

Choi yulhee

UI/UX 웹표준 디자인

이전 양재역

01 CSS transform (2D변형)

02 CSS transform (3D변형)

01

1) CSS transform (2D변형)

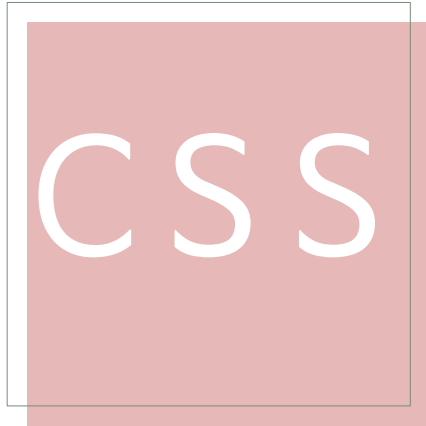
C	S	S	3
기	본	문	법

C	S	S	3
---	---	---	---

INDEX

05_CSS transform

1) CSS transform (2D변형)



01

1) CSS transform (2D 변형)

C S S 3
기 본 문 법

C S S 3

INDEX

핵심용어	용어설명
Transform (2d 변형)	특정 요소의 모양을 평면상에서 자유롭게 변형하기
translate	x축이나 y축으로 이동
skew	x축이나 y축으로 기울임
scale	x축이나 y축으로 확대/축소
rotate	평면상에서의 회전
transform-origin	해당 요소의 모양을 변형 시, 변형의 기준점 변경

01 | translate

- **translate** : x축이나 y축으로 이동

translate는 현재 요소의 위치 값을 기준으로 x축 또는 y축으로 해당요소를 이동시킨다.

translate 사용법

x축으로 이동

transform : **translateX**(x축 이동거리);

y축으로 이동

transform : **translateY**(y축 이동거리);

x,y축으로 이동

transform : **translate**(x축 이동거리, y축 이동거리);

translate는 해당요소의 원래 위치 값을 기준으로 삼아 이동합니다.

01 1) CSS transform (2D변형)

C	S	S	3
기	본	문	법

C	S	S	3
---	---	---	---

INDEX

• 예제

```
<body>

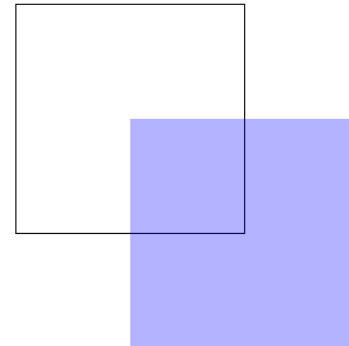
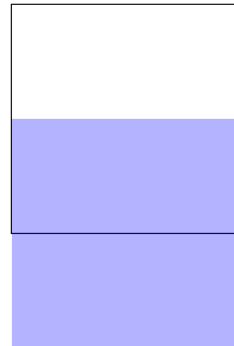
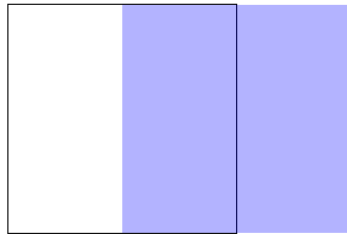
  <div> <p> </p> </div>
  <div> <p> </p> </div>
  <div> <p> </p> </div>

</body>
```

▲ html

```
* { margin:0; padding:0 }
div { width:200px; height:200px; margin:100px; border:1px solid #000;
      float:left; }
div p { width:100%; height:100%; background: blue; opacity: 0.3; }
div:nth-child(1) p { transform: translateX(100px); }
div:nth-child(2) p { transform: translateY(100px); }
div:nth-child(3) p { transform: translate(100px, 100px); }
```

▲ css



01

1) CSS transform (2D변형)

C S S 3
기 본 문 법

C S S 3

INDEX

02 | skew

- skew : x축이나 y축으로 기울임

skew는 현재 요소의 각도 값을 기준으로 x축 또는 y축으로 해당요소를 원하는 각도만큼 기울인다.

skew 사용법

x축으로 기울이기

transform : skewX(각도deg);

y축으로 기울이기

transform : skewY(각도deg);

x,y축으로 동시에 기울이기

transform : skew(x축 각도deg, y축 각도deg);

각도를 나타내는 단위는 degree의 약자인 deg입니다.

01

1) CSS transform (2D변형)

C	S	S	3
기	본	문	법

C	S	S	3
---	---	---	---

INDEX

- **예제** tip: skew의 수치 값을 마이너스로 설정하면 반대 방향으로 기울어 짐

```
<body>

  <div> <p> </p> </div>
  <div> <p> </p> </div>
  <div> <p> </p> </div>

</body>
```

▲ html

```
* { margin:0; padding:0 }
div { width:200px; height:200px; margin:100px; border:1px solid #000;
      float:left; }
div p { width:100%; height:100%; background: blue; opacity: 0.3; }
div:nth-child(1) p { transform: skewX(30deg) ;}
div:nth-child(2) p { transform: skewY(30deg); }
div:nth-child(3) p { transform: skew(30deg, 30deg); }
```

▲ css

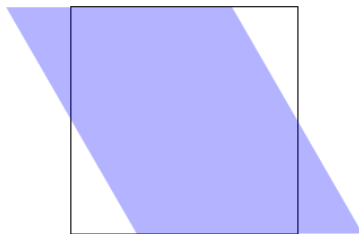
01

1) CSS transform (2D변형)

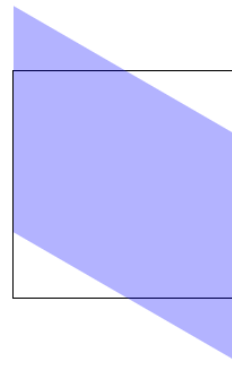
C S S 3
기 본 문 법

C S S 3

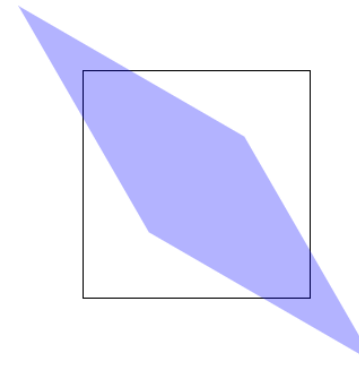
INDEX



x축으로 기울이기



y축으로 기울이기



x,y축으로 동시에 기울이기

03 scale

- **scale** : x축이나 y축으로 확대/축소

scale은 현재 요소의 크기를 기준으로 x축 또는 y축으로 해당 요소를 확대 또는 축소시킨다.

scale 사용법

x축으로 확대/축소

transform : scaleX(x축 비율);

y축으로 확대/축소

transform : scaleY(y축 비율);

x,y축으로 확대/축소

transform : scale(x축 비율, y축 비율);

scale의 비율은 원본 크기 1을 기준으로 1보다 크면 확대 작으면 축소되며 비율 값은 소수점으로 설정 가능.

01

1) CSS transform (2D변형)

C	S	S	3
기	본	문	법

C	S	S	3
---	---	---	---

INDEX

- **예제** tip: scale은 마이너스 값을 지원하지 않음.

```
<body>

  <div><p></p></div>
  <div><p></p></div>
  <div><p></p></div>

</body>
```

▲ html

```
* { margin:0; padding:0 }
div { width:200px; height:200px; margin:100px; border:1px solid #000;
      float:left; }
div p { width:100%; height:100%; background: blue; opacity: 0.3; }
div:nth-child(1) p { transform: scaleX(1.5); }
div:nth-child(2) p { transform: scaleY(0.5); }
div:nth-child(3) p { transform: scale(2, 2); }
```

▲ css

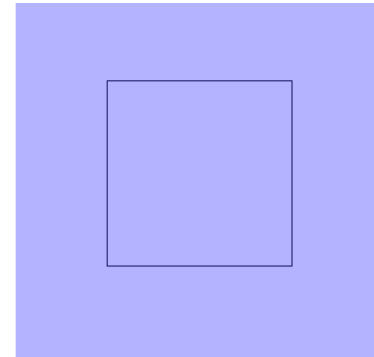
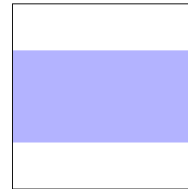
01

1) CSS transform (2D변형)

C S S 3
기 본 문 법

C S S 3

INDEX



04 rotate

- rotate : 평면상에서의 회전

rotate은 현재 요소의 각도를 기준으로 해당 요소를 평면상에서 회전시킨다.

이전 속성들과는 달리 평면상에서 회전시킬 경우에는 따로 x축, y축을 지정하지 않습니다.

rotate 사용법

```
transform : rotate(회전 각도deg);
```

회전각이 플러스 값을 때에는 시계방향,

마이너스 값일 때에는 반시계 방향으로 회전합니다.

01 1) CSS transform (2D변형)

C	S	S	3
기	본	문	법

C	S	S	3
---	---	---	---

INDEX

- **예제** tip: scale은 마이너스 값을 지원하지 않음.

```
<body>

  <div> <p> </p> </div>
  <div> <p> </p> </div>

</body>
```

▲ html

```
* { margin:0; padding:0 }
div { width:200px; height:200px; margin:100px; border:1px solid #000;
      float:left; }
div p { width:100%; height:100%; background: blue; opacity: 0.3; }
div:nth-child(1):hover p { transform: rotate(15deg); }
div:nth-child(2) hover p { transform: rotate(-30deg); }
```

▲ css

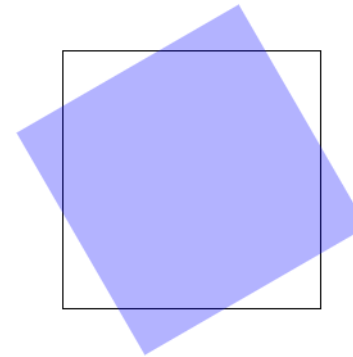
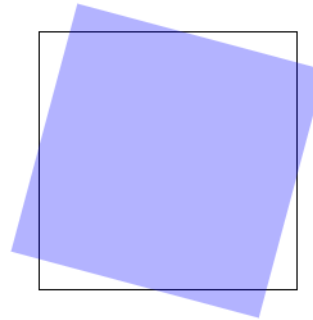
01

1) CSS transform (2D변형)

C S S 3
기 본 문 법

C S S 3

INDEX



05 transform-origin

transform-origin : 해당 요소의 모양을 변형 시, 변형의 기준점 변경

- **transform**의 속성들인 translate(이동), skew(기울기), scale(확대/축소), rotate(회전)은 특정 요소의 모양을 변형시키는 속성이며, transform-origin을 주지 않을 시 해당요소의 정 중앙을 기준으로 변형이 일어난다.
- transform-origin은 변형이 일어나는 기준점을 사용자 임의로 변경해 주는 기능이다.

transform-origin 사용법

transform-origin : left top;

방향 설정

transform-origin : 20px 30px;

고정 px 설정

transform-origin : 0% 100%;

퍼센트 설정

transform-origin의 수치 중 첫 번째는 가로 기준점, 두 번째는 세로 기준점이며. 각각의 수치 사이에는 콤마를 집어 넣지 않는 점 유의

01 1) CSS transform (2D변형)

C S S 3
기 본 문 법

C S S 3

INDEX

• 예제

```
<body>

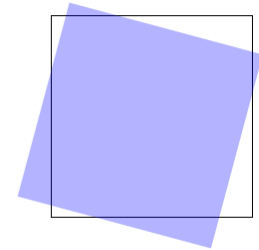
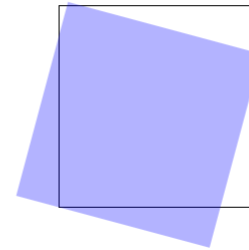
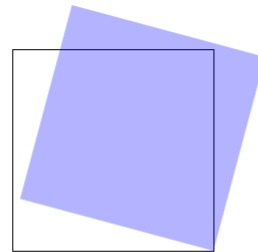
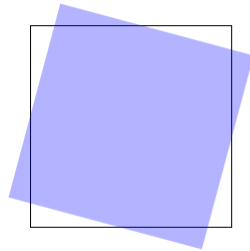
  <div> <p> </p> </div>
  <div> <p> </p> </div>
  <div> <p> </p> </div>
  <div> <p> </p> </div>

</body>
```

▲ html

```
* { margin:0; padding:0 }
div { width:200px; height:200px; margin:100px; border:1px solid #000;
      float:left; }
div p { width:100%; height:100%; background: blue; opacity: 0.3; }
div:nth-child(1):hover p { transform: rotate(15deg); }
div:nth-child(2):hover p { transform: rotate(15deg);
                           transform-origin: right bottom; }
div:nth-child(3):hover p { transform: rotate(15deg);
                           transform-origin: 20px 30px; }
div:nth-child(4):hover p { transform: rotate(15deg);
                           transform-origin: 30% 30%; }
```

▲ css



01

1) CSS transform (2D변형)

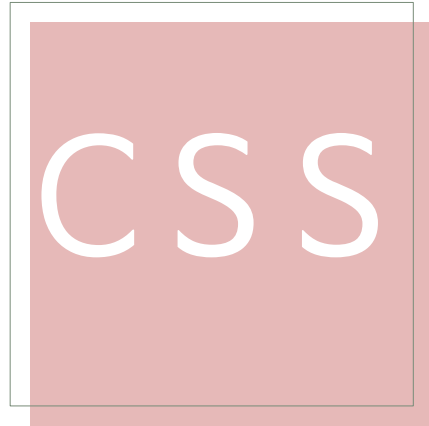
C S S 3
기 본 문 법

C S S 3

INDEX

05_CSS transform

1) CSS transform (3D변형)



01

1) CSS transform (2D 변형)

C S S 3
기 본 문 법

C S S 3

INDEX

핵심용어	용어설명
transform (3d 변형)	특정 요소의 모양을 3차원 공간에서 입체적으로 변형하기
perspective	원근감 설정하기
translateZ	z축으로 이동하기
rotateX , rotateY	3d 공간상에서 x축과 y축 기준으로 회전하기
perspective- origin	원근의 시작점을 x와 y축으로 선언 하는 것
transform- style	해당 요소의 모양을 변형 시, 변형의 기준점 변경
backface- visibility	객체의 뒷면이 보이지 않도록 하는 속성

01 perspective

• perspective : 원근감 설정하기

1. 기존 x축 y축에 z축 추가 시 평면에서 입체적 공간으로 변경.
2. 브라우저에 원근감을 설정해 주는 요소.

perspective 사용법

perspective : 원근감 수치;

어떤 물체를 가까이서 바라보면 해당 물체의 왜곡이 커지고 멀리서 바라보면 물체의 왜곡이 적어짐

perspective 의 수치가 적을수록 광각렌즈로 촬영한 것과 같이 왜곡의 효과가 커지며

수치가 커질수록 망원렌즈로 촬영한 것과 같이 왜곡이 줄어 듭

01 1) CSS transform (3D변형)

C S S 3
기 본 문 법

C S S 3

perspective



- **예제** tip: perspective는 3d 변형을 한 요소의 부모 태그에 설정해 주는 것이 일반적임.

```
<body>

  <div><p></p></div>
  <div><p></p></div>
  <div><p></p></div>

</body>
```

▲ html

```
* { margin:0; padding:0 }
div { width: 200px; height: 200px; margin: 100px; border: 1px solid #000;
      float: left; }
div p { transition: all 2s; }
div:hover p { width: 100%; height: 100%; background: blue; opacity: 0.3;
              transform: rotateX(50deg); }
div:nth-child(1) { perspective: 150px; }
div:nth-child(2) { perspective: 0px; }
div:nth-child(3) { perspective: 600px; }
```

▲ css

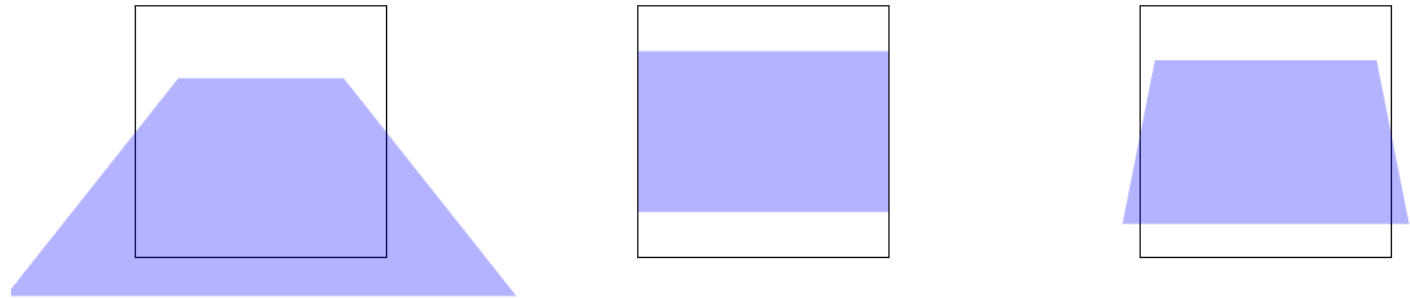
01

1) CSS transform (3D 변형)

C S S 3
기 본 문 법

C S S 3

perspective



02 | translateZ

• translateZ : z축으로 이동하기

1. transform(2d)의 translate : x(위)나 y(아래)로 평면상의 움직임
2. transform(3d)의 translate : z축으로의 이동은 브라우저를 바라보는 사용자 시점에서 앞 뒤로 움직임

translateZ 사용법

transform : translateZ(이동거리);

01

1) CSS transform (3D변형)

C S S 3
기 본 문 법

C S S 3

translateZ

- 사용법은 기존의 translate와 동일
- translateZ는 3d 공간에서의 z축. 즉 사용자가 브라우저를 바라보는 시점을 기준으로 앞 또는 뒤로 움직임

- **예제** tip: translateZ의 마이너스 값은 사용자 시점을 기준으로 뒤쪽으로 멀리 이동하며, 플러스 값은 사용자 쪽으로 가깝게 이동

```
<body>

  <div>
    <p>box1</p>
    <p>box2</p>
    <p>box3</p>
  </div>

</body>
```

▲ html

```
* { margin:0; padding:0 }
div { width: 600px; height: 400px; margin: 100px auto; border: 1px solid #000;
      position: relative; perspective: 1600px;}
div p { width: 100%; height: 100%; background: blue; opacity: 0.3;
        position: absolute; top: 0; left: 0; font: bold 60px arial; color: #fff;
      }
div p:nth-child(1) { transform: translateZ(-500px); }
div p:nth-child(2) { transform: translateZ(0px); }
div p:nth-child(3) { transform: translateZ(500px); }
```

▲ CSS



01

1) CSS transform (3D변형)

C S S 3
기 본 문 법

C S S 3

translateZ

03 rotateX, rotateY

- rotateX, rotateY : 3d 공간상에서 x축과 y축 기준으로 회전하기
 1. transform(2d)의 rotate : 평면상의 시계나 반 시계 방향으로 회전
 2. transform(3d)의 rotateX, rotateY : 3d공간에서 x축과 y축 기준으로 회전하기

translateZ 사용법

```
transform : rotateX(x축 회전각도deg);
transform : rotateY(y축 회전각도deg);
```

01 1) CSS transform (2D변형)

C S S 3
기 본 문 법

C S S 3

rotateX, rotateY

- 사용법은 기존의 rotate 와 동일
- 부모요소 perspective의 수치도 영향을 받음

- **예제** tip:부모에 perspective 속성이 있어야 함!

```
<body>

  <div><p></p></div>
  <div><p></p></div>

</body>
```

▲ html

```
* { margin:0; padding:0 }
div { width: 300px; height: 300px; margin: 100px; border: 1px solid #000;
      float: left; perspective: 700px; }
div p { transition: all 2s; }
div:hover p { width: 100%; height: 100%; background: blue; opacity: 0.3; }
div:nth-child(1) p { transform: rotateX(45deg); }
div:nth-child(2) p { transform: rotateY(45deg); }
```

▲ css

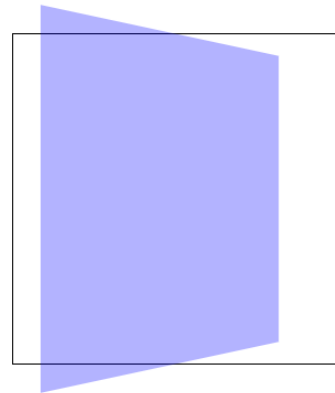
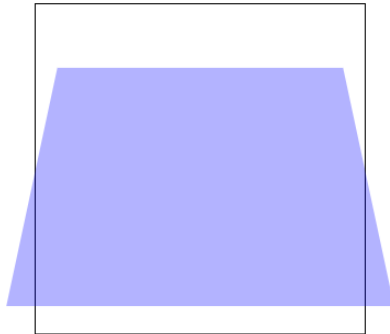
01

1) CSS transform (3D변형)

C S S 3
기 본 문 법

C S S 3

rotateX, rotateY



04 perspective-origin

- **perspective-origin**: 원근의 시작점을 x와 y축으로 선언 하는 것
 1. 3차원 공간에 소실점을 설정하는 것
 2. perspective-origin 속성은 반드시 perspective 속성과 함께 적용

perspective-origin 사용법

perspective-origin : left top;

포지션 값

perspective-origin : 20px 30px;

길이 값

perspective-origin : 50% 50%;

퍼센트 값

- 50% 50%에서 첫 번째 50%는 x축(수평)의 위치, 두 번째 50%는 y축(수직)
- x, y 모두 각각 50%이므로 포함된 모든 자식 객체는 정 중앙에 소실점을 갖는 원근 형태로 보여지게 됨

01 2) CSS transform (3D변형)

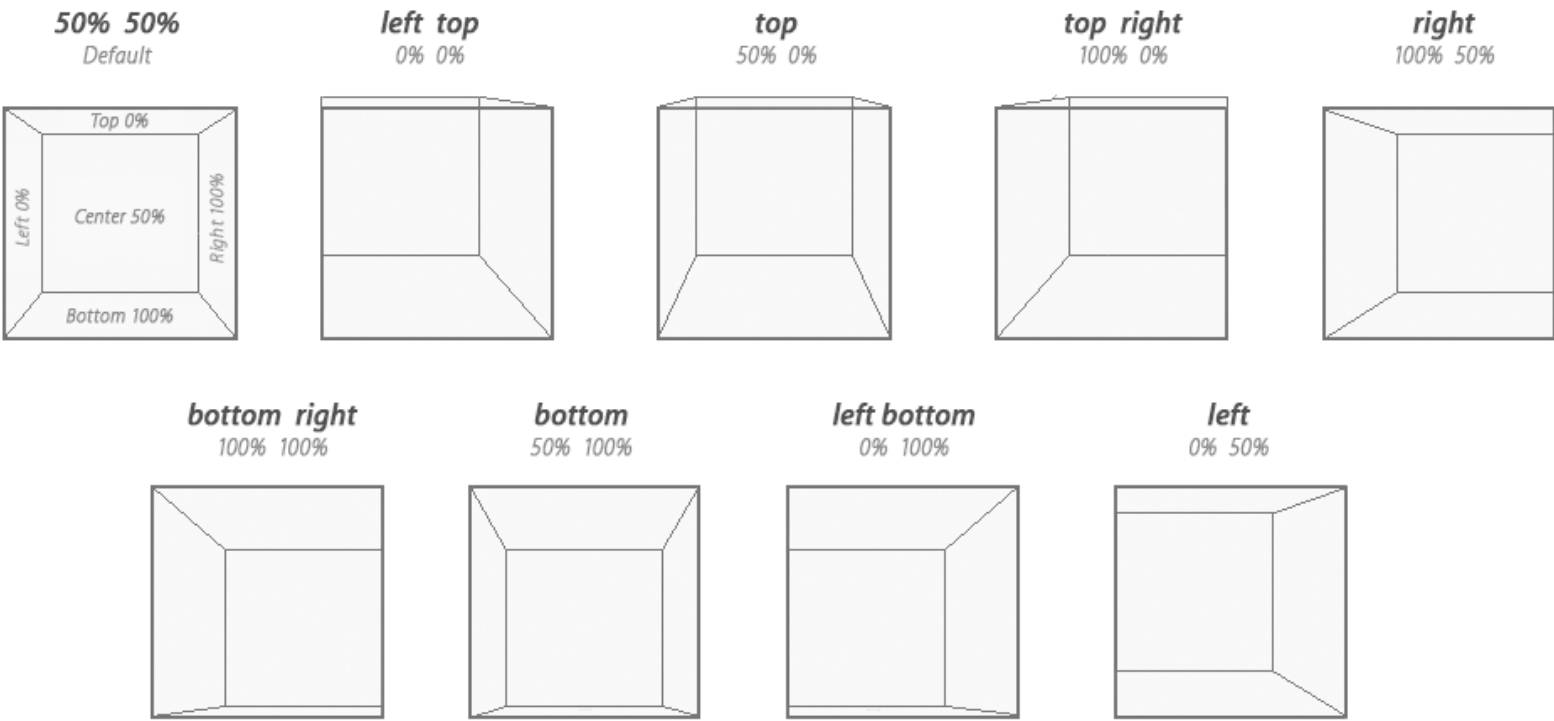
C S S 3
기 본 문 법

C S S 3

perspective-origin

- 예제 transform02.html /transform02.css

perspective-origin - 3D Transform



01 2) CSS transform (3D변형)

C S S 3
기 본 문 법

C S S 3

perspective-origin

05 transform-style

- **transform-style** : 자식 객체를 3차원 공간에서 어떻게 표현할 지 선택
 1. flat과 preserve-3d가 있으며, 기본 값은 flat
 2. 부모 요소에 flat을 선언하거나 혹은 아무것도 선언하지 않게 되면, 그 자식 요소는 3D 공간 값을 가지지 못하므로, transform: 속성(값) 가 선언되어 있어도 표현되지 않음

transform-style 사용법

transform-style: flat;

transform-style: preserve-3d;

01 2) CSS transform (3D변형)

C S S 3
기 본 문 법

C S S 3

transform-style

- transform-style을 주지 않으면 flat속성값을 가진다.

• 예제 transform-style.html

```
<body>
  <div class="container">
    <div class="parent">
      <div class="child">
        I'm your sun.
      </div>
    </div>
  </div>
</body>
```

▲ html

```
* { margin: 0; padding: 0 }
.container { perspective: 300px; }
.parent { margin: 0 auto; width: 300px; height: 300px;
  border: 1px solid #ddd; transform: rotateX(45deg);
  transform-style: preserve-3d; }
.child { margin-top: 40px; padding: 100px 0; width: 100%;
  background-color: #ddd; text-align: center;
  color: #999; transform: rotateX(-30deg); }
```

▲ CSS

01 2) CSS transform (3D변형)

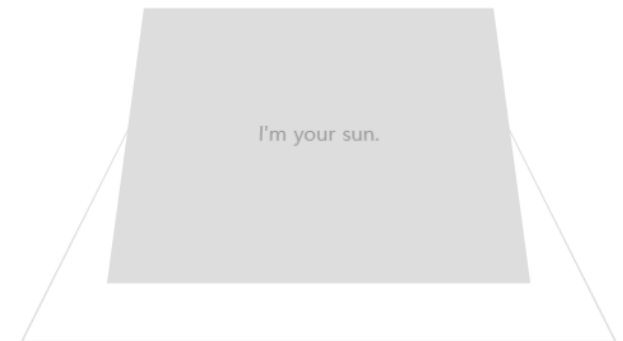
C S S 3
기 본 문 법

C S S 3

transform-style



flat



preserve-3d

06 backface-visibility

- **backface-visibility** : 객체의 뒷면이 보이지 않도록 하는 속성
 1. backface-visibility 속성이 선언되지 않은 기본 값은 visible
 2. visible은 실제 뒷면이 아니고 앞면의 뒤쪽이며, 뒷면으로 적용하려면 뒷면객체를 하나 더 만들어줘야 함

backface-visibility 사용법

backface-visibility: visible;

backface-visibility: hidden;

01 2) CSS transform (3D변형)

C S S 3
기 본 문 법

C S S 3

backface-visibility

- backface-visibility는 단순히 뒷면을 의미하는 것이 아니라, 객체의 보이지 않는 부분을 뜻한다는 점에 유의.

- 예제 backface-visibility01.html backface-visibility02.htm

01

2) CSS transform (3D변형)

CSS3

기본문법

CSS3

backface-visibility

