



HTML / CSS

⌚ 작성일시	@2023년 2월 20일 오후 8:00
▼ 강의 번호	이세빈 강사님
▼ 유형	강의
📎 자료	
☒ 복습	<input type="checkbox"/>

[웹 컴바인 #1]

- **html**

<!DOCTYPE html> 사용할 형식 선언

<html lang="ko"> 사용할 언어 선택

<meta charset="UTF-8"> 언어설정/셋팅

- ----- block 요소-----

1. h태그 = header

- 제목글, 머릿말을 적을 때 사용하는 태그
- h1 ~ h6까지 사용됨
- h1의 글자가 가장 크고 h6의 글자가 가장 작음
- 중요도에 따라서 순서 부여
- h1은 logo에 주로 사용
- <hn(숫자넘버)></hn(숫자넘버)>

2. p태그 = paragraph

- 본문을 나타내기 위해서 단락으로 구분하는 태그
- <p></p>
- ----- block 요소-----
- ----- inline 요소-----

3. br태그 = break

- 줄바꿈 태그
- 단독태그, self closing tag
-
 ,
 둘 다 사용가능

4. b태그 = bold

- 글자를 줄바꿈 없이 두껍게 처리할 때 사용
- 단순 강조를 위한 태그
-

5. strong 태그

- 경고 및 주의사항에 사용
-

6. i, em 태그

- 글자를 기울임
- <i></i>
-

7. mark 태그

- 형광펜 효과
- <mark></mark>

8. mark 태그

- 글자를 더 작게 만듬
- <small></small>

• 목록 태그의 종류

==> 목록 + 아이템

1. 순서가 있는 목록

- ol = order list (block 요소)

2. 순서가 없는 목록

- ul = unorder list (block 요소)
- 메뉴를 구성할 때 많이 사용됨
- 자식태그로 반드시 li만 을 수 있음(li : list item)
- <부모태그>


```
<자식태그><자식태그/>
</부모태그>
```
- li는 ol, ul 내부에서만 유효한 태그 , 단독 사용 불가
- a 내부에 있을 때는 순서를 출력
- ul 내부에 있을 때는 marker 출력

3. 설명목록 = 정의목록

- dl = definition list
 - dt = definition term = 정의되는 용어의 제목
 - dd = definition description = 정의되는 제목에 대한 설명
- ※ dt, dd 태그는 dl 내부에서만 유효한 태그 , 단독 사용 불가
- ※ dl 태그 내부에 하나의 dt, 여러개의 dd로 구성 가능

• CSS (cascading style sheet)

==> 특정 태그에 스타일을 입혀주는 역할

- CSS 적용방법

- 1) inline 스타일 (줄바꿈 x)

- span 태그

==> width, height, margin 적용 불가

==> 그룹을 특정짓거나 묶어줄 때 사용

==> 단순 꾸밈 요소를 넣을 때 사용

- a 태그 = anchor

==> <a 속성="속성값">

==> a 태그의 속성 2가지

==> href="이동할 주소" ★필수 속성

==> target="_blank" : 새창으로 이동할 주소 열람 ★선택 속성

target="_self" : 현재창에서 이동할 주소 열람

- img 태그

==> img 태그 속성 2가지

==> scr="이미지주소"

==> alt="이미지에 대한 설명" ★필수 속성

==>

==> 스크린 리더기가 화면에서 이미지를 설명할 경우

==> 이미지가 유실 될 경우를 대비해서 이미지 설명을 적어줌

==> 이미지가 유실되지 않을 경우에는 기본적으로 보이지 않음

==> 화면 사이즈 비율에 따라 적용된다

※ 이미지 삽입 방법 2가지

- 웹 사이트에서 이미지 주소 복사 후 src에 붙여넣기

- 이미지 저장 후 저장된 폴더 경로 지정 후 불러오기

==> ./ , ./(현재경로) , 단축키 Ctrl + Space : 현재 경로 파일 지정

==> ex) 현재 경로

- 2) 내부스타일

- 3) 외부스타일

- 선택자

==> html----선택자(selector)-----css

- 선택자의 종류

- 1) tag선택자

- 태그명을 css에 불러서 적용함

- ex) h2{스타일}

- 2) id선택자

- <태그 id="선택자명"></태그>

- css에서 선택하는 방법

==> #선택자명{스타일}

- 개발자가 주로 사용
- 같은 id명 중복 사용 불가
- 추후 오류 방지를 위해 선택자명을 중복 사용 금지하도록 약속
- id 선택자의 우선순위가 더 높음

3) class선택자

- <태그 class="선택자명"></태그>
 - css에서 선택하는 방법
- ==> .선택자명(스타일)
- 퍼블리셔가 주로 사용
 - 같은 클래스면 중복 가능
 - 퍼블리싱의 효율을 높이기 위해 주로 사용

※ 선택자의 우선순위

- id선택자 > class선택자 > tag선택자
- 동일한 선택자를 기준으로는 마지막에 쓰여진 선택자가 우선 적용됨

```
태그{
    width: 100px;
    height: 100px;
    background-color:gold;
}
```

- 여백의 적용 방향

==> top / bottom / left / right

- 여백의 종류

1. 외부 여백

- margin

==> 박스모델의 외부에 여백값이 적용됨

==> 박스의 기본 크기 값은 변하지 않음

2. 내부 여백

- padding

==> 박스모델의 내부에 여백값이 적용됨

==> 박스의 기본 크기값에 변화를 줌

==> 위/아래/좌/우 4가지 방향으로 설정 가능

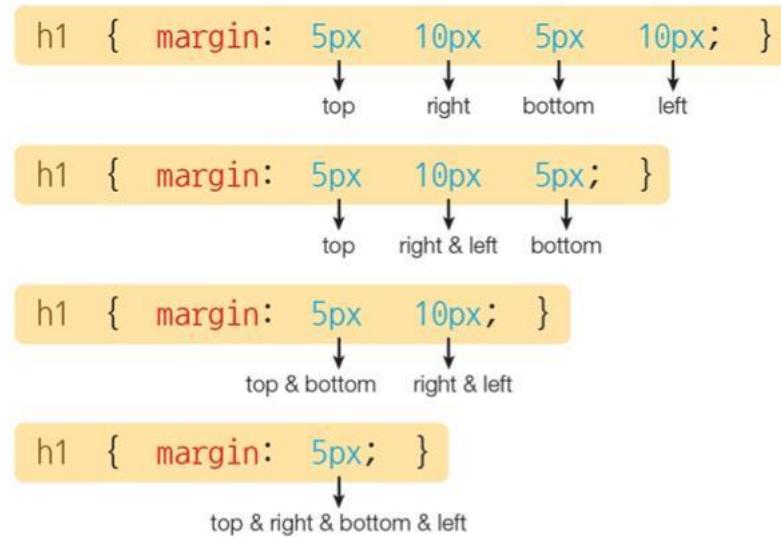
※ margin, padding의 축약

- margin : top right bottom left; ※ 적용값이 모두 다를 경우
 - margin : top (right/left) bottom; ※ 좌, 우 여백이 같고 위, 아래의 여백값만 다를 경우
- ==> ex) margin : 10px 50px 20px;
- margin : (top/bottom) (left/right); ※ 위, 아래 여백이 같고, 좌, 우 여백이 같을 경우
- ==> ex) margin : 50px 20px

==> ex) margin : 0px auto

- margin : (top/right/bottom/left); ※ 적용값이 모두 같을 경우

==> ex) margin : 50px



※ margin으로 중앙 정렬하기

- margin : 0 auto
- 위 아래의 여백값은 없고, 좌,우 여백은 자동으로 중앙 맞춤
- 사용시, width값 필수

※ margin 버그

- 자식태그가 부모요소에서 중앙 정렬이 되고싶을 때
- 보통 margin : 0 auto; 를 사용하게 되는데 부모값에 border 속성이 없으면 제대로 적용되지 않음 (border(테두리) 속성이 선행되어야 함)
- 자식태그의 margin-top 속성 부여 시 부모태그의 margin-top 값에 상관 없이 자식태그의 속성이 적용된다

• CSS 속성

==> width = 가로값;

==> height = 세로값;

==> background-color = 배경색;

==> color = 폰트의 색상;

• CSS에서 컬러 다루는 법

==> 색상명 적어주기 : red

==> #HEX

==> #000(검은색), #fff(흰색)

==> rgb

==> rgb(0,0,0) rgb(255,255,255)

• border = 테두리;

==> border-width : 테두리 가로

==> border-color : 테두리 색상

==> border-style : 테두리 스타일(가장많이 쓰는 속성 : solid(실선))

※ 반드시 위 세가지 속성이 모두 적용야지만 적용 가능

- **border 속성 출력**

==> border : width style color;

※ **box-sizing: border-box;** 테두리를 박스 내부로 적용하는 속성

==> 박스의 전체 사이즈에서 적용된 테두리값만큼 빠지기 때문에 정렬할 때 해당 값을 빼줄것

==> ex)

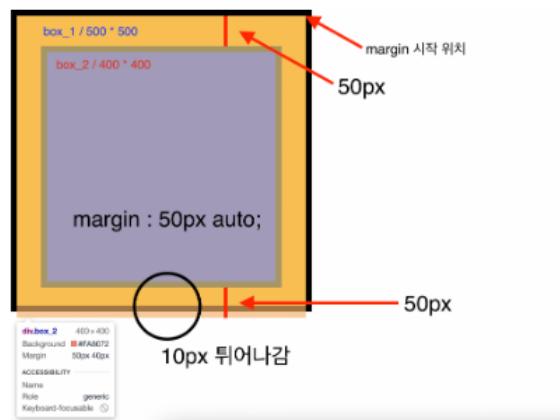
width : 300 , height : 300 border : 10px 인 경우

1) 박스사이즈는 300 * 300 이 아닌 위(top), 아래(bottom) 사이즈 각각 10을 제외한 값인 280 * 280 사이즈가 적용됨

2) 해당 박스를 사이즈가 400 * 400인 박스 가운데 정렬하려면 400 * 400 박스 역시 테두리값을 제외한 380 * 380

즉, $380 - 280 = 100$, 50px로 계산되어지나, margin의 경우 박스 테두리 외부 가장자리가 아닌 테두리 안쪽의 영역부터 시작하기 때문에 테두리 값 10만큼

오버하게되며, 이 때 위, 아래의 정중앙을 맞추기 위해 테두리 값 10을 2로 나눈 값 5를 내부 박스의 마진값에서 빼준다.



<margin 값 설정 관련 참고 이미지>

- **text-align = 글씨의 가로 정렬**

==> text-align : center; 글자 중앙 정렬

- **line-height = 글씨의 세로 정렬**

==> line-height 를 이용한 글자 중앙 정렬의 경우 px값을 height와 동일하게 해주는게 공식

※ 태그안에 태그 중첩 가능

```
<div class="out_box">
  <div="middle_box">
    <div="in_box"></div>
  </div>
</div>
out_box = 부모 태그
middle_box = 자식태그
in_box = 자손 태그
```

```
<div class="out_box">  
<div="in_box"></div>      = 형제태그  
<div="in_box"></div>  
</div>
```

- html -----selector----- css

1. tag

2. id

3. class

※ 적용 우선 순위(level)

- id > class > tag

※ 퍼블리싱 : class

※ 개발 : id

- .클래스명 + tap : div 자동완성

- 태그명앞글자 + tap : 태그 자동완성

- 태그명앞글자.클래스명 + tap : 태그 class:클래스명 자동완성

• 선택자의 종류

1. tag

2. id

3. class

----- id, class 만 유일하게 태그에 직접 적용 가능한 선택자 -----

4. 전체선택자(*)

- 주로 브라우저에 내장된 기본 여백값 제거 시 사용
- 사용규칙 :

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

5. 그룹선택자(,)

- 동일한 스타일이 적용될 태그들을 ,로 구분하여 그룹으로 스타일 적용

6. 자식, 자손 선택자

• 6-1) 자식선택자 (>)

==> 계층구조 안에서 바로 아래에 오는 자식 요소만 선택

==> 선택자와 선택자는 '>' 로 구분

==> 선택자 1 > 선택자 2 : 선택자 1번 바로 아래에 오는 선택자 2번 선택함

==> 부모선택자 > 자식선택자 (자식선택자 < 부모선택자 x)

• 6-2) 자손선택자 (공백)

==> 계층구조에서 하위에 오는 모든 자손 요소 선택

==> 선택자와 선택자는 공백으로 구분

==> 선택자 1 선택자 2 : 선택자 1의 자손 중 선택자 2에 해당하는 모든 요소 선택
==> 상위계층인 자식태그를 포함하여 스타일을 반영함
 ==> ex) 부모태그 span{}
==> 부모태그 안에 중복되는 여러개의 태그가 있는 경우 모든 태그를 선택하는 기능도 수행함
==> 부모선택자 자손선택자

7. 구조선택자

• 7-1) 일반 구조 선택자

==> :first-child = 형제 관계 중에서 첫번째에 위치하는 태그 선택
==> :last-child = 형제 관계 중에서 마지막에 위치하는 태그 선택
==> nth-child(n) = 형제 관계 중에서 n번째에 위치하는 태그 선택
※ 짹수 선택) nth-child(2n)
홀수 선택) nth-child(2n+1)

• 7-2) 형태 구조 선택자

==> :first-of-type = 형제 관계 중에서 첫번째로 등장하는 특정 태그 선택
==> :last-of-type = 형제 관계 중에서 마지막으로 등장하는 특정 태그 선택
==> :nth-of-type = 형제 관계 중에서 n번째로 등장하는 특정 태그 선택

8. 형제선택자

• 8-1) 인접 형제 선택자 (+)

==> 기준선택자 + 형제선택자
==> 두 선택자가 같은 계층에 있을 때 기준 선택자 바로 뒤에 형제 선택자를 선택함

• 8-2) 일반 형제 선택자 (~)

==> 기준선택자 ~ 형제선택자
==> 두 선택자가 같은 계층에 있을 때 기준 선택자 뒤에 있는 모든 형제 선택자를 선택함

9. 가상선택자

- ::before, ::after
- html문서 수정 없이 css만으로 디자인적인 요소를 추가할 때 사용
- 코드 가독성을 높이는데 좋음

※ 특정 선택자의 앞에 요소를 추가할 때 사용

```
선택자::before{  
    content:"";  
}
```

※ 특정 선택자의 뒤에 요소를 추가할 때 사용

```
선택자::after{  
    content:"";  
}
```

★★ before, after 가상 선택자 사용시에는 content:""; 필수 (반영할 컨텐츠 없을 경우 포함)

- 이중클래스 / 다중 클래스

==> <태그 class="box_1 box_2"></태그>

==> .box1

==> .box2

※ 두개의 클래스명 모두 적용

==> 클래스명은 중복 사용이 가능하기 때문에 공통되는 css영역에 반복해서

사용해야하는 경우라면 class명을 하나더 추가해서 사용할 수 있음

==> css코드 단순화에 도움된다

==> <태그 id="box_1 box_2"></태그>

==> id 선택자는 다중 사용 불가

★ 클래스 선택자는 중복 사용 가능한 성질을 가지고 있기 때문에 다중 사용 가능하지만

id 중복 사용 불가해서 위 방식 적용 불가

- 요소 = display

1. block

- 대표 : div

- width, height, 여백값 적용 가능

2. inline

- 대표 : span

- width, height, 여백값 적용 불가 ※ padding(내부여백)은 적용됨

3. inline-block

- 인라인 요소의 성질과 블럭 요소의 성질을 모두 가짐

- block 처럼 width, height, margin이 적용 되면서 inline 요소처럼 한줄로 정렬됨

• float (최근에는 잘 이용하지 않는 추세)

==> 태그를 마치 떠 있는 것과 같은 상태로 만들어서 왼쪽과 오른쪽으로 정렬하는 속성

==> float을 적용한 요소는 문서의 흐름상에서 벗어난 상태

==> 레이아웃을 무너뜨릴 수 있음

==> float속성을 활용하여 레이아웃 정리가 끝났다면, 반드시 해지해줘야 함

• float 사용 해지

==> 1) clear:both

==> float한 요소 바로 다음에 나오는 요소에게 적용

★ 6일차 ‘과제 1’ 풀이 관련

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

※ 기본 여백값만 제거하면 가상선택자(:::marker)가 살아있으므로 완전히 제거하기 위해 아래와 같은 옵션 적용

```
ul, li{  
    list-style: none;  
}
```

※ ul 태그 사용 시에는 반드시 해당 옵션을 적용해 줄 것

```
.. 공통 속성 묶음  
  
1) .box_1 > li{  
    width: 500px;          * %단위로 맞출때 부모의 width 크기와 동일하게 적용해주면 된다  
    height: 100px;         * 부모값만 조정하면 자식값은 자동으로 조정되는 장점  
    text-align: center;  
    line-height: 100px;  
    font-size: 30px;  
    font-weight: bold;  
  
2) .box_1 > li:nth-child(n): 으로 사용 가능
```

```
.box{  
    width: 300px;  
    height: 1000px;          * padding 옵션으로 대체 가능 (활용도 더 높음)  
    background-color: gold;  
    float:left;
```

- **background-attachment : 배경이미지 고정**

1) background-attachment: scroll;

- 기본값, 스크롤의 움직임에 따라 이미지가 보일 수 있고 안보일 수 있음

2) background-attachment:fixed;

- 배경이미지 고정

- 이미지 관련 태그

1. img tag

- 반응형 작업 할 때 배율에 맞게 줄어듬
- 원본 비율을 해치지 않고 그대로 유지

2. background-image

- 배경이미지를 넣을 때 쓰는 속성
- height 필수 / 없으면 출력 X

==> 어떤 요소에 배경이미지를 넣으면 padding 영역을 포함해서 배경이미지 채움

----- background-image 사용 속성 -----

1. background-image

- **1-1) background-image: url("저장한 경로 지정.확장자");**

==> 이미 저장된 이미지를 불러 올 때

==> 단축키 : ctrl + space 후 이미지 파일 선택

2. background-repeat

- **2-1) background-repeat : 배경이미지 반복 속성**

==> 기본값 : background-repeat: repeat;

==> 이미지가 영역을 반복하면서 모두 채움

- 2-2) **background-repeat: repeat-x;**

==> 이미지가 width만큼 반복 배열

- 2-3) **background-repeat: repeat-y;**

==> 이미지가 height만큼 반복 배열

- 2-4) **background-repeat: no-repeat;**

==> 이미지 반복 x (가장 많이 사용)

3. **background-position**

- 3-1) **background-position : 배경이미지 위치 조정**

==> background-position : 좌표 좌표;

==> ex) background-position : top center;

- 3-2) **background-position : Xpx Ypx**

==> ex) background-position : 100px 100px;

- 3-3) **background-position : X% Y%**

==> ex) background-position : 10% 10%

4. **background-size : 배경 크기 조정**

- 4-1) **background-size : auto;**

==> 기본값, 이미지 원래 크기 그대로 출력

- 4-2) **background-size : cover;**

==> 배경 지역(영역)이 이미지로 완전히 덮히도록 이미지를 가능한 크게 표시함

==> 이미지 일부 잘릴 수 있음

- 4-3) **background-size : contain;**

==> 이미지의 비율을 유지하면서 배경영역을 채움

- 4-4) **background-size : X Y;**

==> px, %로 사이즈 조정 가능

※ bacground 축약

```
background : url() repeat position / size;  
* 사이즈가 생략된 경우 가져온 원본이미지 사이즈 유지
```

★ 7일차 ‘과제 1’, ‘과제 2’ 풀이 관련

```
.footer{  
    width: 500px;
```

```
height: 50px;
clear: both;
background-color: skyblue;
font-size: 20px;
font-weight: bold;
```

*block는 반드시 'width' 값을 넣어줘야 함

```
<div class="box">          *자식 태그가 속성이 제대로 적용되도록 전체 자식태그를 감싸주는 부모 태그를 사용하는 것을 추천
    <div class="header"><h2>header 영역</h2></div>
    <div class="nav"><h2>nav 영역</h2></div>
    <div class="section"><h2>section 영역</h2></div>
    <div class="footer"><h2>footer 영역</h2></div>
</div>
```

```
*{
    margin: 0;
    padding: 0;
    box-sizing: border-box;      *border 속성 사용 시 테두리를 내부로 적용해주는 옵션을 초기화 선언 시 기본적으로 적용해 줄 것
}
```

```
<div class="section"><h2>section 영역</h2>
    <p>Lorem ipsum dolor sit, amet consectetur adipisicing elit. Tenetur asperiores iure ex facilis earum ea officia odio deleniti
```

* 부모의 'height'를 따로 지정하지 않으면 자식의 높이값을 따라간다.

```
.section{
    width: 75%;
    height: 100px;           *padding: 30px 0; (위, 아래 30px씩 내부여백을 줌)
    border: 3px solid skyblue;
    float: left;
    font-weight: bold;
    text-align: center;
    box-sizing: border-box;
```

- **css hover** : 사용자의 마우스 포인터가 요소에 팝업되었을 때 나타나는 효과

=> 모바일이나 테블릿에서는 효과 적용이 어려움

- **hover 적용**

```
선택자:hover{}
```

- **hover 옵션**

=> color : 글자색 변경

=> background-color : 배경색 변경

★ 8일차 '과제 1', '과제 2' 풀이 관련

```
.box-1:hover{
    background-color: blue;
    cursor: pointer;        * 마우스 팝업 시 커서모양 변경
                            * 글자위에 마우스 팝업 시에도 속성 유지됨
}
```

```
<!DOCTYPE html>
<html lang="ko">
<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>hover 예제 2</title>
    <style>
```

```
*{
```

```

        margin: 0;
        padding: 0;
        box-sizing: border-box;
    }

    a{
        text-decoration: none;
        color: initial;
    }

    .ul{
        list-style: none;
    }
    /* 초기화영역

    .wrap{
        width: 1200px;
        height: 100px;
        background-color: gray;
        margin: 0 auto;
        /* margin: 0 auto 의 경우 width 값 필수 */
    }

    .logo{
        width: 100px;
        height: 50px;
        /* hight는 되도록 px값으로 설정 */
        background-color: coral;
        text-align: center;
        line-height: 50px;
        margin: 25px;
        float: left;
    }

    ul{
        width: 1000px;
        height: 50px;
        background-color: yellow;
        margin: 25px;
        float: left;
    }

    ul > li{
        width: 184px;
        height: 50px;
        /* %로 사용할 경우 확실하게 값을 정할 수 있을 때 사용한다 */
        background-color: violet;
        float: left;
        line-height: 50px;
        text-align: center;
        list-style-type: none;
        /* list-style-type : ul, cl에서 mark 없애기 */
    }

    ul > li:first-child{
        margin-right: 20px;
    }

    /* 클래스 지정을 통해 전체영역에 margin-right 값을 지정 후
    가장 마지막 li에 margin-right 값을 0으로 지정하면 코드 간소화

    ul > li:first-child:hover{
        background-color: skyblue;
    }

    ul > li:nth-child(2){
        margin-right: 20px;
    }

    ul > li:nth-child(2):hover > a{
        color: white;
        /* hover 글자색 변경 */
    }

    ul > li:nth-child(3){
        margin-right: 20px;
    }

    ul > li:nth-child(3):hover{
        border-radius: 30%;
    }

    ul > li:nth-child(4){
        margin-right: 20px;
    }

```

```

        ul > li:nth-child(4):hover{
            background-color: blue;
            color:white;
        }

        ul > li:nth-child(4):hover a{
            color:white;
        }

    
```

</style>

</head>

<body>

<div class="wrap">

<h1 class="logo">로고</h1>

<!-- logo를 만들때는 대부분 h1 사용 -->

메뉴1

메뉴2

메뉴3

메뉴4

메뉴5

<!-- ★ ul또한 div이기 굳이 div 내부에 중첩할 필요 없음-->

</div>

</body>

</html>

• 시멘틱태그

==> 웹 사이트의 구조를 설계하기 위한 태그

==> 의미가 있는 태그

==> 사이트의 구조 파악을 용이하기 위한 용도

• 시멘틱태그를 사용하는 이유

1. 검색엔진 최적화

- 검색엔진이 태그의 목적에 부합하게 설계되어 있는 구조의 사이트에서 더욱 효율적으로 정보를 파악할 수 있어서 검색 결과 노출에 유리함

2. 웹 접근성

- 스크린리더기와 같은 환경에서 웹 접근성과 사용성 향상

3. 유지보수의 용이성

- 많은 div 태그의 사용으로 관리가 어려워지는 문제점에서 벗어나 태그의 이름만으로 어떤 영역인지 바로 확인 가능 특정 영역의 태그 특성에 맞는 업 구분하여 진행하기 좋음

• 시멘틱태그의 종류

1. header

- <header></header>

==> 페이지의 제목, 소개

==> 로고

==> header 태그 안에는 h1~h6에 해당하는 하나의 제목이 필요함

※ 통상적으로 로고를 h1태그에 담아서 작업

★ header 구상 기본 구조

```
<div class="wrap">
  <header>
    <h1 class="logo">
      <a href="#">
        <!-- # : 임시링크 지정 -->
        
      </a>
    </h1>
    <nav>
      <ul class=""menu_wrap">
        <li>
          <a href="#">메뉴</a>
        </li>
      </ul>
    </nav>
  </header>
</div>
```

2. nav

- <nav></nav>

==> 목차, 메뉴, 리스트 등 다른 페이지로 이동을 위한 링크 공간 표현

※ 통상적으로 header안에 사용

★ nav, main, footer 구상 기본 구조

```
</nav>
</header>
<main>
  <div class="main_01"></div>
  <div class="main_02"></div>
</main>
<footer></footer>
</div>
```

3. aside

- <aside></aside>

==> 간접적으로 문서와 관한 내용을 나타냄

==> 사이드바

4. main

- <main></main>

==> 지배적인 콘텐츠 영역을 나타낼 때 사용

==> html 문서의 주된 내용을 나타낼 때 사용

5. section

- <section></siction>

==> 페이지의 일부분을 지정

==> 같은 테마를 가진 여러개의 콘텐츠를 그룹화

==> 항상 제목이 있는 것이 일반적

6. figure

- <figure></figure>

==> 그림, 차트 등을 감싸는 시멘틱태그

7. footer

- <footer></footer>

=> 바닥글

=> 저작권정보, 사업자 정보, 주소 등

★ 시멘틱태그 예제

```
<!DOCTYPE html>
<html lang="ko">
<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>예제</title>
    <style>
        *{
            margin: 0;
            padding: 0;
            box-sizing: border-box;
        }

        .wrap{
            width: 1200px;
            margin: 0 auto;
        }

        header{
            width: 100%;
            height: 100px;
            background-color: lightgray;
        }

        header > h1{
            text-align: center;
            line-height: 100px;
        }

        nav, aside{
            width: 200px;
            height: 500px;
            background-color: gray;
            float: left;
        }

        .section_1{
            width: 800px;
            height: 500px;
            background-color: lightblue;
            float: left;
        }

        .box{
            width: 400px;
            height: 300px;
            background-color: teal;
            margin: 100px auto;
        }

        footer{
            width: 100%;
            height: 100px;
            background-color: lightgreen;
            clear: both;
        }
    </style>
</head>
<body>

    <div class="wrap">
        <header>
            <h1>header</h1>
        </header>
        <nav>
            <h3>nav</h3>
        </nav>
        <section class="section_1">
            <div class="box"></div>
    
```

```
</section>
<aside>
  <h3>aside</h3>
</aside>
<footer></footer>
</div>
```

- **overflow**

==> 부모요소 밖을 넘쳐나는 자식을 가려줄 때 사용

==> 반드시 부모요소에 적용할 것

- **overflow: scroll;**

==> 부모 밖으로 넘치는 자식을 가려주고 스크롤이 생성되어 자식을 확인할 수 있음

- **overflow: visible;**

==> 부모바깥을 넘치는 자식을 그대로 보여줌

- **overflow: hidden;**

==> 부모바깥을 넘치는 자식을 가려줌

==> 클리핑마스크와 비슷한 효과

※ **hover**효과와 접목해서 사용하는 경우가 많음

※ 속성을 부모태그에 적용

```
.box{
  width: 300px;
  height: 300px;
  border: 10px solid #000;
  overflow: hidden;
}
```

★ .(선택자)box(클래스명).r_box + tab : 이중클래스 생성

- **position** : 레이아웃을 배치하거나 객체를 위치시킬 때 사용할 수 있는 속성

==> top, left, bottom, right 네가지 방향으로 움직임

==> 좌표값으로 position 적용된 객체를 움직일 수 있음

※ **float**은 가로 배치, **position**은 겹쳐줄 때 주로 사용

- **position** 속성

1) **static**

- position 속성의 기본값
- 좌표값의 영향 받지 않음
- 요소를 문서의 흐름에 맞게 배치

★★2) **relative**

- 자신이 원래 있어야 할 자리를 기준으로 상대적으로 위치 값 변경 (상대값)

★★3) **absolute**

- 절대값
- 요소를 기준으로 위치값 변경

- 부모요소가 relative인 태그를 찾아서 계속 올라감
 - 부모요소에 relative가 없다면 브라우저의 원쪽 상단을 기준으로 위치값 변경
 - 문서상에서 독립되어서 앞, 뒤에 나오는 요소와 상호작용 X
 - ※ absolute가 사용되는 요소의 부모값에 relative 필수
 - ※ absolute가 적용되는 요소에 width값 필수
 - absolute 사용시, 요소가 겹칠 수 있음
- ==> 이때, 앞으로 끄내고 싶은 요소가 있다면 z-index 적용
 ==> ex) z-index : 수치
 ※ 보통 99,999 많이 사용

4) fixed

- 고정속성
- 주로 웹 페이지에서 header, top_btn을 고정할 때 사용
- 절대값

5) sticky

- 사용시, 스크롤 필요함
- 스크롤에 반응하여 스티커처럼 붙어서 내려옴

※ position을 이용한 중앙 맞춤

```
position: absolute;
top:50px;
left:50px;
margin-top: -(height/2)px;
margin-left: -(width/2)px;
```

```
.square{
  width: 100px;
  height: 100px;
  background-color: teal;
  position: absolute;
  right: 0;
  bottom: 0;
  color: #fff;
  text-align: center;
  line-height: 100px;
  font-weight: bold;
}

.square:first-child{
  background-color: gold;
  z-index: 1;
  top:50%;
  left: 50%;
  margin-top: -50px;          /* 부모인 square 클래스의 height / 2
                                * 양수값은 현재 위치에서 아래로 반대 위치는 음수 적용
}
```

※ 버튼 생성 방법

```
1. div 사용
<div class="top_btn">
  <a href="#">!</a>
</div>

.top_btn{
  width: 50px;
  height: 50px;
  background-color: salmon;
```

```

border-radius: 50%;
text-align: center;
line-height: 50px;
position: fixed;
bottom: 50px;
right: 50px;
}

2. 버튼 이용
<button class="btn">
    <a href="#">버튼</a>
</button>

.btn{
    border: none;
    width: 50px;
    height: 50px;
    border-radius: 50%;
    position: fixed;
    bottom: 80px;
    right: 50px;
}

3. a 태그 이용
<a href="#" class="btn_top">top</a>

.btn_top{
    display: inline-block;
    padding: 50px 50px;
    background-color: gold;
    border-radius: 50%;
    position: fixed;
    bottom: 50px;
    right: 80px;
}

```

- 웹사이트에 동영상 게시하기

1. iframe

- 유튜브에서 동영상 추출할 때 사용
- [사용방법]

==> 유튜브에서 원하는 동영상 클릭 -> 소스코드 복사 -> vscode에 붙여넣기

※ vscode = visual studio code

※ 동영상 게시자가 동영상을 내리거나 비공개 처리한 경우 재생 불가

2. video

- [사용방법]

```

<video autoplay muted loop>
    <source src="비디오소스">
</video>

```

==> 비디오 태그에 autoplay, muted(음소거), loop(반복) 설정

==> 비디오태그를 원하는 크기에 맞추고 싶을 때는 object-fit : fill / cover 사용

- transition / 변화

==> 객체의 속성을 서서히 변화시키는 속성

==> 객체의 변화가 일정시간 걸쳐서 일어나도록 하는 것

- transition 속성

1) transition-duration

- 변화 지속 시간

==> 단위 : s / ms
==> 1s = 1000ms
==> 값을 지정하지 않는 경우 기본 값이 0s 가 적용되어 시각적으로 변화 확인 불가
==> hover상태에 적용하게 되면 부드럽게 변화가 일어나지 않으므로 반드시 기본값에 적용할 것

2) transition-property

- transition을 적용시킬 속성을 정함
- transition-property : all;

3) transition-timing-function

- transition의 진행 속도를 정함

4) transition-delay

- transition의 시작시간을 연기함

※ transition 촉약

- transition : property duration timing-function delay;

```
.gage{  
    width: 0;  
    height: 100%;  
    background-color: navy;  
    /* text-decoration: 1s;  
    transition-property: width; */  
    transition: all 1s;  
    /*  
  
    transition 촉약  
    transition : property duration timing-function delay;  
  
    */  
}
```

★ 12일차 과제풀이

```
.wrap:hover > .in_box > h3{  
    display: block;  
}  
  
/* .wrap:hover h3{  
    display: block;  
} */
```

==> 자손태그 선택 방법 활용

- transform / 변형

==> 요소 움직임
==> 요소 회전
==> 요소 기울임
==> 요소 크기변경

- transform 속성

1) rotate / 회전

- 단위 : deg
- transform: rotateX(180deg) / rotateX, Y, Z(각도deg)

2) translate / 이동

- 단위 : px
- transform : translate(x,y);

3) skew / 왜곡

- 단위 : deg
- transform : skew(45deg);
- 역방향으로 기울이고 싶을 때는 각도에 - 넣어줌

4) scale / 배율

- transform : scale();
- 원본 : 1
- 원본보다 더 커지길 원하면 1보다 큰 수를 원본보다 더 작아지길 원하면 1보다 작은 수 입력

※ 적용하고 싶은 순서대로 적어서 축약버전 사용 가능

★ 13일 과제 풀이

```
.in_box{
    width: 10%;
    height: 10%;
    background-color: salmon;
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    /* width와 height값이 정확할 경우에는 margin을 이용해서 중앙을 맞추고
       %로 값이 정해져있을 경우 transform: translate(-50%, -50%); 적용 */
}
```

• css animation + keyframe

==> 애니메이션을 나타내는 css 스타일과 애니메이션의 중간상태를 나타내는 keyframe

==> transition보다 규모가 더 크고 복잡하며, 다양한 움직임 구현 가능

• animation 속성

1) animation-name

- 애니메이션 이름 설정
- <유효한 이름>

==> 단순 단어 : name, move 등

==> 언더바 : name_name;

==> 하이픈 : name-name;

• <사용 불가한 이름>

==> 특수문자 : %name, *name

==> 첫글자로 숫자가 나오는 경우 : 1name

※ 애니메이션의 이름이 있어야 호출이 가능하기 때문에 항상 이름을 제일 먼저 적어줄 것!

2) animation-duration

- 애니메이션 재생 지속 시간

- 애니메이션을 한번 재생하는데 걸리는 시간 설정
- transition-duration과 유사한 효과
- animation-duration:0 , animation-duration 적지 않을 시 애니메이션 재생 불가

3) animation-iteration-count

- 애니메이션의 재생 횟수 설정
- 숫자
- 기본값 : 1
- infinite : 애니메이션 무한 반복(가장 많이 사용하는 옵션)

4) animation-direction

- 애니메이션 재생 방향 설정
- ※ from - to => 순방향
to - from => 역방향

4-1) normal

- 기본값 / 애니메이션을 순방향 재생
- 한번 재생이 끝나면 첫번째 프레임부터 다시 시작

4-2) reverse

- 애니메이션 역방향 재생
- 한번 재생이 끝나면 마지막 프레임부터 다시 시작

4-3) alternate

- 0 -> 100 -> 0
- 순방향 시작, 역방향과 순방향을 번갈아가며 재생
- 사용성 가장 높음

4-4) alternate-reverse

- 100 -> 0 -> 100
- 역방향 시작, 순방향과 역방향을 번갈아가며 재생

※ animation 속성

```
animation : name duration (timing-function) infinite direction;
```

※ timing-function 커스텀 마이징

```

<!DOCTYPE html>
<html lang="ko">
  <head> ...
  </head>
  <body>
    <div class="box"></div> == ⑥
    <!--
      - . css animation + keyframe
      --> 애니메이션을 나타내는 css 스타일과 애니메이션의 중간상태를 나타내는 keyframe으로 구성
      --> transition보다 규모가 더 크고 복잡하며, 다양한 움직임 구현 가능

      - . animation 속성
        1. animation-name
        --> 애니메이션 이름 설정

      <유효한 이름>
      - 단순 단어 : name, move 등
      - 언더바 : name_name;
      - 하이픈 : name-name;
    </div>
  </body>
</html>

```

--> 애니메이션을 나타내는 css 스타일과 애니메이션의 중간상태를 나타내는 keyframe으로 구성
--> transition보다 규모가 더 크고 복잡하며, 다양한 움직임 구현 가능

- . animation 속성
1. animation-name
--> 애니메이션 이름 설정

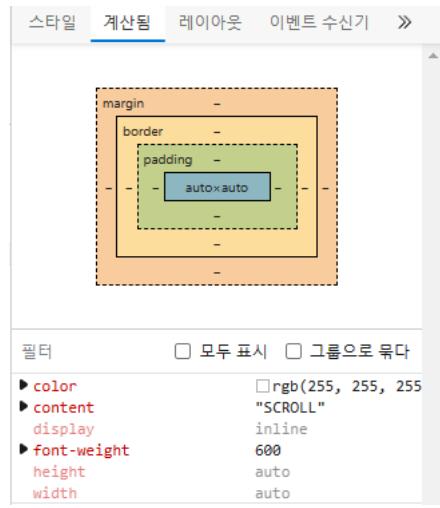
<유효한 이름>
- 단순 단어 : name, move 등
- 언더바 : name_name;
- 하이픈 : name-name;

★레이아웃 예제 기본 구성

header (div, header, h, ul 등 다양하게 활용)	※ 요소 종정렬 방법 1. margin --> margin : top (right/left) bottom; --> margin : (top/bottom) (left/right); --> margin : 0 auto 2. padding --> padding: 30px 0; (위, 아래 30px씩 내부여백을 줌) 3. position --> relative : 자신이 있어야 할 자리 기준으로 상대적으로 위치값 변경 --> absolute : 부모요소가 relative 인 태그를 기준으로 위치값 변경 (width 값 필수) --> position: absolute top:50%; left:50%; margin-top: -(height/2}px; margin-left: -(width/2}px);
section (div, section, h, ul 등 다양하게 활용) 단, ul 태그안에는 li만 사용 가능	※ float은 가로 배치, position은 겹쳐질 때 주로 사용 * margin: 0 auto; 와 float는 같이 사용 불가 * width와 height값이 정확할 경우에는 margin을 이용해서 중앙을 맞추고 %로 값이 정해져있을 경우 transform: translate(-50%, -50%); 적용
footer (div, footer, h, ul 등 다양하게 활용)	※ hover 작용 --> 마우스 팔업 여부 상관없이 부드러운 효과를 주기 hover했을 때 적용이 아닌 해당 태그에 바로 적용
	※ transition 속성 --> transition : property duration timing-function delay;
	※ transform 기본 사용법 --> 부모요소에 반드시 transition-duration 값 필요 --> transform: translate(100px,100px) rotate(360deg) skew(45deg) scale(3) (이동) (회전) (왜곡) (배율) * 순서 상관없이 적힌 순서대로 효과 적용됨
	※ animation 효과 --> animation-name: wheel; (이름) animation-duration: 1s; (재생 지속시간) animation-iteration-count: infinite; (재생 횟수 : 무한대 설정) animation-direction: alternate; (재생 방향 : 순방향 시작, 역방향과 순방향 번갈아가며 재생) animation-timing-function: linear; (속도 : 일정하게 반복)

★ animation 과제풀이

※ 요소의 속성 확인 방법



※ 가상선택자 사용 및 position을 이용한 중앙정렬

```
.icon_mouse{
    width: 30px;
    height: 50px;
    border: 2px solid #fff;
    position: absolute;
    bottom: 50px;
    left: 50%;
    transform: translateX(-50%);
    border-radius: 20px;
    /* 범용성있는 사용을 위해(요소 단독배치를 위해서) */
}

.icon_mouse::before{
    content: "SCROLL";
    color: #fff;
    font-weight: 600;
    position: absolute;
    left: 50%;
    top: -30px;
    transform: translateX(-50%);
}

.ball{
    width: 4px;
    height: 4px;
    background-color: #fff;
    border-radius: 50%;
    position: absolute;
    left: 50%;
    top: 8px;
    transform: translateX(-50%);
    animation: mouse_act 1s linear infinite alternate;
    /* animation : name duration time-function count direction */

    /* animation-name: mouse_act;
    animation-duration: 1s;
    animation-direction: alternate;
    animation-iteration-count: infinite;
    animation-timing-function: linear; */
}
```

※ animation 적용 예시

```

※ 설정 아이콘에 마우스 팝업 시 회전하는 효과 적용
.friend_icon .fa-gear:hover{
    animation: rotateSetting 1s linear infinite;
}

@keyframes rotateSetting {
    from{
        transform: none;
    }
    to{
        transform: rotateZ(360deg);
    }
}

※ 알蛩선 아이콘을 아래에서 위로 올라가면서 Y축 기준으로 회전
.alert{
    width: 30px;
    height: 30px;
    background-color: tomato;
    border-radius: 50%;
    color: #fff;
    display: flex;
    justify-content: center;
    align-items: center;
    font-weight: 600;
    animation: alertRotate 2s ease-in-out infinite;
}

@keyframes alertRotate{
    50%{
        transform: translateY(-5px) rotateY(360deg);
    }
}

※ 채팅창 회면 splash 효과
.splash{
    width: 100vw;
    height: 100vh;
    background-color: var(--mainColor);
    position: absolute;
    top: 0;
    left: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    font-size: 130px;
    /*
        *animation-fill-mode
        => 애니메이션이 끝난 후 처음(from)으로 돌아가느냐, 끝난 상태를 유지하느냐 결정하는 속성
        1) none : 별도의 옵션 지정 x
        2) forwards : 애니메이션 키 프레임이 100%(to)에 도달했을 때 마지막 키프레임 유지
        3) backwards : 애니메이션 스타일을 애니메이션이 시작되기 전의 상태로 적용
    */
    animation: hideSplash 0.3s ease-in-out forwards;
    animation-delay: 0.5s;
}

@keyframes hideSplash {
    from{
        opacity: 1;
    }
    to{
        opacity: 0;
        /* 레이아웃에 영향을 주지 않고 요소를 없애주는 속성 */
        visibility: hidden;
    }
}

※ 아이콘 순차적 팝업
.nav_btn{
    /*
        position: relative;
        top:50px;
    */
    transform: translateY(50px);
    opacity: 0;
    animation: showBtNS .3s ease-in-out forwards;
}

```

```

.nav_btn:nth-child(2){
    animation-delay: .2s;
}

.nav_btn:nth-child(3){
    animation-delay: .5s;
}

.nav_btn:nth-child(4){
    animation-delay: .8s;
}

@keyframes showBtns {
    from{}
    to{
        opacity: 1;
        transform: translateY(0);
    }
}

}

```

- **flex**

=> 요소의 크기가 불분명하거나 동적인 경우에도 각 요소를 정리할 수 있는 효율적인 방법

※ 요소 중앙정렬

1. 텍스트만 정렬이 필요한 경우
`text-align:center / line-height 옵션 사용 ※ height 값이 변할 때마다 수정 필요`
2. float
 요소를 띠 있는 것처럼 만들어서 왼쪽과 오른쪽으로 정렬하는 속성 ※ 레이아웃을 무너뜨릴 수 있음
3. margin
`margin : 0 auto (width값 필수)`
4. padding
`padding 30px 0`
 ※ height 값 대신 위, 아래 30px씩 내부여백 적용
5. position
 부모요소에 `position : relative` 적용 (단, 요소를 독립적으로 배치하고 싶을 경우 생략가능하나 웹안해선 부모 요소에 값 적용해줄 것)
 ※ absolute가 사용되는 요소의 부모값에 `relative` 필수
 ※ absolute가 적용되는 요소에 `width값 필수`
`position: absolute
 top:50px;
 left:50px;
 margin-top: -(height/2)px;
 margin-left: -(width/2)px;`
`position: absolute;
 top: 50%;
 left: 50%;
 transform: translate(-50%, -50%);
 /* width와 height값이 정확할 경우에는 margin을 이용해서 중앙을 맞추고
 %로 값이 정해져있을 경우 transform: translate(-50%, -50%); 적용 */
}`
6. flex
`display : flex`
`justify-content : center`
`align-items:center`

1. margin : width 값 필수, 수평 정렬만 가능한 단점

```
<html>
<style>
.wrap{
    width: 100px;
    margin: 0 auto;
```

2-1. flex 를 이용한 수평 중앙 정렬

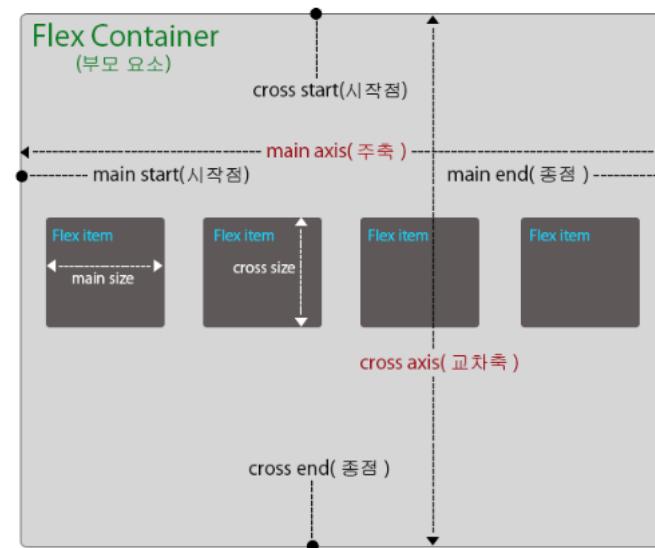
```
<html>
<style>
.wrap{
    display: flex;
    justify-content: center;
```

2-2. flex 를 이용한 수평 수직 정렬

```
<html>
<style>
.wrap{
    height: 100%;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
```

Flexible BOX

(플렉서블 박스)



row	텍스트의 자식 요소와 같은 방향으로, 일반적으로 flex 아이템은 왼쪽에서 오른쪽으로 수평으로 배치됩니다. 초기값.
row-reverse	row 속성값의 반대로, flex 아이템은 오른쪽에서 왼쪽으로 수평 방향으로 배치됩니다.
column	flex 아이템은 위에서 아래로 수직 방향으로 배치됩니다.
column-reverse	flex 아이템은 아래에서 위로 수직 방향으로 배치됩니다.

- **flex 사용 필요 요소**

==> flex-container(부모요소) / flex-item(자식요소)

- **flex 속성**

1. **flex-container에 적용하는 속성**

- 1-1) **display:flex**

- block요소라서 밑으로 여백을 가지고 떨어지던 아이템들이 각 내용물의 width만큼 가지고 가로로 한줄 정렬됨

- 1-2) **flex-direction / 배치 방향 설정**

- 아이템들이 배치되는 축의 방향 설정

==> row(기본값) : 아이템들을 가로(행) 배치

==> row-reverse : 아이템들을 역순으로 가로배치

==> column : 아이템들을 세로(열) 배치

==> column-reverse : 아이템들을 역순 세로배치

`flex-direction: row;`

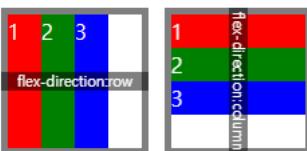


`flex-direction: column`

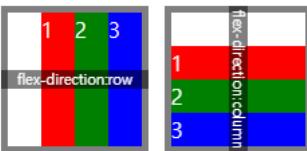


★ **flex-direction**

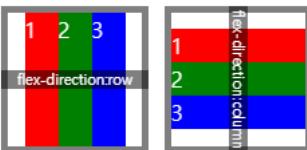
justify-content: flex-start // default



justify-content: flex-end



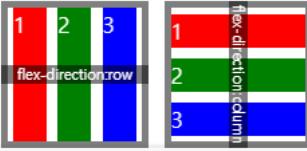
justify-content: center



justify-content: space-between



justify-content: space-around



1-3) flex-wrap / 줄넘김 처리 설정

- 컨테이너가 더 이상 아이템들을 한 줄에 담을 여유공간이 없을 때 줄바꿈을 어떻게 할지 결정하는 속성
- nowrap : 줄바꿈 안함(기본값)
- wrap : 줄바꿈함 (float으로 배치한 요소와 비슷하게 움직임)
- wrap-reverse : 줄바꿈 하고 아이템들을 역순 배치

※ flex의 경우 flex_container에 flex-wrap: wrap; 옵션을 적용해줘야 자식의 요소에 고유 크기값을 적용할 수 있다 (기본적으로 부모의 크기값을 벗어나지 않으려고 한다)

DevTools - 127.0.0.1:5500/15%EC%9D%BC%EC%B0%A8/flex.html

프로젝트의 루트 폴더를 식별하여 Visual Studio Code 소스 파일을 열고 변경 내용을 동기화합니다.

루트 폴더 설정 다시 표시 안 함

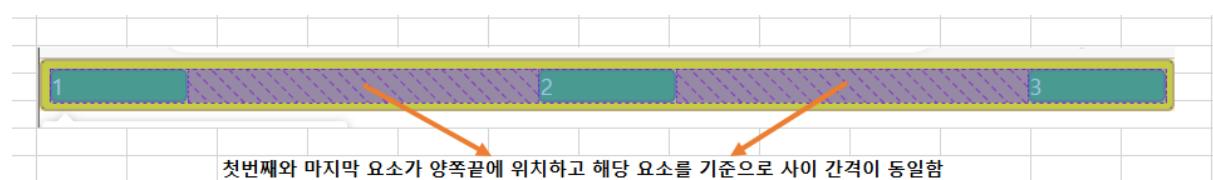
스타일 계산됨 레이아웃 이벤

필터 :hover .cls

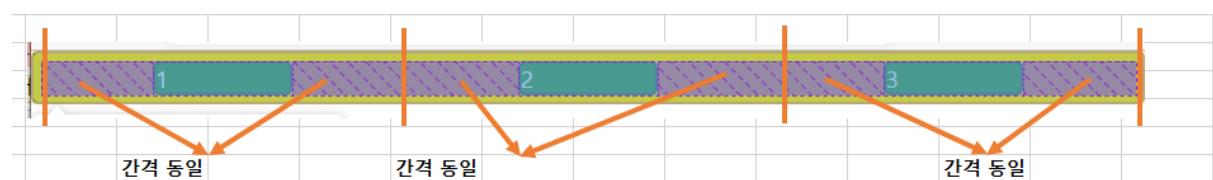
```
element.style {  
}  
.flex_item {  
    width: 200px;  
    background-color: green;  
    border: 2px solid #333;  
    border-radius: 5px;  
    color: #fff;  
}
```

1-4) justify-content / 가로축 방향 정렬

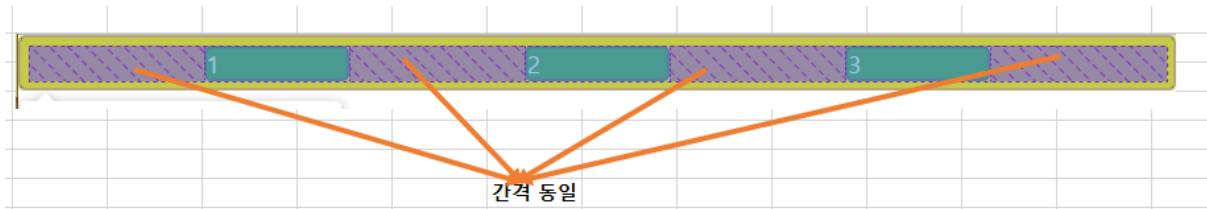
- flex-start : 아이템들을 flex 시작점으로 정렬
※ flex-direction이 column일 때는 위, row일 때는 왼쪽이 start
- flex-end : 아이템들을 flex 끝 지점으로 정렬
※ flex-direction이 column일 때는 아래, row일 때는 오른쪽이 start
- center : 아이템들을 가운데로 정렬
- space-between : 아이템들 사이에 동일한 간격



- space-around : 아이템들 둘레에 동일한 간격



- space-evenly : 아이템들의 사이와 양끝에 동일한 간격



★★ justify-content 적용 안될 경우 부모요소(container)에 width 값 적용 유무 확인

1-5) align-items / 세로축 방향 설정

- stretch : 아이템들이 수직축 방향으로 끝까지 늘어남(기본값)
- flex-start : 아이템들을 시작점으로 배치

※ flex-direction : row = 위

※ flex-direction : column = 왼쪽

- flex-end : 아이템들을 끝점으로 배치

※ flex-direction : row = 아래

※ flex-direction : column = 오른쪽

- center : 아이템들을 세로 중앙 정렬
- baseline : 아이템들을 텍스트 베이스라인 기준으로 정렬함



★★ align-items 적용 안될 경우 부모요소(container)에 height 값 적용 유무 확인

※ flex에 가장 많이 적용하는 3종 속성

```
.flex_container{
    height: 400px;
    background-color: #gold;
    border: 2px solid #333;
    border-radius: 5px;
    padding: 5px;
    display: flex;
    flex-direction: row;
    justify-content: space-between;
    align-items: center;
}

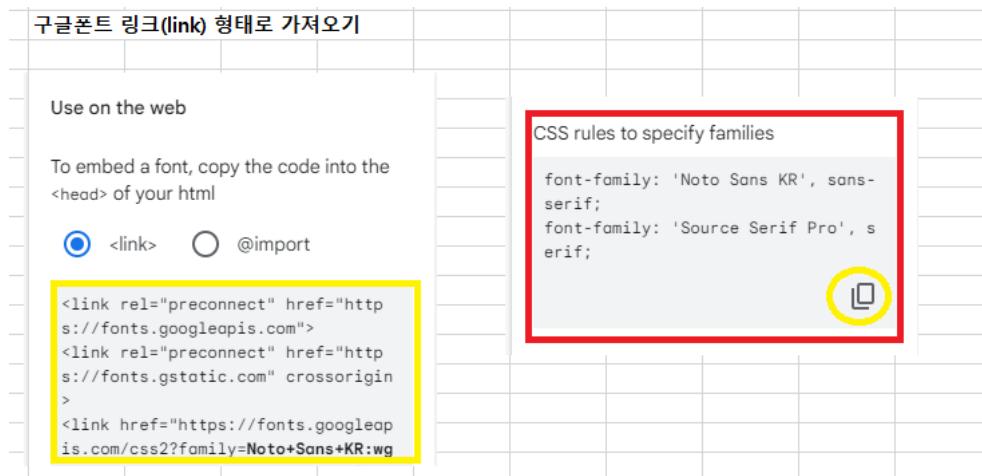
.flex_item{
    width: 100px;
    background-color: #green;
    border: 2px solid #333;
    border-radius: 5px;
    color: #fff;
}
```

★ index(방앗간 사이트 만들기) 과제 풀이

```
.inner{  
    max-width: 1280px;  
    width: 90%;  
    height: 100%;  
    margin: 0 auto;  
    background-color: #gray;  
    display: flex;  
    align-items: center;  
    justify-content: space-between;  
}
```

※ 웹 폰트 적용

```
<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">  
    <link rel="preconnect" href="https://fonts.googleapis.com">  
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>  
    <link href="https://fonts.googleapis.com/css2?family=Noto+Sans+KR:wght@100;300;400;500;700;900&family=Source+Serif+Pro:ital,wght@0,200;0,300;0,400;0,600;0,700;0,900" href="https://fonts.googleapis.com/css2?family=Noto+Sans+KR:wght@100;300;400;500;700;900&family=Source+Serif+Pro:ital,wght@0,200;0,300;0,400;0,600;0,700;0,900" type="font-style">  
    <title>방앗간화장품</title>
```



※ 속성 통합 적용

```
:root{  
    --whiteColor : #fff;  
    --grayColor : #f8f8f8;  
    --kor : 'Noto Sans KR';  
    --eng : 'Roboto'  
}
```

```
.section_2{  
    width: 100%;  
    height: 900px;  
    background-color: var(--grayColor);  
    padding-top: 150px;  
}
```

※ background 축약

```
.section_4_bg{  
    width: 100%;  
    height: 500px;  
    background: url(img/section_4_img.jpg) no-repeat 100% / cover ;  
    /*  
     *  
     */  
}
```

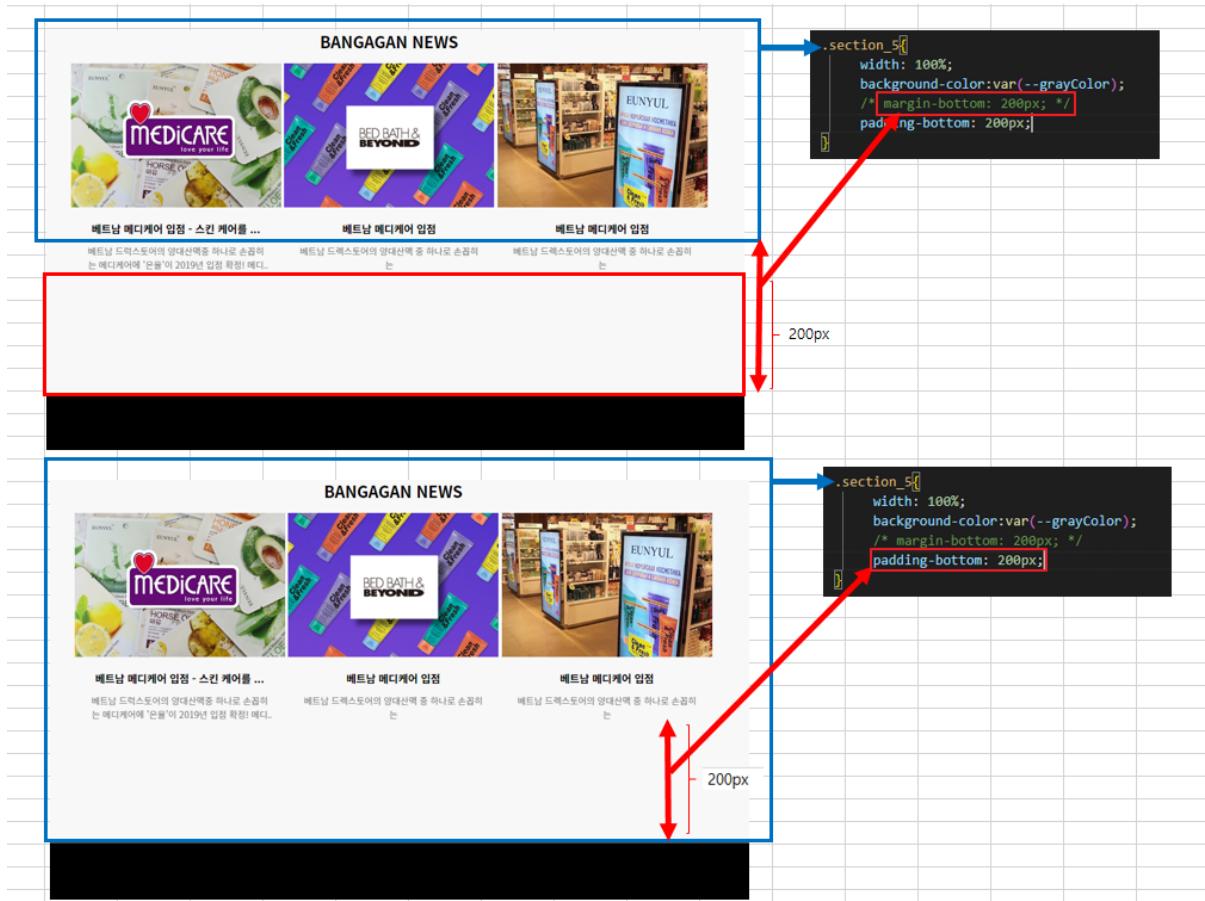
※ box 그림자 효과

```
.section_4_iframe{  
    padding: 60px;  
    background-color: var(--whiteColor);  
    box-shadow: 8px 8px 12px rgba(0, 0, 0, .1);  
    /*  
     *  
     */  
}
```

※ 넘치는 텍스트 처리

```
.n_wrap_dt{  
    font-size: 20px;  
    font-weight: 600;  
  
    text-overflow: ellipsis;  
    /* 넘치는 텍스트들을 말줄임표 원으로 바꿈 */  
    white-space: nowrap;  
    /* 자동 줄바꿈 처리 */  
    overflow: hidden;  
    margin-bottom: 15px;  
}  
  
.n_wrap_dd{  
    font-size: 17px;  
    opacity: .6;  
  
    text-overflow: ellipsis;  
    overflow: hidden;  
    display: -webkit-box;  
    /* 해당 영역을 box 형태로 관리함 */  
    -webkit-line-clamp: 2;  
    /* 영역 내의 컨텐츠의 최대 라인 수 결정 */  
    -webkit-box-orient: vertical;  
    /* 영역 내 정렬을 수직으로 함 */  
}
```

※ margin 과 padding 의 차이



※ 방앗간 과제 마크업 설명 #1

마크업 #1 (header ~ section_4)

1. 전체를 감싸는 박스(wrap) ★ 웹 표준 준수

1-1. header

- 상단메뉴 고정 : position:fixed;
- 공통영역 지정

```
.inner{  
    max-width: 1200px;  
    width: 96%;  
    margin: 0 auto;  
  
    /* ----- 골통 영역 ----- */  
  
header > .inner{  
    /* 골통속성을 적용하되 특정영역에만 지정하고 싶을 경우 */  
}
```

1-2. section_1

2. section_2

- Flex 요소를 사용하기 위해서는 내부 요소를 감싸는 박스(wrap) 필요
- 이미지가 삽입되는 요소는 DIV, LI 등의 태그 생성 후 자식태그로 이미지 삽입

```
<section class="section_2">  
    <div class="inner">  
        <p class="main_txt">  
            welcome to BANGAGAN  
            <!-- 단언말표 : b / 결코문 : strong -->  
        </p>  
        <div class="con_box_wrap">  
            <div href="#">  
                  
                <p class="con_txt">  
                    <b>EUNYUL</b>  
                    베트남 전통 환경 친화적인 화장품 브랜드  
                    <br><br>  
                    <b>CHALDOUK</b>  
                    전통의 힘과 현대의 기술이 만나 세련된 화장품  
                    <br><br>  
                    <b>CLEAN & FRESH</b>  
                    청량하고 깨끗한 화장품 브랜드  
                </p>  
            </div>  
            <div href="#">  
                  
                <p class="con_txt">  
                    <b>MAYBONA</b>  
                    베트남 환경 친화적인 화장품 브랜드  
                    <br><br>  
                    <b>YARNOOK</b>  
                    천연 속 깊은 자연에서 영감을 받은  
                    화장품 브랜드  
                </p>  
            </div>  
        </div>  
    </div>  
</section>
```

3. 이미지 선택 시 불투명 효과 적용

```
.center_box img{  
    display: block;  
    width: 100%;  
    transition: all .5s;  
  
.center_box:hover img{  
    opacity: .5;  
}
```

4. section_4

- 겹침 효과를 위해 값을 -값으로 적용

```
.section_4 > .inner{  
    margin-top: -350px;  
}
```

※ 방앗간 과제 마크업 설명 #2

마크업 #2 (section_5 ~ footer)

```

5. section_5
- 넘지는 텍스트 설정
.n.wrap dt{
    font-size: 20px;
    font-weight: 600;

    text-overflow: ellipsis;
    /* 넘지는 텍스트를 말줄일표 원으로 바꿈 */
    white-space: nowrap;
    /* 자동 줄바꿈 처리 */
    overflow: hidden;
    margin-bottom: 15px;
}

.n.wrap dd{
    font-size: 17px;
    opacity: .6;

    text-overflow: ellipsis;
    overflow: hidden;
    display: -webkit-box;
    /* 헤더 영역을 box 형태로 관리함 */
    -webkit-line-clamp: 2;
    /* 영역 내의 컨텐츠의 최대 라인 수 결정 */
    -webkit-box-orient: vertical;
    /* 영역 내 정렬을 수직으로 함 */
}

```

6. footer

6-1.

```

footer > .inner{
    display: flex;
    justify-content: space-between;
}

```

★ 방앗간 과제 마크업 구조 (css)

```

<!DOCTYPE html>
<html lang="ko">
<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">
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css?family=Noto+Sans+KR:wght@100;300;400;500;700;900&family=Source+Serif+Pro:ital,wght@0,0;1,0;2,0;300,0;400,0;600,0;700,0;900,1,200;1,300;1,400;1,600;1,700;1,900&display=swap" rel="stylesheet">
<title>방앗간화장품</title>
<style>
    *{
        margin: 0;
        padding: 0;
        box-sizing: border-box;
        font-family: 'Noto Sans KR', 'Roboto';
    }

    :root{
        --whiteColor : #fff;
        --grayColor : #f8f8f8;
        --kor : 'Noto Sans KR';
        --eng : 'Roboto';
    }

    ul{
        list-style: none;
    }

    a{
        text-decoration: none;
        color: initial;
    }

    /* ----- 초기화 영역 ----- */

    .inner{
        max-width: 1200px;
        width: 90%;
        margin: 0 auto;
    }

    /* ----- 공통 영역 ----- */

```

```

header{
    width: 100%;
    height: 100px;
    position:fixed ;
    top: 0;
    left: 0;
}

header > .inner{
/* 공통속성을 적용하되 특정영역에만 지정하고 싶을 경우 */
    max-width: 1200px;
    width: 90%;
    height: 100%;
    margin: 0 auto;
    display: flex;
    align-items: center;
    justify-content: space-between;
}

.menu_wrap{
    width: 300px;
    display: flex;
    justify-content: space-between;
/* width 값 필수 */
}

/* .menu_wrap > li{
    margin-left: 20px;
} */

/* ----- header_end ----- */

.section_1{
/* height or padding 값 필수 */
    width: 100%;
    height: 830px;
    background-image: url(img/main_visual.jpg);
    background-repeat: no-repeat;
    background-position: center;
    background-size: cover;
}

/* ----- section_1_end ----- */

.section_2{
    width: 100%;
    height: 900px;
    background-color: var(--grayColor);
    padding-top: 150px;
}

.main_txt{
    font-size: 30px;
    text-align: center;
    /* div 내부에 텍스트가 존재할 경우 flex 사용 추천 */
    text-transform: uppercase;
    /* 소문자 변환 : lowercase
       대문자 변환 : uppercase
    */
    margin-bottom: 20px;
}

.con_box_wrap{
    width: 100%;
    display: flex;
    flex-wrap: wrap;
}

.con_box{
    width: 50%;
    padding: 10px;
    position: relative;
}

.con_box img{
    display: block;
    width: 100%;
}

.con_box dl{
    position: absolute;
    bottom: 30px;
    left: 30px;
    color: #fff;
}

.con_box dt{

```

```

        width: 100%;
        font-size: 25px;
        font-weight: 700;
        text-transform: uppercase;
        font-family: var(--eng);
    }

    .con_box dd{
        width: 100%;
        font-size: 15px;
    }

    /* ----- section_2_end ----- */

    .section_3{
        width: 100%;
        padding: 50px 0 100px;
    }

    .center_box_wrap{
        width: 100%;
        display: flex;
        flex-wrap: wrap;
        justify-content: space-between;
    }

    .center_box{
        width: 33%;
        margin-bottom: 30px;
    }

    .center_box > a{
        display: flex;
        flex-direction: column;
        justify-content: center;
        align-items: center;
    }

    .center_box dl{
        text-align: center;
    }

    .center_box dt{
        font-size: 21px;
        font-weight: bold;
    }

    .center_box dd{
        opacity: .7;
    }

    .center_box img{
        display: block;
        width: 100%;
        transition: all .5s;
    }

    .center_box:hover img{
        opacity: .5;
    }

    /* ----- section_3_end ----- */

    .section_4{
        width: 100%;
        /* margin-bottom: 200px; */
        padding-bottom: 200px;
    }

    .section_4_bg{
        width: 100%;
        height: 500px;
        background: url(img/section_4_img.jpg) no-repeat 100% / cover ;
        /*
        background : url() repeat position / size;
        */
    }

    .section_4 > .inner{
        margin-top: -350px;
    }

    .section_4 .main_txt{
        color: var(--whiteColor);
    }

```

```

.section_4 iframe{
    padding: 60px;
    background-color: var(--whiteColor);
    box-shadow: 8px 8px rgba(0, 0, 0, .1);
    /*
        box-shadow : Xpx Ypx color
    */
}
/* ----- section_4_end ----- */

.section_5{
    width: 100%;
    background-color: var(--grayColor);
}

.section_5 .main_txt{
    font-family: var(--eng);
}

.n_wrap{
    width: 100%;
    display: flex;
    /* justify-content 속성을 적용하기 위해선 반드시 width 값 필요 */
    /* 단순히 가로 배치만 한다면 width 값 생략가능 */
    justify-content: space-between;
}

.n_box{
    width: 33%;
}

.n_wrap img{
    display: block;
    width: 100%;
}

.n_wrap dl{
    padding: 25px;
    text-align: center;
}

.n_wrap dt{
    font-size: 20px;
    font-weight: 600;

    text-overflow: ellipsis;
    /* 넘치는 텍스트들을 말줄임표 원으로 바꿈 */
    white-space: nowrap;
    /* 자동 줄바꿈 처리 */
    overflow: hidden;
    margin-bottom: 15px;
}

.n_wrap dd{
    font-size: 17px;
    opacity: .6;

    text-overflow: ellipsis;
    overflow: hidden;
    display: -webkit-box;
    /* 해당 영역을 box 형태로 관리함 */
    -webkit-line-clamp: 2;
    /* 영역 내의 컨텐츠의 최대 라인 수 결정 */
    -webkit-box-orient: vertical;
    /* 영역 내 정렬을 수직으로 함 */
}

/* ----- section_5_end ----- */

footer{
    width: 100%;
    background-color: #1c1c1c;
    padding: 50px 0;
}

footer > .inner{
    display: flex;
    justify-content: space-between;
}

.footer_left_wrap{
    width: 75%;
    font-size: 14px;
    /* 폰트를 별도로 지정하지않으면 16px 정도 */
    color: var(--whiteColor);
}

```

```

        opacity: .8;
    }

.footer_right_wrap{
    width: 15%;
    display: flex;
    justify-content: space-between;
    align-items: flex-end;
}

/* ----- footer_end ----- */

.top_btn{
    display: inline;
    width: 45px;
    height: 45px;
    background-color: salmon;
    color: var(--whiteColor);
    display: flex;
    justify-content: center;
    align-items: center;
    position:fixed;
    bottom: 100px;
    right: 70px;
}

```

</style>

★ 방앗간 과제 마크업 구조 #2 (html)

```

<body>
    <div class="wrap">
        <header>
            <div class="inner">
                <h1 class="logo">
                    <a href="#">
                        
                    </a>
                </h1>
                <ul class="menu_wrap">
                    <li>
                        <a href="#">BRAND STROY</a>
                    </li>
                    <li>
                        <a href="#">PRODUCT</a>
                    </li>
                    <li>
                        <a href="#">NEWS</a>
                    </li>
                    <li>
                        <a href="#">MEDIA</a>
                    </li>
                </ul>
            </div>
        </header>
        <!-- header_end -->

        <section class="section_1"></section>
        <!-- 태그명.(선택자)선택자명 + tab -->
        <!-- section_1_end -->

        <section class="section_2">
            <div class="inner">
                <p class="main_txt">
                    welcome to <b>bangagan</b>
                    <!-- 단순강조 : b / 경고문 : strong -->
                </p>
                <div class="con_box_wrap">
                    <div class="con_box">
                        <a href="#">
                            
                            <p class="con_txt">
                                <dl>
                                    <dt>eunyul</dt>
                                    <dd>자연에서 찾은 천연 원료의 힘 건강한 아름다움</dd>
                                </dl>
                            </p>
                        </a>
                    </div>
                    <div class="con_box">
                        <a href="#">
                            
                            <p class="con_txt">
                                <dl>

```

```

        <dt>maybona</dt>
        <dd>평범한 일상을 특별하게 만들어 줄 당신만의 브랜드</dd>
    </dl>
    </p>
</a>
</div>
<div class="con_box">
    <a href="#">
        
        <p class="con_txt">
            <dl>
                <dt>chaldduk</dt>
                <dd>선조들의 피부 관리 노하우에서 영감을 얻다</dd>
            </dl>
        </p>
    </a>
</div>
<div class="con_box">
    <a href="#">
        
        <p class="con_txt">
            <dl>
                <dt>clean & fresh</dt>
                <dd>면경은 지우고 개성은 살려 나를 가장 나답게!</dd>
            </dl>
        </p>
    </a>
</div>
</div>
</div>
</section>
<!-- section_2_end -->
<section class="section_3">
    <div class="inner">
        <p class="main_txt">
            new <b>items</b>
        </p>
        <div class="center_box_wrap">
            <div class="center_box">
                <a href="#">
                    
                    <dl>
                        <dt>나이아신아마이드 솔루션 시트 마스크</dt>
                        <dd>피부를 깔끔하게 케어</dd>
                    </dl>
                </a>
            </div>
            <div class="center_box">
                <a href="#">
                    
                    <dl>
                        <dt>히알루로닉 솔루션 시트 마스크</dt>
                        <dd>피부를 촉촉하게!</dd>
                    </dl>
                </a>
            </div>
            <div class="center_box">
                <a href="#">
                    
                    <dl>
                        <dt>햄프시드 솔루션 시트 마스크</dt>
                        <dd>피부 보호막 형성</dd>
                    </dl>
                </a>
            </div>
            <div class="center_box">
                <a href="#">
                    
                    <dl>
                        <dt>콜라겐 솔루션 시트 마스크</dt>
                        <dd>피부의 탄력 저하 개선</dd>
                    </dl>
                </a>
            </div>
            <div class="center_box">
                <a href="#">
                    
                    <dl>
                        <dt>알로에 베라 솔루션 시트 마스크</dt>
                        <dd>민감한 피부를 진정!</dd>
                    </dl>
                </a>
            </div>
            <div class="center_box">
                <a href="#">
                    
                    <dl>

```

```

        <dt>벌꿀 마스크팩</dt>
        <dd>피부를 윤기있고 건강하게</dd>
    </dl>
</a>
</div>
</section>
<!!-- section_3_end -->

<section class="section_4">
<div class="section_4_bg"></div>
<div class="inner">
    <p class="main_txt">bangagan's movie</p>
    <iframe width="100%" height="650" src="https://www.youtube.com/embed/QJPk1K1QF-U" title="NewJeans (뉴진스) &#39;Ditto&#39; MV" alt="NewJeans (뉴진스) 'Ditto' MV iframe">
</div>
</section>
<!!-- section_4_end -->

<section class="section_5">
<div class="inner">
    <p class="main_txt"><b>bangagan news</b></p>
    <div class="n_wrap">
        <div class="n_box">
            <a href="#">
                
            <dl>
                <dt>베트남 메디케어 입점 - 스킨 케어를 ...</dt>
                <dd>
                    베트남 드렉스토어의 양대산맥 중 하나로 손꼽히는 메디케어에 '온울'이 2019년 입점 확정! 메디케어는 2001년에 설립된 베트남 최초의 ...
                </dd>
            </dl>
        </a>
    </div>
    <div class="n_box">
        <a href="#">
            
        <dl>
            <dt>베트남 메디케어 입점</dt>
            <dd>베트남 드렉스토어의 양대산맥 중 하나로 손꼽히는</dd>
        </dl>
    </a>
    </div>
    <div class="n_box">
        <a href="#">
            
        <dl>
            <dt>베트남 메디케어 입점</dt>
            <dd>베트남 드렉스토어의 양대산맥 중 하나로 손꼽히는</dd>
        </dl>
    </a>
    </div>
</div>
</section>
<!!-- section_5_end -->

<footer>
<div class="inner">
    <div class="footer_left_wrap">
        <b>Company</b>Bangagan Cosmetics Co., ltd.
        <b>Representative</b>Im Young Sik
        <b>Corporate business number</b>220-87-75028[사업자정보확인하기]
        <b>Online business number</b>제 2016-경기시흥-0089호
        <b>Privacy officer</b>Bangagan cosmetic
        <b>E-mail</b>bangagan@bangacos.com
        <b>Address</b>36-5, Geumwha-ro, 627 beon-gil, Siheung-si, Gyeonggi-do, South Korea
        Copyright © Bangagan Cosmetics Co., ltd.. All Rights Reserved.
    </div>
    <div class="footer_right_wrap">
        <figure>
            
        </figure>
        <figure>
            
        </figure>
        <figure>
            
        </figure>
    </div>
</div>
</footer>
<!!-- footer_end -->

<a href="#" class="top_btn" >↑</a>
</div>
</body>
</html>

```

• input

==> 사용자 입력 태그
==> 회원가입, 설문조사, 검색창 등등..
==> inline 요소
==> 단독태그

```
TEXT : <input type="text"/>
<!-- 한줄의 기본 텍스트 박스 -->
<br/>

COLOR : <input type="color"/>
<!-- 컬러 픽업창 생성 -->
<br/>

SEARCH : <input type="search" placeholder="검색어를 입력하세요."/>
<!-- 검색어 입력창 -->
<!-- placeholder : 미리보기, 텍스트, 글자를 미리 넣어두고 글을 적으면 사라짐 -->
<input type="submit" value="검색"/>
<!-- 양식을 제출할때 사용(로그인, 검색창, 게시판 등..) -->
<!-- value : 미리 설정해놓은 값, 클릭 및 입력해도 글자가 지워지지않고 살아있음 -->
<br/>

TEL : <input type="tel">
<!-- 전화번호 입력창 -->
<br/>

PW : <input type="password">
<!-- 비밀번호 입력창 -->
<br/>

E-MAL : <input type="email">
<!-- 이메일 입력창 -->
<br/>

<textarea cols="30" rows="10"></textarea>
<!-- 여러줄의 텍스트 박스 -->
<br/>

<input type="time">
<!-- 시간 출력 -->
<input type="date">
<!-- 날짜 출력 -->
<input type="week">
<!-- 주 단위 표현 -->
<br/>

<select>
    <option>1번옵션</option>
    <option>2번옵션</option>
    <option>3번옵션</option>
</select>
<br/>

<input type="email" placeholder="이메일 아이디를 입력하세요">@
<select>
    <option>--이메일 주소 선택--</option>
    <option>naver.com</option>
    <option>gmail.com</option>
    <option>daum.net</option>
</select>

<br/><br/>

<h4>설문조사</h4>
<p>수업 만족도 참여 조사</p>
<input type="radio" name="search" id="tab_1">
<label for="tab_1">만족</label>
<!-- 반드시 id로 연결(class 연결시 동작 x) -->

<input type="radio" name="search" id="tab_2">
<label for="tab_2">보통</label>

<input type="radio" name="search" id="tab_3">
<label for="tab_3">불만족</label>
<!--
    - . radio 버튼
    ==> 한번 체크되면 해제 불가
    ==> 설문조사에 주로 쓰임
    ==> 여러개 중에서 하나만 선택하게 만들고 싶다면 그룹 작업 필수
    - . name="" : radio버튼을 하나의 그룹으로 묶어주는 역할
-->
```

```

    . label 태그
    ==> input 태그와 연결해서 사용
    ==> input과 연결해주기 위해서 id 사용
    *class는 사용 불가
    ==> <input id="text_1"/>
    ==> <label for="text_1"></label>
-->
<br/><br/>

<h4>사이드 메뉴</h4>
<input type="checkbox" id="menu_1"/>
<label for="menu_1">사이디</label>

<input type="checkbox" id="menu_2"/>
<label for="menu_2">클라</label>

<input type="checkbox" id="menu_3"/>
<label for="menu_3">환타</label>
<!--
    . check box
    ==> 선택 후 해제 가능
    ==> 다중옵션 선택 시 사용
    ==> name으로 그룹화 작업해도 사용 불가
-->

```

※ input #1 (TEXT ~ select)

The screenshot shows a code editor with several examples of different input types:

- TEXT :** TEXT : <input type="text"/>
- COLOR :** COLOR : <input type="color"/>
- SEARCH :** 검색어를 입력하세요. 검색 SEARCH : <input type="search" placeholder="검색어를 입력하세요."/> <input type="submit" value="검색"/>
- TEL :** TEL : <input type="tel"/>
- PW :** PW : <input type="password"/>
- E-MAL :** E-MAL : <input type="email"/>
- TEXTAREA :** <textarea cols="30" rows="10"></textarea>
- DATE/TIME/WEAK :** <input type="time"> <input type="date"> <input type="week">
- Dropdown:** -- --:-- (Clock icon) 년-월-일 (Calendar icon) -- 주, ---- (Calendar icon) 1번옵션 ▾
- Email Input:** 이메일 아이디를 입력하세요 @ ---이메일 주소 선택--- ▾
- Select Example:** <select> <option>1번 옵션</option> <option>2번 옵션</option> <option>3번 옵션</option> </select>
- Complex Email Input:** <input type="email" placeholder="이메일 아이디를 입력하세요">@ <select> <option>---이메일 주소 선택----</option> <option>naver.com</option> <option>gmail.com</option> <option>daum.net</option> </select>

※ input #2 (radio버튼 ~ check box)

설문조사
수업 만족도 참여 조사
<input type="radio"/> 만족 <input type="radio"/> 보통 <input type="radio"/> 불만족
사이드 메뉴
<input type="checkbox"/> 사이다 <input type="checkbox"/> 콜라 <input type="checkbox"/> 환타

```
<h4>설문조사</h4>
<p>수업 만족도 참여 조사</p>
<input type="radio" name="search" id="tab_1">
<label for="tab_1">만족</label>
<!-- 반드시 id로 연결(class 연결시 동작 x) --&gt;

&lt;input type="radio" name="search" id="tab_2"&gt;
&lt;label for="tab_2"&gt;보통&lt;/label&gt;

&lt;input type="radio" name="search" id="tab_3"&gt;
&lt;label for="tab_3"&gt;불만족&lt;/label&gt;</pre>

```

```
<h4>사이드 메뉴</h4>
<input type="checkbox" id="menu_1"/>
<label for="menu_1">사이다</label>

<input type="checkbox" id="menu_2"/>
<label for="menu_1">콜라</label>

<input type="checkbox" id="menu_3"/>
<label for="menu_1">환타</label>
```

※ class 이름이 지정되지 않은 input 설정

```
.search_form > input[type="text"]{
    border: none;
    padding: 8px 0;
    border-radius: 0.3rem;
    font-size: 0.8rem;
}

.search_form > input[type="submit"]{
    border: none;
    background-color: gold;
    width: 4rem;
    padding: 0.5rem 0;
    border-radius: 0.3rem;
    cursor: pointer;
}
```

[웹 컴바인 #1 총정리]

[글자 관련 태그]

- 제목태그
=> h1 ~ h6
=> h1 가장 큼, h6 가장 작음
=> h1 는 logo 담아줄 때 주로 사용
- 문단태그
=> p
=> 문단에서 줄바꿈
,

- span
=> inline요소
=> width, height, margin 적용 불가
=> 부분적으로 꾸밈요소를 줄때 주로 사용
=> 줄바꿈 없이 요소를 정렬하고 싶을 때

- **strong / b**

=> strong >>> b

=> strong : 경고 및 주의사항

=> b : 단순강조

- **i, em**

=> 글자 기울임

=> css) font-style : italic; 대체 가능

- **mark**

- **small**

[display / 태그의 성질, 요소]

- **block**

=> 여백값, 줄바꿈 O

=> div

=> 시멘틱태그

- **inline**

=> 여백값, 줄바꿈 X

=> span

=> block 요소의 우선 순위가 inline 요소보다 높음

=> inline > inline O

=> block > block O

=> p > p X

- **inline-block**

=> inline + block

=> inline 요소처럼 한 줄로 태그가 전개, block 요소처럼 width, height 적용 가능

=> 불필요(알수 없는 값)한 여백이 생기는 문제로 레이아웃 작성에는 추천하지 않음

- **flex**

=> container, item

=> container = 부모, item = 자식

=> 부모에게 적용하는 속성, 자식에게 적용하는 속성

=> 부모요소에게 display:flex;

=> 가로축 : justify-content

★ 간혹 space-between 적용 안되는 경우 flex-container에 width값 있는지 확인할 것! ★

=> 세로축 : align-items

★ 간혹 align-item:center 적용 안되는 경우 flex-container에 height값 있는지 확인할 것! ★

[이미지 관련]

- **img tag**

=>

=> 단독태그

=> inline

=> 브라우저의 크기를 줄여도 자신의 크기값 유지함

- **background-image / css**

=> 필수 조건 : height , padding

=> background : url() repeat position / size;
=> backgruond : url() no-repaet top center;
※ 사이즈가 생략된 경우 가져온 원본이미지 사이즈 유지

[video]

1. 유튜브 / iframe

2. video, source

```
<video autoplay muted loop>
    <source src="비디오소스">
</video>
object-fit : fill / cover 사용
```

[position]

=> 좌표 : top, left, right, bottom

1. 상대값

- **relative**

=> 본인 위치를 기준으로 움직임

2. 절대값

- **absolute**

=> 부모요소에 relataive가 있는 요소를 기준으로 움직임

=> 만약, 부모요소에 relataive 가 없는 경우 브라우저 최상단을 기준으로 움직임

=> 위, 아래 요소들과 상호작용 X

=> 문서상에서 독립되어 있음

- **fixed**

=> top_btn, header 고정

=> 브라우저 전역을 기준으로 함

- **sticky**

=> 윈도우의 스크롤 값에 반응하여 고정되는 속성

[변화, 변형 속성]

- **transition**

=> 변화를 일정시간에 걸쳐서 일어나게 만듬

- **transition-duration : 1s;**

transition-duration : 1000ms; (1s = 1000ms)

- **transition-property : all;**

- **timing-function : ease(기본값) / linear / cubic-bezier**

==> transition : property duration timing-function delay;

- **transform / 변형**

=> 객체를 기울이거나 왜곡하거나 배율을 조정하는 등 모형을 변형할 때 사용

※ transition 동반 필수!!!

- **transform : scale();**

=> 배율 조정

=> 기본값 : 1 / 더 작게 0.00 소수점 / 더 크게 1.5

- **transform : rotate(90deg);**

=> 단위 : deg

- **transform : translate(X,Y);**

=> px, %

※ 명확한 width, height를 알고 있는 경우

margin-left : -(width/2)px;

margin-top : -(height/2)px;

※ 크기 값이 불분명한 경우

transform: translate(-50%, -50%);

★추천

```
position: absolute;  
top : 50%;  
left : 50%;
```

[animation]

- 자동으로 움직여줄 때 사용
- animation + keyframe
- animation-name

※ 사용 불가한 이름

1. 숫자가 첫번째로 오는 경우

2. 특수문자

- animation-duration

- animation-iteration-count

=> 숫자

=> infinite

```
animation : name duration (timing-function) infinite direction;
```

```
@keyframes 애니메이션 이름 {  
    from{}  
    50%{}  
    to{}  
}
```

=> 키프레임은 %로 조절도 가능함

=> from, to, % 혼용 사용 가능

- 애니메이션 기본 방향

=> from -> to / 순방향

[웹 컴바인 #2]

• 미디어쿼리(media query)

==> 단말기의 유형, 화면 해상도, 뷰 포트 너비에 따라서 css 스타일을 달리 적용하는 것
==> 반응형 웹 디자인의 기본

```
<<미디어쿼리 기본 틀>>
```

```
@media 미디어유형 조건 {  
    css 명령어  
}
```

<<미디어의 유형>>

1. print

- 인쇄물 및 출력 미리보기 화면에 표시중인 문서

```
ex)  
@media print {  
    body{  
        font-size:12pt;  
    }  
}
```

=> 페이지가 인쇄된 경우에만 본문을 12pt로 설정

2. screen

- 스크린 화면

3. speech

- 스크린 리더기

```

```

4. all

- 위 모든 미디어 포함

screen, all(범용성 가장 높음)

<<분기점(break point)>>

- 4개

==> pc / notebook / tab / mobile

- 3개

==> pc / tab / mobile

- PC break point-----

- @media all and (min-width:1024px){}

==> 가로 해상도가 1024px 보다 큰 경우에 적용

==> pc 사이즈라면 최소 1024픽셀이고, 그보다 더 큰 경우에 적용하는 미디어 쿼리

- tab break point-----

- @media all and (min-width:768px) and (max-width:1023px){}

==> 최대 가로크기가 1023px 이고 최소 가로크기가 768일 때 tab 미디어 쿼리 적용

- mobile break point-----

- @media all and (max-width:767px){**

=> 모바일 분기점에서 최대로 가질 수 있는 가로의 크기는 767px이고 이보다 더 작은 경우에 적용

*** max-width**

=> 뷰 영역에서의 최대 넓이, 즉 지정한 사이즈보다 작을 경우 적용

*** min-width**

=> 뷰 영역에서의 최소 넓이, 즉 지정한 사이즈보다 클 경우 적용

- 메를로웹 사이트 클론**

- reset 설정**

- <https://meyerweb.com/eric/tools/css/reset/>

```
/* http://meyerweb.com/eric/tools/css/reset/
v2.0 | 20110126
License: none (public domain)
*/

a{
    text-decoration: none;
    color: initial;
}

img{
    display: block;
    width: 100%;
}

*{
    box-sizing: border-box;
    font-family: 'Noto Sans KR', 'Titillium Web', sans-serif;
}

html, body, div, span, applet, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
a, abbr, acronym, address, big, cite, code,
del, dfn, em, img, ins, kbd, q, s, samp,
small, strike, strong, sub, sup, tt, var,
b, u, i, center,
dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td,
article, aside, canvas, details, embed,
figure, figcaption, footer, header, hgroup,
menu, nav, output, ruby, section, summary,
time, mark, audio, video {
    margin: 0;
    padding: 0;
    border: 0;
    vertical-align: baseline;
}
/* HTML5 display-role reset for older browsers */
article, aside, details, figcaption, figure,
footer, header, hgroup, menu, nav, section {
    display: block;
}
body {
    line-height: 1;
}
ol, ul {
    list-style: none;
}
blockquote, q {
    quotes: none;
}
blockquote:before, blockquote:after,
q:before, q:after {
    content: '';
    content: none;
}
table {
    border-collapse: collapse;
    border-spacing: 0;
}
```

2. common 설정

```
/*
:root

1. 색상 설정
- 주조색, 보조색, 강조색
- 기본 색상 설정(#fff)

2. 폰트 설정
- font-weight
*/
:root{
    --mainColor : #583cf2;
    --btnColor : #1e3a8b;
    --whiteColor : #fff;
    --blackColor : #222;
    --bold : 700;
    --kor : 'Noto Sans KR';
    --eng : 'Titillium Web';
}
```

3. style 설정

```
@import url(reset.css);
@import url(common.css);
```

4. html #1 (wrap ~ section_1)

```
<!DOCTYPE html>
<html lang="ko">
<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>메를로랩 클론</title>
    <script src="https://kit.fontawesome.com/8c36059e36.js" crossorigin="anonymous"></script>
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=Noto+Sans+KR:wght@100;300;400;500;700;900&family=Titillium+Web:wght@200;300;400;500;600;700;800;900" href="css/style.css"/>
</head>
<body>
    <div class="wrap">
        <header>
            <div class="inner">
                <h1 class="logo">
                    <a href="#">
                        
                    </a>
                </h1>
                <ul class="menu_wrap">
                    <li>
                        <a href="#">company</a>
                    </li>
                    <li>
                        <a href="#">business</a>
                    </li>
                    <li>
                        <a href="#">technology</a>
                    </li>
                    <li>
                        <a href="#">ir center</a>
                    </li>
                    <li>
                        <a href="#">cs center</a>
                    </li>
                </ul>
                <!-- header_Mobile 영역 -->
                <div class="header_m_menu">
```

```

        <span class="mobile_menu"></span>
    </div>
</div>
</header>
<!-- header_end -->

<section class="section_1">
    <div class="inner">
        <div class="main_txt_wrap">
            <!-- section_Mobile 영역 -->
            <h3>next <br class="mo_break">network</h3>
            <p>새로운 네트워크의 시작 시작의 중심 메를로랩</p>
            <div class="arrow_wrap">
                <span>view more</span>
                <div class="circle_arrow">
                    <i class="fa-solid fa-chevron-right"></i>
                </div>
            </div>
        </div>
        <ul class="bottom_nav">
            <li>
                <a href="#">01. next work</a>
            </li>
            <li>
                <a href="#">02. next work</a>
            </li>
            <li>
                <a href="#">03. next work</a>
            </li>
        </ul>
    </div>
    <a href="#" class="side_btn">
        <h3>Inquiry</h3>
        <i class="fa-solid fa-chevron-right"></i>
    </a>
</section>
<!-- section_1_end -->

<section class="section_2">
    <div class="inner">
        <div class="business_wrap_1">
            <div class="main_business_txt">
                <h1>
                    
                </h1>
                <h4>BUSINESS</h4>
                <p>
                    메를로랩의 독자적인 최신 네트워크 기술로 <br/>
                    현재와 미래를 바꾸는 혁신을 만들어 갑니다.
                </p>
                <a href="#" class="more_btn">VIEW MORE</a>
            </div>
            <div class="main_business_img">
                <figure>
                    <a href="#">
                        
                        <!-- section_2_Mobile 영역 #1 -->
                        
                    </a>
                </figure>
                <div class="business_img_txt">
                    <p>메를로랩 IOT</p>
                    <h5>Building IoT</h5>
                    <p class="sub_txt">
                        <!-- tab 영역 -->
                        메를로랩의 세계 최고 무선 기술을 통해 구현되는 스마트 빌딩<br class="tab_hidden"/>
                        조명의 실시간 서버 관리로 사용자 환경에 최적화 합니다.
                    </p>
                    <span>
                        VIEW MORE
                        <i class="fa-solid fa-plus"></i>
                    </span>
                </div>
            </div>
        </div>
        <!-- business_wrap_1_end -->
    <div class="business_wrap_2">
        <div class="business_box">
            <a href="#">
                <figure>

```

```


<!-- section_2_Mobile 영역 #2 -->

</figure>
<div class="business_txt_box">
<p>메를로랩 홈 IOT</p>
<h5>Home IoT</h5>
<p class="sub_txt">
    메를로랩의 IoT 솔루션을 이용하면 시스템의 도입 비용<br/>
    부담 없이 기존 가전 그대로 IoT 환경으로 변경할 수 있습니다.
</p>
<span>
    VIEW MORE
    <i class="fa-solid fa-plus"></i>
</span>
</div>
</a>
</div>
<div class="business_box">
<a href="#">
<figure>

<!-- section_2_Mobile 영역 #3 -->

</figure>
<div class="business_txt_box">
<p>메를로랩 에너지 솔루션</p>
<h5>Energy Solution</h5>
<p class="sub_txt">
    메를로랩의 네트워크 기술을 통해 실현되는 친환경 에너지 <br class="tab_hidden"/>
    서비스로 신재생 에너지 산업에 새로운 청사진을 제시합니다.
</p>
<span>
    VIEW MORE
    <i class="fa-solid fa-plus"></i>
</span>
</div>
</a>
</div>
</div>
<!-- business_wrap_2_end -->

</div>
</section>
<!-- section_2_end -->

```

4. html #2 (section_3 ~ section_4)

```

<section class="section_3">
<div class="section_3_bg">
<div class="inner">
<p class="slide_sub_txt">
    메를로랩의 혁신적인 미래기술
</p>
<h4>TECHNOLOGY</h4>
<div class="arrow_wrap">
    <i class="fa-solid fa-chevron-left"></i>
    <!-- ※ fa-chevron-left : i 태그의 class 명 -->
    <i class="fa-solid fa-chevron-right"></i>
</div>
</div>
</div>
<!-- section_3_bg_end -->

<!-- 눈에 바로 보여지는 영역 -->
<div class="slide_mask">
<div class="inner">
    <!-- 슬라이드 이미지를 잡고있는 영역 -->
    <ul class="slide_wrap">
        <li>
            <a href="#">
                <figure class="slide_img">
                    
                </figure>
                <div class="slide_txt_wrap">
                    <small>/01</small>
                    <h5>메쉬 네트워크</h5>
                    <p>
                        기존 표준 무선 기술의 한계를 단번에 뛰어넘은<br class="tab_hidden"/>
                        세계 최고의 무선 네트워크입니다.
                    </p>
                </div>
            </a>
        </li>
    </ul>
</div>

```

```

                </div>
            </a>
        </li>
        <li>
            <a href="#">
                <figure class="slide_img">
                    
                </figure>
                <div class="slide_txt_wrap">
                    <small>/02</small>
                    <h5>Dual-Stack 통신</h5>
                    <p>
                        메를로랩의 제품 설치만으로 건물 전체를<br class="tab_hidden"/>
                        하나의 IoT 시스템으로 통합할 수 있습니다.
                    </p>
                </div>
            </a>
        </li>
        <li>
            <a href="#">
                <figure class="slide_img">
                    
                </figure>
                <div class="slide_txt_wrap">
                    <small>/03</small>
                    <h5>스마트 조명</h5>
                    <p>
                        메를로랩의 혁신적인 네트워크 기술은 메를로랩의<br/>
                        스마트 조명으로 가치가 극대화됩니다.
                    </p>
                </div>
            </a>
        </li>
        <li>
            <a href="#">
                <figure class="slide_img">
                    
                </figure>
                <div class="slide_txt_wrap">
                    <small>/04</small>
                    <h5>메를로랩 IoT</h5>
                    <p>
                        메를로랩은 조명을 신경망으로 활용해 스마트 빌딩을<br/>
                        사람과 같은 하나님의 유기체로 만들어냅니다.
                    </p>
                </div>
            </a>
        </li>
        <li>
            <a href="#">
                <figure class="slide_img">
                    
                </figure>
                <div class="slide_txt_wrap">
                    <small>/05</small>
                    <h5>서비스 앱</h5>
                    <p>
                        IoT 서비스를 위한 메를로랩만의 기술이 집약된<br/>
                        서버기반의 서비스와 앱을 제공합니다.
                    </p>
                </div>
            </a>
        </li>
    </ul>
</div>
</div>
<!-- section_3_end -->

<section class="section_4">
<div class="inner">
    <div class="section_4_left_wrap">
        <a href="#">
            <!-- <figure>
                
            </figure> --!>
            <h4>IR Center</h4>
            <p>
                가치 중심 경영을 통해 주주가치를<br class="tab_hidden"/>
                극대화하겠습니다.
            </p>
            <div class="ir_arrow_wrap">
                <span>VIEW MORE</span>
                <i class="fa-solid fa-plus"></i>
            </div>
        </a>
    </div>

```

```

<div class="section_4_right_wrap">
    <div class="right_top_wrap">
        <ul class="right_top_menu">
            <li>
                <a href="#">
                    <span>보도자료</span>
                </a>
            </li>
            <li>
                <a href="#">
                    <span>공지사항</span>
                </a>
            </li>
        </ul>
        <div class="top_arrow_wrap">
            <span>VIEW MORE</span>
            <i class="fa-solid fa-plus"></i>
        </div>
    </div>
    <div class="right_bottom_wrap">
        <ul class="bottom_news_list">
            <li>
                <a href="#">
                    <figure class="news_img">
                        
                    </figure>
                    <div class="news_txt">
                        <p class="news_title">
                            메를로랩 'Fast DR' 구현에 기술력 인정...산업자원부 장관상
                        </p>
                        <p class="news_con">
                            [이데일리 이정현 기자] IoT 플랫폼 기업 메를로랩(대표 신소봉)은 산업통상자원부 장관상을 수상했다고 20일 밝혔다...
                        </p>
                        <p class="news_time">2023-03-17</p>
                    </div>
                </a>
            </li>
            <li>
                <a href="#">
                    <figure class="news_img">
                        
                    </figure>
                    <div class="news_txt">
                        <p class="news_title">
                            IoT 기반 '에너지 플랫폼'이 메를로랩 정체성
                        </p>
                        <p class="news_con">
                            "IoT 기반 '에너지 플랫폼'이 메를로랩 정체성" "IoT 기술을 보유한 에너지 플랫폼 회사가 우리의 정체성이다." 어떤
                        </p>
                        <p class="news_time">2022-01-27</p>
                    </div>
                </a>
            </li>
        </ul>
    </div>
</div>
<!-- section_4_end -->

```

4. html #3 (section_5 ~ footer)

```

<section class="section_5">
    <div class="inner">
        <div class="location_wrap">
            <div class="location_txt_wrap">
                <h4>Location<span>오시는 길</span></h4>
                <p>
                    서울특별시 금천구 디자털로9길 68 (가산동) <br class="mo_hidden tab_hidden">
                    대륭포스트 타워 5차 2002-2005호 <br>
                </p>
            <div class="location_btn_wrap">
                <a href="#">
                    문의하기
                    <i class="fa-solid fa-arrow-right"></i>
                <!-- 화살표 아이콘 -->
            </a>
                <a href="#">
                    FAQ
                    <i class="fa-solid fa-arrow-right"></i>
                <!-- 화살표 아이콘 -->
            </a>

```

```

        </div>
    </div>
    <!-- location__txt_wrap_end -->
</div>
<!-- location_wrap_end -->
</div>
<!-- inner_end -->

<div class="location_map">
    <a href="#" class="location_map_icon">
        <i class="fa-solid fa-location-dot"></i>
        <!-- 지도 아이콘 -->
        카카오맵 보기
    </a>
    <div class="location_map_img"></div>
</div>
</section>
<!-- section_5_end -->

<footer>
    <div class="inner">
        <div class="footer_top_wrap">
            <h1 class="footer_logo">
                <a href="#">
                    
                </a>
            </h1>
            <ul class="footer_top_list">
                <li>
                    <a href="#">COMPANY</a>
                </li>
                <li>
                    <a href="#">사이트맵</a>
                </li>
                <li>
                    <a href="#">이용약관</a>
                </li>
                <li>
                    <a href="#">이메일 무단 수집거부</a>
                </li>
            </ul>
            <!-- <div class="footer_side_wrap">
                <a href="#">
                    <h4>COMPANY</h4>
                    <h4>|</h4>
                    <h4>사이트맵</h4>
                    <h4>|</h4>
                    <h4>이용약관</h4>
                    <h4>|</h4>
                    <h4>이메일 무단 수집거부</h4>
                </a>
            </div> -->
        </div>
        <!-- footer_top_wrap_end -->
        <div class="footer_bottom_wrap">
            <div class="footer_bottom_left_box">
                <ul>
                    <li>대표자 : 신소봉</li>
                    <li>문의전화 : <a href="#">02) 862-1700</a></li>
                    <li>사업장 등록번호 : 119-86-57418</li>
                </ul>
                <!-- 주소내용 사용 시 -->
                <address>
                    서울특별시 금천구 디지털로9길 68 (가산동) <br class="pc_hidden tab_hidden"/> 대륭포스트 타워 5차 2002~2005호
                </address>
                <p class="copy_right">(C) Copyright merlot.lab All Rights Reserved.</p>
                <!-- <p>
                    대표자 : 신소봉 문의전화 : 02) 862-1700 사업장 등록번호 : 119-86-57418 <br>
                    서울특별시 금천구 디지털로9길 68 (가산동) 대륭포스트 타워 5차 2002~2005호 <br>
                </p>
                <h4>(C) Copyright merlot.lab All Rights Reserved.</h4> -->
            </div>
            <div class="footer_bottom_right">
                <ul>
                    <li>
                        <a href="#">
                            <span class="footer_icon_1"></span>
                            자사몰 바로가기
                        </a>
                    </li>
                    <li>
                        <a href="#">
                            <span class="footer_icon_2"></span>
                            유튜브 채널
                        </a>
                    </li>
                </ul>
            </div>
        </div>
    </div>

```

```

        </ul>
        <!-- <a href="#">
        <figure>
            
        </figure>
        <h4>자사몰 바로가기</h4>
    </a> -->
    <!-- <a href="#">
    <figure>
        
    </figure>
    <h4>유튜브 채널</h4>
    </a> -->
</div>
</div>
</div>
</body>
</html>

```

5. css(style) #1 (wrap ~ section_1)

```

@import url(reset.css);
@import url(common.css);

/*
 * wrap은 뛰어나가는 요소에 대해 overflow:hidden 처리 후 가장 마지막에 작성
 */

header{
    width: 100%;
    height: 100px;
    position: fixed;
    top: 0;
    left: 0;
    z-index: 9999;

    /*
        배경색이나 특정 요소만 투명하게 만들때 사용함
        opacity 의 경우 전체 요소에 투명속성을 적용할 때 사용
    */
    background-color: rgba(0, 0, 0, .5);
}

header > .inner{
    height: 100%;
    display: flex;
    align-items: center;
    justify-content: space-between;
}

.logo{
    width: 155px;
    /*
        ※ 이미지를 display:block;, width:100%로 하는 이유
        ==> 이미지태그는 가로크기만 맞추면 세로는 자동 맞춤
    */
    flex-shrink: 0;
    /*
        ※ 화면사이즈가 작아질 경우 flex-item이 찌그러지는 경우 방지하기 위해 사용
        (logo 사이즈가 지정된 155px 이하로 줄어들지 않음)
    */
}

.menu_wrap{
    max-width: 1020px;
    width: 90%;
    height: 100%;
    display: flex;
    justify-content: space-between;
    align-items: center;
    /* background-color: gray; */
}

.menu_wrap > li{
    width: 20%;
    /* text-align: center; */
    height: 100%;
    display: flex;
    justify-content: center;
}

```

```

        align-items: center;
        transition: all .2s;
    }

    .menu_wrap a{
        color: var(--whiteColor);
        text-transform: uppercase;
    }

    .menu_wrap > li:hover{
        background-color: var(--mainColor);
        cursor: pointer;
    }

    /* tab 사이즈 적용 */
    @media all and (min-width:768px) and (max-width:1023px){

        /* .logo */

        .menu_wrap{
            display: none;
        }

        .header_m_menu{
            display: block;
        }

        .header_m_menu{
            width: 30px;
            height: 22px;
            display: flex;
            justify-content: center;
            align-items: center;
            position: relative;
        }

        .header_m_menu::before{
            content: "";
            width: 100%;
            height: 2px;
            background-color: var(--whiteColor);
            position: absolute;
            top: 0;
            left: 0;
        }

        .header_m_menu::after{
            content: "";
            width: 100%;
            height: 2px;
            background-color: var(--whiteColor);
            position: absolute;
            bottom: 0;
            left: 0;
        }

        .mobile_menu{
            display: inline-block;
            width: 100%;
            height: 2px;
            background-color: var(--whiteColor);
        }
    }

    /* mobile 사이즈 적용 */
    @media all and (max-width:767px){

        header{
            height: 80px;
        }

        .logo{
            width: 125px;
        }

        .menu_wrap{
            display: none;
        }

        .header_m_menu{
            width: 30px;
            height: 22px;
            display: flex;
            justify-content: center;
            align-items: center;
            position: relative;
        }
    }
}

```

```

        }

.header_m_menu::before{
    content: "";
    width: 100%;
    height: 2px;
    background-color: var(--whiteColor);
    position: absolute;
    top: 0;
    left: 0;
}

.header_m_menu::after{
    content: "";
    width: 100%;
    height: 2px;
    background-color: var(--whiteColor);
    position: absolute;
    bottom: 0;
    left: 0;
}

.mobile_menu{
    display: inline-block;
    width: 100%;
    height: 2px;
    background-color: var(--whiteColor);
}

/*
-----header_end-----
*/

.section_1{
    width: 100%;
    height: 100vh;
    /*
        ※ view port height : 0 ~ 100까지 적용 가능
    */
    background: url(..../img/main_visual.jpg) no-repeat
    top center;
    background-size: cover;
    position: relative;
}

.section_1 > .inner{
    position: relative;
    height: 100%;

    /* ※ 공통 영역 inner에 적용된 0, 20px 조건 제외 */
    padding: initial;
}

.main_txt_wrap{
    position: absolute;
    top: 50%;
    left: 0;
    transform: translateY(-50%);
}

.main_txt_wrap > h3{
    font-size: 60px;
    color: var(--whiteColor);
    text-transform: uppercase;
    font-family: var(--eng);
    font-weight: var(--bold);
}

.main_txt_wrap > p{
    font-size: 18px;
    color: var(--whiteColor);
    /* ※ margin : 위 원 오른쪽 아래 */
    margin: 20px 0 50px;
    /* ※ 메인과 서브 텍스트의 충돌 효과 방지 */
    opacity: .6;
}

.arrow_wrap{
    display: flex;

    /*
        ※ .arrow_wrap > span 와 .circle_arrow의 중앙을 맞출 height값이 존재하기 때문에
        별도로 height 값을 지정하지 않아도 align-items: center; 가능
    */
    align-items: center;
}

```

```

}

.arrow_wrap > span{
  color: var(--whiteColor);
  text-transform: uppercase;
  font-family: var(--eng);
}

.circle_arrow{
  width: 45px;
  height: 45px;
  background-color:
    rgba(255, 255, 255, 0.2);
  border-radius: 50%;
  display: flex;
  justify-content: center;
  align-items: center;
  margin-left: 20px;
  transition: all .5s;
}

.circle_arrow > i{
  color: #fff;
}

.circle_arrow:hover{
  background-color: var(--mainColor);
  cursor: pointer;
}

.bottom_nav{
  position: absolute;
  bottom: 60px;
  right: 0;
  display: flex;
}

.bottom_nav > li{
  width: 200px;
  border-bottom: 1px solid var(--whiteColor);
  margin-left: 30px;
  padding-bottom: 10px;
  opacity: .4;
}

.bottom_nav > li:first-child{
  opacity: 1;
}

.bottom_nav a{
  color: var(--whiteColor);
  text-transform: uppercase;
}

.side_btn{
  width: 165px;
  height: 45px;
  background-color: var(--whiteColor);
  position: absolute;
  top: 50%;
  left: -80px;
  transform: rotate(90deg) translateY(-50%);
  display: flex;
  align-items: center;
  justify-content: space-between;
  padding: 0 15px;
  transition: all .5s;
}

.side_btn > h3{
  font-family: var(--eng);
}

.side_btn:hover{
  background-color: var(--mainColor);
}

.side_btn:hover > *{
  color: var(--whiteColor);
}

.mo_break{
  display: none;
}

/* tab 사이즈 적용 */

```

```

@media all and (min-width:768px) and (max-width:1023px){

    .section_1{
        height: 85vh;
    }

    .section_1 > .inner{
        padding: 0 20px;
    }

    .main_txt_wrap{
        left: 20px;
    }

    .main_txt_wrap > h3{
        font-size: 48px;
    }

    .main_txt_wrap > p{
        font-size: 17px;
        margin: 15px 0 30px;
    }

    .side_btn{
        display: none;
    }

    .bottom_nav{
        left: 50%;
        transform: translateX(-50%);
        width: calc(100% - 40px);
        gap: 20px;
    }

    .bottom_nav > li{
        width: 33%;
        margin-left: 0;
    }

    .bottom_nav a{
        font-size: 13px;
    }

}

/* mobile 사이즈 적용 */

@media all and (max-width:767px){

    .section_1{
        height: 85vh;
    }

    .section_1 > .inner{
        padding: 0 20px;
    }

    .side_btn{
        display: none;
    }

    .main_txt_wrap{
        left: 20px;
    }

    .main_txt_wrap > h3{
        font-size: 30px;
    }

    .mo_break{
        display: block;
    }

    .main_txt_wrap > p{
        font-size: 14px;
        margin: 20px 0;
    }

    .arrow_wrap > span{
        font-size: 14px;
        margin-left: 0;
    }

    .circle_arrow{
        background: none;
    }

    .bottom_nav{
}

```

```

        flex-direction: column;
        left: 20px;
    }

    .bottom_nav > li{
        width: 180px;
        margin-left: 0;
        margin-bottom: 15px;
    }

    .bottom_nav > li > a{
        font-size: 13px;
    }

}

/* -----section_1_end----- */

```

5. css(style) #2 (section_2)

```

.section_2{
    width: 100%;
    padding: 150px 0;
    background: url(..../img/section_2/main_business_bg1.jpg) no-repeat top center;
    position: relative;
}

.business_wrap_1{
    width: 100%;
    display: flex;
    justify-content: space-between;
    align-items: center;
}

.main_business_txt{
    width: 40%;
}

.main_business_txt > h1{
    width: 130px;
}

.main_business_txt > h4{
    font-size: 60px;
    font-family: var(--eng);
    font-weight:var(--bold);
    margin: 20px 0;
}

.main_business_txt > p{
    font-size: 18px;
    opacity: .7;
    line-height: 25px;
}

.more_btn{
    display: flex;
    width: 180px;
    height: 50px;
    background-color: var(--mainColor);
    justify-content: center;
    align-items: center;
    color: var(--whiteColor);
    border-radius: 25px;
    margin-top: 45px;
    transition: all .5s;
}

.more_btn:hover{
    background-color: navy;
}

.main_business_img{
    width: 60%;
}

.main_business_img > figure{
    width: 100%;
}

```

```

}

.business_img_txt{
    max-width: 500px;
    width: 80%;
    background-color: var(--whiteColor);
    padding: 40px 30px;
    position: relative;
    top: -150px;
}

.business_img_txt > p:first-child{
    font-size: 18px;
    color: var(--mainColor);
}

.business_img_txt > h5{
    font-size: 40px;
    font-family: var(--eng);
    font-weight:var(--bold);
    margin: 15px 0 30px;
}

.business_img_txt > .sub_txt{
    opacity: .7;
    line-height: 25px;
    margin-bottom: 30px;
}

.business_img_txt > span{
    cursor: pointer;
}

.business_img_txt > span > i{
    transition: all .5s;
}

.business_img_txt > span:hover i{
    transform: rotate(90deg);
}

.business_wrap_2{
    width: 100%;
    display: flex;
    justify-content: space-between;

    /*
     * ::after로 bg를 겹쳐주면서 가려지게되는 영역을 앞으로 끄내주기 위해 사용
     */
    position: relative;
    z-index: 30;
}

.business_txt_box{
    max-width: 500px;
    width: 90%;
    background-color: var(--whiteColor);
    padding: 40px 30px;
    position: relative;
    top: -150px;
}

.business_txt_box > p:first-child{
    font-size: 18px;
    color: var(--mainColor);
}

.business_txt_box > h5{
    font-size: 40px;
    font-family: var(--eng);
    margin: 10px 0 20px;
}

.business_txt_box > .sub_txt{
    opacity: .7;
    line-height: 25px;
    margin-bottom: 30px;
}

.business_txt_box i:hover{

```

```

        transform: rotate(90deg);
        transition: all .5s;
    }

/* * ::after, ::before의 경우 해당 섹션이 끝나는 지점(닫히는 태그) 바로 전에 적용 */
/* 1) div 태그를 사용해서 box 생성 후 position으로 위치를 잡아준다 */
/* 2) after 처리 */
.section_2::after{
    content: "";
    /*
        ※ 적용하려는 내용이 bg(백그라운드 이미지)이기 때문에 display 속성 변경
        ※ bg의 경우 무조건 height, padding 값 필수
    */
    display: inline-block;
    width: 100%;
    height: 850px;
    background: url(..../img/section_2/main_business_bg2.jpg) no-repeat top center;

    /*
        bg 이미지를 겹쳐주기 위해 ::after의 부모요소인 section_2에 position:relative; 적용
        자식(자손) 요소인 ::after에 position: absolute; 적용
    */
    position: absolute;
    bottom: 0;
}

.section_2_mo_img{
    display: none;
}

/* tab 사이즈 적용 */
@media all and (min-width:768px) and (max-width:1023px){
/*
    * tap
    - min-width: mobile max-width + 1
    - max-width: pc max-width -1
*/
    .section_2{
        padding: 100px 0;
    }

    .business_wrap_1{
        flex-direction: column;
        align-items: flex-start;
    }

    .main_business_txt{
        width: 100%;
    }

    .main_business_txt > h4{
        font-size: 44px;
    }

    .tab_hidden{
        display: none;
    }

    .main_business_txt > p{
        font-size: 17px;
    }

    .more_btn{
        margin-top: 35px;
    }

    .main_business_img{
        width: 100%;
        margin-top: 60px;
    }

    .business_img_txt{
        /*
            ※ br태그 등을 사용하여 설정된 max-width 범위안에서 줄바꿈을 해줄 경우 값을 명확하게 적되,
            줄바꿈이 없을 경우 width의 영역을 %로 지정하게 되면 해당 영역에 가득 차도록 텍스트가 쓰여진다
            (텍스트가 영역을 넘칠 경우 자동 줄바꿈)
        */
        max-width: none;
        width: 90%;
    }

    .business_img_txt > p:first-child{
        font-size: 17px;
    }

    .business_img_txt > h5{

```

```

        font-size: 30px;
    }

    .sub_txt{
        font-size: 15px;
    }

    .business_img_txt > span{
        font-size: 14px;
        font-family: var(--eng);
        font-weight: var(--bold);
    }

    .business_wrap_2{
        gap: 25px;
    }

    .business_box{
        width: 50%;
    }

    .business_box:nth-child(2){
        padding-top: 130px;
    }

    .business_box:nth-child(2) .section_2_pc_img{
        display: none;
    }

    .business_box:nth-child(2) .section_2_mo_img{
        display: block;
    }

    .business_txt_box{
        position: relative;
        top: -100px;
    }

    .business_txt_box > p:first-child{
        font-size: 17px;
    }

    .business_txt_box > h5{
        font-size: 30px;
    }

    .business_txt_box > .sub_txt{
        font-size: 14px;
    }

    .business_txt_box > span{
        font-size: 14px;
        font-family: var(--eng);
    }

}

/* mobile 사이즈 적용 */
@media all and (max-width:767px){

    .section_2{
        padding: 75px 0 0;
        /* padding-top:75px */
    }

    .main_business_txt > h4{
        font-size: 35px;
        margin: 15px 0;
    }

    .main_business_txt > p{
        font-size: 14px;
        line-height: 20px;
    }

    .more_btn{
        width: 150px;
        height: 40px;
        font-size: 13px;
        margin-top: 35px;
    }

    .main_business_img{
        margin-top: 35px;
    }

    .business_img_txt{

```

```

        padding: 30px 20px;
        width: 90%;
        top: -90px;
    }

    .business_img_txt > p:first-child{
        font-size: 16px;
        /* 중복옵션은 가능한 제거해줄 것 */
        /* color: var(--mainColor); */
    }

    .business_img_txt > h5{
        font-size: 25px;
        /* font-family: var(--eng);
        font-weight: var(--bold); */
        margin: 15px 0 25px
    }

    .business_img_txt > .sub_txt, .business_txt_box > .sub_txt{
        font-size: 14px;
        /* opacity: .7; */
        line-height: 20px;
        /* margin-bottom: 30px; */
    }

    .business_img_txt > span{
        /* cursor: pointer; */
        font-size: 13px;
    }

    .business_wrap_1{
        flex-direction: column;
    }

    .main_business_txt{
        width: 100%;
    }

    .main_business_img{
        width: 100%;
    }

    .section_2_mo_img{
        display: block;
    }

    .section_2_pc_img{
        display: none;
    }

    .business_wrap_2{
        flex-direction: column;
    }

    .business_txt_box{
        position: relative;
        top: -90px;
    }

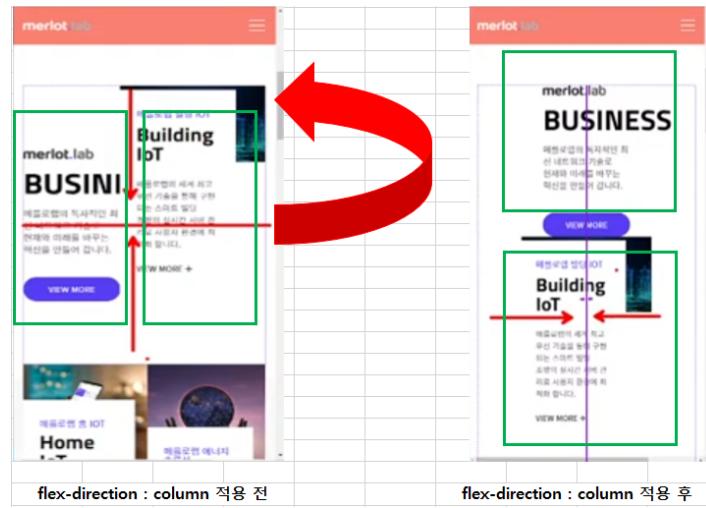
    .business_txt_box > p:first-child{
        font-size: 16px;
    }

    .business_txt_box > h5{
        font-size: 16px;
    }
}

/* -----section_2_end----- */

```

※ flex 처리된 요소 **flex-direction: column;** 적용 시 정렬(축) 기준



5. css(style) #3 (section_3)

```

.section_3{
  width: 100%;

  /*
    통상적으로 넘치는 요소의 부모에게 적용해주는 것이 맞으나 마크업이 길어지는 경우
    각 section 별로 요소 전체를 감싸고 있는 부모요소에게 적용해주면 됨
    ※ wrap을 가장 마지막에 적용하는 이유또한 자식요소가 부모를 넘칠 경우 overflow: hidden; 처리를 위해서임
  */
  overflow: hidden;
}

.section_3_bg{
  width: 100%;
  height: 500px;
  /*
    ※ 이미지를 유지하면서 영역을 확장하고 싶을 때 사용
    bottom 140px : 아래로 140px 영역만큼 확장하겠다는 의미
  */
  background: #171d3d url(..../img/section_3/main_technology_bg.jpg) no-repeat bottom 140px right;
  padding: 100px;
}

.slide_sub_txt{
  color: var(--whiteColor);
  font-size: 18px;
  opacity: .6;
}

.section_3_bg h4{
  font-size: 50px;
  color: var(--whiteColor);
  font-family: var(--eng);
  /* margin : 위 원 오른 아래 */
  margin: 10px 0 40px;
}

.arrow_wrap > i{
  color: var(--whiteColor);
}

.fa-chevron-left::after{
  content: "|";
  /* ※ inline 요소에 margin을 적용하기 위한 속성 변경 */
  display: inline-block;
  margin: 0 14px;
  opacity: .4;
}

.slide_mask{
  /* overflow:hidden */
  position: relative;
  top : -170px
}

.slide_mask > .inner{
  background-color: var(--whiteColor);
}

```

```

padding: 60px 20px 0 80px;

/* slide_mask .inner::after를 통해 확장한 영역을 뒤로 밀여주기위한 선형작업 */
position: relative;
}

.slide_wrap{
  display: flex;
  position: relative;
  z-index: 2;
}

.slide_wrap > li{
  /* flex-shrink: 0; or flex-wrap : wrap을 적용해주지 않으면 값이 정상적으로 적용되지 않음 */
  width: 40%;
  flex-shrink: 0;
  padding-right: 25px;
}

/* ::after, ::before (가상선택자) 사용 시 별도의 코드 간소화 용이
* 가상선택자는 inline요소이기 때문에 width, height 값을 넣을 수 없어서 속성 변경
.slide_mask > .inner::after{
  content: "";
  display: inline-block;
  width: 100%;
  height: 100%;
  background-color: var(--whiteColor);
  /* slide_mask > .inner 와 .inner::after는 flex 처리 불가능하기때문에 position 사용 */
  position: absolute;
  top: 0;
  /* 본인이 가지고 있는 가로 크기만큼 나란히 밀여줌 */
  left: 100%;
}

.slide_wrap{
  display: flex;
  /* 가리는 이미지 요소를 앞으로 꺼내주기 위해 사용 */
  position: relative;
  z-index: 2;
}

.slide_wrap > li{
  width: 40%;
  flex-shrink: 0;
  padding-right: 25px;
}

.slide_txt_wrap{
  padding: 30px 0 0 55px;

  /* .slide_txt_wrap > small 요소 원쪽 상단 정렬을 위한 선행 작업 */
  position: relative;
}

.slide_txt_wrap > small{
  position: absolute;
  left: 0;
  font-weight: var(--bold);
  color: var(--mainColor);
}

.slide_txt_wrap > h5{
  font-size: 25px;
  margin-bottom: 15px;
}

.slide_txt_wrap > p{
  font-size: 18px;
  opacity: .7;
  line-height: 24px;
}

.slide_img{
  /* 이미지를 감싸고 있는 부모요소에 overflow: hidden; 처리 */
  overflow: hidden;
}

.slide_wrap img{
  transition: all .5s;
}

.slide_wrap > li:hover img{
  transform: scale(1.2);
}

/* tab 사이즈 적용 */

```

```

@media all and (min-width:768px) and (max-width:1023px){

    .section_3{
        padding-bottom: 100px;
    }

    .section_3_bg{
        height: 300px;
        background: #171d3d url(..../img/section_3/main_technology_bg.jpg) no-repeat bottom right;
        background-position: bottom right;
        background-size: 100% 100%;
    }

    .section_3_bg > .inner > .slide_sub_txt{
        font-size: 16px;
    }

    .section_3_bg > .inner > h4{
        font-size: 36px;
    }

    .slide_mask{
        position: static;
    }

    .slide_mask > .inner{
        padding: 40px 20px 0;
    }

    .slide_wrap{
        overflow: hidden;
        gap: 10px;
    }

    .slide_wrap > li{
        width: 50%;
        padding-right: 0;
    }

    .slide_txt_wrap{
        padding: 25px 0 0 0;
    }

    .slide_txt_wrap > small{
        /* small에 기존 설정되어 있던 position: absolute; 속성 제거 */
        position: static;
    }

    .slide_wrap > h5{
        font-size: 20px;
        margin: 8px 0 13px;
    }

    .slide_txt_wrap > p{
        font-size: 15px;
        line-height: 19px;
    }

}

/* mobile 사이즈 적용 */
@media all and (max-width:767px){

    .section_3{
        padding-bottom: 100px;
    }

    .section_3_bg{
        height: 220px;
        background: url(..../img/section_3/main_technology_bg.jpg) no-repeat bottom right;
        background-size: auto 100%;
        padding-top: 50px;
    }

    .slide_sub_txt{
        font-size: 16px;
    }

    .section_3_bg h4{
        font-size: 30px;
    }

    .slide_mask{
        position: static;
        /* position: initial; */
    }

}

```

```

/*
    position: static; : position의 기본값 상태, 좌표값 영향을 받지 않음
    position: initial; : position의 설정을 초기화
 */

.slide_mask > .inner{
    padding: 40px 20px 0;
}

.slide_wrap{
    overflow: hidden;
    gap: 20px;
}

.slide_wrap > li{
    width: 50%;
    padding-right: 0;
}

.slide_txt_wrap{
    padding: 20px 0 0;
}

.slide_txt_wrap > small{
    position: static;
}

.slide_txt_wrap > h5{
    font-size: 20px;
    margin-top: 10px;
}

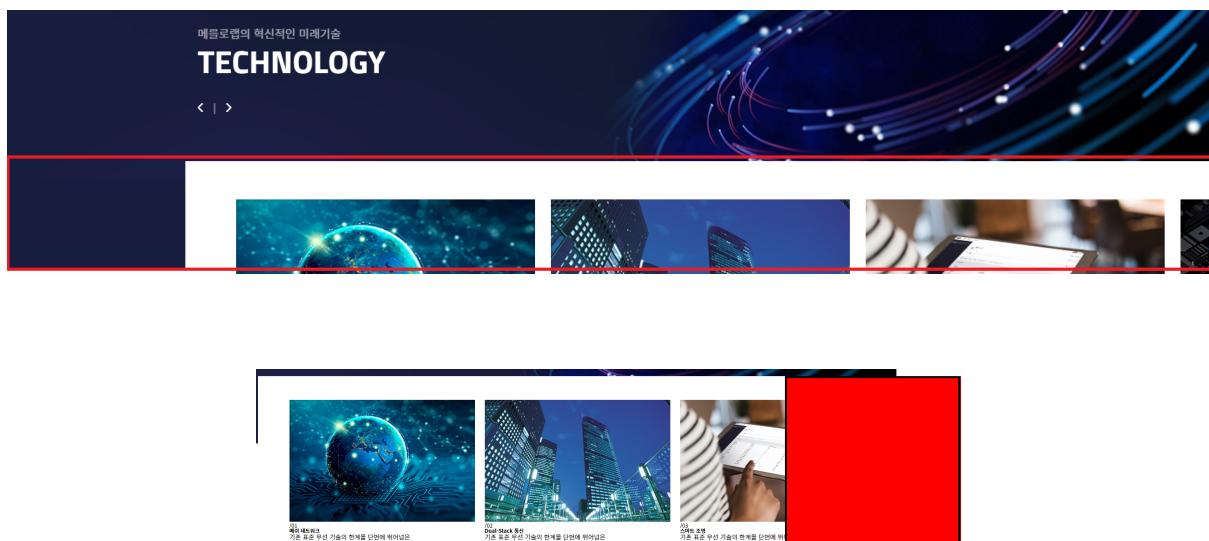
.slide_txt_wrap > p {
    font-size: 13px;
    line-height: 19px;
}

}

/* -----section_3_end----- */

```

※ 넘치는 요소 처리



The screenshot shows two windows side-by-side. On the left is the Visual Studio Code editor with the file 'style.css' open. A red box highlights a section of the code related to a slide mask. On the right is a web browser displaying a news website. A green box highlights the slide mask effect on the right side of the page. A red arrow points from the highlighted code in the editor to the highlighted area in the browser. A green arrow points from the browser back to the code editor.

```

.slide_mask > .inner{
  background-color: var(--whiteColor);
  padding: 60px 20px 0 80px;

  /* slide_mask .inner::after를 통해 확장한 영역을 위로 끌어주기 위한 선택자 */
  position: relative;
}

.slide_mask .inner::after{
  content: "";
  /* ※ ::after, ::before는 기술선택자 사용 시 별도의 코드 간소화 풀이 */
  /* 기본선택자는 inline-block이 때문에 width, height 값을 넣을 수 있어서 속성 변경 */
  display: inline-block;
  width: 100%;
  height: 100%;
  background-color: var(--whiteColor);
  position: absolute;
  top: 0;
  left: 100%;
}

.slide_wrap{
  display: flex;
  position: relative;
  z-index: 2;
}

```

```

.section_3_bg{
  width: 100%;
  height: 500px;
  background: #171d3d url(../img/section_3/main_technology_bg.jpg) no-repeat bottom 140px right;
  padding: 100px;
}


```

5. css(style) #4 (section_4)

```

.section_4{
  /* ※ 완전히 정중앙을 맞추지 않아도 될 경우 유용함 */
  padding: 120px 0 230px;
  /* width: 100%; */

  /* ※ 전체영역을 채우지 않을 경우 값 수정 필요 */
  /*
    height: 100vh;
    * section_4의 경우 부 포트 전체영역을 채우지 않기 때문에 70 ~ 80으로 값 설정
  */
  background: url(..../img/section_4/main_ir_bg.jpg) no-repeat top center;
}

/* section_4은 항상 common(공통)의 속성을 유지하여야 한다 */
.section_4 > .inner{
  /* width: 100%; */
  height: auto; /*
  display: flex;
  /* background-color: beige; */
  justify-content: space-between;

  /*
    * flex-item 사이에 간격을 주는 속성
    반드시 flex-container(부모)에 속성을 적용해 주어야함
  */
  gap: 60px;
}

.section_4_left_wrap{
  /*
    * max-width를 사용할 경우 절대값으로 할당된 px를 제외한 나머지 영역을 중앙 정렬이
    무너지지 않은 상태로 일관성 있는 레이아웃을 유지할 수 있음
  */
  max-width: 370px;
  /* width: 30%; */
  width: 100%;
  padding: 70px 40px;

  /* 나란히 flex 처리 된 section_4_right_wrap에 의해 부모요소에 크기를 맞추는 과정에서 줄어들 것을 감안한 속성 적용 */
  flex-shrink: 0;
  /* margin-right: 60px; */
  background: url(..../img/section_4/main_ir_img1.jpg) no-repeat top center;

  /*
    * 태그와 태그의 중첩은 position 사용
    position을 사용하려면 반영할 이미지를 img 태그대신 background 속성(css)으로 사용하는 것을 추천
  */
}

/* .section_4_left_wrap img{
  width: 100%;
  height: 400px;
}

```

```

        object-fit: cover;
    } */

.section_4_left_wrap h4{
    font-size: 40px;
    color: var(--whiteColor);
    font-family: var(--eng);
    /* position: absolute;
    top: 50px;
    left: 50px; */
}

.section_4_left_wrap p{
    color: var(--whiteColor);
    /* position: absolute;
    top: 100px;
    left: 50px; */
    opacity: .7;
    margin-top: 20px;
    line-height: 24px;
    /* line-height: 25px; */
}

.ir_arrow_wrap{
    margin-top: 170px;
}

.ir_arrow_wrap > *{
    color: var(--whiteColor);
}

.ir_arrow_wrap > span{
    font-size: 14px;
    font-family: var(--eng);
    /* color: var(--whiteColor);
    position: absolute;
    bottom : 200px;
    left: 50px; */
}

.ir_arrow_wrap > i{
    color: var(--whiteColor);
    /* position: absolute;
    bottom: 200px;
    left: 150px; */
}

/* -----section_4_left_wrap_end----- */

.section_4_right_wrap{
    /* width: 60%;
    height: 100%; */

    /* section_4_left_wrap에게 적용한 max-width를 제외한 모든 영역 설정 */
    width: 100%;
    padding-top: 5px;
    /* flex:1; */
    /* width:100%와 동일한 효과 적용 */
}

/*
 * flex-container / flex-item

flex-item
-> 부모요소에 항상 크기를 맞추려고 하기때문에 본연에 값을 잃어버리는 경우가 있음
    flex-container를 벗어나지 않기 위해 본인의 원래 크기값을 줄이면서 container안에 정렬됨

-> flex-wrap : wrap (item 속성의 원본크기를 유지하되 부모 크기를 넘을 경우 줄바꿈)
-> flex-shrink : 0 (줄바꿈 없이 item 속성의 원본크기를 유지함)
    * flex-item에게만 적용할 수 있는 속성
*/
.right_top_wrap{
    width: 100%;
    /* height: 50px; */
    display: flex;
    justify-content: space-between;
    align-items: center;
    /* position: relative;
    border-bottom: 3px solid #000;
    font-weight: bold;
    opacity: .7; */
}

.right_top_menu{
    display: flex;

```

```

}

.right_top_menu > li{
  padding: 0 10px;
}

.top_arrow_wrap:hover{
  cursor: pointer;
}

.top_arrow_wrap > span{
  font-size: 14px;
  /* position: absolute;
  top: 0;
  right: 30px; */
  /* text-align: center; */
  /* line-height: 50px; */
  font-family: var(--eng);
  font-weight: var(--bold);
}

/* 보도자료, 공지사항 마크 효과 적용 영역 */
.right_top_menu a{
  position: relative;
}

.right_top_menu a::after{
  content: "";
  display: inline-block;
  width: 0%;
  height: 50%;
  background-color: plum;
  position: absolute;
  left: 0;
  bottom: 0;
  transition: all .3s;
}

.right_top_menu span{
  position: relative;
  z-index: 30;
}

.right_top_menu a:hover::after{
  width: 100%;
}

.right_bottom_wrap{
  max-width: 870px;
  /* margin: 20px 10px;
  width: 100%; */
  height: 100%; */
  margin-top: 15px;
  border-top: 1px solid #000;
}

.bottom_news_list > li{
  padding: 25px 10px;
  border-bottom: 1px solid #ddd;
}

.bottom_news_list > li > a{
  display: flex;
  /* align-items: center; */
  /* gap: 30px; */
}

/* .bottom_news_list{
  position: relative;
} */

.bottom_news_list .news_img{
  max-width: 230px;
  /* width: 30%; */
  width: 100%;
  border: 1px solid #ddd;
  margin-right: 30px;
  flex-shrink: 0;

  /* news_img : img를 감싸는 엑자 역할 */
  position: relative;
  overflow: hidden;
}

.news_img > img{
  position: absolute;
  top: 50%;
}

```

```

        left: 50%;
        transform: translate(-50%, -50%);
    }

/* .bottom_news_list > li:first-child{
    padding-bottom: 20px;
    border-bottom: 1px solid lightgray;
} */

.bottom_news_list .news_title{
    font-size: 20px;
    font-weight: var(--bold);
    /* margin-bottom: 10px; */
}

.bottom_news_list .news_con{
    font-size: 14px;
    margin: 10px 0;
    opacity: .7;
    line-height: 20px;
}

.news_txt > .news_con:hover{
    text-decoration: underline;
}

.news_txt > .news_time{
    font-size: 14px;
    opacity: .7;
}

.news_txt > .news_time::before{
/* * 아이콘을 태그대신 css에 반영할 경우 */
    content: "\f017";
}

/* .bottom_news_list .news_txt{
    width: 70%;
    margin-left: 30px;
    line-height: 25px;
} */

/* .bottom_news_list .news_con , .news_time{
    font-size: 12px;
    opacity: .6;
} */

/* .bottom_news_list > li:nth-child(2) img{
    width: 100%;
    max-height: 130px;
    object-fit: cover;
    margin-top: 20px;
} */

/* -----section_4_right_wrap_end----- */

/* tab 사이즈 적용 */
@media all and (min-width:768px) and (max-width:1023px){

    .section_4{
        padding: 100px 0;
    }

    .section_4 > .inner{
        flex-direction: column;
    }

    .section_4_left_wrap{
        max-width: none;
        height: 240px;
        background: #190544 url(..../img/section_4/main_ir_img1.jpg) no-repeat;
        background-position: bottom right -250px;
        padding: 40px;
    }

    .section_4_left_wrap h4{
        font-size: 28px;
    }

    .section_4_left_wrap p{
        font-size: 15px;
    }

    .ir_arrow_wrap{
        margin-top: 70px;
    }
}

```

```

.news_txt > .news_title{
    font-size: 17px;
    font-weight: 400;
}

.news_txt > .news_con{
    font-size: 14px;
    margin: 20px 0;
}

.news_txt > .news_time{
    font-size: 14px;
}

/* mobile 사이즈 적용 */
@media all and (max-width:767px){

    .section_4{
        padding: 100px 0;
    }

    .section_4 > .inner{
        flex-direction: column;
    }

    .section_4_left_wrap{
        max-width: none;
        padding: 30px;
        height: 200px;
        background: url(../img/section_4/main_ir_img1.jpg) no-repeat;
        background-position: bottom right -250px;
    }

    .section_4_left_wrap h4{
        font-size: 25px;
    }

    .section_4_left_wrap p{
        font-size: 14px;
        line-height: 19px;
    }

    .ir_arrow_wrap{
        margin-top: 40px;
    }

    .top_arrow_wrap{
        display: none;
    }

    .news_img{
        display: none;
    }

    .news_txt > .news_title{
        font-size: 18px;
        font-weight: 400px;
    }
}

/* -----section_4_end----- */

```

※ background 이미지 위치 조정

```

} */

/* -----section_4_right_wrap_end----- */

@media all and (min-width:768px) and (max-width:1023px){

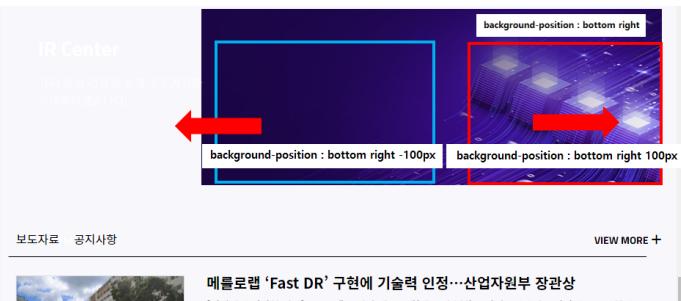
    .section_4{

    }

    .section_4 > .inner{
        flex-direction: column;
    }

    .section_4_left_wrap{
        max-width: none;
        height: 240px;
        background: url(../img/section_4/main_ir_img1.jpg) no-repeat bottom right -250px;
        background-position: bottom right -100px;
        padding: 40px;
    }
}

```



5. css(style) #5 (section_5)

```
.section_5{
    word-wrap: 100%;
    height: 560px;
    background-color: #242424;
}

.section_5 > .inner{
    position: relative;
}

.location_wrap{
    position: absolute;
    top: 0;
    left: 0;
    padding-top: 125px;
}

.location_txt_wrap > h4{
    color: var(--whiteColor);
    font-size: 50px;
    font-family: var(--eng);
}

.location_txt_wrap > h4 > span{
    font-size: 16px;
    font-family: var(--kor);
    opacity: .6;
    display: inline-block;
    margin-left: 5px;
}

.location_txt_wrap > p{
    color: var(--whiteColor);
    opacity: .6;
    margin-top: 25px;
    line-height: 20px;
}

.location_btn_wrap{
    display: flex;
    gap: 30px;
    margin-top: 80px;
}

.location_btn_wrap > a{
    /*
     * 부모요소에 flex를 적용해주고 a 태그에 max-width, width 값을 적용해줄 때 따라
     * a 태그가 block 요소로 바뀔 수 있음
    */
    max-width: 200px;
    width: 50%;
    color: var(--whiteColor);
    padding: 10px;
    display: flex;
    justify-content: space-between;
    border-bottom: 1px solid var(--whiteColor);
}

.location_map{
    /*
     * float:right; 적용 시 location_txt_wrap의 width 영역에 의해 이미지가 오른쪽 상단 정렬 불가
     *
     * location_wrap 내부요소의 가로 정렬
     * 1) section_5에 position : relative; 를 적용하고 location_btn_wrap에 position : absolute 적용
     * 2) location_map에 float: right를 적용해준 뒤 inner에 relative 적용 후 location_wrap에 position: absolute 적용
     *   ※공통속성으로 적용한 inner에 변동성을 최소화 시켜주기 위해서 2번 방법 사용
    */
    float: right;;
    width: 60%;
    background-color: gold;
    display: flex;

    /* float 적용상태로 독립적인 요소이기 때문에 margin으로 위치 조정 가능 */
    margin-top: -50px;
}

/*
 * location_map에 float처리를 하고 location_wrap에 position:absolute 를 적용한 이유
 ==> section_5에 position:relative; 를 적용하고 location_btn_wrap에 position:absolute 적용을 해도 되나
 section_5의 경우 반응형으로 변경 시 높이값에 변동성이 있기 때문에 공통 속성을 최대한 유지할 수 있는 inner와 location_wrap에 position 적용

```

```

/*
.location_map_icon{
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
    width: 110px;
    height: 110px;
    flex-shrink: 0;
    background-color: #1e3a8b;
    color: var(--whiteColor);
}

.location_map_icon > *{
    color: var(--whiteColor);
}

.fa-location-dot{
    display: inline-block;
    margin-bottom: 8px;
}

.location_map_img{
    width: 100%;
    height: 500px;
    background: url(..../img/section_5/map.jfif) no-repeat center / cover;
}

.location_map_icon:hover{
    background-color: var(--mainColor);
}

/* tab 사이즈 적용 */
@media all and (min-width:768px) and (max-width:1023px){

    .section_5{
        /* width: 100%; */
        height: 750px; /*
        display: flex;
        flex-direction: column-reverse;
        /* padding-top: 150px; */
        padding: 100px 0;
        height: initial;
        /* height: initial 또는 height: auto */
    }

    .location_map{
        width: calc(100% - 40px);
        float: initial;
        display: block;
        margin: 0 auto;
        position: relative;
    }

    .location_map_img{
        height: 400px;
        /* max-width: 1340px;
        width: calc(100% - 40px); */
        margin: 0 auto; /*
    }

    .location_map_icon{
        position: absolute;
        top: 0;
        left: 0;
        z-index: 999;
    }

    .location_wrap{
        /* width: 100%; */
        position: static;
        /* float: initial; */
        /* padding: 0; */
        padding-top: 50px;
        /* margin-top: 0; */
    }

    .location_txt_wrap > h4{
        font-size: 34px;
    }

    .location_txt_wrap > h4 > span{
        font-size: 14px;
        font-weight: lighter;
        /*
    }
}

```

```

        font-weight default 값 : 500
        * 500보다 커지면 더 두껍게, 500보다 작으면 더 얇게
    */
}

.tab_hidden{
    display: none;
}

.location_btn_wrap{
    margin-top: 60px;
}
/* .location_txt_wrap{
    position: relative;
    margin-top: 30px; */

/* .location_txt_wrap > p{
    font-size: 14px;
    margin-top: 15px;
}

.location_btn_wrap{
    width: 40%;
    position: absolute;
    bottom: 0;
    right: 10px;
}

.location_btn_wrap > a{
    font-size: 16px;
} */

}

/* mobile 사이즈 적용 */
@media all and (max-width:767px) {

    .section_5{
        /* location_wrap과 location_map을 세로로 나란히 반전 배치 */
        display: flex;
        flex-direction: column-reverse;
    }

    .location_wrap{
        position: static;
        padding: 0px;
    }

    .location_map{
        float: initial;
        margin: initial;
        width: 100%;
    }

    .location_map_icon{
        display: none;
    }

    .location_map_img{
        height: 300px;
        max-width: 1340px;
        /*
            calc : 상대단위에서 절대값을 빼줄 때 사용
            * 연산기호 앞, 뒤에 공백 필수
            * .location_map_img의 경우 PC버전에서 공통설정해놓은 inner의 외부에 존재하는 요소이기 때문에 중앙에 정렬하기 위해서 calc 사용
        */
        width: calc(100% - 40px);
        margin: 0 auto;
    }

    .location_txt_wrap > h4{
        font-size: 26px;
        margin-top: 20px;
    }

    .location_txt_wrap > h4 > span{
        font-weight: lighter;
    }

    .mo_hidden{
        display: none;
    }

    .location_btn_wrap{
        margin-top: 40px;
    }
}

```

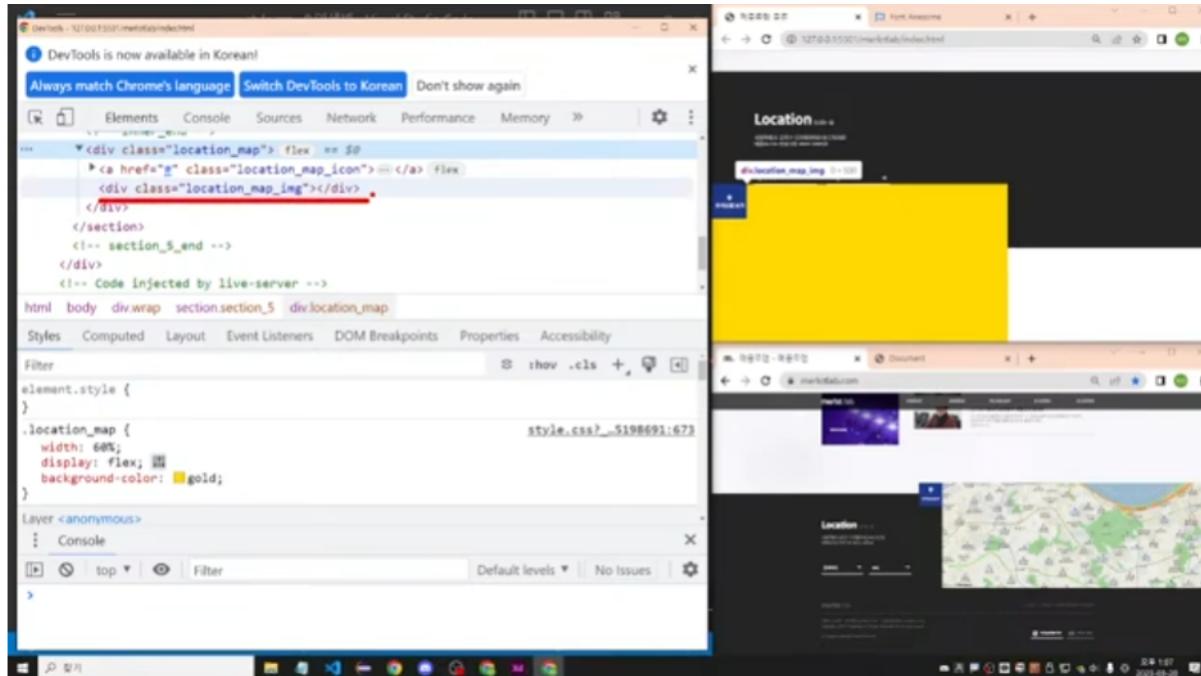
```
}
```

```
/* -----section_5_end----- */
```

※ 삽입한 이미지 반영 안될 경우

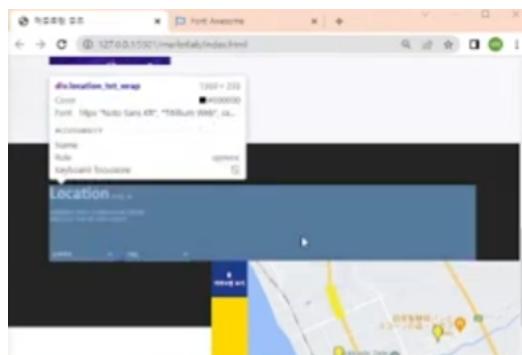
```
.location_map_img{  
    width: 100%;  
    height: 500px;  
    background: url(..../img/section_5/map.jfif) no-repeat center / cover;  
}
```

* 이미지 크기만큼 영역은 잡혀있으나 width 값이 없을 경우 적용이 안될 수 있음



※ location_map에 float : right 적용 시 오른쪽 상단 정렬이 안되는 이유

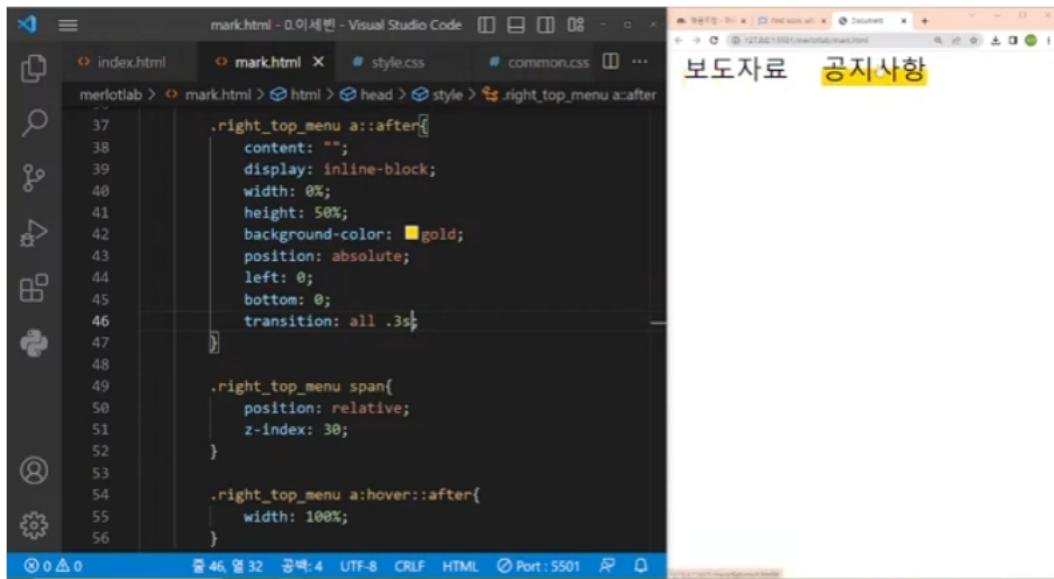
==> location_txt_wrap에 적용된 width 값(1300px) 때문 ==> location_map에 position : absolute 를 적용해서 오른쪽 상단 정렬 (location_map 과 location_wrap의 부모요소인 inner에 position:relative 적용)



※ mark 효과 적용

1) 별도의 div생성을 통해 각각의 div를 감싸는 부모요소에 position: relative 를 적용 후 자식요소에 position: absolute 속성 적용

2) 가상선택자 활용



The screenshot shows a Visual Studio Code interface with several tabs open: index.html, mark.html (the active tab), style.css, and common.css. The code editor displays the following CSS snippet:

```
marklab > mark.html > html > head > style > .right_top_menu a::after
37   .right_top_menu a::after{
38     content: "";
39     display: inline-block;
40     width: 0%;
41     height: 50%;
42     background-color: gold;
43     position: absolute;
44     left: 0;
45     bottom: 0;
46     transition: all .3s;
47   }
48
49   .right_top_menu span{
50     position: relative;
51     z-index: 30;
52   }
53
54   .right_top_menu a:hover::after{
55     width: 100%;
56 }
```

The browser preview window shows a navigation bar with two items: "보도자료" and "공지사항". The "공지사항" item is highlighted with a yellow background color.

5. css(style) #6 (footer)

```
footer{
  background-color: #242424;
  /* width: 100%; */
  height: 200px; /*
  padding: 30px 0 80px;

}

/* footer > .inner{
  width: 100%;
  padding: initial;
} */

.footer_top_wrap{
  width: 100%;
  display: flex;
  justify-content: space-between;
  align-items: center;
  background-color: #242424;
  border-bottom: 1px solid #333;
  padding-bottom: 20px;
}

/* .footer_top_wrap > h1{
  width: 130px;
  opacity: .9;
} */

.footer_logo{
  width: 130px;
}

.footer_top_list{
  display: flex;
}

.footer_top_list a{
  color: var(--whiteColor);
  opacity: .3;
}

/* footer_top_list > li 중 마지막 요소를 제외하고 그 뒤 after 속성 적용 */
.footer_top_list > li:not(:last-child)::after{
  content: "|";
  color: var(--whiteColor);
  display: inline-block;
  margin: 0 9px;
}
```

```

/*
 * footer_top_list > li:not(:last-child)::after{} 대체 방법
 ==> 1-1) .footer_top_list a::after{
     content: "|";
 }

 1-2) footer_top_list의 마지막 자손 a태그에 별도의 클래스명(ex. test) 지정 후
      .test::after{
          display: none;
      }
 */

/* .footer_side_wrap > a{
    display: flex;
    justify-content: space-between;
}

.footer_side_wrap h4{
    font-size: 12px;
    margin-right: 20px;
    color: var(--whiteColor);
    font-family: var(--kor);
    opacity: .3;
} */

/* .footer_side_wrap h4:hover{
    text-decoration: underline;
} */

.footer_top_list a:hover{
    text-decoration: underline;
}

/* -----footer_top_wrap_end----- */

.footer_bottom_wrap{
    width: 100%;
    /* height: 100%; */
    display: flex;
    justify-content: space-between;
    align-items: flex-end;
    padding-top: 20px;
    /* font-size: 12px;
    font-family: var(--kor);
    color: var(--whiteColor);
    line-height: 20px; */

}

.footer_bottom_left_box > ul{
    display: flex;
}

.footer_bottom_left_box > ul > li{
    margin-right: 5px;
    font-size: 14px;
    color: var(--whiteColor);
    opacity: .3;
}

.footer_bottom_left_box a, .copy_right{
    font-size: 14px;
    color: var(--whiteColor);
    opacity: .3;
}

/* .footer_bottom_wrap p{
    opacity: .3;
} */

/* .footer_bottom_left_box > h4{
    margin-top: 30px;
    opacity: .3;
} */

.footer_bottom_left_box address{
    font-size: 14px;
    color: var(--whiteColor);
    /* 기울어진 글자체 원상태로 변경 */
    font-style: normal;
    margin: 5px 0 20px;
    opacity: .3;
}

/* .footer_icon > ul{
    display: flex;
}

```

```

} */

/* .footer_icon{
    display: flex;
    justify-content: space-between;
    margin-top: 50px;
} */

.footer_bottom_right > ul{
    display: flex;
    gap: 20px;
}

.footer_bottom_right a{
    color: var(--whiteColor);
    font-size: 14px;
    display: inline-block;
    /* display: flex;
    justify-content: space-between;
    align-items: center; */
    border-bottom: 1px solid var(--whiteColor);
    /* margin-right: 20px; */
    padding-bottom: 10px;
    align-items: center;
    opacity: .3;
}

.footer_icon_1{
    display: inline-block;
    margin-right: 5px;
    width: 25px;
    height: 25px;
    background: url(..../img/footer/footer_icon1_ov.png) no-repeat center;
}

.footer_icon_2{
    display: inline-block;
    margin-right: 5px;
    width: 30px;
    height: 25px;
    background: url(..../img/footer/footer_icon2_ov.png) no-repeat center;
}

/* .footer_icon > a:last-child{
    opacity: .3;
}

.footer_icon img{
    width: 30px;
    height: 30px;
}

.footer_icon h4{
    color: var(--whiteColor);
    margin-left: 8px;
} */

/* tab 사이즈 적용 */
@media all and (min-width:768px) and (max-width:1023px){

    .footer_top_list a{
        font-size: 14px;
        /*
            * font-default : 16px
            px : 18 ~ 21 (pc)
            px : 18 ~ 19 (tab)
            px : 17      (mobile)
            * footer의 폰트 사이즈는 최대 13px ~ 14px까지만 적용
        */
    }

    .footer_bottom_wrap{
        flex-direction: column;
        align-items: flex-start;
    }

    .footer_top_wrap{
        margin-top: 50px;
    }

    /* .footer_top_wrap a{
        font-size: 12px;
    } */

    .footer_bottom_right{
        margin-top: 50px;
    }
}

```

```

        }

/* .footer_bottom_left_box li{
    margin-left: 10px;
}

.footer_bottom_left_box > address{
    margin-top: 10px;
}

.footer_bottom_left_box > address > .pc_hidden{
    display: none;
}

.footer_bottom_left_box > p{
    font-size: 12px;
} */

}

/* mobile 사이즈 적용 */
@media all and (max-width:767px){

    footer{
        padding: 100px 0;
    }

    .footer_top_wrap{
        flex-direction: column;
        align-items: flex-start;
    }

    .footer_top_list{
        margin-top: 20px;
        width: 100%;
        flex-wrap: wrap;
        justify-content: space-between;
    }

    .footer_top_list > li:not(:last-child)::after{
        content: "";
        display: none;
    }

    .footer_top_list > li{
        width: 49%;
        border: 1px solid gray;
        padding: 8px;
        text-align: center;
    }

    .footer_top_list > li:nth-child(3), .footer_top_list > li:nth-child(4){
        margin-top: 10px;
    }

    .footer_bottom_wrap{
        flex-direction: column;
        align-items: flex-start;
    }

    .footer_bottom_left_box > ul{
        flex-direction: column;
        gap: 5px;
    }

    .footer_bottom_right{
        width: 100%;
    }

    .footer_bottom_right ul{
        gap: 0;
        justify-content: space-between;
        margin-top: 20px;
    }

    .footer_bottom_right li{
        width: 49%;
        flex-shrink: 0;
    }

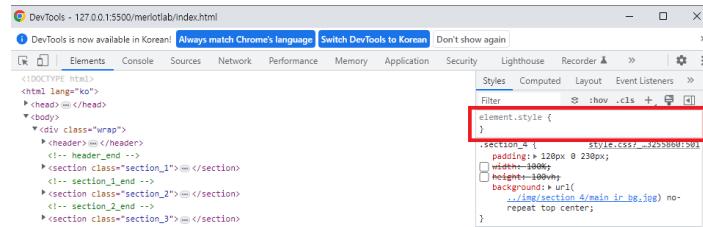
    .footer_bottom_right a{
        border: 1px solid gray;
        width: 100%;
        padding: 8px;
        justify-content: center;
        align-items: center;
    }
}

```

```
}
```

※ 검사항을 통한 속성 적용 테스트

- element.style에 속성 적용 시



- grid

1. grid-container

2. grid-item

[grid-container 속성]

1-1. display:grid

- 그리드 사용 설정
- 바로 형태가 정의 되지 않은 준비 상태

1-2. grid-template-columns / grid-template-rows

- 그리드 형태 정의
- grid-template-columns : 열 배치
- grid-template-rows : 행 배치
- 사용 가능 단위 : px, %, fr, auto
- fr : fraction (비, 비율)
- 단위 혼용 사용 가능

• repeat(반복횟수, 반복값)

==> 반복되는 값을 자동으로 처리

• minmax(최소값, 최대값)

==> ex) minmax(100px, auto)

==> 최소 100px를 확보하고 그보다 커진다면 auto로 늘어나게

• auto-fill, auto-fit

==> 갯수를 미리 정하지 않고 설정된 너비가 허용하는 한 최대한 셀을 채우는 방식

• auto-fill

==> item의 갯수가 모자르면 남는 공간을 비워둠

- **auto-fit**

==> 남는 공간 없이 컨테이너를 꽉 채움

1-3. gap

- 간격
- row-gap
- column-gap
- gap : row column; / row, column의 간격이 다른 경우
- gap : 00px; / row, column의 간격이 같은 경우

※ grid 사용 예시

[속성 사용 예시]

```
<가로>
1-1. grid-template-columns: 200px 200px 400px;
==> 가로방향으로 3개

1-2. grid-template-columns: 200px 50% 400px;
==> 가로방향으로 3개, %단위로 사용 시 현재 컨테이너 크기의 해당 % 만큼 영역 차지

2-1. grid-template-columns: 1fr 1fr 1fr;
==> 가로 3개 반복하되 1:1 비율 #1

2-2. grid-template-columns: repeat(3, 1fr);
==> 가로 3개 반복하되 1:1 비율 #2

3. grid-template-columns: 70px auto;
==> 70px을 확보하고 나머지 영역은 자동으로 채움

4. grid-template-columns: repeat(auto-fill,minmax(20%, auto));
==> 갯수를 정하지 않은 상태로 최소 20%를 확보하여 자동으로 배치
==> 갯수를 정할경우 정해진 갯수의 비율로 적용

5. grid-template-columns: repeat(auto-fit,minmax(20%, auto));
==> auto-fill과 동일한 조건이나 남는공간 없이 배치

6. grid-template-areas : 'header header header';

<div class="item item_1 header">1</div>
<div class="item item_2 header">2</div>
<div class="item item_3 header">3</div>

.header{
    grid-area: header;
}

==> 'header' 이름을 가진 요소 3개를 연결
==> 공통 클래스명을 가진 요소끼리 확장(결합) 배치

<세로>
1. grid-template-rows: 200px 200px 400px ;
==> 세로방향으로 3개

2. grid-template-rows: 1fr 1fr 1fr;
==> 세로 3개 반복하되 1:1 비율

3. grid-template-rows: repeat(3, minmax(100px, auto));
==> 세로방향으로 3개씩 최소 100px를 확보하고 넘는다면 auto(자동)로 늘어나게

4. grid-auto-rows : minmax(100px, auto);
==> 세로방향으로 추가되는 갯수에 상관 없이 100px 크기로 컨테이너 영역에 들어갈 수 있는 만큼 자동 배치

.item:first-child{

grid-column: 1 / span 3;
==> 1번 라인부터 3칸 확장한 4번 라인까지 확장

grid-column-start: 1;
grid-column-end: 3;
* grid-column: 1 / 3;
==> 1번 라인부터 3번 라인까지 가로 확장

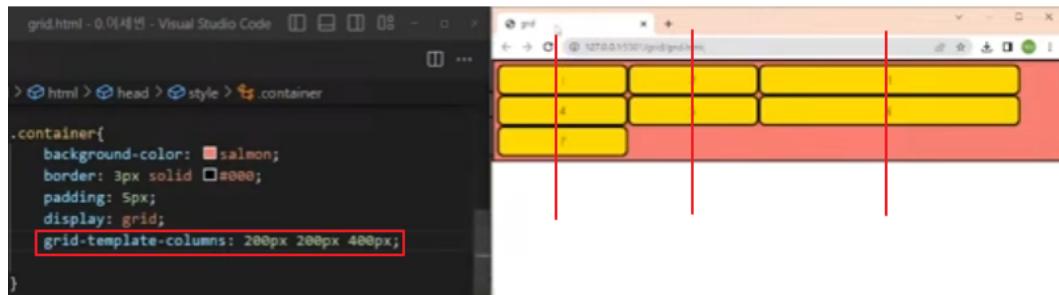
grid-row-start: 1;
```

```

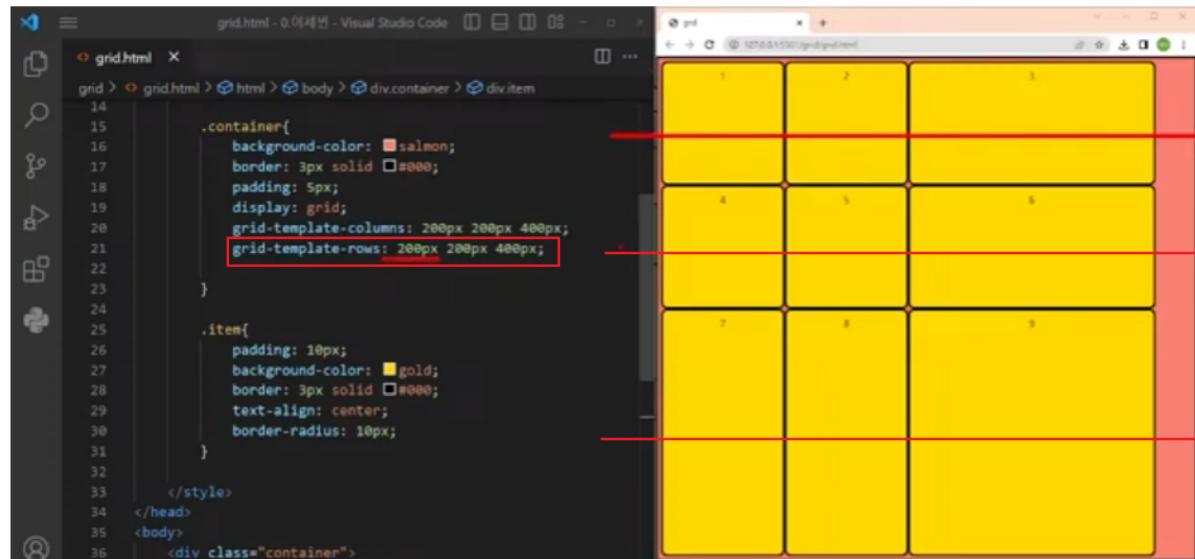
grid-row-end: 3;
* grid-row: 1 / 3;
==> 1번 라인부터 3번 라인까지 세로 확장
}

```

※ grid-template-columns



※ grid-template-rows



- flex-item에 적용하는 속성

1. flex-basis

- flex-item들의 크기 설정
- (flex-direction : row / width)
- (flex-direction : column / height)
- flex-item의 기본 크기 점유
- 사용단위 : px, %, auto(기본값), em, rem

2. flex-grow

- 유연하게 늘리기
- flex-basis의 크기보다 더 커질 수 있는지 결정하는 속성
- 사용단위 : 0 ~ 숫자 (0 이상의 숫자)

- 비율
 - flex-basis를 제외한 여백 부분을 flex-grow에 지정된 숫자 비율로 나눠가짐

※ flex-basis를 지정하지 않은 경우 내부요소의 크기를 제외한 여백을 지정한 비율로 나눠가짐

3. flex-shrink

- 유연하게 줄이기
 - flex-basis의 크기보다 더 작아질 수 있는지 결정하는 속성
 - 사용단위 : 0 ~ 숫자
 - 0보다 큰 값이 셋팅 = 유연한 박스
 - 0 셋팅 = 고정폭이 정해진 박스

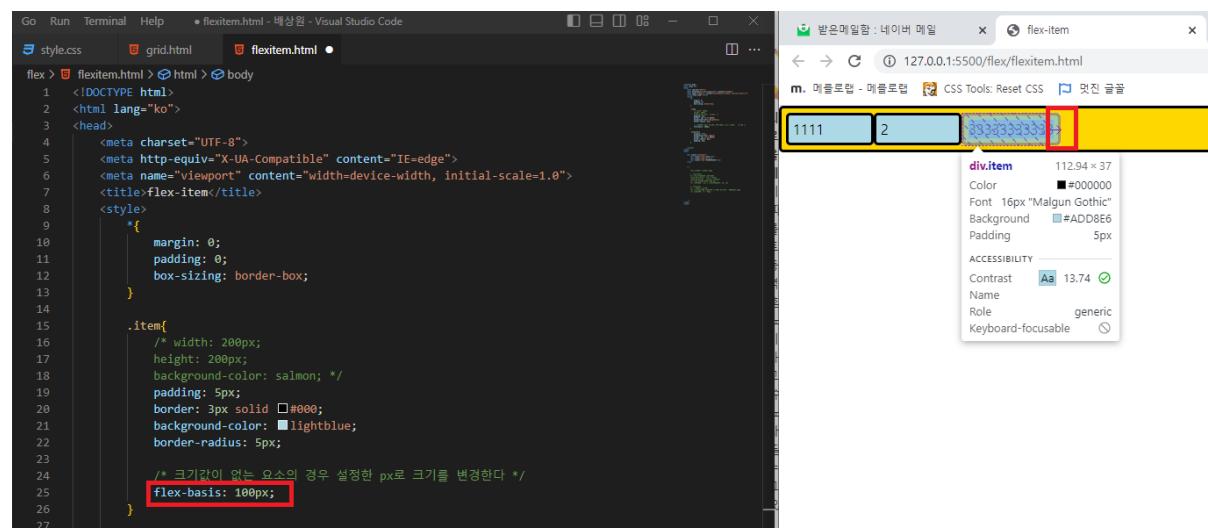
※ **flex-basis**가 있을 경우 설정된 크기를 보장해준다

4. flex

- flex-basis, flex-grow, flex-shrink 축약
 - flex : grow shrink basis;
 - flex : 1;
 - flex-grow : 1; flex-shrink : 1; flex-basis : 0%;
 - flex : 1 1 auto;
 - flex-grow : 1; flex-shrink : 1; flex-basis : auto;

※ flex-basis 예시

최소 100px로 크기를 설정하고 남는 부분 표기



- 절대단위 vs 상대단위

1) 절대단위

$\Rightarrow px$

2) 상대단위

==> 어떤 기준에 따라서 유동적으로 바뀔 수 있는 단위

==> %, vh, vw

- em , rem

[공통점]

==> css font-size를 기준으로해서 결정되는 상대단위

[차이점]

==> 어디에 있는 font-size 속성인지에 따라 차이가 발생

==> em = 해당 단위가 사용되고 있는 요소의 font-size

==> rem = 최상위 요소의 font-size / 최상위 요소 = html/

```
ex)

html {
    font-size : 16px
}

0.5rem = 16px * 0.5 = 8px
1rem = 16 * 1 = 16px
2rem = 16 * 2 = 32px
3rem = 16 * 3 = 48px

width, height, margin, padding, font-size .... 모두 사용 가능

html{
    font-size: 16px;
}

/* root */

.box_1{
    font-size: 20px;
    /* 1em = 20px */
    width: 10em;
    height: 10em;
    /* 각 10 * 20 = 200px */
    background-color: gold;
}

.box_2{
    /* 1rem = 16px */
    width: 10rem;
    height: 10rem;
    /* 각 160px로 계산 */
    background-color: salmon;
}
```

※ Ui Page

1. html

```
<!DOCTYPE html>
<html lang="ko">
<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>ui page</title>
    <link rel="stylesheet" href="./css/style.css"/>
</head>
<body>

    <div class="wrap">

        <header>
            <h1 class="logo">
                <a href="#">UI PAGE</a>
```

```

        </h1>
        <form class="search_form">
            <input type="text" placeholder="검색" />
            <input type="submit" value="찾기" />
        </form>
    </header>
    <!-- header_end -->

    <ul class="menu_wrap">
        <li class="menu_item">
            <a href="#">Home</a>
        </li>
        <li class="menu_item">
            <a href="#">About</a>
        </li>
        <li class="menu_item">
            <a href="#">Product</a>
        </li>
        <li class="menu_item">
            <a href="#">Contact</a>
        </li>
    </ul>
    <!-- menu_wrap_end -->

    <section class="main_con">
        <ul class="card_list">
            <li class="card_item">
                <!--
                    DB 또는 특정 경로에서 데이터 불러오기방식으로 이미지를 지정해야할 경우
                    불러온 이미지는 figure class="card_img" 태그 안쪽으로 인라인 스타일로 반영된다
                <!--

                    따라서 불러온 이미지를 따로 분리하기 어렵기 때문에 태그 뒤에 style을 붙여서 백그라운드 이미지 형태로 가져온다
                <-->
                <figure class="card_img" style="background-image: url(img/mimoticon_1.png);">
                    <!-- 요소를 중앙정렬할 경우 img 태그보다 백그라운드 이미지 방식으로 사용하는 것을 추천 -->
                    <!-- 이미지 태그의 경우 단독으로 사용하지 않고 부모요소에 넣어주고 width:100%를 적용하여 사용 -->
                    
                </figure>
                <div class="card_desc">
                    Lorem ipsum dolor sit amet consectetur adipisicing elit. Voluptates maxime blanditiis non consectetur dolor
                    doloribus aspernatur nobis, cumque id commodi excepturi iusto earum quam tempora enim incidentur debitis. Odio, veniam?
                </div>
            </li>

            <li class="card_item">
                <figure class="card_img" style="background-image: url(img/mimoticon_2.png);">
                    
                </figure>
                <div class="card_desc">
                    화요일!!
                </div>
            </li>

            <li class="card_item">
                <figure class="card_img" style="background-image: url(img/mimoticon_3.png);">
                    
                </figure>
                <div class="card_desc">
                    Lorem ipsum dolor sit amet consectetur adipisicing elit. Voluptates maxime blanditiis non consectetur dolor
                    doloribus aspernatur nobis, cumque id commodi excepturi iusto earum quam tempora enim incidentur debitis. Odio, veniam?
                </div>
            </li>

            <li class="card_item">
                <figure class="card_img" style="background-image: url(img/mimoticon_4.png);">
                    
                </figure>
                <div class="card_desc">
                    Lorem ipsum dolor sit amet consectetur adipisicing elit. Voluptates maxime blanditiis non consectetur dolor
                    doloribus aspernatur nobis, cumque id commodi excepturi iusto earum quam tempora enim incidentur debitis. Odio, veniam?
                </div>
            </li>

            <li class="card_item">
                <figure class="card_img" style="background-image: url(img/mimoticon_5.png);">
                    
                </figure>
                <div class="card_desc">
                    Lorem ipsum dolor sit amet consectetur adipisicing elit. Voluptates maxime blanditiis non consectetur dolor
                    doloribus aspernatur nobis, cumque id commodi excepturi iusto earum quam tempora enim incidentur debitis. Odio, veniam?
                </div>
            </li>

            <li class="card_item">
                <figure class="card_img" style="background-image: url(img/mimoticon_1.png);">
                    
                </figure>
            </li>
        </ul>
    </section>

```

```

        <div class="card_desc">
            Lorem ipsum dolor sit amet consectetur adipisicing elit. Voluptates maxime blanditiis non consectetur dolor
            doloribus aspernatur nobis, cumque id commodi excepturi iusto earum quam tempora enim incidentur debitis. Odio, veniam?
        </div>
    </li>

    <li class="card_item">
        <figure class="card_img" style="background-image: url(img/mimoticon_1.png);">
            
        </figure>
        <div class="card_desc">
            Lorem ipsum dolor sit amet consectetur adipisicing elit. Voluptates maxime blanditiis non consectetur dolor
            doloribus aspernatur nobis, cumque id commodi excepturi iusto earum quam tempora enim incidentur debitis. Odio, veniam?
        </div>
    </li>
</ul>

</section>
<!-- main_con_end -->

<!-- 공통항목과 독립된 항목을 각각 지정해주기 위해 클래스명 2개 지정 -->
<div class="side side_1">
    <ul class="todo_list">
        <li class="todo_item">할일 1</li>
        <li class="todo_item">할일 2</li>
        <li class="todo_item">할일 3</li>
    </ul>
</div>
<!-- side_1_end -->

<div class="side side_2">
    <ul class="message_list">
        <li class="messge_item">
            <!-- 프로필의 경우에도 대부분 데이터에서 불러오는 경우가 많기 때문에 백그라운드 이미지 형태로 사용 -->
            <figure class="user_photo" style="background-image: url(img/mimoticon_2.png)"></figure>
            <p class="message_con">오늘은 화요일 :D</p>
        </li>
        <li class="messge_item">
            <figure class="user_photo" style="background-image: url(img/mimoticon_3.png)"></figure>
            <p class="message_con">오늘은 화요일 :D</p>
        </li>
        <li class="messge_item">
            <figure class="user_photo" style="background-image: url(img/mimoticon_4.png)"></figure>
            <p class="message_con">오늘은 화요일 :D</p>
        </li>
    </ul>
</div>
<!-- side_2_end -->

<footer>
    copyright all right reserved sebhin.
</footer>
</div>

</body>
</html>

```

2. css (style)

```

*{
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

a{
    text-decoration: none;
    color: initial;
}

ul{
    list-style: none;
}

/* reset */

header{
    width: 100%;
    height: 80px;
    background-color: #eee;
    padding: 0 15px;
}

```

```

        display: flex;
        justify-content: space-between;
        align-items: center;
        grid-area: header;
    }

    .logo > a{
        font-size: 30px;
    }

    /* 클래스 이름이 지정되지 않은 input 호출 */
    .search_form > input[type="text"]{
        border: none;
        padding: 8px 0;

        /* 16px * 0.3 = 4.8px */
        border-radius: 0.3rem;

        /* 16px * 0.8 = 12.8px */
        font-size: 0.8rem;
    }

    .search_form > input[type="submit"]{
        border: none;
        background-color: gold;
        width: 4rem;
        padding: 0.5rem 0;
        border-radius: 0.3rem;
        cursor: pointer;
    }

    /* header_end */

    .menu_wrap{
        width: 100%;
        display: flex;
        grid-area: menu;
    }

    .menu_wrap > li{
        width: 25%;
        background-color: gold;
        transition: all .5s;
    }

    .menu_wrap > li > a{
        display: block;
        padding: 1rem;
        text-align: center;
        font-size: 1.2rem;
        font-weight: bold;
        color: #555;
    }

    .menu_wrap > li:hover{
        background-color: crimson;
        width: 35%;
    }

    .menu_wrap > li:hover a{
        color: #fff;
    }

    /* menu_wrap_end */

    .main_con{
        background-color: lightblue;
        padding: 15px;
        grid-area: main;
    }

    .card_list{
        width: 100%;
        display: flex;
        flex-wrap: wrap;
    }

    .card_item{
        /* 소수점 이하 3자리까지만 반영됨 */
        width: 33.333%;

        /* card_item 자식(item) 요소를 flex-item으로 만들어 주고, flex-item에 flex-basis , flex-grow 옵션을 적용하기 위한 선행작업 */
        display: flex;
        flex-direction: column;
        padding: 0 15px;
    }

```

```

/* background-color: salmon; */
margin-bottom: 30px;
}

.card_img img{
    display: none;
}

/* 백그라운드 이미지 스타일이 반영된 클래스명(card_img)을 호출하여 css 적용 */
.card_img{
    height: 300px;
    background-repeat: no-repeat;
    background-position: center;
    background-size: cover;
    background-color: lightgrey;
}

.card_desc{
    width: 100%;
    background-color: #fff;
    padding: 15px;
    /* flex-grow: 1; */
    flex: 1 1 auto;
    /* flex-grow:1 , flex-shirnk:1, flex-basis:auto */
}

/* main_con_end */

.side{
    padding: 15px;
}

.side_1{
    background-color: darkgray;
    grid-area: side_1;
}

.todo_item::before{
    content: "📌";
}

/* side_1_end */

.side_2{
    background-color: #333;
    color: #fff;
    grid-area: side_2;
}

.message_item{
    width: 100%;
    display: flex;
    /* user_photo에 width, height 값이 지정되어 있기 때문에 중앙정렬 가능 */
    align-items: center;
    margin-bottom: 15px;
    margin-right: 5px;
}

.user_photo{
    /*
        width, height를 지정해줘도 이미지가 나오지 않는 경우 이미지 사이즈에 비해 이미지를 담는 영역을 작게 설정하였기 때문
        background-repeat, background-position, background-size 옵션을 넣어줘야 정상적으로 반영 됨
    */
    width: 50px;
    height: 50px;
    background-color: gold;
    border-radius: 50%;
    background-repeat: no-repeat;
    background-position: center;
    background-size: cover;
}

/* side_2_end */

footer{
    padding: 30px 0;
    text-align: center;
    grid-area: footer;
}

/* default = tab, mobile */

/* pc 사이즈 설정 */
@media all and (min-width:1024px){

    .wrap{

```

```

        display: grid;
        grid-template-columns: 20% 1fr 20%;
        grid-template-areas:
        'header header header'
        'menu menu menu'
        'side_1 main side_2'
        'footer footer footer';
    }

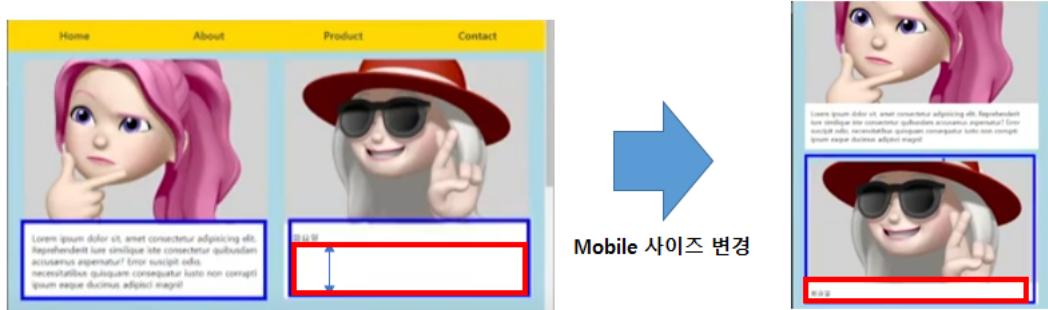
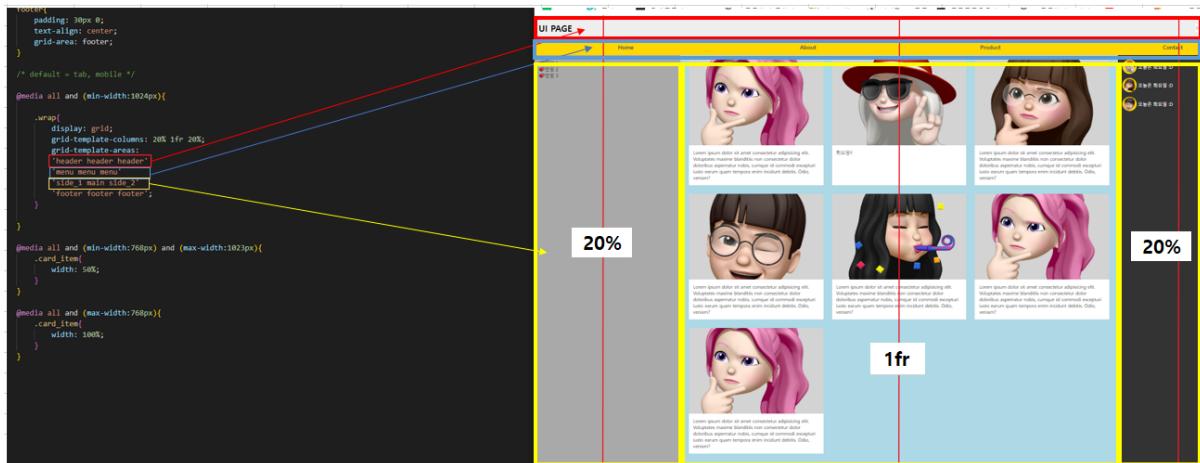
}

/* tab 사이즈 설정 */
@media all and (min-width:768px) and (max-width:1023px){
    .card_item{
        width: 50%;
    }
}

/* mobile 사이즈 설정 */
@media all and (max-width:768px){
    .card_item{
        width: 100%;
    }
}

```

※ grid 적용 예시



flex-basis : auto 적용의 경우 바로 옆에 비교 대상이 존재할 경우 비교대상의 텍스트 높이값에 맞춰서 사이즈가 늘어난다
모바일 버전으로 사이즈가 변경되면서 바로 옆에 비교 대상이 없어지게 되면 본연의 텍스트 높이 값으로 적용 된다

※ kakao 클론

★ css 연결

```

# reset.css      # style.css    index.html
# 커스터마이징 css > # style.css
1 @import url(reset.css);
2
3 /* screen */
4 @import url./screen/login.css;
5 @import url./screen/mobile_only.css;
6
7
8
9 /*
10 [css 연결 및 폴더 관리]
11
12 css / screen(화면) 폴더 생성
13
14 css (root folder)
15 - reset.css
16 - style.css ←
17
18 screen folder
19 - login.css
20
21
22 */
23 */

# style.css
# 커스터마이징
1 @import url(reset.css);
2
3 /* screen */
4 @import url(screen/login.css);
5 @import url(screen/no_mobile.css);
6 @import url(screen/friends.css);
7 @import url(screen/chatting.css);
8 @import url(screen/chatting.css);
9 @import url(screen/find.css);
10
11

```

1. html #1 (index)

```

<!DOCTYPE html>
<html lang="ko">
<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>카카오톡 클론 코딩</title>
    <!--
        .js(자바스크립트)의 경우 통상적으로 body 가장 하단부에 적어준다
        폰트어썸의 경우 head 안쪽에 넣는 것을 권고사항으로 지정하였기 때문에 head 내부에 넣은 것임
    -->
    <script src="https://kit.fontawesome.com/8c36059e36.js" crossorigin="anonymous"></script>
    <link rel="stylesheet" href="css/style.css">
</head>
<body>

    <div class="wrap">
        <header>
            <div class="status_bar">
                <span>LG U</span>
                <i class="fa-solid fa-wifi"></i>
                <!-- 와이파이 아이콘 -->
            </div>
            <div class="status_bar">
                <span>12:54pm</span>
            </div>
            <div class="status_bar">
                <span>90%</span>
                <i class="fa-solid fa-battery-three-quarters"></i>
                <!-- 배터리 아이콘 -->
            </div>
        </header>
        <!-- header_end -->

        <div class="welcome">
            <h1>가입을 시작합니다!</h1>
            <p>
                카카오 계정 하나로 KaKao, Daum, Melon 서비스를 모두 이용할 수 있어요.
            </p>
        </div>
        <!-- welcome_end -->

        <form action="friends.html" class="login_form ">
            <input type="text" placeholder="이메일 혹은 전화번호를 입력하세요">
            <input type="password" placeholder="비밀번호를 입력해주세요">
            <input type="submit" value="Log in">
            <a href="#">카카오톡 아이디를 잊어버리셨나요?</a>
        </form>
        <!-- login_form_end -->

        <div class="mobile_only">
            <span>화면 크기를 줄여주세요 :D</span>
        </div>
    </div>

```

```
</body>
</html>
```

1. html #2 (friends)

```
<!DOCTYPE html>
<html lang="ko">
<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>친구</title>
    <script src="https://kit.fontawesome.com/8c36059e36.js" crossorigin="anonymous"></script>
    <link rel="stylesheet" href="css/style.css">
</head>
<body>
    <div class="wrap">
        <header>
            <div class="status_bar">
                <span>LG U</span>
                <i class="fa-solid fa-wifi"></i>
                <!-- 와이파이 아이콘 -->
            </div>
            <div class="status_bar">
                <span>12:54pm</span>
            </div>
            <div class="status_bar">
                <span>90%</span>
                <i class="fa-solid fa-battery-three-quarters"></i>
                <!-- 배터리 아이콘 -->
            </div>
        </header>
        <!-- header_end -->

        <div class="top_wrap">
            <h1>친구</h1>
            <div class="friend_icon">
                <span>
                    <i class="fa-solid fa-magnifying-glass"></i>
                    <!-- 둘보기 -->
                </span>
                <span>
                    <i class="fa-solid fa-user-plus"></i>
                    <!-- 프로필 -->
                </span>
                <span>
                    <i class="fa-solid fa-music"></i>
                    <!-- 음악 -->
                </span>
                <span>
                    <i class="fa-solid fa-gear"></i>
                    <!-- 설정 -->
                </span>
            </div>
        </div>
        <a href="#" class="find_friends">
            <i class="fa-solid fa-circle-info"></i>
            <!-- 인포메이션 아이콘 -->
            친구를 찾아보세요
            <i class="fa-solid fa-chevron-right"></i>
            <!-- chevron -->
        </a>
        <div class="profile">
            <div class="user_profile">
                <figure class="user_profile_img"></figure>
                <h4>상원</h4>
            </div>
        </div>
        <div class="friends_list_wrap">
            <p>친구</p>
            <div class="friend_list">
                <div class="friends_profile">
                    <figure class="friends_profile_img"></figure>
                    <p class="friends_name">친구1</p>
                </div>
                <div class="friends_num">
                    <span>2</span>
                    <i class="fa-solid fa-chevron-right"></i>
                    <!-- chevron -->
                </div>
            </div>
        </div>
    </div>
</body>
```

```

        </div>
    </div>
</div>

<nav class="bottom_nav">
    <ul class="nav_list">
        <li class="nav_btn">
            <a href="friends.html">
                <i class="fa-solid fa-user"></i>
                <!-- 프로필 -->
            </a>
        </li>
        <li class="nav_btn">
            <a href="chatting.html">
                <span class="alert">1</span>
                <i class="fa-regular fa-comment"></i>
                <!-- 말풍선 -->
            </a>
        </li>
        <li class="nav_btn">
            <a href="#">
                <i class="fa-solid fa-magnifying-glass"></i>
                <!-- 둘보기 -->
            </a>
        </li>
        <li class="nav_btn">
            <a href="more.html">
                <i class="fa-solid fa-ellipsis"></i>
                <!-- 더보기 -->
            </a>
        </li>
    </ul>
</nav>

<div class="splash">
    <i class="fa-solid fa-comment"></i>
</div>

<div class="mobile_only">
    <span>화면 크기를 줄여주세요 :D</span>
</div>
</div>
</body>
</html>

```

1. html #3 (chatting)

```

<!DOCTYPE html>
<html lang="ko">
<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>chatting</title>
    <script src="https://kit.fontawesome.com/8c36059e36.js" crossorigin="anonymous"></script>
    <link rel="stylesheet" href="css/style.css">
</head>
<body>

    <div class="wrap">
        <header>
            <div class="status_bar">
                <span>LG U+</span>
                <i class="fa-solid fa-wifi"></i>
                <!-- 와이파이 아이콘 -->
            </div>
            <div class="status_bar">
                <span>12:54pm</span>
            </div>
            <div class="status_bar">
                <span>90%</span>
                <i class="fa-solid fa-battery-three-quarters"></i>
                <!-- 배터리 아이콘 -->
            </div>
        </header>
        <!-- header_end -->

        <div class="top_wrap">
            <h1>채팅</h1>
            <div class="friend_icon">
                <span>
                    <i class="fa-solid fa-magnifying-glass"></i>

```

```

        <!-- 둘보기 -->
    </span>
    <span>
        <i class="fa-solid fa-user-plus"></i>
        <!-- 프로필 -->
    </span>
    <span>
        <i class="fa-solid fa-music"></i>
        <!-- 음악 -->
    </span>
    <span>
        <i class="fa-solid fa-gear"></i>
        <!-- 설정 -->
    </span>
</div>
</div>

<!-- <div class="chat_list_wrap">
<a href="#">
    <div class="chat_list">
        <figure class="chat_img"></figure>
        <div class="chat_message">
            <h4>friend1</h4>
            <p>몇시에 만날래?</p>
        </div>
    </div>
    <div class="chat_alert">
        <p>12:26</p>
        <span class="alert_1">1</span>
    </div>
</a>
</div> -->

<div class="chat_main">
<a href="chatting_room.html">
    <div class="chat_profile">
        <figure></figure>
        <div class="chat_txt">
            <h4>친구1</h4>
            <p>몇시에 만날래?</p>
        </div>
    </div>
    <div class="chat_alert">
        <p>12:54</p>
        <div class="badge">1</div>
    </div>
</a>
</div>

<nav class="bottom_nav">
<ul class="nav_list">
    <li class="nav_btn">
        <a href="friends.html">
            <i class="fa-solid fa-user"></i>
            <!-- 프로필 -->
        </a>
    </li>
    <li class="nav_btn">
        <a href="chatting.html">
            <span class="alert">1</span>
            <i class="fa-solid fa-comment"></i>
            <!-- 말풍선 -->
        </a>
    </li>
    <li class="nav_btn">
        <a href="#">
            <i class="fa-solid fa-magnifying-glass"></i>
            <!-- 둘보기 -->
        </a>
    </li>
    <li class="nav_btn">
        <a href="more.html">
            <i class="fa-solid fa-ellipsis"></i>
            <!-- 더보기 -->
        </a>
    </li>
</ul>
</nav>

<div class="mobile_only">
    <span>화면 크기를 줄여주세요 :D</span>
</div>
</div>
</body>
</html>

```

1. html #4 (chatting_room)

```
<!DOCTYPE html>
<html lang="ko">
<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>chatting</title>
    <script src="https://kit.fontawesome.com/8c36059e36.js" crossorigin="anonymous"></script>
    <link rel="stylesheet" href="css/style.css">
</head>
<body>

    <div class="chat_room_wrap">
        <header>
            <div class="status_bar">
                <span>LG U</span>
                <i class="fa-solid fa-wifi"></i>
                <!-- 와이파이 아이콘 -->
            </div>
            <div class="status_bar">
                <span>12:54pm</span>
            </div>
            <div class="status_bar">
                <span>90%</span>
                <i class="fa-solid fa-battery-three-quarters"></i>
                <!-- 배터리 아이콘 -->
            </div>
        </header>

        <div class="chat_header">
            <div class="chat_con">
                <a href="chatting.html">
                    <i class="fa-solid fa-chevron-left"></i>
                </a>
            </div>

            <div class="chat_con">
                <h2>친구1</h2>
            </div>

            <div class="chat_con">
                <i class="fa-solid fa-magnifying-glass"></i>
                <i class="fa-solid fa-bars"></i>
            </div>
        </div>

        <div class="chatting">
            <div class="date">2023.03.30 목요일</div>
            <div class="message message_1">
                <figure class="m_profile"></figure>
                <div class="m_con">
                    <p class="m_name">친구1</p>
                    <div class="m_info">
                        <div class="m_bubble">안녕!</div>
                        <div class="m_time">13:05</div>
                    </div>
                </div>
            </div>
            <div class="message message_2">
                <div class="m_info_2">
                    <div class="m_bubble_2">몇시에 어디서 만날래??</div>
                    <div class="m_time_2">13:16</div>
                </div>
            </div>
        </div>

        <!-- 내부에 input 태그가 들어가는 경우 활용 -->
        <form class="reply">
            <div class="reply_con">
                <!-- 더하기 아이콘 -->
                <i class="fa-solid fa-plus"></i>
            </div>

            <div class="reply_con">
                <input type="text" placeholder="채팅 내용을 입력하세요">
                <span class="reply_wink">
                    <!-- 이모티콘 얼굴 -->
                    <i class="fa-solid fa-face-smile"></i>
                </span>
            </div>
        </form>
    </div>
</body>
```

```

        <button>
            <!-- 위쪽 화살표 -->
            <i class="fa-solid fa-arrow-up"></i>
        </button>
    </div>
</form>

<!-- <div class="chat_top_wrap">
    <span>
        <i class="fa-solid fa-chevron-left"></i> -->
        <!-- chevron -->
    <!-- </span>
    <h1>친구1</h1>
    <div class="chat_top_right"> -->
        <!-- 둘보기 -->
        <!-- <i class="fa-solid fa-magnifying-glass"></i> -->
        <!-- 메뉴바 -->
        <!-- <i class="fa-solid fa-bars"></i>
    </div>
</div>

<div class="chat_center_wrap">
    <figure></figure>
    <h4>친구1</h4>
    <h4>안녕!</h4>
    <div class="chat_day">2020.06.08 화요일</div>
    <div class="chat_msg">몇시에 어디서 만날래??</div>
</div>
<div class="chat_bottom_wrap"> -->
    <!-- 더하기 아이콘 -->
    <!-- <i class="fa-solid fa-plus"></i>
    <input type="text" placeholder="채팅 내용을 입력하세요."> -->
    <!-- 이모티콘 얼굴 -->
    <!-- <i class="fa-solid fa-face-smile"></i> -->
    <!-- 위쪽 화살표 -->
    <!-- <i class="fa-solid fa-arrow-up"></i>
</div> -->

<div class="mobile_only">
    <span>화면 크기를 줄여주세요 :D</span>
</div>
</div>
</body>
</html>

```

1. html #5 (more)

```

<!DOCTYPE html>
<html lang="kr">
<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>더보기</title>
    <script src="https://kit.fontawesome.com/8c36059e36.js" crossorigin="anonymous"></script>
    <link rel="stylesheet" href="css/style.css">
</head>
<body>
    <div class="wrap">
        <header>
            <div class="status_bar">
                <span>LG U+</span>
                <i class="fa-solid fa-wifi"></i>
                <!-- 와이파이 아이콘 -->
            </div>
            <div class="status_bar">
                <span>12:54pm</span>
            </div>
            <div class="status_bar">
                <span>90%</span>
                <i class="fa-solid fa-battery-three-quarters"></i>
                <!-- 배터리 아이콘 -->
            </div>
        </header>
        <div class="top_wrap">
            <h1>더보기</h1>
            <div class="friend_icon">
                <span>
                    <i class="fa-solid fa-magnifying-glass"></i>
                    <!-- 둘보기 -->
                </span>
            </div>
        </div>
    </div>

```

```


    <i class="fa-solid fa-music"></i>
    <!-- 음악 -->


    <i class="fa-solid fa-gear"></i>
    <!-- 설정 -->


```

```

</div>
</div>

<div class="more_main">
    <div class="user_component">
        <div class="user_info">
            <figure></figure>
            <div class="user_component_txt">
                <h4>상원</h4>
                <p>+555 666 777</p>
            </div>
        </div>
        <div class="user_bubble">
            <!-- 네모밀풍선 -->
            <i class="fa-regular fa-message"></i>
        </div>
    </div>
</div>

```

```

<ul class="icon_list">
    <li class="icon">
        <i class="fa-regular fa-calendar"></i>
        <!-- 달력 이모티콘 -->
        <span>달력</span>
    </li>

    <li class="icon">
        <i class="fa-regular fa-face-smile-beam"></i>
        <!-- 스마일 -->
        <span>이모티콘</span>
    </li>

    <li class="icon">
        <i class="fa-solid fa-paint-roller"></i>
        <!-- 테마 -->
        <span>테마</span>
    </li>

    <li class="icon">
        <i class="fa-solid fa-file-invoice-dollar"></i>
        <!-- 계좌 -->
        <span>계좌</span>
    </li>
</ul>

```

```

<div class="suggestion">
    <p>제안</p>
    <div class="suggestion_icons">
        <div class="s_icon">
            <div class="icon_box">
                <!-- 따옴표 -->
                <i class="fa-solid fa-quote-right"></i>
            </div>
            <span>카카오 스토리</span>
        </div>
        <div class="s_icon">
            <div class="icon_box">
                <!-- 따옴표 -->
                <i class="fa-solid fa-quote-right"></i>
            </div>
            <span>카카오 스토리</span>
        </div>
    </div>
</div>

```

```

<!-- <div class="more_top_wrap">
    <h1>더보기</h1>
    <div class="more_top_icon">
        <span>
            <i class="fa-solid fa-magnifying-glass"></i> -->
            <!-- 둘보기 -->
        </span>
        <span>
            <i class="fa-solid fa-music"></i> -->
            <!-- 음악 -->
        </span>
        <span>
            <i class="fa-solid fa-gear"></i> -->
            <!-- 설정 -->
        </span>
    </div>
-->

```

```

        <!-- </span>
    </div>
</div> -->

<!-- <div class="more_center_wrap">
    <figure class="more_profile_img"></figure>
    <div class="more_center_txt">
        <h4>상원</h4>
        <p>+555 666 777</p>
    </div>
    <span>
        <i class="fa-regular fa-message"></i> -->
        <!-- 네모 말풍선-->
    </span>
</div> -->

<!-- <div class="more_center_icon_wrap">
    <span>
        <i class="fa-regular fa-calendar"></i> -->
        <!-- 달력 -->
    </span>
    <span>
        <i class="fa-regular fa-face-smile-beam"></i> -->
        <!-- 스마일 -->
    </span>
    <span>
        <i class="fa-solid fa-paint-roller"></i> -->
        <!-- 테마 -->
    </span>
    <span>
        <i class="fa-solid fa-file-invoice-dollar"></i> -->
        <!-- 계좌 -->
    </span>
</div>

<div class="more_center_box">
    <p>제안</p>
    <div class="center_box_quote">
        <span>
            <i class="fa-solid fa-quote-right"></i> -->
            <!-- 따옴표 -->
        </span>
        <p>Kokoa Story</p>
        <span>
            <i class="fa-solid fa-quote-right"></i> -->
            <!-- 따옴표 -->
        </span>
        <p>Kokoa Story</p>
    </div>
</div> -->

<nav class="bottom_nav">
    <ul class="nav_list">
        <li class="nav_btn">
            <a href="friends.html">
                <i class="fa-solid fa-user"></i>
                <!-- 프로필 -->
            </a>
        </li>
        <li class="nav_btn">
            <a href="chatting.html">
                <span class="alert">1</span>
                <i class="fa-solid fa-comment"></i>
                <!-- 말풍선 -->
            </a>
        </li>
        <li class="nav_btn">
            <a href="#">
                <i class="fa-solid fa-magnifying-glass"></i>
                <!-- 둡보기 -->
            </a>
        </li>
        <li class="nav_btn">
            <a href="more.html">
                <i class="fa-solid fa-ellipsis"></i>
                <!-- 더보기 -->
            </a>
        </li>
    </ul>
</nav>

<div class="mobile_only">
    <span>화면 크기를 줄여주세요 :D</span>
</div>
</div>

```

```
</body>
</html>
```

2. css > reset

```
* {
    outline: none;
    -webkit-box-sizing: border-box;
    -moz-box-sizing: border-box;
    box-sizing: border-box; }

.root{
    --mainColor : #fae100;
    --grayColor : rgb(206,206,206);
}

html {
    height: 100%; }

html, body {
    width: 100%;
    height: 100%;
    margin: 0;
    padding: 0; }

body, div, dl, dt, dd, ul, ol, li, h1, h2, h3, h4, h5, h6, pre, code, form, fieldset, legend, input, button, textarea, p, blockquote, th, td, select, label, img, figure {
    margin: 0;
    padding: 0; }

button {
    background: none;
    border: 0;
    font-size:1em;
    outline:none;
}
.btn:focus, .btn:active:focus, .btn.active:focus, .btn.focus, .btn:active.focus, .btn.active.focus {outline:none}
input,textarea,select {outline:0; outline:none; border:0; border:none; vertical-align:middle; font-size:1em; font-family: 'Nanum Squarebold'; color:#000}

table {
    border-collapse: collapse;
    width: 100%;
    background: #fff;
    margin: 0;
    padding: 0; }

fieldset, img, table, button {
    border: 0; }

ul, ol, li {
    list-style: none; }

img {
    width:100%;
    vertical-align: middle; }

caption, legend {
    display: none; }

h1, h2, h3, h4, h5, h6 {
    font-size: 100%;
    font-weight: bold; }

address, em {
    font-style: normal;
    padding: 0;
    margin: 0; }

article, aside, details, figcaption, figure, footer, header, hgroup, menu, navi, section {
    display: block; }

legend, hr {
    display: none;
    height: 0;
    font-size: 0; }

a {
    text-decoration: none; }

p {
```

```
padding: 0; }

button {
  color: #555;
  text-decoration: none;
  cursor: pointer; }
```

2. css > style

```
@import url(reset.css);

/* screen */
@import url("./screen/login.css");
@import url("./screen/friends.css");
@import url("./screen/mobile_only.css");
@import url("./screen/chatting.css");
@import url("./screen/chatting_room.css");
@import url("./screen/more.css");

/*
 [css 연결 및 폴더 관리]

 css / screen(화면) 폴더 생성

css (root folder)
- reset.css
- style.css

screen folder
- login.css

*/
```

2. css > screen > login

```
header{
  display: flex;
  justify-content: space-between;
  padding: 5px 3px;
  /* background-color: gold; */
}

.status_bar{
  width: 33%;
}

.status_bar:nth-child(2){
  font-size: 14px;
  text-align: center;
}

.status_bar:last-child{
  font-size: 14px;

  /* 상단 우측 끝으로 정렬을 위한 선행 작업 */
  display: flex;
  justify-content: flex-end;
  align-items: center;
  gap: 5px;
}

/* header_end */

.welcome{
  padding: 60px 0;
  text-align: center;
}

.welcome > h1{
  font-size: 25px;
}

.welcome > p{
  width: 60%;
  /* width 값이 있으므로 margin 사용 가능 */
  margin: 20px auto 0;
  opacity: .7;
```

```

}

.login_form{
  display: flex;
  flex-direction: column;
  padding: 0 30px;
}

.login_form input{
  border: none;
  padding: 15px 0;
  margin-bottom: 25px;
}

/* login_form input 안에 type="submit"이 아닌 요소 선택 */
.login_form input:not([type="submit"]){
  border-bottom: 1px solid var(--grayColor);
}

.login_form input[type="submit"]){
  background-color: var(--mainColor);
  border-radius: 10px;
  cursor: pointer;
}

.login_form a{
  font-size: 14px;
  text-align: center;
  color: #777;
}

```

2. css > screen > friends

```

.top_wrap{
  width: 100%;
  padding: 25px;
  display: flex;
  justify-content: space-between;
  align-items: center;
  /* background-color: salmon; */
}

.top_wrap > h1{
  font-size: 30px;
}

.friend_icon > span{
  display: inline-block;
  margin-left: 15px;
}

/* i 태그의 경우 width, height 값 적용으로 이미지 사이즈를 키우는 방식이 아닌 font-size를 통해 크기를 조정 */
.friend_icon i{
  font-size: 22px;
  transition: all .5s;
}

/* 계속적으로 동작을 유지해야 할 경우 애니메이션 사용 */
.friend_icon .fa-gear:hover{
  animation: rotateSetting 1s linear infinite;
}

@keyframes rotateSetting {
  to{
    transform: rotateZ(360deg);
  }
}

.find_friends{
  display: block;
  background-color: #fafafa;
  padding: 15px 0;
  text-align: center;
  color: rgba(0,0,0,.5);
  font-size: 15px;
}

.profile{
  padding: 15px 25px;
}

```

```

.user_profile{
  width: 100%;
  display: flex;
  align-items: center;
  border-bottom: 1px solid var(--grayColor);
  padding-bottom: 25px;
}

.user_profile_img{
  width: 70px;
  height: 70px;
  border-radius: 25px;
  background: lightgray url(../../img/mimoticon_4.png) no-repeat center / cover;
}

.user_profile > h4{
  font-size: 22px;
  margin-left: 10px;
}

/*
  글자 사이(자간) 간격 조정 옵션
  영문 폰트의 경우 해당 옵션 자주 사용
*/
letter-spacing: -3px;
}

.friends_list_wrap{
  padding: 0 25px;
}

.friends_list_wrap > p{
  font-size: 14px;
  opacity: .7;
  margin-bottom: 20px;
}

.friends_profile{
  display: flex;
  align-items: center;
}

.friends_profile_img{
  width: 55px;
  height: 55px;
  background: lightgray url(../../img/mimoticon_3.png) no-repeat center / cover;
  border-radius: 18px;
  margin-right: 8px;
}

.friend_list{
  width: 100%;
  display: flex;
  justify-content: space-between;
  align-items: center;
}

.friends_num{
  opacity: .7;
  font-size: 15px;
}

.bottom_nav{
  width: 100%;
  padding: 20px 50px;
  background-color: #f9f9fa;
  position: fixed;
  bottom: 0;
  left: 0;
  border-top: 1px solid var(--grayColor);
}

.nav_list{
  width: 100%;
  justify-content: space-between;
  display: flex;
}

.nav_list i{
  font-size: 25px;
  color: #2e363e;
}

.nav_btn{
  /*
    position: relative;
    top:50px;
  */
}

```

```

/*
transform: translateY(50px);
opacity: 0;
animation: showBtns .3s ease-in-out forwards;
}

.nav_btn:nth-child(2){
    animation-delay: .2s;
}

.nav_btn:nth-child(3){
    animation-delay: .5s;
}

.nav_btn:nth-child(4){
    animation-delay: .8s;
}

@keyframes showBtns {
    from{}
    to{
        opacity: 1;
        transform: translateY(0);
    }
}

.nav_btn a{
    position: relative;
}

.nav_btn:nth-child(2) span{
    position: absolute;
    left: 15px;
    bottom: 10px;
}

.alert{
    width: 30px;
    height: 30px;
    background-color: tomato;
    border-radius: 50%;
    color: #fff;
    display: flex;
    justify-content: center;
    align-items: center;
    font-weight: 600;
    animation: alertRotate 2s ease-in-out infinite;
}

@keyframes alertRotate{
    50%{
        transform: translateY(-5px) rotateY(360deg);
    }
}

.splash{
    width: 100vw;
    height: 100vh;
    background-color: var(--mainColor);
    position: absolute;
    top: 0;
    left: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    font-size: 130px;
    /*
     *animation-fill-mode
     => 애니메이션이 끝난 후 처음(from)으로 돌아가느냐, 끝난 상태를 유지하느냐 결정하는 속성
      1) none : 별도의 옵션 지정 x
      2) forwards : 애니메이션 키 프레임이 100%(to)에 도달했을 때 마지막 키프레임 유지
      3) backwards : 애니메이션 스타일을 애니메이션이 시작되기 전의 상태로 적용
    */
    animation: hideSplash 0.3s ease-in-out forwards;
    animation-delay: 1s;
}

@keyframes hideSplash {
    from{
        opacity: 1;
    }

    to{
        opacity: 0;
    }
}

```

```

    /* 레이아웃에 영향을 주지 않고 요소를 없애주는 속성 */
    visibility: hidden;
}

}

```

2. css > screen > chatting

```

/* .chat_list_wrap{
    padding: 0 25px;
}

.chat_list_wrap > a{
    display: flex;
    align-items: center;
} */

/* .chat_list{
    width: 100%;
    display: flex;
    gap:20px;
    align-items: center;
}

.chat_img{
    width: 70px;
    height: 70px;
    border-radius: 25px;
    background: lightgray url(../../img/mimoticon_2.png) no-repeat center / cover;
}

.chat_message > h4{
    font-size: 18px;
    color: #000;
}

.chat_message > p{
    font-size: 14px;
    color: grey;
} */

/* .chat_alert{
    position: relative;
} */

/* .chat_alert > p{
    font-size: 14px;
    color: grey;
    opacity: .7;
} */

/* .alert_1{
    width: 30px;
    height: 30px;
    background-color: tomato;
    border-radius: 50%;
    color: #fff; /*
    /* position: absolute; */
    /* display: flex;
    justify-content: center;
    align-items: center;
} */

.chat_main{
    padding: 0 25px;
}

.chat_main > a{
    width: 100%;
    display: flex;
    justify-content: space-between;
    align-items: center;
}

.chat_profile{
    display: flex;
    align-items: center;
    gap: 10px;
}

```

```

.chat_profile > figure{
    width: 60px;
    height: 60px;
    background: lightgray url(../../img/mimoticon_5.png) no-repeat center / cover;
    border-radius: 20px;
}

.chat_txt{
    color: #333;
}

.chat_txt > h4{
    font-size: 22px;
}

.chat_txt > p{
    font-size: 14px;
    opacity: .7;
}

.chat_alert{
    display: flex;
    flex-direction: column;
    align-items: center;
}

.chat_alert > p{
    font-size: 14px;
    color: #333;
    opacity: .7;
}

.badge{
    width: 30px;
    height: 30px;
    background-color: tomato;
    border-radius: 50%;
    display: flex;
    justify-content: center;
    align-items: center;
    color: #fff;
    font-weight: 700;
}

```

2. **css > screen > catting_room**

```

/* .chat_room_wrap{
    width: 100%;
    height: 100%;
    background-color: #abc1d1;
}

.chat_top_wrap{
    padding: 0 20px;
    width: 100%;
    display: flex;
    justify-content: space-between;
    font-size: 22px;
} */

/* .chat_center_wrap{
    width: 100%;
    display: flex;
} */

/* .chat_center_wrap{
    display: flex;
    flex-direction: column;
}

.chat_center_wrap > figure{
    width: 50px;
    height: 50px;
    background: salmon url(../../img/mimoticon_4.png) no-repeat center / cover;
    border-radius: 20px;
}

.chat_bottom_wrap{

```

```

width: 100%;
height: 50px;
background-color: #fff;
position: fixed;
bottom: 0;
left: 0;
display: flex;
align-items: center;
justify-content: space-around
}

.chat_bottom_wrap > input[type="text"]{
width: 90%;
height: 90%;
border: 1px solid gray;
border-radius: 20px;
position: relative;
}

.fa-face-smile{
position: absolute;
right:80px;
}

.fa-arrow-up{
width: 40px;
height: 40px;
background-color: yellow;
border: 1px solid gray;
text-align: center;
line-height: 40px;
border-radius: 50%;
position: absolute;
right: 20px;
} */

.chat_room_wrap{
background-color: #abc1d1;
height: 300vh;
}

.chat_header{
width: 100%;
display: flex;
justify-content: space-between;
align-items: center;
padding: 25px;
font-size: 25px;
}

.chat_con{
width: 33%;
}

.chat_con:first-child i{
color: #000;
}

.chat_con:nth-child(2) h2{
text-align: center;
}

.chat_con:nth-child(3){
display: flex;
justify-content: flex-end;
}

.fa-bars{
display: inline-block;
margin-left: 10px;
}

.chatting{
padding: 100px 25px 0;
display: flex;
flex-direction: column;
align-items: center;
}

.date{
padding: 15px;
background-color: #92a4b2;
color: #fff;
border-radius: 25px;
}

```

```

        font-size: 14px;
    }

.message_1{
    /* width 값을 넣어줌으로써 컨텐츠가 왼쪽 기준으로 정렬하게 된다 */
    width: 100%;
    display: flex;
}

.m_profile{
    width: 50px;
    height: 50px;
    background: url(../../img/mimoticon_3.png) no-repeat center / cover;
}

.m_info{
    display: flex;
    align-items: flex-end;
}

.m_bubble{
    padding: 13px;
    background-color: #fff;
    border-radius: 15px;
    border-top-left-radius: 0;
    margin-right: 5px;
}

.m_time, .m_time_2{
    font-size: 14px;
}

.message_2{
    width: 100%;
    display: flex;
    /*
        width: 100%;를 적용해줘야 속성이 제대로 반영됨
        * width 값 반영 전까진 해당 컨텐츠의 크기만큼만 영역 차지
    */
    justify-content: flex-end;
}

.m_info_2{
    display: flex;
    flex-direction: row-reverse;
    align-items: flex-end;
}

.m_bubble_2{
    background-color: var(--mainColor);
    padding: 13px;
    border-radius: 15px;
    border-top-right-radius: 0;
    margin-left: 5px;
}

.reply{
    width: 100%;
    background-color: #fff;
    position: fixed;
    bottom: 0;
    left: 0;
    display: flex;
    padding: 5px 25px;
    justify-content: space-between;
    align-items: center;
}

.reply_con:first-child{
    width: 30px;
}

.reply_con:last-child{
    /* 플러스 아이콘 영역 30px를 제외한 모든 영역 차지 */
    flex: 1;
    position: relative;
}

.reply_con:last-child input{
    width: 100%;
    padding: 10px;
    border: 1px solid rgba(0,0,0,.2);
    border-radius: 25px;
}

.reply_wink{
    position: absolute;
}

```

```

        top: 15px;
        right: 50px;
    }

    .reply_con:last-child button{
        width: 35px;
        height: 35px;
        border-radius: 50%;
        background-color: var(--mainColor);
        position: absolute;
        top: 3px;
        right: 3px;
    }
}

```

2. CSS > screen > more

```

/* .wrap{
    height: 300vh;
}

.more_top_wrap{
    width: 100%;
    display: flex;
    justify-content: space-between;
    padding: 25px;
    background-color: gold;
} */

.more_main{
    padding: 0 25px;
}

.user_info{
    display: flex;
    align-items: center;
}

.user_info > figure{
    width: 80px;
    height: 80px;
    background: url(../../../../img/mimoticon_1.png) no-repeat center / cover;
}

.user_component{
    width: 100%;
    display: flex;
    justify-content: space-between;
    align-items: center;
}

.user_component_txt h4{
    font-size: 25px;
}

.user_component_txt p{
    opacity: .7;
}

.user_bubble > .fa-message{
    font-size: 25px;
}

.icon_list{
    padding: 0 25px;
    width: 100%;
    display: flex;
    justify-content: space-between;
    margin-top: 35px;
}

.icon{
    display: flex;
    flex-direction: column;
    align-items: center;
}

.icon i{
    font-size: 35px;
}

.icon span{

```

```

        display: inline-block;
        margin-top: 15px;
    }

    .suggestion{
        padding: 40px 25px;
        border-top: 1px solid var(--grayColor);
        margin-top: 50px;
    }

    .suggestion p{
        margin-bottom: 30px;
        opacity: .7;
    }

    .suggestion_icons{
        display: flex;
        gap: 25px;
    }

    .s_icon{
        display: flex;
        flex-direction: column;
        align-items: center;
    }

    .s_icon span{
        font-size: 14px;
        margin-top: 10px;
        opacity: .7;
    }

    .icon_box{
        width: 60px;
        height: 60px;
        background-color: var(--mainColor);
        border-radius: 10px;
        display: flex;
        justify-content: center;
        align-items: center;
    }

    .icon_box i{
        color: #fff;
        font-size: 25px;
    }

```

2. css > screen > mobile_only

```

.mobile_only{
    width: 100vw;
    height: 100vh;
    background-color: var(--mainColor);
    position: fixed;
    top: 0;
    left: 0;
    z-index: 999;
    display: flex;
    justify-content: center;
    align-items: center;
    font-size: 30px;
    font-weight: 700;
}

@media all and (max-width:767px){
    .mobile_only{
        display: none;
    }
}

```

- **svg**

=> 확장 가능한 벡터 그래픽

=> 웹페이지에서 벡터 표준 = svg

=> ui / 아이콘, 그래프

- => css, js 조작 가능
- => 장점) 확대해도 절대 깨지지 않고 늘어나지 않음
- => 단점) 모양이 복잡하고 개체수가 많을수록 성능 저하
- => 벡터 → 일러스트 / 비트맵 → 포토샵

※ svg 작성 예시

```

<!DOCTYPE html>
<html lang="ko">
<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>svg</title>
    <style>
        *{
            margin: 0;
            padding: 0;
            box-sizing: border-box;
        }

        .svg{
            width: 300px;
            height: 300px;
            background: url(img/bear.svg) no-repeat center / cover;
        }

        .svg_canvas{
            background-color: lightblue;
        }

        /* 포토샵 또는 일러스트에서의 테두리 속성 적용 */
        ellipse{
            stroke: red;
            /* 테두리 */
            stroke-width: 10;
        }

        path{
            stroke: #000;
            stroke-width: 5;

            /* 선 내부 채우기를 투명하게 하는 속성 */
            fill: transparent;
        }
    </style>
</head>
<body>

    <!-- #1. img tag -->
    

    <!-- #2. background-image -->
    <div class="svg"></div>

    <!-- svg 작성할 수 있는 영역(도화지) -->
    <svg width="600" height="500" class="svg_canvas">

        <!--
            색상 지정 없을 경우 기본 검정색
            fill : 색상 변경
            x: x축 시작점
            y: y축 시작점
        -->
        <rect x="10" y="30" width="200" height="100" fill="salmon"></rect>
        <!-- 일반 사각형 -->

        <circle cx="450" cy="150" r="30" fill="rgba(0,0,0,.5)"></circle>
        <!--
            원형
            cx = center x (원의 정중앙을 기준으로 x축)
            cy = center y (원의 정중앙을 기준으로 y축)
            r = radius = 반지름 / 원의 크기 결정
        -->

        <rect x="50" y="170" rx="10" width="100" height="100" fill="orange"></rect>
        <!-- 둘지점의 반지름이 둘근 사각형 -->

        <ellipse cx="200" cy="200" rx="100" ry="50" fill="green"></ellipse>
    </svg>
</body>

```

```

<!-- 타원 -->
<line x1="10" x2="400" y1="30" y2="300" stroke="blue" stroke-width="10"></line>
<!--
    직선
    일직선을 그리려고 하는 경우 x값만 필요하지만 최상단에 선이 정렬되기 때문에
    y축으로 일직선을 그리려면 y1, y2를 동일한 값을 적용해줄 것
-->

<path d="M 300 200 L 500 100 H 50 V 300 Z"></path>
<!--
    자유그리기 = path
    M : path 시작 위치
    L : 직선
    H : 수평선 (X 위치만 필요)
    V : 수직선 (Y 위치만 필요)
    Z : 마무리
-->

<path d="M 50 400 C 50 400, 200 200, 500 250"></path>
<!--
    커브선
    1. M : 커브선 시작점으로 이동
    2. C : 선의 시작점 지정, 커브가 그려질 위치(고점 선택), 끝나는 점 지정
-->

</svg>

<svg id="_레이어_1" data-name="레이어 1" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 383.16 78.05">
<defs>
    <style>.cls-1 {fill: #fff;stroke: #000;stroke-miterlimit: 10;}</style>
</defs>
    <path class="cls-1" d="m.29,46.92 s103.03-73.08,210.86,5.49S382.71.22,382.71.22"/>
</svg>

<!--
    jpg, png, gif / svg
    svg

=> 확장 가능한 벡터 그래픽
=> 웹페이지에서 벡터 표준 = svg
=> ui / 아이콘, 그래프
=> css, js 조작 가능

=> 장점) 확대해도 절대 깨지지 않고 늘어나지 않음
=> 단점) 모양이 복잡하고 개체수가 많을수록 성능 저하

=> 벡터 → 일러스트 / 비트맵 → 포토샵
-->

</body>
</html>

※ svg animation 효과 적용
<!DOCTYPE html>
<html lang="ko">
<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>curve text</title>
    <style>
        *{
            margin: 0;
            padding: 0;
            box-sizing: border-box;
        }

        .svg{
            width: 100%;
            height: 100vh;
            position: absolute;
            background-color: lightgray;
        }

        .curve-line{
            fill: transparent;
            stroke: #1d1d1d;
            stroke-width: 3;
            stroke-dasharray: 700;
            /*
            선을 점선 형태로 만들
            값을 부여한 만큼 dash 길이가 늘어남
            값이 작으면 선을 수록 점선에 가까운 모양으로 바뀜
            */
        }
    </style>
</head>
<body>
    <div>
        <img alt="A curved text effect where the word 'curve' is written in a wavy, flowing font." data-name="curve text" style="width: 100%; height: 100%;"/>
    </div>
</body>
</html>

```

```

        stroke-dashoffset: 300;
    /*
        svg의 시작점 설정
        offset 설정된 숫자의 값이 작으면 왼벽한 svg 그래픽이 보이고
        숫자가 값이 크면 svg 그래픽이 잘려서 보임
    */
}

circle{
    stroke: #000;
    fill: transparent;
    stroke-width: 10;
    stroke-dasharray: 1260;
    stroke-dashoffset: 0;
    animation: dash 3s linear infinite alternate;
}

}

@keyframes dash{
    from{
        stroke-dashoffset: 1260;
    }
    to{
        stroke-dashoffset: 0;
    }
}

.special{
    font-size: 20px;
    font-weight: bold;
    fill: salmon;
}

#text-curve{
    stroke-dasharray: 700;
    animation: dash-animation 2s linear alternate;
}

@keyframes dash-animation {
    from{
        stroke-dashoffset: 700;
    }
    to{
        stroke-dashoffset: 0;
    }
}

</style>
</head>
<body>
<!-- <svg class="svg" id="_레이어_1" data-name="레이어 1" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 383.16 78.05"> -->
<!-- 나중에 참조할 그래픽 요소를 담아 놓는 공간(태그) -->
<defs>
    <!-- <style>.cls-1 {fill: #fff;stroke: #000;stroke-miterlimit: 10;}</style> -->
    <!-- <path id="text-curve" class="curve-line" d="m.29,46.92 s103.03-73.08,210.86,5.49S382.71.22,382.71.22"/> -->
</defs>

<!-- <text x="20">
    <textPath href="#text-curve">
        Lorem ipsum <span class="special">dolor</span> sit amet consectetur adipisicing elit. Explicabo, ad? Quidem molestia
    </textPath>
</text>
</svg> -->

<svg class="svg">
    <circle cx="400" cy="250" r="100"> </circle>
</svg>

</body>
</html>

```

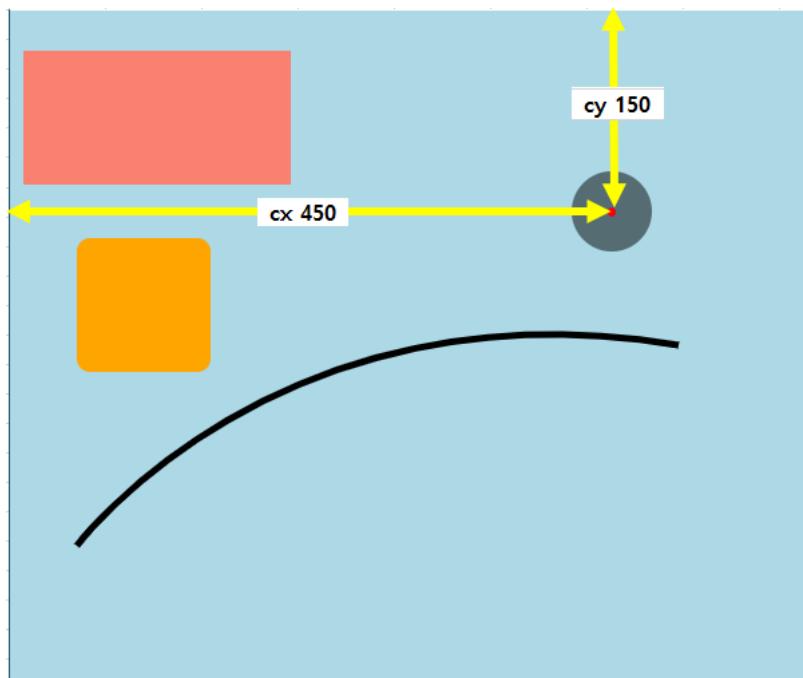
※ 원형 예시

```

<circle cx="450" cy="150" r="30" fill="rgba(0,0,0,.5)"></circle>
<!--
    원형
    cx = center x (원의 정중앙을 기준으로 x축)
    cy = center y (원의 정중앙을 기준으로 y축)

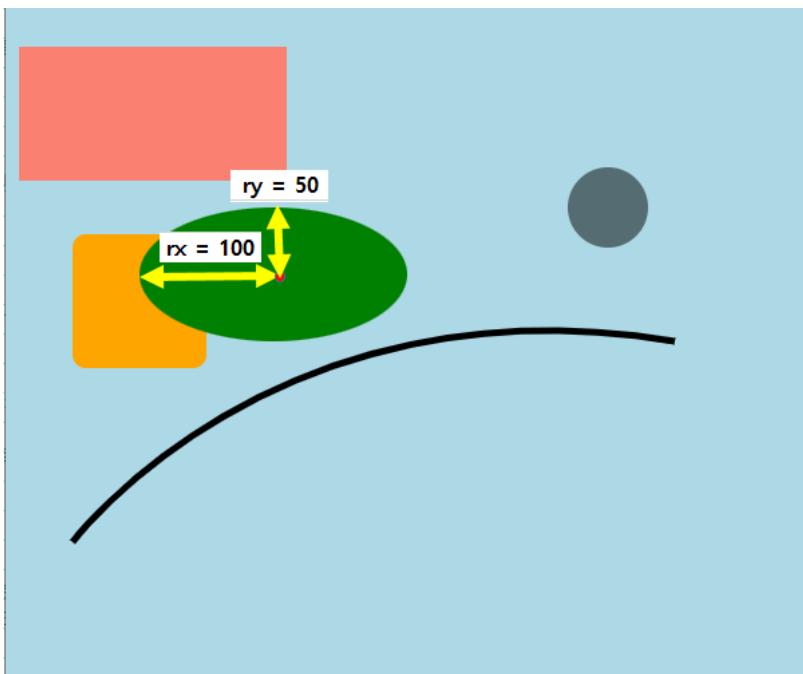
```

r = radius = 반지름 / 원의 크기 결정
-->



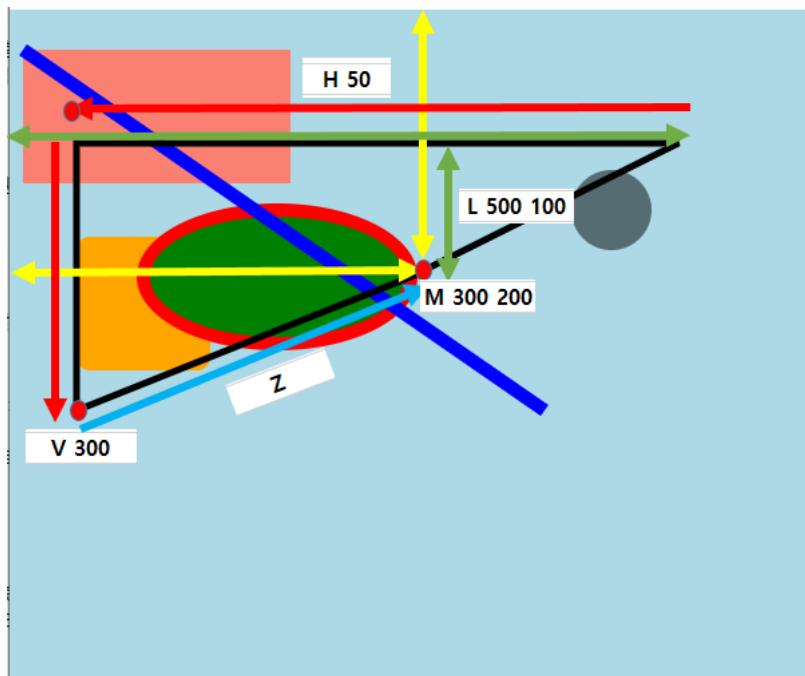
※ 타원형 예시

```
<ellipse cx="200" cy="200" rx="100" ry="50" fill="green"></ellipse>
```



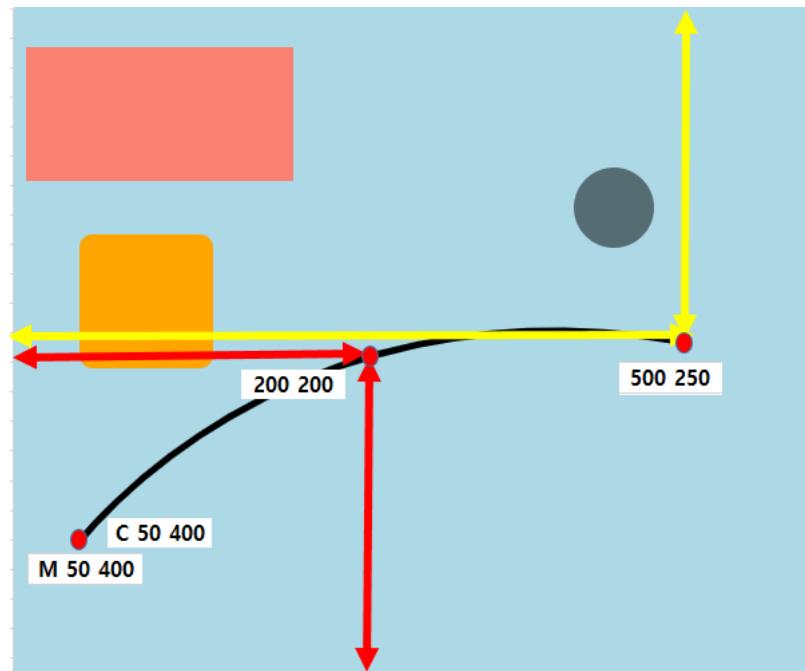
※ 자유그리기(직선) 예시

```
<path d="M 300 200 L 500 100 H 50 V 300 Z"></path>
```



※ 자유그리기(곡선) 예시

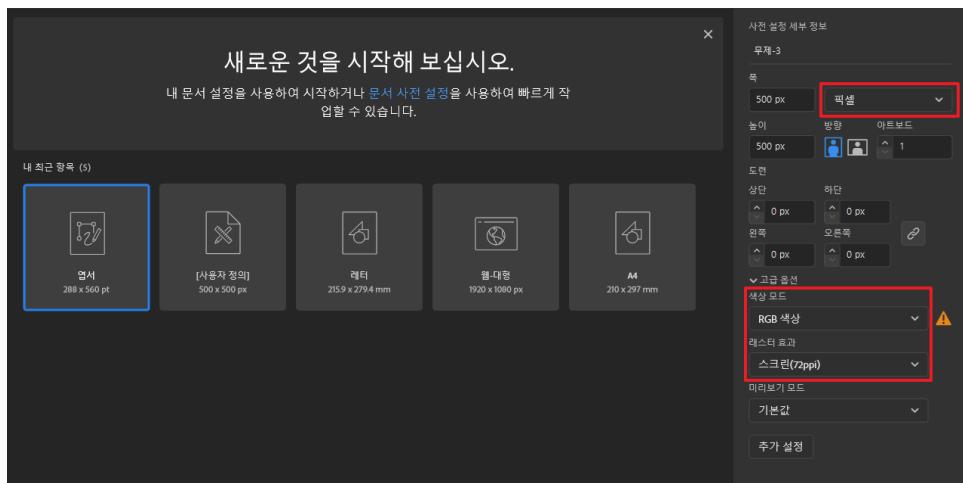
```
<path d="M 50 400 C 50 400, 200 200, 500 250"></path>
```



- 일러스트 사용방법

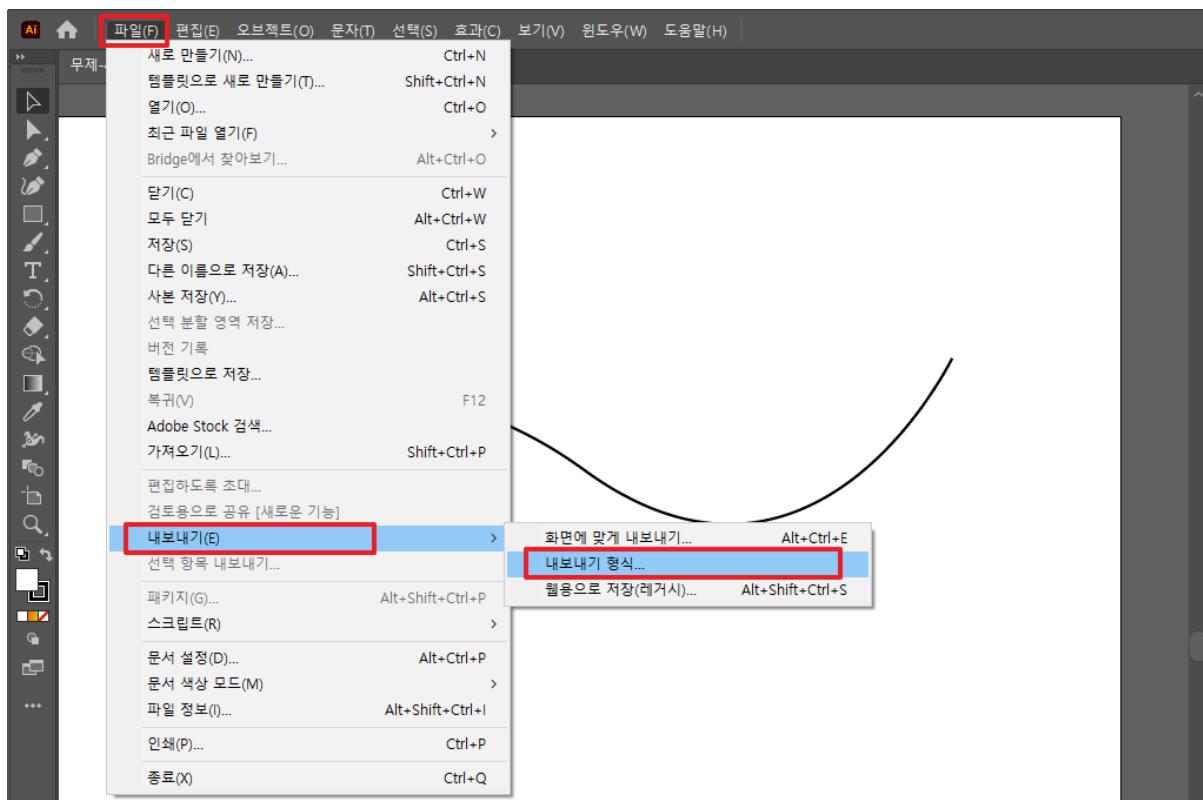
1) 기본설정

- Ctrl + n (새로운 문서 만들기) → 폭 단위 변경(밀리미터 → 픽셀) → 색상모드(RGB), 레스터 효과(72ppi)



2) SVG 파일 만들기

- 상단 메뉴 - 파일 - 내보내기 - 내보내기 형식 *.SVG로 파일 명 지정 후 저장



<https://mytory.net/archives/11175> : 일러스트레이터를 이용한 SVG 파일 만들기

※ **stroke-dasharray**, **stroke-dashoffset**

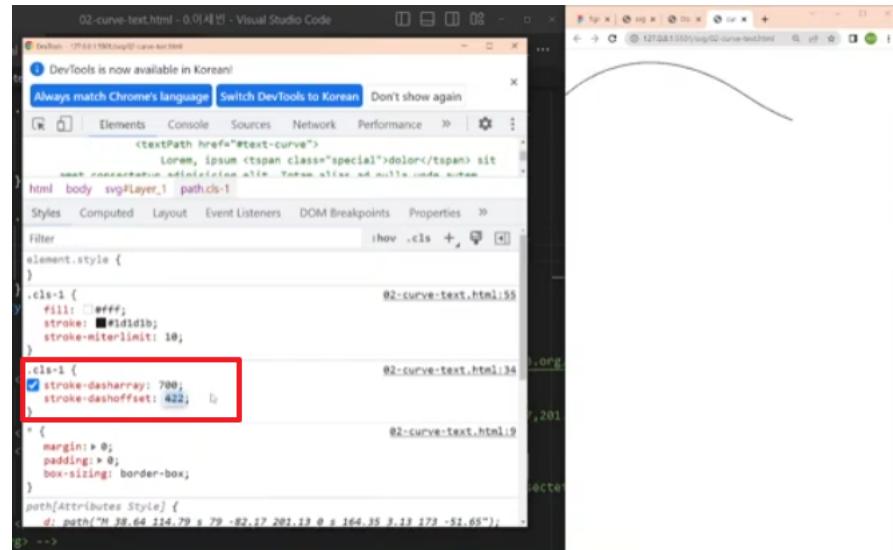
```
.curve-line{  
    fill: transparent;  
    stroke: #1d1d1d;  
    stroke-width: 3;  
    stroke-dasharray: 700;  
    /*  
        선을 점선 형태로 만들
```

```

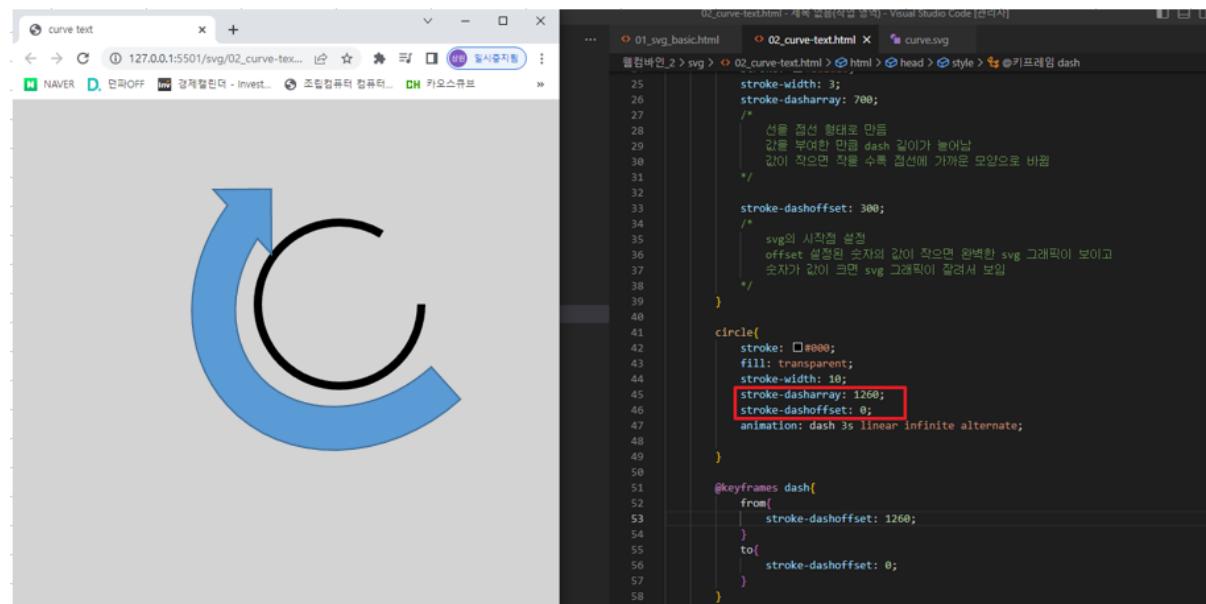
값을 부여한 만큼 dash 길이가 늘어남
값이 작으면 작을 수록 절선에 가까운 모양으로 바뀜
*/
stroke-dashoffset: 300;
/*
svg의 시작점 설정
offset 설정된 숫자의 값이 작으면 완벽한 svg 그래픽이 보이고
숫자가 값이 크면 svg 그래픽이 절려서 보임
*/
}

※ stroke-dashoffset : 숫자가 커질수록 선은 잘 보이지 않는다
stroke-dasharray : 숫자가 커질수록 완벽한 선에 가까움

```



※ stroke-dashoffset animation 효과



- viewBox

min-x min-y width height

=> 밀도, 배율
=> svg 그래픽을 viewBox 속성 없이도 그릴 수 있지만 반응형에 취약함
=> viewBox 속성을 이용하여 화면 크기에 맞게 svg요소 크기 조정 가능
=> viewBox의 width, height에 따라 안에 있는 도형들의 크기가 결정됨
=> min-x, min-y : svg가 그려지는 영역의 시작점
=> width, height : 각 영역의 가로 세로 길이

- **transform-origin**

=> 회전의 중심점(기준점) 지정
=> 기본값 : 50% 50%;
=> 사용단위 : %, 키워드, px

[백분율과 대응 키워드]

0% => left, top
50% => center
100% => right, bottom

※ 곰돌이 움직이는 애니메이션

```
<!DOCTYPE html>
<html lang="ko">
<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>stroke animation</title>
    <style>
        *{
            margin: 0;
            padding: 0;
            box-sizing: border-box;
        }

        .svg{
            position: absolute;
            width: 100%;
            height: 100vh;
            background-color: #16345a;
        }

        .ears{
            animation: moveEars 1s linear infinite alternate;
        }

        /* 타원의 경우 cx="315" cy="90" 값을 기준점으로 지정해줘야 제자리에서 움직이는 애니메이션 적용 가능 */
        .ears:first-child{
            transform-origin: 315px 90px;
            /*
                transform-origin
                => 회전의 중심점(기준점) 지정
                => 기본값 : 50% 50%;
                => 사용단위 : %, 키워드, px
            */
            [백분율과 대응 키워드]

            0% => left, top
            50% => center
            100% => right, bottom
        }

        .ears:nth-child(2){
            transform-origin: 105px 90px;
        }

        .cheek{
            animation: cheek 1s linear infinite alternate;
        }
    </style>

```

```

@keyframes moveEars {
    to{
        transform: scale(1.1);
    }
}

@keyframes cheek{
    from{
        transform: translateY(0)
    }
    to{
        transform: translateY(-10px)
    }
}

</style>
</head>
<body>
<svg class="svg" width="420" height="420" viewBox="0 0 420 420" fill="none" xmlns="http://www.w3.org/2000/svg">
<!-- <rect width="420" height="420" fill="white"/> -->
<circle class="ears" cx="315" cy="90" r="40" fill="#765049" stroke="black" stroke-width="10"/>
<circle class="ears" cx="105" cy="90" r="40" fill="#765049" stroke="black" stroke-width="10"/>
<circle cx="210" cy="210" r="150" fill="#B68278"/>
<circle cx="210" cy="210" r="150" fill="#B68278"/>
<circle cx="210" cy="210" r="150" fill="#B68278"/>
<circle cx="210" cy="210" r="150" stroke="black" stroke-width="10"/>
<mask id="path-4-inside-1_1_7" fill="white">
<path fill-rule="evenodd" clip-rule="evenodd" d="M186.693 238H233.522C231.866 228.918 231 219.56 231 210C231 157.078 257.522 1</mask>
<path fill-rule="evenodd" clip-rule="evenodd" d="M186.693 238H233.522C231.866 228.918 231 219.56 231 210C231 157.078 257.522 1</path>
<path d="M186.693 238L176.906 235.946L174.376 248H186.693V238ZM233.522 238V248H245.512L243.36 236.205L233.522 238ZM298 82.3857<circle cx="210.5" cy="253.5" r="22.5" fill="#B68278" stroke="black" stroke-width="10"/>
<path d="M171 232.5C171 215.655 184.655 202 201.5 202H209V244C209 254.493 200.493 263 190 263C179.507 263 171 254.493 171 244V<path d="M250 232.5C250 215.655 236.345 202 219.5 202H212V244C212 254.493 220.507 263 231 263C241.493 263 250 254.493 250 244V<path d="M191.5 200.5H228.5L217 223H204L191.5 200.52" fill="#0B0B0B"/>
<circle class="cheek" cx="290.5" cy="235.5" r="35.5" fill="url(#paint0_linear_1_7)"/>
<circle class="cheek" cx="130.5" cy="238.5" r="35.5" fill="url(#paint1_linear_1_7)"/>
<rect x="243" y="177" width="24" height="10" rx="5" fill="black"/>
<rect x="154" y="177" width="24" height="10" rx="5" fill="black"/>
</defs>
<linearGradient id="paint0_linear_1_7" x1="290.5" y1="200" x2="290.5" y2="271" gradientUnits="userSpaceOnUse">
    <stop stop-color="#FF0000" stop-opacity="0.37"/>
    <stop offset="1" stop-color="#D9D9D9" stop-opacity="0"/>
</linearGradient>
<linearGradient id="paint1_linear_1_7" x1="130.5" y1="203" x2="130.5" y2="274" gradientUnits="userSpaceOnUse">
    <stop stop-color="#FF0000" stop-opacity="0.37"/>
    <stop offset="1" stop-color="#D9D9D9" stop-opacity="0"/>
</linearGradient>
</defs>
</svg>

<!--
    viewBox
    min-x min-y width height
    => 밀도, 배율
    => svg 그래픽을 viewBox 속성 없이도 그릴 수 있지만 반응형에 취약함
    => viewBox 속성을 이용하여 화면 크기에 맞게 svg요소 크기 조정 가능
    => viewBox의 width, height에 따라 안에 있는 도형들의 크기가 결정됨
    => min-x, min-y : svg가 그려지는 영역의 시작점
    => width, height : 각 영역의 가로 세로 길이

    viewPort
-->
</body>
</html>

```

※ 일러스트 펜툴로 그린 글자 애니메이션

```

<!DOCTYPE html>
<html lang="ko">
<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>stroke animation</title>
    <style>
        *{
            margin: 0;
            padding: 0;
            box-sizing: border-box;
        }
    </style>

```

```

.svg{
    position: absolute;
    width: 100%;
    height: 100vh;
}

.cls-1 {
    fill: none;
    stroke: #2900b9;
    stroke-miterlimit: 10;
    stroke-dasharray: 30px;
    stroke-dashoffset: 0;
    animation: nameDraw 5s infinite alternate;
}

@keyframes nameDraw {
    from{
        stroke-dashoffset: 30px;
    }
    to{
        stroke-dashoffset: 0;
    }
}

</style>
</head>
<body>
<svg id="_레이어_1" data-name="레이어 1" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 39.52 11.78">
<defs></defs>
<path class="cls-1" d="m3.25,2.78S2.35-.48.89.8s1.72,5.9,1.72,5.9c0,0,2.04,4.05-2.11,2.55"/>
<path class="cls-1" d="m8.23,10V3.83s-.1-3.96-3.29-.41"/>
<path class="cls-1" d="m7.75,4.5s-2.81.76-2.68,2.81,1.72,2.81,2.68,1.41.29-.32.29-.32"/>
<polyline class="cls-1" points="11.07 1.99 11.07 3.8 11.07 10"/>
<path class="cls-1" d="m10.91,3.54C3.38-2.49,3.13-1.02,3.13-1.02v7.48"/>
<path class="cls-1" d="m18.89.73s2.39-.32.61,1.5c0,0-3.03-.29-3,2.62s2.3,2.46,2.3,2.46c0,0,1.18-.09,1.12-2.58s-.93-2.14-.93-2.</p>
<path class="cls-1" d="m17.13,7.33s-2.42,1.82,2.92,2.1c0,0,1.55.53-.21,1.49s-3.35-.26-3.35-.26c0,0-.45-.55.67-1.23"/>
<line class="cls-1" x1="22.59" y1="1.94" x2="23.8" y2="10"/>
<line class="cls-1" x1="25.4" y1="1.94" x2="23.74" y2="9.59"/>
<line class="cls-1" x1="26.58" y1="10" x2="25.3" y2="2.38"/>
<line class="cls-1" x1="28.27" y1="1.99" x2="26.51" y2="9.59"/>
<path class="cls-1" d="m31.84,2.38s-1.72-.46-1.63,3.38c0,0-.35,3.77,1.63,3.82s1.91-4.54,1.91-4.54c0,0-.06-2.86-1.91-2.66Z"/>
<polyline class="cls-1" points="36.22 2.09 36.22 3.45 36.22 10"/>
<path class="cls-1" d="m36.34,4.02c2.49-2.94,2.68-1.64,2.68-1.64v7.61"/>
</svg>
<!-- <svg class="svg" id="Layer_1" data-name="Layer 1" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 42.25 10.75">
<defs></defs>
<path id="sebhin" class="cls-1" d="M941.57,539.25s-1.79,-1-.94,2.58c0,0,1.37,1.5.94,2.45,0,0-3.32,1.82-2.15,1.17s3.91-1.79,5.63
<circle id="dot" class="cls-2" cx="30.29" cy="1.2" r="0.38"/>
</svg> -->
<!-- 일러스트에서 레이어명 변경 시,
id명으로 변환됨
-->

<script>
    console.log(document.querySelector('.cls-1').getTotalLength())
</script>
</body>
</html>

```

※ 텍스트 stroke 애니메이션

```

<!DOCTYPE html>
<html lang="ko">
<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>text-animation</title>
    <style>
        *{
            margin: 0;
            padding: 0;
            box-sizing: border-box;
        }

.svg{
    width: 100%;
    height: 100vh;
    position: absolute;
    background-color: #16345a;
}

```

```

        }

    text{
        /* font-size: 30px; */
        /* fill: gold; */
        font-size: 6rem;
        font-weight: bold;
        fill: transparent;
        stroke: gold;
        stroke-width: 3px;
        stroke-dasharray: 390px;
        animation: strokeAni 3s linear infinite;
    }

@keyframes strokeAni {
    from{
        stroke-dashoffset: 500px;
    }
    to{
        stroke-dashoffset: 0;
    }
}

</style>
</head>
<body>
    <svg class="svg" viewBox="0 0 500 500">
        <text>Hello SVG</text>
    </svg>
</body>
</html>

```

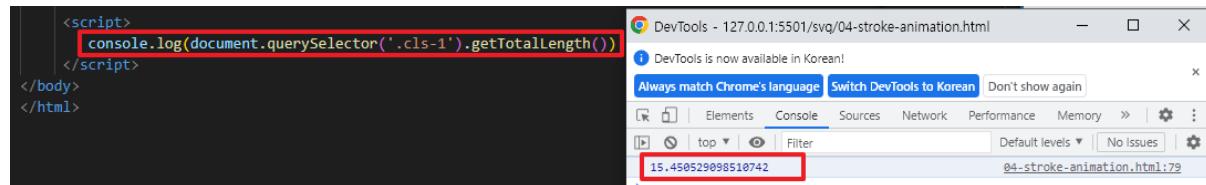
※ stroke 길이 계산 js

- 웹에서 f12(검사창) → 콘솔탭

```

<script>
    console.log(document.querySelector('.cls-1').getTotalLength())
</script>

```



- pattern

```

<!DOCTYPE html>
<html lang="ko">
<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>pattern</title>
    <style>
        *{
            margin: 0;
            padding: 0;
            box-sizing: border-box;
        }

        svg{
            width: 500px;
            height: 500px;
            background-color: #ddd;
        }

        .pattern-rect{
            fill: url(#pattern);
        }
    </style>

```

```

</style>
</head>
<body>
<svg>
<defs>
    <!-- 참조 -->
    <!-- viewBox="0 0 100 100" id="pattern" x="0" y="0" width="0.1" height="0.1">
        circle 100 x 100 사이즈로 10개를 viewBox에 채우겠다는 뜻
    -->
    <pattern viewBox="0 0 100 100" id="pattern" x="0" y="0" width="0.1" height="0.1">
        <circle cx="50" cy="50" r="50" fill="hotpink"/>
    <!-- defs 내부에 참조형태로 존재하기 때문에 시각적으로 표현되지 않음 -->
    </pattern>
</defs>

<rect class="pattern-rect" x="0" y="0" width="100%" height="100%"/></rect>

</svg>
<!--
    pattern
    width="0.1"
    height="0.1"

    1 = 100%
    0.1 = 1/10

    1/10의 크기로 10개씩 넣겠다
-->
[패턴 출력 방법]

1. cx, cy, r 조절
=> 초기값 : 50
=> 변경값 : 25

2. pattern width, height 조절

3. pattern viewBox 설정
=> 출력하고 싶은 패턴의 갯수를 뷔박스 상대적인 크기로 동일하게 설정
-->
</body>
</html>

※ 이미지 반복 패턴 적용

<!DOCTYPE html>
<html lang="kr">
<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>patternBg</title>
    <style>
        *{
            margin: 0;
            padding: 0;
            box-sizing: border-box;
        }

        body{
            background: url(img/bear.svg);
        }
    </style>
</head>
<body>

</body>
</html>

```

```

01-svg_basic.html 03-stroke-animation.html 05-text-animation.html 06-pattern.html x 04-stroke-animation.html
01-설명_2.svg 06-pattern.html > html > head > style > pattern-rect
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47

```

The screenshot shows a browser window displaying a grid of pink circles. The developer tools are open, and a yellow arrow points from the code editor to the pattern definition in the browser's developer tools.

※ mask

```

<!DOCTYPE html>
<html lang="kr">
<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>mask</title>
    <style>
        *{
            margin: 0;
            padding: 0;
            box-sizing: border-box;
        }

        svg{
            width: 100%;
            height: 100%;
            position: absolute;
            top: 0;
            left: 0;
            background-color: #ddd;
        }

        text{
            fill: salmon;
            font-size: 50px;
            font-weight: bold;
        }
    </style>
</head>
<body>

    <svg>
        <defs>
            <mask id="circle-mask">
                <!-- <circle cx="350" cy="100" r="100" fill="#fff"></circle> -->
                <image xlink:href="./img/DHFO.jpg"></image>
            </mask>
        </defs>

        <g mask="url(#circle-mask)">
            <text x="10" y="100">
                Lorem ipsum, dolor sit amet consectetur adipisicing elit. Aliquid quo quasi ullam, explicabo libero officia illo totam
            </text>
        </g>
    </svg>

</body>
</html>

```

※ mask stroke

```

<!DOCTYPE html>
<html lang="ko">
<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>mask stroke</title>
<style>
    .cls-1{
        fill:#fff;
    }

    .cls-2{
        fill:none;
        stroke:gold;
        stroke-miterlimit:10;
        stroke-width:3px;
        animation: hello 5s linear;
        stroke-dasharray: 157;
    }

    @keyframes hello {
        from{
            stroke-dashoffset: 157;
        }
        to{
            stroke-dashoffset: 0;
        }
    }
</style>
</head>
<body>
    <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 40.37 28.5">
        <defs>
            <mask id="text-mask">
                <g id="Layer_1" data-name="Layer 1">
                    <path class="cls-1" d="M43.69,23.69v2.46h3.76V23.69a1.13,1.13,0,0,1,.24-.79.78.78,0,0,1,.62-.27.85.85,0,0,1,.64.26
                    <path class="cls-1" d="M56.72,24.12H52.8v2.11h3.61a.83.83,0,0,1,.59.18.59.59,0,0,1,-.24.47.62.62,0,0,1-.19.48.86.86,
                    <path class="cls-1" d="M60.77,23.69v6.25H64.3a.9.9,0,0,1,.64.21.65.65,0,0,1,.23.51.64.64,0,0,1-.22.51,1,1,0,0,1-.6
                    <path class="cls-1" d="M68,23.69v6.25h3.52a.91.91,0,0,1,.65.21.68.68,0,0,1,0,1,.94.94,0,0,1-.65.2h-4.2a1.08,1.08,0
                    <path class="cls-1" d="M76.74,22.63a4.53,4.53,0,0,1,2.3.55,3.54,3.54,0,0,1,1.45,1.54A5.31,5.31,0,0,1,81,27.06a5.69
                </g>
            </mask>
        </defs>

        <g id="stroke" mask="url(#text-mask)">
            <path class="cls-2" d="M42.68,22.14V32.27l1.4-5.63h5.28v-4.5l-.29,10.13,9.61-9-5.8.14-.2,3.79,5.73-.2-5.82.67.09,3.13L58,30
        </g>
    </svg>
<!-->
    텍스트 stroke animation + mask
    [마스크를 활용한 텍스트 애니메이션]

    1. 일러스트에서 텍스트 쓰고 윤곽선 깨기(Ctrl + shift + 0)
    2. 글씨 고정(Ctrl + 2)
    3. 펜툴로 글씨 따기
    **글자 크기보다 Stroke가 더 두껍게 들어가야함
    4. 레이어 이름 변경(필요한 경우)
    5. svg 내보내기
    6. 윤곽선 꺼낸 글자를 defs 안쪽 mask에 넣어주기
    7. mask에 id 부여
    8. defs 밖에 있는 stroke에 mask="url(#id명)" 연결
    9. 애니메이션 효과 적용
    10. 윤곽선 꺼낸 글자에 fill : #fff; 흰색 처리
    ==> 마스크는 흰색이면 잘 보이고 검은색에 가까울수록 잘 안보임
-->
<script>
    // let 변수명
    let pathEl = document.querySelector('.cls-2')
    console.log(pathEl.getTotalLength());
</script>
</body>
</html>

```

<https://cdnjs.com/> : js, jquery 참조

• 이벤트(hover) 예제

※ 1) 메뉴 hover 할 때 높이값 + 그림자 효과

```

<!DOCTYPE html>
<html lang="ko">
<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>hover menu</title>
    <style>
        *{
            margin: 0;
            padding: 0;
            box-sizing: border-box;
        }

        ul{
            list-style: none;
        }

        a{
            text-decoration: none;
            color: initial;
        }

        /* reset */

        .menu_wrap{
            width: 1200px;
            margin: 50px auto;
            display: flex;
        }

        .menu_wrap > li{
            width: 200px;
            height: 200px;
            font-size: 20px;
            transition: all 0.3s;
            position: relative;
        }

        .menu_wrap > li > a{
            display: flex;
            flex-direction: column;
            justify-content: center;
            align-items: center;
        }

        .menu_wrap > li:first-child{
            background-color: rgb(208,101,3);
        }

        .menu_wrap > li:nth-child(2){
            background-color: rgb(233,147,26);
        }

        .menu_wrap > li:nth-child(3){
            background-color: rgb(22,145,190);
        }

        .menu_wrap > li:nth-child(4){
            background-color: rgb(22,107,162);
        }

        .menu_wrap > li:nth-child(5){
            background-color: rgb(27,54,71);
        }

        .menu_wrap > li:last-child{
            background-color: rgb(21,40,54);
        }

        .icon{
            width: 80px;
            height: 80px;
            display: flex;
            justify-content: center;
            align-items: center;
            margin: 30px auto;
            background-color: rgba(255,255,255,.1);
            /* x축, y축으로 일만큼 이동해서 그림자를 만들 것이냐 하는 속성 */
            box-shadow: 0 0 80px rgba(255,255,255,0);
            /* default : 그림자 블러 80px, 흰색, 투명도 0 상태 */
            border-radius: 50%;
            transition: all .5s;
        }
    </style>

```

```

        }

        .icon > img{
            width: 40px;
            height: 40px;
        }

        .menu_wrap .text{
            color: #fff;
            width: 100%;
            text-align: center;
            position: absolute;
            bottom: 30px;
        }

        /* hover */

        .menu_wrap:hover .icon{
            opacity: .4;
        }

        .menu_wrap > li:hover{
            height: 250px;
            border-bottom: 5px solid rgba(0,0,0,.2);
        }

        /* .menu_wrap > li:hover .text{
            transform: translateY(30px);
        } */

        .menu_wrap > li:hover .icon{
            opacity: 1;
            box-shadow: 0 0 0 rgba(255,255,255,.5);
            /* li:hover 그림자 블러가 사라지고 opacity 50% */
        }

    
```

</style>

</head>

<body>

```

<div class="wrap">
    <ul class="menu_wrap">
        <li>
            <a href="#">
                <figure class="icon">
                    
                </figure>
                <p class="text">New</p>
            </a>
        </li>

        <li>
            <a href="#">
                <figure class="icon">
                    
                </figure>
                <p class="text">Camera</p>
            </a>
        </li>

        <li>
            <a href="#">
                <figure class="icon">
                    
                </figure>
                <p class="text">Download</p>
            </a>
        </li>

        <li>
            <a href="#">
                <figure class="icon">
                    
                </figure>
                <p class="text">Heart</p>
            </a>
        </li>

        <li>
            <a href="#">
                <figure class="icon">
                    
                </figure>
                <p class="text">Setting</p>
            </a>
        </li>
    </ul>
</div>

```

```

        </li>

    <li>
        <a href="#">
            <figure class="icon">
                
            </figure>
            <p class="text">Share</p>
        </a>
    </li>
</ul>
</div>

</body>
</html>

```

※ 2) 텍스트 블로 및 그림자 효과 + 한글자씩 팝업

=> 텍스트의 경우 한글자씩 태그로 선택해서 효과 적용

```

<!DOCTYPE html>
<html lang="ko">
<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>hover box-shadow</title>
    <style>
        *{
            margin: 0;
            padding: 0;
            box-sizing: border-box;
        }

        /* a{
            text-decoration: none;
            color: initial;
        }

        .wrap{
            width: 100%;
            height: 500px;
        }

        .wrap p{
            font-size: 50px;
            margin-top: 50px;
            margin-bottom: 50px;
            text-align: center;
        }

        .center_wrap{
            width: 100%;
            display: flex;
            justify-content: center;
            align-items: center;
        }

        .center_wrap > a{
            transition: all 5s;
        }

        .center_wrap:hover p{
            padding: 20px;
            box-shadow: 8px 8px rgba(0,0,0,.5);
        }

        .bottom_wrap{
            width: 100%;
            display: flex;
            justify-content: center;
            align-items: center;
        }

        .bottom_wrap:hover p{
        } */

        .btn_1, .btn_2{
            width: 200px;
            height: 80px;
        }
    </style>

```

```

        margin: 50px auto;
        display: flex;
        justify-content: center;
        align-items: center;
        font-size: 30px;
        font-weight: bold;
        transition: all .5s;
        cursor: pointer;
    }

    .btn_1:hover{
        box-shadow: 5px 5px 5px #000;
    }

    .btn_2 span{
        transition: all .5s;
    }

    .btn_2:hover span{
        transform: translateY(-20px);
    }

    .btn_2 span:nth-child(2){
        transition-delay: 0.1s;
    }

    .btn_2 span:nth-child(3){
        transition-delay: 0.2s;
    }

    .btn_2 span:nth-child(4){
        transition-delay: 0.3s;
    }

    .btn_2 span:nth-child(5){
        transition-delay: 0.4s;
    }

    .btn_2 span:nth-child(6){
        transition-delay: 0.5s;
    }

    .btn_2 span:nth-child(7){
        transition-delay: 0.6s;
    }


```

</style>

</head>

<body>

```

<!-- <div class="wrap">
    <div class="top_wrap">
        <a href="#">
            <p>Hover me</p>
        </a>
    </div>
    <div class="center_wrap">
        <a href="#">
            <p>Hover me</p>
        </a>
    </div>
    <div class="bottom_wrap">
        <a href="#">
            <p>Hover me</p>
        </a>
    </div>
</div> -->

<div class="btn_1">Hover Me</div>
<div class="btn_2">
    <span>H</span>
    <span>o</span>
    <span>v</span>
    <span>e</span>
    <span>r</span>
    <span>M</span>
    <span>e</span>
</div>


```

</body>

</html>

※ 3) 언더라인 우측 → 좌측 효과

=> 가상선택자 또는 span 태그 활용

```
<!DOCTYPE html>
<html lang="ko">
<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>hover_menu</title>
<style>
  *{
    margin: 0;
    padding: 0;
    box-sizing: border-box;
  }

  ul{
    list-style: none;
  }

  a{
    text-decoration: none;
    color: initial;
  }

  .menu_wrap{
    width: 1200px;
    background-color: lightskyblue;
    margin: 30px auto;
    display: flex;
    justify-content: space-between;
  }

  .menu_wrap > li{
    width: 15%;
    height: 60px;
    background-color: lightgrey;
    /* text-align: center;
    line-height: 60px; */
    position: relative;
  }

  .menu_wrap > li > a{
    display: flex;
    align-items: center;
    justify-content: center;
    width: 100%;
    height: 100%;
    font-size: 20px;
    font-weight: bold;
  }

  .menu_wrap > li span{
    /* span의 부모요소에 display:flex를 적용해줘서 inline 요소인 span에 width, height, margin 등 적용 가능 */
    width: 0%;
    height: 5px;
    background-color: blue;
    position: absolute;
    right: 0;
    bottom: 0;
    transition: all .3s;
  }

  .menu_wrap > li:hover span {
    width: 100%;
  }

</style>
</head>
<body>

<ul class="menu_wrap">
  <li>
    <a href="#">menu1<span></span></a>
  </li>
  <li>
    <a href="#">menu2<span></span></a>
  </li>
  <li>
    <a href="#">menu3<span></span></a>
  </li>
  <li>
    <a href="#">menu4<span></span></a>
  </li>
</ul>


```

```

        </li>
        <li>
            <a href="#">menu5<span></span></a>
        </li>
        <li>
            <a href="#">menu6<span></span></a>
        </li>
    </ul>

```

※ 4) 메뉴 hover event : scale 사용 (과제 풀이)

```

<!DOCTYPE html>
<html lang="kr">
<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>hover_menu_과제</title>
    <style>
        *{
            margin: 0;
            padding: 0;
            box-sizing: border-box;
        }

        ul{
            list-style: none;
        }

        a{
            text-decoration: none;
            color: initial;
        }

        /* reset */

        .menu_wrap, .menu_wrap_2 , .menu_wrap_3{
            width: 1200px;
            background-color: lightskyblue;
            margin: 30px auto;
            display: flex;
            justify-content: space-between;
        }

        .menu_wrap > li , .menu_wrap_2 > li, .menu_wrap_3 > li{
            width: 15%;
            height: 60px;
            background-color: lightgrey;
            position: relative;
        }

        .menu_wrap > li > a, .menu_wrap_2 > li > a , .menu_wrap_3 > li > a{
            display: flex;
            align-items: center;
            justify-content: center;
            width: 100%;
            height: 100%;
            font-size: 20px;
            font-weight: bold;
        }

        .menu_wrap > li span{
            /* width: 0%; */
            height: 5px;
            background-color: blue;
            position: absolute;
            left: 50%;
            bottom: 0;
            transition: all .2s linear;
            border-bottom: 3px solid #000; /*
            position: absolute;
            left: 0;
            bottom: 0;
            width: 100%;
            height: 5px;
            opacity: 0;
            /* scale : 중앙을 기준으로 움직임 효과 적용 가능 */
            transform: scaleX(0);
            transition: all .3s;
            background-color: hotpink;
        }
    </style>

```

```

.menu_wrap > li:hover span{
    /* left: 0;
    width: 100%; */
    opacity: 1;
    transform: scaleX(1);
}

/* menu_wrap hover event_end */

.menu_wrap_2 > li span{
    /* width: 100%;
    height: 5px;
    position: absolute;
    top: 50%;
    transition: all .2s linear; */
    position: absolute;
    left: 0;
    bottom: 0;
    width: 100%;
    height: 100%;
    background-color: lightseagreen;
    transition: all .3s;
    transform: scale(1,0);
    /* 가로는 적용된 width 값을 유지하고 세로는 0 */
}

.menu_wrap_2 a{
    position: absolute;
    left: 0;
    bottom: 0;
    z-index: 999;
}

.menu_wrap_2 > li:hover span{
    /* top: 0;
    height: 100%;
    background-color: rgba(238,130,238,.5) */
    transform: scale(1,1);
}

/* menu_wrap_2 hover event_end */

.menu_wrap_3 > li span{
    /* width: 0%;
    height: 5px;
    background-color: blue;
    position: absolute;
    left: 50%;
    top: 40px;
    transition: all .1s linear;
    border-bottom: 3px solid #000; */
    width: 100%;
    height: 5px;
    background-color: salmon;
    position: absolute;
    left: 0;
    bottom: 50%;
    transition: all .3s;
    transform: scale(0,0);
    opacity: 0;
}

.menu_wrap_3 > li:hover span{
    /* left: 0;
    top: 55px;
    width: 100% */
    opacity: 1;
    transform: scale(1,1);
    bottom: 0;
}

</style>
</head>
<body>

<ul class="menu_wrap">
    <li>
        <a href="#">menu1<span></span></a>
    </li>
    <li>
        <a href="#">menu2<span></span></a>
    </li>
    <li>
        <a href="#">menu3<span></span></a>
    </li>
    <li>

```

```

        <a href="#">menu4<span></span></a>
    </li>
    <li>
        <a href="#">menu5<span></span></a>
    </li>
    <li>
        <a href="#">menu6<span></span></a>
    </li>
</ul>

<ul class="menu_wrap_2">
    <li>
        <span></span>
        <a href="#">menu1</a>
    </li>
    <li>
        <span></span>
        <a href="#">menu2</a>
    </li>
    <li>
        <span></span>
        <a href="#">menu3</a>
    </li>
    <li>
        <span></span>
        <a href="#">menu4</a>
    </li>
    <li>
        <span></span>
        <a href="#">menu5</a>
    </li>
    <li>
        <span></span>
        <a href="#">menu6</a>
    </li>
</ul>

<ul class="menu_wrap_3">
    <li>
        <a href="#">menu1<span></span></a>
    </li>
    <li>
        <a href="#">menu2<span></span></a>
    </li>
    <li>
        <a href="#">menu3<span></span></a>
    </li>
    <li>
        <a href="#">menu4<span></span></a>
    </li>
    <li>
        <a href="#">menu5<span></span></a>
    </li>
    <li>
        <a href="#">menu6<span></span></a>
    </li>
</ul>

</body>
</html>

```

※ 5) 드롭박스 메뉴

```

<!DOCTYPE html>
<html lang="ko">
<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>드롭박스 메뉴</title>
    <style>
        *{
            margin: 0;
            padding: 0;
            box-sizing: border-box;
        }

        ul{
            list-style: none;
        }

```

```

a{
    text-decoration: none;
    color: initial;
}

.drop_menu_1{
    width: 1200px;
    height: 50px;
    display: flex;
    justify-content: space-between;
    border-top: 1px solid #000;
    border-bottom: 1px solid #000;
}

.drop_menu_1 > li{
    width: 15%;
    background-color: lightgray;
}

.drop_menu_1 h3{
    line-height: 50px;
}

.drop_menu_1 h3 a{
    display: flex;
    justify-content: center;
    width: 100%;
    height: 100%;
}

.drop_menus{
    padding: 20px 0;
    background-color: lightblue;
    text-align: center;
    display: none;
}

.nav_1 li:hover a{
    color: crimson;
}

/*
nav_1 의 ul(drop_menu_1)에 마우스 오버 시 하단 메뉴(drop_menus) 팝업
이 경우 어떤 ul을 선택하더라도 모든 li(drop_menus)가 동시에 반응한다
.nav_1 .drop_menu_1:hover .drop_menus{
    display: block;
}
*/

/*
nav_1의 자식 ul(drop_menu_1)의 자식 li에 마우스 오버 시 하단메뉴(drop_menus) 팝업
이 경우 선택하는 ul에 따라 li(drop_menus)가 각각 반응한다
*/
.nav_1 .drop_menu_1 > li:hover .drop_menus{
    display: block;
}


```

</style>

</head>

<body>

```

<nav class="nav_1">
    <ul class="drop_menu_1">
        <li>
            <h3><a href="#">제품소개</a></h3>
            <ul class="drop_menus">
                <li><a href="#">sub_menu_1</a></li>
                <li><a href="#">sub_menu_2</a></li>
                <li><a href="#">sub_menu_3</a></li>
            </ul>
        </li>

        <li>
            <h3><a href="#">구매안내</a></h3>
            <ul class="drop_menus">
                <li><a href="#">sub_menu_1</a></li>
                <li><a href="#">sub_menu_2</a></li>
                <li><a href="#">sub_menu_3</a></li>
            </ul>
        </li>

        <li>
            <h3><a href="#">고객센터</a></h3>
            <ul class="drop_menus">
                <li><a href="#">sub_menu_1</a></li>
                <li><a href="#">sub_menu_2</a></li>
            </ul>
        </li>
    </ul>
</nav>

```

```

        <li><a href="#">sub_menu_3</a></li>
    </ul>
</li>

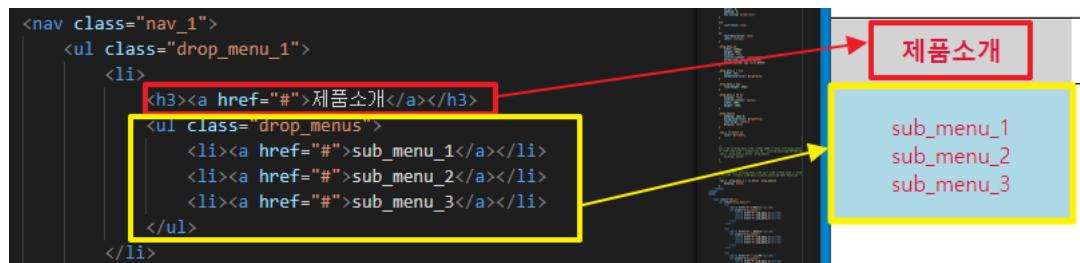
<li>
    <h3><a href="#">서비스</a></h3>
    <ul class="drop_menus">
        <li><a href="#">sub_menu_1</a></li>
        <li><a href="#">sub_menu_2</a></li>
        <li><a href="#">sub_menu_3</a></li>
    </ul>
</li>

<li>
    <h3><a href="#">홍보센터</a></h3>
    <ul class="drop_menus">
        <li><a href="#">sub_menu_1</a></li>
        <li><a href="#">sub_menu_2</a></li>
        <li><a href="#">sub_menu_3</a></li>
    </ul>
</li>

<li>
    <h3><a href="#">기업소개</a></h3>
    <ul class="drop_menus">
        <li><a href="#">sub_menu_1</a></li>
        <li><a href="#">sub_menu_2</a></li>
        <li><a href="#">sub_menu_3</a></li>
    </ul>
</li>
</ul>
</nav>
</body>
</html>

```

※ 드롭박스 메뉴 Markup 구조



[웹 컴파인 #2 총정리]

1. 중앙 정렬

- margin**

ex) inner

=> height 값 x (내부 컨텐츠 높이에 자동 설정 되도록)

=> float : left, margin : auto 혼용 사용 불가

=> width 확인 (block 요소이기 때문에 width 없을 경우 화면을 가득 채움)

- flex**

=> tag > tag 중앙 정렬 시 사용하기 좋음

=> container(부모)에 display : flex; 적용 시 안쪽에 있는 item(자식)의 display가 변경되는 경우가 있으니 확인하고 작업할 것!

- position**

=> 요소끼리 겹치는 경우에만 사용

=> relative(부모), absolute(자식)

==> relative 는 필요한 경우에만 적절히 사용
==> absolute된 자식 요소를 잡아주는 경우
==> 일부 레이아웃 조정을 위해서 사용하는 경우 (메틀로랩의 맵 위치 지정)

2. height

==> inner, section의 경우 height 값 x 권장
==> padding으로 부피감 효과를 줌으로써 height 대체 가능

3. width

- 00px; , calc(100% - 100px); ※calc 사용 시 연산자 사이에 여백 필수
- 단위 : %, vw, em, rem
- max-width / width / margin : 0 auto => inner

==> grid.guide

==> pc : 1200 ~ 1800px

==> 1000px 이하로 inner 사용 하지 않는 것을 권장

4. 미디어쿼리

```
1) PC
@media all and (min-width:1024px){}

2) tab
@media all and (min-width:768px) and (max-width:1023px){}

3) Mobile
@media all and (max-width:767px){}
```

==> 3개 중 하나를 기본 값(default)으로 잡고 작업하는 것이 좋음

5. :root

- common.css
- reset.css

```
:root{
    --mainColor : #5452f;
    --whiteColor :#fff;
    --eng : 'Noto Sans'
}

color : var(--mainColor);
```

6. em, rem

- 기준 : font-size
- root / rem **
- em / 개별div
- width, height, padding, margin, font-size ... (px를 사용하는 모든 요소에 사용 가능)

7. flex, grid

- 실용도 : flex(8) : grid(2)