Team. 자바스(JAVAS)

기술서

작성자: 이지은

① Swiper.js란?

- swiper.js는 편리하고 강력한 슬라이더를 구현하는 라이브러리로 모바일도 지원하며, jqurey와 같은 별도의 프레임워크가 필요 없다.

② 한 페이지에 swiper가 2개 이상 들어가는 경우

- 서브페이지를 퍼블리싱 하던 도중 한 페이지에 swiper가 2개가 들어가는 경우가 있었다. swiper가 한 개만 들어갈 때와 2개 이상이 들어갈 때의 차이점이 발생해서 한 페이지에 swiper가 2개 이상이 필요한 경우에는 어떻게 적용해야 할지 기술해보려고 한다.

③ Swiper가 하나일 때

- swiper가 제대로 작동되기 위해 CDN방식으로 css파일과 js파일을 연결해 주어야 한다.
※ 참고사항 : npm으로 다운받아 사용도 가능하다.

```
<link
  rel="stylesheet"
  href="https://cdn.jsdelivr.net/npm/swiper@10/swiper-bundle.min.css"
/>

<script src="https://cdn.jsdelivr.net/npm/swiper@10/swiper-bundle.min.js"></script>
```

- html의 기본 구조는 다음과 같다. 필요한 슬라이드 개수만큼 자유롭게 추가가 가능하다.
※ 주의사항 : swiper-container, swiper-wrapper, swiper-slide 클래스는 임의로 변경할 수 없다.

```
<!-- Slider main container -->
<div class="swiper">
  <!-- Additional required wrapper -->
  <div class="swiper-wrapper">
    <!-- Slides -->
    <div class="swiper-slide">Slide 1</div>
    <div class="swiper-slide">Slide 2</div>
    <div class="swiper-slide">Slide 3</div>
    ...
  </div>
  <!-- If we need pagination -->
  <div class="swiper-pagination"></div>

  <!-- If we need navigation buttons -->
  <div class="swiper-button-prev"></div>
  <div class="swiper-button-next"></div>

  <!-- If we need scrollbar -->
  <div class="swiper-scrollbar"></div>
</div>
```

- JS파일에 작성해야 할 내용은 다음과 같다.

```
const swiper = new Swiper('.swiper', {
  // Optional parameters
  direction: 'vertical',
  loop: true,

  // If we need pagination
  pagination: {
    el: '.swiper-pagination',
  },

  // Navigation arrows
  navigation: {
    nextEl: '.swiper-button-next',
    prevEl: '.swiper-button-prev',
  },

  // And if we need scrollbar
  scrollbar: {
    el: '.swiper-scrollbar',
  },
});
```

※ 참고사항 : swiper.js의 옵션은 다양하게 존재하며 사용자가 원하는대로 설정이 가능하다.

- swiper.js 기본 옵션들

옵션	값	설명
direction	'horizontal' (기본값) 'vertical'	슬라이드 방향 지정
autoplay	boolean 또는 object	true: 자동 슬라이드 설정 object: 슬라이드 옵션 추가
loop	boolean	true: 무한 반복 슬라이드 false: 무한 반복하지 않음
delay	number	자동 슬라이드 시, 한 슬라이드에 머무르는 시간(ms) 기본값 : 300
speed	number	슬라이드 속도 기본값 : 300
slidePerView	number 또는 'auto'	한 번에 몇 개의 슬라이드를 보여줄지 지정

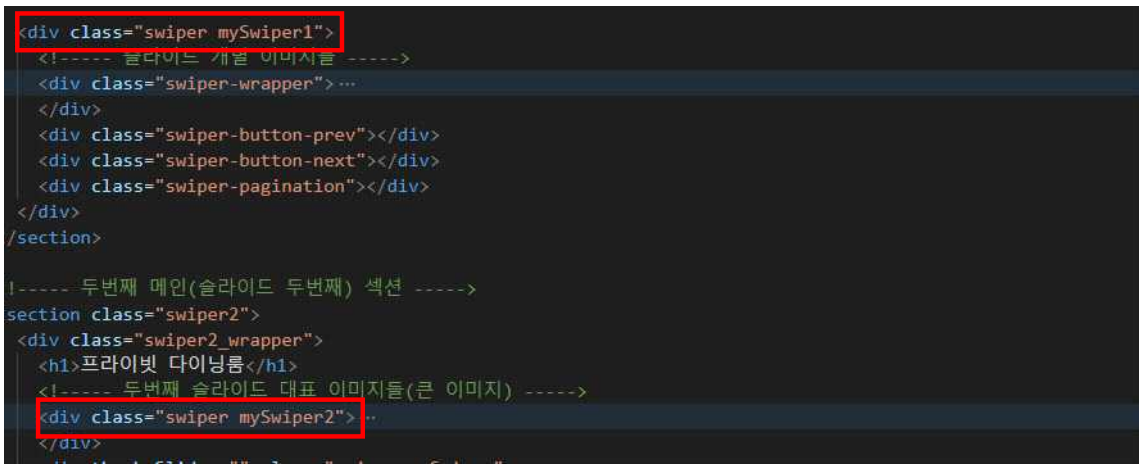
pagination	{
	el: '.swiper-pagination',
	clickable: true
	}
		페이징을 클릭하면 해당
		영역으로 이동

navigation	{
	prevEl:
	'.swiper-button-prev',
	nextEl: '.swiper-button-next'
	}
		네비게이션 설정
		이전 화살표 버튼과 다음
		화살표 버튼

※ 참고사항 : 더 많은 추가 옵션은 https://swiperjs.com/swiper-api 에서 확인 가능하다.

④ Swiper가 두 개인 경우

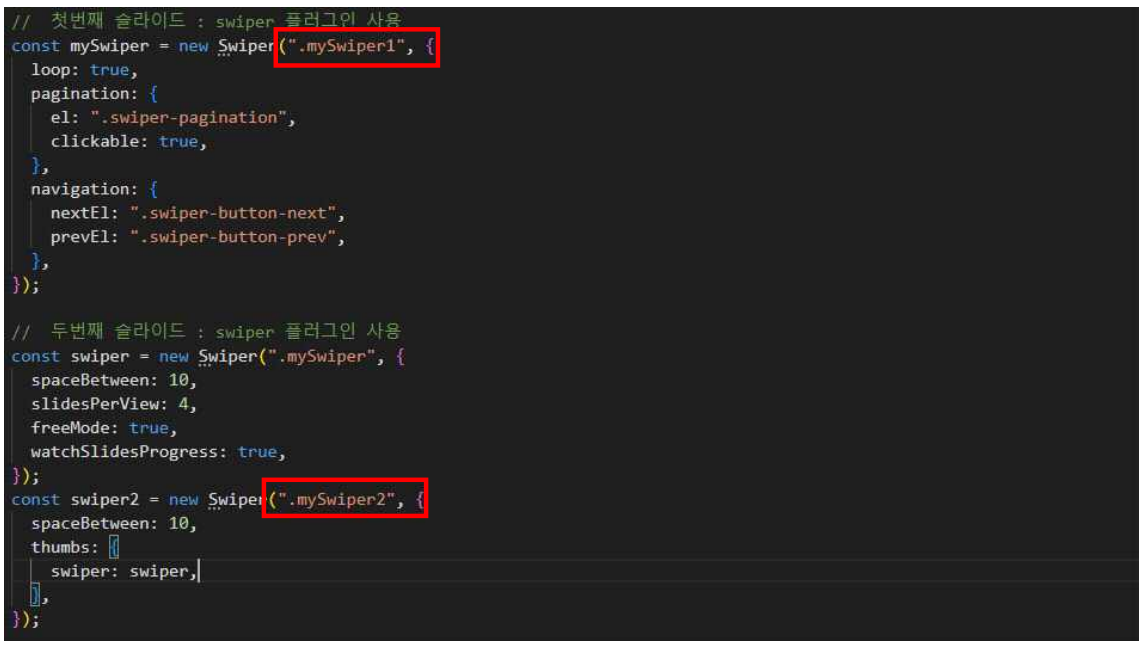
- 서브페이지를 퍼블리싱 하면서 한 페이지에 swiper가 2개가 들어가서 어떻게 해야 할지 고민이 많았는데 방법은 어렵지 않았다.



```
<div class="swiper mySwiper1">
  <!----- 슬라이드 개별 이미지들 ----->
  <div class="swiper-wrapper">···
  </div>
  <div class="swiper-button-prev"></div>
  <div class="swiper-button-next"></div>
  <div class="swiper-pagination"></div>
</div>
</section>

!----- 두번째 메인(슬라이드 두번째) 섹션 ----->
section class="swiper2">
<div class="swiper2_wrapper">
  <h1>프라이빗 다이닝룸</h1>
  <!----- 두번째 슬라이드 대표 이미지들(큰 이미지) ----->
  <div class="swiper mySwiper2">···
  </div>
```

위에서 설명한 swiper.js가 제공하는 기본 html 구조는 똑같이 작성하되, 클래스명을 각각 다르게 부여해 주면 된다. 나는 기본 구조인 ** **에 첫 번째 슬라이드에는 **mySwiper1,** 두 번째 슬라이드에는 **mySwiper2**라는 클래스명을 부여해 주었더니 문제없이 작동되었다.



```
// 첫번째 슬라이드 : swiper 플러그인 사용
const mySwiper = new Swiper(".mySwiper1", {
  loop: true,
  pagination: {
    el: ".swiper-pagination",
    clickable: true,
  },
  navigation: {
    nextEl: ".swiper-button-next",
    prevEl: ".swiper-button-prev",
  },
});

// 두번째 슬라이드 : swiper 플러그인 사용
const swiper = new Swiper(".mySwiper", {
  spaceBetween: 10,
  slidesPerView: 4,
  freeMode: true,
  watchSlidesProgress: true,
});
const swiper2 = new Swiper(".mySwiper2", {
  spaceBetween: 10,
  thumbs: {
    swiper: swiper,
  },
});
```

-js에서도 html에서 내가 부여한 클래스명으로 작성해 주면 된다.

출처: https://swiperjs.com/