

2. Pin 애니메이션

[gsap02.html](#)

1. 이미지 애니메이션

```
// 01 : 이미지 애니메이션 주기
//타임라인 메서드를 사용하면 요소에 순차적으로 애니메이션을 줄 수 있다
const ani1 = gsap.timeline();

//애니메이션 효과가 순차적으로 실행
ani1.to("#section1 .parallax__item__img", {rotation: 720, scale: 0, borderRadius: 200})
    .to("#section1 .parallax__item__img", {rotation: 0, scale: 1, borderRadius: 20})

//예제 1번에 사용했던 방법과 동일합니다, 단지 방법의 차이가 있을 뿐입니다
ScrollTrigger.create({
  animation: ani1,
  trigger: "#section1",
  start: "top top",
  end: "+=2000",
  scrub: true,
  pin: true,
  anticipatePin: 1, //핀 효과를 조금 더 자연스럽게 연출하기 위한
  속성
  markers: false
});
```

2. 이미지 순차적으로 나오기

```
// 02 : 이미지 순차적으로 나오기
const ani2 = gsap.timeline();
```

```

//from메서드는 to메서드의 반대로 애니메이션의 초기 속성을 적용하는 겁니다.
//.i1의 경우 현재 y값이 0이기 때문에 -100에서 0으로 적용하는 것과 같습니다
//autoAlpha = 투명도를 설정합니다.
ani2.from("#section2 .i1", {y: -100, autoAlpha:0, borderRadius: 200})
    .from("#section2 .i2", {y: 100, autoAlpha:0, borderRadius: 200})
    .from("#section2 .i3", {y: -100, autoAlpha:0, borderRadius: 200});

ScrollTrigger.create({
    animation: ani2,
    trigger: "#section2",
    start: "top top",
    end: "+=2000", //요소가 끝나는 부분부터 2000px을 더주는 것(속도와 길이 조정)
    scrub: true,
    pin: true,
    anticipatePin: 1,
    markers: false
});

```

3. 이미지 랜덤으로 떨어지기

ease옵션

Easing | GSAP | Docs & Learning

"slow", "rough", and "expoScale" eases are not in the core - they are packaged together in an EasePack file in order to minimize file size. "CustomEase", "CustomBounce", and

 <https://gsap.com/docs/v3/Eases/>



```
// 03 : 이미지 랜덤으로 떨어지기
const ani3 = gsap.timeline();

ani3.from("#section3 .parallax__item__img", {
  autoAlpha: 0,
  y: -100,
  ease: "back.out(4)",
  stagger: { //이미지들을 뜻합니다
    amount: 3, //이미지의 갯수
    from: "random" //랜덤으로 이미지를 연출할 수 있다
  }
});

ScrollTrigger.create({
  animation: ani3,
  trigger: "#section3",
  start: "top top",
  end: "+=3000",
  scrub: true,
  pin: true,
  markers: false,
  anticipatePin: 1
});
```

4. 이미지 축소하기

```
// 04 : 이미지 축소하기
const ani4 = gsap.timeline();

ani4.from("#section4 .parallax__item__img", {
  autoAlpha: 0,
  scale: 5,
  width: "100vw",
  height: "100vh"
});
```

```

ScrollTrigger.create({
  animation: ani4,
  trigger: "#section4",
  start: "top top",
  end: "+=3000",
  scrub: true,
  pin: true,
  markers: false,
  anticipatePin: 1
});

```

5. 텍스트 애니메이션

```

// 05 : 텍스트 애니메이션
const ani5 = gsap.timeline();

// xPercent = 퍼센트를 표현할 때 사용,
// "text" 문자열을 추가한 이유 = 동시에 애니메이션 효과를 설정하기 위
함,
// "text" 이름은 랜덤입니다
ani5.to("#section5 .t1", {xPercent: 300}, "text")
    .to("#section5 .t2", {xPercent: -300}, "text")
    .to("#section5 .t3", {xPercent: 300}, "text")
    .to("#section5 .t4", {xPercent: -300}, "text")

ScrollTrigger.create({
  animation: ani5,
  trigger: "#section5",
  start: "top top",
  end: "+=3000",
  scrub: true,
  pin: true,
  markers: true,

```

```

        anticipatePin: 1
    });

```

6. 텍스트 확대하기

```

// 06 : 텍스트 확대하기
const ani6 = gsap.timeline();

//scale: 60 = 크기를 60배 확
ani6.to("#section6 .parallax__item__text", {scale: 60, duration: 2, autoAlpha: 1})
    .to("#section6 .parallax__item__text", {autoAlpha: 0})

ScrollTrigger.create({
    animation: ani6,
    trigger: "#section6",
    start: "top top",
    end: "+=4000",
    scrub: true,
    pin: true,
    anticipatePin: 1,
    markers: false
});

```

7. 텍스트 제자리 애니메이션

```

// 07 : 텍스트 제자리 애니메이션
const ani7 = gsap.timeline();

// "+=1" : 이전 애니메이션이 끝나고 1초 뒤에 실행
ani7.from("#section7 .t1", {autoAlpha: 0, duration: 1, y: 50}, "+=1")
    .from("#section7 .t2", {autoAlpha: 0, duration: 1, y: 50}

```

```

0}, "+=1")
    .from("#section7 .t3", {autoAlpha: 0, duration: 1, y: 5
0}, "+=1")
    .from("#section7 .t4", {autoAlpha: 0, duration: 1, y: 5
0}, "+=1")
    .from("#section7 .t5", {autoAlpha: 0, duration: 1, y: 5
0}, "+=1")
    .from("#section7 .t6", {autoAlpha: 0, duration: 1, y: 5
0}, "+=1")
    .from("#section7 .t7", {autoAlpha: 0, duration: 1, y: 5
0}, "+=1")

ScrollTrigger.create({
    animation: ani7,
    trigger: "#section7",
    start: "top top",
    end: "+=6000",
    scrub: true,
    pin: true,
    markers: false,
    anticipatePin: 1
});

```

8. 텍스트 애니메이션

```

// 08 : 텍스트 애니메이션
const ani8 = gsap.timeline();

ani8.from("#section8 .t1", {x: innerWidth * 1})
    .from("#section8 .t2", {x: innerWidth * -1})
    .from("#section8 .t3", {x: innerWidth * 1})
    .from("#section8 .i1", {x: innerWidth * 1, rotation: 36
0, scale: 1.5})

ScrollTrigger.create({
    animation: ani8,

```

```

    trigger: "#section8",
    start: "top top",
    end: "+=4000",
    scrub: true,
    pin: true,
    markers: false,
    anticipatePin: 1
  });

```

9. 이미지 확대하기

```

//09 : 이미지 확대하기
const ani9 = gsap.timeline();

ani9.to("#section9 .parallax__item__img", {scale: 13})
    .to("#section9 .parallax__item__img", {autoAlpha: 0})

ScrollTrigger.create({
  animation: ani9,
  trigger: "#section9",
  start: "top top",
  end: "+=4000",
  scrub: true,
  pin: true,
  markers: false,
  anticipatePin: 1
});

```