

9. 메뉴 이동 효과

gsap09.html

1. 작업하기

```
let links = gsap.utils.toArray("#parallax__nav ul li a");

//forEach 메서드를 사용하여 각 앵커 링크 요소에 대해 반복 작업을 수행
links.forEach(link => {
    //링크의 href 속성 값을 가져와서, 이 값으로 해당 섹션을 선택
    합니
    let element = document.querySelector(link.getAttribute(
        "href")),

    // 스크롤 트리거를 설정, 섹션의 스크롤 위치를 기준으로 정의
    linkST = ScrollTrigger.create({
        trigger: element,
        start: "top top"
    });

    //또 다른 스크롤 트리거를 설정, element가 화면에 나타나는 상
    태에 따라 활성화됩니다
    ScrollTrigger.create({
        trigger: element,
        start: "top center", //element의 상단이 뷰포트의 중앙에
        도달할 때 애니메이션이 시작
        end: "bottom center", //element의 하단이 뷰포트의 중앙에
        도달할 때 애니메이션이 끝납니다
        onToggle: self => setActive(link) //스크롤 트리거가 활
        성화되거나 비활성화될 때 호출
    });

    //각 링크에 클릭 이벤트 리스너를 추가
    link.addEventListener("click", e => {
        e.preventDefault();
    });
});
```

```

        //gsap.to 메서드를 사용하여 페이지를 스크롤
        //linkST.start는 linkST트리거의 시작 지점으로 페이지를 스
        크롤 합니다
        gsap.to(window, {duration: 1, scrollTo: linkST.star
t, overwrite: "auto"});
    });
});

//현재 활성화된 링크를 설정하는 함수
function setActive(link) {
    //links 배열의 모든 링크에서 active 클래스를 제거
    links.forEach(el => el.classList.remove("active"));
    //현재 활성화된 링크(link)에 active 클래스를 추가
    link.classList.add("active");
}

```