

# 10강. css3와 애니메이션

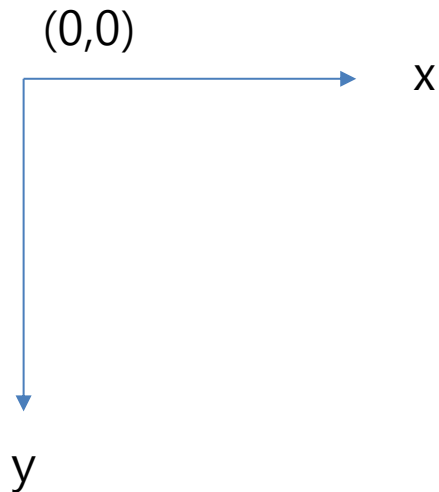


# 변형(transform)

## 변형(transform)

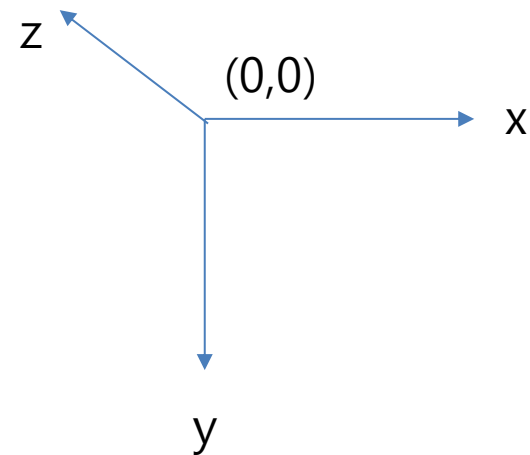
### 2차원 변형

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



### 3차원 변형

- X축과 y축에 원근감 추가
- Z축은 앞뒤로 이동, 보는 사람 쪽으로 다가올수록 값이 더 커짐



# 변형(transform)

## transform(변형) 속성

함수	설명
rotate()	각도만큼 웹 요소를 시계 방향이나 시계 반대 방향으로 회전 <code>transform: rotate(45deg)</code>
translate()	지정한 방향으로 이동할 거리를 지정하면 해당 요소를 이동 <code>transform: translateX(50px), transform: translateY(20px)</code>
scale()	지정한 크기만큼 요소를 확대/축소 <code>transform: scaleX(1.5), scaleY(1.5), transform: scale(2)</code>



# 변형(transform)

## transform(변형) 속성

```
<h2>변형전 개체</h2>
<div class="a">Hello World!</div>
<br>

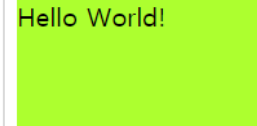
<h2>transform: rotate(20deg):</h2>
<div class="b">Hello World!</div>
<br>

<h2>transform: translateX(50px):</h2>
<div class="c">Hello World!</div>
<br>

<h2>transform: scaleY(1.5):</h2>
<div class="d">Hello World!</div>
```

### 변형전 개체

Hello World!



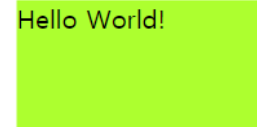
### transform: rotate(20deg):

Hello World!




### transform: translateX(50px):

Hello World!



### transform: scaleY(1.5):

Hello World!



# 변형(transform)

## transform(변형) 속성

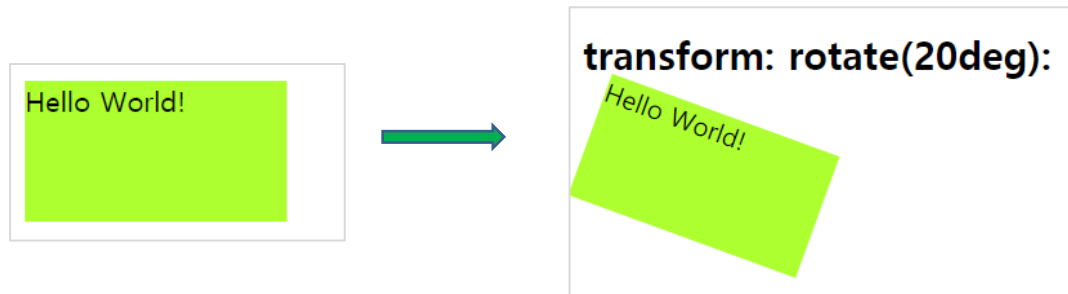
```
div.a{
  width: 150px;
  height: 80px;
  background-color: ■greenyellow;
}
div.b{
  width: 150px;
  height: 80px;
  background-color: ■greenyellow;
  transform: rotate(20deg);
}
div.c{
  width: 150px;
  height: 80px;
  background-color: ■greenyellow;
  transform: translateX(50px);
}
div.d{
  width: 150px;
  height: 80px;
  background-color: ■greenyellow;
  transform: scaleY(1.5);
}
```



# 변형(transform)

transform(변형) : rotate()

hover 선택자 적용하기



```
<h2>transform: rotate(20deg):</h2>
<div class="a">Hello World!</div>
<br>

<h2>transform: translateX(50px):</h2>
<div class="b">Hello World!</div>
<br>

<h2>transform: scaleY(1.5):</h2>
<div class="c">Hello World!</div>
```

```
div.a{
  width: 150px;
  height: 80px;
  background-color: greenyellow;
}
div.a:hover{
  transform: rotate(20deg);
}
```



# 변형(transform)

transform(변형) – translate()

hover 선택자 적용하기

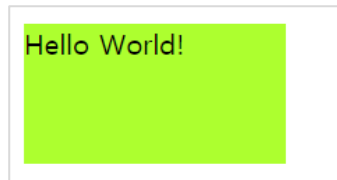


```
div.b{  
  width: 150px;  
  height: 80px;  
  background-color: greenyellow;  
}  
div.b:hover{  
  transform: translateX(50px);  
}
```

# 변형(transform)

transform(변형) – scale(1.5)

hover 선택자 적용하기



**transform: scaleY(1.5):**



```
div.c{  
  width: 150px;  
  height: 80px;  
  background-color: greenyellow;  
}  
div.c:hover{  
  transform: scaleY(1.5);  
}
```





# 브라우저 접두사

## 브라우저 접두사(prefix)

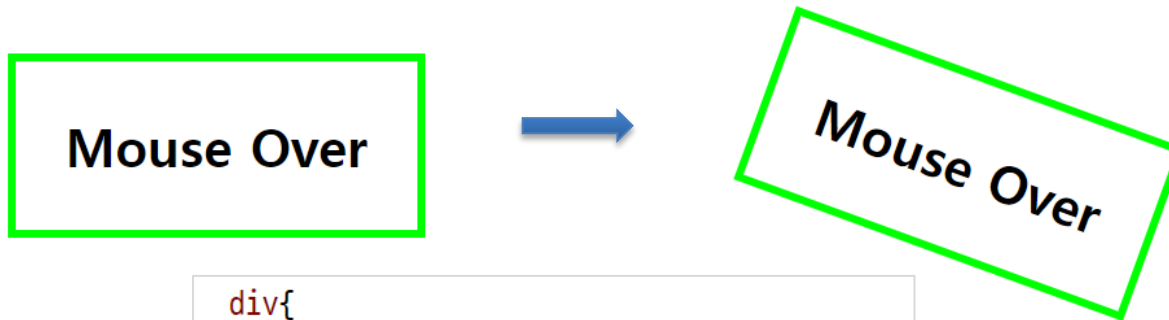
표준 규약이 아닌 속성들은 브라우저에 따라 다른 방식으로 지원되기 때문에 속성 이름 앞에 접두사를 붙여 브라우저별로 구분한다.

속성	설명
-webkit-	웹킷 방식 브라우저용(사파리, 크롬 등)
-moz-	게코 방식 브라우저용(모질라, 파이어폭스 등)
-o-	오페라 브라우저
-ms-	마이크로소프트 인터넷 익스플로러



# 브라우저 접두사

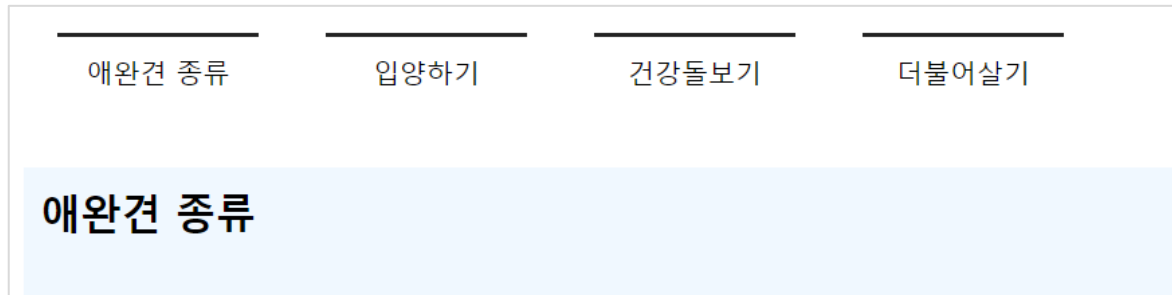
브라우저 접두사(prefix)



```
div{  
  border: 5px solid ■ #00ff00;  
  padding: 30px;  
  position: absolute;  
  top: 50px;  
  left: 100px;  
}  
div:hover{  
  -webkit-transform: rotate(20deg);  
  -ms-transform: rotate(20deg);  
  -moz-transform: rotate(20deg);  
  transform: rotate(20deg);  
}  
h1{margin: 0;}
```

# 메뉴 버튼 확대하기

- transform: scale()을 이용하여 메뉴 확대하기



```
<header>
  <nav>
    <ul>
      <li><a href="#">애완견 종류</a></li>
      <li><a href="#">입양하기</a></li>
      <li><a href="#">건강돌보기</a></li>
      <li><a href="#">더불어살기</a></li>
    </ul>
  </nav>
</header>
<section>
  <h1>애완견 종류</h1>
</section>
```



# 메뉴 버튼 확대하기

navi-tr.css

```

*{margin: 0; padding: 0;}
header{height: 60px;}
/* 메뉴 스타일 */
nav ul{list-style: none; padding: 10px 20px;}
▼ nav ul li{
    /* transform 속성 사용을 위해 크기를 만들어줌 */
    width: 100px;
    height: 30px;
    display: inline-block;
    margin: 10px 20px;
    border-top: 3px solid ■ #222;
    padding: 10px;
    text-align: center;
}
nav ul li a{text-decoration: none; color: ■ black;}
nav ul li:hover{background-color: □ #eee; transform: scale(1.1);}
/* section 스타일 */
section{background-color: □ aliceblue; height: 300px; padding: 10px; margin: 20px;}

```



# 트랜지션

## 트랜지션이란

웹 요소의 속성 - 크기나 배경색-이 **일정한 시간** 동안 조금씩 자연스럽게 바뀌는 것을 말한다.

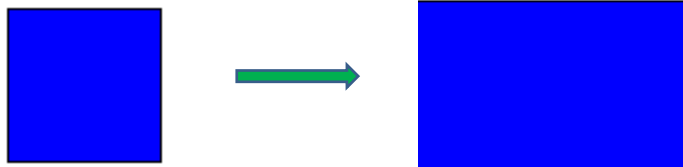
속성 값	설명
transition-property	트랜지션 대상을 설정합니다.
transition-duration	트랜지션 진행 시간을 설정합니다.
transition-timing-function	트랜지션 속도 곡선을 설정합니다.
transition-delay	트랜지션 지연시간을 설정합니다.
transition	트랜지션 속성을 한꺼번에 설정합니다.



# 트랜지션

## transition-duration 속성

- 시간 단위는 초(seconds) 또는 밀리초(milliseconds) – 2s, 100ms
- 트랜지션이 여러 개라면 쉼표(,)로 구분해서 진행시간 지정
- 모든 속성에 동일한 시간을 지정하려면 예) **transition**: all 2s;



```
<body>  
  <div class="tr1"></div>  
</body>
```

# 트랜지션

## transition-duration 속성

```
.tr1{
  width: 100px;
  height: 100px;
  border: 1px solid ■#000;
  background: ■#0000ff;
  position: absolute;
  top: 100px;
  left: 100px;

  transition-property: width, height;
  transition-duration: 1s, 2s;

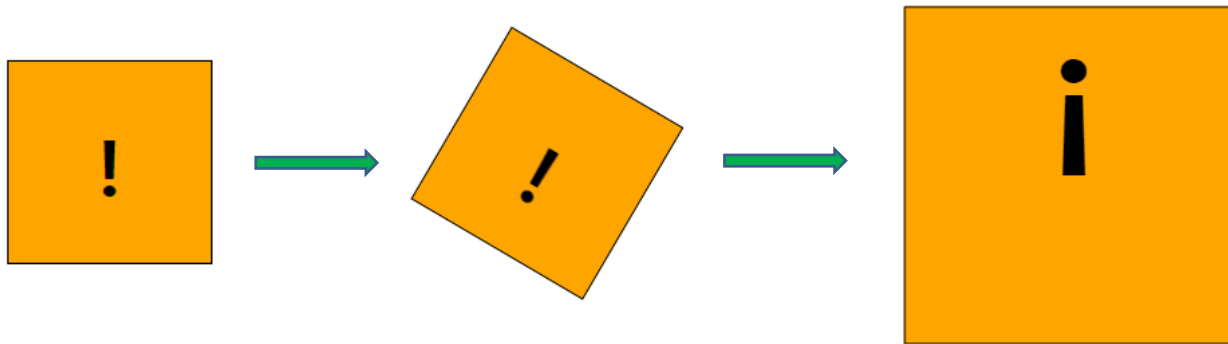
  /*transition: widht, height, 1s, 2s; /* 한번에 속성 설정 - 코머로 구분 */
  /*transition: all 2000ms; 모든 속성에 대해 시간을 똑같이 설정 */
}

.tr1:hover{
  width: 200px;
  height: 120px;
}
```



# 트랜지션

정해진 시간 동안 속성과 변형(transform) 변화



```
<body>  
  <div class="tr1"><h1>!</h1> </div>  
</body>
```



# 트랜지션

## 정해진 시간 동안 속성과 변형(transform) 변화

```
.tr1{
  position: absolute;
  top: 100px;
  left: 100px;
  width: 150px;
  height: 150px;
  background-color: orange;
  font-size: 1.5em;
  text-align: center;

  transition-property: width, height, transform /*font-size*/;
  /*transform 함수 사용 가능*/
  transition-duration: 2s, 2s, 5s;
}

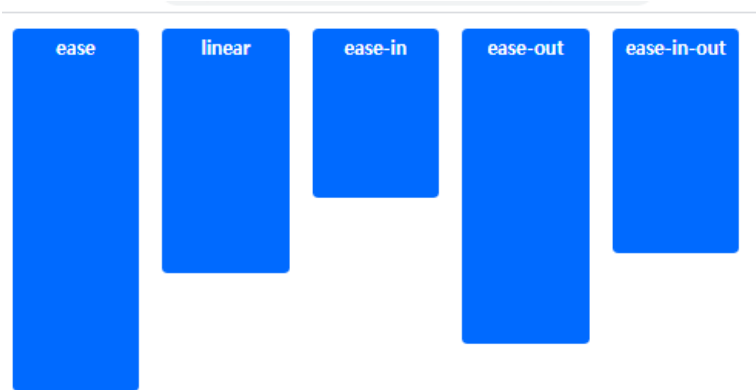
.tr1:hover{
  width: 200px;
  height: 200px;
  transform: rotate(180deg);
  /* font-size: 3em; */
}
```

시간을 대응할때는 반드시 콤마로 구분한다.



# 트랜지션

## Transition-timing-function 속성 – 트랜지션 속도 곡선 지정하기



```
<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>
```

```
#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 .ease {transition: 3s ease;}
#ex .linear{transition:3s linear;}
#ex .ease-in{transition:3s ease-in;}
#ex .ease-out{transition:3s ease-out;}
#ex .ease-in-out{transition:3s ease-in-out;}
```



# 트랜지션

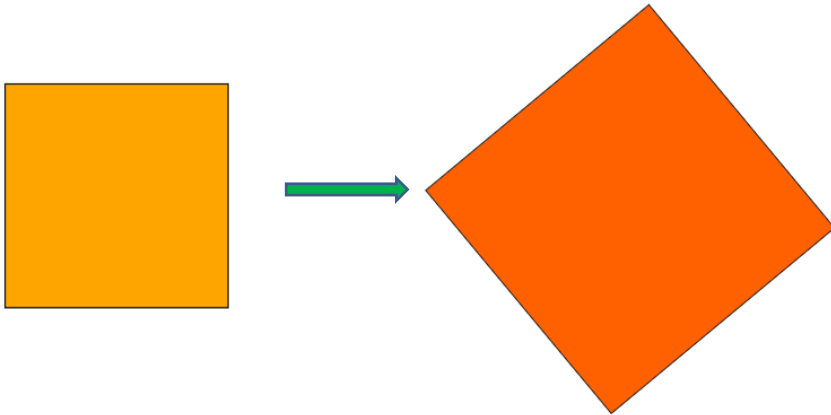
## transition-timing-function 속성

트랜지션의 시작과 중간, 끝에서의 속도 지정

속성 값	설명
linear	시작부터 끝까지 똑같은 속도로 트랜지션을 진행
ease	처음에는 천천히 시작하고 점점 빨라지다가 마지막에는 천천히 끝남(기본값)
ease-in	시작을 느리게 함.
ease-out	느리게 끝남
ease-in-out	느리게 시작하고 느리게 끝남

# 트랜지션

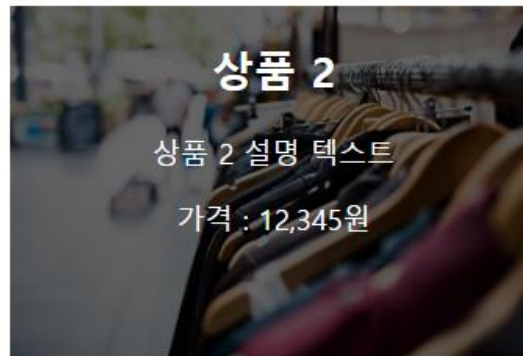
## transition 예제



```
.box{  
  width: 100px;  
  height: 100px;  
  border: 1px solid #222;  
  background-color: yellow;  
  margin: 100px;  
  /*transition-property: width, height, transform;  
  transition-timing-function: ease-in;  
  transition-duration: 2s;*/  
  transition: all 2s ease-in;  
  /* all에 2s 적용, ease-in-시작을 느리게 */  
}  
.box:hover{  
  width: 200px;  
  height: 200px;  
  background-color: orange;  
  transform: rotate(270deg);  
}
```

# 상품 페이지에 가격 떠오르게 만들기

## 신상품 목록



# 상품 페이지에 가격 떠오르게 만들기

```
<h1>신상품 목록</h1>
<div id="container">
  <ul class="prod-list">
    <li>
      
      <div class="caption">
        <h1>상품 1</h1>
        <p>상품 1 설명 텍스트</p>
        <p>가격 : 12,345원</p>
      </div>
    </li>
    <li>
      
      <div class="caption">
        <h1>상품 2</h1>
        <p>상품 2 설명 텍스트</p>
        <p>가격 : 12,345원</p>
      </div>
    </li>
  </ul>
</div>
```



# 상품 페이지에 가격 떠오르게 만들기

```
*{margin:0; padding:0;}
#container{width: 1000px; margin: 30px auto;}
h1{text-align: center;}
.prod-list{list-style: none; margin-top: 30px}
.prod-list li{display: inline-block; position: relative; overflow: hidden; margin:5px 10px;}
.prod-list li:nth-of-type(1){margin-left: 20px;}
.prod-list img{float: left;}
.prod-list .caption{width: 300px; height: 200px; background: ■ rgba(0, 0, 0, 0.6);
  position: absolute; top: 200px; transition: all 0.6s; }
.prod-list li:hover .caption{transform: translateY(-200px); /*설명글이 위로 200px 이동*/}
.prod-list .caption h2{ padding: 20px 0 10px 0;}
.prod-list .caption h2, p{color: □ #fff; text-align: center; line-height: 2em;}
```



# 애니메이션

## CSS와 애니메이션

- 애니메이션을 시작해 끝내는 동안 원하는 곳 어디서든 스타일을 바꾸며 애니메이션을 정의할 수 있다.
- **키프레임(keyframes):** 애니메이션 중간에 스타일이 바뀌는 지점

속성 값	설명
@keyframes	애니메이션이 바뀌는 지점을 설정합니다.
animation-name	@keyframes로 설정해 놓은 중간 상태의 이름을 지정함
animation-duration	애니메이션 지연 시간을 지정합니다.(반듯이 들어감)
animation-direction	애니메이션 종료 후 처음부터 시작할지, 역방향으로 진행할지를 지정합니다.(normal / alternate)
animation-iteration-count	애니메이션 반복 횟수를 지정함(기본 1번, infinite-무한)
animation	애니메이션 속성들을 한꺼번에 묶어 지정함





# 애니메이션

## @keyframes 속성

- 애니메이션의 시작과 끝을 비롯해 **상태가 바뀌는 지점**을 설정
- **이름**으로 애니메이션 구별

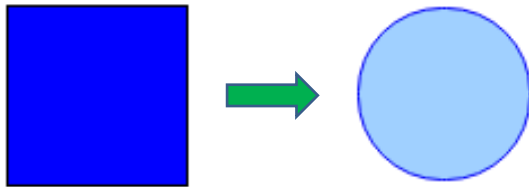
```
@keyframes 이름{  
  from{  
    변경전 속성  
  }  
  to{  
    변경후 속성  
  }  
}
```

- 시작 위치는 0%, 끝 위치 100%로 놓고 지정
- 시작과 끝 위치만 사용한다면 from, to 키워드 사용 가능



# 애니메이션

## @keyframes 속성 – 애니메이션 지점 설정하기



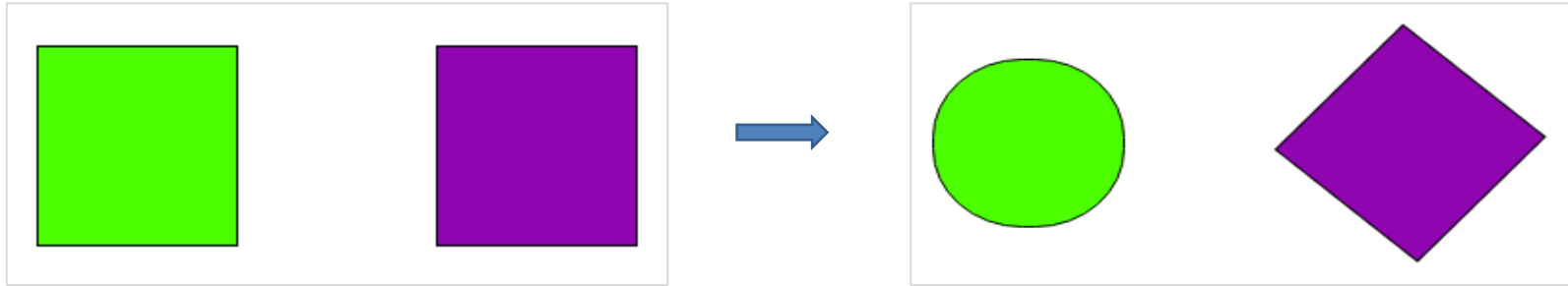
```
div {  
  width: 100px;  
  height: 100px;  
  background-color: blue;  
  animation-name: change-bg;  
  animation-duration: 3s;  
}
```

```
@keyframes change-bg {  
  from {  
    background-color: blue;  
    border: 1px solid black;  
  }  
  to {  
    background-color: #a5d6ff;  
    border: 1px solid blue;  
    border-radius: 50%;  
  }  
}
```

# 애니메이션

## animation-name 속성

@keyframes 속성에서 만든 애니메이션 이름을 사용



# 애니메이션

```
div{
  width: 100px;
  height: 100px;
  border: 1px solid ■#222;
  margin: 50px;
  float: left;
}
#box1{
  background-color: □azure;
  animation-name: shape;
  animation-duration: 3s;
  animation-iteration-count: 3;
  /* animation: shape 3s 3; */
}
#box2{
  background-color: ■blueviolet;
  animation-name: rotate;
  animation-duration: 2s;
  animation-iteration-count: infinite;
  /* animation: rotate 2s infinite; */
}
```

```
@keyframes shape {
  from{
    border: 1px solid ■#222;
  }
  to{
    border: 1px solid ■#222;
    border-radius: 50%;
  }
}
@keyframes rotate {
  from{
    transform: rotate(0deg);
  }
  50%{
    transform: rotate(45deg);
  }
  to{
    transform: rotate(90deg);
  }
}
```

# 애니메이션

## 무한 반복하는 애니메이션 만들기

CSS3 Animation



CSS3 Animation

```
<div class="box">  
  <h3>CSS3 Animation</h3>  
</div>
```



```
.box {  
  margin-top: 50px;  
  margin-left: 100px;  
  padding: 20px;  
  height: 60px;  
  animation-name: moving;  
  animation-duration: 3s;  
  animation-direction: normal;  
  animation-iteration-count: infinite;  
}
```

```
animation: moving 2s alternate infinite
```



# 애니메이션

## 무한 반복하는 애니메이션 만들기

```
@keyframes moving {  
  from {  
    width: 200px;  
    background:  #faef7c;  
    opacity: 0.5;  
    transform: scale(0.5) rotate(15deg);  
  }  
  
  to {  
    width: 400px;  
    background:  #ff9400;  
    opacity: 1;  
    transform: scale(1) rotate(0deg);  
  }  
}
```



# 애니메이션

## animation 속성

- 여러 개의 애니메이션 속성을 하나의 속성으로 줄여서 사용
- 특히, animation-duration 속성은 반드시 지정해야 함.



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

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

```
@keyframes background {  
  from { background: ■ red;}  
  50% { background-color: ■ green;  
  to { background-color: ■ blue;}  
}
```

