

## ch01. 환경설정

### 1. 준비

- ① starbucks 폴더만들기 → images 폴더와 favicon 준비하기
- ② 기초 이미지 다운로드 받기
- ③ main.html 생성 → favicon 설정

### 2. 브라우저 설정하기

- ① 브라우저마다 다른 css 표현방식 통일하기
- 구글검색 : reset .css cdn → <https://www.jsdelivr.com/package/npm/reset-css> →



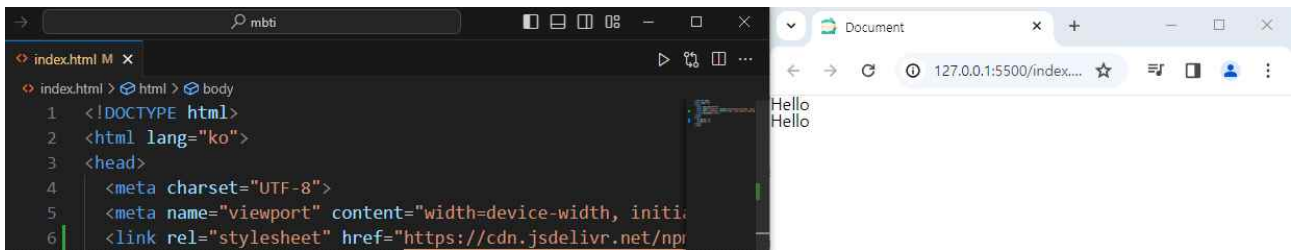
선택 후 HTML Copy하기 → index.html에 적용하기

- 변환 전 : 적용 전



- 적용 후 :

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/reset-css@5.0.2/reset.min.css">
```



- 적용할 css 생성하고 연결하기 → css 폴더생성 → main.css 파일 생성

```
body{  
  color : red;  
}
```

→ index.html에 css 연결하기

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

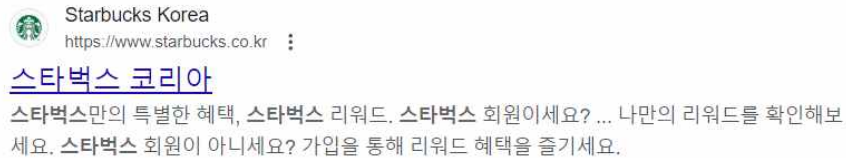
→ live server 확인

▶ index.html

```
<!DOCTYPE html>  
<html lang="ko">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <title>Starbucks Coffee Korea</title>  
  <!-- favicon.ico 이외 해상도 좋은 png 사용할 경우 -->  
  <link rel="icon" href="/favicon.png">  
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/reset-css@5.0.2/reset.min.css">  
  <link rel="stylesheet" href="/css/main.css">  
</head>  
<body>  
  Hello World  
</body>  
</html>
```

### 3. 오픈 그래프(The Open Graph protocol)

- ① 웹페이지가 소셜미디어(페이스북 등)로 공유될 때 우선적으로 활용되는 정보를 지정



- 사이트의 기본적인 메타정보를 끌어다가 간단하게 보여주는 것.



- ② 오픈 그래프 속성 및 작성하기

▷ 일반적인 오픈 그래프

- og:type: 페이지의 유형(E.g, website, video.movie)
- og:site\_name: 속한 사이트의 이름
- og:title: 페이지의 이름(제목)
- og:description: 페이지의 간단한 설명
- og:image: 페이지의 대표 이미지 주소(URL)
- og:url: 페이지 주소(URL)

▷ 트위터 카드(Twitter Cards) : 웹페이지가 소셜 미디어(트위터)로 공유될 때 우선적으로 활용되는 정보를 지정합니다.

- twitter:card: 페이지(카드)의 유형(E.g. summary, player)
- twitter:site: 속한 사이트의 이름
- twitter:title: 페이지의 이름(제목)
- twitter:description: 페이지의 간단한 설명
- twitter:image: 페이지의 대표 이미지 주소(URL)
- twitter:url: 페이지 주소(URL)

- ③ 우리가 사용할 기본 내용 처리하기(main.html <title> 태그 아래에 붙이기)

```
<meta property="og:type" content="website" />
<meta property="og:site_name" content="Starbucks" />
<meta property="og:title" content="Starbucks Coffee Korea" />
<meta property="og:description" content="스타벅스는 세계에서 가장 큰 다국적 커피 전문점으로,
64개국에서 총 23,187개의 매점을 운영하고 있습니다." />
```

```

<meta property="og:image" content="/images/starbucks_seo.jpg" />
<meta property="og:url" content="https://starbucks.co.kr" />
-----

<meta property="twitter:card" content="summary" />
<meta property="twitter:site" content="Starbucks" />
<meta property="twitter:title" content="Starbucks Coffee Korea" />
<meta property="twitter:description" content="스타벅스는 세계에서 가장 큰 다국적 커피 전문점으로, 64개국에서 총 23,187개의 매점을 운영하고 있습니다." />
<meta property="twitter:image" content="/images/starbucks_seo.jpg" />
<meta property="twitter:url" content="https://starbucks.co.kr" />

```

#### 4. 사용할 폰트의 지정? - 브라우저 마다 보여지는 폰트가 달라지므로...

- ① 대표적으로 크롬에서는 고딕체가 사파리에서는 바탕체로 보이기도 한다.
- ② 구글 폰트 적용 → 구글에서 google font 검색 → fonts.google.com → nanum 입력 → nanum gothic 클릭 → font size 400, 700, 800 지원 → 우리는 400과 700사용할 예정 임. → 우 상단 get Embed code → copy & paste

※ 폰트는 사이즈가 크다

▶ main.css

```

body {
  color : red;
  font-size: 100px;
  font-family: "Nanum Gothic", sans-serif;
}

```

▶ index.html

```

<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Starbucks Coffee Korea</title>

  <meta property="og:type" content="website" />
  <meta property="og:site_name" content="Starbucks" />
  <meta property="og:title" content="Starbucks Coffee Korea" />
  <meta property="og:description" content="스타벅스는 세계에서 가장 큰 다국적 커피 전문점으로, 64개국에서 총 23,187개의 매점을 운영하고 있습니다." />
  <meta property="og:image" content="/images/starbucks_seo.jpg" />
  <meta property="og:url" content="https://starbucks.co.kr" />

```

```

<meta property="twitter:card" content="summary" />
<meta property="twitter:site" content="Starbucks" />
<meta property="twitter:title" content="Starbucks Coffee Korea" />
<meta property="twitter:description" content="스타벅스는 세계에서 가장 큰 다국적 커피 전문점
      으로, 64개국에서 총 23,187개의 매점을 운영하고 있습니다." />
<meta property="twitter:image" content="/images/starbucks_seo.jpg" />
<meta property="twitter:url" content="https://starbucks.co.kr" />

<!-- favicon.ico 이외 해상도 좋은 png 사용할 경우 -->
<link rel="icon" href="/favicon.png">
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/reset-css@5.0.2/reset.min.css">
<link rel="preconnect" href="https://fonts.googleapis.com">
<link href="https://fonts.googleapis.com/css2?family=Nanum+Gothic&display=swap"
      rel="stylesheet">
<link rel="stylesheet" href="/css/main.css">
</head>
<body>
  Hello World
</body>
</html>

```

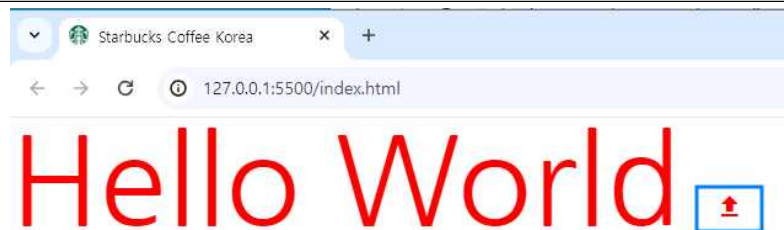
## 5. Google Material Icons

- ① 자주 사용하는 icon을 구글에서 제공하는 코딩형태의 것을 이용해 봅시다.
- ② 작고 자주 사용되는 아이콘을 매번 디자이너에게 요청할 수 없으니 이것이 편하다.
- ③ 사용법 → google material icons 검색 → 찾아서 사용할 수 있으나 버전의 문제로 아래의 것 사용

```
<link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons" />
```

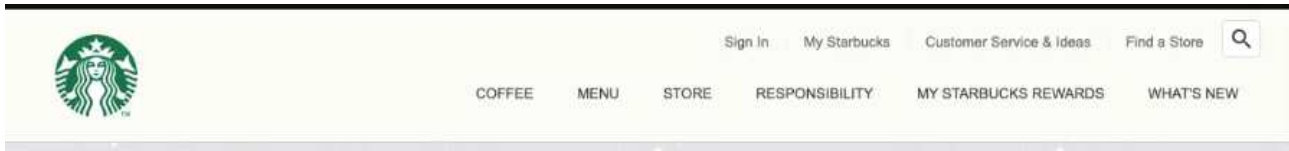
- ④ <link rel="stylesheet" href="/css/main.css"> 위 라인에 붙여넣기
- ⑤ 테스트 해 보기

```
<div class="material-icons">upload</div>
```



```
<div class="material-icons">search</div>
```

## Ch02. 헤더와 드롭다운 메뉴



### 1. 로고

#### ① 로고 삽입하고 영역설정하기

##### ▶ index.html

```
<body>
  <!-- 특정 기능은 없지만 이곳이 헤더영역이다라는 의미 -->
  <!-- HEADER -->
  <header>
    <div class="inner">
      <a href="/" class="logo">
        <!-- alt 대체속성 : 필수 태그 -->
        
      </a>
    </div>
  </header>
</body>
```

##### ▶ main.css

```
/* COMMON */
body {}
img {}

/* HEADER */
header {}
header .inner {}
header .logo {}
```

```
/* COMMON */
body {
  /* 어두운 회색 : 같은 값이 6개 연속이면 3개만 써도 됨. */
  color: #333333;
  font-size: 16px;
  font-weight: 400;
  /* 16픽셀의 1.4배 줄높이 - 불편함 없는 줄높이*/
  line-height: 1.4;
```

```

    font-family: "Nanum Gothic", sans-serif;
}

img {
    /* 모든 이미지를 인라인요소가 아닌 블록요소로... */
    display: block;
}

/* HEADER */
header {
    background-color: royalblue;
}

header .inner {
    /* 로고 포함된 헤더 영역을 창사이즈 및 확대/축소에 따라 가운데 위치시키는 내용 */
    /* 로고는 평균 높이가 75px */
    /* 이미지는 인라인 요소이며, 글자는 baseline이 존재해서 아래쪽 약간의 여백이 생긴다. */
    /* 해결방법 : img 요소를 블록으로 처리 */
    width: 1100px;
    margin: 0 auto;
    background-color: orange;
}

header .logo {
}

```

## ② inner 설정하기

### ▶ main.css

```

header .inner {}
header .logo {}

```

```

header .inner {
    /* 로고 포함된 헤더 영역을 창사이즈 및 확대/축소에 따라 가운데 위치시키는 내용 */
    /* 로고는 평균 높이가 75px */
    /* 이미지는 인라인 요소이며, 글자는 baseline이 존재해서 아래쪽 약간의 여백이 생긴다. */
    /* 해결방법 : img 요소를 블록으로 처리 */
    width: 1100px;
    height: 120px;
    margin: 0 auto;
    background-color: orange;
}

```

```

/* 부모는 relative 한 position을 가져야 한다. */
position: relative;
}

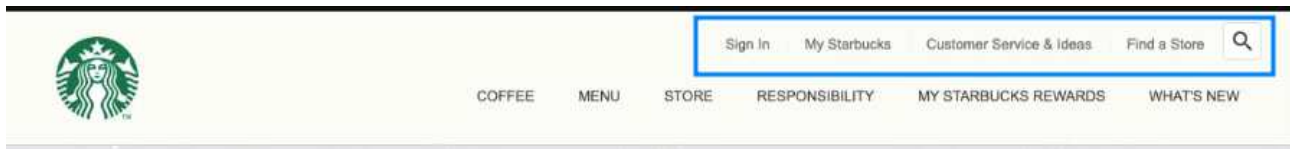
header .logo {
  /* 부모요소 기준 배치 */
  height: 75px;
  position: absolute;
  top: 0;
  bottom: 0;
  /* 부모기준 왼쪽에 붙어 있다. - 명시 */
  left: 0;
  margin: auto;
}

```

## 2. 서브메뉴

> 화면 상단 서브메뉴 작업

- 네 개의 메뉴, 세로 구분선, 링크, 검색 아이콘



① 상단 네 개 메뉴 만들기

▶ index.html

```

<body>

  <!-- 특정 기능은 없지만 이곳이 헤더영역이다라는 의미 -->
  <!-- HEADER -->
  <header>
    <div class="inner">
      <a href="/" class="logo">
        <!-- alt 대체속성 : 필수 태그 -->
        
      </a>
      <!-- 상단 서브메뉴 구역 -->
      <div class="sub-menu">
        <ul class="menu">
          <!-- 메뉴가 들어갈 영역 -->
          <li>
            <a href="/signin">Sign In</a>

```



```

        </li>
        <li>
            <a href="#">My Starbucks</a>
        </li>
        <li>
            <!-- #과 동일한 기능이나 #은 미세한 페이지 변화 있을 수 있음. -->
            <a href="javascriptvoid(0)">Customer Service & Ideas</a>
        </li>
        <li>
            <a href="javascriptvoid(0)">Find a Store</a>
        </li>
    </ul>
    <div class="search">
        <input type="text" />
        <div class="material-icons">search</div>
    </div>
</div>
</header>
</body>

```

► main.css

```

/* HEADER */
header .sub-menu ul.menu{}
header .sub-menu ul.menu li{}
header .sub-menu ul.menu li a{}
header .sub-menu .search{}
header .sub-menu .search input{}
header .sub-menu .search .material-icons{}

```

```

/* COMMON */
a {
    /* 모든 페이지의 a 태그 밑줄 사라짐 */
    text-decoration: none;
}
/* HEADER */
header .sub-menu {

```

**/\* 3) 메뉴를 수평으로 정렬 \*/**

```
header .sub-menu ul.menu{  
  font-family: Arial,sans-serif;  
  display: flex;  
}
```

**/\* 4-1) \*/**

```
header .sub-menu ul.menu li{  
  position: relative;  
}
```

**/\* 4) 세로구분선 만들기 :: 가상요소선택자\*/**

```
header .sub-menu ul.menu li::before{  
  /* 없으면 안됨 */  
  content: "";  
  display: block;  
  width: 1px;  
  height: 12px;  
  /* background-color: black; */  
  background-color: #e5e5e5;  
  /* 부모요소인 li의 가운데로 정렬, absolute는 block요소로 강제로 바꾼다. */  
  position: absolute;  
  top: 0;  
  bottom: 0;  
  margin: auto;  
}
```

**/\* 4-2) Sign In 좌측 세로선 없애기 \*/**

```
header .sub-menu ul.menu li:first-child::before{  
  display: none;  
}
```

**/\* 1) \*/**

```
header .sub-menu ul.menu li a{  
  /* background-color: red; */  
  font-size: 12px;  
  /* 글자 클릭 시 잘 안되는 것을 방지하기 위해 영역주고 block으로 변환 */  
  padding: 11px 16px;  
  display: block;
```

```

    color: #656565;
}
/* 2) 마우스 올리면 효과*/
header .sub-menu ul.menu li a:hover{
    color: #000;
}

header .sub-menu .search{}

header .sub-menu .search input{}

header .sub-menu .search .material-icons{}

```

### 3. 검색

① 검색 아이콘 및 텍스트 상자 처리하기

▶ main.css

```

header .sub-menu {}
header .sub-menu .search{}
header .sub-menu .search input{}
header .sub-menu .search .material-icons{}

```

```

header .sub-menu {
    display: flex;
    /* 8) 서브메뉴 우측 정렬 */
    position: absolute;
    top: 10px;
    right: 0;
}

/* 5) 서치메뉴 영역 */
header .sub-menu .search{
    /* 영역을 얼마나 차지하는가 */
    /* background-color: red; */
    height: 34px;
    position: relative;
}

```

#### **/\* 6) 서치 상자 처리\*/**

```
header .sub-menu .search input{
  width: 36px;
  height: 34px;
  padding: 4px 10px;
  border: 1px solid #ccc;
  /* 내부와 외부크기를 유지하면서 내부여백과 테두리선 추가 */
  box-sizing: border-box;
  border-radius: 5px;
  /* 선택하면 상자주변 선 생김 없애기 */
  outline: none;
  background-color: #fff;
  color: #777;
  /* 선택하면 입력영역이 적다. 크게 늘리자 */
  /* 자연스럽게 영역이 늘어나도록 transition 적용 */
  font-size: 12px;
  transition: width .4s;
}
```

#### **/\* 6-1) 선택하면 크기 늘이기 \*/**

```
header .sub-menu .search input:focus{
  width: 190px;
  border-color: #690;
}
```

#### **/\* 7) 돋보기 아이콘 \*/**

```
header .sub-menu .search .material-icons{
  /* 5)에다 position: relative 추가 */
  /* 구글 폰트 사이즈랑 높이같다. */
  height: 24px;
  position: absolute;
  top: 0;
  bottom: 0;
  right: 5px;
  margin: auto;
  /* submenu 클래스의 display: flex; 추가 변경 */
  /* 돋보기 클릭 시 텍스트가 열리지 않는다.
  * 애플은 css의 범위를 벗어나는 것으로 javascript로 처리
  * 해야 한다.
```

```

    */
    transition: 0.4s;
}

header .sub-menu .search.focused .material-icons{
    /* 포커스 받으면 돋보기 안보이게 하기 */
    opacity: 0;
}

```

▷ header 색상 변경 및 header .inner 정리

```

/* HEADER */
header {
    /* background-color: royalblue; */
    background-color: #f6f5f0;
    border-bottom: 1px solid #c8c8c8;
}

header .inner {
    /* 로고 포함된 헤더 영역을 창사이즈 및 확대/축소에 따라 가운데 위치시키는 내용 */
    /* 로고는 평균 높이가 75px */
    /* 이미지는 인라인 요소이며, 글자는 baseline이 존재해서 아래쪽 약간의 여백이 생긴다. */
    /* 해결방법 : img 요소를 블록으로 처리 */
    width: 1100px;
    height: 120px;
    margin: 0 auto;
    /* background-color: orange; */
    /* 부모는 relative 한 position을 가져야 한다. */
    position: relative;
}

```

② js 폴더 생성 → main.js 생성 → index.html <head>에 추가

▶ index.html

```

<head>
    -----
    <script defer src="./js/main.js"></script>
</head>

```

▶ main.js

```
const searchEl = document.querySelector('.search');
const searchInputEl = searchEl.querySelector('input');

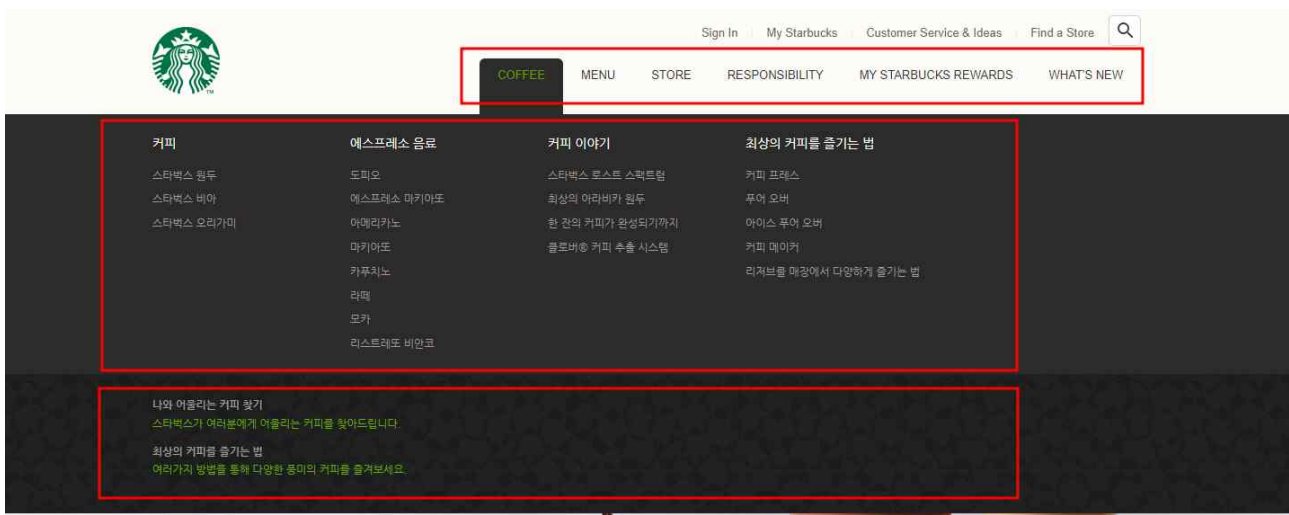
searchEl.addEventListener('click', function(){
  searchInputEl.focus();
})

// 입력상자가 포커스를 받으면 Placeholder에 통합검색 보이게...
searchInputEl.addEventListener('focus', function(){
  // search 클래스 밑에 focused 라는 클래스를 만들어 붙이겠다.
  searchEl.classList.add('focused');
  searchInputEl.setAttribute('placeholder', '통합검색');
})

searchInputEl.addEventListener('blur', function(){
  // search 클래스 밑에 focused 라는 클래스를 삭제하겠다.
  searchEl.classList.remove('focused');
  searchInputEl.setAttribute('placeholder', '');
})
```

## 4. 메인메뉴(1)

▷ 메뉴와 DropDown 메뉴 구성하기



① 상단 서브메뉴 구역 설정

▶ index.html

```
<!-- 상단 서브메뉴 구역 -->
```

```
<div class="sub-menu">
  <ul class="menu">
    <!-- 메뉴가 들어갈 영역 -->
    <li>
      <a href="/signin">Sign In</a>
    </li>
    <li>
      <a href="#">My Starbucks</a>
    </li>
    <li>
      <!-- #과 동일한 기능이나 #은 미세한 페이지 변화 있을 수 있음. -->
      <a href="javascript:void(0)">Customer Service & Ideas</a>
    </li>
    <li>
      <a href="javascript:void(0)">Find a Store</a>
    </li>
  </ul>
  <div class="search">
    <input type="text" />
    <div class="material-icons">search</div>
  </div>
</div>

<!-- 메뉴와 드롭다운메뉴 구역 -->
<ul class="main-menu">
  <li class="item">
    <div class="item__name">COFFEE</div>
    <div class="item__contents">

    </div>
  </li>

  <li class="item">
    <div class="item__name">MENU</div>
    <div class="item__contents">

    </div>
  </li>

  <li class="item">
```

```

        <div class="item__name">STORE</div>
        <div class="item__contents">

    </div>
</li>

<li class="item">
    <div class="item__name">RESPONSIBILITY</div>
    <div class="item__contents">

    </div>
</li>

<li class="item">
    <div class="item__name">MY STARBUCKS REWARDS</div>
    <div class="item__contents">

    </div>
</li>

<li class="item">
    <div class="item__name">WHAT'S NEW</div>
    <div class="item__contents">

    </div>
</li>
</ul>

```

#### ▶ main.css – 기본 메인메뉴 배치

```

/* HEADER - MAINMENU */
header .main-menu{}
header .main-menu .item{}
header .main-menu .item .item__name{}
header .main-menu .item .item__contents{}

```

```

/* HEADER - MAINMENU */
header .main-menu{
    position: absolute;
    bottom: 0;
    right: 0;
    /* 요소의 쌓인 순서 배치*/
}

```



```

z-index: 1;
display: flex;
}

header .main-menu .item{

}

header .main-menu .item .item_name{
padding: 10px 20px 34px 20px;
font-family: Arial, sans-serif;
font-size: 13px;
}

/* .item에 hover를 해야 하위에도 적용됨. */
header .main-menu .item:hover .item_name{
background-color: #2c2c29;
color: #690;
border-radius: 6px 6px 0 0;
}

header .main-menu .item .item_contents{
}

```



## ② 메인메뉴 선택 시 나타나는 드롭다운 메뉴의 구성(menu, texture)

▶ index.html – COFFEE, MENU, STORE 구성하고 드롭다운 구성하기

```

<header>
  <div class="inner">
    <a href="/" class="logo">
      <!-- alt 대체속성 : 필수 태그 -->
      
    </a>
    <!-- 상단 서브메뉴 구역 -->
    <div class="sub-menu">
      <ul class="menu">

```

```

<!-- 메뉴가 들어갈 영역 -->
<li>
  <a href="/signin">Sign In</a>
</li>
<li>
  <a href="#">My Starbucks</a>
</li>
<li>
  <!-- #과 동일한 기능이나 #은 미세한 페이지 변화 있을 수 있음. -->
  <a href="javascript:void(0)">Customer Service & Ideas</a>
</li>
<li>
  <a href="javascript:void(0)">Find a Store</a>
</li>
</ul>
<div class="search">
  <input type="text" />
  <div class="material-icons">search</div>
</div>
</div>

<!-- 메뉴와 드롭다운메뉴 구역 -->
<ul class="main-menu">
  <li class="item">
    <div class="item__name">COFFEE</div>
    <div class="item__contents">
      <!-- 메뉴밀 세부메뉴와 그 밑 텍스트 -->
      <div class="contents_menu"></div>
      <div class="contents_texture"></div>
    </div>
  </li>

  <li class="item">
    <div class="item__name">MENU</div>
    <div class="item__contents">
      <!-- 메뉴밀 세부메뉴와 그 밑 텍스트 -->
      <div class="contents_menu"></div>
      <div class="contents_texture"></div>
    </div>
  </li>
</ul>

```

```

<li class="item">
  <div class="item__name">STORE</div>
  <div class="item__contents">
    <!-- 메뉴밑 세부메뉴와 그 밑 텍스트 -->
    <div class="contents_menu"></div>
    <div class="contents_texture"></div>
  </div>
</li>

<li class="item">
  <div class="item__name">RESPONSIBILITY</div>
  <div class="item__contents">

    </div>
</li>

```

---

#### ► main.css

```

header .main-menu .item .item__contents{}
header .main-menu .item .item__contents .contents_menu{}
header .main-menu .item .item__contents .contents_texture{}

```

```

header .main-menu .item .item__contents{
  /* 아래 두개 메뉴 층 viewport 기준으로 펼침 */
  /* top이나 bottom 속성을 사용하지 않아 수직 위치 값이 없다면
  요소의 원래 위치를 그대로 사용합니다.
  만약 position: absolute;를 사용했다면, (위치상)부모 요소를
  기준으로 하므로 화면의 뷰포트 좌우 끝까지 늘어날 수 없게 됩니다.
  */
  width: 100%;
  position: fixed;
  left: 0;
  /* hover 효과로 나타내기 위해 숨김 */
  display: none;
}

header .main-menu .item:hover .item__contents{
  display: block;
}

```

```

header .main-menu .item .item__contents .contents__menu{
  /* 시각적 위치 확인 */
  background-color: red;
  height: 200px;
}

header .main-menu .item .item__contents .contents__texture{
  /* 시각적 위치 확인 */
  background-color: orange;
  height: 100px;
}

```

### ③ 임의로 내용 넣고 테스트하면서 처리하기

#### ▶ index.html

```

<li class="item">
  <div class="item__name">COFFEE</div>
  <div class="item__contents">
    <!-- 메뉴밀 세부메뉴와 그 밑 텍스트 -->
    <div class="contents__menu">COFFEE</div>
    <div class="contents__texture">COFFEE</div>
  </div>
</li>

<li class="item">
  <div class="item__name">MENU</div>
  <div class="item__contents">
    <!-- 메뉴밀 세부메뉴와 그 밑 텍스트 -->
    <div class="contents__menu">MENU</div>
    <div class="contents__texture">MENU</div>
  </div>
</li>

<li class="item">
  <div class="item__name">STORE</div>
  <div class="item__contents">
    <!-- 메뉴밀 세부메뉴와 그 밑 텍스트 -->
    <div class="contents__menu">STORE</div>
    <div class="contents__texture">STORE</div>
  </div>
</li>

```

▷ 각각의 텍스트가 두개 층 맨 좌측 상단에 고정되어 있다.

※ inner 라는 클래스는 한 줄의 콘텐츠를 가운데 몰아주는 역할을 하는 것이다.

※ header .inner 요소를 분리해서 자주 사용할 수 있도록 위쪽에 재정의의를 한다.

▶ index.html

```
<div class="item_name">COFFEE</div>
  <div class="item_contents">
    <!-- 메뉴밑 세부메뉴와 그 밑 텍스트 -->
    <div class="contents_menu">
      <div class="inner">
        COFFEE
      </div>
    </div>
    <div class="contents_texture">
      <div class="inner">
        COFFEE
      </div>
    </div>
  </div>
```

▶ main.css

```
/* COMMON */
a {
  /* 모든 페이지의 a 태그 밑줄 사라짐 */
  text-decoration: none;
}

.inner{
  width: 1100px;
  margin: 0 auto;
  position: relative;
}

/* HEADER */

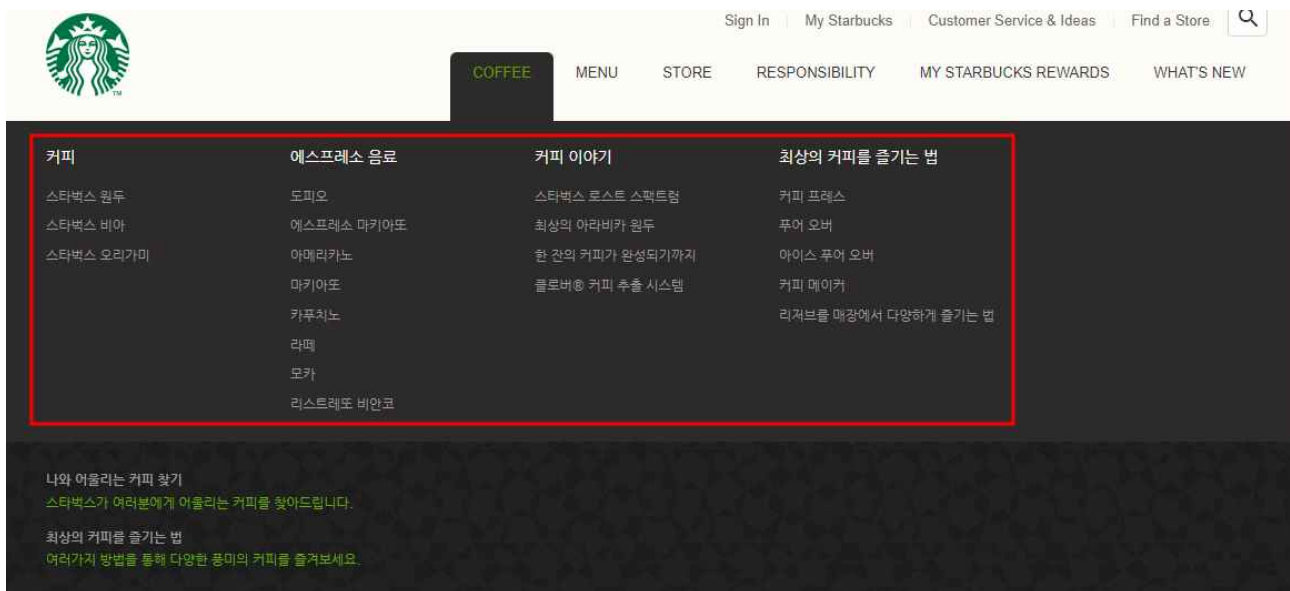
/* header .inner { */ → 삭제
/* header의 바로 자식요소인 inner 에만 적용 */
header > .inner {
  /* 로고 포함된 헤더 영역을 창사이즈 및 확대/축소에 따라 가운데 위치시키는 내용 */
  /* 로고는 평균 높이가 75px */
  /* 이미지는 인라인 요소이며, 글자는 baseline이 존재해서 아래쪽 약간의 여백이 생긴다. */
```

```

/* 해결방법 : img 요소를 블록으로 처리 */
/* width: 1100px; */ → 중복 부분 모두 삭제
height: 120px;
/* margin: 0 auto; */ → 중복 부분 모두 삭제
/* background-color: orange; */
/* 부모는 relative 한 position을 가져야 한다. */
/* position: relative; */ → 중복 부분 모두 삭제
}

```

#### ④ 세부 메뉴 구성하기 – 메뉴를 선택하면 나오는 상단 메뉴 소개



#### ▶ index.html

```

<div class="contents__menu">
  <ul class="inner"> → div 태그를 ul 태그로 변경
    <li>
      <h4>커피</h4>
      <ul>
        <li>스타벅스 원두</li>
        <li>스타벅스 비아</li>
        <li>스타벅스 오리가미</li>
      </ul>
    </li>
    <li>
      <h4>에스프레소 음료</h4>
      <ul>
        <li>도피오</li>

```

```

        <li>에스프레소 마키아또</li>
        <li>아메리카노</li>
        <li>마키아또</li>
        <li>카푸치노</li>
        <li>라떼</li>
        <li>모카</li>
        <li>리스트레또 비안코</li>
    </ul>
</li>
</ul>
</div>

```

#### ▶ main.css

```

header .main-menu .item .item__contents .contents__menu{
    /* 시각적 위치 확인 */
    background-color: red;
    height: 200px;
} → 애 아래에 위치....

header .main-menu .item .item__contents .contents__menu > ul{
}
header .main-menu .item .item__contents .contents__menu > ul > li{
}
header .main-menu .item .item__contents .contents__menu > ul > li h4{
}

header .main-menu .item .item__contents .contents__menu > ul > li ul{
}

header .main-menu .item .item__contents .contents__menu > ul > li ul li{
}

```

#### ▶ main.css

```

header .main-menu .item .item__contents .contents__menu{
    /* 시각적 위치 확인 */
    background-color: red;
    height: 200px;
} → 애 아래에 위치....

header .main-menu .item .item__contents .contents__menu > ul{

```

```

}
header .main-menu .item .item__contents .contents__menu > ul > li{
}
header .main-menu .item .item__contents .contents__menu > ul > li h4{
}

header .main-menu .item .item__contents .contents__menu > ul > li ul{
}

header .main-menu .item .item__contents .contents__menu > ul > li ul li{
}

```

► main.css

```

header .main-menu .item .item__contents .contents__menu{
  /* 시각적 위치 확인 */
  /* background-color: red; */
  background-color: #2c2c29;
  /* 메뉴마다 길이가 다르니 auto 자동설정 */
  /* height: 200px; */
} → 맨 마지막 수정

header .main-menu .item .item__contents .contents__menu > ul{
  /* 메뉴 수평 배치 */
  display: flex;
  padding: 20px;
}
header .main-menu .item .item__contents .contents__menu > ul > li{
  /* 메뉴 영역 각각 200px로 고정 */
  width: 200px;
}
header .main-menu .item .item__contents .contents__menu > ul > li h4{
  padding: 3px 0 12px 0;
  font-size: 14px;
  color: #fff;
}

header .main-menu .item .item__contents .contents__menu > ul > li ul{
}

header .main-menu .item .item__contents .contents__menu > ul > li ul li{
}

```



```
padding: 5px 0;
font-size: 12px;
color: #999;
cursor: pointer;
}

header .main-menu .item .item__contents .contents__texture{
  /* 시각적 위치 확인 */
  background-color: orange;
  height: 100px;
}
```

##### ⑤ 세부 메뉴 구성하기 - 메뉴를 선택하면 나오는 하단 texture

커피	에스프레소 음료	커피 이야기	최상의 커피를 즐기는 법
스타벅스 원두	도피오	스타벅스 로스트 스펙트럼	커피 프레스
스타벅스 비아	에스프레소 마키아또	최상의 아라비카 원두	푸어 오버
스타벅스 오리가미	아메리카노	한 잔의 커피가 완성되기까지	아이스 푸어 오버
	마키아또	클로버® 커피 추출 시스템	커피 메이커
	카푸치노		리저브를 매장에서 다양하게 즐기는 법
	라떼		
	모카		
	리스트레또 비안코		

나와 어울리는 커피 찾기  
 스타벅스가 여러분에게 어울리는 커피를 찾아드립니다.

최상의 커피를 즐기는 법  
 여러가지 방법을 통해 다양한 풍미의 커피를 즐겨보세요.

##### ▶ index.html

```
<div class="contents__texture">
  <div class="inner">
    <h4>나와 어울리는 커피 찾기</h4>
    <!-- div 태그도 무방함. -->
    <p>스타벅스가 여러분에게 어울리는 커피를 찾아드립니다.</p>
    <h4>최상의 커피를 즐기는 법</h4>
    <p>여러가지 방법을 통해 다양한 풍미의 커피를 즐겨보세요.</p>
  </div>
</div>
```

##### ▶ main.css

```
header .main-menu .item .item__contents .contents__texture{
  /* 시각적 위치 확인 */
  background-color: orange;
  height: 100px;
}
header .main-menu .item .item__contents .contents__texture h4{}
header .main-menu .item .item__contents .contents__texture p{}
```

#### ▶ main.css

```
header .main-menu .item .item__contents .contents__texture{
  /* 시각적 위치 확인 */
  /* background-color: orange;
  height: 100px; */ → 삭제
  padding: 26px 0;
  font-size: 12px;
  /* 패턴 사용해서 처리 */
  background-image: url("../images/main_menu_pattern.jpg");
}

header .main-menu .item .item__contents .contents__texture h4{
  color: #999;
  font-weight: 700;
}

header .main-menu .item .item__contents .contents__texture p{
  color: #690;
  /* 세개 입력시 상, 중(좌우), 하 */
  margin: 4px 0 14px;
}
```

⑥ 나머지 전체 메뉴 구성 코딩하기

⑦ 메뉴 선택시 색 변화 시키기

```
header .main-menu .item .item__contents .contents__menu > ul > li ul li:hover{
  color: #690;
}
```

## 5. BEM

① BEM(Block Element Modifier) : HTML 클래스 속성의 작명법

- 요소\_일부분 : Underscore(Lodash) 기호로 요소의 일부분을 표시

- 요소—상태 : Hyphen(Dash) 기호로 요소의 상태를 표시

```
<body>
  <div class="container">
    <div class="name"></div>
    <div class="item">
      <div class="name"></div>
    </div>
  </div>

  <div class="btn primary"></div>
  <div class="btn success"></div>
  <div class="btn error"></div>
</body>
```

```
<body>
  <div class="container">
    <div class="container__name"></div>
    <div class="item">
      <div class="item__name"></div>
    </div>
  </div>

  <div class="btn btn--primary"></div>
  <div class="btn btn--success"></div>
  <div class="btn btn--error"></div>
</body>
```

## 6. 전역 배지(GSAP)



※ 화면 우측에 보이는 이미지 세로 배열 두개

- 화면 하단으로 스크롤 시 따라오며
- 어느 정도 따라오다가 사라짐
- 다시 해당위치로 스크롤 업 되었을 때 다시 나타나.
- index.html 의 inner 가 끝나는 부분에서 작업 시작

### ① 이미지 배치하기

▶ index.html

```
<div class="badges">
  <div class="badge">
    
  </div>
  <div class="badge">
    
  </div>
</div>
```

▶ main.css

```
header .badges{
```

```
header .badges.badge{
```

► main.css

```
/* HEADER */
```

```
header {
```

```
    position: relative;
```

```
    /* background-color: royalblue; */
```

```
    background-color: #f6f5f0;
```

```
    border-bottom: 1px solid #c8c8c8;
```

```
}
```

```
header .badges{
```

```
    position: absolute;
```

```
    /* header에 position이 있는지 확인 */
```

```
    top: 132px;
```

```
    right: 12px;
```

```
}
```

```
/* 그림자 등 이미지 설정 */
```

```
header .badges .badge{
```

```
    border-radius: 10px;
```

```
    /* 둥글게 깎은 밖으로 이미지가 넘치지 않도록 설정 */
```

```
    overflow: hidden;
```

```
    margin-bottom: 12px;
```

```
    box-shadow: 4px 4px 10px rgba(0, 0, 0, 0.15);
```

```
    /* 아래로 내려가면서 따라오는지 확인하기 위해서
```

```
        body 부분의 height 를 임시로 3000px 지정
```

```
    */
```

```
}
```

```
/* COMMON */
```

```
body {
```

```
    /* 어두운 회색 : 같은 값이 6개 연속이면 3개만 써도 됨. */
```

```
    color: #333333;
```

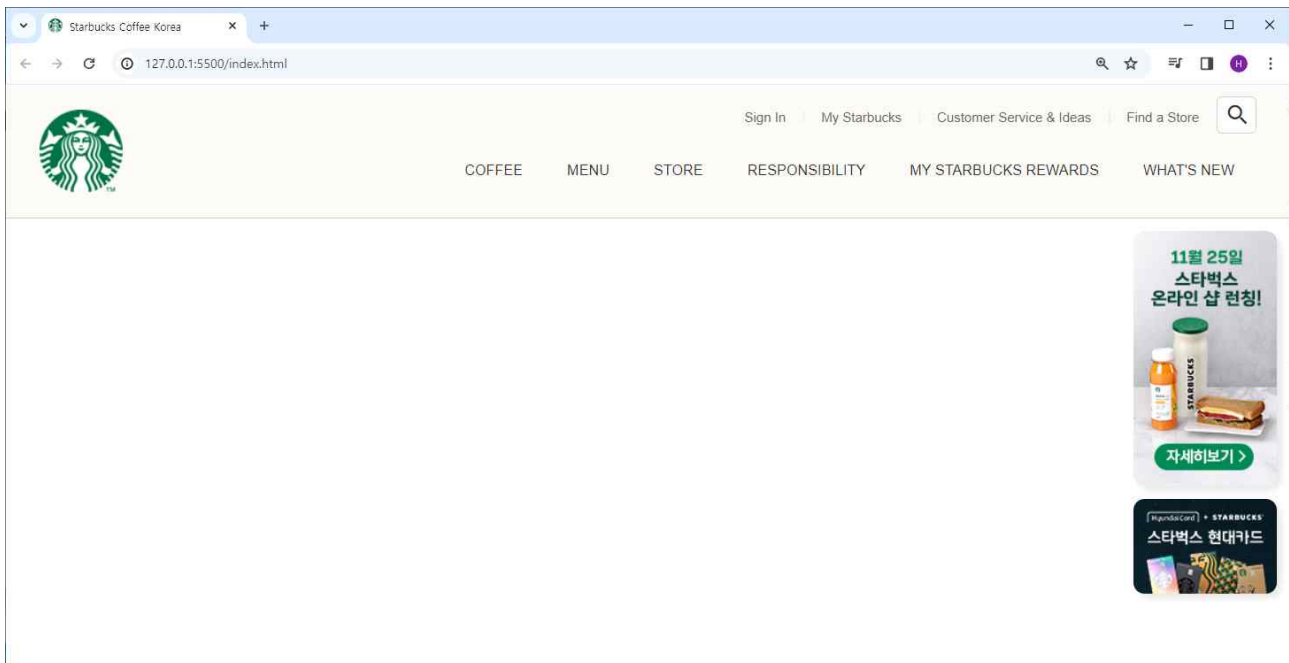
```
    font-size: 16px;
```

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

```
    /* 16픽셀의 1.4배 줄높이 - 불편함 없는 줄높이*/
```

```
    line-height: 1.4;
```

```
font-family: "Nanum Gothic", sans-serif;
height: 3000px;
}
```



## ② 이미지 스크롤 처리하기

▶ main.css

```
/* HEADER */
header {
  /* 배지처리 - 현재 position 주석처리 - fixed로 변경예정*/
  /* position: relative; */
  /* position
    fixed : 브라우저의 ViewPort 기준으로 정렬하겠다.
    fixed : 부모 노드의 기준요소로 사용
    fixed or absolute : 가로너비를 최소로 줄어들을려고 시도함.
  */

  /* 스크롤 시 우측 두개의 이미지 고정하기 */
  position: fixed;
  width: 100%;
  top: 0;

  /* background-color: royalblue; */
  background-color: #f6f5f0;
```

```
border-bottom: 1px solid #c8c8c8;
}
```

③ 특정 위치까지 내려가면 이미지 사라짐 처리하기(JavaScript 사용)

- scroll 이벤트가 수십 개 이상 발생하니 화면이 버벅이는 현상이 생김
- 이를 방지하기 위해 외부에서 제공하는 자바 라이브러리를 사용해 보자
- google 검색 창 → lodash cdn 검색



검색결과 약 215,000개 (0.23초)



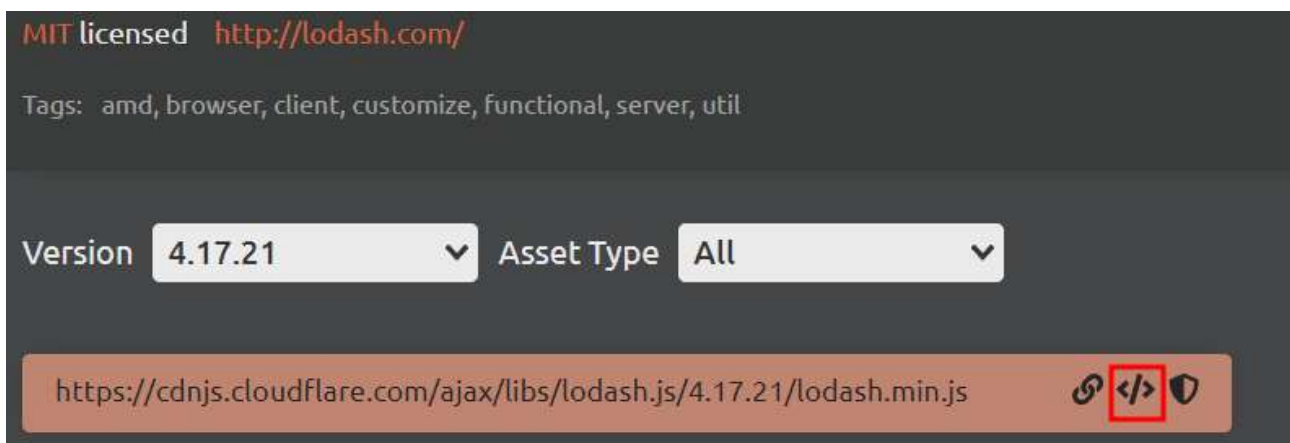
**lodash.js - Libraries - cdnjs - The #1 free and open source ...**

A utility library delivering consistency, customization, performance, & extras. - Simple. Fast. Reliable. Content delivery at its finest. cdnjs is a free ...



**Lodash**

A JavaScript utility library delivering consistency, modularity, performance, & extras.



▶ main.html

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/lodash.js/4.17.21/lodash.min.js" integrity="sha512-WFN04846sdKMIP5LKNpMaWzU7YpMyCU245etK3g/2ARYbPK9Ub18eG+ljU96qKR CWh+quCY7yefSmlkQw1ANQ==" crossorigin="anonymous" referrerpolicy="no-referrer"></script>
```

► main.js

```
const badgeEl = document.querySelector('header .badges');

// scroll 이벤트가 수십개 이상발생하니 화면이 버벅이는 현상이 생김
// 이를 방지하기 위해 외부에서 제공하는 자바라이브러리를 사용해 보자
// 0.3초 마다 scroll 이벤트가 발생되도록 처리
// _.throttle(함수, 시간)
window.addEventListener('scroll', _.throttle(function(){
  // console.log('scroll');
  console.log(window.scrollY);
  if(window.scrollY > 500){
    //배지 숨기기
    badgeEl.style.display = 'none';
  } else {
    //배지 보이기
    badgeEl.style.display = 'block';
  }
},300));
```

④ gsap cdn 사용하기

- 부자연스러운 배지의 나타남과 사라짐을 유연하게 자바스크립트 애니메이션 사용하기
- google 검색 창 → gsap cdn 검색
- index.html → main.js 보다 먼저 불러지게 위에 복사

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/gsap/3.12.5/gsap.min.js" integrity="sha512-7eHRwcbYkK4d9g/6tD/mhkf++eoTHwpNM9woBxtPUBWm67zeAfFC+HrdoE2GanKe oclY/VxeLvlqwvCdk7qScg==" crossorigin="anonymous" referrerpolicy="no-referrer"></script>
```

► main.js

```
const badgeEl = document.querySelector('header .badges');

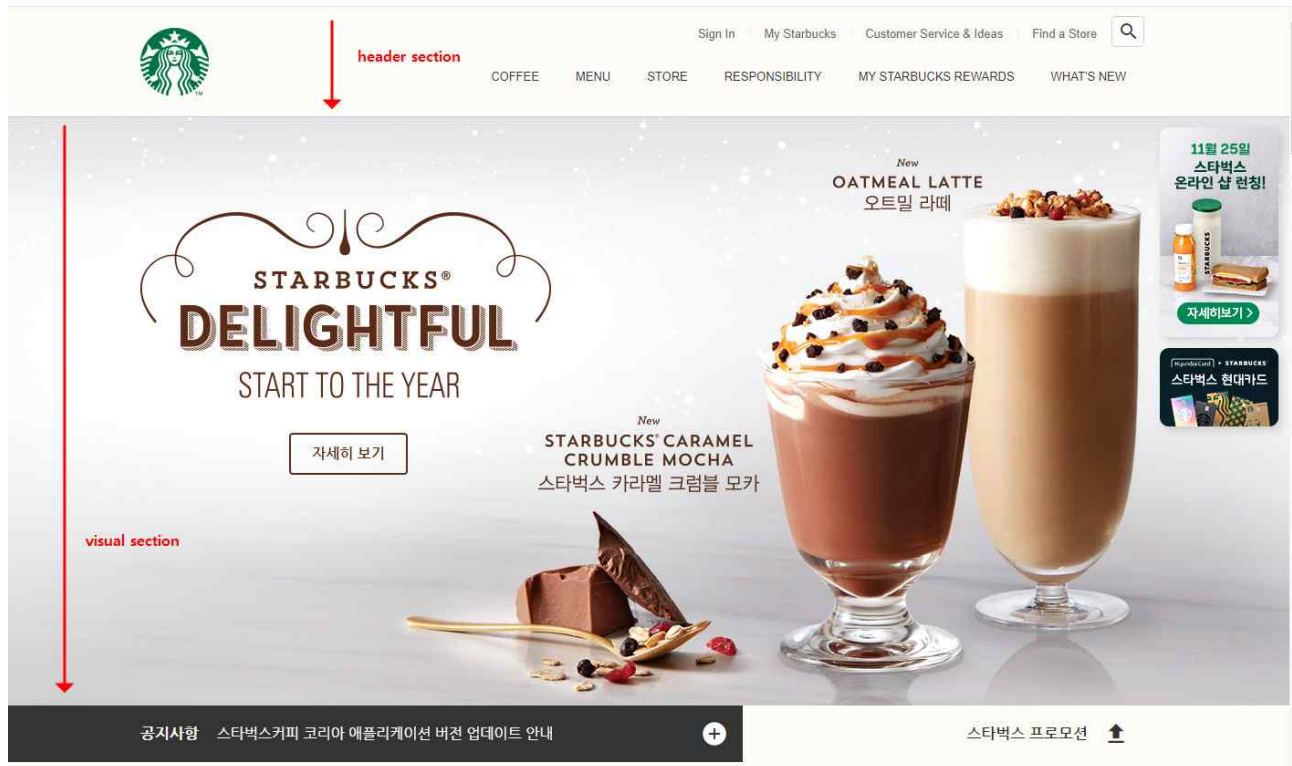
// scroll 이벤트가 수십개 이상발생하니 화면이 버벅이는 현상이 생김
// 이를 방지하기 위해 외부에서 제공하는 자바라이브러리를 사용해 보자
// 0.3초 마다 scroll 이벤트가 발생되도록 처리
// _.throttle(함수, 시간)
window.addEventListener('scroll', _.throttle(function(){
```

```
// console.log('scroll');
console.log(window.scrollY);
if(window.scrollY > 500){
    //배지 숨기기
    // badgeEl.style.display = 'none';
    // gsap.to(요소, 지속시간(초), 옵션)
    gsap.to(badgeEl, 0.6, {
        opacity : 0,
        display : 'none',
    });
} else {
    //배지 보이기
    // badgeEl.style.display = 'block';
    gsap.to(badgeEl, 0.6, {
        opacity : 1,
        display : 'block',
    });
}
},300));
```



## Ch03. 순차적 애니메이션

### 1. 전역 버튼 스타일(1)



① visual section의 이미지들이 순차적으로 나타나는 애니메이션 구현(Javascript)

▶ index.html

```
</header>

<!-- VISUAL -->
<section class="visual">
  <div class="inner">
    <div class="title">
      <!-- alt : 대체텍스트는 이미지 안의 문자열로 ... -->
      
      <a href="javascript:void(0)" class="btn">자세히 보기</a>
    </div>

    
```

```



</div>
</section>
```

► main.css

```
/* VISUAL */
.visual {}
.visual .inner{}
.visual .title{}
.visual .title .btn{}
.visual .cup1.image{}
.visual .cup1.text{}
.visual .cup2.image{}
.visual .cup2.text{}
.visual .spoon{}
```

► main.css

```
/* VISUAL */
.visual {
  /* header의 높이인 120px로 ... */
  margin-top: 120px;
  background-image: url("../images/visual_bg.jpg");
  background-position: center;
}

/* 위에서 지정한 .inner가 공통 적용됨 */
.visual .inner{
  height: 646px;
  background-color: orange;
}

.visual .title{
  position: absolute;
  top: 88px;
  left: -10px;
}
```

```

.visual .title .btn{}

.visual .cup1.image{
    position: absolute;
    bottom: 0;
    right: -47px;
}

.visual .cup1.text{
    position: absolute;
    top: 38px;
    right: 171px;
}

.visual .cup2.image{
    position: absolute;
    bottom: 0;
    right: 162px;
}

.visual .cup2.text{
    position: absolute;
    top: 321px;
    right: 416px;
}

.visual .spoon{
    position: absolute;
    bottom: 0;
    left: 275px;
}

```

② 우측 배지부분이 가려질 것을 대비해서 header의 z-index 값을 비교적 큰 값인 9로 변경

► main.css

```

/* HEADER */
header {
    /* 배지처리 - 현재 position 주석처리 - fixed로 변경예정*/
    /* position: relative; */
    /* position
        fixed : 브라우저의 ViewPort 기준으로 정렬하겠다.
        fixed : 부모 노드의 기준요소로 사용
    */
}

```

fixed or absolute : 가로너비를 최소로 줄어들을려고 시도함.

\*/

/\* 스크롤 시 우측 두개의 이미지 고정하기 \*/

position: fixed;

width: 100%;

top: 0;

/\* background-color: royalblue; \*/

background-color: #f6f5f0;

border-bottom: 1px solid #c8c8c8;

**z-index: 9;**

}

## 2. 전역 버튼 스타일(2)



### ① 전역 버튼 만들기

- 사이트 전체에서 고루 사용되고 있는 버튼의 모양을 통일
- 클래스로 정의해서 반복 재 활용
- COMMON 영역에 정의

### ▶ main.css

/\* COMMON \*/

-----

.btn {

width: 130px;

padding: 10px;

border: 2px solid #333;

border-radius: 4px;

color: #333;

```

font-size: 16px;
font-weight: 700;
text-align: center;
cursor: pointer;
/* 요소의 크기계산 기준을 border-box로
   패딩이 들어가고 보더가 들어간 만큼 요소가
   커지지 않도록...
*/
box-sizing: border-box;
display: block;
transition: 0.4s;
}

.btn:hover {
  background-color: #333;
  color: #fff;
}

```

## ② 다양한 버튼 스타일 만들기 놓기

▶ main.css

```

/* 다양한 버튼 스타일 만들기
   -- 기본 버튼 스타일보다 우선 적용된다.
*/
.btn.btn--reverse {
  background-color: #333;
  color: #fff;
}

.btn.btn--reverse:hover {
  background-color: transparent;
  color: #333;
}

.btn.btn--brown {
  color: #592b18;
  border-color: #592b18;
}

.btn.btn--brown:hover {
  color: #fff;
}

```

```

background-color: #592b18;
}

.btn.btn--gold {
  color: #d9aa8a;
  border-color: #d9aa8a;
}

.btn.btn--gold:hover {
  color: #fff;
  background-color: #d9aa8a;
}

.btn.btn--white {
  color: #fff;
  border-color: #fff;
}

.btn.btn--white:hover {
  color: #333;
  background-color: #fff;
}

```

### ③ 기본 버튼형을 브라운 타입의 버튼으로 변경하기

▶ main.html

```

<div class="title">
  <!-- alt : 대체텍스트는 이미지 안의 문자열로 ... -->
  
  <a href="javascript:void(0)" class="btn">자세히 보기</a>
</div>

```

```

<a href="javascript:void(0)" class="btn btn--brown">자세히 보기</a>

```

### ④ 버튼 위치 지정하기

▶ main.css

```

.visual .title .btn{

```

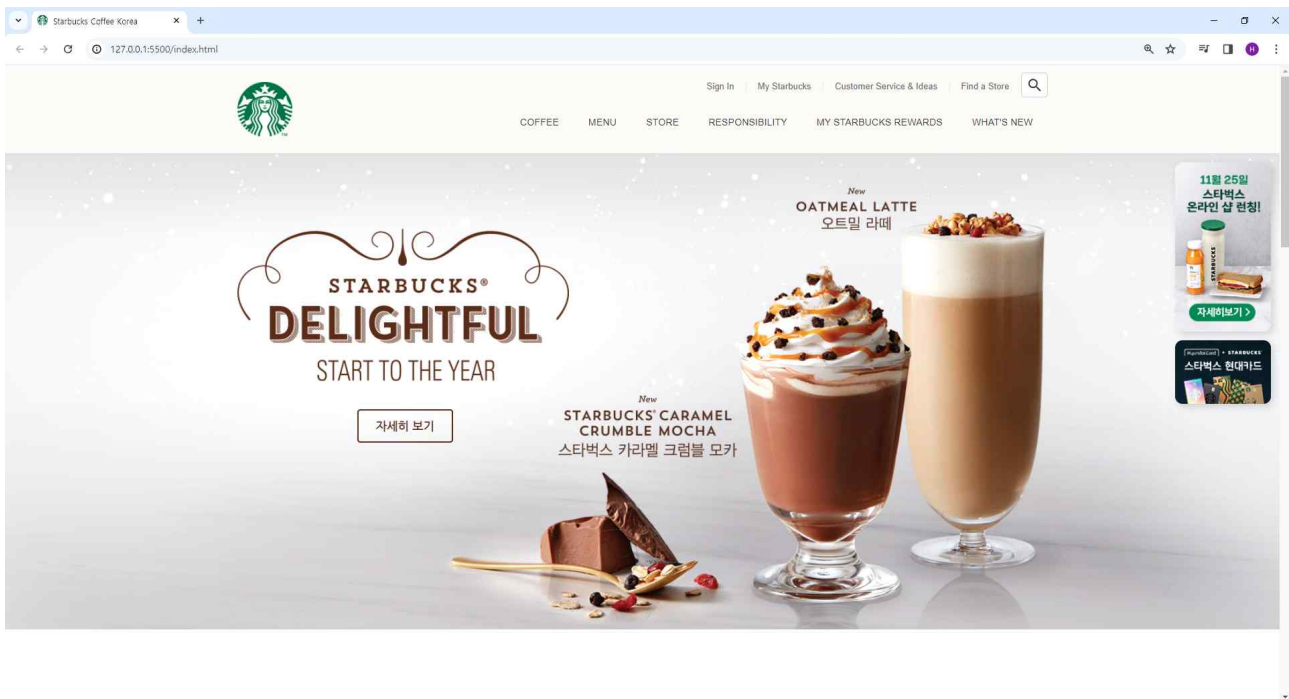
```
/* 타이틀이 이동되면 버튼도 따라 움직인다. */
```

```
position: absolute;
```

```
top: 259px;
```

```
left: 173px;
```

```
}
```



### 3. 순차적으로 요소 보이기

① 자바스크립트를 활용한 애니메이션 처리

- 페이지가 리로드(F5) 되면 타이틀 → 컵1 → 컵2 → 스푼 순서대로 나타나도록

- 자바 스크립트 활용

- 컵1과 텍스트1의 짝이므로 둘을 하나의 <div>로 묶어준다.
- class="fade-in" 추가
- css 에서 안보이게 처리 : opacity : 0
- 자바스크립트에서 한 개씩 나타나게 처리

▶ index.html

```
<!-- VISUAL -->
```

```
<section class="visual">
```

```
<div class="inner">
```

```
<div class="title fade-in">
```

```
<!-- alt : 대체텍스트는 이미지 안의 문자열로 ... -->
```

```

    <a href="javascript:void(0)" class="btn btn--brown">자세히 보기</a>
</div>

<div class="fade-in">
    
    
</div>

<div class="fade-in">
    
    
</div>

<div class="fade-in">
    
</div>
</div>
</section>

```

#### ▶ main.css

```

.visual .fade-in {
    opacity: 0;
}

```

#### ▶ main.js

```

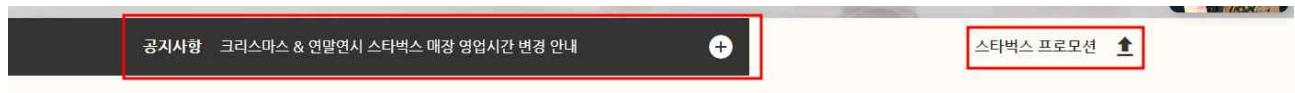
const fadeEls = document.querySelectorAll('.visual .fade-in');
fadeEls.forEach(function(fadeEl, index){
    gsap.to(fadeEl, 1, {
        delay : (index + 1) * 0.7, // 0.7초, 1.4초, 2.1초....
        opacity : 1,
    });
})

```



## Ch04. 요소 슬라이드

### 1. 공지사항



#### ① 슬라이드 공지사항 기초 만들기

- 일정 시간이 지나면 공지사항이 슬라이드 되면서 바뀐다.
- 우측 스타벅스 프로모션을 누르면 아래쪽 프로모션이 접어졌다 펴졌다 한다.
- section 태그가 끝난 후 위치부터 작업한다.

#### ▶ index.html

```
<section class="notice">
  <div class="notice-line">
    <div class="bg-left"></div>
    <div class="bg-right"></div>
    <!-- notice 가운데 정렬 용도 -->
    <div class="inner"></div>
  </div>
</section>
```

#### ▶ main.css

```
/* NOTICE */
.notice {}
.notice .notice-line {}
.notice .notice-line .bg-left {}
.notice .notice-line .bg-right {}
.notice .notice-line .inner{}
```

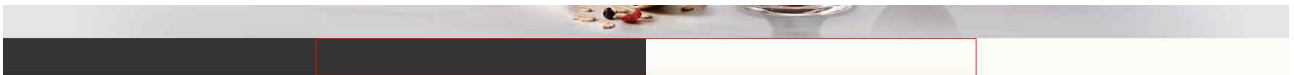
#### ▶ main.css

```
/* NOTICE */
.notice {}
.notice .notice-line {
  position: relative;
}
.notice .notice-line .bg-left {
  position: absolute;
  top: 0;
  left: 0;
  width: 50%;
```

```

height: 100%;
background-color: #333;
}
.notice .notice-line .bg-right {
position: absolute;
top: 0;
right: 0;
width: 50%;
height: 100%;
background-color: #f6f5ef;
}
.notice .notice-line .inner{
height: 62px;
border: 2px solid red;
}

```



② 슬라이드 inner 영역 좌우 분할하고 채우기

▶ index.html

```

<!-- NOTICE -->
<section class="notice">
  <div class="notice-line">
    <div class="bg-left"></div>
    <div class="bg-right"></div>
    <!-- notice 가운데 정렬 용도 -->
    <div class="inner">
      <div class="inner_left">
        <h2>공지사항</h2>
        <div class="swiper"></div>
        <a href="javascriptvoid(0)" class="notice-line__more">
          <div class="material-icons">add_circle</div>
        </a>
      </div>
      <div class="inner_right">
        <h2>스타벅스 프로모션</h2>
        <div class="toggle-promotion">
          <div class="material-icons">upload</div>
        </div>
      </div>
    </div>
  </div>
</section>

```

```
        </div>
    </div>
</div>
</section>
```

► main.css

```
/* NOTICE */
.notice {}
.notice .notice-line {
    position: relative;
}
.notice .notice-line .bg-left {
    position: absolute;
    top: 0;
    left: 0;
    width: 50%;
    height: 100%;
    background-color: #333;
}
.notice .notice-line .bg-right {
    position: absolute;
    top: 0;
    right: 0;
    width: 50%;
    height: 100%;
    background-color: #f6f5ef;
}
.notice .notice-line .inner{
    height: 62px;
    /* border: 2px solid red; */
    display: flex;
}

/* 공지사항 : 내용 + ---- 부분 */
.notice .notice-line .inner_left{
    width: 60%;
    height: 100%;
    background-color: #333;
    display: flex;
    /* center로 맞추면 swiper가 사라짐. 애도 높이 지정 */
}
```

```

    align-items: center;
}

.notice .notice-line .inner__left h2{
    color: #fff;
    font-size: 17px;
    font-weight: 700;
    margin-right: 20px;
}

.notice .notice-line .inner__left .swiper{
    background-color: orange;
    /* 애는 무조건 줄어들려는 습성이 있어서 최대한 늘어나라는 처리 필요 */
    flex-grow: 1;
    height: 62px;
}

.notice .notice-line .inner__left .notice-line__more {
    width: 62px;
    height: 62px;
    display: flex;
    justify-content: center;
    align-items: center;
}

.notice .notice-line .inner__left .notice-line__more .material-icons {
    /* 아이콘 색이 a 태그가 붙으면 기본 파랑인데 흰색으로 변경 */
    color: #fff;
    /* icon의 기본크기는 24px 이다. */
    font-size: 30px;
}

/* 스타벅스 프로모션 ↓ 부분 */
.notice .notice-line .inner__right{
    width: 40%;
    height: 100%;
    display: flex;
    justify-content: flex-end;
    align-items: center;
}

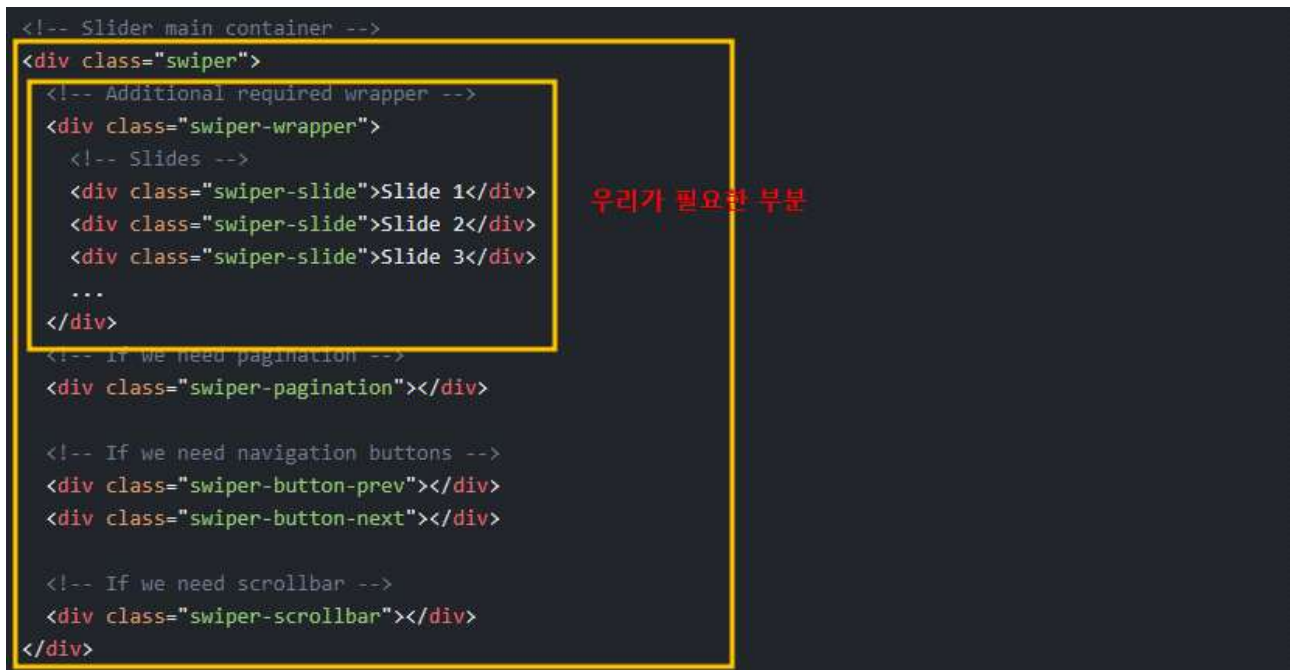
```

```
.notice .notice-line .inner_right h2 {
  font-size: 17px;
  font-weight: 700;
}
.notice .notice-line .inner_right .toggle-promotion{
  width: 62px;
  height: 62px;
  cursor: pointer;
  display: flex;
  align-items: center;
}
.notice .notice-line .inner_right .toggle-promotion .material-icons{
  font-size: 30px;
}
```

## 2. Swiper.js 라이브러리 확인 및 세로슬라이드 만들기

### ① swiper 셋팅하기

- <https://swiperjs.com>
- Get Started → 좌측 Add Swiper HTML Layout → 구조 확인하기



- Resources – Demos → 좌측 Vertical → 확인

### ② 사용하기

- Get Started → 좌측 Use Swiper from CDN → -----min----- → 복사 → index.html 붙여넣기
- ▶ index.html

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/swiper@11/swiper-bundle.min.css" />
<script src="https://cdn.jsdelivr.net/npm/swiper@11/swiper-bundle.min.js"></script>
```

```
<div class="inner_left">
  <h2>공지사항</h2>
  <!-- Swiper 구성부분 -->
  <div class="swiper">
    <div class="swiper-wrapper">
      <div class="swiper-slide">
        <a href="javascript:void(0)">
          크리스마스 & 연말연시 스타벅스 매장 영업시간 안내</a>
        </div>
      <div class="swiper-slide">
        <a href="javascript:void(0)">
          [당첨자 발표] 2024년 스타벅스 플래너 영수증 이벤트</a>
        </div>
      <div class="swiper-slide">
        <a href="javascript:void(0)">
          스타벅스 코리아 애플리케이션 버전 업데이트 안내</a>
        </div>
      <div class="swiper-slide">
        <a href="javascript:void(0)">
          [당첨자 발표] 뉴이어 전자영수증 이벤트</a>
        </div>
    </div>
  </div>
  <a href="javascript:void(0)" class="notice-line__more">
    <div class="material-icons">add_circle</div>
  </a>
</div>
```

### ③ js 및 css 구성하기

- <https://swiperjs.com>
- Get Started → 좌측 Initialize Swiper → copy

#### ▶ main.js

```
const swiper = new Swiper('.notice-line .swiper', {
  // Optional parameters
  direction: 'vertical',
```

```

    autoplay: true,
    loop: true,
  });

```

#### ▶ main.css

```

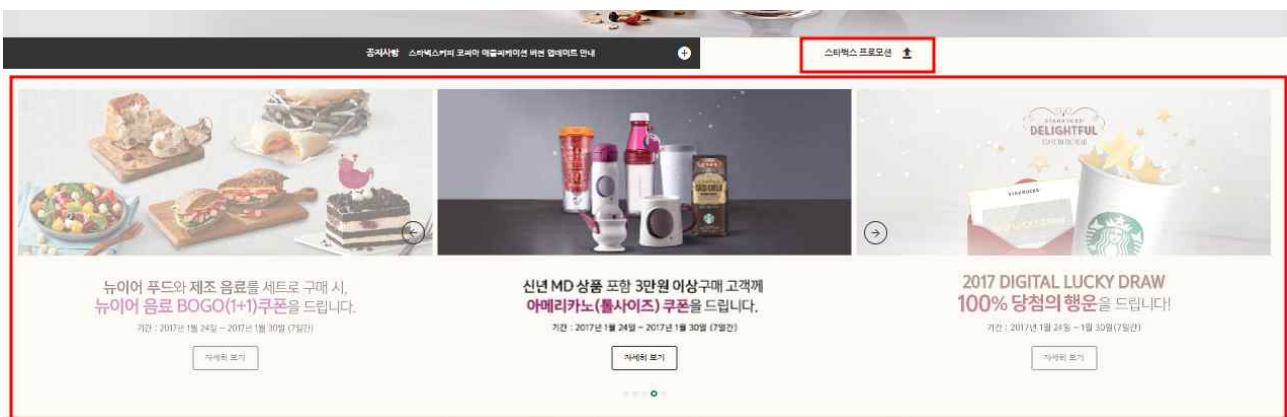
.notice .notice-line .inner_left .swiper{
  /* background-color: orange; */
  /* 애는 무조건 줄어들려는 습성이 있어서 최대한 늘어나라는 처리 필요 */
  flex-grow: 1;
  height: 62px;
}

.notice .notice-line .inner_left .swiper .swiper-slide{
  height: 62px;
  display: flex;
  align-items: center;
}

.notice .notice-line .inner_left .swiper .swiper-slide a{
  color: #fff;
}

```

### 3. 수평 슬라이드(Swiper)로 요소 가운데 배치



- 스타벅스 프로모션과 함께 묶여 있는 가로 슬라이드 만들기
- 화면을 확대해도 가운데 슬라이드가 계속 가운데에 위치하게 하기
- notice-line 형제 요소로 만들기
- 이미지 세 개 존재 : 이미지 사이즈 - 819pixel, 이미지 사이 간격 - 10pixel → 전체 2477pixel

#### ① 레이아웃 만들기

▶ index.html

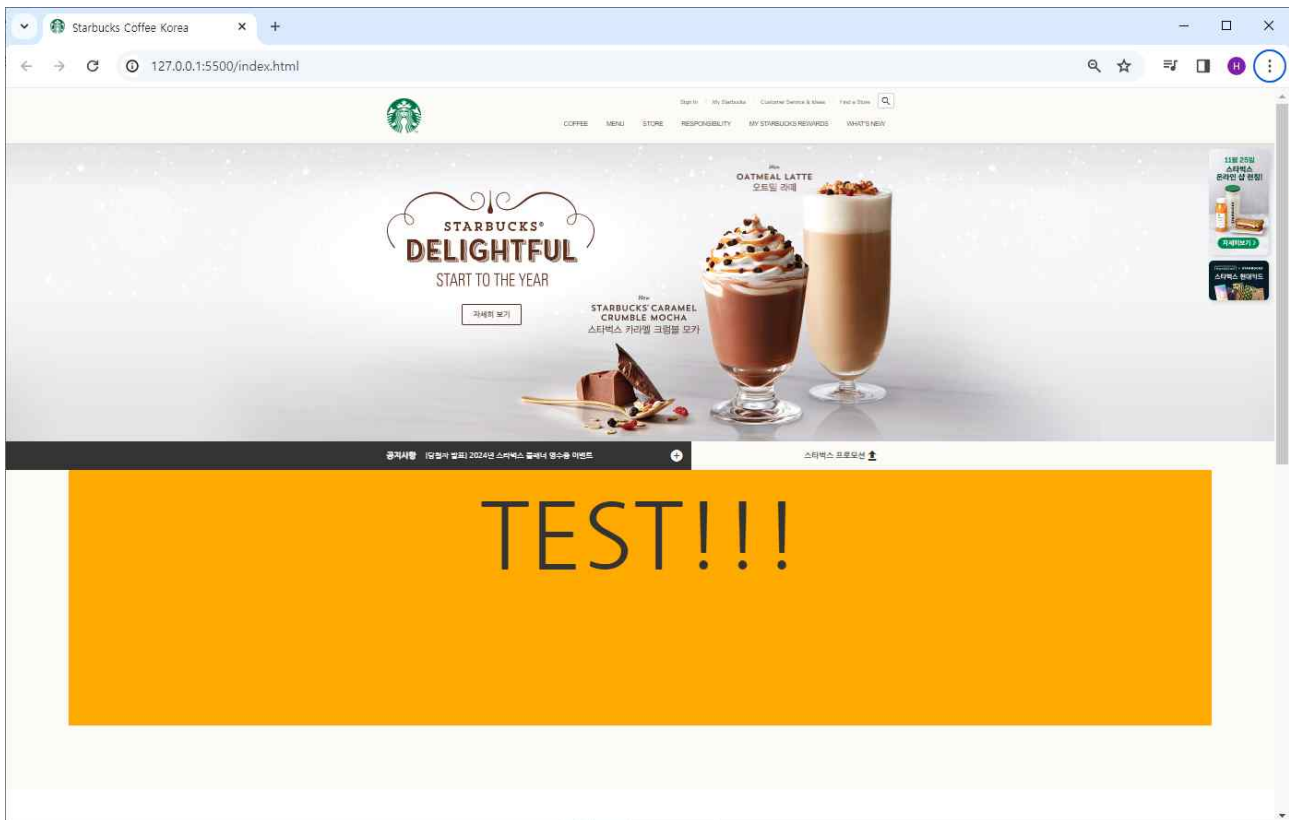
```
<!-- 프로모션 영역 -->
<div class="promotion">
  <div class="swiper">
    TEST!!
  </div>
</div>
```

▶ main.css

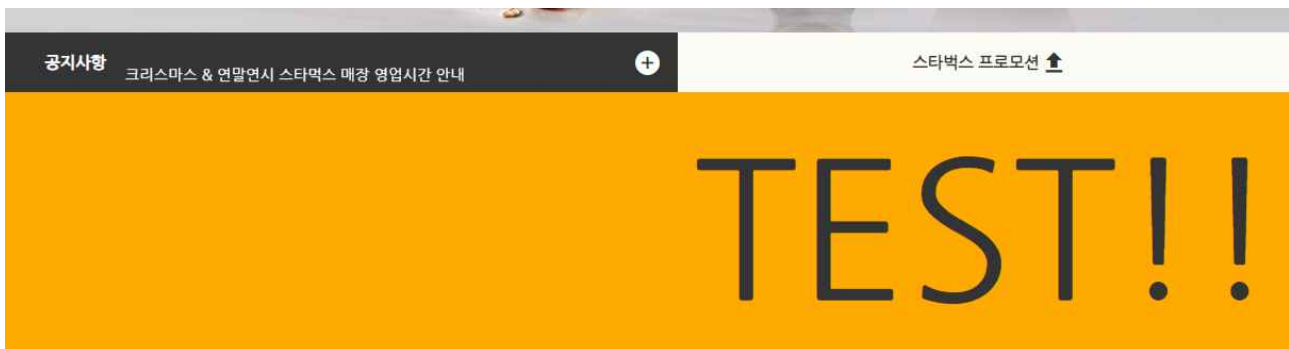
```
.notice .promotion {
  height: 693px;
  background-color: #f6f5ef;
}

.notice .promotion .swiper {
  /* 문제가 다소 많음 - 함수 사용해 처리 */
  /* width: 2477px; */
  width: calc(819px * 3 + 20px);
  /* width: calc(100% - 50px); */
  height: 553px;
  background-color: orange;
  text-align: center;
  font-size: 200px;
}
```





② 화면 확대 시 한쪽으로 몰리지 않고 정 중앙 유지하게 하기



▶ main.css

```
.notice .promotion {
  height: 693px;
  background-color: #f6f5ef;
  position: relative;
}

.notice .promotion .swiper {
  /* 문제가 다소 많음 - 함수 사용해 처리 */
}
```

```

/* width: 2477px; */
width: calc(819px * 3 + 20px);
/* width: calc(100% - 50px); */
height: 553px;
background-color: orange;
text-align: center;
font-size: 200px;

position: absolute;
top: 40px;
left: 50%;
/* margin-left: -1238.5px; */
margin-left: calc((819px * 3 + 20px) / -2);
}

```

#### 4. 프로모션 이미지 슬라이드

① 내용 넣고 동작 시키기

▶ index.html

```

<!-- 프로모션 영역 -->
<div class="promotion">
  <div class="swiper">
    <div class="swiper-wrapper">
      <div class="swiper-slide">
        
        <a href="javascript:void(0)" class="btn">자세히 보기</a>
      </div>
      <div class="swiper-slide">
        
        <a href="javascript:void(0)" class="btn">자세히 보기</a>
      </div>
      <div class="swiper-slide">
        
        <a href="javascript:void(0)" class="btn">자세히 보기</a>
      </div>
    </div>
  </div>
</div>

```

```

<div class="swiper-slide">
  
  <a href="javascripit:void(0)" class="btn">자세히 보기</a>
</div>
<div class="swiper-slide">
  
  <a href="javascripit:void(0)" class="btn">자세히 보기</a>
</div>
</div>
</div>
</div>

```

► main.css

```

.notice .promotion .swiper {
  width: calc(819px * 3 + 20px);
  height: 553px;

  /* background-color: orange;
  text-align: center;
  font-size: 200px; */ → 삭제

  position: absolute;
  top: 40px;
  left: 50%;
  margin-left: calc((819px * 3 + 20px) / -2);
}

```

► main.js

```

// const swiper = new Swiper('.notice-line .swiper', {
new Swiper('.notice-line .swiper', {
  // Optional parameters
  direction: 'vertical',
  autoplay: true,
  loop: true,
});

new Swiper('.promotion .swiper', {
  // Optional parameters

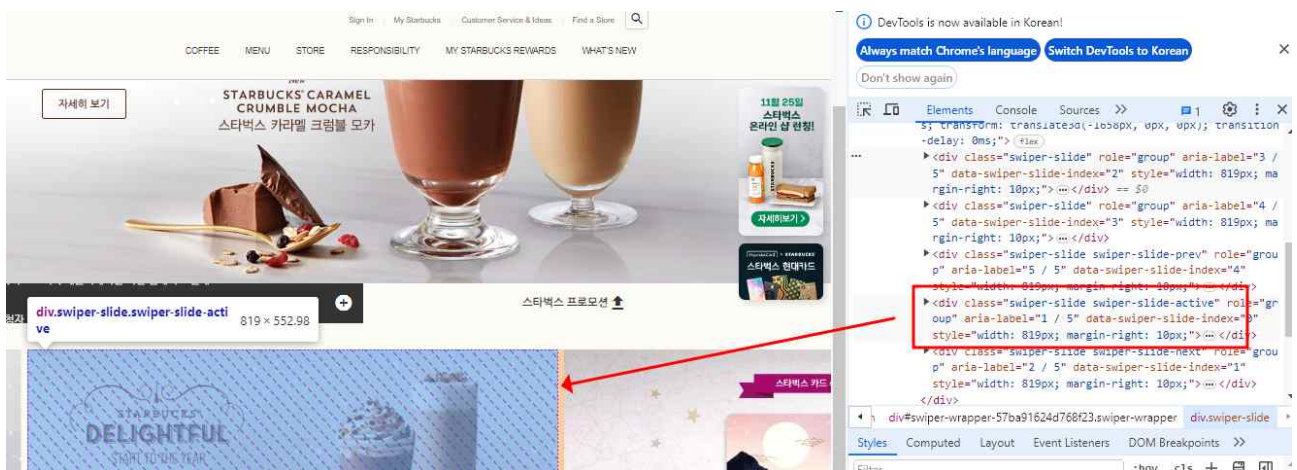
```

```

direction: 'horizontal',
slidesPerView : 3,
spaceBetween : 10,
centeredSlides : true,
loop: true,
autoplay: {
  delay : 5000
}
});

```

## ② 가운데 이미지만 진하게 주변은 blur 처리하기



- 개발자 도구에서 중앙이미지를 가리키는 동적 생성 class 이름 확인하기

### ▶ main.css

```

.notice .promotion .swiper-slide {
  opacity: 0.5;
  transition: opacity 1s;
}
.notice .promotion .swiper-slide-active {
  opacity: 1;
}

```

## ③ 버튼 처리하기

### ▶ main.css

```

.notice .promotion .swiper-slide {
  opacity: 0.5;
  transition: opacity 1s;
  position: relative;
}

```

```

}

.notice .promotion .swiper-slide .btn {
  position: absolute;
  bottom: 0;
  left: 0;
  right: 0;
  margin: auto;
}

```

### ③ 페이지 처리 버튼 구성

- swiper가 끝나는 부분 아래쪽에 기술

► index.html

```

<div class="swiper-pagination"></div>
<div class="swiper-prev"></div>
<div class="swiper-next"></div>

```

► main.js

```

new Swiper('.promotion .swiper', {
  // Optional parameters
  direction: 'horizontal',
  slidesPerView : 3,
  spaceBetween : 10,
  centeredSlides : true,
  loop: true,
  autoplay: {
    delay : 5000
  },
  //페이지 번호 요소 선택자 처리
  pagination: {
    el : '.promotion .swiper-pagination',
    clickable: true,
  },
  navigation: {
    prevEl: '.promotion .swiper-prev',
    nextEl: '.promotion .swiper-next',
  }
});

```

► main.css

```

.notice .promotion .swiper-pagination .swiper-pagination-bullet{
  background-color: transparent;
  background-image: url('../images/promotion_slide_pager.png');
  width: 12px;
  height: 12px;
  margin-right: 6px;
  outline: none;
}

.notice .promotion .swiper-pagination .swiper-pagination-bullet:last-child{
  margin-right: 0;
}

.notice .promotion .swiper-pagination .swiper-pagination-bullet-active{
  background-image: url('../images/promotion_slide_pager_on.png');
  width: 12px;
  height: 12px;
  margin-right: 6px;
  outline: none;
}

```

### ③ 좌우 슬라이드 이동 버튼

#### ▶ main.css

```

/* 다중선택자의 사용 */
.notice .promotion .swiper-prev,
.notice .promotion .swiper-next {
  width: 42px;
  height: 42px;
  border: 2px solid #333;
  border-radius: 50%;
  position: absolute;
  top: 300px;
  z-index: 1;
  cursor: pointer;
  outline: none;
}

.notice .promotion .swiper-prev {
  left: 50%;
  margin-left: -480px;
}

```

```

}

.notice .promotion .swiper-next {
  right: 50%;
  margin-right: -480px;
}

```

#### ▶ index.html

```

<div class="swiper-pagination"></div>
  <div class="swiper-prev">
    <div class="material-icons">arrow_back</div>
  </div>
  <div class="swiper-next">
    <div class="material-icons">arrow_forward</div>
  </div>

```

#### ▶ main.css : 화살표 위치 및 정리

```

/* 다중선택자의 사용 */
.notice .promotion .swiper-prev,
.notice .promotion .swiper-next {
  width: 42px;
  height: 42px;
  border: 2px solid #333;
  border-radius: 50%;
  position: absolute;
  top: 300px;
  z-index: 1;
  cursor: pointer;
  outline: none;
  display: flex;
  justify-content: center;
  align-items: center;
  transition: 0.4s
}

.notice .promotion .swiper-prev:hover,
.notice .promotion .swiper-next:hover {
  background-color: #333;
  color: #fff;
}

```

## 5. 슬라이드 영역 토글

### ① 토글시킴

- 슬라이드 우측 → 스타벅스 프로모션 ↑ → 클릭 시 토글
- 프로모션 클래스 내부에 토글 버튼 클릭 시 hide 클래스 추가
- 프로모션 클래스 내부에 토글 버튼 클릭 시 hide 클래스 제거

▶ main.js

```
const promotioinEl = document.querySelector('.promotion');
const promotionToggleBtn = document.querySelector('.toggle-promotion');
let isHidePromotion = false;

promotionToggleBtn.addEventListener('click', function(){
  isHidePromotion = ! isHidePromotion;
  if(isHidePromotion){
    // 숨김처리
    // promotionEl 내 모든 클래스리스트에 hide 클래스를 추가
    promotioinEl.classList.add('hide');
  } else {
    // 다시 보임 처리
    promotioinEl.classList.remove('hide')
  }
});
```

▶ F12 개발자 모드에서 hide 생성 및 제거 확인

▶ main.css

```
.notice .promotion {
  height: 693px;
  background-color: #f6f5ef;
  position: relative;
  /* 높이를 0.4초에 걸쳐서 줄임 */
  transition: height 0.4s;
  /* 크기가 줄면서 안의 내용이 넘치는 잘라버림 */
  overflow: hidden;
}

/* .promotion.hide 띄어쓰기 금지 */
.notice .promotion.hide {
  height: 0;
}
```



## Ch05. 유튜브 영상 배경

### 1. 리워즈 작업하기



#### ① 기본 틀 만들기

##### ▶ index.html

```
<!-- REWARDS -->
<section class="rewards">
  <div class="inner"></div>
</section>
```

##### ▶ main.css

```
/* REWARDS */
.rewards {}
.rewards .inner{
  background-image: url('../images/rewards.jpg');
  height: 241px;
}
```

#### ② 좌우 여백 없애기(늘이기)

##### ▶ index.html

```
<!-- REWARDS -->
<section class="rewards">
  <!-- 좌우측 여백 작업하기 -->
  <div class="bg-left"></div>
  <div class="bg-right"></div>
  <div class="inner"></div>
</section>
```

##### ▶ main.css

```

/* REWARDS */
.rewards {
  position: relative;
}

.rewards .bg-left{
  width: 50%;
  height: 100%;
  background-color: #272727;
  position: absolute;
  top: 0;
  left: 0;
}

.rewards .bg-right {
  width: 50%;
  height: 100%;
  background-color: #d5c798;
  position: absolute;
  top: 0;
  right: 0;
}

```



### ③ 우측 버튼 그룹 작업하기

▶ index.html

```

<!-- REWARDS -->
<section class="rewards">
  <!-- 좌우측 여백 작업하기 -->
  <div class="bg-left"> </div>
  <div class="bg-right"> </div>
  <div class="inner">

```

```

<div class="btn-group">
  <div class="btn btn--reverse sign-up">회원가입</div>
  <div class="btn sign-in">로그인</div>
  <div class="btn gift">e-Gift 선물하기</div>
</div>
</div>
</section>

```

► main.css

```

.rewards .btn-group{
  /* 수평맞춤하고 길이 250px이 넘으면
  줄바꿈 하겠다.
  기본 버튼 길이가 130px 이므로 한곳에 두개가 들어갈 수 없다.
  개별 버튼 길이 조정 필요
  */
  position: absolute;
  bottom: 24px;
  right: 0;
  width: 250px;
  display: flex;
  flex-wrap: wrap;
}
.rewards .btn-group .btn.sign-up{
  margin-right: 10px;
}
.rewards .btn-group .btn.sign-in{
  width: 110px;
}
.rewards .btn-group .btn.gift{
  /* flex-grow : 1 → btn그룹의 너비인 250px까지 늘려라 */
  margin-top: 10px;
  flex-grow: 1;
}

```

## 2. Youtube 재생 위치 작업하기

### ① 영화프레임 만들기

#### ▶ screen.html

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="/css/screen.css">
  <title>Document</title>
</head>
<body>
  <div class="container">
    <div class="item"></div>
  </div>
</body>
</html>
```

#### ▶ screen.css

```
/* height는 한없이 작아지려는 성향이 있다.
   그런데, padding-top을 비율로 지정하면,
   높이는 상위컨테이너의 너비의 비율만큼 생긴다.
   영화화면의 비율은 16:9 이는 56.25% 이다.
*/

.container{
  width: 500px;
  background-color: royalblue;
}
.container .item{
  width: 100%;
  height: 0;
  padding-top: 50%;
}
```

### ② Youtube 영상 삽입할 기본 프레임 만들기

#### ▶ index.html

```
<!-- YOUTUBE VIDEO -->
<section class="youtube">
```

```
<div class="youtube_area">
  <div id="player"> </div>
</div>
<div class="youtube_cover"> </div>
</section>
```

► main.css

```
/* YOUTUBE VIDEO */
.youtube {
  position: relative;
  height: 700px;
  background-color: #333;
  /* .youtube 범위를 넘치는 동영상은 모두 잘라버림 */
  overflow: hidden;
}

.youtube .youtube_area {
  width: 1920px;
  background-color: orange;
  position: absolute;
  left: 50%;
  margin-left: calc(1920px / -2);
  /* .youtube 영역 정가운데 .youtube_area 위치시키기 */
  top: 50%;
  margin-top: calc(1920px * 9 / 16 / -2);
}

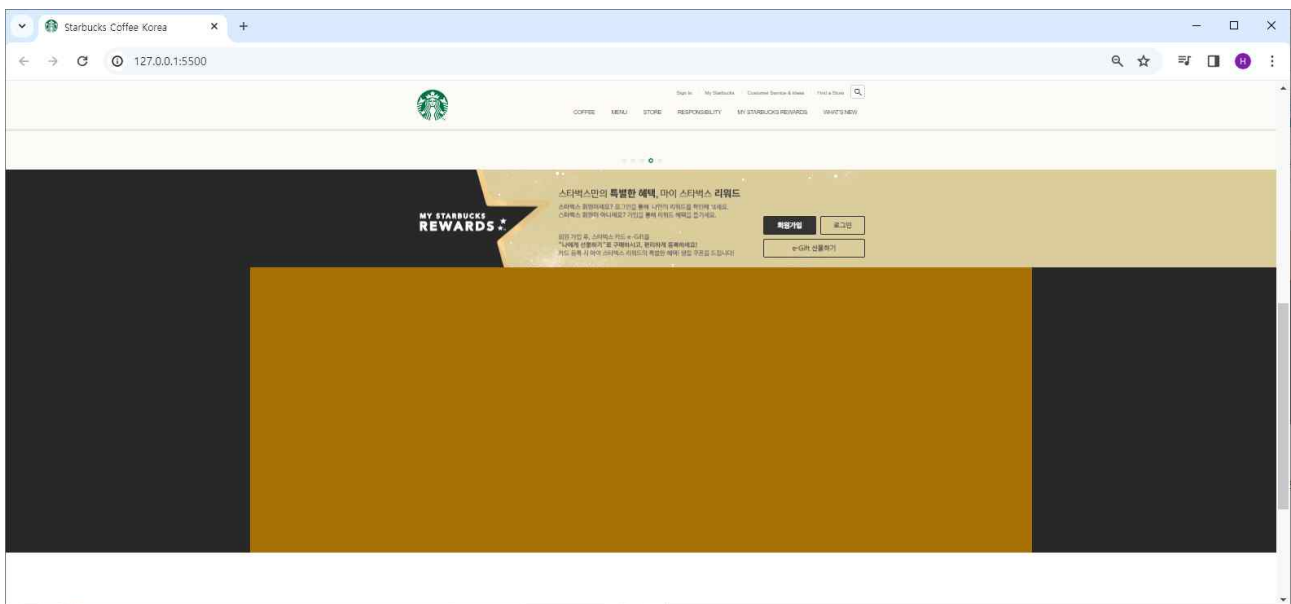
/* 실제로 요소를 HTML에 추가하지 않고 가상요소 선택자를
이용해 처리 함. screen.html의 container -> youtube_area
item에 해당하는 부분을 가상요소로 대체
*/
.youtube .youtube_area::before {
  content: "";
  display: block;
  width: 100%;
  height: 0;
  padding-top: 56.25%
}

.youtube .youtube_cover{
  background-image: url(../images/video_cover_pattern.png);
```

```

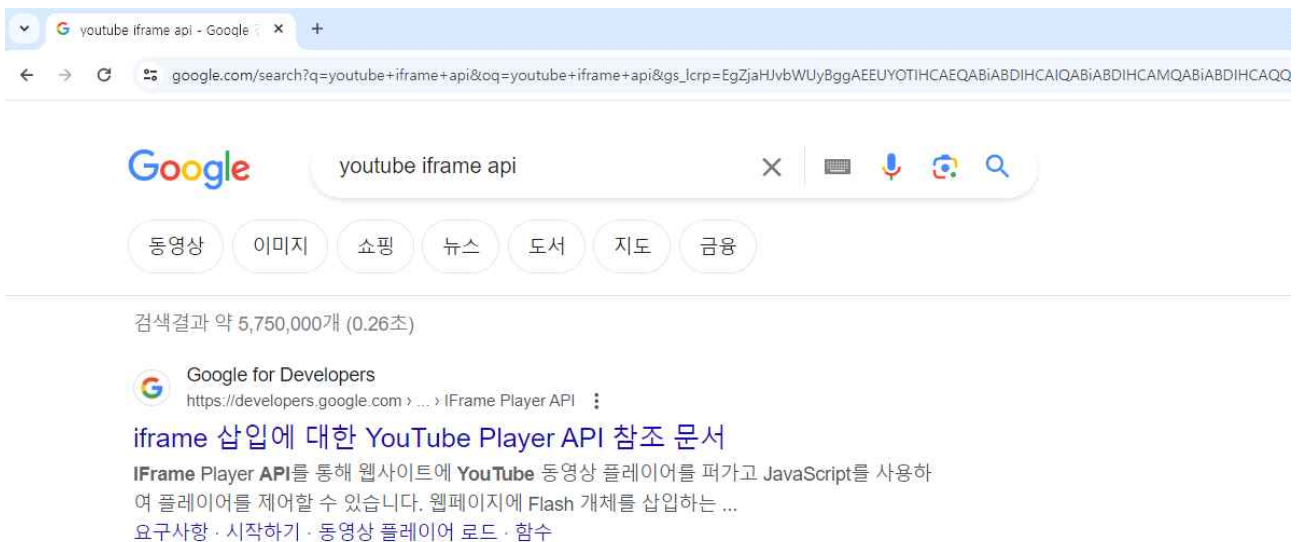
background-color: rgba(0, 0, 0, 0.3);
position: absolute;
top: 0;
left: 0;
width: 100%;
height: 100%;
}

```



### 3. Youtube iFrame API를 활용한 영상재생

#### ① youtube iframe api 설정



- 시작하기 → 아래쪽 코드 중 script 부분만 복사하기
- js 폴더 내 → youtube.js 파일 생성 후 붙이기

▶ youtube.js → 변경 전

```
var tag = document.createElement('script');

tag.src = "https://www.youtube.com/iframe_api";
var firstScriptTag = document.getElementsByTagName('script')[0];
firstScriptTag.parentNode.insertBefore(tag, firstScriptTag);

var player;
function onYouTubeIframeAPIReady() {
  player = new YT.Player('player', {
    height: '360',
    width: '640',
    videoid: 'M7lc1UVf-VE',
    events: {
      'onReady': onPlayerReady,
      'onStateChange': onPlayerStateChange
    }
  });
}
```

▶ youtube.js → 변경 후

```
var tag = document.createElement('script');

tag.src = "https://www.youtube.com/iframe_api";
var firstScriptTag = document.getElementsByTagName('script')[0];
firstScriptTag.parentNode.insertBefore(tag, firstScriptTag);

function onYouTubeIframeAPIReady() {
  new YT.Player('player', {
    videoid: 'EJF919p_hb0',
    playerVars: {
      autoplay: true,
      loop: true,
      playlist: 'EJF919p_hb0',
    },
    events: {
      onReady: function(event){
```

```

        event.target.mute()
    }
}
});
}

```

▶ index.html

```
<script defer src="./js/youtube.js"></script>
```

▶ main.css → id : player 인 css 설정

```

#player {
    width: 100%;
    height: 100%;
    position: absolute;
    top: 0;
    left: 0;
}

```

## 4. 애니메이션 만들기

① youtube 위에서 둥둥 떠다니는 애니메이션 만들기

▶ index.html

```

<!-- YOUTUBE VIDEO -->
<section class="youtube">
    <div class="youtube_area">
        <div id="player"></div>
    </div>
    <div class="youtube_cover"></div>
    <div class="inner">
        
        
        
    </div>
</section>

```

▶ main.css

```

.youtube .inner {
    height: 700px;
    /* background-color: royalblue; */
}

```



```

}

.youtube .floating1 {
  position: absolute;
  top: 50px;
  left: 0;
}

.youtube .floating2 {
  position: absolute;
  top: 350px;
  left: 150px;
}

.youtube .floating3 {
  position: absolute;
  bottom: -200px;
  right: 0;
}

```

► main.js

```

// 범위 랜덤 함수(소수점 2자리까지)
function random(min, max) {
  // `.toFixed()`를 통해 반환된 문자 데이터를,
  // `parseFloat()`을 통해 소수점을 가지는 숫자 데이터로 변환
  return parseFloat((Math.random() * (max - min) + min).toFixed(2))
}

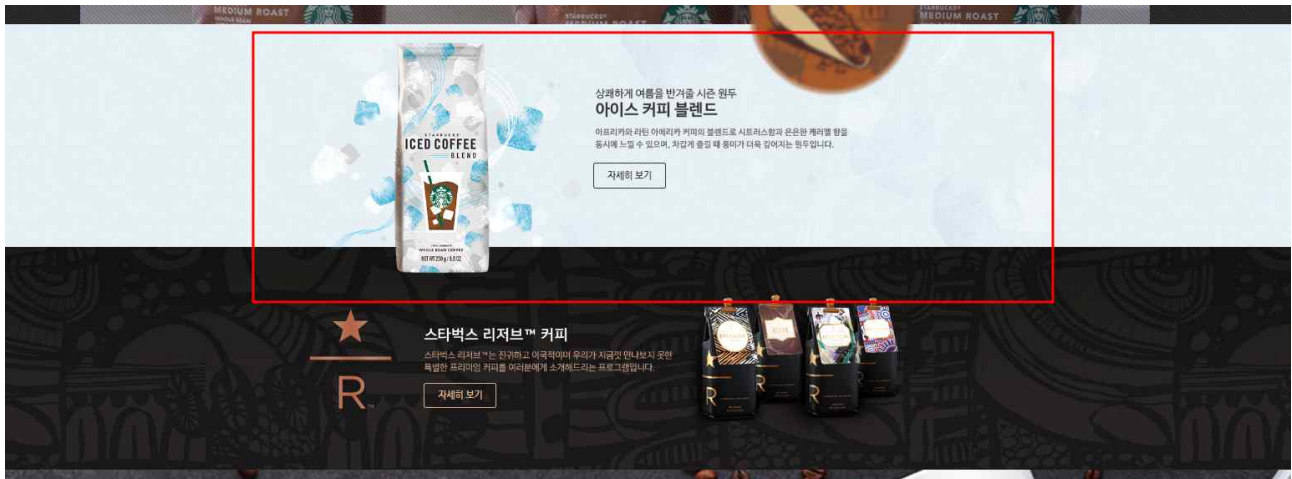
function floatingObject(selector, delay, size){
  // gsap.to(요소, 시간, 옵션);
  gsap.to(selector, random(1.5, 2.5), {
    y: size,
    repeat: -1,
    yoyo: true,
    ease: Power1.easeInOut,
    delay: random(0, delay)
  })
}

floatingObject('.floating1', 1, 15);

```

```
floatingObject('.floating2', 0.5, 15);
floatingObject('.floating3', 1.5, 20);
```

## 5. 고정이미지 배경



### ① 이미지 넣고 작업하기

#### ▶ index.html

```
<!-- SEASON PRODUCT -->
<section class="season-product">
  <div class="inner">
    
    <div class="text-group">
      
      
      <div class="more">
        <a href="javascript:void(0)" class="btn">자세히 보기</a>
      </div>
    </div>
  </div>
</section>
```

#### ▶ main.css

```
body {
  height: 6000px;
}

/* SEASON PRODUCT */
.season-product {
```

```

    background-image: url('../images/season_product_bg.jpg');
}
.season-product .inner {
    height: 400px;
}
.season-product .text-group {
    position: absolute;
    top: 110px;
    right: 100px;
}
.season-product .text-group .title {
    margin-bottom: 10px;
}
.season-product .text-group .description {
    margin-bottom: 15px;
}
.season-product .text-group .more {}

```

## ② youtube 섹션에서 잘린 원형이미지 복원하기



※ overflow : hidden 속성때문에 안 보였던 이미지를 옮겨 준다.

```



 ---> 삭제
</div>
</section>

<!-- SEASON PRODUCT -->
<section class="season-product">
    <div class="inner">
        
        
    </div>
</section>

```

```

<div class="text-group">
  
  
  <div class="more">
    <a href="javascript:void(0)" class="btn">자세히 보기</a>
  </div>
</div>

```

#### ▶ main.css 변경

```

/* .youtube .floating3 { -----> 삭제 후 이동
position: absolute;
bottom: -200px;
right: 0;
} */

/* SEASON PRODUCT */
.season-product {
  background-image: url("../images/season_product_bg.jpg");
}

.season-product .inner {
  height: 400px;
}

.season-product .floating3 {
  position: absolute;
  top: -200px;
  right: 0;
}

```

#### ③ 리저브커피 섹션 작업하기



#### ▶ index.html

```

<section class="reserve-coffee">

```

```

<div class="inner">
  
  <div class="text-group">
    
    <div class="more">
      <a href="javascript:void(0)" class="btn btn--gold">자세히 보기</a>
    </div>
  </div>

  
</div>
</section>

```

► main.css

```

/* RESERVE COFFEE */
.reserve-coffee {
  background-image: url('/images/reserve_bg.jpg');
}
.reserve-coffee .inner {
  height: 400px;
}
.reserve-coffee .reserve-logo {
  position: absolute;
  top: 110px;
  left: 0;
}
.reserve-coffee .text-group {
  position: absolute;
  top: 124px;
  left: 208px;
}
.reserve-coffee .product {
  position: absolute;
  top: 0;
  right: 0;
}

```

④ PICK YOUR FAVORITE SECTION 작업하기



## ► index.html

```
<!-- YOUR FAVORITE -->
<section class="pick-your-favorite">
  <div class="inner">
    <div class="text-group">
      
      
      <div class="more">
        <a href="javascript:void(0)" class="btn btn--white">자세히 보기</a>
      </div>
    </div>

    
  </div>
</section>
```

## ► main.css

```
/* PICK YOUR FAVORITE */
.pick-your-favorite {
  background-image: url('../images/favorite_bg.jpg');
  background-repeat: no-repeat;
  background-position: center;
  /* 이미지가 뷰포트에 고정(마우스 스크롤할 때)
  즉, 화면따라 움직이지 않음.
  시간차 애니메이션 기법
  */
  background-attachment: fixed;

  /* 배경의 이미지를 요소의 더 넓은 너비에 맞춰서 출력 */
  background-size: cover;
}
```

```

.pick-your-favorite .inner {
  padding: 110px 0;
}
.pick-your-favorite .text-group {
  /* background-color: orange; */
  width: 362px;
  display: flex;
  /* 주어진 영역에서 줄바꿈 */
  flex-wrap: wrap;
  /* 우측정렬 */
  justify-content: flex-end;
}
.pick-your-favorite .text-group .title {
  margin-bottom: 40px;
}
.pick-your-favorite .text-group .description {
  margin-bottom: 40px;
}

```

##### ⑤ reserve store 작업하기



##### ▶ index.html

```

<!-- RESERVE STORE -->
<section class="reserve-store">
  <div class="inner"></div>
</section>

```

##### ▶ main.css

```
/* RESERVE STORE */
.reserve-store {
  background-image: url(../images/reserve_store_bg.jpg);
  background-repeat: no-repeat;
  background-position: center;
  background-attachment: fixed;
  background-size: cover;
}
.reserve-store .inner {
  height: 600px;
}
```



## 6. 3D 애니메이션



### ① 애니메이션 만들기

- Reserve Store 내부 이미지위에 마우스를 올리면
- 회전하면서 매장안내가 나타나는 효과 구성

#### ▶ index.html

```
<!-- RESERVE STORE -->
<section class="reserve-store">
  <div class="inner">
    <div class="medal">
      <div class="front">
        
      </div>
      <div class="back">
        
      </div>
    </div>
  </div>
</section>
```

#### ▶ main.css

```
.reserve-store .inner {
  height: 600px;
  display: flex;
  justify-content: center;
  align-items: center;
}

.reserve-store .medal{
  width: 334px;
  height: 334px;
  /* 원근거리 600px */
  perspective: 600px;
}

.reserve-store .medal .front,
.reserve-store .medal .back {
  width: 334px;
  height: 334px;
  backface-visibility: hidden;
  transition: 1s;
}

.reserve-store .medal .front {
  position: absolute;
  transform: rotateY(0deg);
}

.reserve-store .medal:hover .front {
  transform: rotateY(-180deg);
}

.reserve-store .medal .back {
  transform: rotateY(-180deg);
}

.reserve-store .medal:hover .back {
  transform: rotateY(0deg);
}
```

## ② 버튼추가하기

▶ index.html

```

<!-- RESERVE STORE -->
<section class="reserve-store">
  <div class="inner">
    <div class="medal">
      <div class="front">
        
      </div>
      <div class="back">
        
        <a href="javascriptvoid()" class="btn">
          매장안내 </a>
        </div>
      </div>
    </div>
  </div>
</section>

```

#### ▶ main.css

```

.reserve-store .medal .back .btn {
  position: absolute;
  top: 240px;
  left: 0;
  right: 0;
  margin: auto;
}

```

### ③ 매장찾기 section 만들기



#### ▶ index.html

```

<!-- FIND THE STORE -->
<section class="find-store">
  <div class="inner">

```

```

<imgsrc="./images/find_store_texture1.png"alt=""class="texture1"/>
<imgsrc="./images/find_store_texture2.png"alt=""class="texture2"/>>
<imgsrc="./images/find_store_picture1.jpg"alt=""class="picture picture1">
<imgsrc="./images/find_store_picture2.jpg"alt=""class="picture picture2">>

<divclass="text-group">
  <imgsrc="./images/find_store_text1.png"alt=""class="title">
  <imgsrc="./images/find_store_text2.png"alt=""class="description">
  <divclass="more"></div>
  <ahref="javascript:void(0)"class="btn">매장찾기</a>
</div>
</div>
</section>

```

#### ► main.css

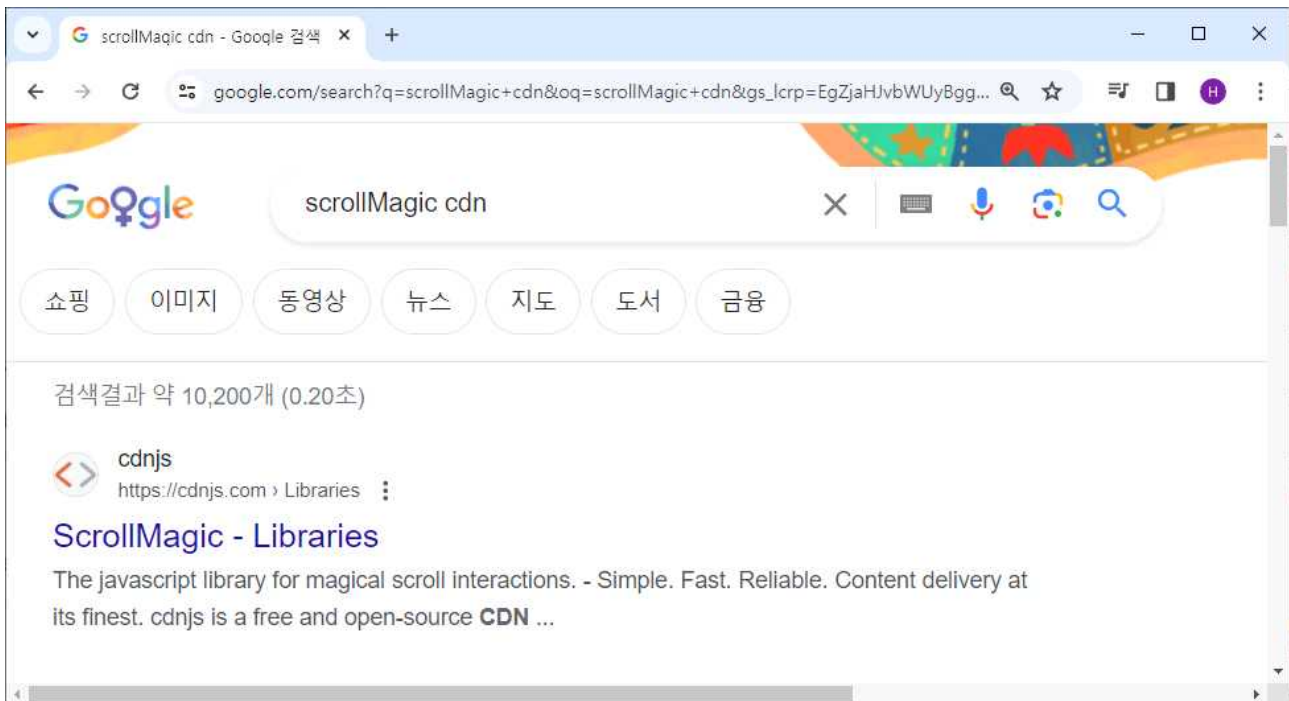
```

/* FIND STORE */
.find-store{
  background-image: url("../images/find_store_bg.jpg")
}
.find-store.inner{
  height: 400px;
}
.find-store.texture1{
  position: absolute;
  top: 0;
  left: 400px;
}
.find-store.texture2{
  position: absolute;
  bottom: 0;
  right: 0;
}
.find-store.picture{
  position: absolute;
  border-radius: 50%;
  box-shadow: 2px2px8pxrgba(0, 0, 0, 0.5);
}
.find-store.picture1{
  /* position: absolute; */
  top: -60px;
}

```



```
    left: 0;
  }
  .find-store.picture2{
    /* position: absolute; */
    top: 150px;
    left: 250px;
  }
  .find-store.text-group{
    position: absolute;
    top: 120px;
    left: 550px;
  }
  .find-store.text-group.title{
    margin-bottom: 20px
  }
  .find-store.text-group.description{
    margin-bottom: 20px;
  };
};
```

## 7. 스크롤 위치 계산 애니메이션



# ScrollMagic

The javascript library for magical scroll interactions.

★ 14k  GitHub  package 0 known vulnerabilities

(MIT OR GPL-3.0+) licensed <http://janpaepke.github.io/ScrollMagic/>

Tags: scroll, scrolling, animation, sticky, pin, fixed, scrollbar, scrub, sync, position, progress, parallax, events, classes

Version  Asset Type

<https://cdnjs.cloudflare.com/ajax/libs/ScrollMagic/2.0.8/ScrollMagic.min.js>



## ① scroll magic 삽입하기

### ▶ index.html

```
<!-- ScrollMagic -->
<scriptsrc="https://cdnjs.cloudflare.com/ajax/libs/ScrollMagic/2.0.8/ScrollMagic.min.js"integrity="sha512-8E3KZoPoZCD+1dgfqhPbejQBnQfBXe8FuWL4z/c8sTrgeDMFEnoyTIH3obB4/fV+6Sg0a0XF+L/6xS4Xx1fUEg=="crossorigin="anonymous"referrerpolicy="no-referrer"></script>
```

### ▶ main.js

```
/**
 * 요소가 화면에 보여짐 여부에 따른 요소 관리
 */
// 관리할 요소들 검색!
const spyEls = document.querySelectorAll('section.scroll-spy')
// 요소들 반복 처리!
spyEls.forEach(function (spyEl) {
  new ScrollMagic
    .Scene({ // 감시할 장면(Scene)을 추가
      triggerElement: spyEl, // 보여짐 여부를 감시할 요소를 지정
      triggerHook: .8 // 화면의 80% 지점에서 보여짐 여부 감시
    })
    .setClassToggle(spyEl, 'show') // 요소가 화면에 보이면 show 클래스 추가
    .addTo(new ScrollMagic.Controller()) // 컨트롤러에 장면을 할당(필수!)
})
```

### ▶ index.html

```
<!-- SEASON PRODUCT -->
<sectionclass="season-product scroll-spy">
```

```

<!-- RESERVE-COFFEE -->
<sectionclass="reserve-coffee scroll-spy">

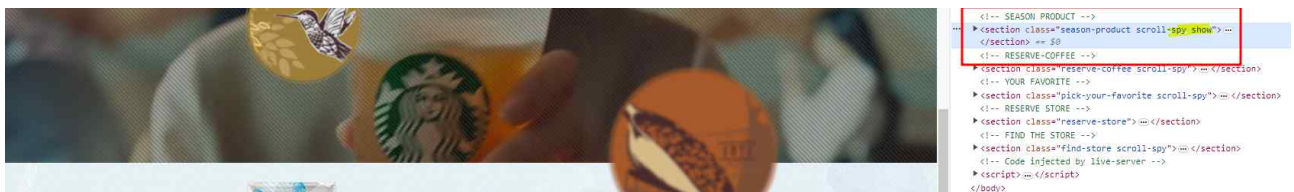
<!-- YOUR FAVORITE -->
<sectionclass="pick-your-favorite scroll-spy">

<!-- RESERVE STORE -->   애는 건너 뛴

<!-- FIND THE STORE -->
<sectionclass="find-store scroll-spy">

```

※ 개발자도구(F12)를 열어 확인. ScrollMagic이라는 자바스크립트 라이브러리가 각각의 해당하는 Scrollspy라는 클래스를 가지고 있는 요소들을 지속적으로 감시하고 있다가 그 요소가 우리가 지정해 놓은 해당하는 지점을 넘어서 화면에 보이는 순간 각각의 부분에 show 라는 클래스를 추가해 준다.



▶ main.css : 헤더 바로 위 부분에 기술

```

/*BACK TO POSITION*/
.back-to-position {
  opacity: 0;
  transition: 1s;
}
.back-to-position.to-right {
  transform: translateX(-150px);
}
.back-to-position.to-left {
  transform: translateX(150px);
}
.show .back-to-position {
  opacity: 1;
  transform: translateX(0);
}

```

```

}
.show .back-to-position.delay-0 {
  transition-delay: 0s;
}
.show .back-to-position.delay-1 {
  transition-delay: .3s;
}
.show .back-to-position.delay-2 {
  transition-delay: .6s;
}
.show .back-to-position.delay-3 {
  transition-delay: .9s;
}

```

#### ► index.html

```

<!-- SEASON PRODUCT -->
<sectionclass="season-product scroll-spy">
  <divclass="inner">
    <imgsrc="/images/floating3.png"alt="Icon"class="floating floating3">
    <imgsrc="/images/season_product_image.png"alt=""
      class="product back-to-position to-right delay-0"/>
    <divclass="text-group">
      <imgsrc="/images/season_product_text1.png"alt=""
        class="title back-to-position to-left delay-1"/>
      <imgsrc="/images/season_product_text2.png"alt=""
        class="description back-to-position to-left delay-2"/>
      <divclass="more back-to-position to-left delay-3">

<!-- RESERVE-COFFEE -->
<sectionclass="reserve-coffee scroll-spy">
  <divclass="inner">
    <imgsrc="/images/reserve_logo.png"alt=""
      class="reserve-logo back-to-position to-right delay-0"/>
    <divclass="text-group">
      <imgsrc="/images/reserve_text.png"alt=""
        class="description back-to-position to-right delay-1"/>
      <divclass="more back-to-position to-right delay-2">
        <a href="javascript:void(0)"class="btn btn--gold">자세히 보기</a>
      </div>

```



```

</div>

<imgsrc="/images/reserve_image.png"alt=""
        class="product back-to-position to-left delay-3"/>

<!-- YOUR FAVORITE -->
<sectionclass="pick-your-favorite scroll-spy">
<divclass="inner">
<divclass="text-group">
<imgsrc="/images/favorite_text1.png"alt=""
        class="title back-to-position to-right delay-0"/>
<imgsrc="/images/favorite_text2.png"alt=""
        class="description back-to-position to-right delay-1"/>
<divclass="more back-to-position to-right delay-2">
<a href="javascript:void(0)"class="btn btn--white">자세히 보기</a>
</div>
</div>

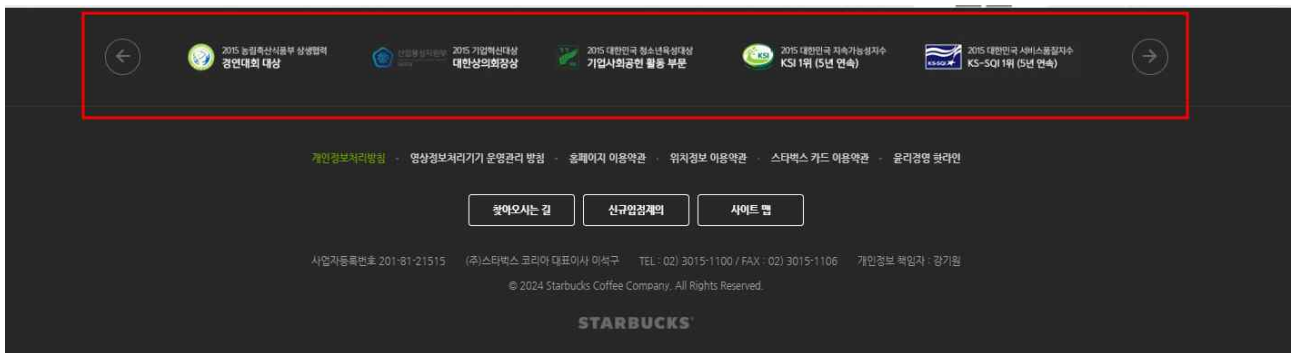
<imgsrc="/images/reserve_image.png"alt=""class="product back-to-position to-left">

<!-- FIND THE STORE -->
<sectionclass="find-store scroll-spy">
<divclass="inner">
<imgsrc="/images/find_store_texture1.png"alt=""class="texture1"/>
<imgsrc="/images/find_store_texture2.png"alt=""class="texture2"/>
<imgsrc="/images/find_store_picture1.jpg"alt=""class="picture picture1
        back-to-position to-right delay-0"/>
<imgsrc="/images/find_store_picture2.jpg"alt=""class="picture picture2
        back-to-position to-right delay-1"/>

<divclass="text-group">
<imgsrc="/images/find_store_text1.png"alt=""class="title back-to-position to-left delay-0"/>
<imgsrc="/images/find_store_text2.png"alt=""
        class="description back-to-position to-left delay-1"/>
<divclass="more back-to-position to-left delay-2"></div>

```

## 8. 다중 요소 슬라이드



## ① Swiper 사용해서 처리하기

### ▶ index.html

```
<!--AWARDS-->
<section class="awards">
  <div class="inner">
    <div class="swiper">
      <div class="swiper-wrapper">
        <div class="swiper-slide">
          
        </div>
        <div class="swiper-slide">
          
        </div>
        <div class="swiper-slide">
          
        </div>
        <div class="swiper-slide">
          
        </div>
        <div class="swiper-slide">
          
        </div>
        <div class="swiper-slide">
          
        </div>
        <div class="swiper-slide">
          
        </div>
        <div class="swiper-slide">
          
        </div>
      </div>
    </div>
  </div>
</section>
```

```

<div class="swiper-slide">
  
</div>
<div class="swiper-slide">
  
</div>
</div>
</div>

<div class="swiper-prev">
  <span class="material-icons">arrow_back</span>
</div>
<div class="swiper-next">
  <span class="material-icons">arrow_forward</span>
</div>
</div>
</section>

```

#### ► main.js

```

new Swiper('.awards .swiper', {
  direction: 'horizontal', // 수평 슬라이드
  autoplay: true, // 자동 재생 여부
  loop: true, // 반복 재생 여부
  spaceBetween: 30, // 슬라이드 사이 여백
  slidesPerView: 5, // 한 번에 보여줄 슬라이드 개수
  // slidesPerGroup: 5, // 한 번에 슬라이드 할 개수(전체 개수로 나뉘어야 함)
  navigation: { // 슬라이드 이전/다음 버튼 사용 여부
    prevEl: '.awards .swiper-prev', // 이전 버튼 선택자
    nextEl: '.awards .swiper-next' // 다음 버튼 선택자
  }
})

```

#### ► main.css

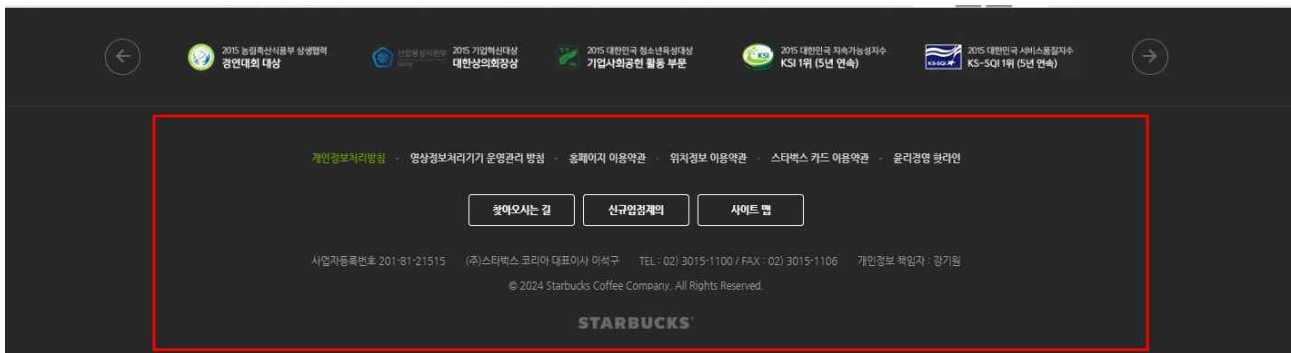
```

/*AWARDS*/
.awards {
  background-color: #272727;
}
.awards .inner {
  padding: 40px 0;
}

```

```
.awards .swiper-container {  
  width: 100%;  
  height: 40px;  
}  
.awards .swiper-prev,  
.awards .swiper-next {  
  width: 42px;  
  height: 42px;  
  outline: none;  
  border: 2px solid #fff;  
  border-radius: 50%;  
  color: #fff;  
  position: absolute;  
  top: 0;  
  bottom: 0;  
  margin: auto;  
  opacity: .3;  
  cursor: pointer;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  transition: .4s;  
}  
.awards .swiper-prev {  
  left: -100px;  
}  
.awards .swiper-next {  
  right: -100px;  
}  
.awards .swiper-prev:hover,  
.awards .swiper-next:hover {  
  background-color: #fff;  
  color: #333;  
}
```

## 9. 푸터



## ① 외형만들고 css 적용하기

### ▶ index.html

```
<!--FOOTER-->
<footer>
  <div class="inner">

    <ul class="menu">
      <li><a href="javascript:void(0)" class="green">개인정보처리방침 </a></li>
      <li><a href="javascript:void(0)">영상정보처리기기 운영관리 방침 </a></li>
      <li><a href="javascript:void(0)">홈페이지 이용약관 </a></li>
      <li><a href="javascript:void(0)">위치정보 이용약관 </a></li>
      <li><a href="javascript:void(0)">스타벅스 카드 이용약관 </a></li>
      <li><a href="javascript:void(0)">윤리경영 핫라인 </a></li>
    </ul>

    <div class="btn-group">
      <a href="javascript:void(0)" class="btn btn--white">찾아오시는 길 </a>
      <a href="javascript:void(0)" class="btn btn--white">신규입점제외 </a>
      <a href="javascript:void(0)" class="btn btn--white">사이트 맵 </a>
    </div>

    <div class="info">
      <span>사업자등록번호 201-81-21515</span>
      <span>(주)스타벅스 코리아 대표이사 이석구</span>
      <span>TEL : 02) 3015-1100 / FAX : 02) 3015-1106</span>
      <span>개인정보 책임자 : 강기원</span>
    </div>

    <p class="copyright">
      &copy; <span class="this-year"></span> Starbucks Coffee Company. All Rights Reserved.
    </p>
```

```


</div>
</footer>
```

► main.css

```
/*FOOTER*/
footer {
  background-color: #272727;
  border-top: 1px solid #333;
}
footer .inner {
  padding: 40px 0 60px 0;
}
footer .menu {
  display: flex;
  justify-content: center;
}
footer .menu li {
  position: relative;
}
footer .menu li::before {
  content: "";
  width: 3px;
  height: 3px;
  background-color: #555;
  position: absolute;
  top: 0;
  bottom: 0;
  right: -1px;
  margin: auto;
}
footer .menu li:last-child::before {
  display: none;
}
footer .menu li a {
  display: block;
  color: #CCC;
  font-size: 12px;
  font-weight: 700;
```

```
padding: 15px;
}
footer .menu li a.green {
  color: #669900;
}
footer .btn-group {
  margin-top: 20px;
  display: flex;
  justify-content: center;
}
footer .btn-group .btn {
  font-size: 12px;
  margin-right: 10px;
}
footer .btn-group .btn:last-child {
  margin-right: 0;
}
footer .info {
  margin-top: 30px;
  text-align: center;
}
footer .info span {
  margin-right: 20px;
  color: #999;
  font-size: 12px;
}
footer .info span:last-child {
  margin-right: 0;
}
footer .copyright {
  color: #999;
  font-size: 12px;
  text-align: center;
  margin-top: 12px;
}
footer .logo {
  margin: 30px auto 0;
}
```

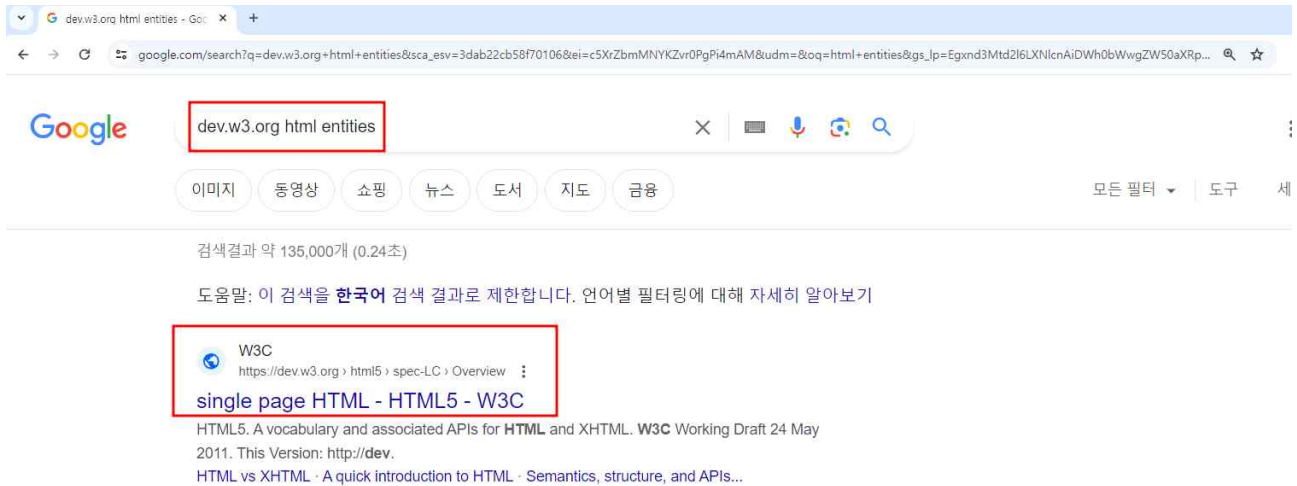
► main.js

```

/**
 * 올해가 몇 년도인지 계산
 */
const thisYear = document.querySelector('.this-year')
thisYear.textContent = new Date().getFullYear()

```

## ② html에서 사용하는 다양한 특수문자 찾기



## 10. 페이지 상단으로 이동

▶ index.html

```

<!--TO TOP BUTTON-->
<div id="to-top">
  <div class="material-icons">arrow_upward</div>
</div>

```

※ 스크롤 기능을 사용하기위한 gsap 스크롤기능 삽입





▶ index.html

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/gsap/3.12.5/ScrollToPlugin.min.js"
integrity="sha512-1PKqXBz2ju2JcAerHKL0ldg0PT/1vr3LghYAtc59+9xy8e19QEtaNUyt1gprouyWnpOPq
NJjL4gXMRMEpHYyLQ==" crossorigin="anonymous" referrerpolicy="no-referrer"> </script>
```

▶ main.css

▶ main.js

```
const badgeEl = document.querySelector('header .badges');
const toTopEl = document.querySelector('#to-top');
// scroll 이벤트가 수십개 이상발생하니 화면이 버벅이는 현상이 생김
// 이를 방지하기 위해 외부에서 제공하는 자바라이브러리를 사용해 보자
// 0.3초 마다 scroll 이벤트가 발생되도록 처리
// _.throttle(함수, 시간)
window.addEventListener('scroll', _.throttle(function(){
  // console.log('scroll');
  console.log(window.scrollY);
  if(window.scrollY > 500){
    //배지 숨기기
    // badgeEl.style.display = 'none';
    // gsap.to(요소, 지속시간(초), 옵션)
    gsap.to(badgeEl, 0.6, {
      opacity : 0,
      display : 'none',
    });
    // 맨 위로 이동하기 버튼 보이기
    gsap.to(toTopEl, 0.2, {
      x : 0
    })
  } else {
    //배지 보이기
    // badgeEl.style.display = 'block';
    gsap.to(badgeEl, 0.6, {
      opacity : 1,
```

```
        display : 'block',
    });
    // 버튼 숨기기
    gsap.to(toTopEl, 0.2, {
        x : 100
    })
    }
},300));

toTopEl.addEventListener('click', function() {
    gsap.to(window, 0.7, {
        scrollTo: 0
    });
});
```

