

Hello,

KDT 웹 개발자 양성 프로젝트

1기! 12th

with





JS







DOM

(Document Object Model)



querySelector



getElementById



classList



AddEventListener



setAttribute



textContent



querySelectorAll



과제 리뷰 + DOM



Todo list

HTML 구성하기



```
<body>
  <div class="header">
    <h1>Tetz Todo List</h1>
  </div>
  <div class="contents">
    <div class="input-part">
      <input type="text" class="input-task" />
      <input type="button" class="input-btn" value="추가" />
    </div>
    <div class="list-part">
      <ul class="todo-list"></ul>
    </div>
  </div>
</body>
```

CSS 간단하게



```
* {  
  margin: 0;  
}  
  
ul {  
  list-style: none;  
  padding-left: 0;  
}  
  
.header {  
  background-color: orange;  
  height: 60px;  
}  
  
.header h1 {  
  text-align: center;  
  line-height: 60px;  
}
```

```
.contents {  
  display: flex;  
  flex-direction: column;  
  align-items: center;  
}  
  
.input-part {  
  padding: 30px 0px;  
}  
  
.list-part .todo-list {  
  text-align: center;  
}
```

JS 시작!



- 기본적으로 사용할 요소들 미리 호출
 - 할 일을 입력하는 입력창
 - 추가 버튼
 - Todo-list
- 추가 버튼에 AddEventListener 걸기
 - 입력이 없으면? → 내용을 입력하라고 할 것!
 - 입력이 있으면 → Task 추가



추가할 요소들 순차적으로 선언!

- Task
 - Li 요소로 추가하기
 - `Document.createElement("태그 이름");`
- 완료 여부를 나타낼 Check-box
 - Check-box 체크하면? → task 에 line-through 효과 주기
 - Check-box 체크를 풀면? → task 에 line-through 효과 빼기

추가할 요소들 순차적으로 선언!



- Task 삭제 버튼
 - 삭제 버튼 추가를 위한 요소 선언하기
- Task 삭제 방법 2가지
 - Onclick + this(먼저 해당 방법으로 시작!)
 - e.target



Li 요소로 추가!

- 그런데 추가가 이상하게 되네요!?
- (요소).textContent / (요소).innerHTML
 - 요소가 초기화 됩니다! 따라서 지금은 쓰면 안되겠죠?
- (요소).append 또는 appendChild 사용
- 추가 되는지 확인해 볼까요?

append() 와 appendChild() 공통점



- 붙여 넣는 공간은 동일 → 부모 노드의 제일 마지막 자식 요소로 추가

append()



- append() 는 Node 와 String 을 전부 추가 할 수 있어요
- append() 는 여러 가지 값을 한번에 붙일 수 있어요
- append() 는 반환(리턴) 값이 없어요!

appendChild()



- appendChild() 는 Node 만 추가할 수 있어요
- appendChild() 는 한 번에 하나만 추가할 수 있어요
- appendChild() 는 추가한 Node 를 반환(리턴) 합니다!

prepend()



- `prepend()` 는 `append()` 와는 반대로 부모 노드의 첫번째 요소로 추가합니다!
- 단, `prependChild()` 는 존재 하지 않아요!



삭제 기능 구현, onclick + this

- 클릭 된 요소를 제거하는 방법 중 첫번째 입니다!
- Onclick 에서 넘어온 this 를 매개변수로 하는 함수 선언
- This 의 값을 출력해서 확인
- This 의 부모 요소를 삭제하면 완료!

```
function deleteTask(t) {  
  t.parentNode.remove();  
}
```




삭제 기능 구현, e.target

- 클릭 된 요소를 제거하는 방법 중 두번째 입니다!
- 부모 요소에 click 으로 addEventListener 를 걸고 익명 함수의 매개 변수로 e(event)를 전달하는 방법
- E.target 으로 this 와 같은 효과를 냅니다!
- 단, 해당 target 의 부모 요소를 지워버리면 글자 또는 check 박스를 클릭해소 삭제 되므로 처리 필요
- TagName / 속성 값 확인 후 제거

새롭게 사용한 기능 목록



- `Document.createElement();`
- `(요소).append / appendChild`
- `OnClick` 속성
- `Html` 문서에서의 `this`
- `Event` 객체!





jQuery

자바스크립트 언어를 간편하게 사용할 수 있도록
단순화시킨 오픈 소스 기반의
자바스크립트 라이브러리

프레임 워크란?

특정 프로그램 개발의 시작부터 끝까지
필요한 모든 기능을 제공하는 도구 및
자원들의 묶음

라이브러리란?

자주 사용하는 기능들을 모아두었
다가 필요할 때 가져다 쓸 수 있도록
한 자원들의 묶음

Jquery 장점



- 사실, JS가 DOM 관련 기능이 구릴 때 정말 압도적으로 편리 했었다!
- 대부분의 브라우저에서 지원 가능
- HTML DOM를 손쉽게 조작 가능
- CSS 스타일도 간단히 적용 가능
- 애니메이션 효과나 대화형 처리를 간단하게 적용
- 같은 동작을 더욱 짧은 코드로 구현 가능
- 오픈 라이선스를 적용해 누구나 자유롭게 이용 가능

Jquery 사용하기



- 파일 다운로드
- CDN(Content Delivery Network) 사용
- <https://releases.jquery.com/>

Code Integration

```
<script  
  src="https://code.jquery.com/jquery-3.6.0.slim.js"  
  integrity="sha256-HwWONEZrpuoh951cQD1ov2HUK5zA5DwJ1DNUXaM6FsY="  
  crossorigin="anonymous"></script>
```



The `integrity` and `corsorigin` attributes are used for [Subresource Integrity \(SRI\) checking](#). This allows browsers to ensure that resources hosted on third-party servers have not been tampered with. Use of SRI is recommended as a best-practice, whenever libraries are loaded from a third-party source. Read more at srihash.org

JQuery 기초



```
$ ( 선택자 ) . 동작함수 ( ) ;
```



값 가져오기 & 설정하기

- `$(선택자).val();`
 - Input 과 같은 value 값을 가져올 수 있다.

```
var value = $("input").val();
```

- `$(선택자).val("설정할 값");`
 - Input 과 같은 곳에 value 값을 설정할 수 있다.

```
$("input").val("설정완료!");
```




값 가져오기 & 설정하기

- `$(선택자).val();`

- Input 과 같은 value 값을 가져올 수 있다.

```
var value = $("input").val();
```

→ `document.querySelector("선택자").value;`

- `$(선택자).val("설정할 값");`

- Input 과 같은 곳에 value 값을 설정할 수 있다.

```
$("input").val("설정완료!");
```

→ `document.querySelector("선택자").value = "원하는 값";`

Text 변경하기



- `$(선택자).text("글자");`

```
$("span").text("jQuery로 바꿨습니다.");
```

```
var span = document.querySelector("span");  
span.innerText = "js로 바꿨습니다."
```

실습



- Text 타입 input 하나, 버튼을 2개(출력, 변경), span 1개 만들기
- 변경을 누르면 input 의 value 값을 "가나다라마바사"로 만들기
- 출력을 누르면 input 에 입력한 내용이 span 의 내용으로 적용되도록 만들기
- 실습 내용을 JS와 JQ로 구현하기!



속성 변경하기

- `$(선택자).attr("속성", "속성값");`

```
$("#li2").attr("style", "color: red");
```

→ `document.querySelector("선택자").setAttribute("속성", "값");`

실습



- Text 타입 input 하나, 버튼을 2개(변경1, 변경2) 만들기
- 변경1을 누르면 text 타입 input 의 placeholder 를 “변경1을 누르셨습니다” 로 변경
- 변경2를 누르면 text 타입 input 을 radio 버튼으로 변경하기
- 실습 내용을 JS와 JQ로 구현하기!

CSS 변경하기



- `$(선택자).css("속성" , "속성값");`

```
$("#span").css("font-size", "28px");  
$("#span").css("color", "blue");
```

```
var span = document.querySelector("span");  
span.style = "font-size: 10px; color: red";
```

실습



- 100 x 100 div 요소 하나, 버튼을 2개(변경1, 변경2) 만들기
- 버튼을 누르면 div 요소의 색상을 오렌지로 변경
- 변경1은 JS 로 구현
- 변경2는 JQ 로 구현

HTML 변경하기



- \$(선택자).html(html);

```
$("#p").html("<h1>jQuery</h1>");
```

```
var p = document.querySelector("p");  
p.innerHTML = "<h3>javascript</h3>"
```


실습



- P 태그로 쓰여진 가나다라마바사, 변경 버튼 만들기
- 버튼을 누르면 p 태그 가나다라마바사를 h1 태그 가나다라마바사로 변경
- JS 와 JQ 로 각각 구현

요소 추가



- `$(선택자).append(요소);`
 - 선택된 요소의 자식 요소 제일 마지막 요소로 추가
 - JS의 append 와 동일
- `$(선택자).prepend(요소);`
 - 선택된 요소의 자식 요소 중 첫번째로 추가
 - JS의 prepend 와 동일
- `$(선택자).before(요소);`
 - 선택된 요소의 이전 형제 요소로 추가
 - JS 에는 없음
- `$(선택자).after(요소);`
 - 선택된 요소의 다음 형제 요소로 추가
 - JS 에는 없음

실습



- 텍스트 인풋 / 버튼 / ul 태그 (li 자식을 4개 가지도록 만들기)
- 버튼 클릭 시, 텍스트 인풋의 내용을
- 태그의 마지막 자식으로 삽입
- 태그의 첫번째 자식으로 삽입
- 위의 기능을 JS, JQ로 구현
- 또 다른 태그의 형제로 삽입(앞, 뒤)
 - 이 것은 JQ 로만 구현하기



요소 삭제

- `$(선택자).remove();`
 - 선택한 요소를 DOM 트리에서 삭제
 - **인자로 선택자를 넘겨주면**, 선택한 요소 중 조건을 만족하는 요소만 삭제 → **JS**에서는 지원하지 않는 기능
- `$(선택자). empty();`
 - 선택한 요소의 자식 요소를 모두 삭제
 - 선택된 요소는 남아 있고 그 안이 다 비워진다
 - JS 에는 없는 기능
 - 단, `(요소).innerHTML = ""` / `(요소).textContent = ""`; 로 동일하게 처리 가능



요소 탐색

- `$(선택자).parent()`
 - 바로 위에 존재하는 하나의 부모 요소 선택
 - JS에서는 `(요소).parentNode`
- `$(선택자).parents()`
 - 바로 위 부모뿐만 아니라 부모의 부모, 모든 조상을 선택
 - JS에서는 없음, `closest("선택자")` 로 가까운 조상 찾기는 가능
- `$(선택자).children()`
 - 선택된 요소의 자식 요소 모두 선택
 - JS에서는 `(요소).childNodes` → 비 노드 요소 포함
 - JS에서는 `(요소).children` → 비 노드 요소 제외

요소 탐색



- **\$(선택자).next()**
 - 선택된 요소 바로 다음에 위치한 형제 요소 선택
 - JS에서는 (요소).nextSibling
- **\$(선택자).prev()**
 - 선택된 요소 바로 이전에 위치한 형제 요소 선택
 - JS에서는 (요소).previousSibling



클래스 추가, 삭제, 확인

- `$(선택자).addClass("클래스명")`
 - 선택된 요소에 클래스 추가
- `$(선택자).removeClass("클래스명")`
 - 선택된 요소에 클래스 삭제
- `$(선택자).hasClass("클래스명")`
 - 선택된 요소에 클래스가 있는지 여부 확인
 - Return 값은 true 와 false 로 나온다.

종합 실습

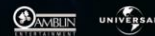


- 주어진 HTML 문서에 아래의 기능을 추가!
 - 해당 날짜를 누르면 날짜가 날짜 입력 창에 자동으로 입력
 - 내용을 입력하고 작성 버튼을 누르면 해당 날짜에 작성 내용이 div 요소로 추가 되도록 만들기
- + 작성 내용을 클릭하면 삭제 되는 기능 추가!

미래의 그날이 온다!



이 영화는 유나이티드 아츠스 센터에서 제작되었습니다.
A Universal Picture © 1985 & 2015 Universal Studios.







JS





JS 파일 연결하기

Main.js 파일 연결하기!



- 안전하게 속성 값으로 defer 넣고 시작!

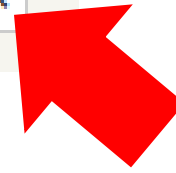


서브메뉴

Search



[Sign In](#) | [My Starbucks](#) | [Customer Service & Ideas](#) | [Find a Store](#)





서브 메뉴, search 변경하기

- 기존은 CSS 의 :focus 를 사용하여 구현
- 따라서, 돋보기를 피해서 input 을 클릭 했을 때에만 작동 → 불편
- 돋보기를 클릭하면 기능이 동작하도록 변경
- 돋보기를 클릭하면 focus 가 가도록 설정하여 기존의 CSS 사용
- 포커스 시 → “통합 검색” 이라는 placeholder 도 추가!(by JS)



공지 사항



공지사항 스타벅스 e-Gift Card 구매 가능 금액 안내



공지 사항, 슬라이드 메뉴 적용하기



- 해당 기능을 전부 구현하는 것은 매우 어렵고 귀찮은 작업이죠?
- 이럴 때 쓰는 것이 바로! Library
- Swiper 라는 라이브러리를 사용해 봅시다!




Swiper

The Most Modern Mobile
Touch Slider

[Get Started](#) [API](#) [Demos](#) [React](#) [Solid](#) [Svelte](#) [Vue](#) [Angular](#)

MIT Licensed, v8.3.1 released on July 13, 2022 [Changelog](#)

 31,724 stars

Top Notch Features

- ✓ Library Agnostic
- ✓ Flexbox Layout
- ✓ Multi Row Slides Layout
- ✓ Two-way Control
- ✓ Rich API
- ✓ Parallax Transitions
- ✓ Virtual Slides
- ✓ Mutation Observer
- ✓ Full True RTL Support
- ✓ 3D Effects
- ✓ Full Navigation Control
- ✓ Most Flexible Slides Layout Grid
- ✓ Images Lazy Loading
- ✓ And many more

<https://swiperjs.com/>




Use Swiper from CDN

If you don't want to include Swiper files in your project, you may use it from CDN. The following files are available:

```
<link
  rel="stylesheet"
  href="https://unpkg.com/swiper@8/swiper-bundle.min.css"
/>

<script src="https://unpkg.com/swiper@8/swiper-bundle.min.js"></script>
```

A large red arrow pointing from the right towards the script tag in the code block above.

If you use ES modules in browser, there is a CDN version for that too:

```
<script type="module">
  import Swiper from 'https://unpkg.com/swiper@8/swiper-bundle.esm.browser.min.js'

  const swiper = new Swiper(...)
</script>
```

Download assets

If you want to use Swiper assets locally, you can directly download them from <https://unpkg.com/swiper@8/>



Add Swiper HTML Layout

Now, we need to add basic Swiper layout to our app:

```
<!-- 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>
```



Swiper CSS Styles/Size

In addition to [Swiper's CSS styles](#), we may need to add some custom styles to set Swiper size:

```
.swiper {  
  width: 600px;  
  height: 300px;  
}
```



Initialize Swiper

Finally, we need to initialize Swiper in JS:

```
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',  
  },  
});
```


Swiper 라이브러리 추가



```
<!-- Icons -->
<link rel="stylesheet"
      href="https://fonts.googleapis.com/css2?family=Material+Symbols+Outlined:opsz,wght,FILL,GRAD@20..48,100..700,0..1,-50..200" />

<!-- SWIPER -->
<link rel="stylesheet" href="https://unpkg.com/swiper@8/swiper-bundle.min.css" />
<script src="https://unpkg.com/swiper@8/swiper-bundle.min.js"></script>

<!-- main -->
<link rel="stylesheet" href="./css/main.css">
<script defer src="./js/main.js"></script>
```

HTML 코드 추가!



- Swiper 클래스 div 추가
- Swiper-wrapper div 자식으로 swiper-slide div 추가



```
<div class="inner">
  <div class="inner__left">
    <h1>공지사항</h1>
    <div class="swiper">
      <div class="swiper-wrapper">
        <div class="swiper-slide">My DT Pass 이용 안내</div>
        <div class="swiper-slide">My DT Pass 관련 서비스 점검 안내</div>
        <div class="swiper-slide">시스템 개선 및 점검 안내</div>
        <div class="swiper-slide">전자 영수증 서비스 서비스 점검 안내</div>
        <div class="swiper-slide">스타벅스 e-Gift Card 구매 가능 금액 안내</div>
      </div>
    </div>
    <a href="#"><span class="material-symbols-outlined">add_circle</span></a>
  </div>
  <div class="inner__right">
    <h1>스타벅스 프로모션</h1>
    <a href="#"><span class="material-symbols-outlined">expand_circle_down</span></a>
  </div>
</div>
```

JS에 Swiper 생성자 함수 추가



- 변수 하나를 만들고 Swiper 생성자 함수 추가
- 설정 값 입력
 - Direction: "vertical"
 - Autoplay: true
 - Loop: true

CSS 수정



- Swiper 클래스의 CSS 수정
- 크기 값이 필요 → 높이 값 주기!
- 아이템 중앙 정렬!
 - 한 줄 텍스트는??





프로모션

슬라이드



공지사항 시스템 개선 및 점검 안내



스타벅스 프로모션



스타벅스와 함께
가하는 것을 즐기는 나만의 여름!

기간 : 2022년 6월 14일(화) - 7월 25일(월)

자세히 보기

STARBUCKS RESERVE™ SUMMER

스타벅스 리저브 서머 원두 & 커피

기간 : 2022년 4월 5일 ~ 2022년 8월 25일

자세히 보기



회원 계정에 등록된 스타벅스 카드
1만원당 별★1개를 즉시 추가로

기간 : 2022년 1월 1일 ~ 2022년 12월

자세히 보기

HTML 구성하기



- Promotion 클래스 div 선언
- Swiper 사용을 위한 swiper 클래스 div 추가
- Swiper-wapper 추가
- Swiper-slide 추가

CSS 선언해 놓기!



- `.notice .promotion {}`
- `.notice .promotion .swiper {}`
- `.notice .promotion .swiper .swiper-wrapper {}`
- `.notice .promotion .swiper .swiper-wrapper .swiper-slide {}`

Promotion / Swiper 크기 세팅



- 스타벅스 페이지를 참고
- Promotion
 - Height : 658px / background-color: #f6f5ef
- Swiper
 - Height : 553px / Width: $\text{calc}(819\text{px} * 3 + 20\text{px})$

중앙 정렬!



- `Text-align: center;` 는 항상 중앙을 가르킬까요?
- 화면 확대 축소에 따른 중앙 정렬 확인!
- Container 의 크기에 따른 중앙 정렬 → `left: 50% / translate(-50%, 0);`

이미지 삽입하기!



- Swiper slide 에 이미지 삽입하기
- 자세히 보기 버튼(black) 추가!



```
<div class="promotion">
  <div class="swiper">
    <div class="swiper-wrapper">
      <div class="swiper-slide">
        
        <a href="#" class="btn btn--black">자세히 보기</a>
      </div>
      <div class="swiper-slide">
        
        <a href="#" class="btn btn--black">자세히 보기</a>
      </div>
      <div class="swiper-slide">
        
        <a href="#" class="btn btn--black">자세히 보기</a>
      </div>
    </div>
  </div>
</div>
```

Swiper 생성자 함수 추가!



- 새로운 swiperPromotion 변수에 Swiper 생성자 추가
- Direction / slidePerView / spaceBetween / centeredSlide / loop / autoplay 옵션 추가하기!



```
// SWIPER PROMOTION
const swiperPromotion = new Swiper(".promotion .swiper", {
  direction: "horizontal", // 기본 값
  slidesPerView: 3, // 한번에 보여줄
  spaceBetween: 10, // 아이템간 거리
  centeredSlides: true, // 슬라이드 센터 여부
  loop: true, // 루프 여부
  autoplay: {
    // 자동 재생, 변경 시간 설정
    delay: 5000,
  }
});
```


CSS 수정하기!



- Swiper 배경색 제거
- Swiper 동작을 확인하기
- 보여지는 슬라이드의 클래스명 확인(active)
- 양 옆 슬라이드 반투명 처리
- 자세히 보기 버튼 위치 및 크기 조절

페이지네이션 / 이동 버튼 추가하기



- Swiper 에서 제공하는 pagination / prev / next 버튼 추가하기

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

페이지네이션 / 이동 버튼 추가하기



- Swiper 에서 제공하는 pagination / prev / next 버튼 추가하기

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

- Swiper 생성자에 옵션 추가하기!



```
// SWIPER PROMOTION
const swiperPromotion = new Swiper(".promotion .swiper", {
  direction: "horizontal", // 기본 값
  slidesPerView: 3, // 한번에 보여줄
  spaceBetween: 10, // 아이템간 거리
  centeredSlides: true, // 슬라이드 센터 여부
  loop: true, // 루프 여부
  autoplay: {
    // 자동 재생, 변경 시간 설정
    delay: 5000,
    disableOnInteraction: false,
  },
  pagination: {
    el: ".promotion .swiper-pagination", // pagination을 할 엘리먼트 클래스 설정
    clickable: true, // 클릭 가능 여부 설정
  },
  navigation: {
    prevEl: ".promotion .swiper-button-prev", // 이전 버튼 클래스 설정
    nextEl: ".promotion .swiper-button-next", // 이후 버튼 클래스 설정
  },
});
```

페이지네이션 CSS 수정



- Pagination 동작 및 클래스 확인하기
- 각각 bullet 크기 설정
- 선택 효과 설정
 - 백그라운드 이미지 조절
 - 백그라운드 컬러 제거 → 투명하게 처리



이동 버튼 CSS 수정

- 버튼 위치부터 확인 → `position: absolute; / top: 300px;`
- 버튼 스타일 수정
 - `Width / height : 55px;`
 - `Border: 2px solid #333;`
 - `Color: #333;`
 - `Border-radius: 50%;`
 - `Cursor: pointer;`
- 화살표 크기도 수정하기 → `after` 의 폰트 사이즈 수정



이동 버튼 위치 지정

- Position: absolute; 는 이미 지정된 상태!
- Left: 400px? / right: 400px?
- 반응형 확인하기!
- Left: 50% / translate(-550px, 0);
- Right: 50% / translate(550px, 0);
- Promotion 크기를 기반으로 배치하기 → 단 일정 크기 부터는 반응형 필요!

Promotion 에 overflow 지정



- Swiper 의 크기로 인해 가로 스크롤 발생!
- Promotion 에 overflow: hidden; 설정으로 스크롤 없애기!

Autoplay?



- Pagination 에 재생, 멈춤 버튼을 만드는 것은 고통스러우니 공지사항의 + 버튼에 autoplay 멈춤, 재시작 기능을 넣어 봅시다!
- <a> 태그에 onclick 주기!
- Js 파일에 함수 생성
- Swiper의 autoplay 관련 함수
 - .autoplay.start() / .autoplay.stop()
- Autoplay.stop() 적용 → 어? 그런데 안먹네요?

Autoplay?



- <a> 태그의 특성 상 href 로 인해 페이지를 새로 고침
- 이를 해결하기 위해 href="javascript:함수명();" 을 사용
- A 태그의 효과는 쓰고 싶지만, 페이지 새로 고침을 막으려면
 - 사용

```
<a href="javascript:stopAutoPlay();" >  
  <span class="material-symbols-outlined">add_circle</span>  
</a>
```

Autoplay?



- swiperPromotion 의 값을 console.log 로 찍어보기
- swiperPromotion.autoplay.running 값에 따라 stop / start 여부 결정!

Promotion Toggle



- Promotion section 토글 기능 추가!
- Toggle 용 버튼에 클래스 주기!
 - Javascript:void(0); 처리
- JS 에서 버튼에 addEventListener 추가
 - Promotion 에 hide 클래스 추가 / 삭제 처리
- CSS 에서 promotion 처리

Promotion Toggle 버튼 애니메이션!



- Promotion section 토글 버튼 애니메이션 추가하기!
- 보여 졌을 때 180deg 돌아가도록 처리하기!
- JS, CSS 수정!



스크롤에 따른
애니메이션 처리!



Scroll 발생을 체크

- 브라우저 레벨에서 발생하는 개념이므로
- document → window 사용!

```
// SCROLL
let scrollYpos;
window.addEventListener("scroll", function () {
  scrollYpos = window.scrollY;
  console.log(scrollYpos);
});
```

- 콘솔 로그에서 확인하기!



페이지가 로드되면 바로!

- Visual 의 애니메이션 재생 필요
- Visual 의 inner 클래스에 :hover 로 처리 했던 것으로 클래스로 변경 → :hover → animate
- 페이지가 로드 되면 바로 실행되는 함수
 - Window.onload() 사용

```
window.onload = () => {  
  const visualInner = document.querySelector(".visual .inner");  
  visualInner.classList.add("animate");  
};
```




페루 애니메이션!

- 스크롤 위치 찾기! → 300 정도가 적당해 보이네요!
- Hover 로 처리해 두었던 애니메이션을 animate 클래스로 변경!
- JS 처리

```
if (scrollYpos > 300) {  
    const peruAnimate = document.querySelector(".peru");  
    peruAnimate.classList.add("animate");  
}
```



**JUST
DID
IT.**

