

3. 스냅 고정 효과 만들기

스크롤이 어중간한 위치에 멈추더라도 현재 스크롤 위치와 가까운 시작 포인트로 스크롤을 이동시킴

gsap.utils.snap (endValue, value): 가장 가까이 지정된 스냅 포인트로 조정

- **endValue** = 스냅하려는 목표 간격 또는 포인트
- **value** = 스냅하려는 현재 값

gsap.utils.normalize :

값의 범위를 특정 최소값과 최대값 사이로 조정하여 일관된 비율로 변환하는 메서드입니다

```
// .parallax__item으로 선택된 모든 요소를 배열로 변환(panels)
let panels = gsap.utils.toArray(".parallax__item");

// panels 배열의 각 패널 요소에 대해 ScrollTrigger.create를 호출하여
// 스크롤 트리거를 생성 합니다
// 각 패널을 지정하고 start를 패널의 상단이 뷰포트의 상단과 일치할 때
// 트리거가 시작되도록 작성하고 tops 변수에 저장
let tops = panels.map(panel => ScrollTrigger.create({trigger: panel, start: "top top"}));

// panels 배열의 각 패널에 대해 ScrollTrigger.create를 호출합니다
panels.forEach((panel, i) => {
  ScrollTrigger.create({
    trigger: panel,
    // 패널의 높이가 윈도우의 높이보다 작으면 "top top"에서 시작하고,
    // 그렇지 않으면 "bottom bottom"에서 시작합니다(이 조건에 따라
    // 패널의 고정 시작위치를 결정합니다)
    start: () => panel.offsetHeight < window.innerHeight ? "top top" : "bottom bottom",
    pin: true, // 스크롤할 때 패널 고정
    pinSpacing: false // 패널 고정 시 여백 제거
  });
});
```

```

//전체 페이지에 대한 또 다른 ScrollTrigger를 생성하여 snap효과를 구현
ScrollTrigger.create({

    snap: { //snap속성은 스크롤 위치가 가장 가까운 패널 시작 위치로 스냅되도록 설정

        snapTo: (progress, self) => { //snapTo 함수는 현재 스크롤 진행 상황(progress)과 self 객체를 인자로 받아 스냅 위치를 계산합니다

            //각 패널의 시작 위치를 배열로 저장합니다
            let panelStarts = tops.map(st => st.start),

            //현재 스크롤 위치(self.scroll)를 가장 가까운 패널 시작 위치로 스냅합니다.
            snapScroll = gsap.utils.snap(panelStarts, self.scroll());

            //gsap.utils.normalize는 스크롤 위치를 페이지 전체 범위 내에서 정규화 합니다.
            return gsap.utils.normalize(0, ScrollTrigger.maxScroll(window), snapScroll);
        },
        duration: 0.5 //스냅 애니메이션의 지속 시간을 설정
    }
});

```