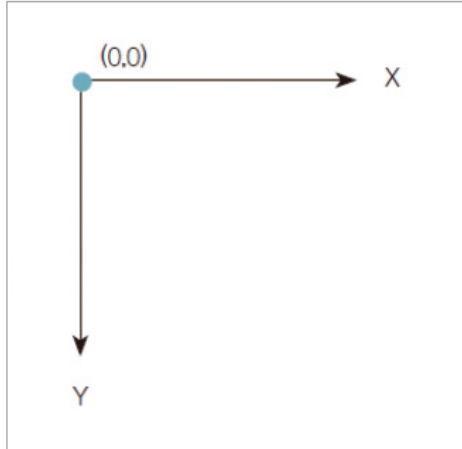


변형

변형(transform, 트랜스폼) : 특정 요소의 크기나 형태 등 스타일이 바뀌는 것

2차원 변형

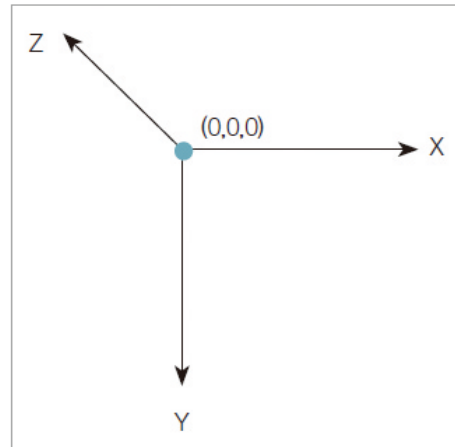
- 수평이나 수직으로 웹 요소 변형
- 크기나 각도만 지정하면 됨
- 2차원 좌표 사용



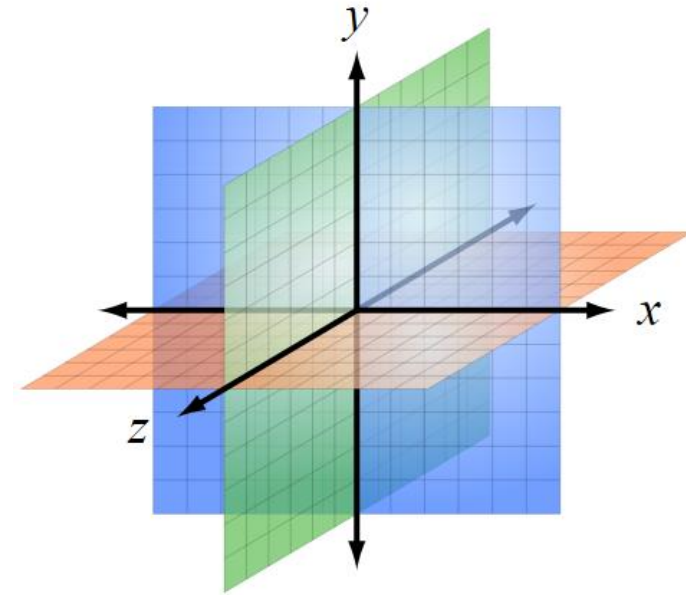
2차원 좌표계

3차원 변형

- x축과 y축에 원근감 추가
- z축은 앞뒤로 이동. 보는 사람 쪽으로 다가올 수록 값이 더 커짐



3차원 좌표계



translate 함수

지정한 방향으로 이동할 거리를 지정하면 해당 요소를 이동시킴

```
기본형  transform:translate(tx, ty)
        transform:translate3d(tx, ty, tz)
        transform:translateX(tx)
        transform:translateY(ty)
        transform:translateZ(tz)
```

- **transform:translate(tx, ty)** - x축 방향으로 tx만큼, y축 방향으로 ty만큼 이동합니다. tx와 ty 두 가지 값을 사용하지만 ty 값이 주어지지 않으면 0으로 간주합니다.
- **transform:translate3d(tx, ty, tz)** - x축 방향으로 tx만큼, y축 방향으로 ty만큼, 그리고 z축 방향(앞뒤)으로 tz만큼 이동합니다.
- **transform:translateX(tx)** - x축 방향으로 tx만큼 이동합니다.
- **transform:translateY(ty)** - y축 방향으로 ty만큼 이동합니다.
- **transform:translateZ(tz)** - z축 방향으로 tz만큼 이동합니다.

```
<style>
  .movex:hover { transform: translateX(50px); }
  .movey:hover { transform: translateY(20px); }
  .movexy:hover { transform : translate(10px, 20px); }
</style>
```

```
<div class="origin">
  <div class="movex">  </div>
</div>
<div class="origin">
  <div class="movey">  </div>
</div>
<div class="origin">
  <div class="movexy">  </div>
</div>
```



scale 함수

지정한 크기만큼 요소를 확대/축소

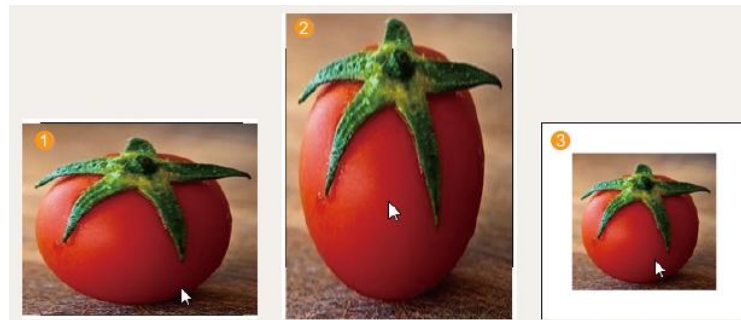
기본형

- transform:scale(sx, sy) —①
- transform:scale3d(sx, sy, sz) —②
- transform:scaleX(sx) —③
- transform:scaleY(sy) —④
- transform:scaleZ(sz) —⑤

- **transform:scale(sx, sy)** - x축 방향으로 sx만큼, y축 방향으로 sy만큼 확대합니다. sy 값이 주어지지 않는다면 sx 값과 같다고 간주합니다. 예를 들어 scale(2.0)는 scale(2,2)와 같은 함수이며 요소를 두 배로 확대합니다.
- **transform:scale3d(sx, sy, sz)** - x축 방향으로 sx만큼, y축 방향으로 sy만큼, 그리고 z축 방향으로 sz만큼 확대합니다.
- **transform:scaleX(sx)** - x축 방향으로 sx만큼 확대합니다.
- **transform:scaleY(sy)** - y축 방향으로 sy만큼 확대합니다.
- **transform:scaleZ(sz)** - z축 방향으로 sz만큼 확대합니다.

```
<style>
.scale: hover { transform: scaleX(1.2); }
.scaley: hover { transform: scaleY(1.5); }
.scale: hover { transform: scale(0.7); }
</style>
```

```
<div class="origin">
  <div class="scalex">  </div>
</div>
<div class="origin">
  <div class="scaley">  </div>
</div>
<div class="origin">
  <div class="scale">  </div>
</div>
```



rotate 함수

- 각도만큼 웹 요소를 시계 방향이나 시계 반대 방향으로 회전
- 일반 각도(degree)나 라디안(radian) 값 사용(1라디안=1/180°)

2차원 함수 기본형 `transform: rotate(각도)`

```
<style>
.rotate1:hover { transform: rotate(20deg); }
.rotate2:hover { transform: rotate(-40deg); }
</style>
```

```
<div class="origin">
  <div class="rotate1">  </div>
</div>
<div class="origin">
  <div class="rotate2">  </div>
</div>
```



skew 함수

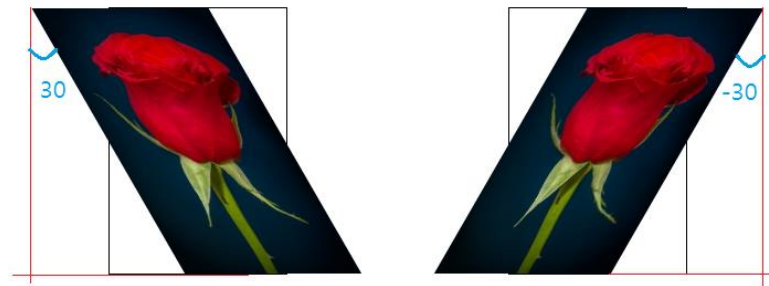
요소를 지정한 각도만큼 비틀어 왜곡

기본형 `transform:skew(ax, ay)` —①
`transform:skewX(ax)` —②
`transform:skewY(ay)` —③

- **transform:skewX(ax)** - x축을 따라 당깁니다.
- **transform:skewY(ay)** - y축을 따라 당깁니다.
- **transform:skew(ax, ay)** - 첫 번째 각도는 x축을 따라 당기는 각도이고 두 번째 각도는 y축을 따라 당기는 각도입니다. 두 번째 값이 주어지지 않으면 y축에 대한 각도를 0으로 간주합니다.

```
<style>  
  .skewx1 { transform: skewX(30deg); }  
  .skewx2 { transform: skewX(-30deg); }  
</style>
```

```
<div class="origin">  
  <div class="skewx1"></div>  
</div>  
<div class="origin">  
  <div class="skewx2"></div>  
</div>
```



변형과 관련된 속성들

transform-origin 속성

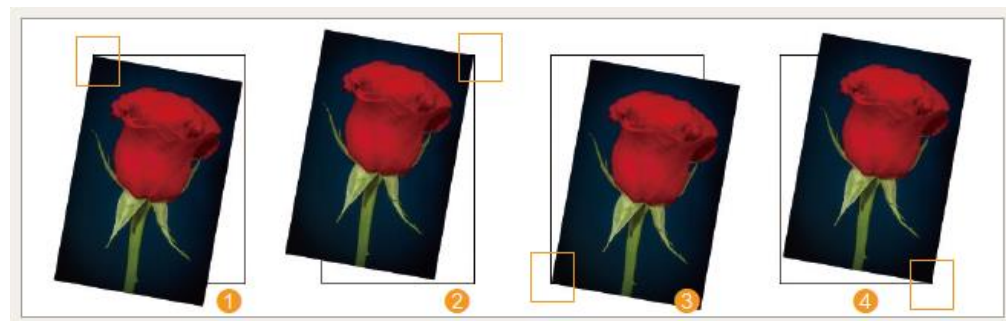
특정 지점을 변형의 기준으로 설정

기본형 transform-origin: <x축> <y축> <z축>

속성 값	설명
<x축>	원점 기준의 x 좌표값으로 길이 값이나 <백분율>, left, center, right 중에서 사용할 수 있습니다.
<y축>	원점 기준의 y 좌표값으로 길이 값이나 <백분율>, top, center, bottom 중에서 선택할 수 있습니다.
<z축>	원점 기준의 z 좌표값으로 길이 값만 사용할 수 있습니다.

```
<style>
.rose {transform: rotateZ(10deg);}
.ltop .rose { transform-origin: left top; }
.rtop .rose { transform-origin: right top; }
.lbottom .rose { transform-origin: left bottom; }
.rbottom .rose { transform-origin : right bottom; }
</style>
```

```
<div class="origin">
  <div class="ltop">  </div>
</div>
<div class="origin">
  <div class="rtop">  </div>
</div>
<div class="origin">
  <div class="lbottom">  </div>
</div>
<div class="origin">
  <div class="rbottom">  </div>
</div>
```



변형과 관련된 속성들

perspective 속성

- 원근감을 갖게 함.
- 속성 값은 0보다 커야 하며 값이 클수록 사용자로부터 멀어짐.

기본형 perspective: <크기> | none;

속성 값	설명
<크기>	원래 위치에서 사용자가 있는 방향으로 얼마나 이동하는지를 픽셀 크기로 지정합니다.
none	perspective를 지정하지 않습니다. 기본 값입니다.★

perspective-origin 속성

- 입체적으로 표현할 요소의 아랫부분(bottom) 위치 지정
- 좀더 높은 곳에서 원근을 조절하는 듯한 느낌을 갖게 함.

기본형 perspective-origin: <x축 값> | <y축 값>;

속성 값	설명
<x축 값>	웹 요소가 x축에서 어디에 위치하는지를 지정합니다. 사용할 수 있는 값은 길이 값이나 백분율, left, right, center입니다. 기본 값은 50%입니다.
<y축 값>	웹 요소가 y축에서 어디에 위치하는지를 지정합니다. 사용할 수 있는 값은 길이 값이나 백분율, top, center, bottom입니다. 기본 값은 50%입니다.

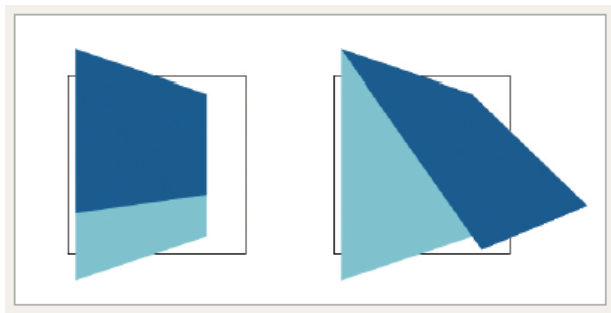
변형과 관련된 속성들

transform-style 속성

부모 요소에 적용한 3D 변형을 하위 요소에도 적용

기본형 transform-style: flat | preserve-3d

속성 값	설명
flat	하위 요소를 평면으로 처리합니다.
preserve-3d	하위 요소들에 3D효과를 적용합니다.



```
<style>
.box1 {
  background:#82cbd8;
  transform: rotateY(45deg);
}
.box2 {
  background: #0d6097;
  transform-origin: left top;
  transform: rotateX(45deg);
}
#tr-style1 { transform-style:flat ; }
#tr-style2 { transform-style:preserve-3d; }
</style>
```

```
<div class="container">
  <div class="box1" id="tr-style1">
    <div class="box2"></div>
  </div>
</div>
<div class="container">
  <div class="box1" id="tr-style2">
    <div class="box2"></div>
  </div>
</div>
```


변형과 관련된 속성들

backface-visibility 속성

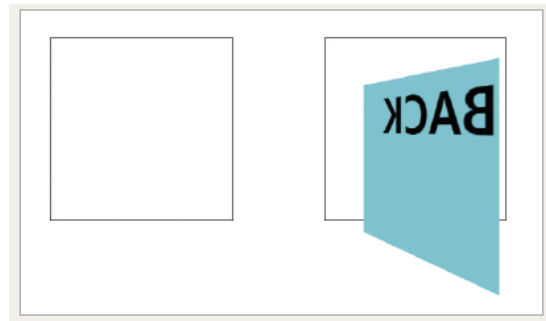
요소의 뒷면, 즉 반대쪽 면을 표시할 것인지 결정

기본형 backface-visibility : visible | hidden

속성 값	설명
visible	뒷면을 표시합니다. 기본 값입니다.★
hidden	뒷면을 표시하지 않습니다.

```
<style>
.box {
  background:#82cbd8;
  transform: rotateY(135deg);
}
#back1 { backface-visibility: hidden; }
#back2 { backface-visibility : visible; }
</style>
```

```
<div class="container">
  <div class="box" id="back1"><h1>BACK</h1></div>
</div>
<div class="container">
  <div class="box" id="back2"><h1>BACK</h1></div>
</div>
```



형성평가 1/4

1. `translate(50px, 100px)` 함수의 역할은 무엇인가요?

- 1) 요소를 50px만큼 회전합니다.
- 2) 요소를 x축으로 50px, y축으로 100px 이동합니다.
- 3) 요소를 x축으로 100px, y축으로 50px 이동합니다.
- 4) 요소를 기울여 위치를 변경합니다.

2. `scale(2, 1.5)` 함수는 무엇을 의미하나요?

- 1) 요소를 2배 확대합니다.
- 2) 요소를 x축으로 2배, y축으로 1.5배 확대합니다.
- 3) 요소를 1.5배 축소합니다.
- 4) 요소를 회전시키고 크기를 조정합니다.

형성평가 2/4

3. rotate(45deg)는 어떤 변형을 의미하나요?

- 1) 요소를 45도 기울입니다.
- 2) 요소를 45도 회전시킵니다.
- 3) 요소를 45px 이동합니다.
- 4) 요소의 크기를 45배로 조정합니다.

4. skew(30deg, 10deg) 함수의 역할은 무엇인가요?

- 1) 요소를 x축으로 30px, y축으로 10px 이동합니다.
- 2) 요소를 x축으로 30도, y축으로 10도 기울입니다.
- 3) 요소를 기울이고 크기를 축소합니다.
- 4) 요소를 x축으로 30도, y축으로 10도 회전합니다.

형성평가 3/4

5. transform-origin: top left;는 무엇을 설정하나요?

- 1) 변형 기준점을 요소의 왼쪽 상단으로 설정합니다.
- 2) 변형 기준점을 요소의 오른쪽 하단으로 설정합니다.
- 3) 변형 기준점을 x축과 y축의 중앙으로 설정합니다.
- 4) 변형 기준점을 숨깁니다.

6. perspective: 500px; 속성의 역할은 무엇인가요?

- 1) 요소를 500px만큼 이동합니다.
- 2) 3D 변환의 원근감을 설정합니다.
- 3) 500px의 회전을 추가합니다.
- 4) 요소의 기울기를 500px로 설정합니다.

형성평가 4/4

7. `backface-visibility: hidden;`은 어떤 역할을 하나요?

- 1) 요소의 앞면을 숨깁니다.
- 2) 요소의 뒷면을 숨깁니다.
- 3) 요소를 3D로 보이게 합니다.
- 4) 변형 효과를 제거합니다.

8. `skew` 함수와 `rotate` 함수의 차이는 무엇인가요?

- 1) `skew`는 회전을, `rotate`는 기울임을 설정합니다.
- 2) `skew`는 기울임을, `rotate`는 회전을 설정합니다.
- 3) 둘 다 요소를 이동합니다.
- 4) 둘 다 크기를 조정합니다.

학습정리 1/1

1 . 변형

함수/속성	설명	예시
translate(x, y) 함수	요소를 x축, y축 방향으로 이동	transform: translate(50px, 100px); (x축으로 50px, y축으로 100px 이동)
scale(x, y) 함수	요소의 크기를 x축, y축 방향으로 비율을 기준으로 확대 또는 축소	transform: scale(2, 1.5); (x축 2배, y축 1.5배 크기 조정)
rotate(deg) 함수	요소를 지정된 각도만큼 회전	transform: rotate(45deg); (요소를 45도 회전)
skew(x, y) 함수	요소를 x축, y축 방향으로 기울임	transform: skew(30deg, 10deg); (x축으로 30도, y축으로 10도 기울임)
transform-origin 속성	변형의 기준점	transform-origin: top left; (변형 기준점을 왼쪽 상단으로 설정)
perspective 속성	3D 변환에 사용할 원근감	perspective: 500px; (카메라가 500px 거리에서 보는 원근감 설정)
perspective-origin 속성	원근감의 기준점	perspective-origin: left top; (원근 기준점을 왼쪽 상단으로 설정)
transform-style 속성	자식 요소가 3D 공간에서 배치여부	transform-style: preserve-3d; (자식 요소가 3D 공간에서 배치됨)
backface-visibility 속성	요소의 뒷면이 보일지 여부	backface-visibility: hidden; (요소의 뒷면을 숨김)

