

5강. 반응형 웹과 미디어 쿼리



모바일 기기와 웹 디자인

반응형 웹 디자인 (Responsive Web Design)

- 웹 사이트의 내용을 그대로 유지하면서 다양한 화면 크기에 맞게 웹사이트를 표시하는 방법
- 화면 크기에 반응해 화면 요소들을 자동으로 바꾸어 사이트를 구현하는 것

반응형 웹디자인의 장단점

장점

- 모든 스마트 기기에서 접속 가능
- 가로 모드에 맞춘 레이아웃 변경 가능
- 사이트 유지, 관리 용이

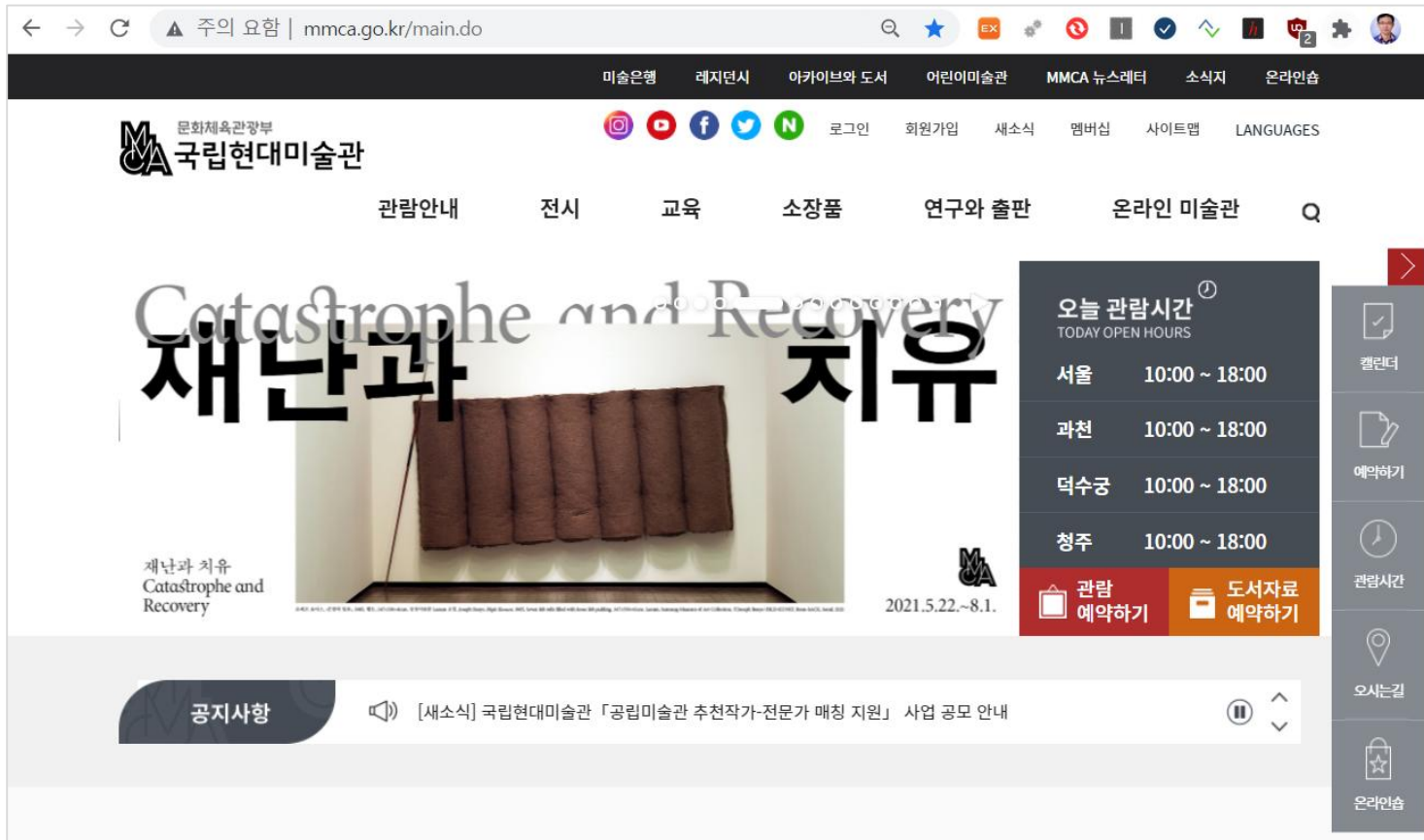
단점

- 반응형 웹 기술이 최신 웹 표준인 CSS3의 일부 -> 최신 브라우저만 지원된다.



반응형 웹 디자인

국립 현대 미술관



PC 기준



반응형 웹 디자인

국립 현대 미술관



태블릿 기준

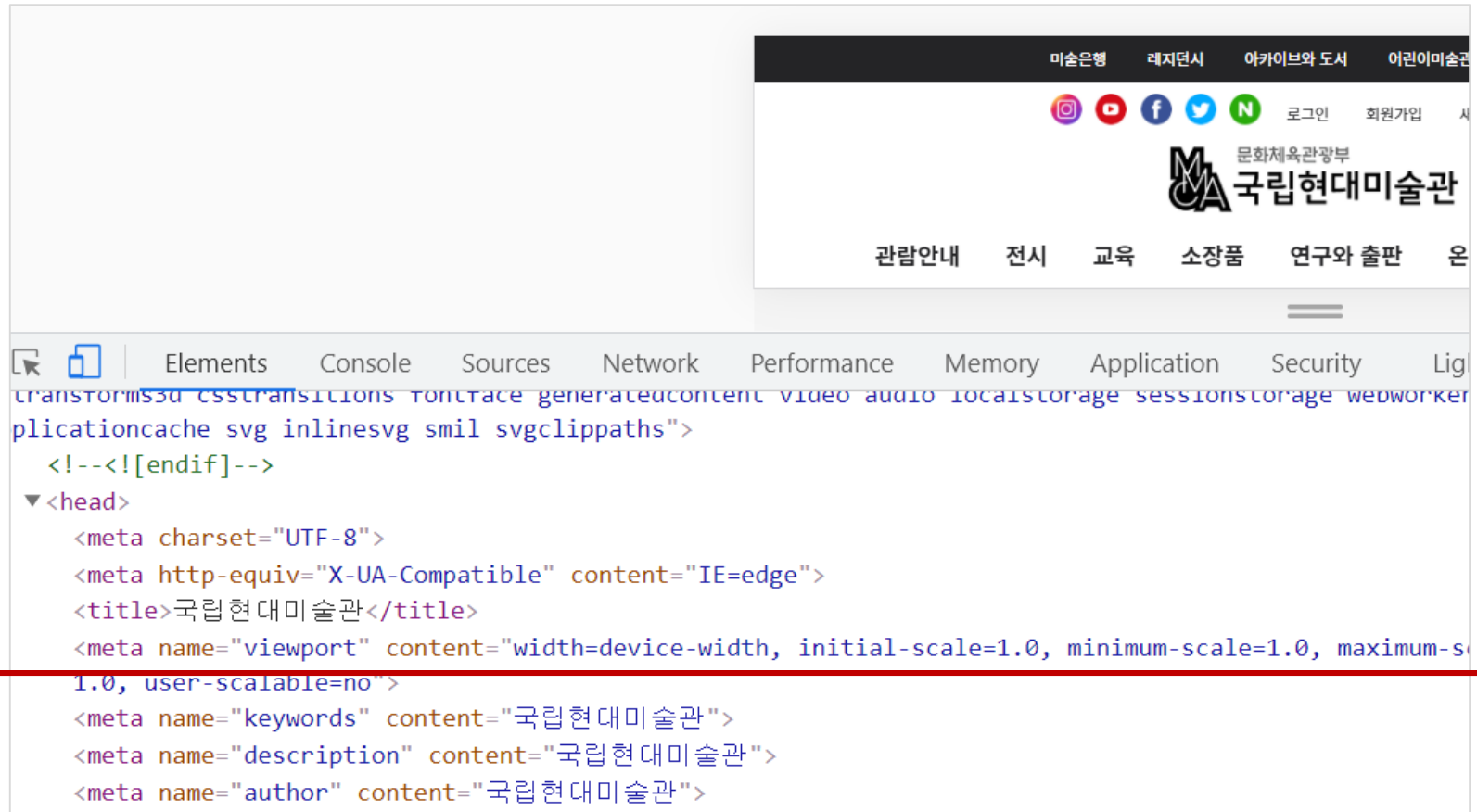


스마트폰 기준



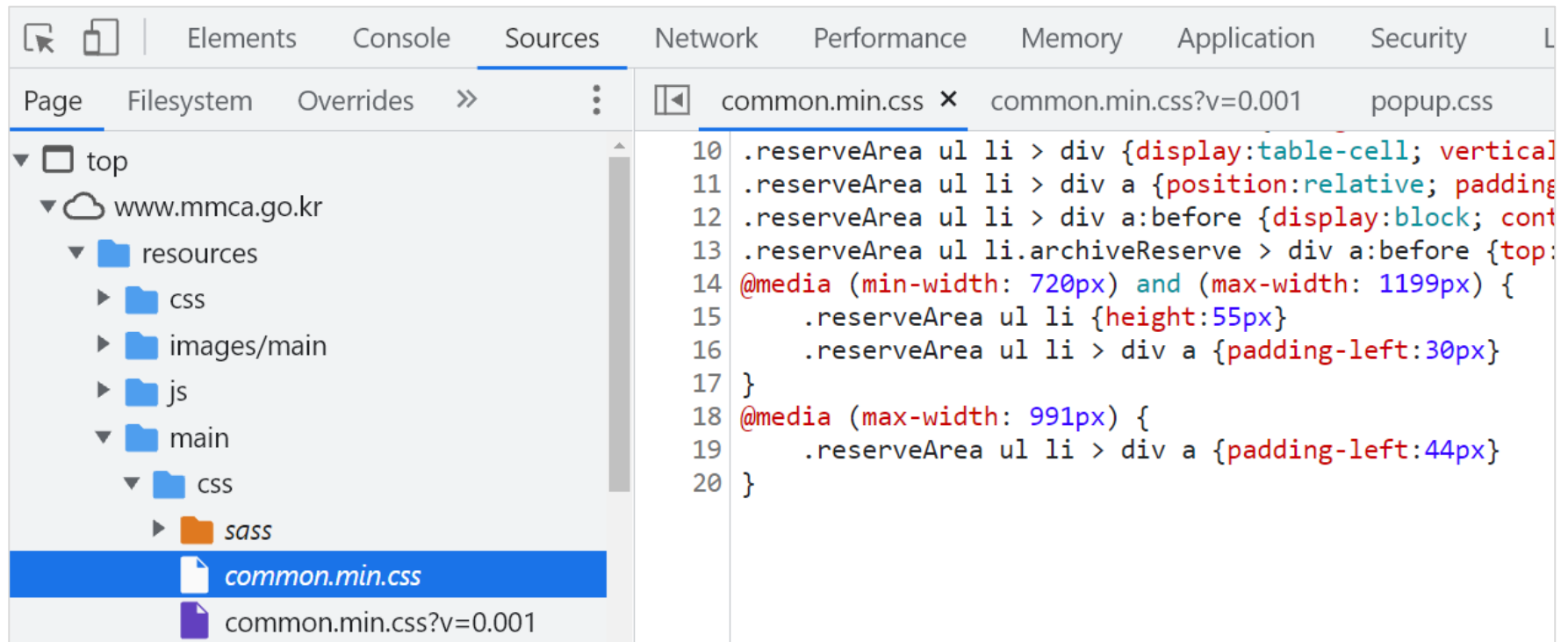
반응형 웹 디자인

국립 현대 미술관



반응형 웹 디자인

Css -> 미디어 쿼리 사용



뷰포트(Viewport) 지정하기

뷰포트(viewport)

- 실제 내용이 표시되는 영역
- PC화면과 모바일 화면의 픽셀 표시 방법이 다르기 때문에 모바일 화면에서 의도한 대로 표시되지 않음
- 뷰포트를 지정하면 기기 화면에 맞춰 확대/축소해서 내용 표시

뷰포트 지정하기

- <head>태그 안에서 <meta>태그를 이용해 뷰포트 지정

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

의미 : 뷰포트의 너비를 스마트폰 화면 너비에 맞추고 초기화면 배율을 1로 지정



뷰포트(Viewport) 지정하기

뷰포트의 속성

종류	설명	사용 가능한 값
width	뷰포트 너비	device-width 또는 크기
height	뷰포트 높이	device-height 또는 크기
initial-scale	초기 확대.축소 값	1~10

뷰포트의 단위

CSS에서 크기를 지정할 때 주로 px, % 단위는 고정 폰트이고, 브라우저의 비율에 따라 글자 크기가 변경되는 가변폰트는 vw, vh이다.

- vw(viewport width) : 1vw는 뷰포트 너비의 1%와 같다.(1200px이면 5%-60px)
웹 브라우저 너비는 100을 기준으로함. $(5 \times 1200) \div 100$
- vh(viewport height): 1vh는 뷰포트 높이의 1%와 같다.



뷰포트(Viewport) 지정하기

em, rem 사용 예제

em과 rem 비교

em 단위

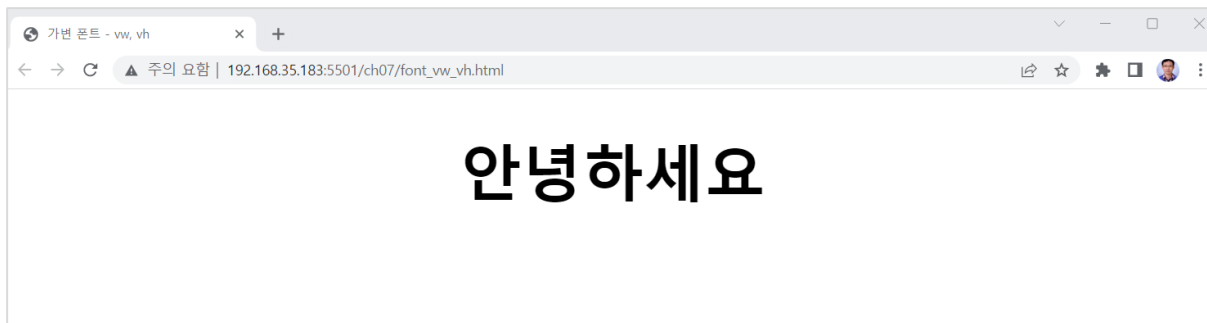
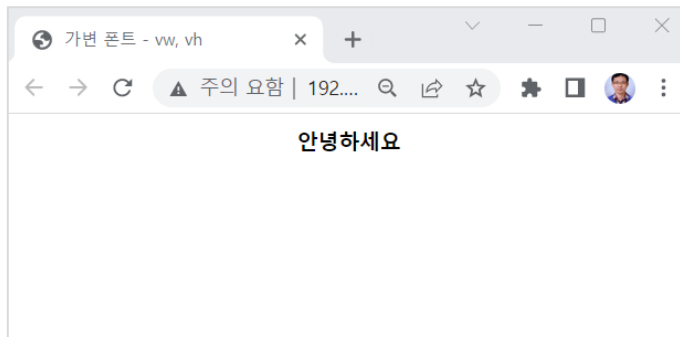
rem 단위

```
<style>
  body{
    font-size: 12px;
  }
  /* em은 부모로부터 상속을 받으나
   rem은 상속 받지 않음 */
  /* 1em = 12px, 1rem = 16px*/
  p.a{font-size: 2em;}
  p.b{font-size: 2rem;}
</style>
</head>
<body>
  <h1>em과 rem 비교</h1>
  <p class="a">em 단위</p>
  <p class="b">rem 단위</p>
</body>
```



뷰포트(Viewport) 지정하기

뷰포트 사용 예제



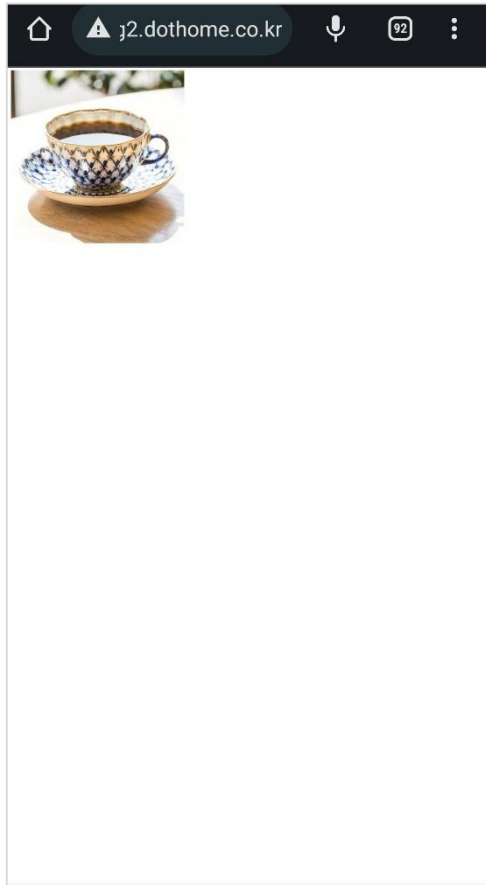
```
<style>
  h1{
    text-align: center;
    /* 5x16px=80px */
    /*font-size: 5em;*/

    /* 브라우저 크기 1200px이면
    | vw,vh 동일하게 적용 (5x1200)/100=60px */
    font-size: 5vw;
    /*font-size: 5vh; /* 브라우저 높이 1200px이면 60px */
  }
</style>
</head>
<body>
  <h1>안녕하세요</h1>
</body>
```

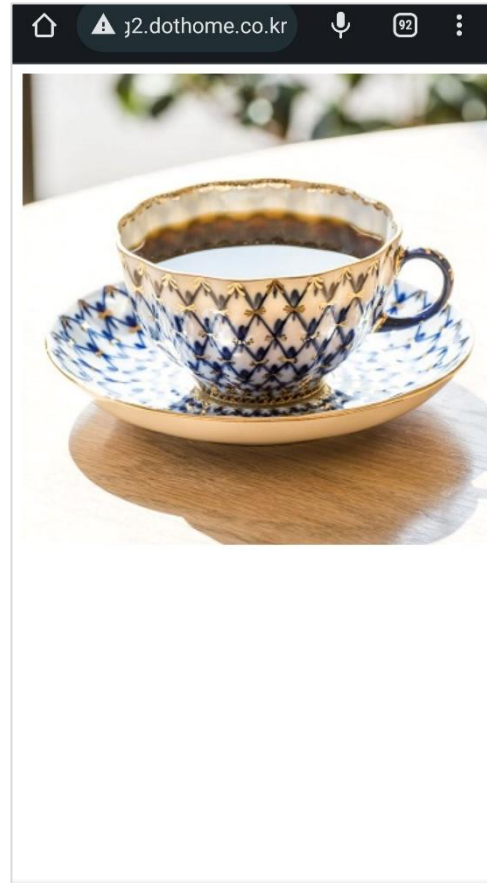


뷰포트(Viewport) 지정하기

뷰포트 사용 안함



뷰포트 사용함



뷰포트(Viewport) 지정하기

```
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>뷰포트 사용 안함</title>
</head>
<body>
  
</body>
```

viewport1.html

```
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>뷰포트</title>
</head>
<body>
  
</body>
```

viewport2.html



호스팅 서버 준비하기

웹호스팅 서버 준비하기

- HTML로 만든 웹문서를 서버 컴퓨터에 옮겨야 한다.
- 개인은 웹서버를 마련하기 쉽지 않아서 서버의 일부 공간을 매달 또는 몇 년마다 일정금액을 내고 사용하는 서비스를 이용한다.
- 웹호스팅 업체로는 대표적으로 cafe24, 고도몰, 가비아 등이 있다.
- 무료 호스팅 서비스 – 닷홈(dothome.co.kr)

웹호스팅 서버의 종류

- 윈도우 서버 호스팅 – ASP
- 리눅스 서버 호스팅 – 톰캣 서버(JSP), PHP
- 클라우드 서버 호스팅 – AWS(아마존웹서버),
구글 클라우드, MS Azure 등



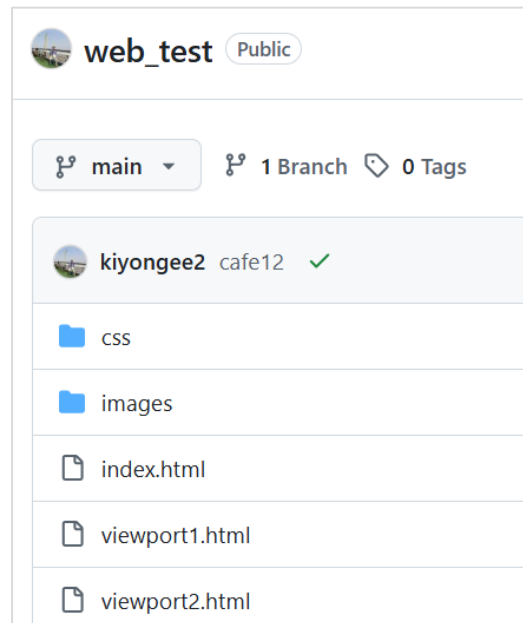
깃허브(Git hub) 호스팅 서버

- 깃허브 무료 호스팅 사용하기

- 도메인 주소

<https://username.github.io/mysite>

- 파일 업로드



깃허브(Git hub) 호스팅 서버

- 깃허브 무료 호스팅 사용하기
 - GitHub Pages 설정
 - 저장소 페이지 → Settings → Pages
 - Branch → main 선택, / (root) 선택
 - 저장 → 잠시 기다리면 배포됨

GitHub Pages

[GitHub Pages](#) is designed to host your personal, organization, or project website.

Your site is live at https://kiyongee2.github.io/web_test/

Last [deployed](#) by [kiyongee2](#) 19 minutes ago


Build and deployment


Source

Deploy from a branch ▾

Branch

Your GitHub Pages site is currently being built from the main branch. [Learn more about your site.](#)

 main ▾


 / (root) ▾

Save




깃허브(Git hub) 호스팅 서버

[카페 소개](#)[이달의 추천](#)[오시는 길](#)



카페 소개




영업 시간 : 오전 9시 ~ 밤 9시
휴무 : 매주 월요일
(월요일이 공휴일인 경우 필요일 영업, 다음날 휴무)


이달의 추천

*** 핸드 드립 아이스 커피 ***

1. 1인분 기준으로 서버에 각열을 5조각(한 조각의 20cc) 넣고 추출을 시작한다.
2. 평상시 보다 원두의 양은 2배 정도(20g)와 추출액은 물을 포함해서 200cc까지 내린다
3. 아이스 잔에 물을 6~7개 섞어서 시원하게 마신다.



오시는 길



서울시 금천구 가산 디지털2로 123
7호선 가산디지털단지역 8번 출구

[카페 소개](#)[개인정보 처리방침](#)[이용 약관](#)[사이트 맵](#)



미디어 쿼리(Media Query)

미디어 쿼리

- 접속하는 장치(미디어)에 따라 **특정한 CSS 스타일을 사용하도록 함**
- 반응형 웹디자인에서 가장 많이 사용하는 방식
(반응형 레이아웃 - 가변그리드 레이아웃, 플렉스박스 레이아웃 방식 등)

미디어 쿼리 구문

```
@ media 미디어 유형 [and 조건] * [and 조건]
```

예) 미디어 유형이 'screen'이면서 최소 너비가 '600px'이고 최대 너비가 '1400px' 정의

```
<style>
```

```
  @media screen and (min-width: 768px) and (max-width: 1400px){
```

```
    ....
```

```
  }
```

```
</style>
```



미디어 쿼리

미디어 유형의 종류

종류	설명
all	모든 미디어 유형에서 사용할 CSS를 정의
tv	음성과 영상이 동시에 출력되는 TV에서 사용할 CSS를 정의
screen	컴퓨터 및 스마트 장치의 스크린에서 사용할 CSS를 정의

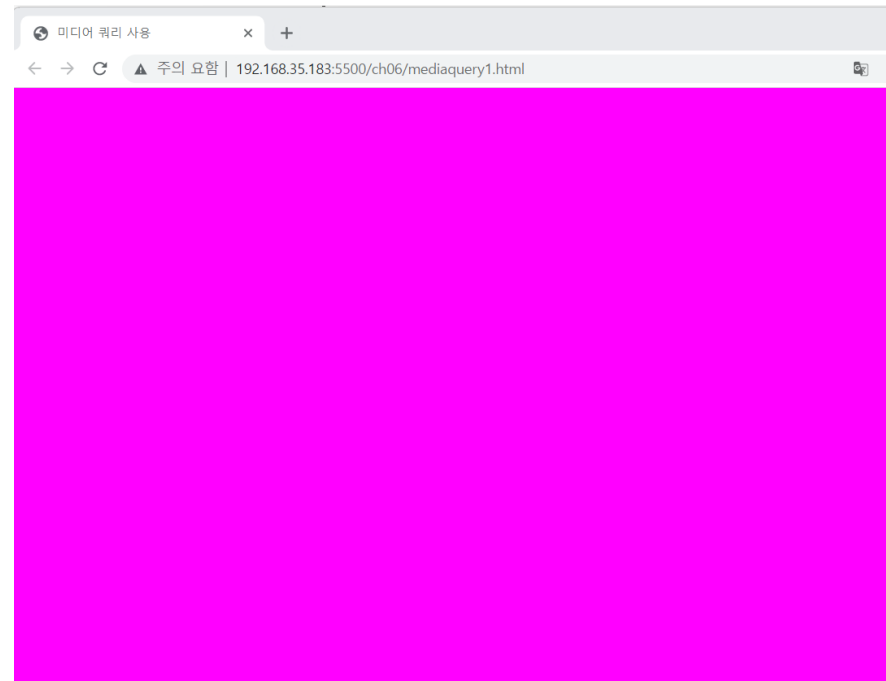
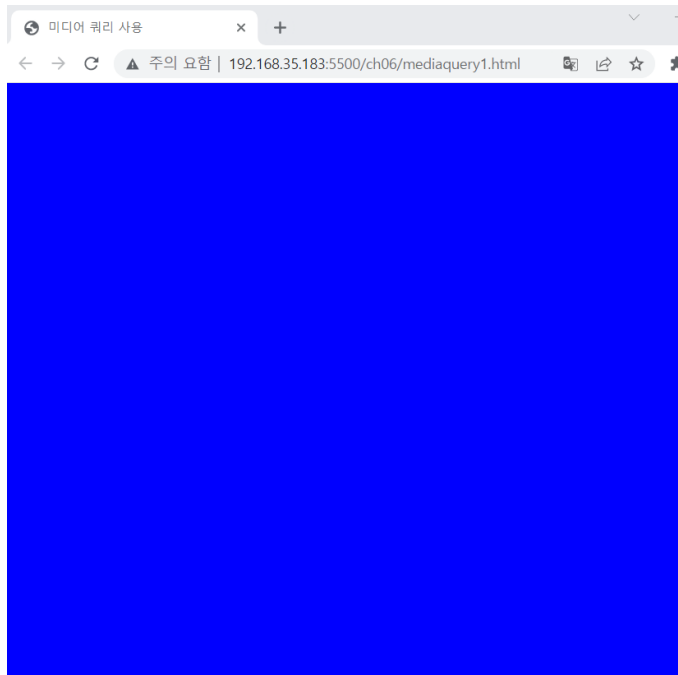
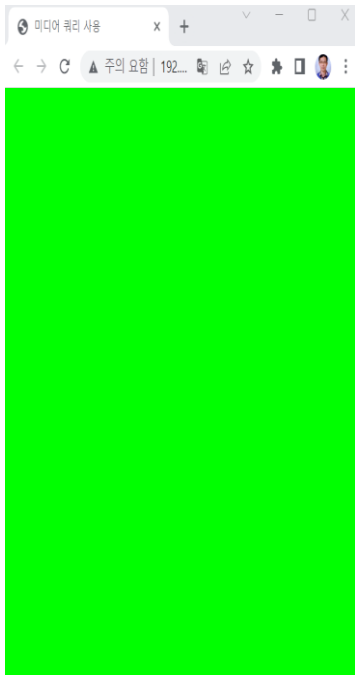
미디어 쿼리 조건

웹페이지의 가로, 세로 값 설정	설명
width, height	웹 페이지의 가로 너비, 세로 높이
min-width, min-height	최소 너비, 최소 높이
max-width, max-height	최대 너비, 최대 높이



미디어 쿼리

미디어 쿼리 사용하기



미디어 쿼리

```
<style>
  @media screen and (max-width: 767px) {
    body{
      background-color: ■ #0f0;
    }
  }
  @media screen and (min-width: 768px) {
    body{
      background-color: ■ #f0f;
    }
  }
  @media screen and (min-width: 960px) {
    body{
      background-color: ■ #ff0;
    }
  }
</style>
```

mediaquery1.html



미디어 쿼리

예제 – 가로 크기가 800px 이상이면 숨기기



미디어 쿼리

예제 - 가로 크기가 800px 이상이면 숨기기

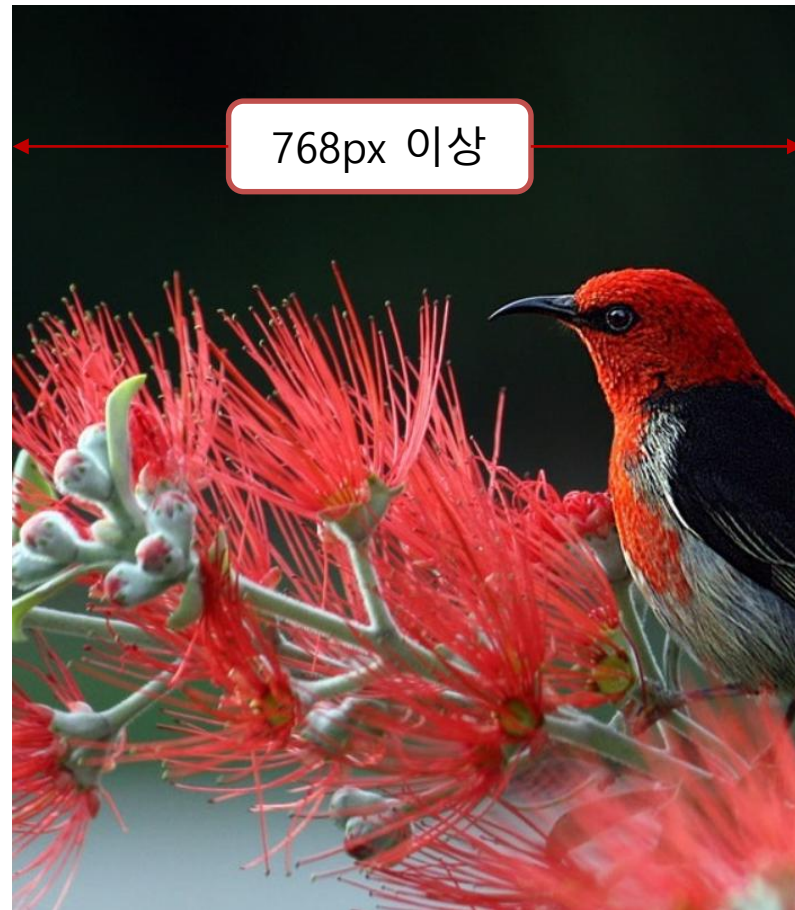
```
<style>
  /* 모바일 기준 */
  div{
    width: 90%;
    margin: 0 auto;
    border: 10px solid #ccc;
    padding: 20px;
    display: block;
  }
  /* 가로 너비가 800px 이상이면 숨기기 */
  @media screen and (min-width: 800px) {
    div{
      display: none;
    }
  }
</style>
```

mq1.html

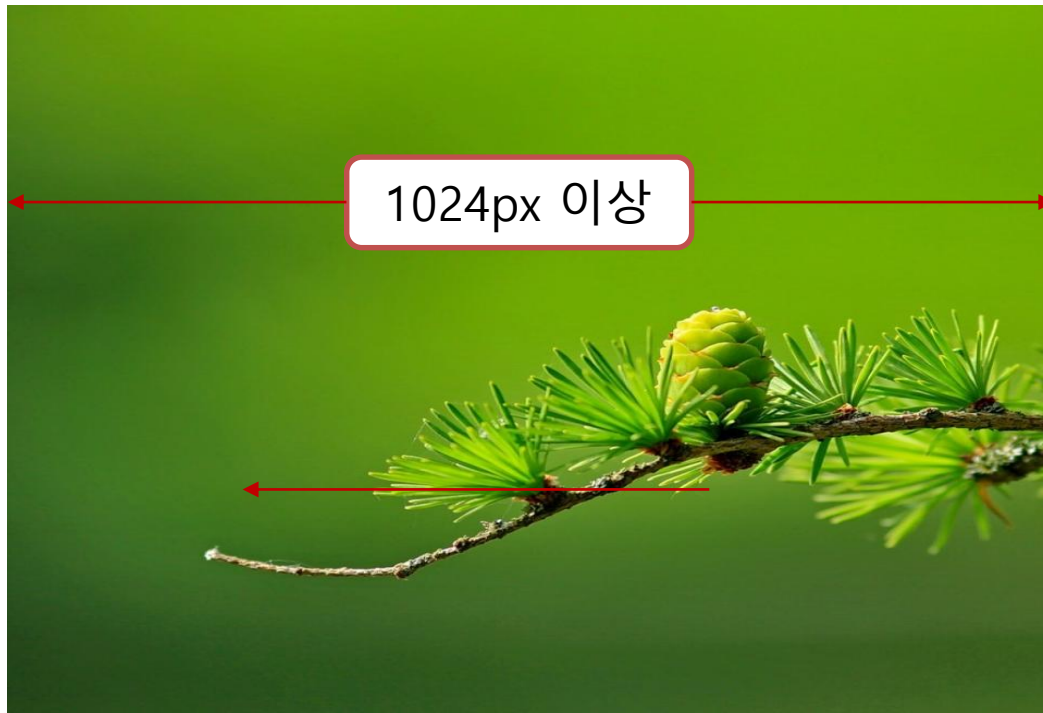


미디어 쿼리

화면 너비에 따라 배경 이미지 바꾸기



미디어 쿼리



미디어 쿼리

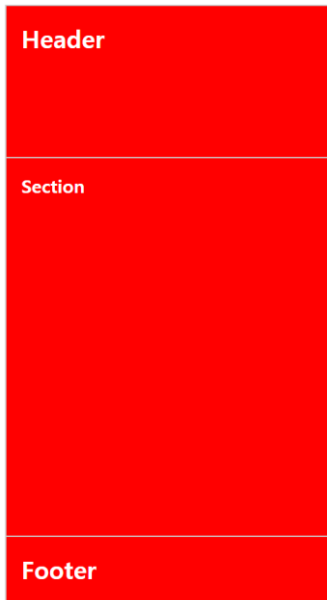
```
<style>
  /* 모바일 기준 */
  body {
    background: url(images/bg3.jpg) no-repeat;
    background-size: cover;
  }
  /* 태블릿 기준 */
  @media screen and (min-width:768px) {
    /* 가로가 768px 이상이면 bg2.jpg 지정 */
    body {
      background: url(images/bg2.jpg) no-repeat;
      background-size: cover;
    }
  }
  /* PC 기준 */
  @media screen and (min-width:1024px) {
    /* 가로가 1024px 이상이면 bg1.jpg 지정 */
    body {
      background: url(images/bg1.jpg) no-repeat;
      background-size: cover;
    }
  }
</style>
```

mq2.html

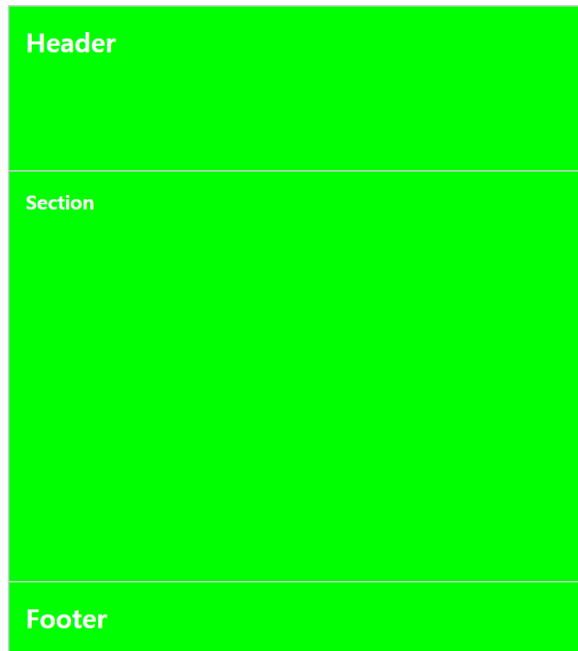


반응형 웹1

반응형 웹 - HTML5 기준 레이아웃



스마트폰



태블릿



PC



반응형 웹 페이지

반응형 웹 - HTML5 기준 레이아웃

```
<body>
  <div id="container">
    <header>
      <h1>Header</h1>
    </header>
    <section>
      <h1>Section</h1>
    </section>
    <footer>
      <h1>Footer</h1>
    </footer>
  </div>
</body>
```

mq-layout.html



반응형 웹 페이지

mq.css

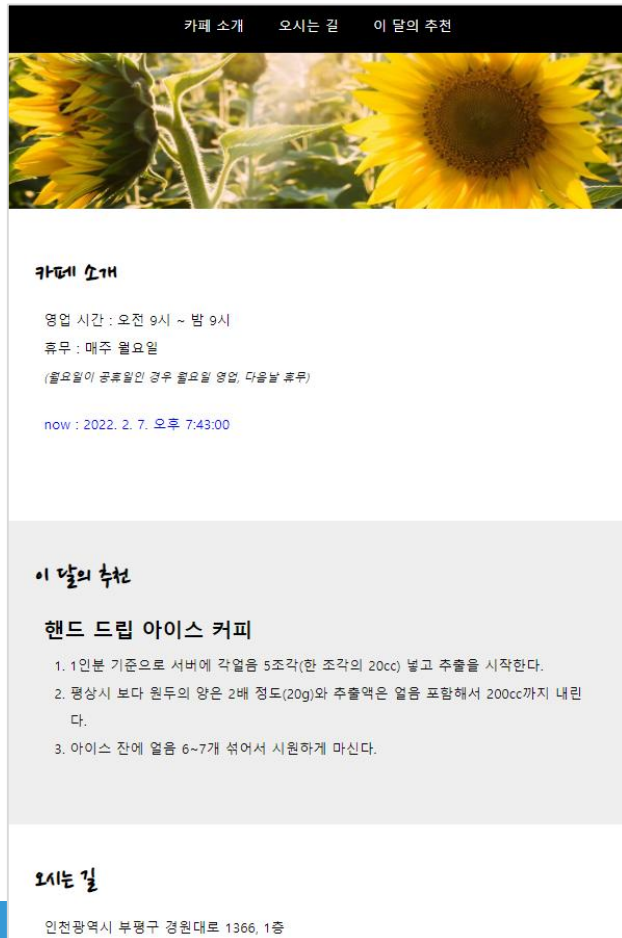
```
/* 모바일 기준*/
*{margin: 0; padding: 0;}
#container{width: 100%; margin:0 auto; background-color: #f00;}
#container h1{font-size: 2em;}
header{height: 100px; border: 1px solid #fff; color: #fff; padding: 10px;}
section{height: 600px; border: 1px solid #fff; color: #fff; padding: 10px;}
footer{height: 100px; border: 1px solid #fff; color: #fff; padding: 10px;}

/* 태블릿 기준 */
@media screen and (min-width: 780px) {
    #container{width: 100%; margin:0 auto; background-color: #0f0;}
}

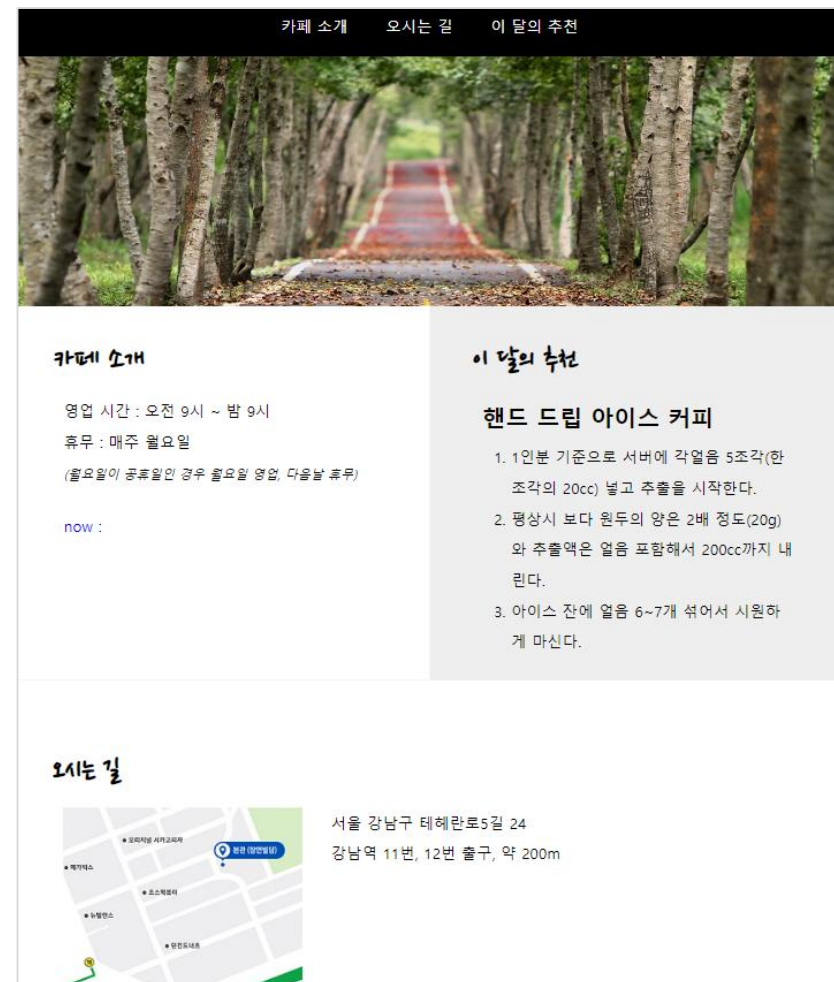
/* PC 기준 */
@media screen and (min-width: 1024px) {
    #container{width: 1000px; margin:0 auto; background-color: #00f;}
}
```

cafe12 사이트

스마트폰 시안



태블릿 시안



cafe12 사이트

PC 시안



cafe12 사이트

index.html

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Cafe 소개</title>
  <link rel="stylesheet" href="css/cafe.css">
</head>
<body>
  <div id="container">
    <header>
      <nav>
        <ul id="main-nav">
          <li><a href="#intro">카페 소개</a></li>
          <li><a href="#map">오시는 길</a></li>
          <li><a href="#choice">이 달의 추천</a></li>
        </ul>
      </nav>
      <div id="main-pic">
        
      </div>
    </header>
    <section id="intro">
      <div class="page-title">
        <h1>카페 소개</h1>
      </div>
    </section>
  </div>
</body>
</html>
```



cafe12 사이트

```
<div class="content">
  <div class="photo">
    
  </div>
  <div class="text">
    <p>영업 시간 : 오전 9시 ~ 밤 9시</p>
    <p>휴무 : 매주 월요일<br>
      <i><small>(월요일이 공휴일인 경우 월요일 영업, 다음날 휴무)</small></i>
    </p>
    <br>
    <p style="color: ■ blue">now : <span id="demo"></span>
  </div>
</div>
</section>
<section id="choice">
  <div class="page-title">
    <h1>이 달의 추천</h1>
  </div>
  <div class="content">
    <div class="photo">
      
    </div>
    <div class="text">
      <h2>핸드 드립 아이스 커피</h2>
      <ol>
        <li>1인분 기준으로 서버에 각얼음 5조각(한 조각의 20cc) 넣고 추출을 시작한다. </li>
        <li>평상시 보다 원두의 양은 2배 정도(20g)와 추출액은 얼음 포함해서 200cc까지 내린다.</li>
        <li>아이스 잔에 얼음 6~7개 섞어서 시원하게 마신다.</li>
      </ol>
    </div>
  </div>
</section>
```


cafe12 사이트

```
<section id="map">
  <div class="page-title">
    <h1>오시는 길</h1>
  </div>
  <div class="content">
    <div class="photo">
      
    </div>
    <div class="text">
      <p>서울시 영등포구 선유동2로 57 이레빌딩 19층</p>
      <p>2호선 당산역 12번, 13번 출구, 도보 10분</p>
    </div>
  </div>
</section>
<footer>
  <ul id="bottom-menu">
    <li><a href="">카페 소개</a></li>
    <li><a href="">개인정보 처리방침</a></li>
    <li><a href="">이용 약관</a></li>
    <li><a href="">사이트 맵</a></li>
  </ul>
</footer>
</div>
```



cafe12 사이트

```
@charset "utf-8";
/* 공통 스타일 */
*{margin: 0; padding: 0; box-sizing: border-box;}
#container{width: 100%;}

/* header 스타일 */
header{width: 100%; height: 320px;}
header nav{height: 60px; background: ■ #000;}
nav #main-nav{list-style: none; text-align: center;}
nav #main-nav li{display: inline-block; line-height: 60px;
| margin: 0 30px;}
nav #main-nav li a{text-decoration: none; color: □ white;}
nav #main-nav li a:hover{color: ■ orange;}
#main-pic img{width: 100%; height: 260px;}
```

style.css



cafe12 사이트

```
/* section 영역 */
section{height: 320px; padding: 15px 5%}
section:nth-of-type(2n){background: #eee;}
section .page-title{margin: 20px 0;}
section p{line-height: 3;}
section .content .photo{display: none;}
#choice .content h2{margin-bottom: 20px;}
#choice .content ol{padding-left: 30px;}
#choice .content ol li{line-height: 2;}

/* footer 스타일 */
footer{height: 80px; border-top: 2px solid #222;}
#bottom-menu{list-style: none; margin: 10px 10px 10px 40px;}
#bottom-menu li{display: inline-block; margin: 10px 0;
padding: 0 10px; border-right: 1px solid #ccc;}
#bottom-menu li a{text-decoration: none; color: #444;}
#bottom-menu li:last-child{border: none;}
```

cafe12 사이트

```
/* 태블릿 기준 - 780px 이상 */
@media screen and (min-width: 780px) {
  header{height: 350px;}
  nav #main-nav li{margin: 0 40px;}
  #main-pic img{height: 290px;}
  section{height: 380px;}
  #intro, #choice{float: left; width: 50%;}
  #map{clear: both;}
  #map .photo{display: block; float: left; margin-right: 50px;}
}

/* pc 기준 - 1200px 이상 */
@media screen and (min-width: 1200px) {
  #container{width: 1200px; margin: 0 auto;}
  header{height: 380px;}
  nav #main-nav li{margin: 0 50px;}
  #main-pic img{height: 320px;}
  section{height: 350px;}
  #intro, #choice, #map{width: 100%; padding: 10px 5%;}
  section .content .photo{display: block;}
  section p{line-height: 2.5;}
  #intro .photo, #map .photo{float:left; margin-right: 20px;}
  #choice .photo{float: right; margin-right: 50px;}
}
```



그리드 레이아웃

■ 그리드 레이아웃(Grid Layout)

- 반응형 웹 디자인에서 웹 문서 요소를 배치하는 기준
- 웹 사이트 화면을 여러 개의 칼럼(column)으로 나눈 후 웹 요소를 배치
- 화면을 규칙적으로 배열하므로 레이아웃을 일관성 있게 유지

▶ 플렉스 박스 레이아웃

- 수평이나 수직 방향 중 하나를 주축으로 정하고 박스를 배치
- 여유 공간이 생길 경우 너비나 높이를 적절하게 늘리거나 줄일 수 있음



그리드 레이아웃 - flex

■ 플렉스 박스 레이아웃

- 배치할 요소를 감싸는 부모 요소를 만든다.(컨테이너)
- display 속성을 이용한다. (**display: flex**)



```
<style>
  #container{display: flex;}
  .box{
    width: 100px;
    height: 100px;
    background-color: blue;
    margin: 20px;
    /*display: inline-block;*/
  }
</style>
</head>
<body>
  <div id="container">
    <div class="box"></div>
    <div class="box"></div>
    <div class="box"></div>
  </div>
</body>
```

display-flex.html



그리드 레이아웃 - flex

- flex-direction : 가로(row), 세로(column) 방향 배치



flex-direction.html

```
<div class="container">
  <div class="box"><p>1</p></div>
  <div class="box"><p>2</p></div>
  <div class="box"><p>3</p></div>
  <div class="box"><p>4</p></div>
  <div class="box"><p>5</p></div>
  <div class="box"><p>6</p></div>
</div>
```

그리드 레이아웃 - flex

- flex-direction : 가로(row), 세로(column) 방향 배치

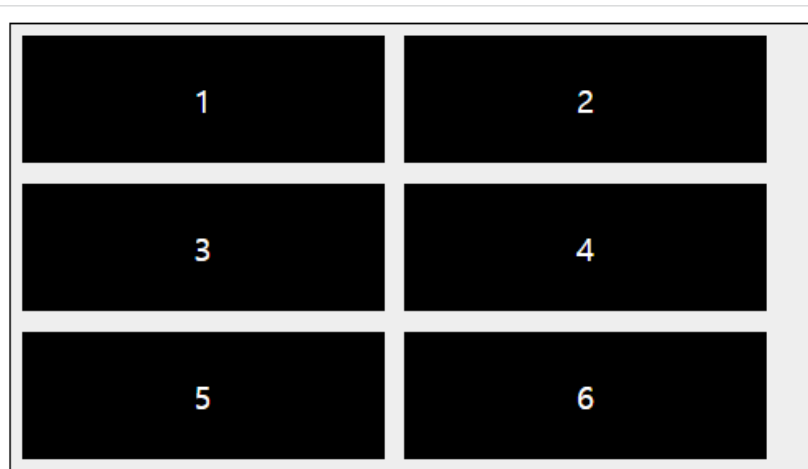
```
<style>
.container {
  width:800px;
  border: 1px solid ■ #000;
  background-color: □ #eee;
  display:flex;
  flex-direction: row; /*주축 : 가로, 순서 : 왼쪽에서 오른쪽으로 배치*/
  /*flex-direction: row-reverse; 오른쪽에서 왼쪽으로*/
  /*flex-direction: column; 주축 : 세로, 순서 : 위에서 아래로*/
  /*flex-direction: column-reverse; /* 주축 : 세로, 순서 : 아래에서 위로 */
}
.box {
  width:80px;
  background-color: ■ #000;
  margin:5px;
  padding:5px 50px;
}
p {color: □ #fff; text-align: center;}
```



그리드 레이아웃 - flex

- flex-wrap : 컨테이너의 범위에 따라 배치 - 줄바꿈

flex-wrap.html



그리드 레이아웃 - flex

- flex-wrap : 컨테이너의 범위에 따라 배치 - 줄바꿈

```
<style>
  .container {
    width:400px;
    border: 1px solid ■ #000;
    background-color: □ #eee;
    display:flex;
    /*flex-wrap: nowrap; /*기본값: 줄을 바꾸지 않음(컨테이너 범위를 벗어남)*/
    flex-wrap: wrap; /*교차축으로 아래로 배치됨(컨테이너 범위내에 배치)*/
    /*flex-wrap: wrap-reverse; 교차축에서 순서를 역순으로 배치 */
  }
  .box {
    width:80px;
    background-color: ■ #000;
    margin:5px;
    padding:5px 50px;
  }
  p {color: □ #fff; text-align: center;}
</style>
```



그리드 레이아웃 - flex

- `justify-content` : 주축의 정렬 방법을 지정 - 수평정렬

`justify-content.html`



그리드 레이아웃 - flex

- justify-content : 주축의 정렬 방법을 지정 - 수평정렬

```
<style>
.container {
    border: 1px solid ■ #000;
    background-color: □ #eee;
    display: flex;
    /*justify-content: flex-start; 주축의 시작점에 맞춰 배치*/
    /*justify-content: flex-end; 주축의 끝점에 맞춰 배치*/
    justify-content: center; /*주축의 중앙에 맞춰 배치*/
    /*justify-content: space-between; 시작점과 끝점에 배치한 후 동일한 간격으로 배치*/
    /*justify-content: space-around; /* 주축에 동일한 간격으로 배치 */
}
.box {
    width: 80px;
    background-color: ■ #000;
    margin: 5px;
    padding: 5px 30px;
}
p {color: □ #fff; text-align: center;}
</style>
```

justify-content.html



그리드 레이아웃 - flex

- `align-items` : 교차축의 정렬 방법을 지정 - 수직 정렬



align-items.html

그리드 레이아웃 - flex

- align-items : 교차축의 정렬 방법을 지정 - 수직 정렬

```
#wrap{
  width: 400px;
  height: 200px;
  border: 1px solid ■#000;
  background: □#eee;
  display: flex;
  /* 수직 정렬(top, middle, bottom과 같음) */
  align-items: start;
  /* align-items: center; */
  /* align-items: end; */
}
.box{
  width: 80px;
  background-color: ■#000;
  margin: 5px;
  padding: 5px 30px;
}
p{color: □#fff; text-align: center;}
```



그리드 레이아웃 - flex

- 플렉스 박스

The flex Property



그리드 레이아웃 - flex

■ 플렉스 박스

```
#wrap {  
  width: 300px;  
  height: 300px;  
  border: 1px solid black;  
  display: flex;  
}  
  
#wrap div {  
  flex: 1;  
}
```

display-flex2.html

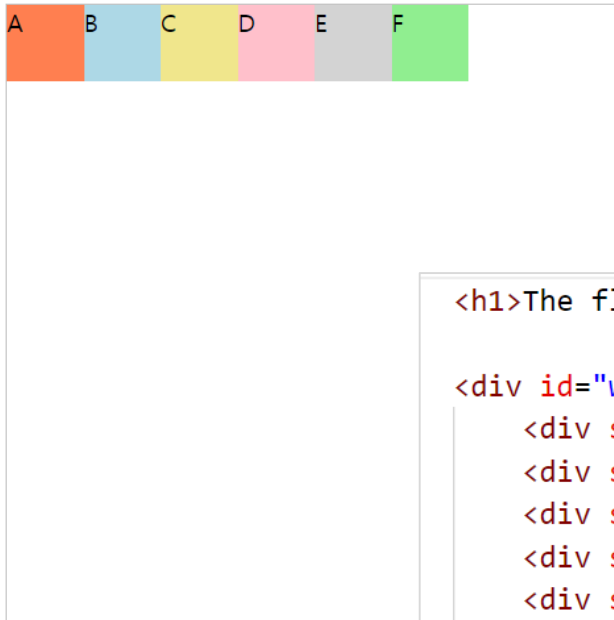
```
<h1>The flex Property</h1>  
  
<div id="wrap">  
  <div style="background-color: coral;">RED</div>  
  <div style="background-color: lightblue;">BLUE</div>  
  <div style="background-color: lightgreen;">Green div with more content.  
</div>
```



그리드 레이아웃 - flex

■ 플렉스 박스 레이아웃

The flex-direction 속성



flex-direction1.html

```
<h1>The flex-direction 속성</h1>

<div id="wrap">
  <div style="background-color: coral;">A</div>
  <div style="background-color: lightblue;">B</div>
  <div style="background-color: khaki;">C</div>
  <div style="background-color: pink;">D</div>
  <div style="background-color: lightgrey;">E</div>
  <div style="background-color: lightgreen;">F</div>
</div>
```



그리드 레이아웃 - flex

■ 플렉스 박스 레이아웃

```
#wrap {  
  width: 400px;  
  height: 400px;  
  border: 1px solid #c3c3c3;  
  display: flex;  
  flex-direction: row;  
  /* flex-direction: row-reverse; */  
  /* flex-direction: column; */  
}  
  
#wrap div {  
  width: 50px;  
  height: 50px;  
}
```

그리드 레이아웃 - flex

- 플렉스 박스 레이아웃

The flex-wrap Property



그리드 레이아웃 - flex

■ 플렉스 박스 레이아웃

```
#main {  
  width: 200px;  
  height: 200px;  
  border: 1px solid #c3c3c3;  
  display: flex;  
  flex-wrap: wrap;  
}
```

```
#main div {  
  width: 50px;  
  height: 50px;  
}
```

flex-wrap1.html

```
<h1>The flex-wrap Property</h1>  
  
<div id="main">  
  <div style="background-color:coral;">A</div>  
  <div style="background-color:lightblue;">B</div>  
  <div style="background-color:khaki;">C</div>  
  <div style="background-color:pink;">D</div>  
  <div style="background-color:lightgrey;">E</div>  
  <div style="background-color:lightgreen;">F</div>  
</div>
```



그리드 레이아웃 - flex

■ 플렉스 박스 레이아웃

FLEXBOX FROGGY

◀ 단계 2 of 24 ▶

justify-content를 한 번 더 사용하여 개구리들이 수평으로 이동하는 것을 도와주세요. 이 CSS 속성은 요소들을 가로선 상에서 정렬하며 다음의 값들을 인자로 받는다는 것을 기억하세요:

- **flex-start**: 요소들을 컨테이너의 왼쪽으로 정렬합니다.
- **flex-end**: 요소들을 컨테이너의 오른쪽으로 정렬합니다.
- **center**: 요소들을 컨테이너의 가운데로 정렬합니다.
- **space-between**: 요소들 사이에 동일한 간격을 둡니다.
- **space-around**: 요소들 주위에 동일한 간격을 둡니다.

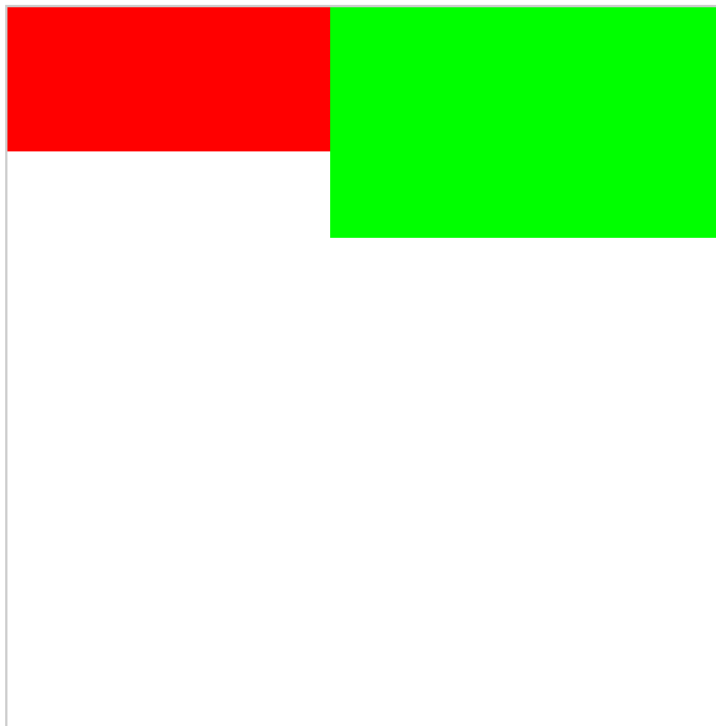
```
1 #pond {  
2   display: flex;  
3   justify-content: center;  
4 }  
5  
6  
7  
8  
9  
10
```

다음



그리드 레이아웃 - flex

▪ 가변 크기 - 가변 패딩



```
#wrap{
  /* width: 660px; */
  width: 60%;
  height: 500px;
  border: 2px solid #ccc;
  margin: 0 auto;
  display: flex;
  /* flex-direction: column; */
  align-items: flex-start;
}

#wrap div{
  /* width: 200px; */
  width: 30.30%;
}

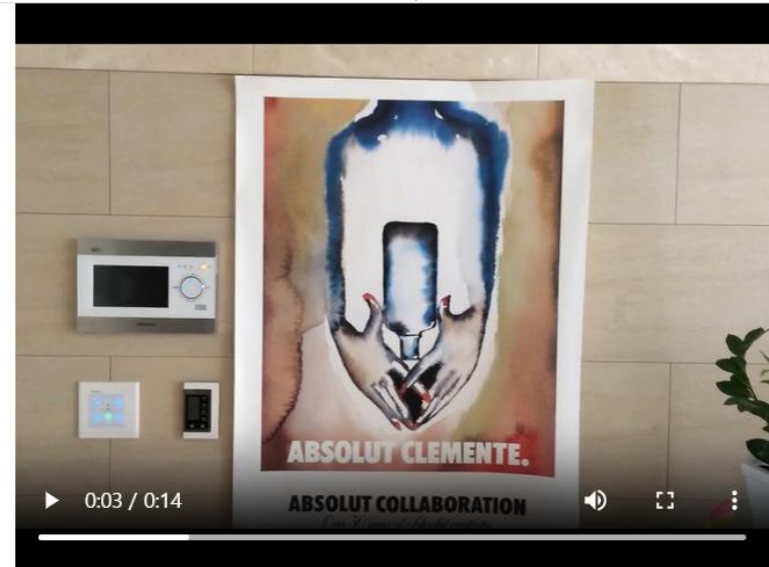
#wrap div:first-child{
  /* width = 300px(200 + 50 + 50) */
  background: #f00;
  /* height: 200px; */
  /* padding: 50px; */ /* 고정 패딩 */
  padding: 50px 7.57%; /* 가변 패딩 */
}

#wrap div:nth-child(2){
  /* width = 360px(200 + 80 + 80) */
  background: #0f0;
  /* padding: 80px; */
  padding: 80px 12.12%;
}
```

가변 크기

▪ 가변 크기 – max-width

```
<style>
  *{margin: 0; padding: 0;}
  #wrap{
    width: 90%;
    margin: 0 auto;
  }
  video{
    width: 100%;
    /* max-width: 80%; */
    max-width: 800px;
  }
</style>
</head>
<body>
  <div id="wrap">
    <div>
      <video controls preload="auto">
        <source src="source/video.mp4" type="video/mp4">
      </video>
    </div>
  </div>
</body>
```



calc() 함수

- CSS의 calc() 함수

박스는 가변적이되 마진이나 패딩은 고정시킬때 사용.



calc() 함수

```
/*{margin:0; padding:0;}
#wrap{
  width: 90%;
  height: 500px;
  margin: 0 auto;
  border: 4px solid ■#000;
}
```

```
/*
```

calc() 함수 - 박스는 가변적이되 마진이나 패딩은 고정시킬때 사용.
50px 크기의 고정 마진을 적용하기 위해 calc() 함수 사용

왼쪽, 오른쪽 패딩값을 더한 값인 100px을 뺀 값을 너비값으로 설정

```
*/
```

```
#wrap div{
  width: 90%;
  /* width: calc(100% - 100px); */
  height: 200px;
  margin: 50px;
  background: ■orange;
}
```

calc() 함수

이것은 배너 광고입니다.

```
/* calc()를 이용하면 고정된 너비의 여백을 가진 요소를 쉽게  
배치할 수 있습니다.  
화면을 가로지르면서 창의 좌우 모서리와 간격이  
40픽셀인 배너를 생성합니다. */  
*{margin:0; padding:0;}  
.banner{  
    position: absolute;  
    top: 100px;  
    left: 80px;  
    width: calc(100% - 160px);  
    border: 1px solid black; #000;  
    padding: 10px;  
    background: yellow;  
    text-align: center;  
    box-sizing: border-box;  
}
```



var() 함수

var() 함수는 "CSS 변수"의 값을 다른 속성의 값으로 지정할 때 사용합니다.

Software Engineer



```
:root{
  /* color */
  --red-color: rgb(212, 60, 60);
  --blue-color: rgb(66, 66, 235);

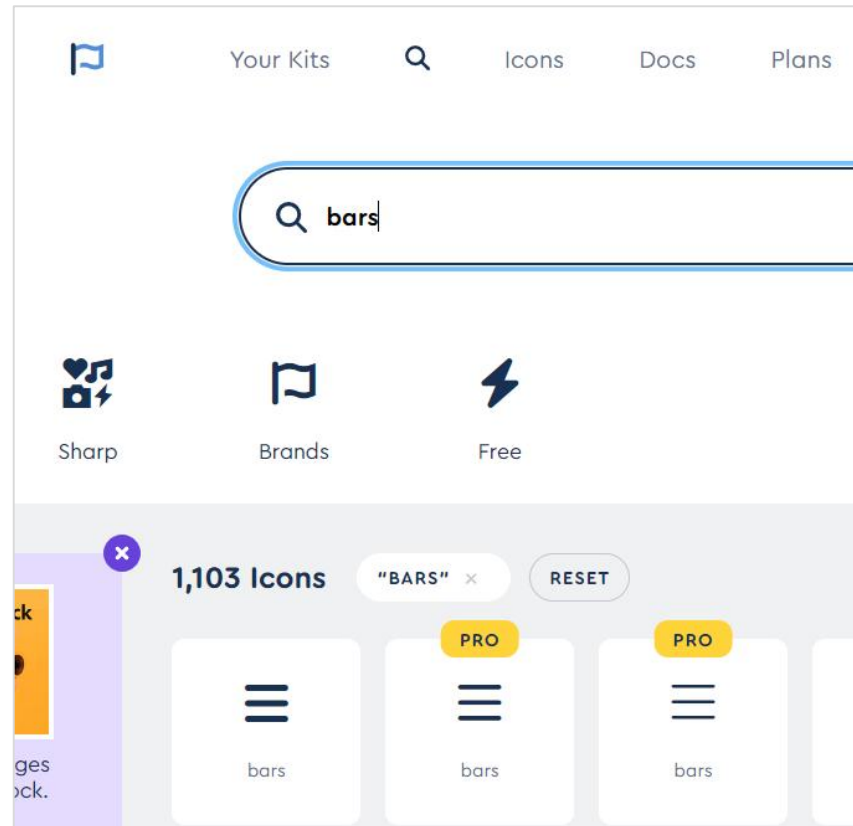
  /* background */
  --green-color: rgb(45, 238, 45);

  /* margin */
  --my-2: 0.5rem;
  --my-3: 1rem;
}
h1{
  color: var(--red-color);
}
div{
  width: 100px;
  height: 100px;
  background: var(--green-color);
  margin: var(--my-3);
  display: inline-block;
}
```



fontawesome

무료 아이콘 얻기 – fontawesom > get started > 이메일 보내서 인증받음



fontawesome

로그인 > 키값 (자바크스립트 소스)

The classic way to use a Font Awesome Kit - just one line of javascript.

```
<script src="https://kit.fontawesome.com/187dabceeb.js" crossorigin="anonymous"></script>
```



부트 스트랩 환경 설정

Bootstrap(부트스트랩) 이란?

부트스트랩은 웹사이트를 쉽게 만들 수 있게 도와주는 HTML, CSS, JS 라이브러리이다. 하나의 CSS로 휴대폰, 태블릿, 데스크탑까지 다양한 기기에서 작동하는 반응형 웹을 구현할 수 있다.

Build fast,
responsive sites
with Bootstrap

Quickly design and customize responsive mobile-first sites with Bootstrap, the world's most popular front-end open source toolkit, featuring Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful JavaScript plugins.

Get started

Download

5.1버전 선택후 다운로드



Download

Bootstrap v5.1 ▼



부트 스트랩 환경 설정

Compiled CSS and JS

Download ready-to-use compiled code for **Bootstrap v4.6.1** to easily drop into your project, which includes:

- Compiled and minified CSS bundles (see [CSS files comparison](#))
- Compiled and minified JavaScript plugins (see [JS files comparison](#))

This doesn't include documentation, source files, or any optional JavaScript dependencies (jQuery and Popper).

Download

다운로드 후 압축풀고 bootstrap.css 파일을 css 디렉터리 안에 삽입



👉 부트스트랩 내부 링크걸기

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



부트 스트랩 환경 설정

CDN via jsDelivr

Skip the download with [jsDelivr](#) to deliver cached version of Bootstrap's compiled CSS and JS to your project.

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-...
```

Copy

If you're using our compiled JavaScript and prefer to include Popper separately, add Popper before our JS, via a CDN preferably.

```
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js" integrity="sha384-...">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js" integrity="sha384-...">
```

Copy



부트 스트랩 Components

- Get started > 좌측 메뉴 : Components > Button, Card 등

Build fast,
responsive sites
with Bootstrap

Quickly design and customize responsive mobile-first sites with Bootstrap, the world's most popular front-end open source toolkit, featuring Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful JavaScript plugins.

[Get started](#) [Download](#)

> Getting started
> Customize
> Layout
> Content
> Forms
▼ **Components**
 Accordion
 Alerts
 Badge
 Breadcrumb
 Buttons
 Button group
 Card
 Carousel
 Close button
 Collapse
 Dropdowns

Buttons

Use Bootstrap's custom button styles for with support for multiple sizes, states, and more.

Audience Network

[Get Started](#)

Meta

Power-up the lifetime value of your app users with rewarded video ads ads via Carbon

Examples

Bootstrap includes several predefined button styles, each thrown in for more control.

[Primary](#) [Secondary](#) [Success](#) [Danger](#)



Components – Buttons

- bootstrap.css > ctrl+F(찾기) > .btn

```
2505 > btn| Aa ab *
```

```
2506 .btn {
```

```
2507     display: inline-block;
```

```
2508     font-weight: 400;
```

```
2509     color: #212529;
```

```
2510     text-align: center;
```

```
2511     vertical-align: middle;
```

```
2512     -webkit-user-select: none;
```

```
2513     -moz-user-select: none;
```

```
2514     -ms-user-select: none;
```

```
2515     user-select: none;
```

```
2516     background-color: transparent;
```

```
.btn-primary {
```

```
    color: #fff;
```

```
    background-color: #007bff;
```

```
    border-color: #007bff;
```

```
}
```

```
.btn-primary:hover {
```

```
    color: #fff;
```

```
    background-color: #0069d9;
```

```
    border-color: #0062cc;
```

```
}
```



```
<button type="button" class="btn btn-primary">Primary</button>
```

```
<button type="button" class="btn btn-secondary">Secondary</button>
```

```
<button type="button" class="btn btn-success">Success</button>
```

```
<button type="button" class="btn btn-danger">Danger</button>
```



components – Badge

▪ 구성요소 - Badge

Headings

Example heading **New**

Example heading **New**

Example heading **New**

Example heading **New**

Example heading **New**

Example heading **New**

```
<h1>Example heading <span class="badge bg-secondary">New</span></h1>  
<h2>Example heading <span class="badge bg-secondary">New</span></h2>  
<h3>Example heading <span class="badge bg-secondary">New</span></h3>  
<h4>Example heading <span class="badge bg-secondary">New</span></h4>  
<h5>Example heading <span class="badge bg-secondary">New</span></h5>  
<h6>Example heading <span class="badge bg-secondary">New</span></h6>
```

Copy



components – card

▪ 구성요소 - Card

Card title

Card subtitle

Some quick example text to build on the card title and make up the bulk of the card's content.

[Card link](#) [Another link](#)

```
<div class="card" style="width: 18rem;">  
  <div class="card-body">  
    <h5 class="card-title">Card title</h5>  
    <h6 class="card-subtitle mb-2 text-muted">Card subtitle</h6>  
    <p class="card-text">Some quick example text to build on the card title and make up the bulk o  
    <a href="#" class="card-link">Card link</a>  
    <a href="#" class="card-link">Another link</a>  
  </div>  
</div>
```

Copy



components – Pagination

▪ 구성요소 - Pagination

Previous 1 2 3 Next

```
<nav aria-label="Page navigation example">
  <ul class="pagination">
    <li class="page-item"><a class="page-link" href="#">Previous</a></li>
    <li class="page-item"><a class="page-link" href="#">1</a></li>
    <li class="page-item"><a class="page-link" href="#">2</a></li>
    <li class="page-item"><a class="page-link" href="#">3</a></li>
    <li class="page-item"><a class="page-link" href="#">Next</a></li>
  </ul>
</nav>
```

Copy



구성 요소 예제

■ 구성요소

Button



Badge

Example heading **New**

Example heading **New**

Example heading **New**

Example heading **New**

Example heading **New**

Example heading **New**



components – Pagination

■ 구성요소

```
<h1 class="title">Button</h1>
<div>
  <button type="button" class="btn btn-primary">Primary</button>
  <button type="button" class="btn btn-secondary">Secondary</button>
  <button type="button" class="btn btn-success">Success</button>
  <button type="button" class="btn btn-danger">Danger</button>
  <button type="button" class="btn btn-warning">Warning</button>
  <button type="button" class="btn btn-info">Info</button>
  <button type="button" class="btn btn-light">Light</button>
  <button type="button" class="btn btn-dark">Dark</button>
</div>

<h1 class="title">Badge</h1>
<div>
  <h1>Example heading <span class="badge bg-secondary">New</span></h1>
  <h2>Example heading <span class="badge bg-secondary">New</span></h2>
  <h3>Example heading <span class="badge bg-secondary">New</span></h3>
  <h4>Example heading <span class="badge bg-secondary">New</span></h4>
  <h5>Example heading <span class="badge bg-secondary">New</span></h5>
  <h6>Example heading <span class="badge bg-secondary">New</span></h6>
</div>
```

components – Pagination

■ 구성요소

Card 제목

Card 부제목

카드 제목을 기반으로 하고 카드 콘텐츠의 대부분을 구성합니다.

[Card link](#)

```
.mx-3 {  
  margin-right: 1rem !important;  
}  
  
.mb-3,  
.my-3 {  
  margin-bottom: 1rem !important;  
}
```

```
<div class="container my-3">  
  <div class="card my-3" style="width: 20rem">  
    <div class="card-body">  
      <h5 class="card-title">Card 제목</h5>  
      <h6 class="card-subtitle mb-2 text-muted">Card 부제목</h6>  
      <p class="card-text">  
        카드 제목을 기반으로 하고 카드 콘텐츠의 대부분을 구성합니다.  
      </p>  
      <a href="#" class="card-link">Card link</a>  
    </div>  
  </div>  
</div>
```



그리드 시스템(Grid System)

■ 그리드 시스템(Grid System)

부트스트랩은 기본적으로 한 행(row)을 12열(column)로 구분하는 그리드 시스템을 제공하는데 반응형 웹 페이지를 구현하기 위한 필수 기능이다.

== 규칙 ==

- row 클래스는 <div class="row"> 형태로 container안에 위치해야 정상적인 배열이 가능함
- 내용은 col-* 형태로 의 자식 요소로 배치되어야 함 <div class="col-6">

```
.col-1 {  
  -ms-flex: 0 0 8.333333%;  
  flex: 0 0 8.333333%;  
  max-width: 8.333333%;  
}  
  
.col-2 {  
  -ms-flex: 0 0 16.666667%;  
  flex: 0 0 16.666667%;  
  max-width: 16.666667%;  
}
```

```
.col-3 {  
  -ms-flex: 0 0 25%;  
  flex: 0 0 25%;  
  max-width: 25%;  
}  
  
.col-4 {  
  -ms-flex: 0 0 33.333333%;  
  flex: 0 0 33.333333%;  
  max-width: 33.333333%;  
}
```



그리드 사용

■ 그리드 사용 예제

구조와 정보

106m² / 32평
복층, 개별바베큐

기준 4명 ~ 최대 4명

입·퇴실 안내

입실 시간 : 오후 3:00 ~ 오후 9:00

퇴실 시간 : 오전 11:00

오후 9:00이후 입실시에는 사전에 꼭 연락 바랍니다



그리드 사용

■ 그리드 사용 예제

```
<div class="container my-3">
  <div class="row">
    <div class="col-6">
      <div class="card">
        <div class="card-body">
          <h5 class="card-title">구조와 정보</h5>
          <p class="card-text">
            106㎡ / 32평<br>
            복층, 개별바베큐<br>
            <br>
            <strong>기준 4명 ~ 최대 4명</strong>
          </p>
        </div>
      </div>
    </div>
    <div class="col-6">
      <div class="card">
        <div class="card-body">
          <h5 class="card-title">입·퇴실 안내</h5>
          <p class="card-text">
            입실 시간 : 오후 3:00 ~ 오후 9:00<br>
            퇴실 시간 : 오전 11:00<br>
            <br>
            <strong>오후 9:00이후 입실시에는 사전에 꼭 연락 바랍니다</strong>
          </p>
        </div>
      </div>
    </div>
  </div>
</div>
```



쇼핑몰 메인 페이지

GoodMall | mall 소개 | 카테고리 | 공지사항

Welcome to Shopping Mall

좋은 품질, 착한 가격을 만나보세요.

다양한 상품

당신과 가장 가까운 마트.. 장보기의 기쁨을 더합니다.

[상세정보 >](#)

질 좋은 상품

당신과 가장 가까운 마트.. 장보기의 기쁨을 더합니다.

[상세정보 >](#)

합리적 가격

당신과 가장 가까운 마트.. 장보기의 기쁨을 더합니다.

[상세정보 >](#)

상품 정보 1



상품 이미지

상품 정보 2



상품 이미지

브랜드명 : GoodMall | 대표자 : 다있소 | 사업자등록번호 : 123-12-12345
주소 : 서울시 구로구 0000-0000
계좌번호 : 우리은행(홍길동) XXX-02XX-XXXX-XX | 예약문의 : 010-0000-0000

HTML



CSS



네비게이션 바(navbar)

■ 네비게이션바(navbar) 메뉴 스타일 적용하기

GoodMall mall 소개 카테고리 공지사항

```
<div class="container">
  <header>
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
      <div class="container-fluid">
        <a class="navbar-brand" href="#">GoodMall</a>
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
          data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false"
          aria-label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
          <ul class="navbar-nav">
            <li class="nav-item">
              <a class="nav-link" href="#">mall 소개</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="#">카테고리</a>
            </li>
          </ul>
        </div>
      </div>
    </nav>
  </header>
</div>
```



네비게이션 바(navbar)

■ 네비게이션바(navbar) 메뉴 스타일 적용하기

```
<li class="nav-item">
  <a class="nav-link" href="#">공지사항</a>
</li>
</ul>
</div>
</nav>
<div class="my-3">
  <h1 class="display-4">Welcome to Shopping Mall</h1></strong>
  <p class="lead">좋은 품질, 착한 가격을 만나보세요.</p></strong>
  <hr class="my-4">
</div>
</header>
```

```
.navbar {
  position: relative;
  display: -ms-flexbox;
  display: flex;
  -ms-flex-wrap: wrap;
  flex-wrap: wrap;
```

```
.nav {
  display: -ms-flexbox;
  display: flex;
  -ms-flex-wrap: wrap;
  flex-wrap: wrap;
  padding-left: 0;
  margin-bottom: 0;
  list-style: none;
}
```



부트 스트랩 – 레이아웃

```
<section>
  <div class="row">
    <div class="col-4">
      <div class="card">
        <div class="card-body">
          <h5 class="card-title">다양한 상품</h5>
          <p class="card-text">
            당신과 가장 가까운 마트..
            장보기의 기쁨을 더합니다.
          </p>
          <a href="#" class="btn btn-primary">상세정보 &raquo;</a>
        </div>
      </div>
    </div>
    <div class="col-4">
      <div class="card">
        <div class="card-body">
          <h5 class="card-title">질 좋은 상품</h5>
          <p class="card-text">
            당신과 가장 가까운 마트..
            장보기의 기쁨을 더합니다.
          </p>
          <a href="#" class="btn btn-success">상세정보 &raquo;</a>
        </div>
      </div>
    </div>
  </div>
```

열(row) 공간 나누기
12/2 -> 6등분
12/4 -> 3등분



부트 스트랩 – 레이아웃

```
<!-- 상품 이미지 -->
<div class="col-6 my-4 text-center">
  <div class="card">
    <div class="card-body">
      <h5 class="card-title">상품 정보 1</h5>
      
      <p class="card-text">
        상품 이미지
      </p>
    </div>
  </div>
</div>
<div class="col-6 my-4 text-center">
  <div class="card">
    <div class="card-body">
      <h5 class="card-title">상품 정보 2</h5>
      
      <p class="card-text">
        상품 이미지
      </p>
    </div>
  </div>
</div>
</div>
</section>
```



부트 스트랩 – footer

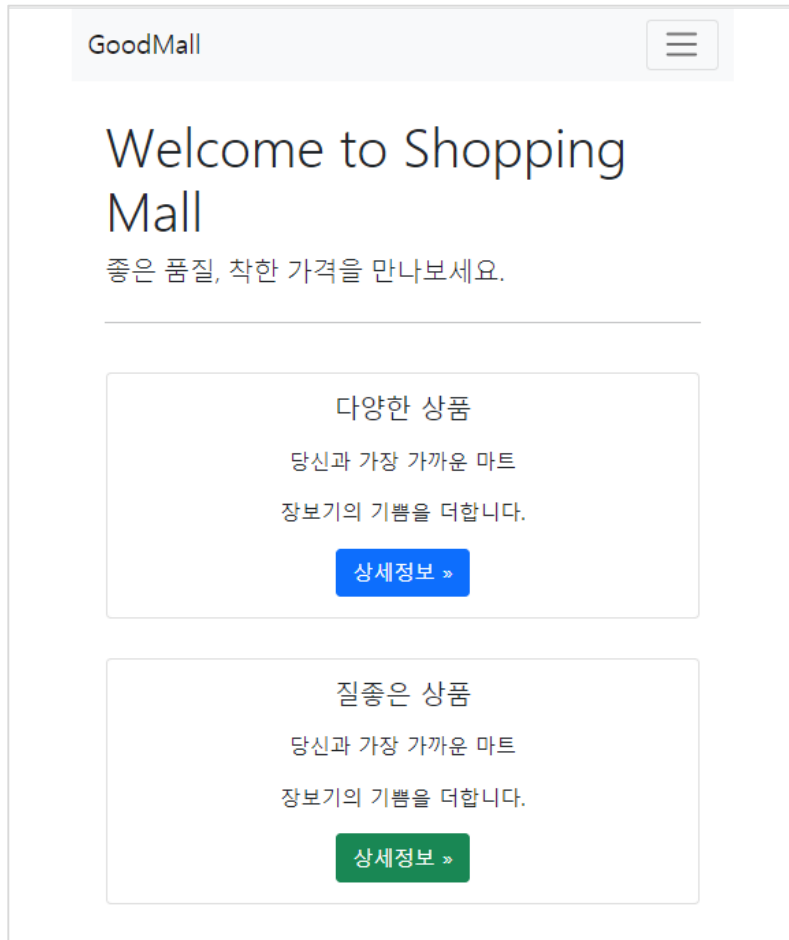
▪ footer – 바닥글 스타일

브랜드명 : GoodMall | 대표자 : 다 있 소 | 사업자등록번호 : 123-12-12345
주소 : 서울시 구로구 0000-0000
계좌번호 : 우리은행(홍길동) XXX-02XX-XXXX-XX | 예약문의 : 010-0000-0000

```
<footer class="py-4 bg-dark">
  <div class="container">
    <p style="color: #ccc">
      브랜드명 : GoodMall | 대표자 : 다 있 소 | 사업자등록번호 : 123-12-12345<br>
      주소 : 서울시 구로구 0000-0000 <br>
      계좌번호 : 우리은행(홍길동) XXX-02XX-XXXX-XX | 예약문의 : 010-0000-0000<br>
    </p>
  </div>
</footer>
```



부트 스트랩 – 모바일 최적화



```
@media screen and (min-width: 320px)
  and (max-width: 780px){
    header .title{padding: 10px 5% 0;}
    .col-4{
      width: 90%;
      text-align: center;
      margin: 15px 5%;
    }
    .col-6{
      width: 90%;
      margin: 0 5%;
    }
    .card img{border-radius: 5px;}
  }
```

