

기타 고려 사항

normalize.css ?

- 브라우저마다 조금씩 다른 기본 태그, 글씨체, .. 등을 모두 같은 환경을 만들어주기 위한 CSS
- CDN 링크로 사용하거나, 다운로드 받아서 직접 normalize.css 를 만들어서 사용할 수 있어요

normalize CDN & Downloads

- <https://cdnjs.com/libraries/normalize>
- <https://necolas.github.io/normalize.css>

reset.css ?

- normalize와 마찬가지로 브라우저마다 다른 기본 스타일을 똑같이 보여주기 위해서 사용
- 완전히 초기화함

reset.css Downloads

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

CSS, 가상 클래스 :root

- 웹 문서에서 가장 상위 태그에 해당하는 태그를 의미합니다
- 어떤 태그를 의미하는 걸까요?
- 문서에서 쓰일 기본적인 스타일(페이지 메인 색상, 글씨 색상, 폰트 사이즈, 배경 색상 등을 지정할 수 있습니다!)

사용 방법

- root.css

```
:root {  
  /* color */  
  --main-red: #dc3545;  
  
  /* size */  
  --base-space: 8px;  
}
```

- :root 가상클래스 선택자 내부에 변수를 만들어서 CSS 값을 넣을 수 있어요!

- Style.css

```
@import url(./root.css);  
  
h1 {  
  color: var(--main-red);  
}
```

- 지정해둔 CSS 변수를 사용하기 위해서 var() 메소드를 통해 색을 지정할 수 있어요!
- root.css 는 html head 내부에 링크를 걸어주거나 common CSS 파일에 import 를 시켜서 사용하면 돼요. (reset 이나 normalize 처럼!)

유용한 Library

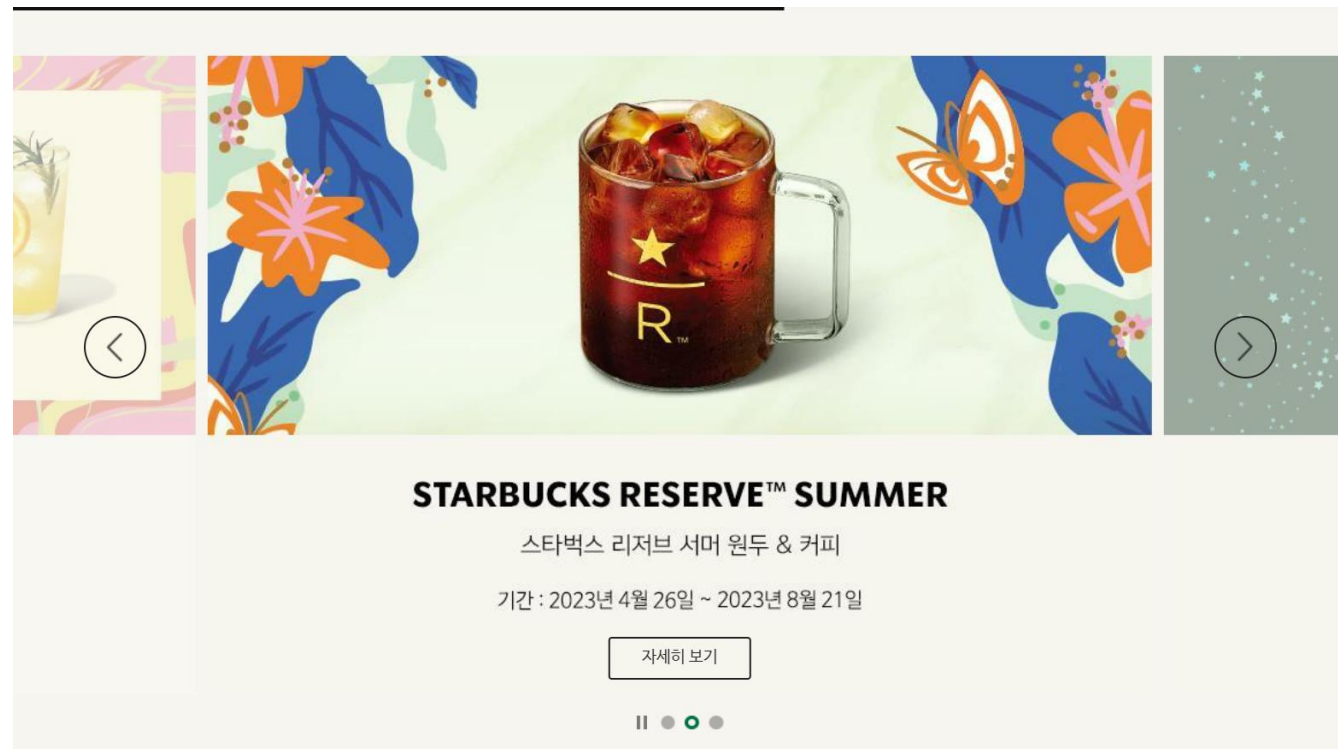
Swiper.js



<https://swiperjs.com/>

Swiper.js

- 그림과 같은 슬라이더를 구현하는 라이브러리!
- 자바스크립트 문법에 익숙하지 않아도 사용 방법만 알면 잘 쓸 수 있습니다.



Swiper.js

- Swiper.js 속성이 잘 정리되어 있는 블로그
 - <https://iridescent-zeal.tistory.com/150>

GSAP (애니메이션 라이브러리)



GSAP 기본

- 먼저 gsap을 이용해서 애니메이션을 다루는 기본 방법부터 알아보겠습니다!
- 공식 문서에 들어가보겠습니다!
 - <https://greensock.com/gsap/>
- Gsap 라이브러리의 CDN이 있는 링크입니다.
 - <https://cdnjs.com/libraries/gsap>

GSAP 기본: gsap.to

- 요소의 현재 상태에서 지정한 상태로 변경시킬 때!
- gsap.to(“선택자”, {animation})
- {animation} 에는 gsap 에서 CSS 에 대해서 지정한 속성과 속성값들이 들어갑니다. (속성값은 단위 없음)
 - X : x축으로 (px)
 - rotation : 회전 각도
 - scaleX : X축 크기
 - duration : 지속 시간 (초)

```
gsap.to(".box2", {  
  x: 1000,  
  rotation: 720,  
  scaleX: 2,  
  duration: 4,  
});
```

GSAP 기본: gsap.from

- 요소의 지정 상태에서 기본 상태로 변경시킬 때! to와 사용 방법은 같아요.
- gsap.from(“선택자”, {animation})

```
gsap.from(".box2", {  
  x: 1000,  
  rotation: 720,  
  scaleX: 2,  
  duration: 4,  
});
```

GSAP 기본: gsap.timeline

- 여러 애니메이션을 연결할 때 사용
- 하나의 타임라인은 애니메이션이 함께 이어지도록
- 여러 개의 timeline을 만들어서 사용 가능

```
let tl = gsap.timeline();
tl.to(".box2", {
  x: 1000,
  rotation: 360,
  duration: 3,
}).to(".box2", {
  backgroundColor: "orange",
  duration: 1,
}).to(".box2", {
  x: 0,
  opacity: 0,
  duration: 3,
});
```


GSAP 기본

- 기본적인 .to .from 만 잘 알고 있으면 대부분의 애니메이션이 구현 가능
- Timeline 을 사용해서 CSS @keyframes 로 구현하는 것 보다 조금 더 유연한 동작 가능



기본 동작 방법을 알았으니 이제 스크롤 기반 애니메이션을 적용시키는 방법에 대해서 알아보까요?

ScrollTrigger!!

- 스크롤에 따라서 gsap 이 작동되도록 도와주는 친구!
- 타임라인에 ScrollTrigger 를 적용하려면 ScrollTrigger.create() 와 함께 사용

ScrollTrigger.create() 없이 사용

- 하나의 animation에 적용
 - ScrollTrigger 값이 animation설정 부분에 들어감
- class1 에 대해서만 스크롤 트리거 사용 가능

```
t1.from(".class1", {
  scrollTrigger: {
    animation: t1,
    trigger: ".class1",
    start: "top 0%",
    end: "bottom 20%",
    scrub: 2,
    markers: true,
    pin: true,
  },
  delay: 0.5,
  scale: 3,
  duration: 1.5,
  opacity: 0,
}).from(".class2", {
  scale: 3,
  duration: 1.5,
  opacity: 0,
});
```

ScrollTrigger.create() 와 함께 사용

- 전체 timeline에 대해서 적용할 때는 ScrollTrigger.create() 메소드 이용

```
t1.from(".s1-slogan1", {  
  delay: 0.5,  
  scale: 3,  
  duration: 1.5,  
  opacity: 0,  
}).from(".s1-slogan2", {  
  scale: 3,  
  duration: 1.5,  
  opacity: 0,  
});
```

```
ScrollTrigger.create({  
  animation: slide1,  
  trigger: ".slide1",  
  start: "top 0%",  
  end: "bottom 20%",  
  scrub: 2,  
  markers: true,  
  pin: true,  
});
```

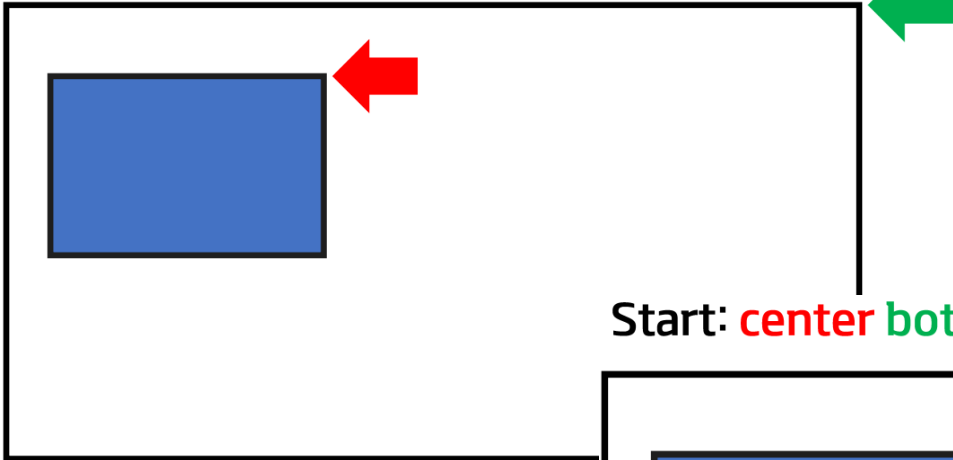
- 모든 타임 라인에 대해서 스크롤 트리거 사용 가능

ScrollTrigger!!

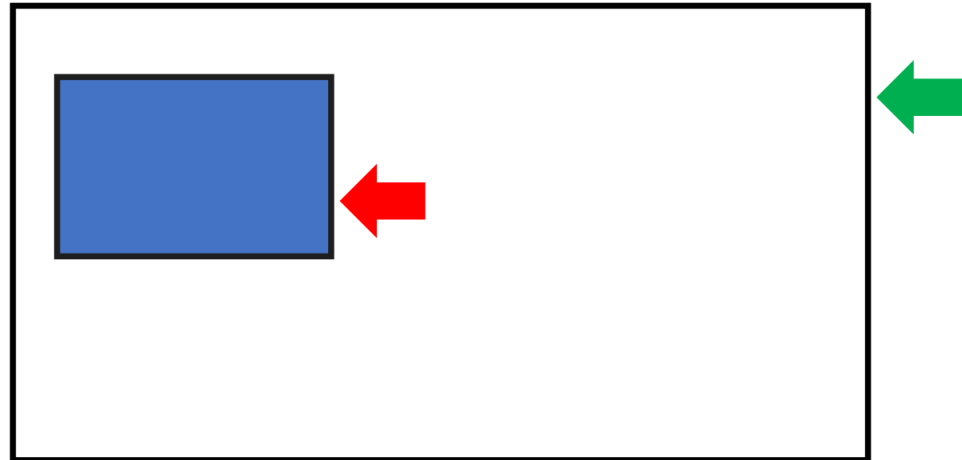
- Start: ‘요소시작점 스크롤 시작점’
- End: ‘요소 끝점 스크롤 끝점’
- scrub: 스크롤 동작을 되돌려도 해당동작을 반복할지 결정
- pin: 해당 스크롤 트리거가 동작되는 동안 고정시킬지 결정

start와 end의 기준

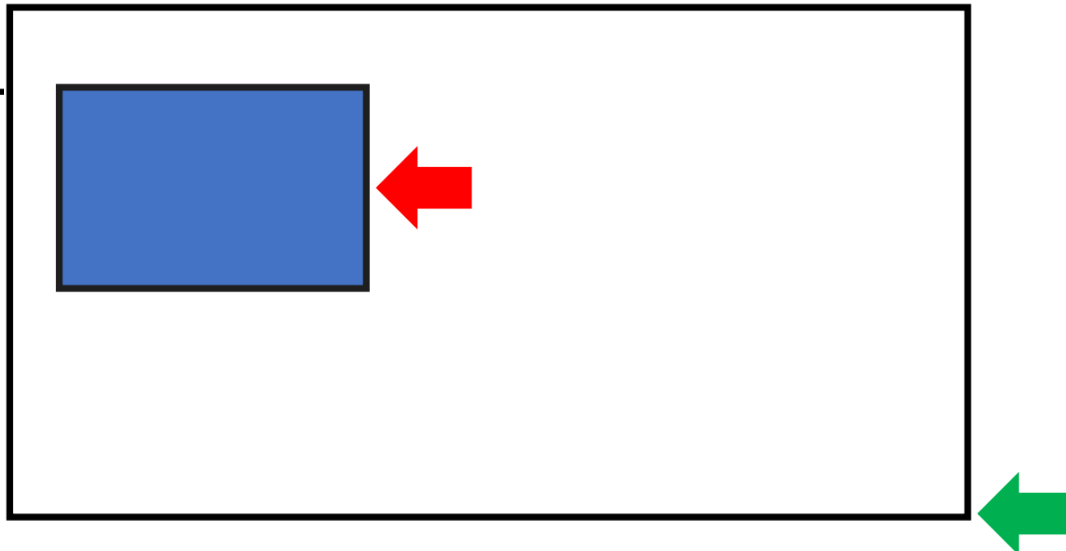
Start: top top,



Start: 70% 20%,



Start: center bottom,



끝!!!

추가 라이브러리 모음



- UI 라이브러리 (like bootstrap)
 - TailWind CSS
 - Materialize
- 슬라이드 라이브러리 (like swiper Library)
 - slick
- Modal
 - Sweet alert
- 스크롤 라이브러리
 - AOS