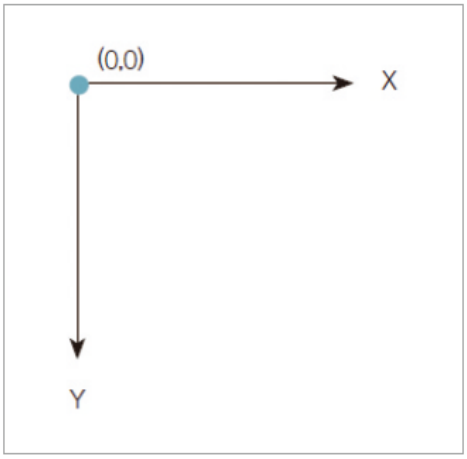
13장 CSS3와 애니메이션

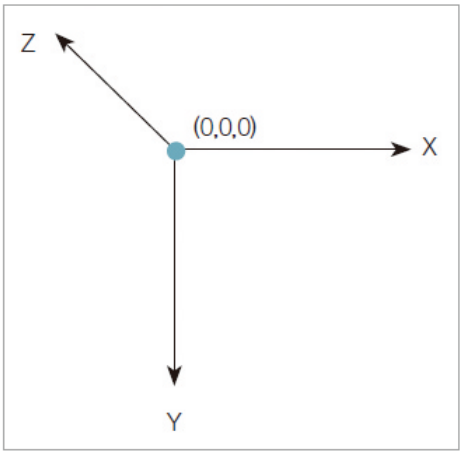
13.1 변형

(1) 2차원 변형과 3차원 변형

- 2차원 변형(2D transform): 단순히 수평이나 수직으로 이동하고 회전하는 것을 말한다.
- 3차원 변형(3D transform): x축과 y축에 원근감을 주는 z축을 추가해 변형시키는 것을 말한다.



2차원 좌표계



3차원 좌표계

(2) transform과 변형 함수

- 기본형: transform: 변형함수;
- 예) .photo { transform: translate(50px,100px); }
- 2차원 변형함수: translate(x,y), translateX(tx), translateY(ty) 등
- 3차원 변형함수: translate3d(tx,ty,tz), rotateX(각도), rotateY(각도) 등

(3) translate 변형 함수 - 요소 이동시키기

- 기본형: transform:translate(tx,ty)
  transform:translate3d(tx,ty,tz)

(4) scale 변형 함수 - 요소 확대/축소하기

- 기본형: transform:scale(sx,sy)
  transform:scale3d(sx,sy,sz)

(5) rotate 변형 함수 - 요소 회전하기

■ 기본형: transform:rotate(각도)
transform:rotate3d(rx,ry,rz,각도)

(6) skew 변형 함수 - 요소를 비틀어 왜곡하기

■ 기본형: transform:skew(ax,ay)
transform:skewX(ax)
transform:skewY(ay)

13.2 변형과 관련된 속성들

(1) transform-origin 속성 - 변형 기준점 설정하기

■ 기본형: transform-origin: 〈x축〉 〈y축〉 〈z축〉 ¦ initial ¦ inherit ;

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.

(2) perspective, perspective-origin 속성 - 원근감 표현하기

■ 기본형: perspective: 〈크기〉 ¦ none;
perspective-origin: 〈x축 값〉 ¦ 〈y축 값〉

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

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.

(3) transform-style 속성 - 3D 변형 적용하기

■ 기본형: transform-style: flat ¦ preserve-3d

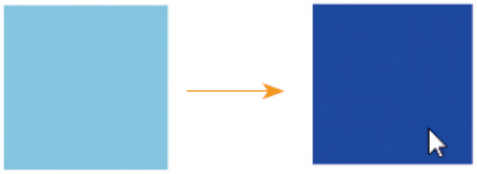
(4) backface-visibility 속성 - 요소의 뒷면 표시하기

■ 기본형: backface-visibility: visible ¦ hidden

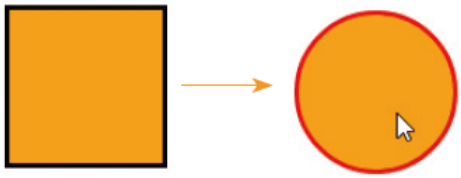
13.3 트랜지션

(1) 트랜지션이란?

- 웹 요소의 배경 색이 바뀌거나 도형의 테두리가 원형으로 바뀌는 것처럼 스타일 속성이 바뀌는 것을 말한다.



사각형의 배경 색이 바뀌는 트랜지션



사각형의 모양과 테두리 색이 바뀌는 트랜지션

(2) transition-property 속성 - 트랜지션을 적용할 속성 지정하기

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

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

(3) transition-duration 속성 - 트랜지션 진행 시간 지정하기

- 기본형: transition-duration: <시간>

(4) transition-timing-function 속성 - 트랜지션 속도 곡선 지정하기

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

속성 값	설명
linear	시작부터 끝까지 똑같은 속도로 트랜지션을 진행합니다.
ease	처음에는 천천히 시작하고 점점 빨라지다가 마지막에는 천천히 끝냅니다. 기본 값입니다.*
ease-in	시작을 느리게 합니다.
ease-out	느리게 끝냅니다.
ease-in-out	느리게 시작하고 느리게 끝냅니다.
cubic-bezier(n,n,n,n)	베지에 함수를 직접 정의해 사용합니다. n에서 사용할 수 있는 값은 0~1입니다.

[13/tr-function.html]

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Transition timing function</title>
<!--[if lt IE 9]>
<script src="html5shiv.js"></scirpt>
<![endif]-->
<style>
#ex div {
          float: left;
          width: 100px;
          height: 50px;
          margin: 5px 10px;
          padding: 5px;
          color: white;
          background-color: #006aff;
          border-radius: 5px;
          text-align: center;
          font-weight: bold;
}

#ex:hover div {
          height: 400px;
}

#ex .ease {
          -moz-transition: 3s ease;
          -o-transition: 3s ease;
          -webkit-transition: 3s ease;
          transition: 3s ease;
}

#ex .linear {
          -webkit-transition: 3s linear;
          -moz-transition: 3s linear;
          -o-transition: 3s linear;
          transition: 3s linear;
}

#ex .ease-in {
          -webkit-transition: 3s ease-in;
          -moz-transition: 3s ease-in;
          -o-transition: 3s ease-in;
          transition: 3s ease-in;
}

#ex .ease-out {
          -webkit-transition: 3s ease-out;
          -moz-transition: 3s ease-out;
          -o-transition: 3s ease-out;
          transition: 3s ease-out;
}

#ex .ease-in-out {
          -webkit-transition: 3s ease-in-out;
          -moz-transition: 3s ease-in-out;
          -o-transition: 3s ease-in-out;
          transition: 3s ease-in-out;
}
</style>
</head>

<body>
```

```
        <div id="ex">
                <div class="ease">ease</div>
                <div class="linear">linear</div>
                <div class="ease-in">ease-in</div>
                <div class="ease-out">ease-out</div>
                <div class="ease-in-out">ease-in-out</div>
        </div>
</body>
</html>
```

(5) transition-delay 속성 - 지연 시간 설정하기

■ 기본형: transition-delay: <시간>

(6) transition 속성 - 트랜지션 속성 한꺼번에 표기하기

■ 기본형: transition: <transition-property 값> ¦ <transition-duration 값> ¦ <transition-timing-function 값> ¦ <transition-delay 값>

13.4 애니메이션

(1) CSS와 애니메이션

■ CSS 애니메이션은 시작해 끝나는 동안 원하는 곳 어디서든 스타일을 바꾸며 애니메이션을 정의할 수 있다.

■ 키프레임(keyframe) : 애니메이션 중간에 스타일이 바뀌는 지점

■ CSS 애니메이션에서 사용하는 속성

속성	설명
@keyframes	애니메이션이 바뀌는 지점을 설정합니다.
animation-delay	애니메이션 지연 시간을 지정합니다.
animation-direction	애니메이션 종료 후 처음부터 시작할지, 역방향으로 진행할지를 지정합니다.
animation-duration	애니메이션 실행 시간을 설정합니다.
animation-fill-mode	애니메이션이 종료되었거나 지연되어 애니메이션이 실행되지 않는 상태일 때 요소의 스타일을 지정합니다.
animation-iteration-count	애니메이션 반복 횟수를 지정합니다.
animation-name	@keyframes로 설정해 놓은 중간 상태의 이름을 지정합니다.
animation-play-state	애니메이션을 멈추거나 다시 시작합니다.
animation-timing-function	애니메이션의 속도 곡선을 지정합니다.
animation	animation 하위 속성들을 한꺼번에 묶어 지정합니다.

(2) @keyframes 속성 - 애니메이션 지점 설정하기

■ 기본형: @keyframes <이름> {
<선택자> { <스타일> }

}

(3) animation-name 속성 - 애니메이션 이름 지정하기

■ 기본형: animation-name: <키프레임 이름> ¦ none

(4) animation-duration 속성 - 애니메이션 실행 시간 설정하기

■ 기본형: animation-duration: <시간>

(5) animation-direction 속성 - 애니메이션 방향 지정하기

■ 기본형: animation-direction: normal ¦ alternate

(6) animation-iteration-count 속성 - 반복 횟수 지정하기

■ 기본형: animation-iteration-count: <숫자> ¦ inifinite

(7) animation-timing-function 속성 - 애니메이션 속도 곡선 지정하기

■ 기본형: animation-timing-function: linear ¦ ease ¦ ease-in ¦ ease-out ¦ ease-in-out ¦ cublic-bezier(n,n,n,n)

(8) animation 속성 - 애니메이션 관련 속성 한꺼번에 표기하기

■ 기본형: animation: <animation-name> ¦ <animation-duration> ¦ <animation-timing-function> ¦ <animation-delay> ¦ <animation-iteration-count> ¦ <animation-direction>

[13/animation.html]

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Transform</title>
<style>
          .box {
                    width: 60px;
                    height: 60px;
                    margin: 60px;
                    -webkit-animation: rotate 1.5s infinite, background 1.5s infinite alternate;
                    -moz-animation: rotate 1.5s infinite, background 1.5s infinite alternate;
                    -o-animation: rotate 1.5s infinite, background 1.5s infinite alternate;
                    animation: rotate 1.5s infinite, background 1.5s infinite alternate;
          }

          @-webkit-keyframes rotate {
                    from {
                              transform: perspective(120px) rotateX(0deg) rotateY(0deg);
```

```
		}

		50% {
			transform: perspective(120px) rotateX(-180deg) rotateY(0deg);
		}

		to {
			transform: perspective(120px) rotateX(-180deg) rotateY(-180deg);
		}
	}

	@-moz-keyframes rotate {
		from {
			transform: perspective(120px) rotateX(0deg) rotateY(0deg);
		}

		50% {
			transform: perspective(120px) rotateX(-180deg) rotateY(0deg);
		}

		to {
			transform: perspective(120px) rotateX(-180deg) rotateY(-180deg);
		}
	}

	@-o-keyframes rotate {
		from {
			transform: perspective(120px) rotateX(0deg) rotateY(0deg);
		}

		50% {
			transform: perspective(120px) rotateX(-180deg) rotateY(0deg);
		}

		to {
			transform: perspective(120px) rotateX(-180deg) rotateY(-180deg);
		}
	}

	@keyframes rotate {
		from {
			transform: perspective(120px) rotateX(0deg) rotateY(0deg);
		}
		50% {
			transform: perspective(120px) rotateX(-180deg) rotateY(0deg);
		}
		to {
			transform: perspective(120px) rotateX(-180deg) rotateY(-180deg);
		}
	}

	@-webkit-keyframes background {
		from {
			background: red;
		}

		50% {
			background-color: green;
		}

		to {
			background-color: blue;
		}
	}
```

```
        @-moz-keyframes background {
                from {
                        background: red;
                }

                50% {
                        background-color: green;
                }

                to {
                        background-color: blue;
                }
        }

        @-o-keyframes background {
                from {
                        background: red;
                }

                50% {
                        background-color: green;
                }

                to {
                        background-color: blue;
                }
        }
        @keyframes background {
                from {
                        background: red;
                }

                50% {
                        background-color: green
                }

                to {
                        background-color: blue;
                }
        }
</style>
</head>

<body>
        <div class="box">   </div>
</body>
</html>
```