 </body>
50 </html>

```



■ flex-wrap 속성

flex-wrap : nowrap | wrap | wrap-reverse

- 플렉스 항목이 많아 플렉스 컨테이너의 너비를 초과할 경우 줄 바꿈 여부를 지정
 - 브라우저의 너비를 넓히거나 줄일 때 플렉스 항목들의 줄 바꿈 발생 여부를 지정

종류	설명
nowrap	플렉스 항목들의 너비가 플렉스 컨테이너의 너비보다 커도 줄 바꿈이 발생하지 않음 (기본값)
wrap	플렉스 항목들의 너비가 플렉스 컨테이너의 너비보다 크면 바꿈이 발생
wrap-reverse	플렉스 항목들의 너비가 플렉스 컨테이너의 너비보다 크면 바꿈이 발생하되 시작점과 끝점이 바뀜

chapter12 > flex-wrap.html > html > body > div#opt1.container

```

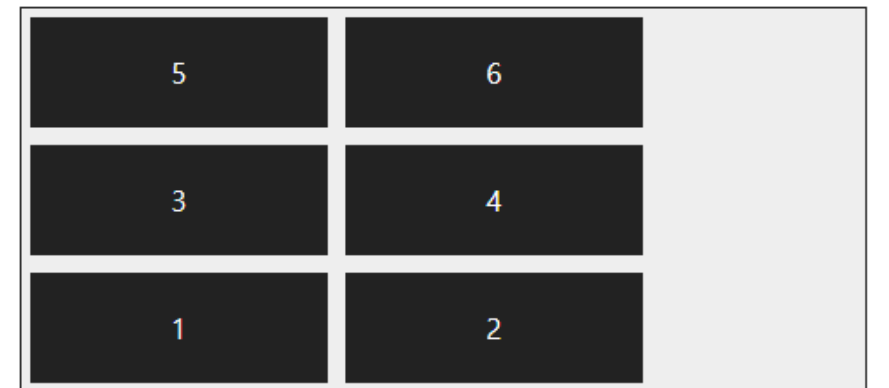
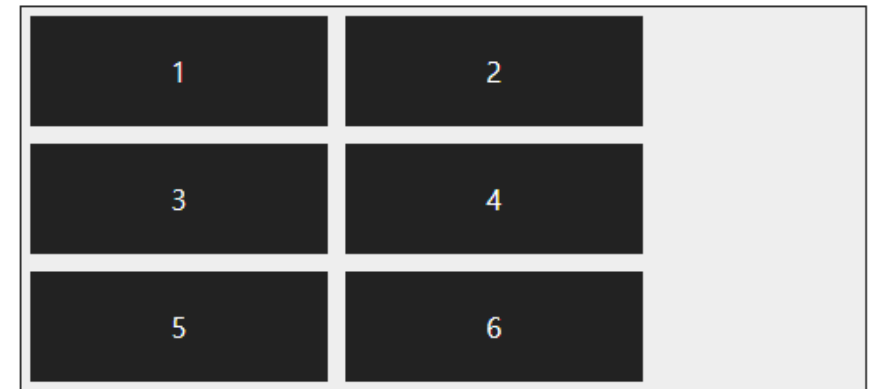
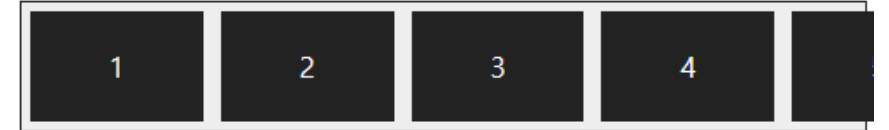
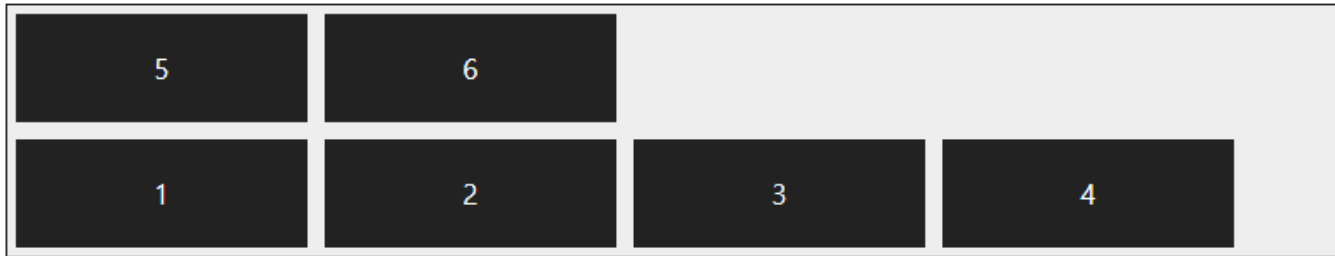
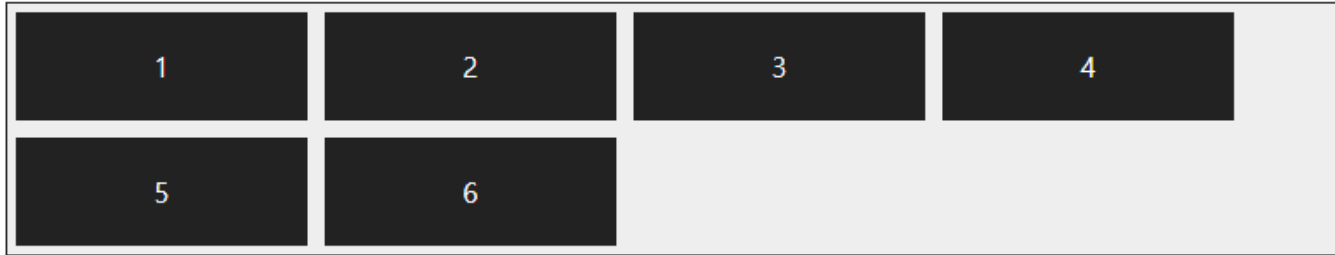
1  <!DOCTYPE html>
2  <html lang="ko">
3  <head>
4    <meta charset="UTF-8">
5    <title>플렉스 박스 레이아웃</title>
6    <meta name="viewport" content="width=device-width,initial-scale=1">
7    <style>
8      .container {
9        display: flex;
10       background-color: #eee;
11       border: 1px solid #222;
12       margin-bottom: 30px;
13     }
14     .box {
15       padding: 5px 45px;
16       margin: 5px;
17       width: 80px;
18       background-color: #222;
19     }
20     #opt1{ flex-wrap: nowrap; }      /*한 줄에 표시*/
21     #opt2{ flex-wrap: wrap; }        /*여러 줄에 표시*/
22     #opt3{ flex-wrap: wrap-reverse; } /*시작점 끝점 바꿔 여러줄 표시*/
23     p {
24       color: #fff; text-align: center;
25     }
26   </style>
27 </head>

```

```

28 <body>
29   <div class="container" id="opt1">
30     <div class="box"><p>1</p></div>
31     <div class="box"><p>2</p></div>
32     <div class="box"><p>3</p></div>
33     <div class="box"><p>4</p></div>
34     <div class="box"><p>5</p></div>
35     <div class="box"><p>6</p></div>
36   </div>
37   <div class="container" id="opt2">
38     <div class="box"><p>1</p></div>
39     <div class="box"><p>2</p></div>
40     <div class="box"><p>3</p></div>
41     <div class="box"><p>4</p></div>
42     <div class="box"><p>5</p></div>
43     <div class="box"><p>6</p></div>
44   </div>
45   <div class="container" id="opt3">
46     <div class="box"><p>1</p></div>
47     <div class="box"><p>2</p></div>
48     <div class="box"><p>3</p></div>
49     <div class="box"><p>4</p></div>
50     <div class="box"><p>5</p></div>
51     <div class="box"><p>6</p></div>
52   </div>
53 </body>
54 </html>

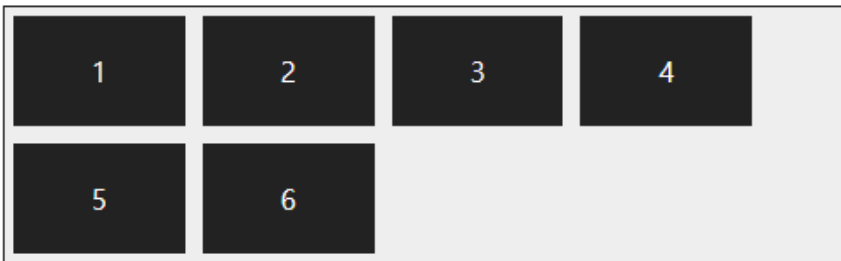
```



flex-flow 속성

flex-flow : <flex-direction> <flex-wrap>

- flex-direction 속성과 flex-wrap 속성을 동시에 지정



```
chapter12 > <> flex-flow.html > ...
1  <!DOCTYPE html>
2  <html lang="ko">
3  <head>
4    <meta charset="UTF-8">
5    <title>플렉스 박스 레이아웃</title>
6    <meta name="viewport" content="width=device-width,initial-scale=1">
7    <style>
8      .container {
9        display: flex;
10       background-color: #eee; border: 1px solid #222;
11       margin-bottom: 10px;
12       flex-flow: row wrap;
13     }
14     .box { padding: 5px 45px; margin: 5px; background-color: #222; }
15     p { color: #fff; text-align: center; }
16   </style>
17 </head>
18 <body>
19   <div class="container">
20     <div class="box"><p>1</p></div>
21     <div class="box"><p>2</p></div>
22     <div class="box"><p>3</p></div>
23     <div class="box"><p>4</p></div>
24     <div class="box"><p>5</p></div>
25     <div class="box"><p>6</p></div>
26   </div>
27 </body>
28 </html>
```

■ justify-content 속성

justify-content : flex-start | flex-end | center | space-between | space-around

- 주축을 기준으로 플렉스 항목의 정렬 방식을 지정
 - 가로 정렬 방식

종류	설명
flex-start	주축의 시작점에 맞춰 배치 (왼쪽 → 오른쪽)
flex-end	주축의 끝점에 맞춰 배치 (오른쪽 → 왼쪽)
center	주축의 중앙에 맞춰 배치
space-between	첫 번째 항목은 시작점에 배치하고 마지막 항목은 끝점에 배치. 나머지 항목은 사이에서 같은 간격으로 배치
space-around	모든 항목을 주축 내에서 같은 간격으로 배치

chapter12 > <> flex-justify-content.html > ...

```

1  <!DOCTYPE html>
2  <html lang="ko">
3  <head>
4    <meta charset="UTF-8">
5    <title>플렉스 박스 레이아웃</title>
6    <meta name="viewport" content="width=device-width,initial-scale=1">
7    <style>
8      .container {
9        display: flex;
10       background-color: #eee;
11       border: 1px solid #222;
12       margin-bottom: 30px;
13     }
14     #opt1{ justify-content: flex-start;}
15     #opt2{ justify-content: flex-end; }
16     #opt3{ justify-content: center; }
17     #opt4{ justify-content: space-between; }
18     #opt5{ justify-content: space-around; }
19     .box {
20       padding: 5px 45px; margin: 5px; background-color: #222;
21     }
22     p {
23       color: #fff; text-align: center;
24     }
25   </style>
26 </head>
27 <body>
28   <div class="container" id="opt1">
29     <div class="box"><p>1</p></div>

```

```

30   <div class="box"><p>2</p></div>
31   <div class="box"><p>3</p></div>
32   <div class="box"><p>4</p></div>
33 </div>
34 <div class="container" id="opt2">
35   <div class="box"><p>1</p></div>
36   <div class="box"><p>2</p></div>
37   <div class="box"><p>3</p></div>
38   <div class="box"><p>4</p></div>
39 </div>
40 <div class="container" id="opt3">
41   <div class="box"><p>1</p></div>
42   <div class="box"><p>2</p></div>
43   <div class="box"><p>3</p></div>
44   <div class="box"><p>4</p></div>
45 </div>
46 <div class="container" id="opt4">
47   <div class="box"><p>1</p></div>
48   <div class="box"><p>2</p></div>
49   <div class="box"><p>3</p></div>
50   <div class="box"><p>4</p></div>
51 </div>
52 <div class="container" id="opt5">
53   <div class="box"><p>1</p></div>
54   <div class="box"><p>2</p></div>
55   <div class="box"><p>3</p></div>
56   <div class="box"><p>4</p></div>
57 </div>
58 </body>
59 </html>

```



■ align-items 속성

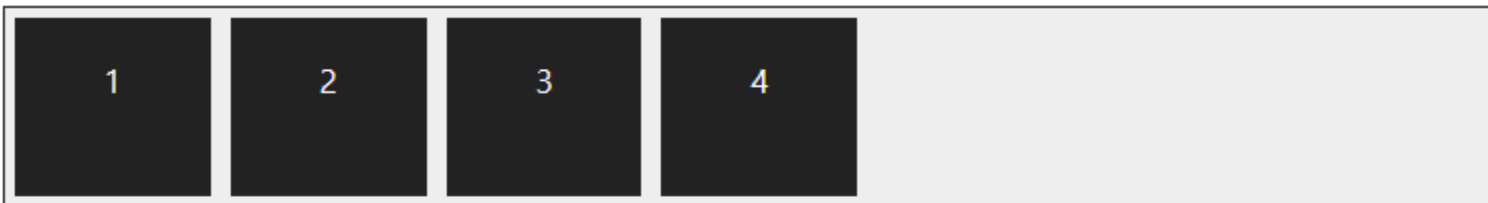
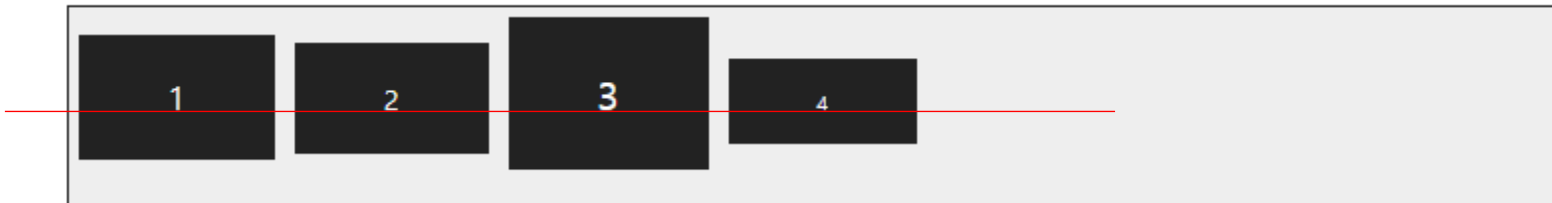
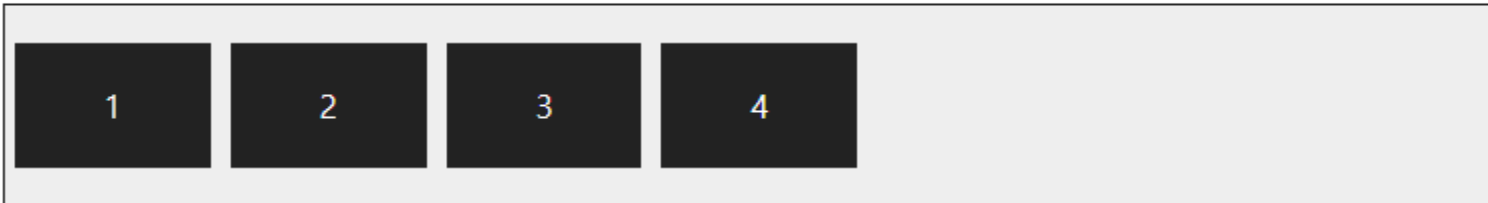
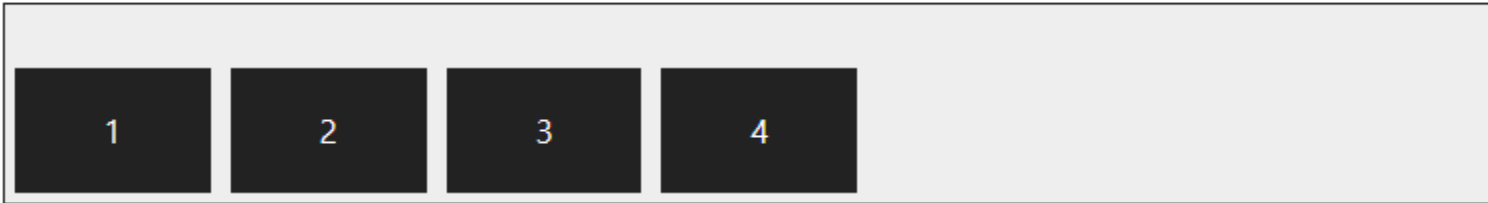
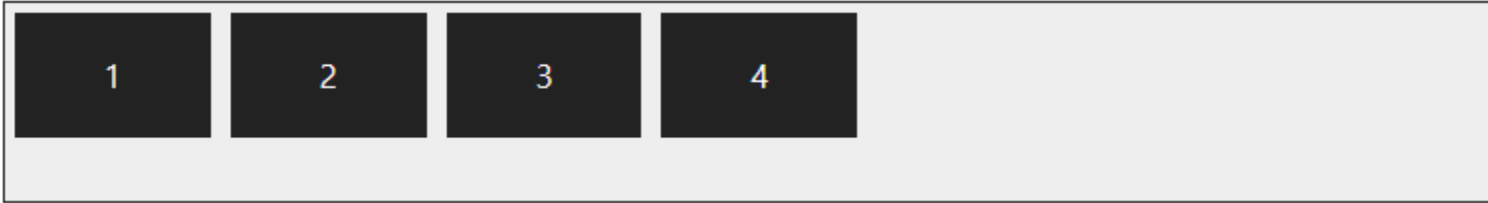
align-items : flex-start | flex-end | center | baseline | stretch

- 교차축을 기준으로 플렉스 항목의 정렬 방식을 지정
 - 세로 정렬 방식

종류	설명
flex-start	교차축의 시작점에 맞춰 배치 (위쪽 배치)
flex-end	교차축의 끝점에 맞춰 배치 (아래쪽 배치)
center	교차축의 중앙에 맞춰 배치
baseline	교차축의 문자 기준선에 맞춰 배치
stretch	플렉스 항목을 늘려 교차축에 가득 차게 재치


```
chapter12 > <> align-items.html > ...
1  <!DOCTYPE html>
2  <html lang="ko">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1">
6    <style>
7      .container {
8        width:100%; height:100px;
9        display:flex;
10       background-color:■ #eee; border:1px solid □ #222;
11       margin-bottom:20px;
12     }
13     #opt1{ align-items: flex-start;}
14     #opt2{ align-items: flex-end; }
15     #opt3{ align-items: center; }
16     #opt4{ align-items: baseline; }
17     #opt5{ align-items: stretch; }
18     .box {
19       padding:5px 45px; margin:5px; background-color:□ #222;
20     }
21     p { color:■ #fff; text-align: center; }
22   </style>
23 </head>
24 <body>
25   <div class="container" id="opt1">
26     <div class="box"><p>1</p></div>
27     <div class="box"><p>2</p></div>
28     <div class="box"><p>3</p></div>
```

```
29   <div class="box"><p>4</p></div>
30 </div>
31 <div class="container" id="opt2">
32   <div class="box"><p>1</p></div>
33   <div class="box"><p>2</p></div>
34   <div class="box"><p>3</p></div>
35   <div class="box"><p>4</p></div>
36 </div>
37 <div class="container" id="opt3">
38   <div class="box"><p>1</p></div>
39   <div class="box"><p>2</p></div>
40   <div class="box"><p>3</p></div>
41   <div class="box"><p>4</p></div>
42 </div>
43 <div class="container" id="opt4">
44   <div class="box"><p>1</p></div>
45   <div class="box"><p style="font-size:14px;">2</p></div>
46   <div class="box"><p style="font-size:20px;">3</p></div>
47   <div class="box"><p style="font-size:10px;">4</p></div>
48 </div>
49 <div class="container" id="opt5">
50   <div class="box"><p>1</p></div>
51   <div class="box"><p>2</p></div>
52   <div class="box"><p>3</p></div>
53   <div class="box"><p>4</p></div>
54 </div>
55 </body>
56 </html>
```



■ align-self 속성

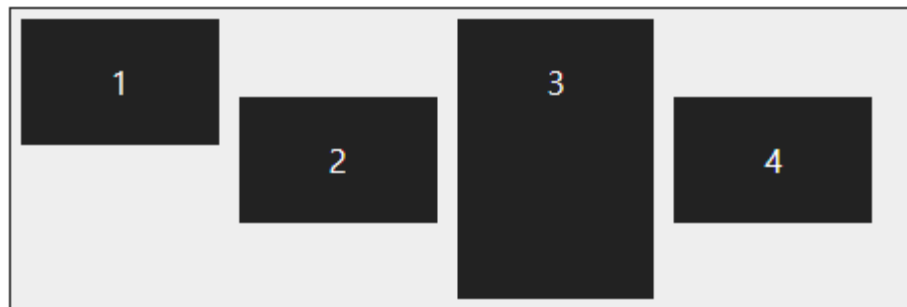
align-self : <align-items 속성값>

- align-item 속성으로 교차점 기준 정렬을 수행하되 특정 항목만 다른 속성 값으로 지정할 경우 사용

chapter12 > <> align-self.html > ...

```
1  <!DOCTYPE html>
2  <html lang="ko">
3  <head>
4    <meta charset="UTF-8">
5    <title>플렉스 박스 레이아웃</title>
6    <meta name="viewport" content="width=device-width, initial-scale=1">
7    <style>
8      .container {
9        width:450px; height:150px;
10       background-color: #eee;
11       border:1px solid #222;
12       margin-bottom:20px;
13       display:flex; align-items: center;
14     }
15     .box {
16       padding:5px 45px; margin:5px;
17       background-color: #222;
18     }
19     #box1 { align-self: flex-start; }
20     #box3 { align-self:stretch; }
21     p {
22       color: #fff;
23       text-align: center;
24     }
25   </style>
26 </head>
```

```
27 <body>
28   <div class="container">
29     <div class="box" id="box1"><p>1</p></div>
30     <div class="box"><p>2</p></div>
31     <div class="box" id="box3"><p>3</p></div>
32     <div class="box"><p>4</p></div>
33   </div>
34 </body>
35 </html>
```



■ align-content 속성

align-content : flex-start | flex-end | center | space-between | space-around | stretch

- 주축을 기준으로 줄 바꿈이 발생해 플렉스 항목이 여러 줄로 표현되었을 때 교차축에서 플렉스 항목 사이의 간격을 지정

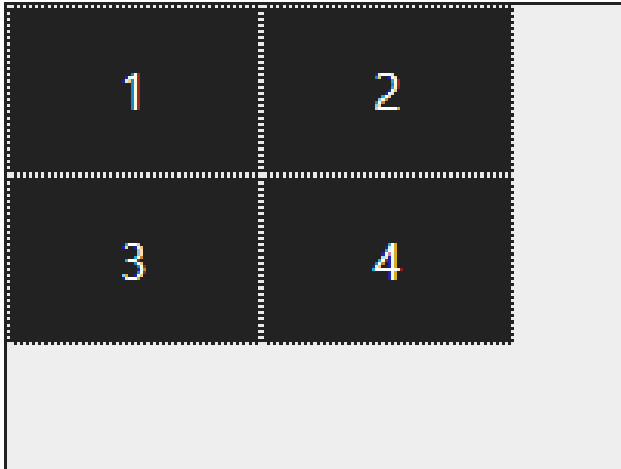
종류	설명
flex-start	교차축의 시작점에 맞춰 배치
flex-end	교차축의 끝점에 맞춰 배치
center	교차축의 중앙에 맞춰 배치
space-between	첫 번째 항목은 교차점의 시작점에 배치하고 마지막 항목은 교차점의 끝점에 배치 나머지 항목은 사이에서 같은 간격으로 배치
space-around	모든 항목을 교차축 내에서 같은 간격으로 배치
stretch	플렉스 항목을 늘려 교차축에 가득 차게 배치

chapter12 > <> align-content.html > ...

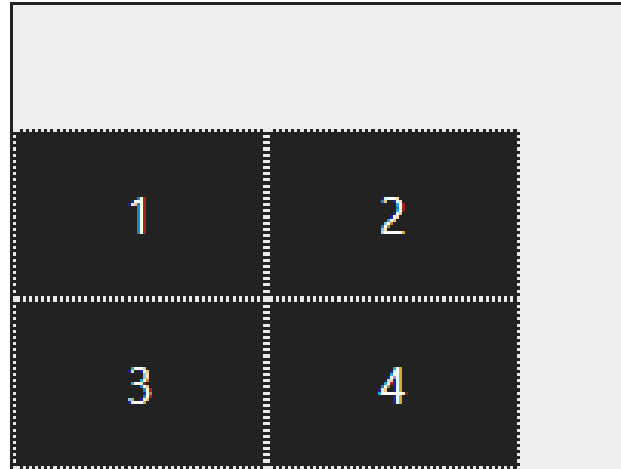
```
1  <!DOCTYPE html>
2  <html lang="ko">
3  <head>
4    <meta charset="UTF-8">
5    <title>플렉스 박스 레이아웃</title>
6    <meta name="viewport" content="width=device-width,initial-scale=1">
7    <style>
8      .container {
9        float:left;
10       width:200px; height:150px;
11       display:flex;
12       flex-flow: row wrap;
13       border:1px solid #222;
14       background-color: #eee;
15       margin:30px;
16       align-content: flex-start;
17     }
18     .box {
19       width:80px;
20       background-color: #222;
21       border:1px dotted #e9e9e9;
22     }
23     p {
24       color: #fff; text-align: center;
25     }
26   </style>
27 </head>
```

```
28   <body>
29     <div class="container">
30       <div class="box"><p>1</p></div>
31       <div class="box"><p>2</p></div>
32       <div class="box"><p>3</p></div>
33       <div class="box"><p>4</p></div>
34     </div>
35   </body>
36 </html>
```

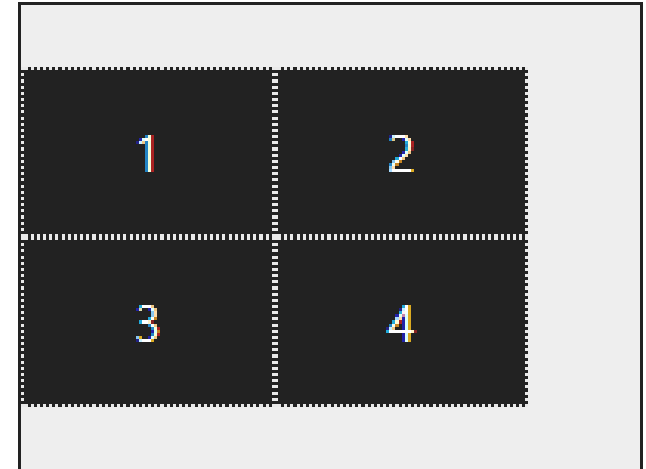
`align-content: flex-start;`



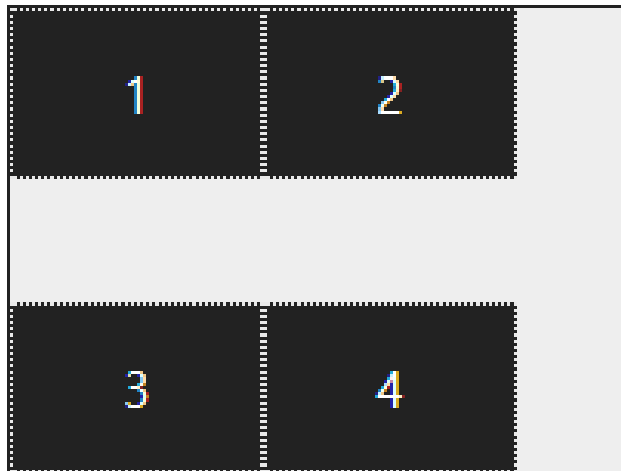
`align-content: flex-end;`



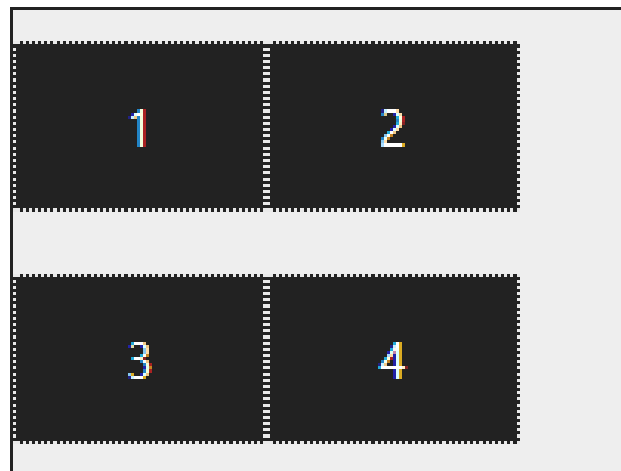
`align-content: center;`



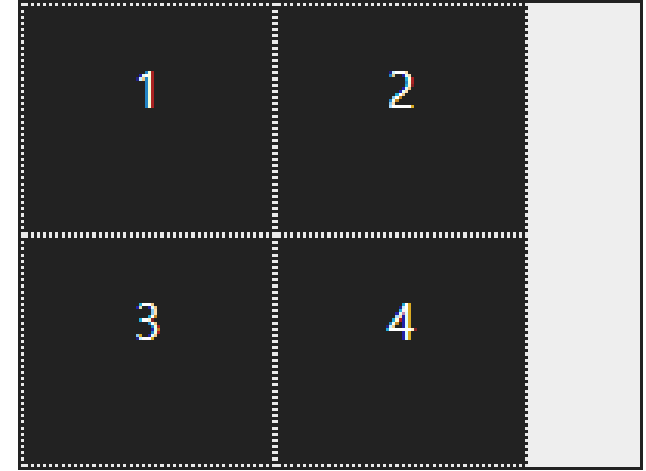
`align-content: space-between;`



`align-content: space-around;`



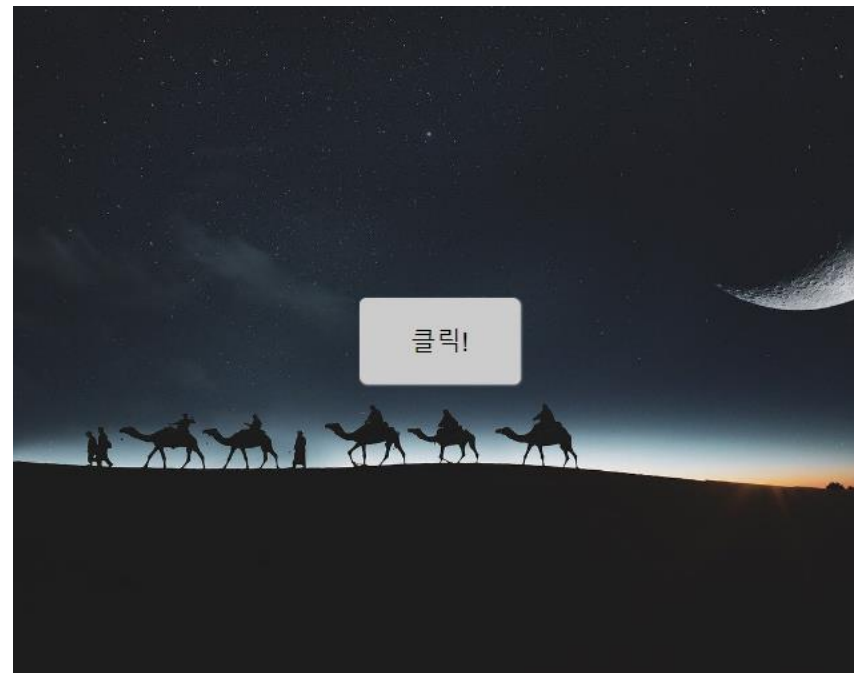
`align-content: stretch;`



■ 플렉스 레이아웃을 사용해 항상 중앙에 표시

```
chapter12 > <> center.html > ...
1  <!DOCTYPE html>
2  <html lang="ko">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width,initial-scale=1.0">
6    <style>
7      * { margin:0; box-sizing: border-box; }
8      body {
9        background:url('images/bg5.jpg') no-repeat left top fixed;
10       background-size:cover;
11       display: flex;
12       justify-content: center;
13       align-items: center;
14       min-width: 100wh;
15       min-height: 100vh;
16     }
17     button {
18       background-color: #ccc;
19       font-size: 1.2em;
20       padding: 1em 2em;
21       border:none; border-radius:5px; box-shadow:1px 1px 2px #fff;
22     }
23   </style>
24 </head>
```

```
25 <body>
26   <button>클릭!</button>
27 </body>
28 </html>
```



12-5 CSS 그리드 레이아웃 사용하기

■ CSS 그리드 레이아웃

- 항목을 배치할 때 가로와 세로를 모두 사용해 배치
 - 플렉스 박스 레이아웃에서는 가로나 세로 중 하나를 기준으로 배치함
- 가로 방향을 나타내는 row와 세로 방향을 나타내는 column을 사용해 구성
 - 줄 사이의 여백과 컬럼 사이의 여백을 조절



■ display 속성

display : grid | inline-grid

- 그리드 컨테이너의 배치를 지정하기 위한 속성

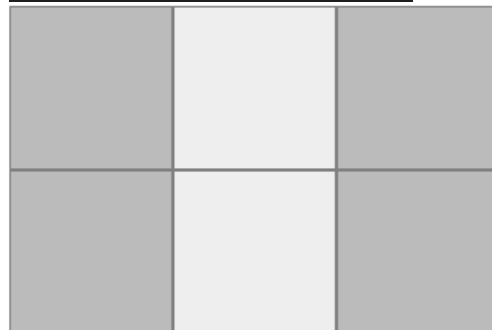
종류	설명
grid	그리드 컨테이너를 블록 요소로 지정
inline-grid	그리드 컨테이너를 인라인 블록 요소로 지정

chapter12 > <> grid-display.html > ...

```
1 <!DOCTYPE html>
2 <html lang="ko">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width,initial-scale=1">
6   <title>CSS Grid Layout</title>
7   <style>
8     #wrapper{
9       display:grid;
10      grid-template-columns: 100px 100px 100px;
11      grid-template-rows:100px;
12    }
13    .items{
14      padding:10px;
15      background-color: #eee;
16      border:1px solid grey;
17    }
18    .items:nth-child(odd){
19      background-color: #bbb; }
20  </style>
21 </head>
22 <body>
23   <div id="wrapper">
24     <div class="items"></div>
25     <div class="items"></div>
26     <div class="items"></div>
27   </div>
```

```
28   <div id="wrapper">
29     <div class="items"></div>
30     <div class="items"></div>
31     <div class="items"></div>
32   </div>
33 </body>
34 </html>
```

display:grid;



display:inline-grid;



■ grid-template-columns/ grid-templates-rows 속성

grid-template-columns : <컬럼 너비>

grid-template-rows : <컬럼 높이>

- grid-template-columns 속성
 - 컬럼의 너비를 지정
 - 첫 번째 컬럼부터 마지막 컬럼까지 컬럼의 수만큼 지정하며 너비를 지정
- grid-templates-rows 속성
 - 행의 높이를 지정
 - 첫 번째 행부터 마지막 행까지 높이를 지정
 - 높이를 지정하지 않으면 auto로 지정됨 (콘텐츠의 내용에 따라 결정)

chapter12 > <> grid-template-columns.html > ...

```

1  <!DOCTYPE html>
2  <html lang="ko">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1">
6    <style>
7      #wrapper{
8        display:grid;
9        grid-template-columns: 200px 200px 200px;
10       grid-template-rows:100px;
11     }
12     .items{ padding:10px; background-color: #eee; border:1px solid grey }
13     .items:nth-child(odd){ background-color: #bbb; }
14   </style>
15 </head>
16 <body>
17   <div id="wrapper">
18     <div class="items">Lorem ipsum dolor sit amet consectetur adipisicing elit. Amet, reprehenderit.Lorem ipsum dolor,
19       sit amet consectetur adipisicing elit. </div>
20     <div class="items">Lorem ipsum dolor, sit amet consectetur adipisicing elit.Lorem ipsum dolor, sit amet consectetur
21       adipisicing elit</div>
22     <div class="items">Lorem ipsum dolor sit amet.Lorem ipsum dolor, sit amet consectetur adipisicing elit</div>
23     <div class="items">Lorem ipsum dolor sit.Lorem ipsum dolor, sit amet consectetur adipisicing elit</div>
24     <div class="items">Lorem, ipsum dolor.</div>
25   </div>
26 </body>
27 </html>

```

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

■ 상대적인 크기를 지정

- fr(fraction) 단위
 - 픽셀은 고정적이므로 반응형 웹 디자인에 맞지 않으므로 상대적인 크기를 지정할 수 있는 fr 단위를 사용

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

3개의 같은 크기의 컬럼을 배치

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

3개의 컬럼을 2:1:2 비율로 배치

- 값이 반복될 때 줄여서 표현하는 repeat() 함수

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

=

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

- 최대값과 최소값을 지정하는 minmax() 함수

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

줄 높이는 최소 100 픽셀로 지정하고, 내용이 많으면 높이가 자동으로 늘어남

chapter12 > <> grid-minmax.html > ...

```

1  <!DOCTYPE html>
2  <html lang="ko">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1">
6    <style>
7      #wrapper{
8        width:600px;
9        display:grid;
10       grid-template-columns: repeat(3, 1fr);
11       grid-template-rows: minmax(100px, auto);
12     }
13     .items{ padding:10px; background-color: #eee; border:1px solid grey }
14     .items:nth-child(odd){ background-color: #bbb; }
15   </style>
16 </head>
17 <body>
18   <div id="wrapper">
19     <div class="items">Lorem ipsum dolor sit amet consectetur adipisicing elit. Amet, reprehenderit.Lorem ipsum dolor,
20       sit amet consectetur adipisicing elit. </div>
21     <div class="items">Lorem ipsum dolor, sit amet consectetur adipisicing elit.Lorem ipsum dolor, sit amet consectetur
22       adipisicing elit</div>
23     <div class="items">Lorem ipsum dolor sit amet.Lorem ipsum dolor, sit amet consectetur adipisicing elit</div>
24     <div class="items">Lorem ipsum dolor sit.Lorem ipsum dolor, sit amet consectetur adipisicing elit</div>
25     <div class="items">Lorem, ipsum dolor.</div>
26   </div>
27 </body>
28 </html>

```

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

■ auto-fit/ auto-fill 속성

```
grid-template-columns : repeat( auto-fit, 크기 )
```

```
grid-template-columns : repeat( auto-fill, 크기 )
```

- 화면 너비에 따라 컬럼 개수를 자동으로 지정
- auto-fit 속성
 - 화면이 넓을 때 남는 공간 없이 가득 채워 컬럼을 표시
- auto-fill 속성
 - 화면이 넓을 때 컬럼의 최소 너비만 표시하고 남는 공간은 그대로 둠

chapter12 > <> auto-fit-fill.html > ...

```
1  <!DOCTYPE html>
2  <html lang="ko">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1">
6    <style>
7      #wrapper1{
8        display:grid;
9        grid-template-columns:repeat(auto-fit, minmax(200px, 1fr));
10       margin-bottom:20px;
11     }
12     #wrapper2{
13       display:grid;
14       grid-template-columns:repeat(auto-fill, minmax(200px, 1fr));
15     }
16     h1 {
17       font-size:24px;
18     }
19     .items{
20       padding:15px;
21       background-color: #eee;
22     }
23     .items:nth-child(odd){
24       background-color: #bbb;
25     }
26   </style>
27 </head>
```

```
28 <body>
29   <h1>auto-fit일 때</h1>
30   <div id="wrapper1">
31     <div class="items">Lorem ipsum dolor sit amet, consectetur adipisicing elit. Soluta nesciunt magni voluptatem.</div>
32     <div class="items">Lorem ipsum dolor sit amet consectetur adipisicing elit.</div>
33     <div class="items">Lorem ipsum dolor, sit amet consectetur adipisicing elit. Eius nulla corrupti porro!</div>
34     <div class="items">Lorem ipsum dolor sit.</div>
35     <div class="items">Lorem ipsum dolor sit amet consectetur adipisicing elit. Accusantium, quos!</div>
36     <div class="items">Lorem, ipsum dolor sit amet consectetur adipisicing elit. Porro odit quam sit.</div>
37   </div>
38
39   <h1>auto-fill일 때</h1>
40   <div id="wrapper2">
41     <div class="items">Lorem ipsum dolor sit amet, consectetur adipisicing elit. Soluta nesciunt magni voluptatem.</div>
42     <div class="items">Lorem ipsum dolor sit amet consectetur adipisicing elit.</div>
43     <div class="items">Lorem ipsum dolor, sit amet consectetur adipisicing elit. Eius nulla corrupti porro!</div>
44     <div class="items">Lorem ipsum dolor sit.</div>
45     <div class="items">Lorem ipsum dolor sit amet consectetur adipisicing elit. Accusantium, quos!</div>
46     <div class="items">Lorem, ipsum dolor sit amet consectetur adipisicing elit. Porro odit quam sit.</div>
47   </div>
48 </body>
49 </html>
```

auto-fit일 때

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Soluta nesciunt magni voluptatem.

Lorem ipsum dolor sit amet consectetur adipiscing elit.

Lorem ipsum dolor, sit amet consectetur adipiscing elit. Eius nulla corrupti porro!

Lorem ipsum dolor sit.

Lorem ipsum dolor sit amet consectetur adipiscing elit. Accusantium, quos!

Lorem, ipsum dolor sit amet consectetur adipiscing elit. Porro odit quam sit.

auto-fill일 때

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Soluta nesciunt magni voluptatem.

Lorem ipsum dolor sit amet consectetur adipiscing elit.

Lorem ipsum dolor, sit amet consectetur adipiscing elit. Eius nulla corrupti porro!

Lorem ipsum dolor sit.

Lorem ipsum dolor sit amet consectetur adipiscing elit. Accusantium, quos!

Lorem, ipsum dolor sit amet consectetur adipiscing elit. Porro odit quam sit.

■ grid-column-gap/ grid-row-gap/ grid-gap 속성

grid-column-gap : <컬럼 간격 크기>

grid-row-gap : <줄 간격 크기>

grid-gap : <줄 간격 크기> <컬럼 간격 크기>

- 항목이 간격을 조절하기 위한 속성

종류	설명
grid-column-gap	컬럼과 컬럼 사이의 간격을 지정
grid-row-gap	줄과 줄 사이의 간격을 지정
grid-gap	컬럼 간격과 줄 사이의 간격을 동시에 지정

```
chapter12 > <> grid-gap.html > ...
1  <!DOCTYPE html>
2  <html lang="ko">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1">
6    <style>
7      #wrapper{
8        display:grid;
9        grid-template-columns:repeat(3, 200px);
10       grid-template-rows: minmax(100px, auto);
11       grid-gap:20px 30px;
12     }
13     .items{ padding:15px; background-color:■ #eee; }
14     .items:nth-child(odd){ background-color:■ #bbb; }
15   </style>
16 </head>
17 <body>
18   <div id="wrapper">
19     <div class="items">Lorem ipsum dolor sit amet, consectetur adipisicing elit. Soluta nesciunt magni voluptatem.</div>
20     <div class="items">Lorem ipsum dolor sit amet consectetur adipisicing elit.</div>
21     <div class="items">Lorem ipsum dolor, sit amet consectetur adipisicing elit. Eius nulla corrupti porro!</div>
22     <div class="items">Lorem ipsum dolor sit.</div>
23     <div class="items">Lorem ipsum dolor sit amet consectetur adipisicing elit. Accusantium, quos!</div>
24     <div class="items">Lorem, ipsum dolor sit amet consectetur adipisicing elit. Porro odit quam sit.</div>
25   </div>
26 </body>
27 </html>
```

Lorem ipsum dolor sit
amet, consectetur
adipisicing elit. Soluta
nesciunt magni
voluptatem.

Lorem ipsum dolor sit
amet consectetur
adipisicing elit.

Lorem ipsum dolor, sit
amet consectetur
adipisicing elit. Eius
nulla corrupti porro!

Lorem ipsum dolor sit.

Lorem ipsum dolor sit
amet consectetur
adipisicing elit.
Accusantium, quos!

Lorem, ipsum dolor sit
amet consectetur
adipisicing elit. Porro
odit quam sit.

■ CSS 그리드 라인을 이용한 배치

- 그리드 레이아웃은 눈에 보이지 않는 그리드 라인이 포함되어 있음
 - 그리드 라인을 이용해 배치를 수행할 수 있음
 - 여러 개의 영역을 하나로 합쳐서 배치할 수 있음
- 가로와 세로로 그리드를 생성한 다음 배치 속성을 사용해 배치

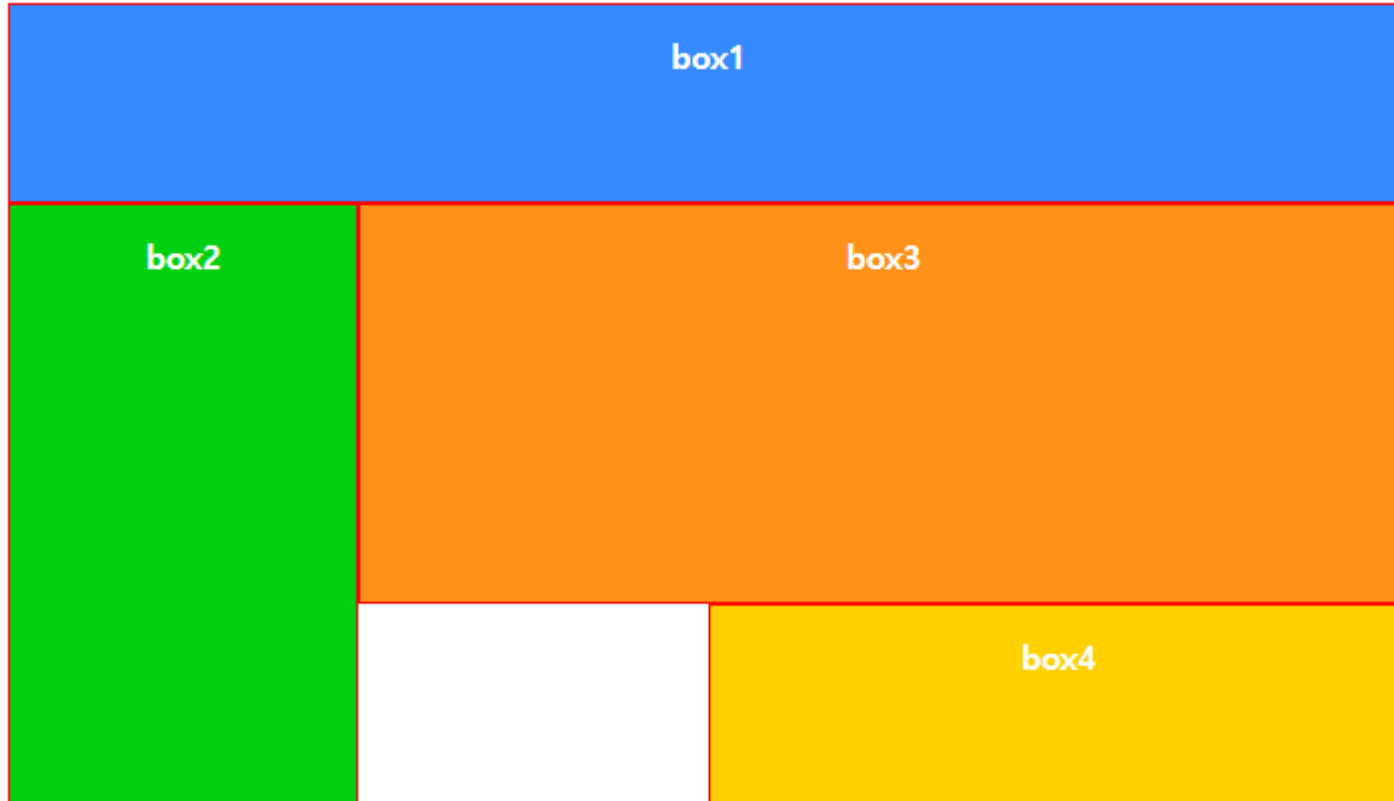


종류	설명	예
grid-column-start	컬럼 시작의 라인 번호를 지정	grid-column-start:1
grid-column-end	컬럼 마지막의 라인 번호를 지정	grid-column-end:4
grid-column	컬럼의 시작 번호와 끝 번호를 슬래시(/)로 구분해 지정	grid-column:1/4
grid-row-start	줄 시작의 라인 번호를 지정	grid-row-start:2
grid-row-end	줄 마지막의 라인 번호를 지정	grid-row-end:4
grid-row	줄의 시작 번호와 끝 번호를 슬래시(/)로 구분해 지정	grid-row:2/4

chapter12 > <> grid-guidline.html > ...

```
1  <!DOCTYPE html>
2  <html lang="ko">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1">
6    <style>
7      #wrapper{
8        width:700px; display:grid;
9        grid-template-columns:repeat(4, 1fr);
10       grid-template-rows:repeat(4, 100px);
11      }
12      .box{
13        padding:15px; color:■ #fff;
14        font-weight:bold;
15        text-align:center;
16        border:1px solid ■ red;
17      }
18      .box1 {
19        background-color:■ #3689ff;
20        grid-column:1/5;
21      }
22      .box2 {
23        background-color:■ #00cf12;
24        grid-row:2/5;
25        grid-column-start:1;
26        grid-column-end: 2; /*하나의 줄이나 칸은 생략 가능*/
27      }
```

```
28 .box3 {
29     background-color: ■ #ff9019;
30     grid-column: 2/5;
31     grid-row: 2/4;
32 }
33 .box4 {
34     background-color: ■ #ffd000;
35     grid-column: 3/5;
36     grid-row-start: 4;
37     grid-row-end: 5; /*하나의 줄이나 칸은 생략 가능*/
38 }
39 </style>
40 </head>
41 <body>
42     <div id="wrapper">
43         <div class="box box1">box1</div>
44         <div class="box box2">box2</div>
45         <div class="box box3">box3</div>
46         <div class="box box4">box4</div>
47     </div>
48 </body>
49 </html>
```



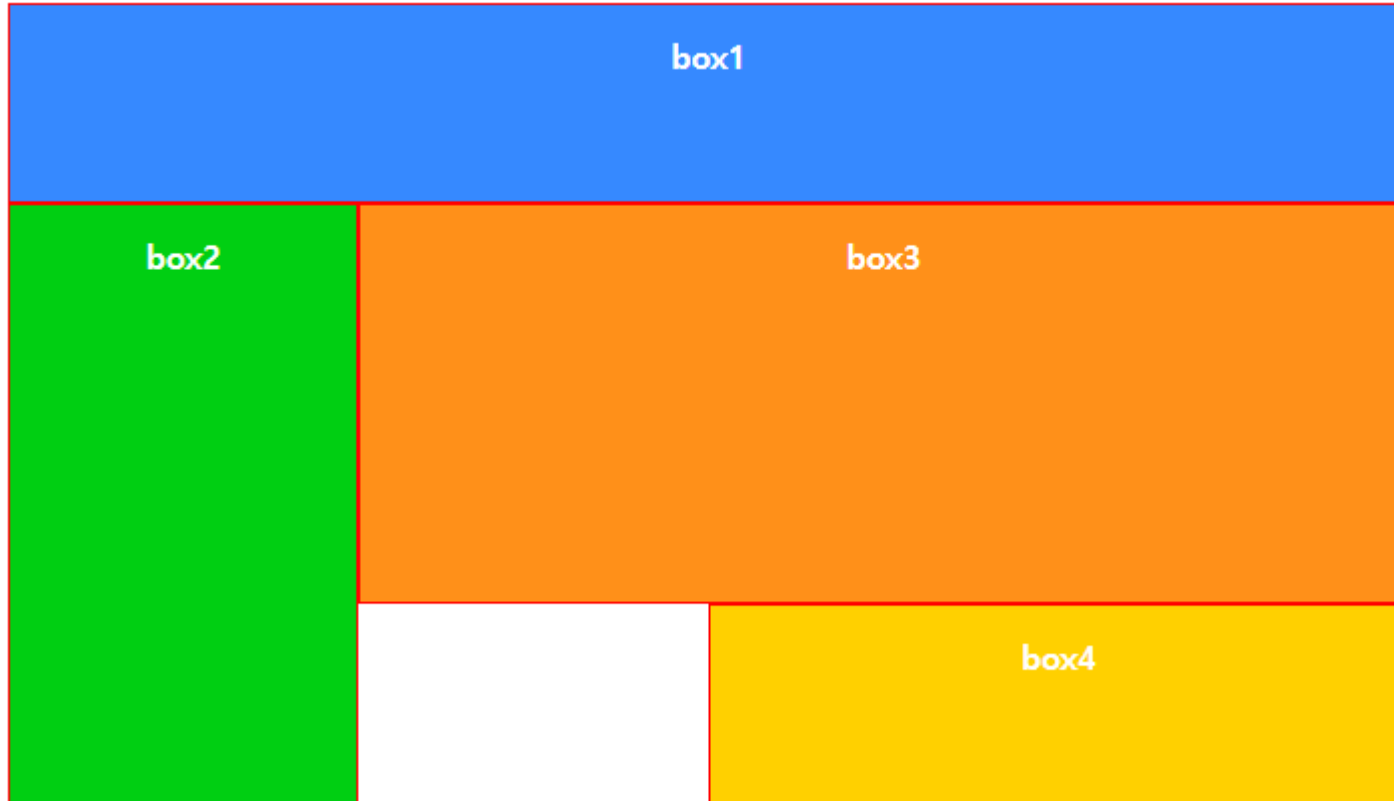
■ 템플릿 영역을 만들어 배치

- 그리드 라인보다 쉽게 배치하는 방법
 - 배치할 템플릿 영역을 생성해 배치
- grid-area 속성을 사용해 각 템플릿 영역에 이름을 지정
 - 템플릿 이름을 이용해 영역을 배치함
- grid-template-area 속성의 값으로 템플릿 영역 배치 위치를 지정
 - 각 줄을 구성하는 컬럼들은 grid-area 속성으로 생성한 템플릿 이름으로 지정
 - 각 줄을 구성하는 컬럼들은 이중인용부로 내에 함께 표현
 - 비어있는 영역(템플릿 영역이 배치 되지 않은 곳은 마침표를 입력

chapter12 > <> template-area.html > ...

```
1 <!DOCTYPE html>
2 <html lang="ko">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width,initial-scale=1">
6   <style>
7     #wrapper{
8       width:700px;
9       display:grid;
10      grid-template-columns:repeat(4, 1fr);
11      grid-template-rows:repeat(4, 100px);
12      grid-template-areas:
13        "box1 box1 box1 box1"
14        "box2 box3 box3 box3"
15        "box2 box3 box3 box3"
16        "box2 . box4 box4";
17    }
18    .box{
19      padding:15px; color:■ #fff;
20      font-weight:bold;
21      text-align:center;
22      border:1px solid ■ red;
23    }
24    .box1 {
25      background-color:■ #3689ff;
26      grid-area:box1;
27    }
```

```
28    .box2 {
29      background-color:■ #00cf12;
30      grid-area:box2;
31    }
32    .box3 {
33      background-color:■ #ff9019;
34      grid-area:box3;
35    }
36    .box4 {
37      background-color:■ #ffd000;
38      grid-area:box4;
39    }
40  </style>
41 </head>
42 <body>
43   <div id="wrapper">
44     <div class="box box1">box1</div>
45     <div class="box box2">box2</div>
46     <div class="box box3">box3</div>
47     <div class="box box4">box4</div>
48   </div>
49 </body>
50 </html>
```



```
chapter12 > css > # gallery-result.css > ...
```

```
1  * {  
2    box-sizing: border-box;  
3  }  
4  body {  
5    background-color: #eee;  
6    font-size: 16px;  
7  }  
8  #wrapper {  
9    display: grid;  
10   grid-template-columns: repeat(auto-fit, minmax(320px, 1fr));  
11   grid-gap: 1rem;  
12 }  
13 .card {  
14   background-color: #fff;  
15   box-shadow: 0px 1px 5px #222;  
16 }  
17 .card > header {  
18   font-size: 1.5rem;  
19   padding: 0.5rem;  
20 }  
21 .card > p {  
22   padding: 0.5rem;  
23   line-height: 1.6em;  
24 }  
25 .card > img {  
26   max-width: 100%;  
27 }
```

chapter12 > <> grid-gallery-result.html > html > body > div#wrapper > div.card > p

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1">
6    <title>상품 갤러리 만들기</title>
7    <link rel="stylesheet" href="css/gallery-result.css">
8  </head>
9  <body>
10   <div id="wrapper">
11     <div class="card">
12       <header>
13         <h3>사진 제목</h3>
14       </header>
15       <figure>
16         
17       </figure>
18       <p>사진 설명 : Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ullam, numquam.
19         Neque mollitia esse blanditiis facere.</p>
20     </div>
21
22     <div class="card">
23       <header>
24         <h3>사진 제목</h3>
25       </header>
26       <figure>
27         
28       </figure>
```



```
29     <p>사진 설명 : Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ullam, numquam.  
30     Neque mollitia esse blanditiis facere.</p>  
31 </div>  
32  
33 <div class="card">  
34     <header>  
35     | <h3>사진 제목</h3>  
36     </header>  
37     <figure>  
38     |   
39     </figure>  
40     <p>사진 설명 : Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ullam, numquam.  
41     Neque mollitia esse blanditiis facere.</p>  
42 </div>  
43  
44 <div class="card">  
45     <header>  
46     | <h3>사진 제목</h3>  
47     </header>  
48     <figure>  
49     |   
50     </figure>  
51     <p>사진 설명 : Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ullam, numquam.  
52     Neque mollitia esse blanditiis facere.</p>  
53 </div>  
54  
55 <div class="card">  
56     <header>  
57     | <h3>사진 제목</h3>  
58     </header>
```

```
59     <figure>
60     |   
61     </figure>
62     <p>사진 설명 : Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ullam, numquam.
63     |   Neque mollitia esse blanditiis facere.</p>
64   </div>
65
66   <div class="card">
67     <header>
68     |   <h3>사진 제목</h3>
69     </header>
70     <figure>
71     |   
72     </figure>
73     <p>사진 설명 : Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ullam, numquam.
74     |   Neque mollitia esse blanditiis facere.</p>
75   </div>
76
77   <div class="card">
78     <header>
79     |   <h3>사진 제목</h3>
80     </header>
81     <figure>
82     |   
83     </figure>
84     <p>사진 설명 : Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ullam, numquam.
85     |   Neque mollitia esse blanditiis facere.</p>
86   </div>
87
```

```
88     <div class="card">
89         <header>
90             <h3>사진 제목</h3>
91         </header>
92         <figure>
93             
94         </figure>
95         <p>사진 설명 : Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ullam, numquam.
96             Neque mollitia esse blanditiis facere.</p>
97     </div>
98 </div>
99 </body>
100 </html>
```

사진 제목



사진 설명 : Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ullam, numquam. Neque mollitia esse blanditiis facere.

사진 제목



사진 설명 : Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ullam, numquam. Neque mollitia esse blanditiis facere.

사진 제목



사진 설명 : Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ullam, numquam. Neque mollitia esse blanditiis facere.

사진 제목



사진 설명 : Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ullam, numquam. Neque mollitia esse blanditiis facere.

사진 제목



사진 설명 : Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ullam, numquam. Neque mollitia esse blanditiis facere.

사진 제목



사진 설명 : Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ullam, numquam. Neque mollitia esse blanditiis facere.

사진 제목



사진 설명 : Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ullam, numquam. Neque mollitia esse blanditiis facere.

사진 제목



사진 설명 : Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ullam, numquam. Neque mollitia esse blanditiis facere.

사진 제목



사진 설명 : Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ullam, numquam. Neque mollitia esse blanditiis facere.

사진 제목



사진 설명 : Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ullam, numquam. Neque mollitia esse blanditiis facere.

수고하셨습니다