



# CHAPTER 09 CSS3 변형과 애니메이션

변형과 애니메이션에 관한 CSS3 고급 내용 살펴보기

## 9.1 변형 속성 기본

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS3 Transition Basic</title>
  <style>
    .box {
      width: 100px; height: 100px;
      background-color: orange;
    }
  </style>
</head>
<body>
  <div class="box"></div>
</body>
</html>

<style>
  .box {
    width: 100px; height: 100px;
    background-color: orange;
  }

  .box:hover {
```

```
    width: 200px;
    height: 300px;
  }

  .box:active {
    background-color: red;
  }
</style>

.box {
  width: 100px; height: 100px;
  background-color: orange;

  -ms-transition-duration: 2s;
  -moz-transition-duration: 2s;

  -webkit-transition-duration: 2s;
  transition-duration: 2s;
}
```

- HTML 페이지 구성
- 스타일 사용: hover 상태 선택자와 active 상태 선택자 사용
- 마우스를 올리면 곧바로 크기가 변경되고 클릭하면 곧바로 색 변경됨
- transition-duration 속성을 사용으로 변형 속성 제공
- 우측 이미지: 서서히 변경



## 9.2 변형 속성

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS3 Transition Basic</title>
  <style>

    </style>
</head>
<body>

</body>
</html>

<body>
  <h1>CSS3 Transition Graph</h1>
  <div id="graph">
    <div class="bar"></div>
    <div class="bar"></div>
    <div class="bar"></div>
    <div class="bar"></div>
    <div class="bar"></div>
  </div>
</body>

#graph {
  width: 610px;
  border: 3px solid black;
}

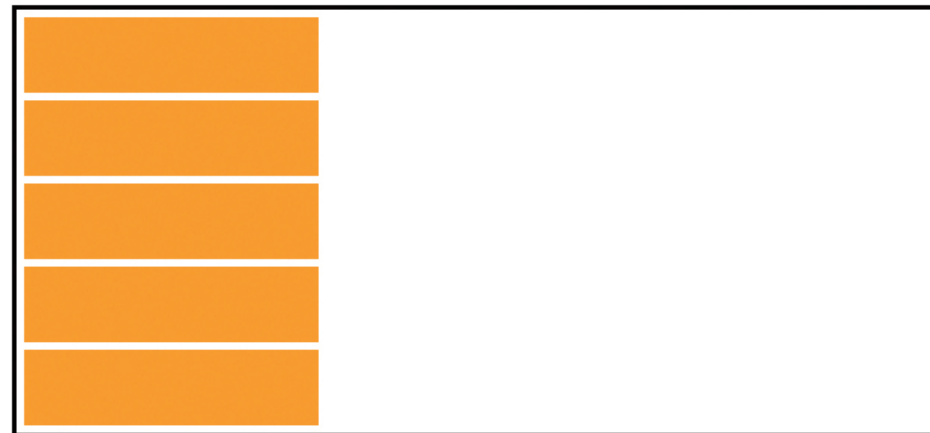
.bar {
  width: 10px; height: 50px;
  background-color: orange;
  margin: 5px;

  transition-duration: 5s;
}

#graph:hover > .bar {
  width: 600px;
}
```

- body 태그 구성
- 각각의 .bar 태그에 변형 속성 사용
- transition-duration 속성 사용:  
#graph 태그에 마우스 올릴 경우  
5초 동안 애니메이션 작동

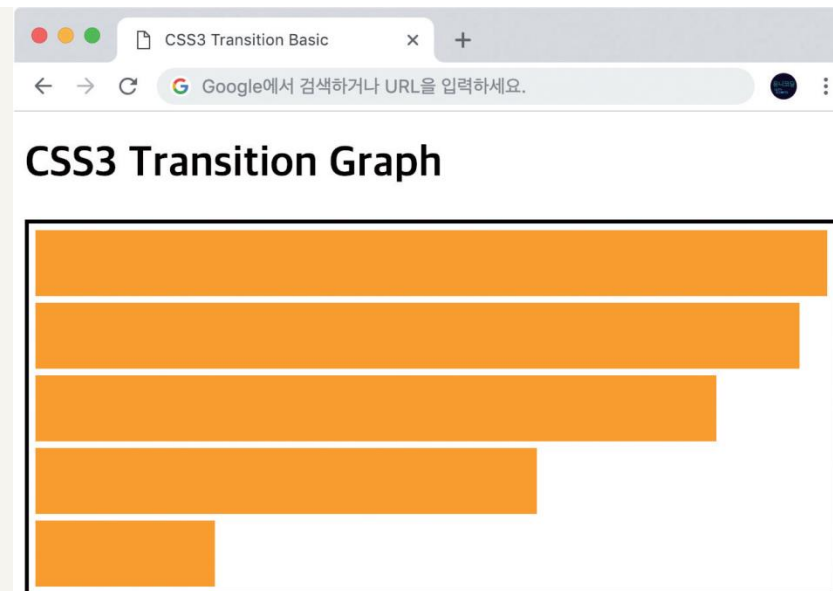
### CSS3 Transition Graph



속성 이름	설명
transition	모든 transition 속성을 한 번에 사용
transition-delay	이벤트 발생 후 몇 초 후에 재생할지 지정
transition-duration	몇 초 동안 재생할지 지정
transition-property	어떤 속성을 변형할지 지정
transition-timing-function	수치 변형 함수를 지정

## 9.2 변형 속성 / transition-delay 속성

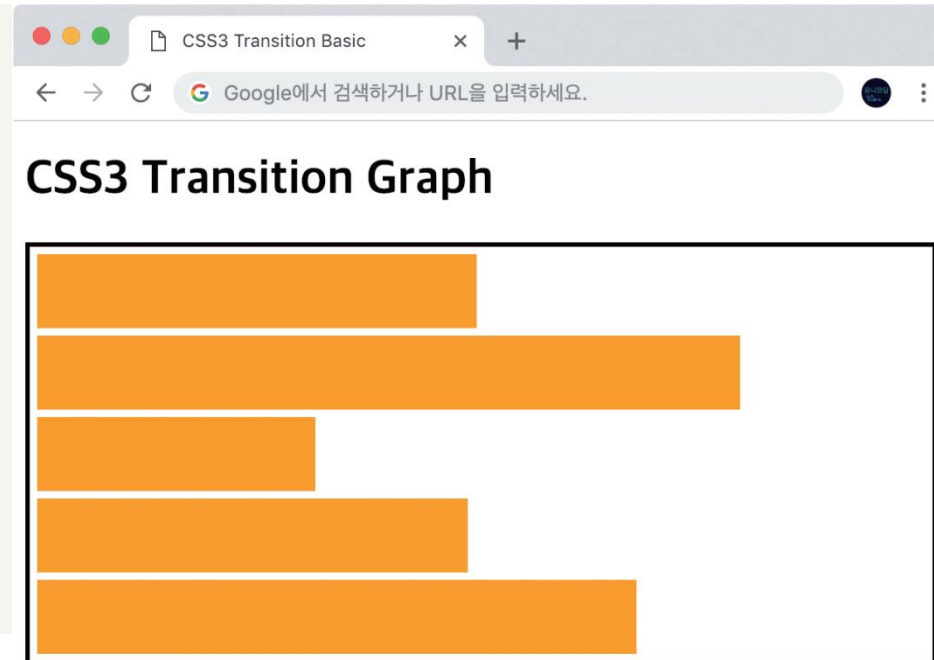
```
.bar:nth-child(1) {  
  transition-delay: 0s;  
}  
.bar:nth-child(2) {  
  transition-delay: 1s;  
}  
.bar:nth-child(3) {  
  transition-delay: 2s;  
}  
.bar:nth-child(4) {  
  transition-delay: 3s;  
}  
.bar:nth-child(5) {  
  transition-delay: 4s;  
}
```



- transition-delay 속성: 이벤트가 발생하고 몇 초 동안 기다린 후 애니메이션이 작동할지 지정하는 속성
- 애니메이션이 느린 속도에서 빠른 속도로 진행함
- 수치 변형 함수를 바꾸고 싶으면 transition-timing-function 속성을 사용

## 9.2 변형 속성 / transition-timing-function 속성

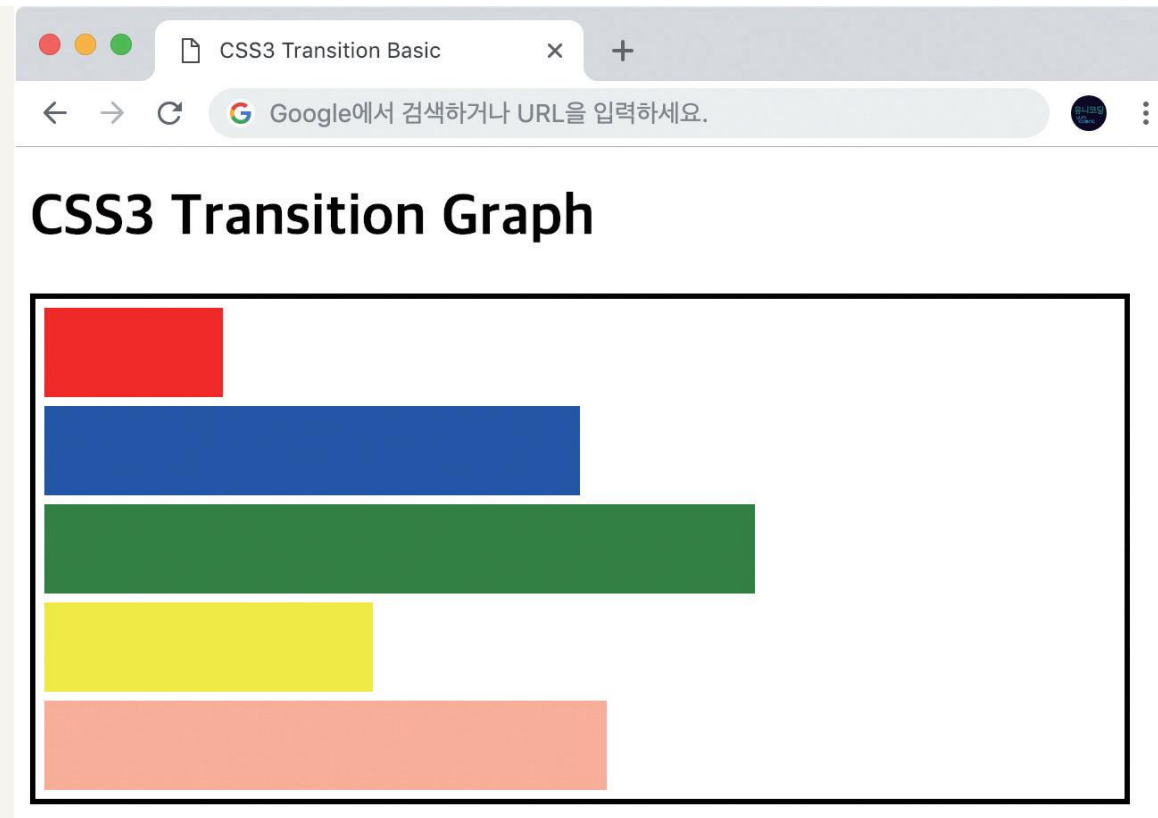
```
.bar:nth-child(1) {  
  transition-timing-function: linear;  
}  
.bar:nth-child(2) {  
  transition-timing-function: ease;  
}  
.bar:nth-child(3) {  
  transition-timing-function: ease-in;  
}  
.bar:nth-child(4) {  
  transition-timing-function: ease-in-out;  
}  
.bar:nth-child(5) {  
  transition-timing-function: ease-out;  
}
```



- 좌측 코드 입력 후 출력: 각각 다른 변형 함수를 가진 바(Bar) 생성

## 9.2 변형 속성 / transition-property 속성

```
#graph {  
  width: 610px;  
  border: 3px solid black;  
}  
  
.bar {  
  width: 10px; height: 50px;  
  background-color: orange;  
  margin: 5px;  
  
  transition-duration: 5s;  
}  
  
#graph:hover > .bar:nth-child(1) {  
  background-color: red;  
  width: 100px;  
}  
#graph:hover > .bar:nth-child(2) {  
  background-color: blue;  
  width: 300px;  
}  
#graph:hover > .bar:nth-child(3) {  
  background-color: green;  
  width: 400px;  
}  
#graph:hover > .bar:nth-child(4) {  
  background-color: yellow;  
  width: 200px;  
}  
#graph:hover > .bar:nth-child(5) {  
  background-color: pink;  
  width: 400px;  
}
```



- 각각의 속성에 다른 형태의 애니메이션을 적용하고 싶을 때는 transition-property 속성 사용
- CSS3 변형 속성은 기본 설정으로 변환 가능한 모든 스타일 속성에 애니메이션 적용이 가능

## 9.3 키 프레임과 애니메이션 속성

속성 이름	설명
animation	모든 animation 속성을 한 번에 적용
animation-delay	이벤트 발생 후 몇 초 후에 재생할지 지정
animation-direction	애니메이션 진행 방향을 설정
animation-duration	애니메이션을 몇 초 동안 재생할지 지정
animation-iteration-count	애니메이션 반복 횟수를 지정
animation-name	애니메이션 이름을 지정
animation-play-state	애니메이션 재생 상태를 지정
animation-timing-function	수치 변형 함수를 지정

## 9.3 키 프레임과 애니메이션 속성

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS3 Transition Basic</title>
  <style>

  </style>
</head>
<body>
  <div id="box">
    <h1>Rotation</h1>
  </div>
</body>
</html>
```

```
* { margin: 0; padding: 0; }
body { position: relative; }
#box {
  position: absolute;
  width: 200px; height: 200px;
  border-radius: 100px;
  text-align: center;
  background: linear-gradient(#cb60b3 0%, #db36a4 100%);
}

#box > h1 {
  line-height: 200px;
}

@keyframes rint {
  from {

  }

  to {

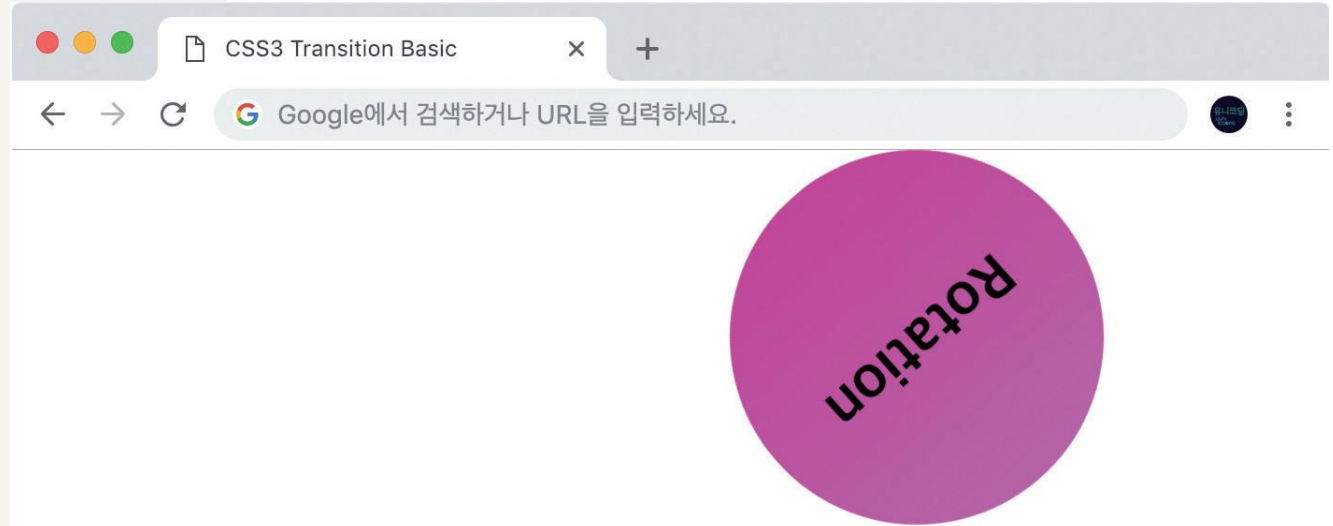
  }
}
```

- @keyframes: 키 프레임 규칙(keyframes @-rule)이라고 부르며, CSS3에서 애니메이션을 지정하는 형식
- 키 프레임은 @keyframes 이름 형태로 입력
- 키 프레임 안에는 퍼센트 단위로 애니메이션을 적용



## 9.3 키 프레임과 애니메이션 속성 / animation-name 속성

```
#box {  
  position: absolute;  
  width: 200px; height: 200px;  
  border-radius: 100px;  
  text-align: center;  
  background: linear-gradient(#cb60b3 0%, #db36a4 100%);  
  
  animation-name: rint;  
  animation-duration: 2s;  
  animation-timing-function: linear;  
}  
  
@keyframes rint {  
  from { left:0; transform: rotate(0deg); }  
  50% { left: 500px; }  
  to { left: 500px; transform: rotate(360deg); }  
}
```



- 키 프레임 생성 이후에는 animation-name 속성을 사용해 태그를 키 프레임에 연결
- 애니메이션 속성은 반드시 animation-name 속성과 animation-duration 속성을 사용해야 함
- animation-timing-function 속성은 transition-timing-function 속성과 동일함

## 9.3 키 프레임과 애니메이션 속성 / animation-iteration-count 속성

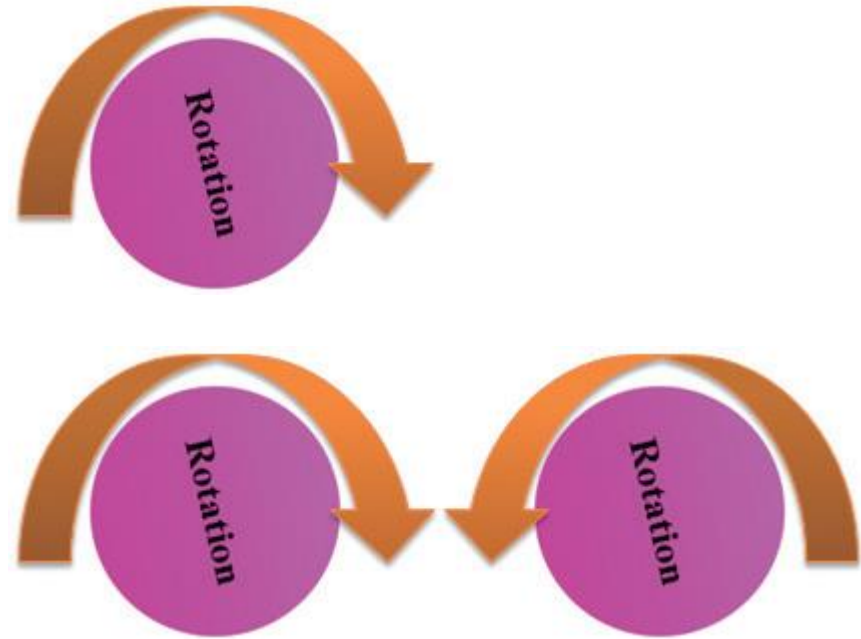
```
#box {  
  position: absolute;  
  width: 200px; height: 200px;  
  border-radius: 100px;  
  text-align: center;  
  background: linear-gradient(#cb60b3 0%, #db36a4 100%);  
  
  animation-name: rint;  
  animation-duration: 2s;  
  animation-iteration-count: infinite;  
  animation-timing-function: linear;  
  
}  
  
@keyframesrint {  
  from { left: 0; transform: rotate(0deg); }  
  50% { left: 500px; }  
  to { left: 500px; transform: rotate(360deg); }  
}
```

- 애니메이션을 특정 횟수만큼 반복하고 싶을 때는 animation-iteration-count 속성 사용
- animation-iteration-count 속성에는 숫자를 적용하며 적용한 숫자만큼 애니메이션 반복
- 애니메이션을 무한번 반복하고 싶을 때는 infinite 키워드를 적용

## 9.3 키 프레임과 애니메이션 속성 / animation-direction 속성

animation-direction 속성	설명
alternate	from에서 to로 이동 후 to에서 from으로 이동을 반복
normal	계속 from에서 to로 이동

```
#box {  
  position: absolute;  
  width: 200px; height: 200px;  
  border-radius: 100px;  
  text-align: center;  
  background: linear-gradient(#cb60b3 0%, #db36a4 100%);  
  
  animation-name: rint;  
  animation-duration: 2s;  
  animation-iteration-count: infinite;  
  animation-direction: alternate;  
  animation-timing-function: linear;  
}  
  
@keyframesrint {  
  from { left: 0; transform: rotate(0deg); }  
  
  50% { left: 500px; }  
  to { left: 500px; transform: rotate(360deg); }  
}
```



## 9.3 키 프레임과 애니메이션 속성 / animation-play-state 속성

```
#box {  
  position: absolute;  
  width: 200px; height: 200px;  
  border-radius: 100px;  
  text-align: center;  
  background: linear-gradient(#cb60b3 0%, #db36a4 100%);  
  
  /* name duration function delay count direct 순서로 입력합니다. */  
  animation: rint 2s linear none infinite alternate;  
}
```

- animation-play-state 속성: 애니메이션을 중지하고 재생할 때 사용함
- animation 속성 사용 방법 사례: 상단 코드
- 사용하고 싶지 않은 속성은 none 키워드를 적용