

마무리 작업(수정, 추가, 삭제)



제품 상세(goods-detail.html) 오버레이 이슈

[추가 사항] 구매하기 팝업 z-index 추가

```
.goods-order-info {  
  background-color: #fff;  
  border-radius: 10px 10px 0 0;  
  box-shadow: 0 -5px 5px rgba(0, 0, 0, 0.05);  
  padding: 20px;  
  position: fixed;  
  bottom: 0;  
  width: 375px;  
  z-index: 1000;  
}
```

.goods-order-info가 .overlay-fold 아래로 위치하는 문제 해결

```
.goods-order-final {  
  display: none;  
  z-index: 1000; 삭제(불필요)  
}
```

[추가 사항] .btn-fold가 수평 중앙에 있지 않은 문제



```
.btn-fold {  
  background-color: transparent;  
  border: none;  
  width: 100%;  
  position: absolute;  
  top: 5px;  
  left: 0;  
}
```

마무리 작업(수정, 추가, 삭제)



전체 너비 사이즈 조정(375px → 100%)

[수정 사항] .container를 고정에서 유동적으로 바꾸기

```
/* Layout CSS */
.container {
  width: 100%;
  position: relative;
}
.preloading,
.signin,
.signup {
  height: 100vh;
}
```

- .container에 너비를 삭제하거나 100% 주기
- 스크롤이 없는 페이지는 뷰포트에 맞게 보이도록 높이를 100vh 주기

[수정 사항] .gnb를 브라우저 하단에 스크롤에 관계 없이 고정하고 너비 100%

```
/* GNB CSS */
.gnb {
  border-top: 1px solid var(--gloudy-gray);
  height: 65px;
  display: flex;
  justify-content: space-between;
  position: fixed;
  bottom: 0;
  width: 100%;
  background-color: #fff;
}
```



[수정 사항] .goods-detail-info를 브라우저 하단에 고정하고 너비 100%

```
/* Goods Order Info */
.goods-order-info {
  background-color: #fff;
  border-radius: 10px 10px 0 0;
  box-shadow: 0 -5px 5px rgba(0, 0, 0, 0.05);
  padding: 20px;
  position: fixed;
  bottom: 0;
  width: 100%;
  z-index: 1000;
}
```

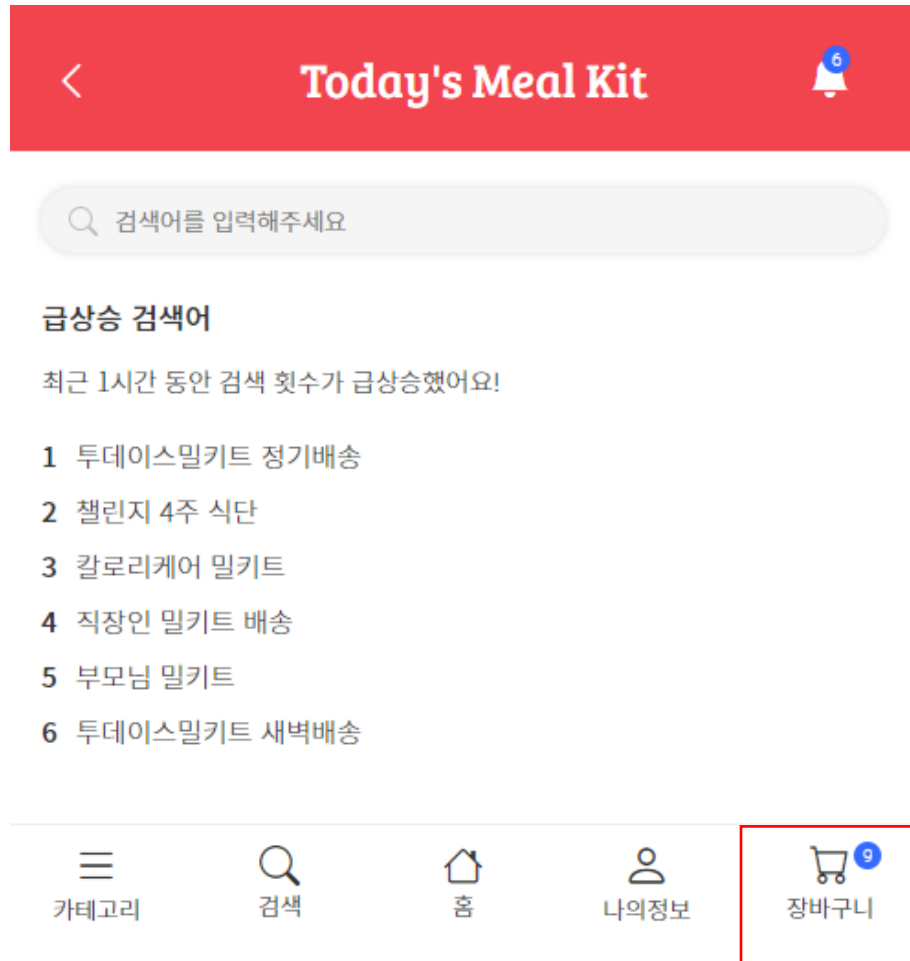


마무리 작업(수정, 추가, 삭제)



GNB(gnb.html) 알림 아이콘 고정되지 않는 이슈

[수정 사항] 장바구니 알림 위치 유동적 → 고정적



```
.user-alarm span,  
.cart-alarm b {  
  border: 1px solid #000;  
  position: relative;  
  width: 30px;  
  margin: auto;  
}
```

```
.cart-alarm b em {  
  color: #fff;  
  top: 0;  
  right: -5px;  
}
```

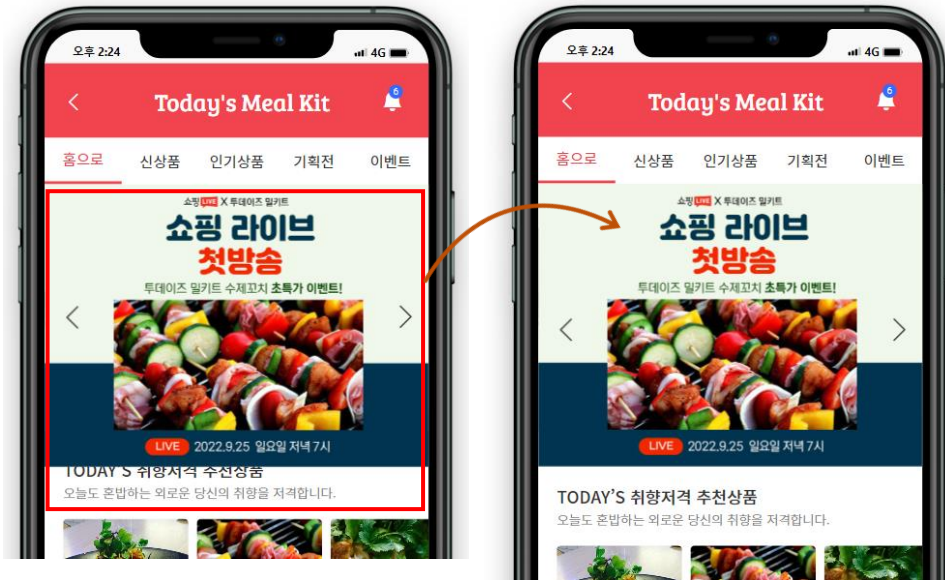
.cart-alarm b 가 %로 유동적으로 움직이면서 .cart-alarm b em 의 위치도 유동적으로 바뀌는 것을 방지



마무리 작업(수정, 추가, 삭제)



프론트 슬라이더 높이 이슈 & 콘텐츠 아래쪽 마진 주기



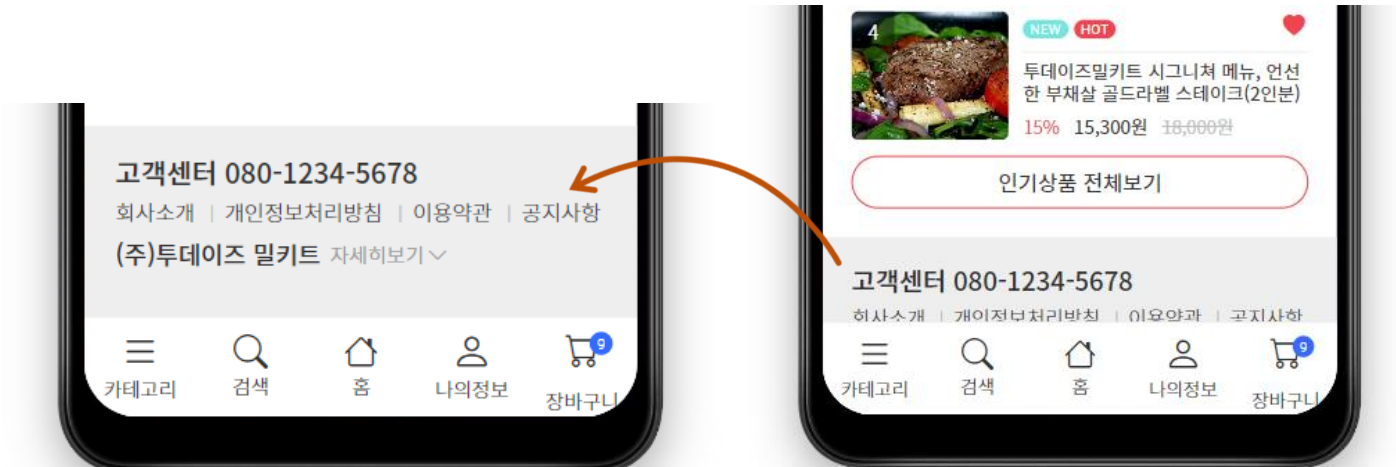
[수정 사항] 모바일 디바이스가 커지면서 슬라이드 이미지가 아래쪽으로 커져서 이어지는 콘텐츠를 가리는 문제

```
.front-slider {
  height: 275px;
}
.front-slider a img {
  width: 100%;
}
```

- .front-slider에 고정된 높이 값 삭제하면 자동으로 높이는 auto로 변경됨
- .front-slider a img에 유동적인 너비로 100%

[수정 사항] 스크롤이 100vh 이상인 페이지에서 마지막 부분이 하단에 Fixed 되어 있는 .gnb에 가려져 있는 부분을 보완

```
/* Control Content Class Name */
[class*=content] {
  padding: 20px;
  margin-bottom: 65px;
}
```



마무리 작업(수정, 추가, 삭제)

슬릭(Slick) 슬라이더 CDN 사용하기

[참고 사항] 슬릭 슬라이더와 같은 유명한 제이쿼리 플러그인은 다운로드 후 파일 링크로 사용할 수도 있지만 CDN을 간편하게 사용할 수도 있습니다.

CDN js 공식 웹사이트 : www.cdnjs.com

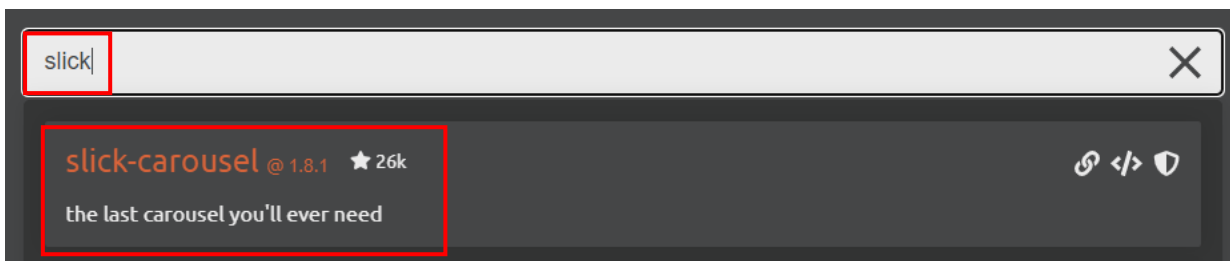
```
<!-- Slick Slider CDN -->
<script src="https://cdnjs.cloudflare.com/ajax/libs/slick-carousel/1.8.1/slick.min.js"></script>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/slick-carousel/1.8.1/slick-theme.css">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/slick-carousel/1.8.1/slick.css">
```

▲ 슬릭 슬라이더 CDN을 링크한 경우

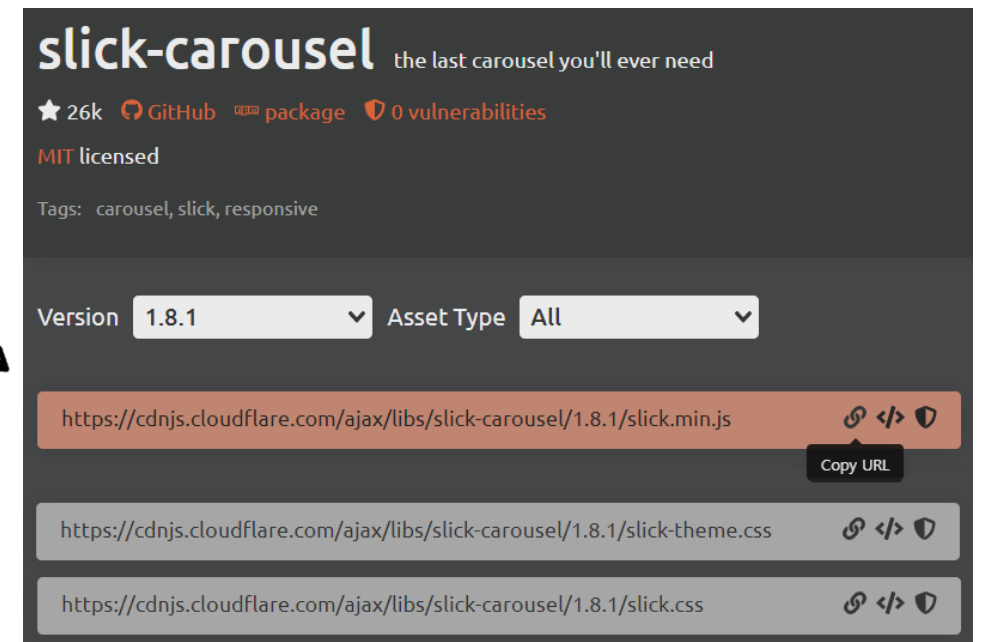
```
<!-- Slick Slider -->
<script src="/js/slick/slick.min.js"></script>
<link rel="stylesheet" href="/js/slick/slick-theme.css">
<link rel="stylesheet" href="/js/slick/slick.css">
```

▲ 슬릭 슬라이더 플러그인 다운로드 후 파일 링크

1) cdn.js에서 slick 검색



2) 아래 3가지 파일 Copy URL 후 head 사이에 CDN 링크해서 사용



```
<div class="container preloading">
  <div class="preloading-inner" onclick="location.href='/html/front.html'">
    <div class="heading-preloading">
      <h2 class="logo">Today's Meal Kit</h2>
      <p class="slogan">
        <span>나를 위해</span>
        <span>든든한 한끼를</span>
        <span>부탁해!</span>
      </p>
    </div>
  </div>
</div>

<script>
  /* 3초 후 해당 front.html로 이동 */
  setTimeout("location.href='/html/front.html'",3000)
</script>
```

인라인 자바스크립트로 페이지 링크하기

자바스크립트 setTimeout 함수를 사용해서 일정 시간 후 지정된 페이지로 이동하기

▲ preloading.html

[파일을 링크하는 인라인 자바스크립트 및 기타 인라인 자바스크립트]

- 현재 탭 <div onclick="location.href='http://www.inflern.com'">DIV</div>
- 새로운 탭 <div onclick="window.open('http://www.inflern.com','_blank')">DIV</div>

쌍 따옴표와 외 따옴표를 잘 구분해서 사용해야 합니다. 하나라도 잘못 사용하면 작동하지 않습니다.

```
<a href="/html/front.html">
  <div class="container preloading">
    <div class="preloading-inner">
      <div class="heading-preloading">
        <h2 class="logo">Today's Meal Kit</h2>
        <p class="slogan">
          <span>나를 위해</span>
          <span>든든한 한끼를</span>
          <span>부탁해!</span>
        </p>
      </div>
    </div>
  </div>
</a>
```

▲ a태그로 링크해도 되지만 추천하는 방식은 아닙니다.

- 페이지를 링크하는 방식은 a태그가 사용된 구조라면 a태그에 링크하고 사용되지 않은 경우 인라인 자바스크립트 onclick을 사용해서 링크
- 일정 시간 후 페이지로 이동은 꼭 이렇게 해야 하는 것은 아니고 상황에 따라 활용
- 새로 고침
- window.close()">창 닫기

마무리 작업(수정, 추가, 삭제)

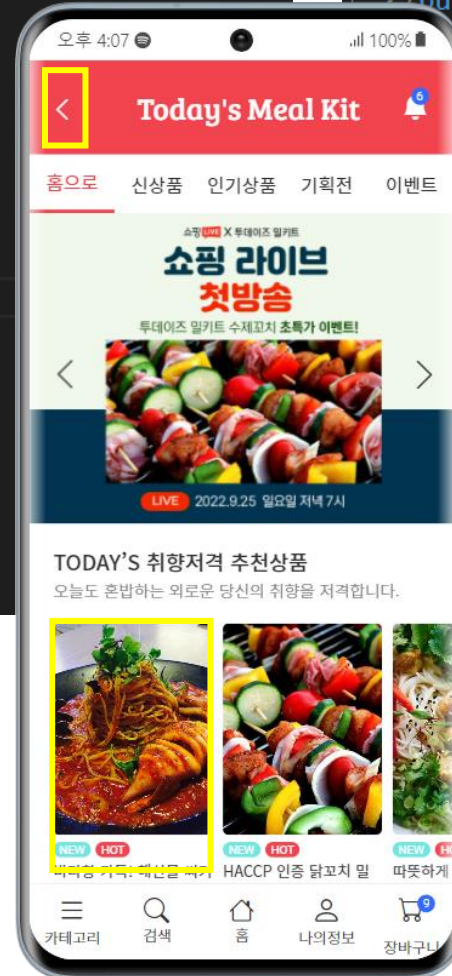
인라인 자바스크립트로 HTML 파일 링크하기(2)

```
<div class="goods-suggestion-items">
  <div class="goods-suggestion-item" onclick="location.href='goods-detail.html'">
    <div class="thum">
      
    </div>
    <div class="desc">
      <div class="goods-badge">
        <span class="badge new">new</span>
        <span class="badge hot">hot</span>
      </div>
      <div class="goods-title">
        <h3>바다향 가득! 해산물 씨키트 해물 스파게티 밀키트</h3>
        <div>
          <span>30%</span><b>11,760원</b><s>16,800원</s>
        </div>
      </div>
    </div>
  </div>
</div>
```

인라인 자바스크립트로 페이지 링크하기

```
<div class="main-header">
  <button class="btn-back" onclick="history.back()">
    <i class="bi bi-chevron-left"></i>
  </button>
  <h2 class="logo">Today's Meal Kit</h2>
  <div class="user-alarm">
    <span>
      <i class="bi bi-bell-fill"></i>
      <em data-count="6"></em>
    </span>
  </div>
</div>
```

이전 페이지로 이동하는 자바스크립트



[a태그에 사용할 경우와 그 외의 경우 예시]

- `뒤로 이동`
- `앞으로 이동`
- `<button onclick="history.back()">뒤로 이동</button>`

[앞뒤 페이지로 이동하는 인라인 자바스크립트]

- `history.go(1);` // 한 단계 앞으로 이동
- `history.go(2);` // 두 단계 앞으로 이동
- `history.back(-1);` // 한 단계 이전으로 이동
- `history.back(-2);` // 두 단계 이전으로 이동
- `history.back();` // `history.back(-1)`과 동일
- `history.forward();` // `history.go(1)`과 동일

※ 여러가지가 있지만 `history.back();` 을 주로 사용함

마무리 작업(수정, 추가, 삭제)



GNB 파일 링크하기 & 모든 작업 페이지 빠짐없이 보여주기

```
<div class="gnb">
  <a href="/html/category.html"><i class="bi bi-list"></i><span>카테고리</span></a>
  <a href="/html/search.html"><i class="bi bi-search"></i><span>검색</span></a>
  <a href="/html/front.html"><i class="bi bi-house"></i><span>홈</span></a>
  <a href="/html/mypage.html"><i class="bi bi-person"></i><span>나의 정보</span></a>
  <a class="cart-alarm" href="/html/cart.html">
    <b>
      <i class="bi bi-cart"></i><em data-count="9"></em>
    </b>
    <span>장바구니</span>
  </a>
</div>
```

퍼블리싱 작업한 모든 파일을 잘 보여줄 수 있도록 꼭! 링크해야 합니다.(실 작업이 아니니 중복된 링크일지라도 모두 링크합니다.)

- 붉은 색 박스로 된 부분은 인라인 자바스크립트로 같은 파일 링크를 모두 연결합니다.
- 탭 메뉴의 경우 기능만 하고 임시 이미지를 넣어서 보여주면 안됩니다. 실 작업이라면 그럴 수도 있지만 퍼블리셔 개인 포트폴리오 퍼블리싱 결과물로 보여주는 것이라면 탭 메뉴 기능 뿐만 아니라 이미지 텍스트를 모두 변경하는 노력은 해야 합니다.

작업한 페이지에 없으면 없는 데로 그냥 슬라이드 기능만 해도 됩니다.



마무리 작업(수정, 추가, 삭제)



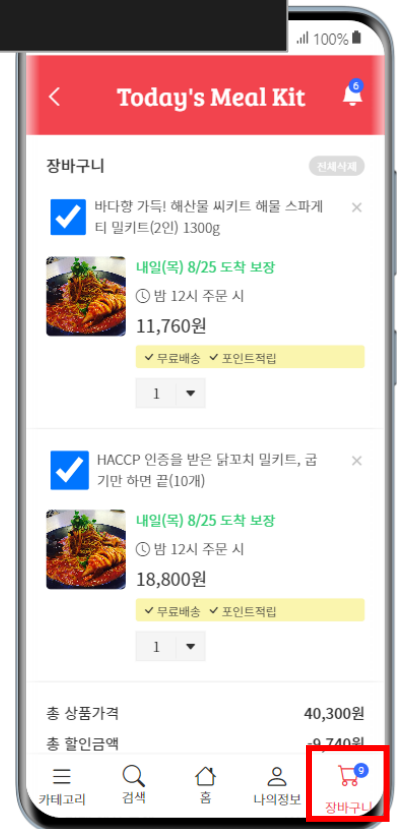
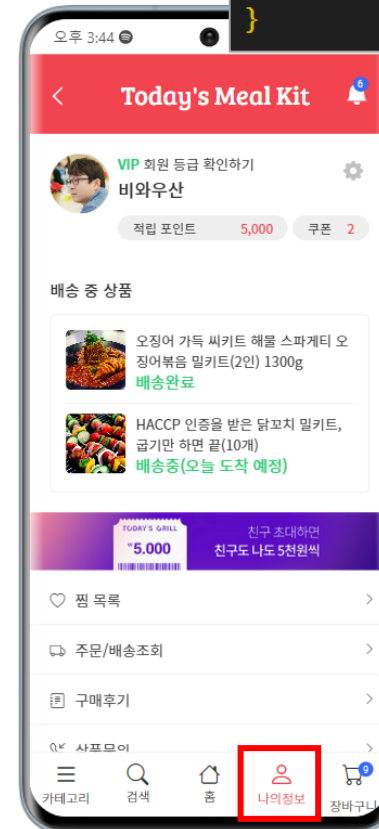
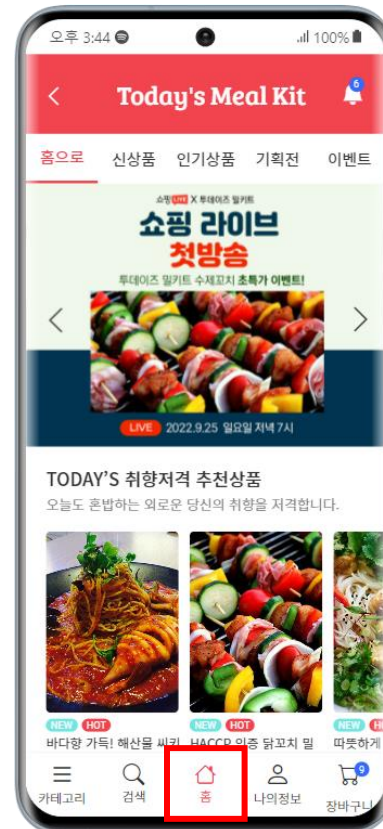
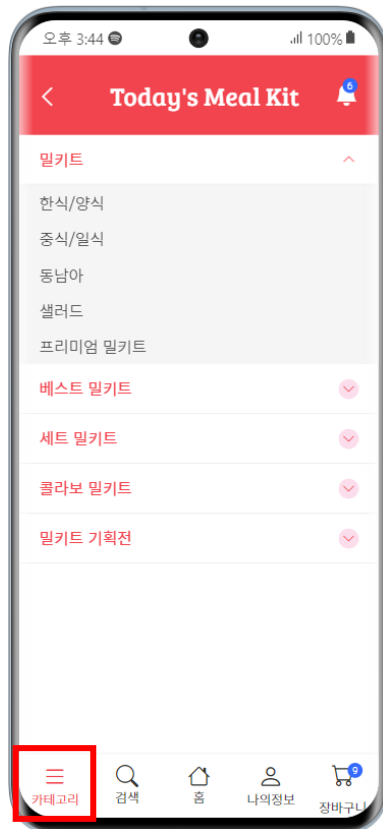
GNB에 연결된 파일에서 GNB 활성화된 네비게이션 체크

```
<div class="gnb">
```

```
① <a href="/html/category.html"><i class="bi bi-list"></i><span>카테고리</span></a>
② <a href="/html/search.html"><i class="bi bi-search"></i><span>검색</span></a>
③ <a href="/html/front.html"><i class="bi bi-house"></i><span>홈</span></a>
④ <a href="/html/mypage.html"><i class="bi bi-person"></i><span>나의 정보</span></a>
⑤ <a class="cart-alarm" href="/html/cart.html">
```

```
/* GNB Navigation Check */
.category .gnb a:nth-child(1),
.search .gnb a:nth-child(2),
.front .gnb a:nth-child(3),
.mypage .gnb a:nth-child(4),
.cart .gnb a:nth-child(5) {
  color: var(--crimson);
}
```

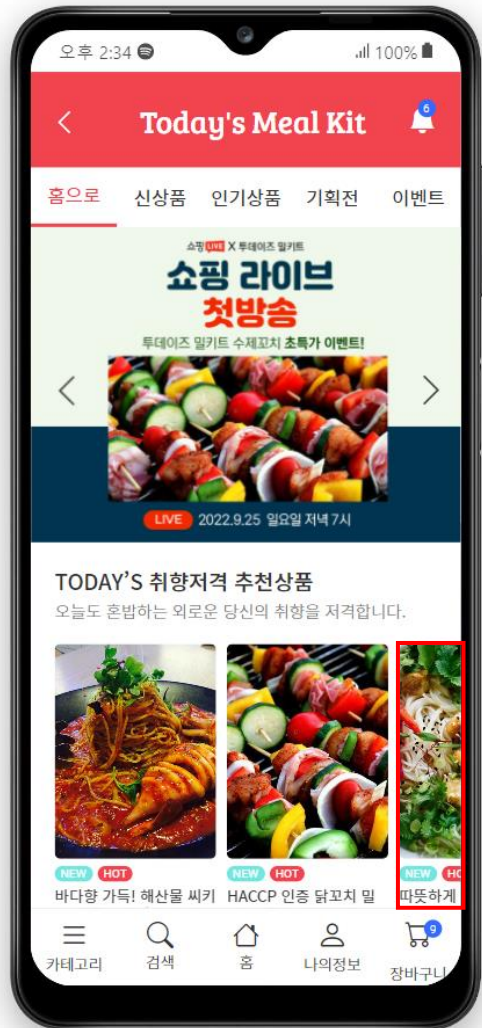
필수 작업
사항은 아님



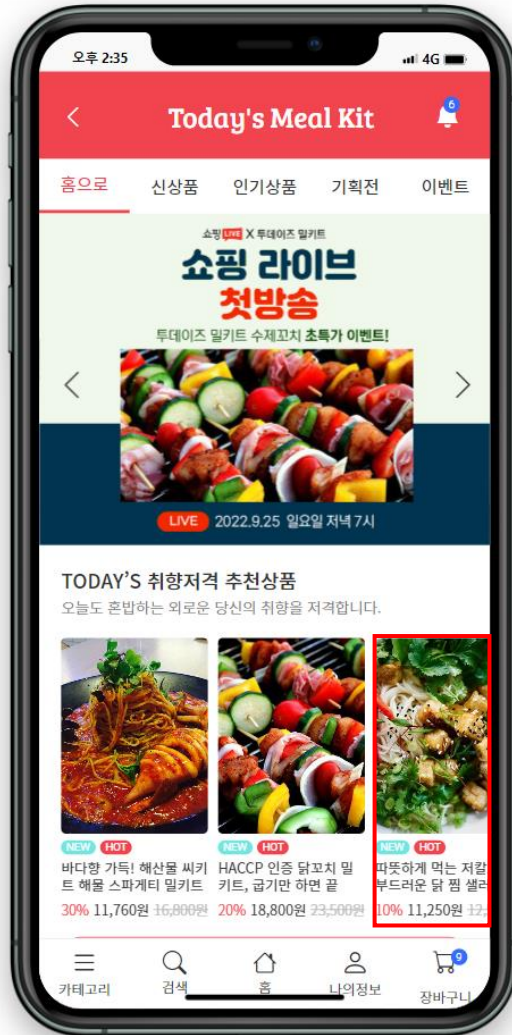
마무리 작업(수정, 추가, 삭제)



변수(Var) 활용한 큰 사이즈 모바일과 태블릿에 맞춘 레이아웃(1)



▲ Galaxy A12



▲ iPhone 11 Pro Max

[보완 사항] 모바일 디바이스가 커지면서 빠르게 슬라이더에서 반 정도만 보였던 슬라이더 아이템이 70~80%까지 보이는 부분을 보완하고 싶다면 미디어쿼리를 사용해서 슬라이더의 너비를 조정해 주어야 함. (html의 head 사이에 viewport meta 태그 필수)

```
.goods-suggestion-items {  
  width: 430px;  
  margin-bottom: 15px;  
}  
.goods-suggestion-item .thumb {  
  width: 135px;  
  height: 180px;  
}  
.goods-suggestion-item .thumb i {  
  width: inherit;  
  height: inherit;  
  object-fit: cover;  
  border-radius: 5px;  
}  
.goods-suggestion-item .desc {  
  width: 135px;  
}  
@media (min-width: 400px) {  
  .goods-suggestion-items {  
    width: 500px;  
  }  
  .goods-suggestion-item .thumb {  
    width: 155px;  
    height: 200px;  
  }  
  .goods-suggestion-item .desc {  
    width: 155px;  
  }  
}
```

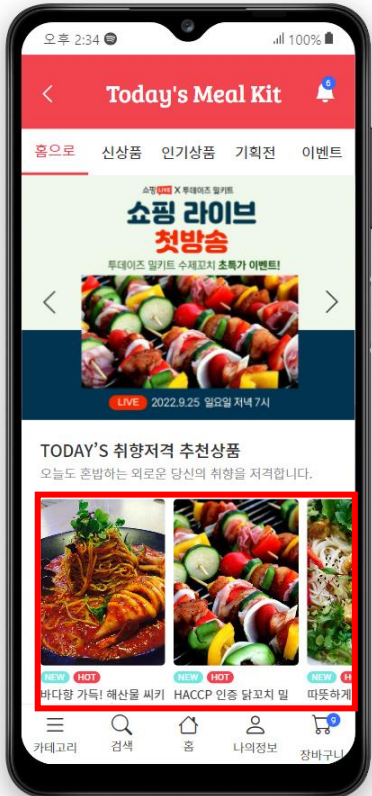
```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

완성된 페이지 별로 라이브 프리뷰를 확인하는 방법으로 브라켓(Brackets) 텍스트 에디터를 사용하면 좋습니다.

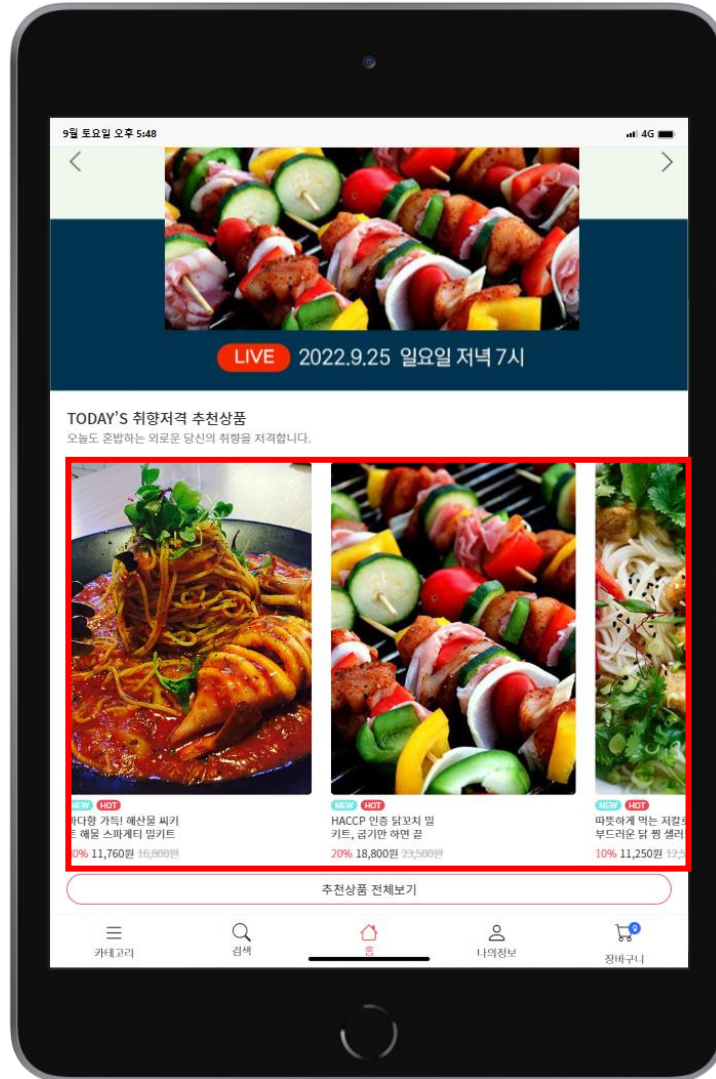
마무리 작업(수정, 추가, 삭제)



변수(Var) 활용한 큰 사이즈 모바일과 태블릿에 맞춘 레이아웃(2)



▲ Galaxy A12



▲ iPad Mini

```
/* For Big Size Mobile & Tablet Device */  
--slide-basis: 370px;
```

▲ 슬라이드 너비 기준이 될 변수 선언

```
/* For Tablet Device */  
.goods-suggestion-items {  
  width: calc(var(--slide-basis) * 1.16);  
  margin-bottom: 15px;  
}  
.goods-suggestion-item .thumb {  
  width: calc(var(--slide-basis) * 0.36);  
  height: calc(var(--slide-basis) * 0.49);  
}
```

▲ 슬라이드 너비 변수에 맞춰 비율로 조정

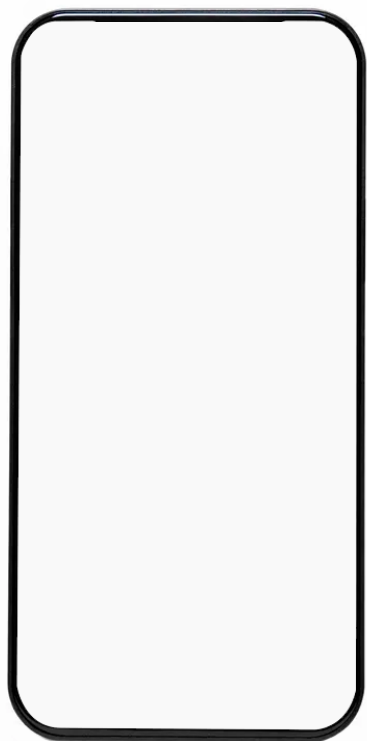
```
@media (min-width: 400px) {  
  :root {  
    --slide-basis: 420px;  
  }  
}  
  
@media (min-width: 768px) and (max-width: 1024px) {  
  :root {  
    --slide-basis: 820px;  
  }  
}
```

▲ 미디어 쿼리 구간에서 슬라이드 변수 개별적으로 재선언

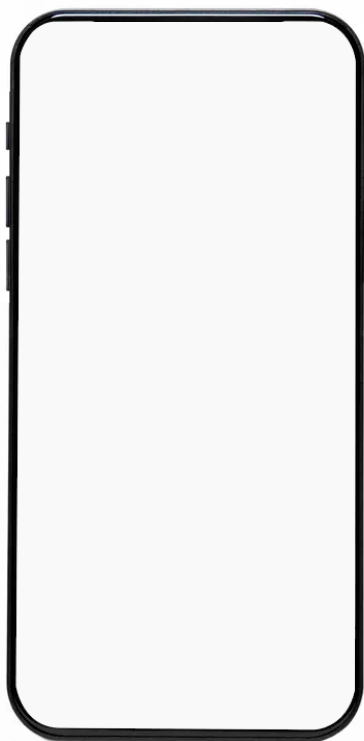
점프업(Jump Up)



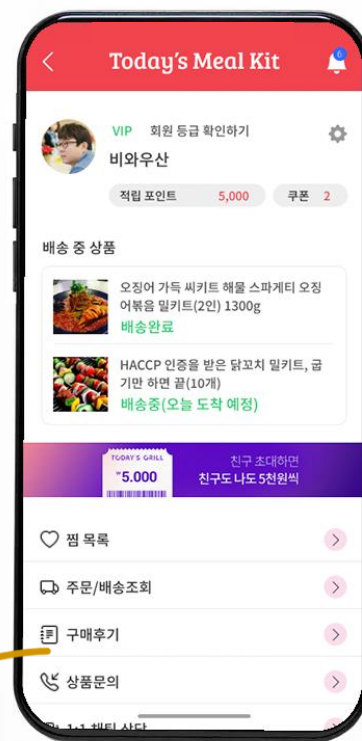
HTML+CSS로 나이스한 모바일 목업(Mockup) 만들고 iframe으로 활용하기



▲ 1단계
모바일 목업 심플형

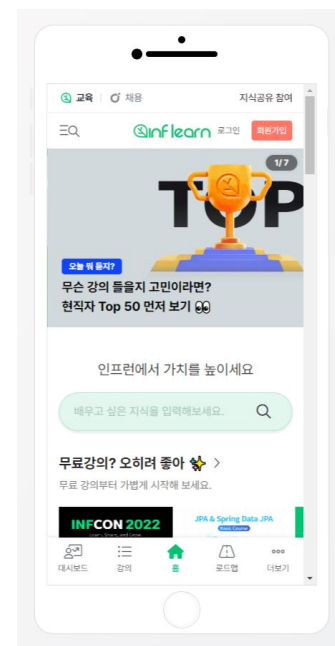


▲ 2단계
모바일 목업 기본형

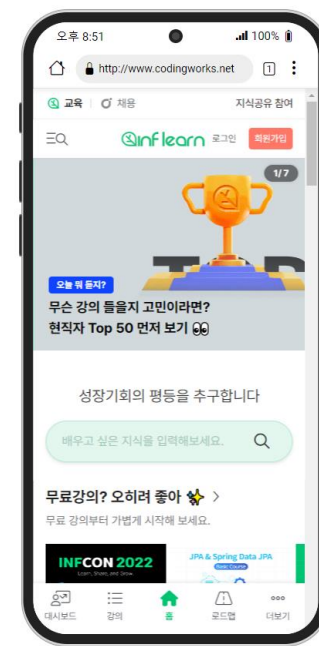


▲ 3단계
iframe으로 퍼블리싱 작업한 html 불러오기

수강생 참고자료 다운로드에서 완성본을 다운받을 수 있습니다.



▲ 모바일 목업 완성본(화이트)



▲ 모바일 목업 완성본(블랙)

```
<iframe src= "불러올 파일 경로" frameborder="0" width= "px or %" height= "px or %"></iframe>
```

iframe에 파일을 불러오지 못할 경우 대부분 경로를 잘못
넣어서 그렇습니다. 파일 경로를 먼저 확인하세요.

border, width, height 속성은 너비와 높이는 CSS로 조절하기 때문에 사용할 필요 없음
(frameborder는 텍스트에디터가 기본적으로 만들어 주니까 그냥 씬)

점프업(Jump Up)

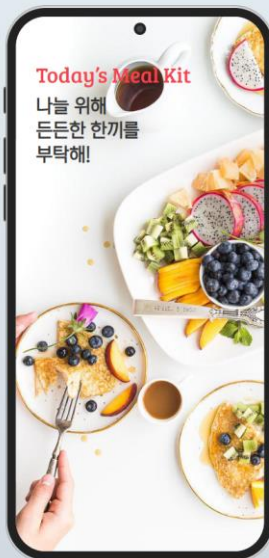


PC 유저와 모바일 유저를 자바스크립트로 체크해서 분기하기(1)

☞ 모바일 웹을 반드시 플레이스토어, 앱스토어에서 서비스하지 않는 경우도 많이 있습니다. PC 유저도 볼 수 있도록 대비해야 합니다.

index.html에 유저가 접속한 디바이스를 체크하는 자바스크립트를 넣고 접속 디바이스를 체크해서 모바일 전용 또는 PC 전용을 체크해서 리다이렉트(Redirect)로 분기합니다.

PC에서 보여주는 방법 #01



single-view.html

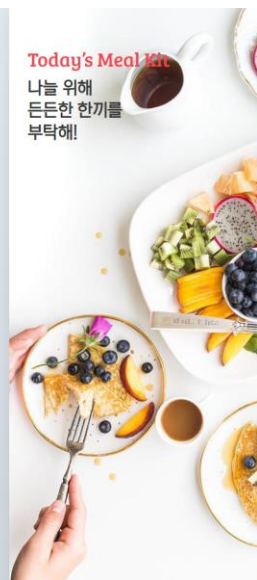
- index.html에서 single-view.html로 리다이렉트 하기
- single-view.html 내의 iframe 링크를 보여줄 첫 페이지로 설정
Ex) `<iframe src="/preloading.html" frameborder="0"></iframe>`
- 모바일 목업은 너무 튀지 않는 선에서 무난한 것으로 선택

Beautiful CSS box-shadow examples(예쁜 박스 쉐도우 구하기)

<https://getcssscan.com/css-box-shadow-examples>

← single-view.html의 배경은 정책에 따라 디자인 컨셉에 따라 알맞게 보여주기 →

PC에서 보여주는 방법 #02



without-mockup.html

- index.html에서 without-mockup.html로 리다이렉트 하기
- without-mockup.html 내의 iframe 링크를 보여줄 첫 페이지로 설정
Ex) `<iframe src="/preloading.html" frameborder="0"></iframe>`

점프업(Jump Up)



PC 유저와 모바일 유저를 자바스크립트로 체크해서 분기하기(1)

☞ 실제 모바일 디바이스에서 보여질 때 반드시 모든 html 파일에 뷰포트 meta 태그가 있어야 합니다.

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>Welcome to Today's Meal Kit</title>
</head>
<body>

  <script>
    /* 모바일 디바이스를 감지해 모바일 페이지로 리다이렉트 되는 스크립트 */
    var UserAgent = navigator.userAgent;
    if (UserAgent.match(
      /iPhone|iPod|Android|Windows CE|BlackBerry|Symbian|Windows Phone|webOS|Opera Mini|
      Opera Mobi|POLARIS|IEMobile|lgtelecom|nokia|SonyEricsson/i
    ) != null || UserAgent.match(/LG|SAMSUNG|Samsung/) != null) {
      // alert("Mobile로 방문하셨습니다.");
      location.href = "/preloading.html";
    } else {
      // alert("PC로 방문하셨습니다.");
      location.href = "/mobile-mockup/single-view.html";
      location.href = "/mobile-mockup/without-mockup.html";
    }
  </script>

</body>
</html>
```

index.html



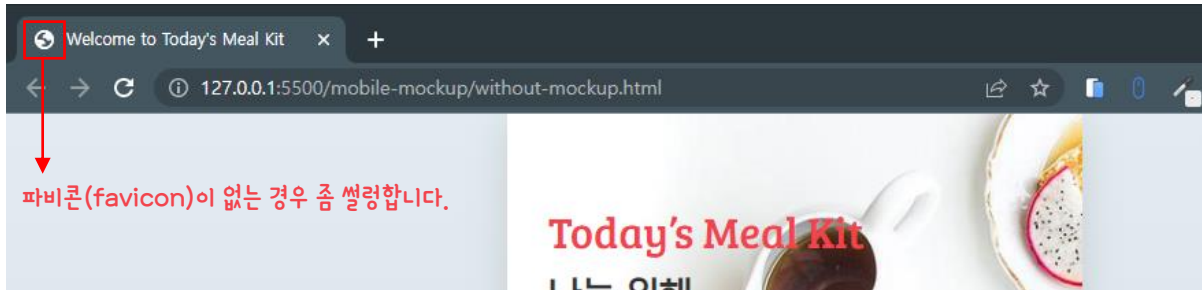
▲ 모바일 목업이 없는 너비 100% 상태의 html이 보여 지므로, 실제 모바일로 접속한 유저는 모바일 디바이스에 딱 맞게 보여집니다.



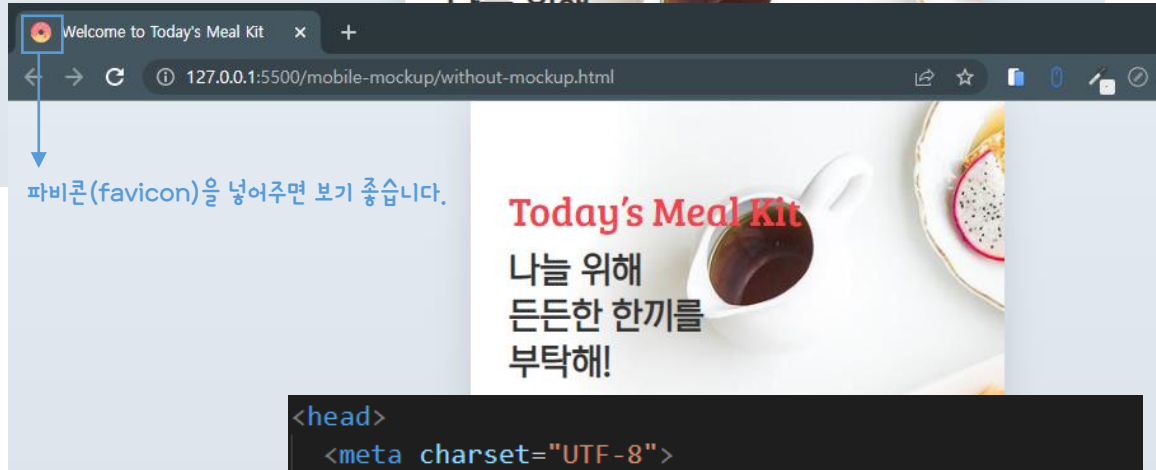
해당 자바스크립트 소스 텍스트는 수강생 참고자료에 있습니다. 복사해서 사용하세요.

마무리 작업(수정, 추가, 삭제)

PC로 접속를 위해 파비콘(favicon) 넣기 / 로고에 링크하기



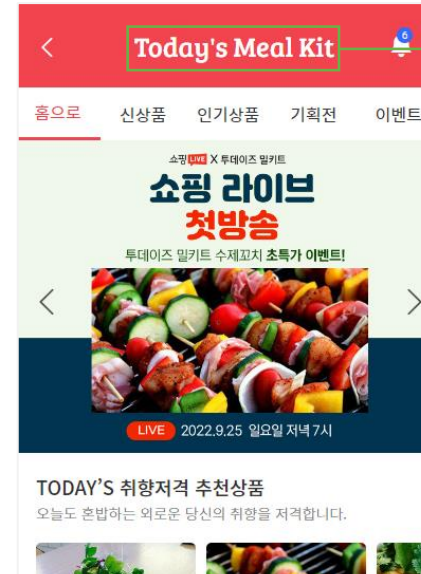
파비콘(favicon)이 없는 경우 좀 썰렁합니다.



파비콘(favicon)을 넣어주면 보기 좋습니다.

```
<head>
  <meta charset="UTF-8">
  <title>Welcome to Today's Meal Kit</title>
  <link rel="stylesheet" href="mobile-mockup.css">
  <link rel="icon" href="/images/favicon.png">
</head>
```

- Flat Icons(파비콘 구하기) <https://www.flaticon.com/>
- Freepik 일러스트 이미지 구하기 <https://www.freepik.com/>



- 일반적으로 사용자가 처음 페이지로 가려고 할 때 헤더의 로고를 누르는 경우가 흔합니다.
- 이런 경우를 위해 header에 있는 .logo에 인라인 자바스크립트로 처음 페이지를 링크해 두는 것이 좋습니다.
- Ex) front.html

```
<div class="main-header">
  <button class="btn-back" onclick="history.back()">
    <i class="bi bi-chevron-left"></i>
  </button>
  <h2 class="logo" onclick="location.href='/html/front.html'">
    Today's Meal Kit
  </h2>
  <div class="user-alarm">
    <span>
      <i class="bi bi-bell-fill"></i><em data-count="6"></em>
    </span>
  </div>
</div>
```

main-header.html

점프업(Jump Up)

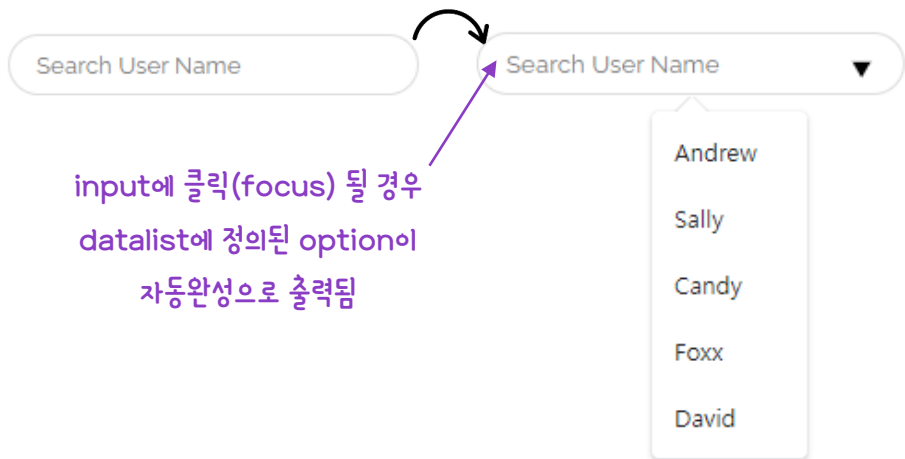


datalist 태그로 데이터 리스트(Data List) 제안하기

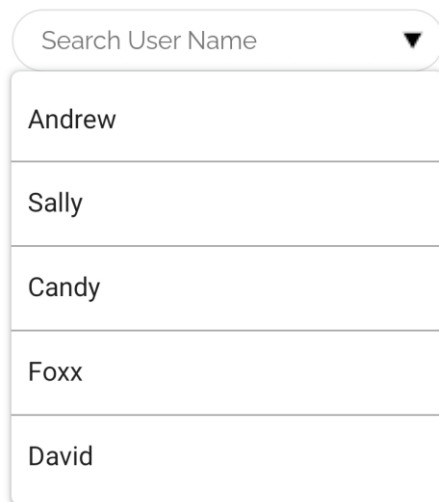
HTML5에서 새롭게 추가된 <datalist> 태그는 input 요소에 클릭(focus) 되었을 경우 datalist의 내용을 제안하는 기능으로 search.html에 활용

[HTML <datalist> 태그]

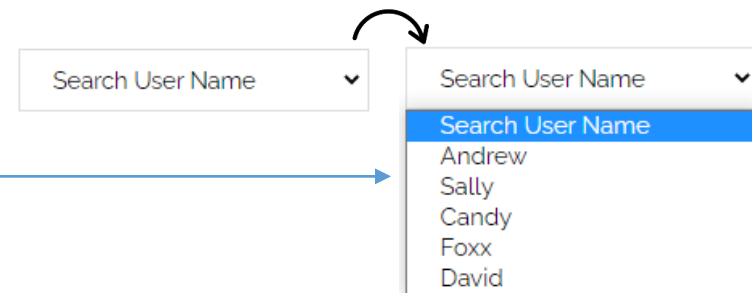
- <datalist> 요소는 HTML5에서 새롭게 추가된 요소
- <datalist> 요소는 <input> 요소에 연결해서 미리 정의된 옵션들 자동완성 기능으로 보여줌
- <input> 요소의 list 속성 값을 <datalist> 요소의 id 속성 값과 매칭
- <select>와 <datalist>의 차이는 select는 미리 정의되어 있는 옵션을 선택할 수 있지만 사용자가 목록에 없는 값을 입력할 수 없습니다. 하지만 datalist는 option 목록에 없는 값을 사용자의 입력을 받을 수 있습니다.



▲ PC에서 input을 클릭한 모습



▲ 실제 Mobile Device에서 input을 클릭한 모습



```
<input list="username" placeholder="Search User Name">
<datalist id="username">
  <option value="Andrew"></option>
  <option value="Sally"></option>
  <option value="Candy"></option>
  <option value="Foxx"></option>
  <option value="David"></option>
</datalist>
```

점프업(Jump Up)



개별 문서를 페이지의 원하는 곳에 불러오는 load() 메서드 & 콜백함수



- 공통 영역으로 쓰이는 개별 문서를 페이지의 원하는 곳에 불러오는 load() 메서드
- load로 불러와진 파일에 있는 제이쿼리는 반드시 콜백 함수로 사용해야 합니다.



```
$('.main-header-include').load('/include/main-header.html') ①  
$('.trigger').click(function(){  
    $('.sidebar').toggleClass('active') ②  
})
```

- ① main-header.html을 로드 완료
- ② main-header.html 내의 요소에 연결된 제이쿼리 구문은 불러오지 못한 상태로 .trigger에 연결된 제이쿼리는 휘발되어 버립니다.

```
$('.main-header-include').load('/include/main-header.html', function(){  
    $('.trigger').click(function(){  
        $('.sidebar').toggleClass('active') ④  
    })  
}) ③
```

괄호를 닫아서 마무리 하기 전에 콤마(,)를
넣고 function을 다시 선언합니다.

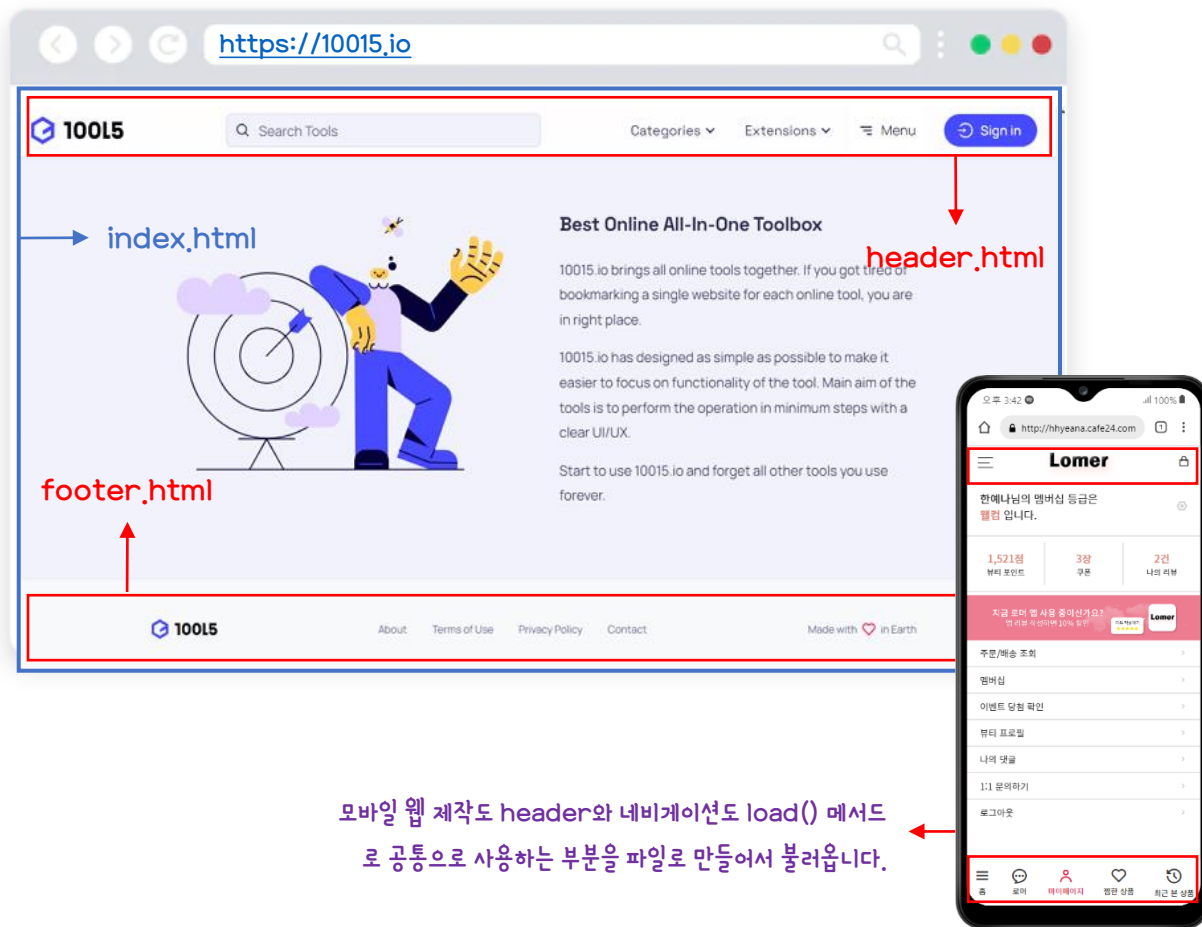
▲ 제이쿼리 load 된 파일 내에 제이쿼리 구문이 있으면 반드시 콜백 함수로 처리해야 작동합니다.

- ③ main-header.html을 로드함과 동시에
- ④ main-header.html 내의 요소에 연결된 제이쿼리 구문도 함께 불러오기 때문에 .trigger에 연결된 제이쿼리는 정상작동 합니다.

점프업(Jump Up)



개별 문서를 페이지의 원하는 곳에 불러오는 load() 메서드 - 실습하기(#1)



모바일 웹 제작도 header와 네비게이션도 load() 메서드로 공통으로 사용하는 부분을 파일로 만들어서 불러옵니다.

[load 메서드]

웹 사이트에서 공통 영역을 따로 파일로 만들어 일정 영역에 불러옴.

공통 영역을 따로 파일로 만들지 않으면 모든 페이지에 매번 같은 html 구조가 들어갑니다. 이러면 매번 같은 작업을 해줘야 하는 것도 있지만 변경 사항이 생기면 파일 개수만큼 고쳐줘야 합니다.

예를 들어 index.html 부터 페이지 수가 50~100개라면 50~100번을 수정해야 합니다. 하지만 load로 공통 영역을 따로 파일로 만들어 일정 영역에 불러오면 공통 영역 파일 한번만 고쳐주면 됩니다. 그래서 header와 footer와 같은 부분은 header.html과 footer.html 로 만들어서 load로 불러옵니다.

[실습을 위한 폴더 구조]

```
LOAD 메서드 연습하기
├── include
│   ├── footer.html
│   └── header.html
├── JS
│   ├── custom.js
│   ├── index.html
│   └── style.css
```

1. load 메서드 연습하기.. 라는 폴더를 만듭니다. 이 폴더를 루트 폴더로 지정하고 VSC로 엽니다.
 2. 폴더 안에 index.html custom.js 파일과 style.css 파일을 만들고 include라는 폴더를 만듭니다.
 3. include 폴더 안에 header.html과 footer.html을 만듭니다.
- ※ 굳이 include라는 폴더를 만들 필요는 없지만 보통 load로 불러와 지는 파일은 한 곳에 모아 두는 것이 좋습니다.

점프업(Jump Up)



개별 문서를 페이지의 원하는 곳에 불러오는 load() 메서드 - 실습하기(#2)

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <title>index.html</title>
  <!-- Custom CSS & JS -->
  <script src="custom.js"></script>
  <link rel="stylesheet" href="style.css">
</head>
<body>
```

```
<div class="container">
```

```
<header>
```

```
<h2>Logo</h2>
```

```
<div class="gnb">
```

```
<a href="#none">Menu 01</a>
```

```
<a href="#none">Menu 02</a>
```

```
<a href="#none">Menu 03</a>
```

```
<a href="#none">Menu 04</a>
```

```
</div>
```

```
</header>
```

```
<section>
```

```
<div class="content">
```

```
<h1>Section Content</h1>
```

```
<p>
```

Lorem ipsum dolor sit amet consectetur adipisicing elit. Earum amet sapiente a veniam commodi! Consequuntur, iusto modi iste quia soluta recusandae illum repudiandae. Beatae corrupti inventore praesentium et distinctio recusandae.

```
</p>
```

```
</div>
```

```
</section>
```

```
<footer>
```

```
<p>Copyright 2023. CodingWorks. All rights reserved.</p>
```

```
</footer>
```

```
</div>
```

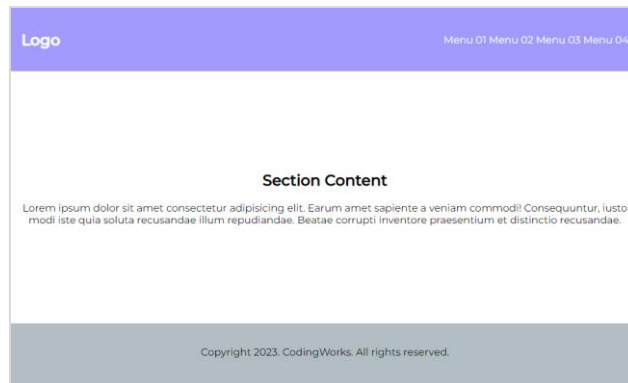
```
</body>
```

```
</html>
```

① index.html의 모든 html 구조를 완성합니다

② CSS로 기본적인 배치를 합니다.

③ header와 footer html을 잘라내서 따로 파일을 만듭니다.



```
<div class="container">
```

```
<section>
```

```
<div class="content">
```

```
<h1>Section Content</h1>
```

```
<p>
```

Lorem ipsum dolor sit amet consectetur adipisicing elit. Earum amet sapiente a veniam commodi! Consequuntur, iusto modi iste quia soluta recusandae illum repudiandae. Beatae corrupti inventore praesentium et distinctio recusandae.

```
</p>
```

```
</div>
```

```
</section>
```

```
</div>
```

③에서 header와 footer html을 잘라내면 헤더와 푸터 부분이 당연히
도 뺄 뚫리게 됩니다.

점프업(Jump Up)

개별 문서를 페이지의 원하는 곳에 불러오는 load() 메서드 - 실습하기(#3)

```
<header>
  <h2>Logo</h2>
  <div class="gnb">
    <a href="#none">Menu 01</a>
    <a href="#none">Menu 02</a>
    <a href="#none">Menu 03</a>
    <a href="#none">Menu 04</a>
  </div>
</header>
```

header.html

```
<footer>
  <p>Copyright 2023. CodingWorks. All rights reserved.</p>
</footer>
```

footer.html

▲ 위의 2개 파일을 index.html 파일에 불러올 곳을 만들기

```
<div class="container">
  <div class="include-header"></div>
  <section>
    <div class="content">
      <h1>Section Content</h1>
      <p>
        Lorem ipsum dolor sit amet consectetur adipisicing elit. Earum amet sapiente a veniam commodi! Consequuntur, iusto modi iste quia soluta recusandae illum repudiandae. Beatae corrupti inventore praesentium et distinctio recusandae.
      </p>
    </div>
  </section>
  <div class="include-footer"></div>
</div>
```

▲ include라는 이름을 가진 곳에 별도로 만든 파일을 load() 메서드로 불러오게 됩니다. 클래스네임은 직관성을 위해서 include라는 이름을 사용하는 것이 좋습니다.

불러와지는 파일은 html은 기본문서구조가 빠진 header와 footer 내용만 있어야 합니다.

[콜백함수]

불러와지는 파일 안에 제이쿼리가 작동하는 부분이 있다면 반드시 콜백함수로 넣어줘야 합니다.

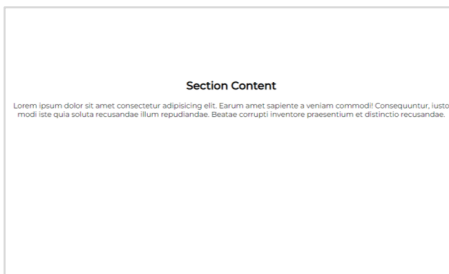
```
$(function(){
  $('.include-header').load('include/header.html', function(){
    // header에 사용된 제이쿼리 구문
  });
  $('.include-footer').load('include/footer.html', function(){
    // footer에 사용된 제이쿼리 구문
  });
});
```

```
$(function(){
  $('.include-header').load('include/header.html');
  $('.include-footer').load('include/footer.html');
});
```

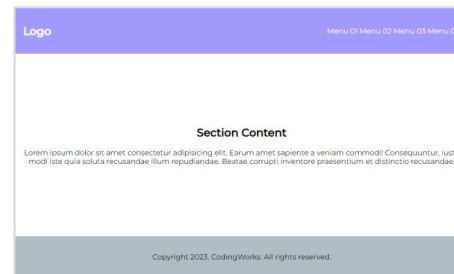
[참고]

load 메서드로 불러와지는 파일은 오프라인 환경에서는 작동하지 않습니다. 반드시 라이브 서버에서 봐야 합니다.

(오프라인 환경이란 파일 탐색기에서 클릭해서 html 오픈하는 것을 말합니다.)



▲ load 메서드로 불러오기 전

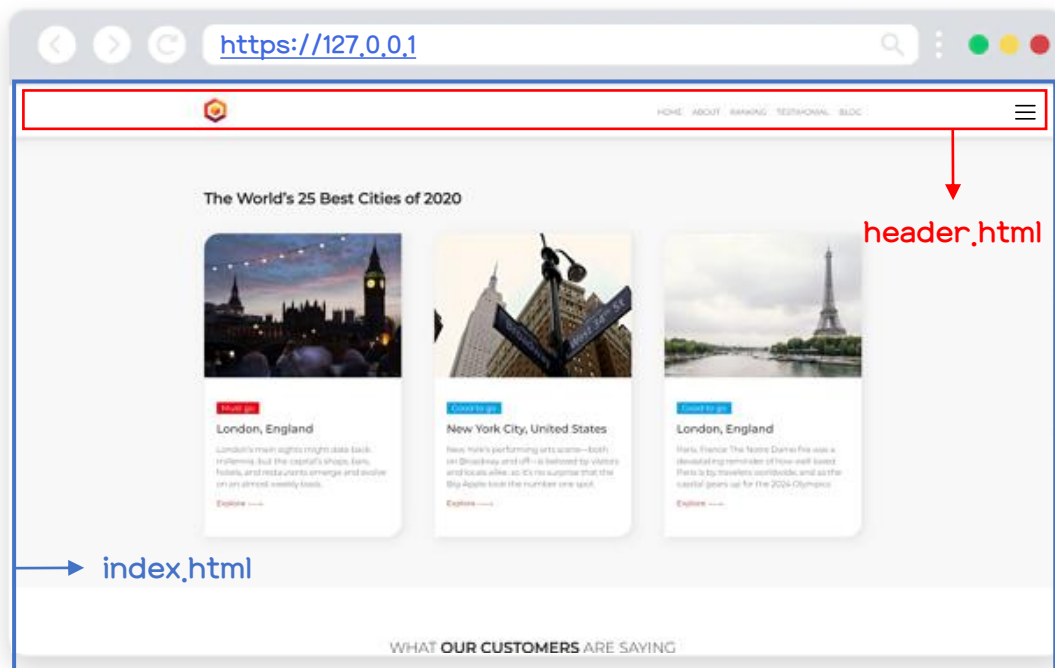


▲ load 메서드로 불러온 후

점프업(Jump Up)



load 메서드로 load된 html 파일 안에 콜백함수 사용하기 - 실습하기(#4)



[load 메서드로 load된 html 파일 안에 제이쿼리 구문이 있는 경우]

- 좌측의 예시처럼 index.html 파일에서 load 메서드로 header.html을 불러온 경우 header.html에 제이쿼리 구문 또는 자바스크립트 구문이 있다면 콜백함수를 사용해야 합니다.
- 예를 들어 header.html에 ≡를 눌러서 사이드바를 열거나 제이쿼리로 뭔가 작동을 시킨다면 반드시 콜백함수를 사용해야 합니다.

// 콜백함수 사용하지 않은 경우

```
$(function(){  
  $('.include-header').load('include/header.html');  
  // header에 사용된 제이쿼리 구문  
});
```

// 콜백함수 사용한 경우

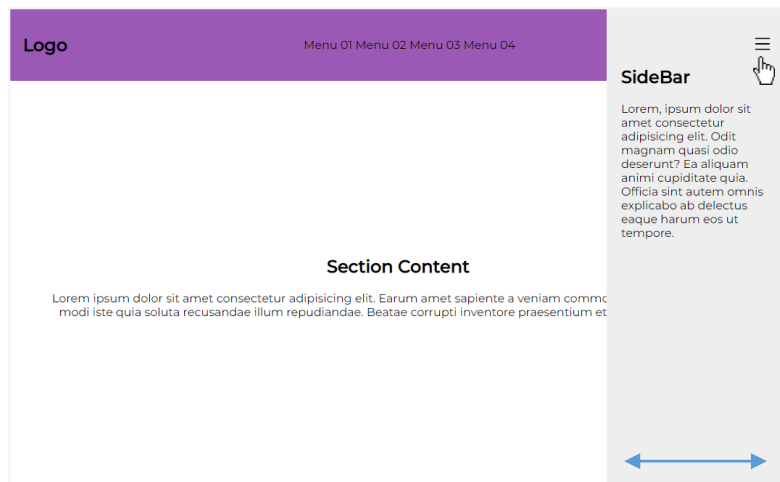
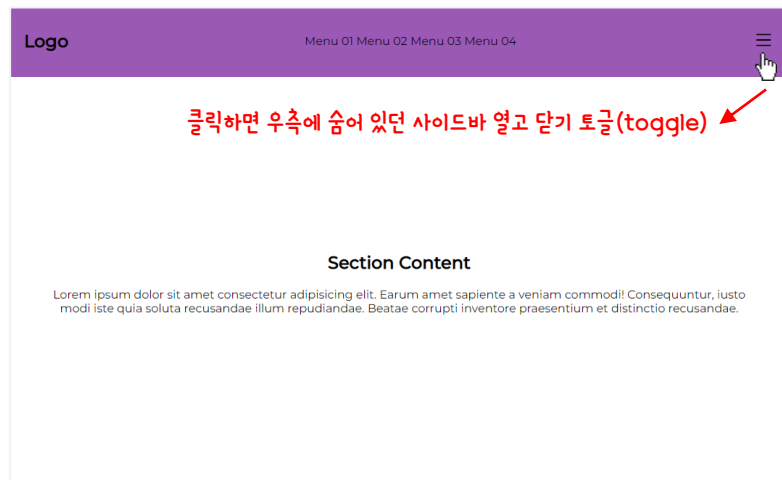
```
$(function(){  
  $('.include-header').load('include/header.html', function(){  
    // header에 사용된 제이쿼리 구문  
  });  
});
```

콜백함수(Callback Function)

점프업(Jump Up)



load 메서드로 load된 html 파일 안에 콜백함수 사용하기 - 실습하기(#5)



```
<!DOCTYPE html>
<html lang="ko">
<head>
  <title>index.html</title>
  <!-- jQuery CDN -->
  <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
  <!-- Custom CSS & JS -->
  <script src="custom.js"></script>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="container">
    <div class="include-header"></div>
    <section>
      <div class="content">
        <h1>Section Content</h1>
        <p>...</p>
      </div>
    </section>
  </div>
</body>
</html>
```

index.html

```
<header>
  <h2>Logo</h2>
  <div class="gnb">
    <a href="#none">Menu 01</a>
    <a href="#none">Menu 02</a>
    <a href="#none">Menu 03</a>
    <a href="#none">Menu 04</a>
  </div>
  <div class="trigger">
    <i class="bi bi-list"></i>
  </div>
</header>
<div class="sidebar">
  <h2>SideBar</h2>
  <p>...</p>
</div>
```

header.html

```
/* Trigger */
.trigger {
  font-size: 30px;
  cursor: pointer;
  position: relative;
  z-index: 1000;
}

/* Sidebar */
.sidebar {
  position: fixed;
  top: 0;
  right: -250px;
  background-color: #eee;
  width: 250px;
  height: 100vh;
  padding: 60px 20px;
  transition: 0.35s;
}

.sidebar.active {
  right: 0;
}
```

CSS

```
// 콜백함수 사용하지 않은 경우
$(function(){
  $('<div>include-header</div>').load('include/header.html');
  $('<div>trigger</div>').click(function(){
    $('<div>sidebar</div>').toggleClass('active');
  });
});
```

해당 제이쿼리 구문은 header.html 내에 있는 요소를 제어하는 코드로 콜백함수로 사용하지 않으면 작동하지 않습니다.

```
// 콜백함수 사용한 경우
$(function(){
  $('<div>include-header</div>').load('include/header.html', function(){
    $('<div>trigger</div>').click(function(){
      $('<div>sidebar</div>').toggleClass('active');
    });
  });
});
```

header.html을 불러오고 load 메서드가 종료되지 않은 상태에서 header.html에 사용된 제이쿼리 구문을 콜백함수로 실행하므로 정상적으로 작동합니다.