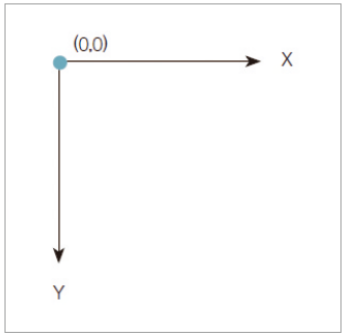
변형 알아보기

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



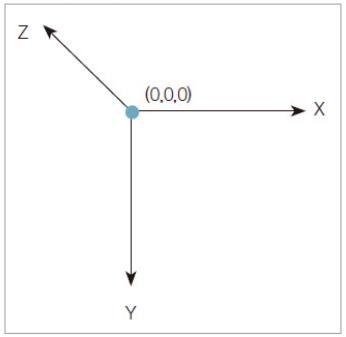
2차원 좌표계

표 11-1 2차원 변형 함수

종류	설명
translate(tx, ty)	지정한 크기만큼 x축, y축으로 이동합니다.
translateX(tx)	지정한 크기만큼 x축으로 이동합니다.
translateY(ty)	지정한 크기만큼 y축으로 이동합니다.
scale(sx, sy)	지정한 크기만큼 x축과 y축으로 확대·축소합니다.
scaleX(sx)	지정한 크기만큼 x축으로 확대·축소합니다.
scaleY(sy)	지정한 크기만큼 y축으로 확대·축소합니다.
rotate(각도)	지정한 각도만큼 회전합니다.
skew(ax, ay)	지정한 각도만큼 x축과 y축으로 왜곡합니다.
skewX(ax)	지정한 각도만큼 x축으로 왜곡합니다.
skewY(ay)	지정한 각도만큼 y축으로 왜곡합니다.

3차원 변형

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



3차원 좌표계

표 11-2 3차원 변형 함수

종류	설명
translate3d(tx, ty, tz)	지정한 크기만큼 x축, y축, z축으로 이동합니다.
translateZ(tz)	지정한 크기만큼 z축으로 이동합니다.
scale3d(sx, sy, sz)	지정한 크기만큼 x축, y축, z축으로 확대·축소합니다.
scaleZ(sz)	지정한 크기만큼 z축으로 확대·축소합니다.
rotate(rx, ry, 각도)	지정한 각도만큼 회전합니다.
rotate3d(rx, ry, rz, 각도)	지정한 각도만큼 회전합니다.
rotateX(각도)	지정한 각도만큼 x축으로 회전합니다.
rotateY(각도)	지정한 각도만큼 y축으로 회전합니다.
rotateZ(각도)	지정한 각도만큼 z축으로 회전합니다.
perspective(길이)	입체적으로 보일 수 있도록 깊잇값을 지정합니다.

변형 알아보기

translate 함수

지정한 방향으로 이동할 거리를 지정하면 해당 요소를 이동시킴

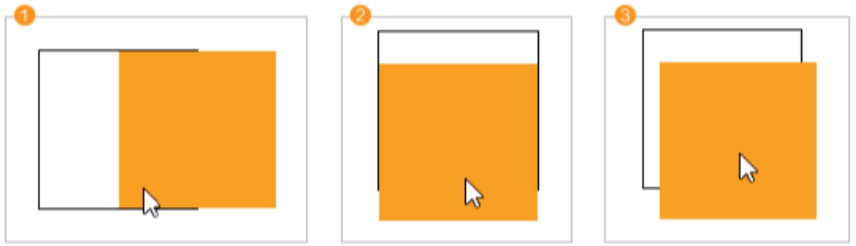
```
기본형  transform: translate(tx, ty)
        transform: translate3d(tx, ty, tz)
        transform: translateX(tx)
        transform: translateY(ty)
        transform: translateZ(tz)
```

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

Do it! translate( ) 함수를 사용해 웹 요소 이동하기 예제 파일 11\translate.html

```
(... 생략 ...)
  #movex:hover { transform: translateX(50px); }  /* x축으로 50px 이동 */ ❶
  #movey:hover { transform: translateY(20px); }  /* y축으로 20px 이동 */ ❷
  #movexy:hover { transform:  t              (10px, 20px); }  /* x축으로 10px, y축
으로 20px 이동 */ ❸
</style>
(... 생략 ...)
```

정답: translate



변형 알아보기

scale 함수

지정한 크기만큼 요소를 확대/축소

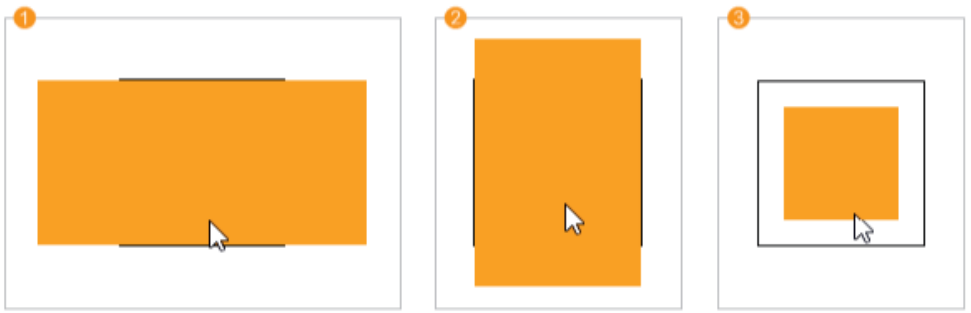
```
기본형  transform: scale(sx, sy)
        transform: scale3d(sx, sy, sz)
        transform: scaleX(sx)
        transform: scaleY(sy)
        transform: scaleZ(sz)
```

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

Do it! scale( ) 함수를 사용해 확대·축소하기 예제 파일 11\scale.html

```
(... 생략 ...)
<style>
  #scalex { transform: scaleX(2); }          /* x축으로 2배 확대 */ ❶
  #scaley { transform: scaleY(1.5); }        /* y축으로 1.5배 확대 */ ❷
  #scale { transform: s          (0.7); }    /* x, y축으로 0.7배 확대 */ ❸
</style>
(... 생략 ...)
```

정답: scale



변형 알아보기

rotate 함수

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

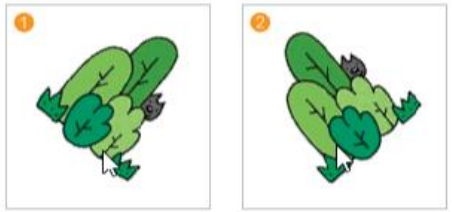
2차원 rotate( ) 함수

기본형 `transform: rotate(각도)`

Do it! 함수를 사용해 2차원에서 회전하기 예제 파일 11\rotate.html

```
(... 생략 ...)
<style>
  #rotate1:hover { transform: rotate(40deg); }        /* 오른쪽으로 40도 회전 */ ❶
  #rotate2:hover { transform: r          (-40deg); } /* 왼쪽으로 40도 회전 */ ❷
</style>
(... 생략 ...)
```

정답: rotate



변형 알아보기

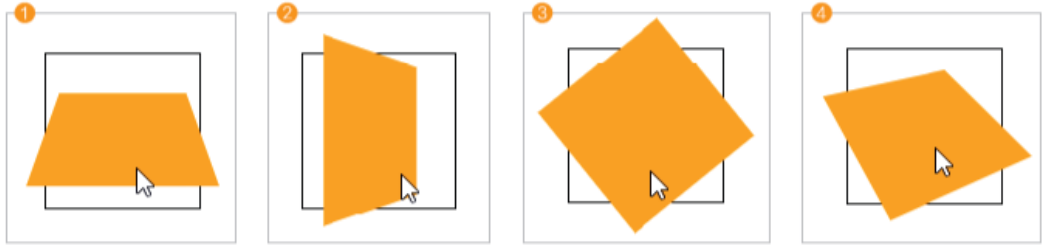
rotate 함수

3차원 rotate( ) 함수

```
기본형  transform: rotate(rx, ry, 각도)
        transform: rotate3d(rx, ry, rz, 각도)
        transform: rotateX(각도)
        transform: rotateY(각도)
        transform: rotateZ(각도)
```

perspective 속성

- 원근감을 표현하기 위해 사용하는 속성
- 원래 있던 위치에서 사용자가 있는 쪽으로 얼마나 이동하는지 나타냄.
- 값(픽셀 단위)은 0보다 커야 하며 값이 클수록 사용자로부터 멀어짐.
- perspective 속성은 변형하는 요소의 부모 요소에 정의해야 한다.



Do it! rotate( ) 함수를 사용해 3차원에서 회전하기 — 예제 파일 11\rotate3d.html

```
(... 생략 ...)
  .origin {
    ......
    perspective: 200px;  /* 원근감 추가 */
  }
  .origin > div {
    ......
    transition: all 3s;  /* 3초 동안 회전하도록 트랜지션 적용 */
  }
  #rotatex:hover { transform: rotateX(55deg); }        /* x축으로 55도 회전 */ ❶
  #rotatey:hover { transform: rotateY(55deg); }        /* y축으로 55도 회전 */ ❷
  #rotatez:hover { transform: r                (55deg); } /* z축으로 55도 회전 */ ❸
  #rotatexyz:hover {transform: rotate3d(2.5, 1.2, -1.5, 55deg); } /* x, y, z축에 방
향 벡터를 지정하고 55도 회전 */ ❹
</style>
......
<div class="origin">
  <div id="rotatex"></div>
</div>
<div class="origin">
  <div id="  r                "></div>
</div>
<div class="origin">
  <div id="rotatez"></div>
</div>
<div class="origin">
  <div id="rotatexyz"></div>
</div>
(... 생략 ...)
```

#rotatex, #rotatey, #rotatez, #rotatexyz의 부모 요소인 .origin에 perspective 속성을 적용합니다!

정답: rotateZ, rotateY

변형 알아보기

skew 함수

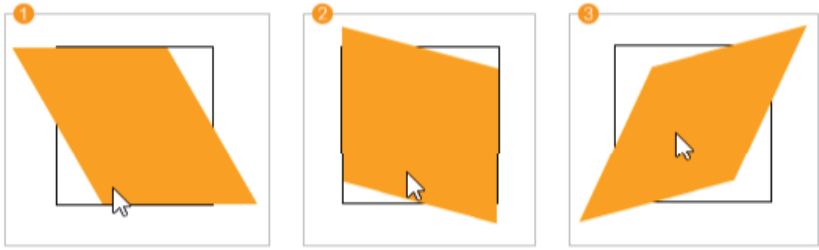
요소를 지정한 각도만큼 비틀어 왜곡

```
기본형  transform: skew(x각도, y각도)
        transform: skewX(x각도)
        transform: skewY(y각도)
```

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

Do it! skew( ) 함수를 사용해 도형 비틀기 예제 파일 11\skew.html

```
(... 생략 ...)
  #skewx:hover { transform: skewX(30deg); } /* x축 기준으로 30도 비틀기 */ ❶
  #skewy:hover { transform: skewY(15deg); } /* y축 기준으로 15도 비틀기 */ ❷
  #skewxy:hover { transform: skew(-25deg, -15deg); } /* x축 기준으로 -25도, y축 기준
으로 -15도 비틀기 */ ❸
</style>
(... 생략 ...)
```

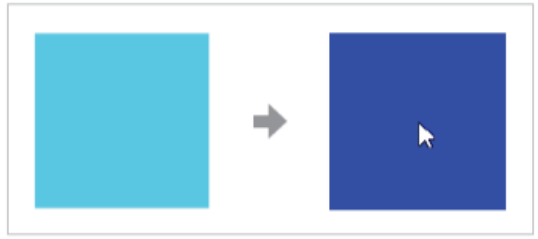


트랜지션 알아보기

트랜지션이란

웹 요소의 스타일 속성이 조금씩 자연스럽게 바뀌는 것

예) 하늘색 도형 위로 마우스를 올려놓으면 도형이 하늘색에서 파란색으로 바뀌고 마우스를 치우면 원래 배경 색으로 되돌아감.



예) 도형 위로 마우스를 올려놓으면 사각형의 테두리와 테두리색이 바뀌고 마우스를 치우면 원래 스타일로 되돌아감.

트랜지션의 속성

종류	설명
transition-property	트랜지션의 대상을 지정합니다.
transition-duration	트랜지션을 실행할 시간을 지정합니다.
transition-timing-function	트랜지션의 실행 형태를 지정합니다.
transition-delay	트랜지션의 지연 시간을 지정합니다.
transition	transition-property, transition-duration, transition-timing-function, transition-delay 속성을 한꺼번에 정합니다.

트랜지션 알아보기

transition-property 속성

- 트랜지션을 적용할 속성 선택
- 이 속성을 지정하지 않으면 모든 속성이 트랜지션 대상이 됨.

기본형 `transition-property: all ¦ none ¦ <속성 이름>`

종류	설명
all	all값을 사용하거나 transition-property를 생략할 경우 요소의 모든 속성이 트랜지션 대상이 됩니다. 기본값입니다.
none	트랜지션을 하는 동안 아무 속성도 바뀌지 않습니다.
속성 이름	트랜지션 효과를 적용할 속성을 지정합니다. 예를 들어 배경색만 바꿀 것인지, width값을 바꿀 것인지 원하는 대상만 골라 지정할 수 있습니다. 속성이 여럿일 경우 쉼표(,)로 구분하여 나열합니다.

```
transition-property: all;              /* 해당 요소의 모든 속성에 트랜지션 적용 */
transition-property: background-color; /* 해당 요소의 배경색에 트랜지션 적용 */
transition-property: width, height;    /* 해당 요소의 너비와 높이에 트랜지션 적용 */
```

transition-duration 속성

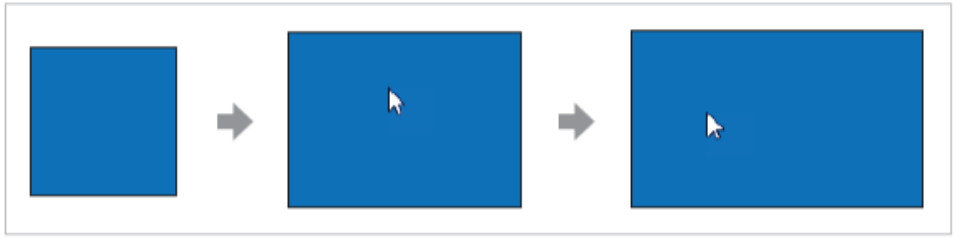
- 트랜지션 진행 시간 지정
- 시간 단위는 초(seconds) 또는 밀리초(milliseconds)
- 트랜지션이 여러 개라면 쉼표(,)로 구분해 진행 시간 지정

기본형 `transition-duration: <시간>`

Do it! 트랜지션 대상과 진행 시간 지정하기 — 예제 파일 11\tr-1.html

```
(... 생략 ...)
  .box {
    ......
    transition-property: width, height;   /* 트랜지션 대상 - 너비, 높이 */
    transition-du                : 2s, 1s;  /* 트랜지션 시간 - 2초, 1초 */
  }
  .box:hover {
    width: 200px;
    height: 120px;
  }
</style>
(... 생략 ...)
```

정답: duration



트랜지션 알아보기

transition-timing-function 속성

트랜지션의 시작과 중간, 끝에서의 속도 지정

기본형 `transition-timing-function: linear ¦ ease ¦ ease-in ¦ ease-out ¦ ease-in-out ¦ cubic-bezier(n, n, n, n)`

종류	설명
ease	처음에는 천천히 시작하고 점점 빨라지다가 마지막엔 천천히 끝냅니다. 기본값입니다.
linear	시작부터 끝까지 똑같은 속도로 진행합니다.
ease-in	느리게 시작합니다.
ease-out	느리게 끝냅니다.
ease-in-out	느리게 시작하고 느리게 끝냅니다.
cubic-bezier(n, n, n, n)	베지에 함수를 정의해서 사용합니다. 이때 n값은 0~1 사이만 사용할 수 있습니다.

transition-delay 속성

- 트랜지션이 언제부터 시작될지 지연 시간 지정
- 시간 단위는 초(seconds) 또는 밀리초(milliseconds). 기본값 0

기본형 `transition-delay: <시간>`

transition 속성

- 트랜지션 관련 속성을 한꺼번에 지정
- 시간 값 속성이 2개이므로, 앞에 오는 시간값은 transition-duration, 뒤에 오는 시간 값은 transition-delay 속성으로 간주

기본형 `transition: <transition-property값> ¦ <transtion-duration값> ¦ <transition-timing-function값> ¦ <transition-delay값>`

Do it! 트랜지션 속성 한꺼번에 지정하기 예제 파일 11\tr-2.html

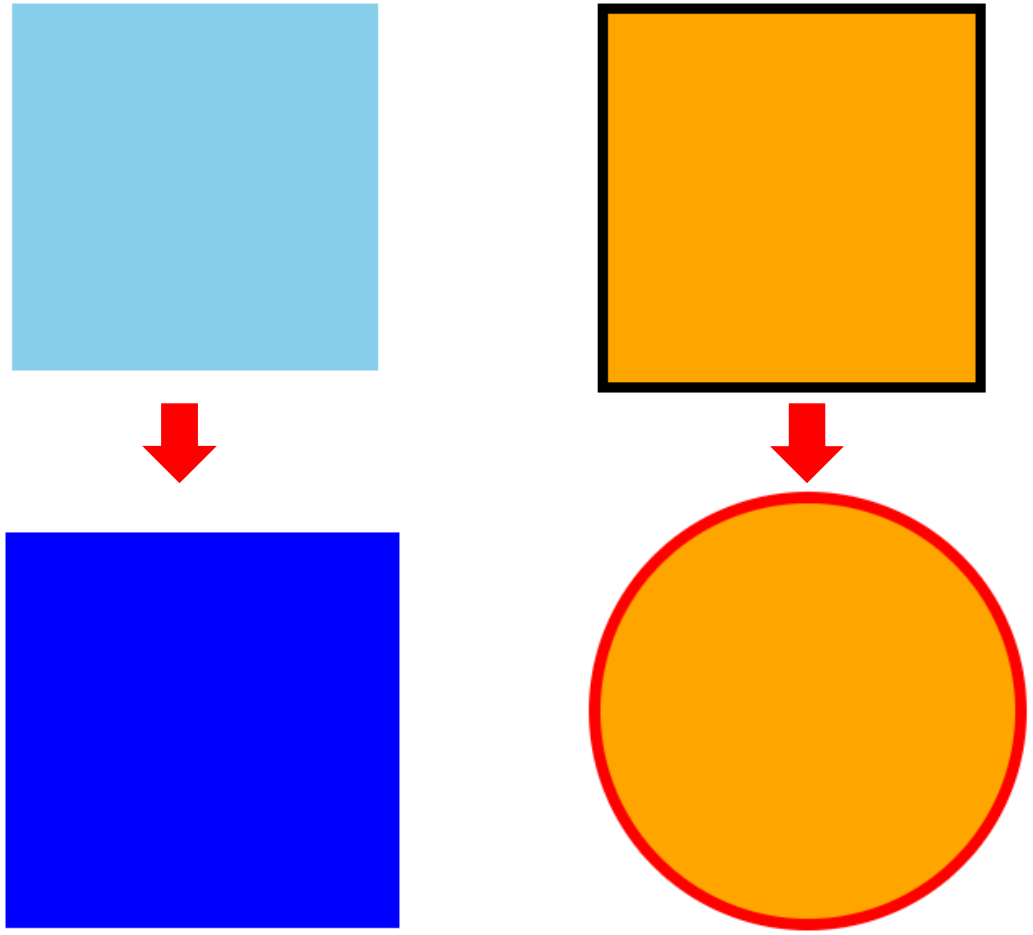
```
(... 생략 ...)
  .box {
    ......
    transition: 2s ease-in;
  }
  .box:hover {
    width: 200px;
    height: 200px;
    background-color: #f50;
    transform: rotate(270deg);
  }
</style>
(... 생략 ...)
```

transition-property: 기본값 all
transition-duration: 2s
transition-timing-function: ease-in
transition-delay: 기본값 0

여기에 있는 속성이 모두 트랜지션 대상

Quiz. tr-ex.html

:hover 했을 때



Quiz-1. newproduct.html & newproduct.css

신상품 목록

상품 1

상품 1 설명 텍스트

가격 : 12,345원

상품 2

상품 2 설명 텍스트

가격 : 12,345원

상품 3

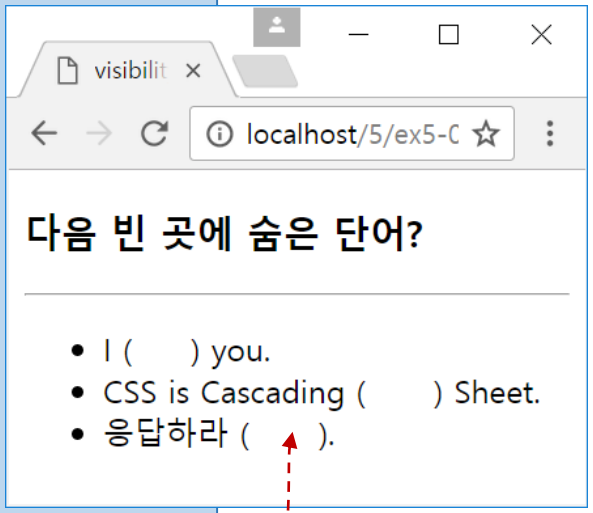
상품 3 설명 텍스트

가격 : 12,345원

- 전체 폭 사이즈는 1000px
- 목록으로 작성

visibility로 텍스트 숨기기 (visibility.html)

```
<!DOCTYPE html>
<html>
<head><title>visibility 프로퍼티</title>
<style>
span {
   visibility : hidden;
}
</style>
</head>
<body>
<h3>다음 빈 곳에 숨은 단어?</h3>
<hr>
<ul>
   <li>I (<span>love</span>) you.
   <li>CSS is Cascading (<span>Style</span>) Sheet.
   <li>응답하라 (<span>1988</span>).
</ul>
</body>
</html>
```



visibility : hidden;
공간은 차지하지만
텍스트는 보이지 않음

overflow

overflow CSS 단축 속성은 요소의 콘텐츠가 너무 커서 요소의 블록 서식 맥락에 맞출 수 없을 때의 처리법을 지정합니다.

Visible : 콘텐츠를 자르지 않으며 안쪽 여백 상자 밖에도 그릴 수 있습니다.

Hidden : 콘텐츠를 안쪽 여백 상자에 맞추기 위해 잘라냅니다. 스크롤바를 제공하지 않고, 스크롤할 방법(드래그, 마우스 휠 등)도 지원하지 않습니다. 코드를 사용해 스크롤할 수는 있으므로 (`offsetLeft` (en-US) 속성 설정 등), 이 상태의 요소도 스크롤 컨테이너입니다.

clip : `hidden`과 마찬가지로, 콘텐츠를 안쪽 여백 상자에 맞춰 자릅니다. 그러나 `clip`은 코드를 사용한 스크롤링도 방지하므로 어떠한 스크롤도 불가능합니다. 이 상태의 요소는 스크롤 컨테이너가 아니며, 새로운 블록 서식 문맥도 생성하지 않습니다. 서식 문맥이 필요하다면 `display:flow-root`을 사용할 수 있습니다.

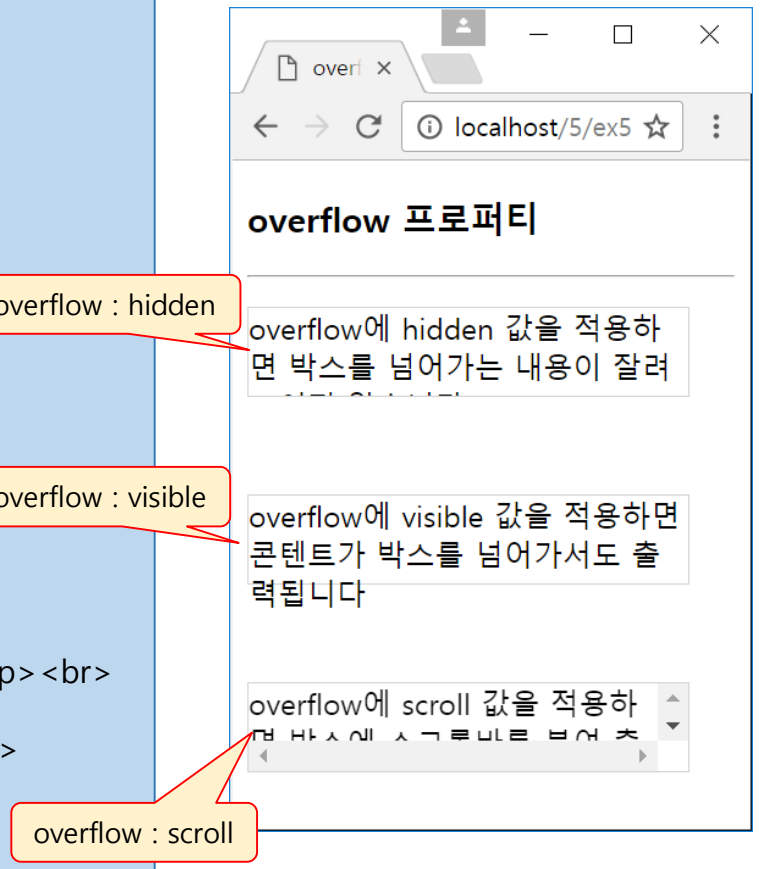
Scroll : 콘텐츠를 안쪽 여백 상자에 맞추기 위해 잘라냅니다. 브라우저는 콘텐츠를 실제로 잘라냈는지 여부를 따지지 않고 항상 스크롤바를 노출하므로 내용의 변화에 따라 스크롤바가 생기거나 사라지지 않습니다. 프린터는 여전히 넘친 콘텐츠를 출력할 수도 있습니다.

Auto : 사용자 에이전트가 결정합니다. 콘텐츠가 안쪽 여백 상자에 들어간다면 `visible`과 동일하게 보이나, 새로운 블록 서식 문맥을 생성합니다. 데스크톱 브라우저의 경우 콘텐츠가 넘칠 때 스크롤바를 노출합니다.

overlay : `auto`와 동일하게 동작하지만, 스크롤바가 공간을 차지하는 대신 콘텐츠 위에 위치합니다. Webkit(Safari 등)과 Blink(Chrome 또는 Opera 등) 기반 브라우저만 지원합니다.

overflow 프로퍼티 활용(overflow.html)

```
<!DOCTYPE html>
<html>
<head>
<title>overflow 프로퍼티</title>
<style>
p {
    width : 15em;
    height : 3em;
    border : 1px solid lightgray;
}
.hidden { overflow : hidden; }
.visible { overflow : visible; }
.scroll { overflow : scroll; }
</style>
</head>
<body>
<h3>overflow 프로퍼티</h3>
<hr>
<p class="hidden">overflow에 hidden 값을 적용하면
      박스를 넘어가는 내용이 잘려 보이지 않습니다.</p><br>
<p class="visible">overflow에 visible 값을 적용하면
      콘텐트가 박스를 넘어가서도 출력됩니다</p><br>
<p class="scroll">overflow에 scroll 값을 적용하면
      박스에 스크롤바를 붙여 출력합니다.</p>
</body>
</html>
```



z-index

CSS z-index 속성은 위치 지정 요소와, 그 자손 또는 하위 플렉스 아이템의 Z축 순서를 지정합니다. 더 큰 z-index 값을 가진 요소가 작은 값의 요소 위를 덮습니다.

위치 지정 요소(position이 static 외의 다른 값인 요소)의 박스에 대해, z-index 속성은 다음 항목을 지정합니다.

1. 현재 쌓임 맥락에서 자신의 위치.
2. 자신만의 쌓임 맥락 생성 여부.

```
/* 키워드 값 */
 z-index: auto;

/* 박스가 새로운 쌓임 맥락을 생성하지 않습니다. 현재 쌓임 맥락에서의 위치는 부모 요소와 동일합니다. */

/* <integer> 값 */
 z-index: 0;
 z-index: 3;
 z-index: 289;
 z-index: -1; /* 음수 값으로 우선순위를 낮출 수 있음 */
```

z-index로 카드 쌓기(z-index.html)

```
<!DOCTYPE html>
<html><head><title>z-index 프로퍼티</title>
<style>
div { position : absolute; }
img { position : absolute; }
#spadeA { z-index : -3; left : 10px; top : 20px; }
#spade2 { z-index : 2; left : 40px; top : 30px; }
#spade3 { z-index : 3; left : 80px; top : 40px; }
#spade7 { z-index : 7; left : 120px; top : 50px; }
</style></head>
<body>
<h3>z-index 프로퍼티</h3>
<hr>
<div>
<img id="spadeA" src="images/som1.jpg"
        width="100" height="140" alt="스페이드A">
<img id="spade2" src="images/som2.jpg"
        width="100" height="140" alt="스페이드2">
<img id="spade3" src="images/som3.jpg"
        width="100" height="140" alt="스페이드3">
<img id="spade7" src="images/som4.jpg"
        width="100" height="140" alt="스페이드7">
</div>
</body>
</html>
```

애니메이션 알아보기

CSS와 애니메이션

- 웹 요소에 애니메이션 추가
- 애니메이션을 시작해 끝내는 동안 원하는 곳 어디서든 스타일을 바꾸며 애니메이션을 정의할 수 있다.
- 키프레임(keyframe) : 애니메이션 중간에 스타일이 바뀌는 지점

애니메이션 관련 속성

종류	설명
@keyframes	애니메이션이 바뀌는 지점을 지정합니다.
animation-delay	애니메이션의 시작 시간을 지정합니다.
animation-direction	애니메이션을 종료한 뒤 처음부터 시작할지, 역방향으로 진행할지 지정합니다.
animation-duration	애니메이션의 실행 시간을 지정합니다.
animation-iteration-count	애니메이션의 반복 횟수를 지정합니다.
animation-name	@keyframes로 설정해 놓은 중간 상태를 지정합니다.
animation-timing-function	키프레임의 전환 형태를 지정합니다.
animation	animation 속성을 한꺼번에 묶어서 지정합니다.

애니메이션 알아보기

@keyframes 속성

- 애니메이션의 시작과 끝을 비롯해 상태가 바뀌는 지점을 설정
- '이름'으로 애니메이션 구별

```
기본형 @keyframes  <이름> {
         <선택자> { <스타일> }
       }
```

- @keyframes의 선택자에서 속성값이 바뀌는 지점을 가리킴
- 시작 위치는 0%, 끝 위치 100%로 놓고 위치 지정
- 시작과 끝 위치만 사용한다면 from, to 키워드 사용 가능

animation-name 속성

- 어떤 애니메이션을 사용할지 구별
- @keyframes 속성에서 만든 애니메이션 '이름'을 사용

```
기본형 animation-name: <키프레임 이름> | none
```

animation-duration 속성

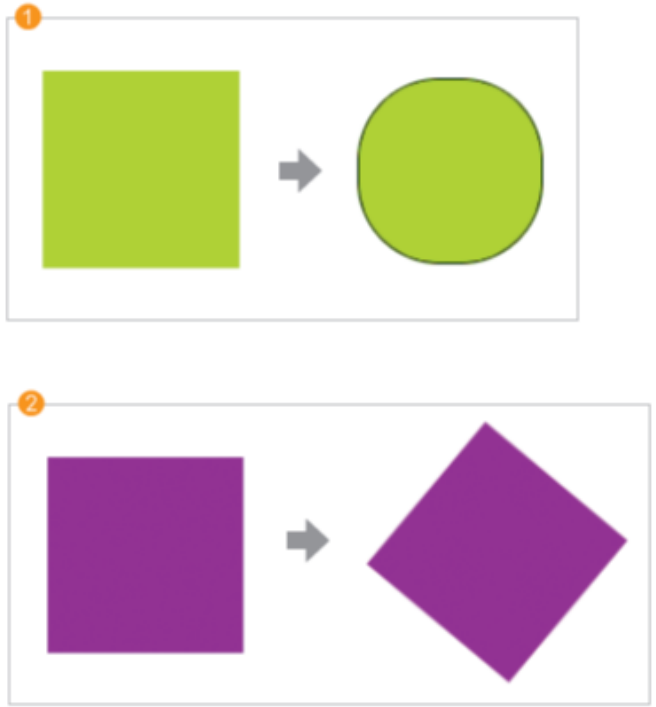
- 애니메이션 실행 시간 설정. 기본값 0
- 사용 가능한 값은 초(s)나 밀리초(ms)

```
기본형 animation-duration: <시간>
```

애니메이션 알아보기

Do it! 애니메이션의 지점, 이름, 실행 시간 적용하기 — 예제 파일 11\ani-1.html

```
(... 생략 ...)
  #box1 {
    background-color: #4cff00;       /* 연두색 박스 */
    border: 1px solid transparent;   /* 1px짜리 투명 테두리 */
    animation-name: shape;           /* 애니메이션 지정 */
    animation-duration: 3s;          /* 애니메이션 실행 시간 */
  }
  #box2 {
    background-color: #8f06b0;       /* 보라색 박스 */
    border: 1px solid transparent;   /* 1px짜리 투명 테두리 */
    animation-name: rotate;          /* 애니메이션 지정 */
    animation-duration: 3s;          /* 애니메이션 실행 시간 */
  }
① @keyframes shape {
    from { border: 1px solid transparent; }  /* 1px짜리 투명 테두리에서 */
    to {
      border: 1px solid #000; /* 검은색 테두리로 변경 */
      border-radius: 50%;     /* 테두리를 둥글게 변경 */
    }
  }
② @keyframes rotate {
    from { transform: rotate(0deg); }  /* 0도에서 시작해서 */
    to { transform: rotate(45deg); }   /* 45도까지 회전하기 */
  }
</style>
......
<div class="box" id="box1"></div>
<div class="box" id="box2"></div>
(... 생략 ...)
```



애니메이션 알아보기

animation-direction 속성

애니메이션이 끝난 후 원래 위치로 돌아가거나 반대 방향으로 실행하도록 지정

기본형 `animation-direction: normal ¦ alternate`

종류	설명
normal	애니메이션을 끝까지 실행하면 원래 위치로 돌아갑니다. 기본값입니다.
alternate	애니메이션을 끝까지 실행하면 왔던 방향으로 되돌아가면서 애니메이션을 실행합니다.

animation-iteration-count 속성

애니메이션 반복 횟수 지정하기

기본형 `animation-iteration-count: <숫자> ¦ infinite`

종류	설명
숫자	애니메이션의 반복 횟수를 정합니다.
infinite	애니메이션을 무한 반복합니다.

Do it! 무한 반복하는 애니메이션 만들기 예제 파일 11\ani-2.html

```
(... 생략 ...)
  #box1 {
    background-color: #4cff00;
    border: 1px solid transparent;
    animation-name: shape;
    animation-duration: 3s;
    animation-iteration-count: infinite;          /* 애니메이션 무한 반복 */
  }
  #box2 {
    background-color: #8f06b0;
    border: 1px solid transparent;
    animation-name: rotate;
    animation-duration: 3s;
    animation-iteration-count: i          ;  /* 애니메이션 무한 반복 */
  }
  @keyframes shape { ..... }
  @keyframes rotate { ..... }
  }
</style>
(... 생략 ...)
```

애니메이션 알아보기

animation-timing-function 속성

애니메이션 속도 곡선 지정

animation 속성

- 여러 개의 애니메이션 속성을 하나의 속성으로 줄여서 사용
- 지정하지 않은 속성은 기본 값 사용.
- animation-duration 속성 값은 반드시 지정해야 함. (기본값이 0)

Do it! 애니메이션 2개를 한꺼번에 지정하기 예제 파일 11\ani-3.html

```
(... 생략 ...)
  .box {
    width: 100px;
    height: 100px;
    margin: 60px auto;
    animation: rotate 1.5s infinite, background 1.5s infinite alternate;
  }

  @keyframes rotate { /* 0도 -> x축 -180도 회전 -> y축 -180도 회전 */
    from { transform: perspective(120px) rotateX(0deg) rotateY(0deg); }
    50% { transform: perspective(120px) rotateX(-180deg) rotateY(0deg); }
    to { transform: perspective(120px) rotateX(-180deg) rotateY(-180deg); }
  }

  @k                    background {
    from { background-color: red; }   /* 시작 배경색은 빨강 */
    50% { background-color: green; }  /* 중간(50%) 배경색은 초록 */
    to { background-color: blue; }    /* 마지막(100%) 배경색은 파랑 */
  }
</style>
(... 생략 ...)
```

rotate 애니메이션 정의, 1.5초 진행, 무한 반복

background 애니메이션 정의, 1.5초 진행, 무한 반복

정답: @keyframes