10 반응형 웹과 미디어쿼리

- 반응형 웹
- 반응형 요소
- 미디어쿼리

❖ 반응형 웹 디자인(responsive web design)이란?

- **화면의 크기에 반응**해 화면 요소들을 자동으로 바꾸어 사이트를 구현하는 것

- **뷰포트(viewport)** : 스마트폰 화면에서 실제 내용이 표시되는 영역
  - 모바일 기기가 아닌 '**웹 브라우저 창 너비**'에 반응하는 것이다.

- 장점
  - 사이트 하나를 만들어 모든 기기에 사용 가능하므로 효율적이다.
  - 사이트 유지・관리가 쉽다.

❖ 뷰 포트 지정하기

기본형 : <meta name="viewport" content="속성1, 속성2 ...">

- head 영역에 지정
- content 속성에서 뷰 포트 속성을 지정

속성	설명	값	기본값
width	뷰포트 너비	device-width, 크기	브라우저 기본값
height	뷰포트 높이	device-height, 크기	브라우저 기본값
user-scalable	확대/축소 가능여부	yes/no	yes
initial-scale	초기 확대/축소 값	1-10	1
minimum-scale	최소 확대/축소 값	0-10	0.25
maximum-scale	최대 확대/축소 값	0-10	1.6

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

❖ 뷰 포트 단위

속성	설명	크기	예시
vw (viewport width)	뷰포트 너비	1vw = 너비의 1%	너비 = 1000px 인 경우 1vw = 10px
vh (viewport height)	뷰포트 높이	1vh = 높이의 1%	높이 = 800px 인 경우 1vh = 8px
vmin (viewport minimum)	뷰포트 최소값	너비와 높이 중 작은 값의 1%	vmin = 8px
vmax (viewport maximum)	뷰포트 최대값	너비와 높이 중 큰 값의 1%	vmax = 10px

```
<style>
    h1{ font-size : 3vw }    →  3 * 10px = 30px
</style>
```

❖ 반응형 요소 (p320)

요소	설명		
가변 글꼴	em	부모요소 폰트의대문자 M의 너비=1em(16px) 글자크기(em)=글자크기(px)/16px	
	rem	root의 기본 크기를 기준으로 지정	
가변 이미지	max-width=100%; height: auto; - 부모요소 만큼만 확대/축소		
object-fit	콘텐츠의 가로세로 비율을 유지하면서 해상도에 맞게 크기 조절		
	fill	요소 전체 영역을 채움(비율 무시), 기본값	
	contain	요소 전체 영역을 맞춤(비율 유지)	
	cover	요소 전체 영역을 채움(비율 유지)	
	none	원래 크기 유지	
	scale-down	none과 contain 중 작은 값 선택	
가변 비디오	max-width=100%;		

❖ 반응형 요소 (p320)

em 단위

- 부모 요소에서 지정한 글꼴의 대문자 M 너비를 1em으로 놓고 상대적인 크기를 계산함
- 부모 요소에서 글자 크기를 지정하지 않으면 body 요소의 기본 크기 사용

```
<style>
  p { font-size: 1em; }
 .content { font-size: 1.5em; }
</style>
……
<h1>레드향</h1>
<p>껍질에 붉은 빛이 돌아 레드향이라 불린다.</p>
<div class="content">
  <p>레드향은 한라봉과 귤을 교배한 것으로</p>
  <p>일반 귤보다 2~3배 크고, 과육이 붉고 통통하다.</p>
</div>
```

레드향

껍질에 붉은 빛이 돌아 레드향이라 불린다.

레드향은 한라봉과 귤을 교배한 것으로

일반 귤보다 2~3배 크고, 과육이 붉고 통통하다.

16px

24px

부모 요소가 다르기 때문에 p 요소의 글자 크기가 달라짐

❖ 반응형 요소 (p322)

rem 단위

- root에서 지정한 크기를 기준으로 상대적인 크기를 계산함

```
<style>
  html { font-size: 16px; }
   p { font-size: 1rem; }
  .content { font-size: 1.5rem; }
</style>
……
<h1>레드향</h1>
<p>껍질에 붉은 빛이 돌아 레드향이라 불린다.</p>
<div class="content">
  <p>레드향은 한라봉과 귤을 교배한 것으로</p>
  <p>일반 귤보다 2~3배 크고, 과육이 붉고 통통하다.</p>
</div>
```

16px

레드향

껍질에 붉은 빛이 돌아 레드향이라 불린다.

레드향은 한라봉과 귤을 교배한 것으로

일반 귤보다 2~3배 크고, 과육이 붉고 통통하다.

em 단위가 부모 요소를 기준으로 한다는 점만 기억한다면 em 단위나 rem 단위는 상황에 따라 어떤 것을 사용해도 무방함

❖ 반응형 요소 (p323)

width 속성과 max-width 속성

- width: 100%로 지정하면 부모 요소의 너비가 커질 때 화질이 떨어질 수 있음.
- max-width: 100%로 지정하면 부모 요소 너비가 커져서 이미지가 커지더라도 이미지 원본 크기만큼만 커짐

```
<style>
  ……
  .top {
      max-width: 100%;
      height: auto;
  }
</style>
```

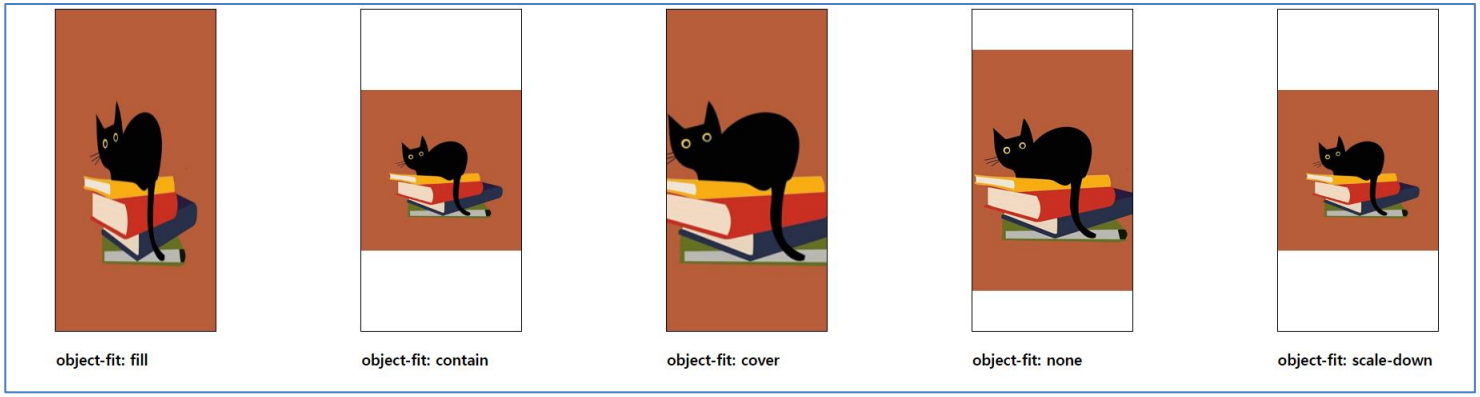

```
<p>브라우저 창의 크기를 조절해보세요.</p>
<img src="images/kitten.jpg" alt="베개 뒤에 숨어 있는 고양이" class="top">
```

❖ 반응형 요소 (p324)

object-fit 속성

- 이미지나 비디오 등의 가로, 세로 비율을 유지하면서 크기를 조절할 수 있다

```
<style>
  ……
  .fill { object-fit: fill; }  /* 요소에 꽉 차게 채움 */
  .contain { object-fit: contain; }  /* 비율 유지. 요소 크기에 맞춤 */
  .cover { object-fit: cover; }  /* 비율 유지. 요소에 꽉 차게 채움 */
  .none { object-fit: none; }  /* 이미지 원래 크기 유지 */
  .scale-down { object-fit: scale-down; }  /* none과 contain 중 작은 것 */
</style>
```



❖ 미디어 쿼리

- 어떤 미디어를 사용하느냐에 따라 화면 크기에 맞는 CSS가 적용되도록 하는 것(방법)

기본형 : @media[only|not] 미디어 유형 [and 조건]*[and 조건]

- <style>…</style> 영역에 지정
- 미디어 유형 : all, print, screen, tv, aural(음성 합성 장치), braille(점자표시장치), handheld, projection, tty, embossed(점자 프린터)

```
<style>
  @media all (min-width:600px) and (max-width:959px){
          적용할 css 속성
   }
</style>
```

❖ 미디어 쿼리

구분	속성
웹 문서	width, height, min[max]-width, min[max]-height
단말기	device-width, device-height, min[max]-device-width, min[max]-device-height
화면 회전	orientation:portrait[landscape] 세로 방향[가로방향]
중단점	서로 다른 css를 적용할 화면 크기(break point) 표준은 없지만 가장 일반적으로 사용할 만한 기기 파악 후 사용

css프레임워크	아주 작은 화면	작은 화면	중간 화면	큰 화면	아주 큰 화면
부트스트랩	576px 미만	576px 이상	768px 이상	992px 이상	1200px 이상
마젠토	640px 미만	640px 이상	768px 이상	1024px 이상	1440px 이상

❖ 미디어 쿼리 적용하기

외부 파일 연결하기

- **<link rel="stylesheet" media="미디어파일 조건" href="css 파일경로">**

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

  <link rel="stylesheet" media="screen and (max-width:768px)"

  href="css/tablet.css">

- **@import url(css파일 경로) 미디어쿼리 조건**

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

  and (max-width:768px);

- @import구문보다 <link> 태그가 안정적이고 빠르므로 link 태그 사용 권장

❖ 미디어 쿼리 적용하기

웹 문서에서 직접 정의

- **<style media="미디어 쿼리 조건">**
  스타일 규칙들
  </style>

```
<style media="screen and (max-width:768px)">
        body{background: orange;}
</style>
```

- **<style>**
  @media 미디어 쿼리 조건{
  스타일 규칙들
  }
  </style>

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

❖ 미디어 쿼리 사용 문서 만들기

media.html

① 레이아웃 구상하기

- 모바일 화면
- 태블릿 화면
- 데스크톱 화면

② 미디어 쿼리 중단점 결정하기

③ 태그로 구성

④ 모바일 화면 css 작성

⑤ 태블릿 화면 css 작성

- 미디어 중단점(768px 이상 ~ 1719px 이하)

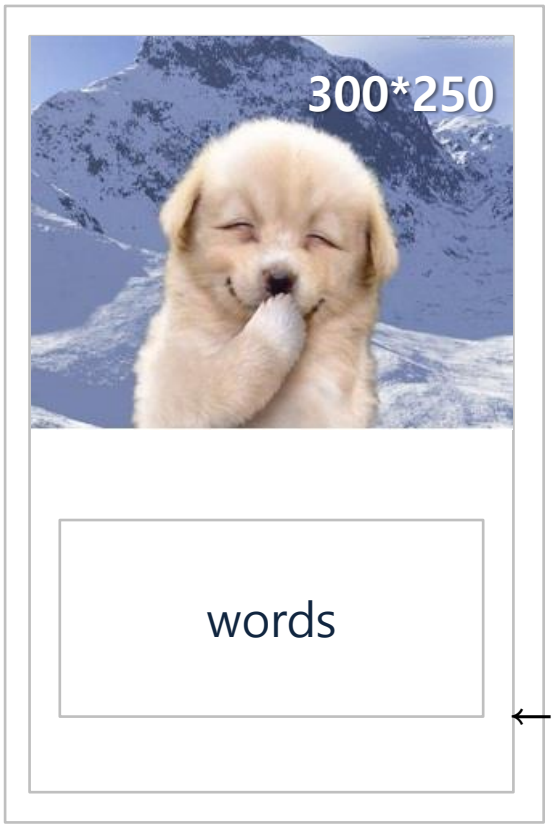
⑥ 데스크톱 화면 css 작성

- 미디어 중단점(1720px 이상)

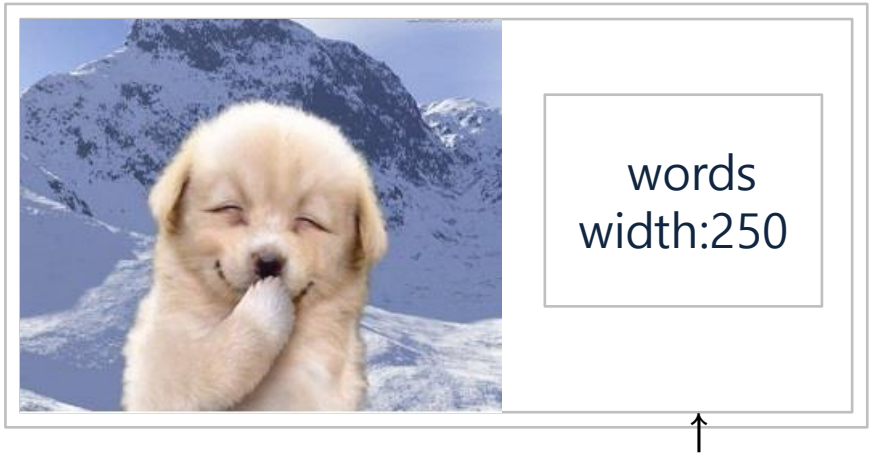
❖ 미디어 쿼리 사용 문서 만들기

카드 설계

#container : 320px



#container : 570px



❖ 미디어 쿼리 이용 사이트 구성하기

```
<body>
    <div id="container">
        <div class="card">
            <img src="img/1.jpg">
            <div class="words">
                <h2>일 분 전만큼 먼 시간은 없다.</h2>
                <h3>- Jim Bishop</h3>
            </div>
        </div>
        <div class="card">
            <img src="img/2.jpg">
            <div class="words">
                <h2>웃음은 마음의 조깅이다.</h2>
                <h3>- Norman Cousins</h3>
            </div>
        </div>
        <div class="card">
            <img src="img/3.jpg">
            <div class="words">
                <h2>낡은 옷은 그냥 입고 새 책을 사라.</h2>
                <h3>- Austin Phelps</h3>
            </div>
        </div>
    </div>
</body>
```

❖ 미디어 쿼리 이용 사이트 구성하기

```
<style>
    *{
        margin: 0;
        padding: 0;
        box-sizing: border-box;
    }
    body {
        background:rgb(9, 100, 160);
    }
    #container{
        width: 320px;
        margin: 50px auto;
    }
    .card{
        width: 300px;
        height: 500px;
        background-color: #fff;
        margin-bottom: 20px;
        position: relative;
    }
```

```
    .words{
        width: 300px;
        position: absolute;
        top:300px;
        text-align: center;
        padding: 20px;
    }
```

❖ 미디어 쿼리 이용 사이트 구성하기

```
@media screen and (min-width : 768px){
        #container{
            width: 570px;
            margin: 50px auto;
        }
        .card{
            width: 550px;
            height: 250px;
            background-color: #fff;
            margin: 20px 10px;
            position: relative;
        }
        .words{
            width: 250px;
            position: absolute;
            left:300px;
            top:50px;
            text-align: center;
        }
    }
```

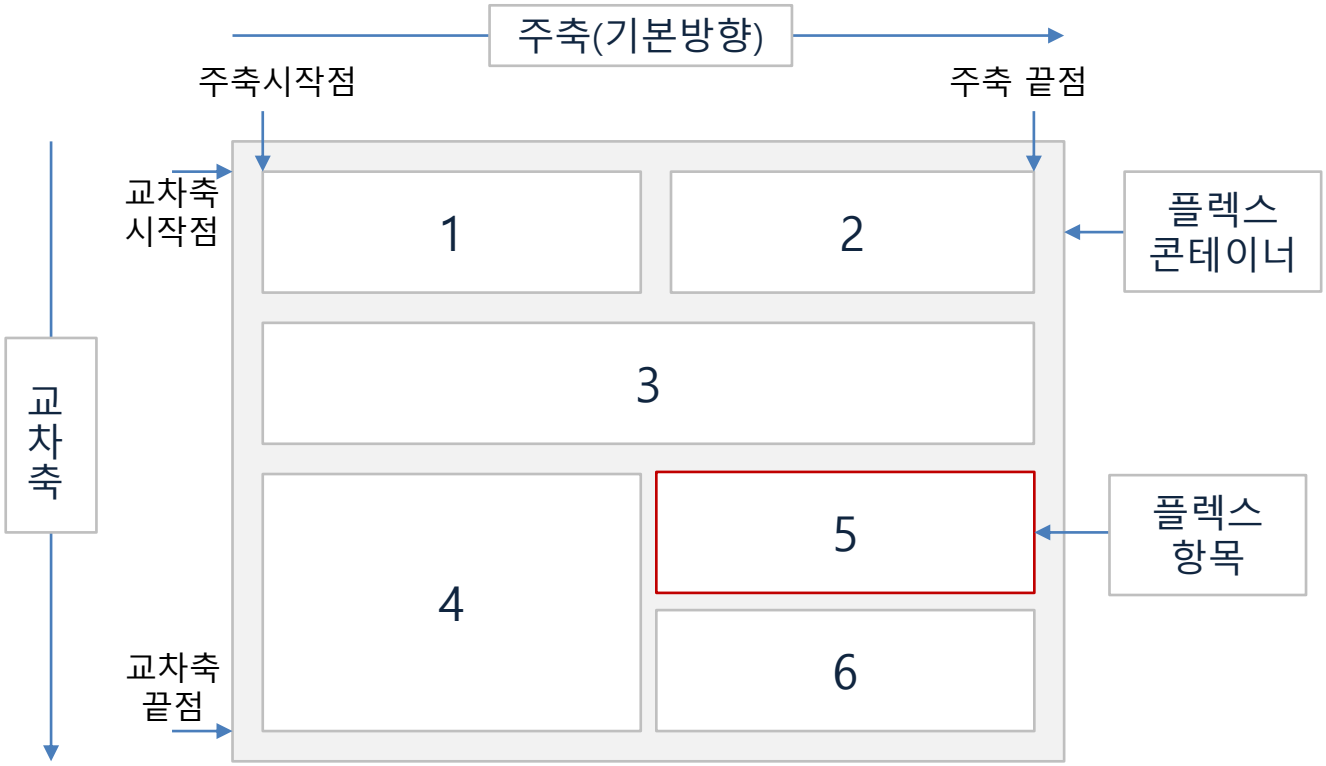
❖ 미디어 쿼리 이용 사이트 구성하기

```
    @media screen and (min-width : 1720px){
        #container{
            width: 1710px;
            margin: 50px auto;
        }
        .card{
            float: left;
        }
    }


</style>
```

❖ 플렉스 박스 레이아웃(flex box layout)_p342

- 그리드 레이아웃을 기본으로 플렉스 박스를 원하는 위치에 배치하는 것
- 플렉스 박스를 이용하면 여유 공간에 따라 너비나 높이, 위치를 자유롭게 변형할 수 있어 편리하게 사용할 수 있다



❖ 플렉스 박스 레이아웃

속성	설명	기본형
display	플렉스 컨테이너 지정	display: flex | inline-flex
flex-direction	주축 방향 지정	flex-direction: row | row-reverse | column| column-reverse
flex-wrap	항목 한 줄/여러 줄 배치	flex-wrap: no-wrap | wrap | wrap-reverse
flex-flow	방향과 배치 함께 지정	flex-flow: <방향> <줄 배치>
order	항목 배치 순서 바꾸기	order: 0 | 숫자 (0 : 입력 순서, 숫자 순서로 배치)
flex	플렉스 항목 크기 조절	flex: [<flex-grow> <flex-shrink> <flex-basis>] | auto(항목 너비값) | initial
flex : 2 2 0; flex: [<flex-grow: 늘릴 비율> <flex-shrink: 줄일 비율> <flex-basis: 기본값>]		

❖ 플렉스 박스 레이아웃

속성	설명	기본형
justify-content	주축 기준 배치 방법	justify-content: flex-start | flex-end | center | space-between | space-around
align-items	교차 축 배치 방법	align-items: stretch | flex-start | flex-end | center | base-line(글자 기준선)
align-self	교차 축 배치 방법 (특정 플렉스 항목 배치 가능)	align-self: auto(부모 속성 상속) | stretch | flex-start | flex-end | center | base-line
align-content	여러 줄일 때 배치 방법 (교차축)	align-content: flex-start | flex-end | center | space-between | space-around

❖ 플렉스 박스 이용 사이트 구성하기

flex.html

① 플렉스 박스 레이아웃 구상하기

- 모바일 화면
- 태블릿 화면
- 데스크톱 화면

② 태그로 구성/기본 css작성

③ 미디어 쿼리 작성하기

④ 플렉스 컨테이너 지정하기

⑤ 플렉스 항목 너비 지정하기

⑥ 브라우저 확인하기

❖ 플렉스 박스 이용 사이트 구성하기

```
<body>
    <div id="container">
        <header>
            <h1>솔로의 식탁</h1>
        </header>
        <section id="menus">
            <div id="menu1">
                <h2>밥/죽</h2>
            </div>
            <div id="menu2">
                <h2>국/찌개</h2>
            </div>
            <div id="menu3">
                <h2>반찬</h2>
            </div>
            <div id="menu4">
                <h2>일품요리</h2>
            </div>
            <div id="menu5">
                <h2>음료/칵테일</h2>
            </div>
        </section>
        <footer>
            <p>솔로의 식탁</p>
        </footer>
    </div>
</body>
```

❖ 플렉스 박스 이용 사이트 구성하기

```
* {
    margin:0;
    padding:0;
    box-sizing:border-box;
}
#container {
    width:100%;
}
header {
    width:100%;
}
header h1 {
    font-size:3em;
    text-align: center;
}
#menus {
    width:100%;
}
```

```
#menus > div {
    position:relative;
    height:400px;
    border:1px solid black;
    margin-bottom:15px;
}
#menu1, #menu2, #menu3,
#menu4, #menu5 {
    width:100%;
}
#menus h2 {
    position:absolute;
    padding:5px;
    font-size:2em;
    color:white;
    text-shadow:3px 3px 5px
                        black;
    right:3%;
    bottom:10px;
}
```

❖ 플렉스 박스 이용 사이트 구성하기

```
#menu1 {
    background:url("img/dish1.jpg") no-repeat center;
    background-size:cover;
}
#menu2 {
    background:url("img/dish2.jpg") no-repeat center;
    background-size:cover;
}
#menu3 {
    background:url("img/dish3.jpg") no-repeat center;
    background-size:cover;
}
#menu4 {
    background:url("img/dish4.jpg") no-repeat center;
    background-size:cover;
}
#menu5 {
    background:url("img/dish5.jpg") no-repeat center;
    background-size:cover;
}
```

❖ 플렉스 박스 이용 사이트 구성하기

```
footer {
        width:100%;
        background:#373737;
        height:200px;
    }
    footer p {
        font-size:1.2em;
        color:#eee;
        text-align:center;
        line-height:200px;
    }
```

❖ 플렉스 박스 이용 사이트 구성하기

```
@media all and (min-width:768px) {
    #menus {
        display: flex;
        flex-wrap: wrap;
        justify-content: space-between;
    }
    #menu1, #menu2, #menu3, #menu4 {
        width: 49%;
    }
    #menu5 {
        width: 100%;
    }
}
```

```
@media all and (min-width:992px) {
        #menu1, #menu2, #menu3,
        #menu4, #menu5 {
            width:33%;
        }
        #menu5 {
            margin-left: 0.5%;
            flex: 2 2 0;
        }
    }
```

❖ CSS 그리드 레이아웃(css grid layout) p373

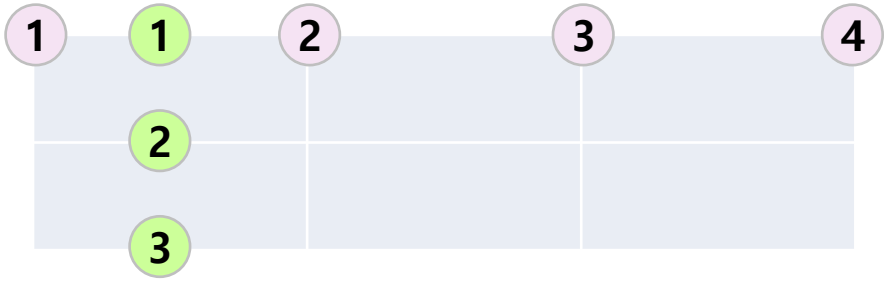
- 그리드 레이아웃은 가로방향을 가리키는 줄(row), 세로 방향을 가리키는 칼럼(column)으로 화면 구성

❖ 그리드 레이아웃 속성과 함수

속성	설명	기본형
display	그리드 컨테이너 지정	display: grid| inline-grid;
grid-template-columns	칼럼 크기와 개수 지정	: 200px 400px 200px; : 1fr 2fr 1fr; fr(fraction) : 상대 크기 지정
grid-template-rows grid-auto-rows	줄 높이 지정	grid-template-rows : 100px;
minmax()	최소값과 최대값 지정	grid-template-rows : minmax(100px, auto);
repeat()	값 반복 함수	grid-template-columns : repeat(auto-fit[fill], 200px);
[column/row]gap	칼럼과 줄 사이 간격 지정	row-gap : 20px; gap : 30px;

❖ 그리드 라인을 이용한 배치

속성	설명	기본형
grid-column-start	칼럼 시작 번호	grid-column-start : 1;
grid-column-end	칼럼 끝 번호	grid-column-end : 4;
grid-column	칼럼 시작/끝	grid-column : 1/4;
grid-row-start	줄 시작 번호	grid-row-start : 1;
grid-row-end	줄 끝 번호	grid-row-end : 4;
grid-row	줄 시작/끝	grid-row : 1/4;



❖ 템플릿 영역 만들어 배치

속성	설명	기본형
grid-area	템플릿 이름 지정	grid-area : box1;
grid-template-area	템플릿 그리드 만듬	grid-template-areas : “box1 box1 box1” “box2 box3 box3” “box2 . box4” . -> 빈 영역 표시

❖ 그리드 레이아웃 갤러리

```
<body>
  <div id="wrapper">
    <div class="card">
      <header>
        <h3>사진 제목</h3>
      </header>
      <figure>
        <img src="img/cup-1.jpg">
        <figcaption>사진 설명 : Lorem ipsum dolor sit amet consectetur
          adipisicing elit. Cumque nemo odit,
            facilis distinctio!</figcaption>
      </figure>
    </div>
```

❖ 그리드 레이아웃 갤러리

```
<div class="card">
      <header>
        <h3>사진 제목</h3>
      </header>
      <figure>
        <img src="img/cup-2.jpg">
      </figure>
      <p>사진 설명 : Lorem ipsum dolor sit amet consectetur
          adipisicing elit. Cumque nemo odit,
            facilis distinctio!</p>
    </div>
       ...
  </div>
</body>
```

❖ 그리드 레이아웃 갤러리

```
* {
        box-sizing: border-box;
    }
  body {
    background-color:#eee;
    font-size:16px;
  }
  #wrapper{
    display:grid;
    grid-template-columns:repeat(auto-fit, minmax(320px, 1fr));
    gap:1rem;
  }
```

❖ 그리드 레이아웃 갤러리

```
.card {
  background-color:#fff;
  box-shadow:0px 1px 5px #222;
}
.card header {
  font-size:1.5rem;
  padding:0.5rem;
}
```

```
.card > p {
  padding:0.5rem;
  line-height:1.6em;
}
.card img {
  max-width:100%;
}
```