

1. Html 작성

곰돌이 얼굴을 형상화 한 도형 이미지를 html에 넣어주었다.

Alt는 사용자의 네트워크 환경이나 sec 속성값의 오류, 시각장애인 등등 이미지를 볼 수 없을 때를 대비하여 이미지대신 대체 정보를 제공 목적으로 자세히 작성해주었다.

```
<div class="ear1">
    
</div>
<div class="face">
    
</div>
<div class="ear2">
    
</div>
```

이미지를 감싼 박스에 직접적으로 크기를 지정해주었다.

```
1. main .welcome .container .ear1 {
2.   width: 174px;
3.   height: 160px;
4. }
5. main .welcome .container .face {
6.   width: 532px;
7.   height: 467px;
8. }
9. main .welcome .container .ear2 {
10.  width: 174px;
11.  height: 160px;
12. }
```

2. CSS 작성

Position을 통해 부모relative 기준 각각 이미지의 위치를 지정해주고, 공통으로 들어가는 값은 묶어서 작성하였다

z-index를 통해 z축 순서를 지정해주었다.

Z-index란 하위 플렉스 아이템의 Z축 순서를 지정한다. 더 큰 z-index 값을 가진 요소가 작은 값의 요소 위를 덮는다.

```
main .welcome {
  height: 830px;
  line-height: 75px;
  padding: 0 5%;
  flex-direction: column;
```

```

    justify-content: center;
    align-items: center;
    position: relative;
    gap: 100px;
}
main .welcome .teddy_img {
    z-index: 9;
}
main .welcome .container .ear1,
main .welcome .container .face,
main .welcome .container .ear2 {
    position: absolute;
    transform: translate(-50%, -50%);
    z-index: 99;
    transition: transform 3s ease-in-out;
}
main .welcome .container .ear1 {
    top: 25%;
    left: 38%;
    transform: translate(-200%, -200%);
}
main .welcome .container .face {
    top: 50%;
    left: 50%;
    transform: translate(-200%, 100%);
}
main .welcome .container .ear2 {
    top: 25%;
    left: 62%;
    z-index: -99;
    transform: translate(100%, -300%);
}
main .welcome .container .ear1.move,
main .welcome .container .face.move,
main .welcome .container .ear2.move {
    transform: translate(-50%, -50%) rotate(360deg);
}

```

3. Java Script 작성

- A. Window.addEventListener()를 이용해 실행할 함수를 등록한다.

```

//첫번째 섹션 곰돌이
window.addEventListener("load", function () {
    var ear1 = document.querySelector(".welcome .container .ear1");
    var face = document.querySelector(".welcome .container .face");
    var ear2 = document.querySelector(".welcome .container .ear2");

```

- B. `moveImagesToCenter`를 이용해, 곰돌이 이미지가 가운데로 이동하는 애니메이션 효과를 보이게 한다.

```
// 이미지 요소들을 모이는 값
function moveImagesToCenter() {
  ear1.classList.add("move");
  face.classList.add("move");
  ear2.classList.add("move");
}
setTimeout(moveImagesToCenter, 100);
});
```

- C. `bearLoading`은 `setInterval`을 사용하여 3초마다 곰돌이 애니메이션이 이동하는 효과를 만들어 주었다.

```
c. const bearLoading = function () {
  var ear1 = document.querySelector(".welcome .container .ear1");
  var face = document.querySelector(".welcome .container .face");
  var ear2 = document.querySelector(".welcome .container .ear2");
  ear1.classList.add("move");
  face.classList.add("move");
  ear2.classList.add("move");
};

setInterval(bearLoading, 3000);
```

- d. `bearRemove`는 6초마다 `move` 클래스를 제거하는 역할을 한다. 6초마다 곰돌이 효과가 제거되고, 콘솔에 `ddd`를 추가하는 효과를 구현하였다.

```
const bearRemove = function () {
  var ear1 = document.querySelector(".welcome .container .ear1");
  var face = document.querySelector(".welcome .container .face");
  var ear2 = document.querySelector(".welcome .container .ear2");
  console.log("ddd");
  ear1.classList.remove("move");
  face.classList.remove("move");
  ear2.classList.remove("move");
};

setInterval(bearRemove, 6000);
```