

6. 텍스트 효과

gsap06.html

1. 텍스트 분리하기

```
let text = document.querySelector(".split");
let splitText = text.innerText;

//split()을 통해 글씨를 하나씩 가져오고 사이에 span을 넣어줍니다.
//글자 중간에 들어가는 거니까 닫는태그부터 시작합니다.
let splitWrap = splitText.split('').join('</span><span>');

//전체에 시작태그와 끝나는 태그로 감싸 줍니다.(한글자씩 span으로 분리됩니다)
text.innerHTML = splitWrap = "<span>" + splitWrap + "</span>>";
```

```

    <figure class="parallax__item__imgWrap">...</figure>
  <p class="parallax__item__desc split"> = $0
    <span>높</span>
    <span>은</span>
    <span> </span>
    <span>목</span>
    <span>표</span>
    <span>를</span>
    <span> </span>
    <span>세</span>
    <span>우</span>
    <span>고</span>
    <span>,</span>
    <span> </span>
    <span>스</span>
    <span>스</span>
    <span>로</span>
    <span> </span>
    <span>채</span>
```

2. 모든 텍스트 분리하기

```
//여러개의 요소를 선택해야하니 querySelectorAll을 사용하고 forEach
문으로 순회하며 적용합니다.
document.querySelectorAll(".split").forEach(text => {
    //웹 표준을 준수하기 위해 aria-hidden='true'를 설정, 분리된 글
    씨는 접근성을 위해 숨겨 놓습니다.
    let splitWrap = text.innerText.split(' ').join("</span><
span aria-hidden='true'>");

    //분리된 텍스트를 innerHTML을 통해 다시 넣어줍니다
    text.innerHTML = "<span aria-hidden='true'>"+ splitWrap
    + "</span>";

    //부모 박스에 aria-label을 붙여 줍니다
    text.setAttribute("aria-label", text.innerText);
});
```

3. 모든 텍스트 분리하기 : 여백 표현하기

JS

```
//split클래스를 가진 모든 요소를 선택하여 forEach메서드를 사용하여 반
복 작업을 수행합니다.
document.querySelectorAll(".split").forEach(text => {

    //현재 요소의 텍스트를 theText에 저장
    let theText = text.innerText;
    //새로운 텍스트를 저장할 빈 문자열을 생성
    let newText = "";

    //요소의 텍스트의 길이만큼 반복
    for(let i=0; i<text.innerText.length; i++){

        //각 문자를 span태그로 감싸기 위해 newText에 span태그를 추
```

가합니다.

```
//aria-hidden='true'속성은 스크린 리더가 이 태그를 무시하
도록 지시합니다
newText += "<span aria-hidden='true'>";
//현재 문자가 공백인 경우
if (text.innerText[i] == " "){
    //공백 문자를 추가 합니다
    newText += " "
} else { //현재 문자가 공백이 아닌 경우
    //원래의 문자를 추가합니다
    newText += text.innerText[i];
}
//각 문자 뒤에 </span>태그를 추가하여 문자를 span태그로 감쌌
니다.
newText += "</span>";
}
//요소의 innerHTML을 newText로 설정하여 텍스트를 새로운 HTML
형식으로 업데이트 합니다
text.innerHTML = newText;
//aria-label속성을 추가하여 원래 텍스트를 값으로 설정합니다.
text.setAttribute("aria-label", theText);
});
```

CSS

```
/* 글씨는 인라인 구조이기 때문에 애니메이션을 주기 위해 inline-block
을 꼭 주셔야 합니다 */
.split > span {
    display: inline-block;
}
```

JS

```
...
//.from은 지정된 시작 상태에서 애니메이션을 시작합니다
gsap.from(".split span", {
```

```

yPercent: 100, //화면 아래쪽으로 100% 만큼 떨어뜨려 놓습니다
autoAlpha: 0, //불투명도 0
duration: 2, // 애니메이션이 2초 동안 진행
ease: "circ.out", //애니메이션이 끝날 때 속도가 줄어드는 easin
g효과

stagger: 0.1, //애니메이션 시작 시각을 요소마다 0.1초씩 차이를
두어 지연 시킵니다(순차적으로 발생하는 효과)

scrollTrigger: {
  trigger: ".split",
  start: "top center", //요소의 상단이 뷰포트의 중앙에 도달
할 때 애니메이션이 시작
  end: "+=400", // 애니메이션의 끝 위치를 시작위치에서 400픽
셀 아래로 설정
  markers: true,
  scrub: 1, //스크롤과 애니메이션을 동기화
}
});

```

4. 모든 텍스트 분리하기 : 여백 표현하기 : 다중 선택

JS

```

...

//.split 클래스를 가진 모든 요소를 배열로 변환하고, forEach를 사용하
여 각 요소에 대해 반복 작업을 수행, text는 현재 반복 중인 .split 요소
를 나타냅니다
gsap.utils.toArray(".split").forEach((text) => {
  //현재 .split 요소 내의 모든 span 요소를 선택하고, gsap.fr
om 메서드를 사용하여 애니메이션을 설정합니다
  gsap.from(text.querySelectorAll("span"), {
    yPercent: 100,
    outoAlpha: 0,

```

```

        duration: 1,
        ease: "circ.out",
        //stagger: 0.04,
        stagger: {
            amount: 1, //지연의 총량을 설정(1초동안 요소들이 분산되
            //도록 지연)
            from: "random" //span 요소의 애니메이션 시작 시점을
            //무작위로 설정
        },

        scrollTrigger: {
            trigger: text,
            start: "top bottom",
            end: "+=400",
            markers: true,
        }
    });
});

```

5. split-type 사용하기

```

...
<!--split-type 플러그인-->
<script src="https://unpkg.com/split-type"></script>

```

```

//split클래스를 가진 모든 요소를 배열로 변환
const targets = gsap.utils.toArray(".split");

//forEach 메서드를 사용하여 각 target 요소에 대해 반복 작업을 수행합
//니다
targets.forEach((target) => {

//SplitType 인스턴스를 생성하여 target 요소의 텍스트를 줄, 단어, 문
//자 단위로 분할합니다
let SplitClient = new SplitType(target, { type: "lines",

```

```

words, chars" });

    //텍스트가 줄 단위로 분할된 결과를 lines 변수에 저장(줄 단위로 애니메이션을 적용)
    let lines = SplitClient.lines;
    //텍스트가 단어 단위로 분할된 결과를 words변수에 저장(단어 단위로 애니메이션을 적용)
    let words = SplitClient.words;
    //텍스트가 문자 단위로 분할된 결과를 chars 변수에 저장(문자 단위로 애니메이션을 적용)
    let chars = SplitClient.chars;

    //words (단어) 요소에 애니메이션을 적용
    gsap.from(words, {
        yPercent: 100,
        opacity: 0,
        duration: 1,
        ease: "circ.out",
        stagger: {
            amount: 1,
            from: "random"
        },
        scrollTrigger: {
            trigger: target,
            start: "top bottom",
            end: "+=400",
            markers: true,
        }
    });
});

```