

트랜지션이란

- 시간이 흐르면서 한 스타일에서 다른 스타일로 바뀌는 것
- 스타일이 바뀌는 시간을 조절해서 애니메이션 효과
- 플래시나 자바스크립트를 사용하지 않고 CSS 소스만으로 애니메이션 효과를 낼 수 있다.
- 트랜지션 속성은 브라우저 접두사를 붙여 사용해야 한다.

transition

속성	설명	사용법
transition-duration	변형이 적용되는 시간	transition-duration : 2s; transition-duration : 1.5s;
transition-delay	변형 적용 전 대기시간	transition-delay : 1s; transition-delay : 0.5s;
transition-timing-function	변형 적용 형태 (점점 빠르게 / 점점 느리게)	transition-timing-function : ease-in; transition-timing-function : ease-out;
transition-property	변형이 적용되는 속성	transition-property : width, height; transition-property : background-color;

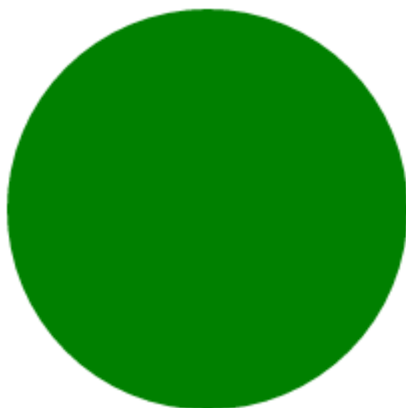
transition



마우스 오버



마우스 클릭

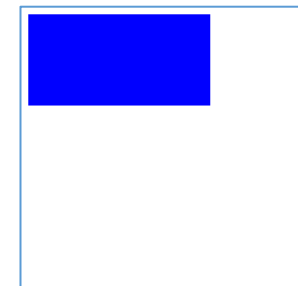
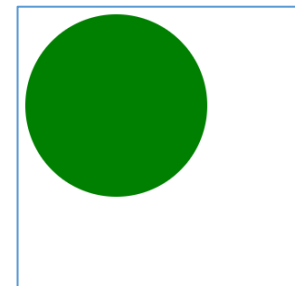
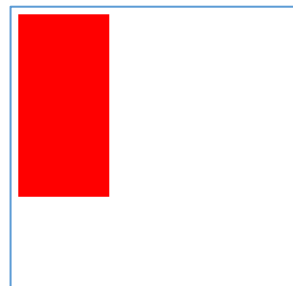


```
<div class="rec"></div>
<style>
  .rec {
    width : 100px; height : 200px;
    background-color : red;
    transition-duration : 1s;
  }
  .rec:hover {
    width : 200px; height : 100px;
    background-color : blue;
  }
  .rec:active {
    width : 200px; height : 200px;
    background-color : green;
    transition-duration : 3s;
    border-radius : 100px;
  }
}
```

transition – 1

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>HTML5 연습</title>
  <style>
    .rec {
      width : 100px; height : 200px;
      background-color : red;
      transition-duration : 1s;
    }
    .rec:hover {
      width : 200px; height : 100px;
      background-color : blue;
      transform:rotate(360deg);
    }
    .rec:active {
      width : 200px; height : 200px;
      background-color : green;
      transition-duration : 3s;
      border-radius : 100px;
    }
  </style>
</head>
```

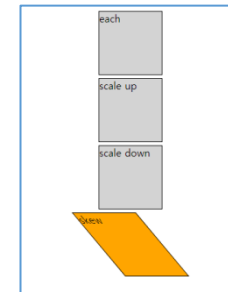
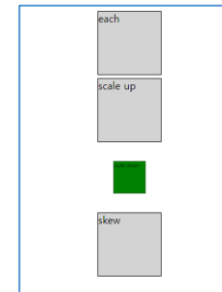
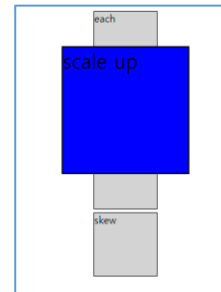
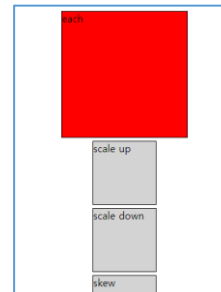
```
<body>
  <div class="rec"> </div>
</body>
</html>
```



transition – 2

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>HTML5 연습 </title>
  <style>
    div {
      margin : 5px auto;
      width: 100px; height: 100px;
      background-color: lightgray;
      transition-duration: 3s;
      border: 1px solid black;
    }
    .each:hover {
      background-color: red;
      transition-property: width, height, background-color;
      width: 200px; height: 200px;
      transition-duration: 1s, 2s, 0.5s;
    }
    .scaleup:hover {
      background-color : blue;
      transform: scale(2);
    }
```

```
.scaledown:hover {
  background-color: green;
  transform: scale(0.5);
}
.skew:hover {
  background-color: orange;
  transform: skew(40deg);
}
</style>
</head>
<body>
  <div class="each">each</div>
  <div class="scaleup">scale up</div>
  <div class="scaledown">scale down</div>
  <div class="skew">skew</div>
</body>
</html>
```



transition 속성과 animation 속성

공통점

- 시작 스타일과 끝나 는 스타일을 지정하면 전체적으로 부드럽게 변화하는 애니메이션 효과를 만든다.
- 애니메이션에 소요되는 시간이나 지연 시간 등을 지정한다.

다른점

- 애니메이션의 시작에서부터 끝날 때까지 어느 지점이든 @keyframes 속성을 사용해 애니메이션을 정의할 수 있다



@keyframes

- 사용자가 애니메이션을 정의한다.
- 애니메이션이 시작할 상태와 애니메이션이 끝날 때의 상태, 최소한 두 가지 상태를 설정해야 한다.
- 시작할 때의 상태 는 0%(또는 from) 스타일을 정의하고 끝나는 상태는 100% (또는 to) 스타일을 정의한다.
- 중간에 원하는 부분에 애니메이션을 추가할 수 있다.

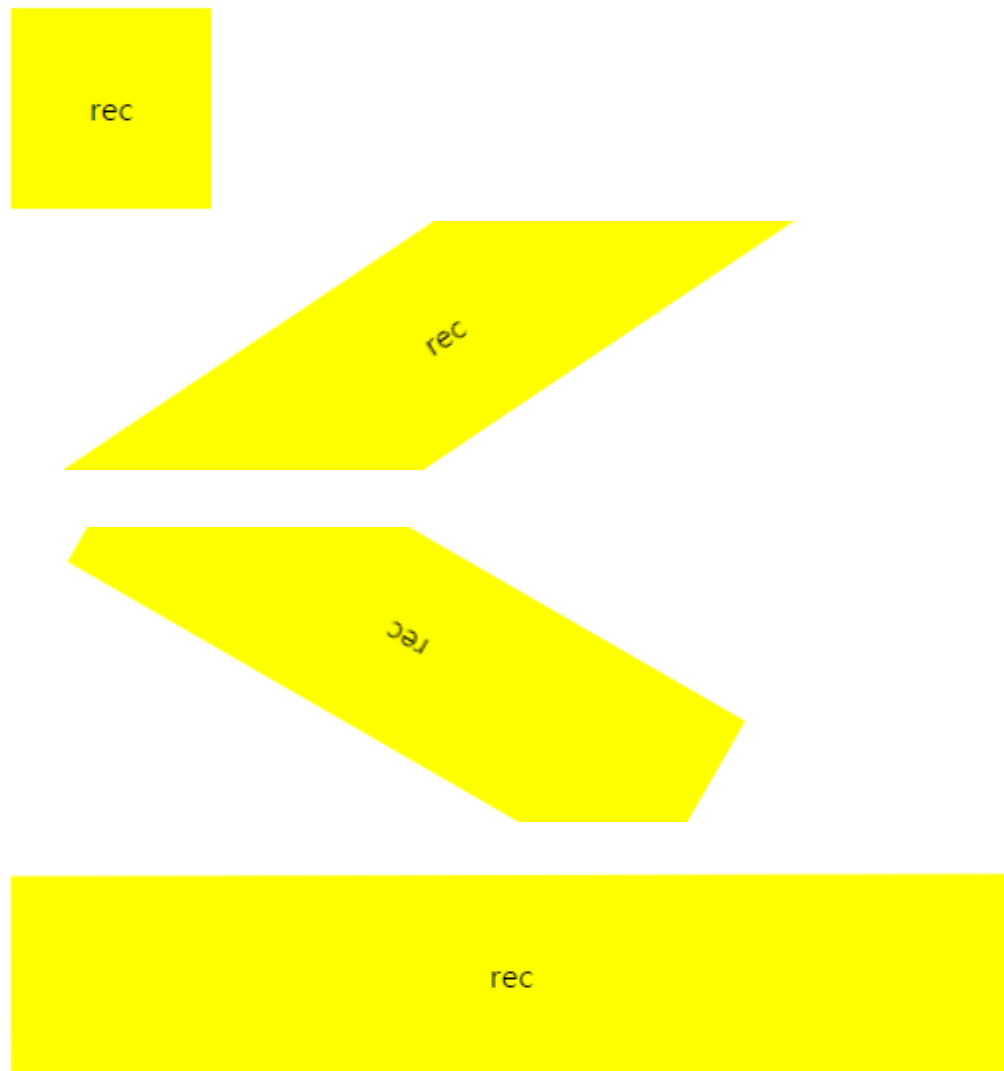
```
@keyframes myani{
  0% {
    background-color:red;
  }
  100%{
    background-color:blue;
  }
}
```

animation

속성	설명	사용법
@keyframes	애니메이션 동작방식 정의	<pre>@keyframes ani1 { from { width : 100px; } to { width : 200px; } }</pre>
animation-name	애니메이션 동작방식이 정의된 애니메이션명 지정	animation-name : ani1;
animation-iteration-count	애니메이션 반복 횟수	animation-iteration-count : 2; animation-iteration-count : infinite;
animation-direction	애니메이션 종료 후 재시작 시 정방향 / 역방향 지정	animation-direction : normal (정) animation-direction : reverse (역) animation-direction : alternate (정-역) animation-direction : alternate-reverse (역-정)

duration / delay / timing-function 등의 속성은 transition 과 동일한 방식으로 사용 가능

animation



```
<div id="wrap">
  <div id="rec">rec</div>
</div>
<style>
  #wrap #rec {
    background-color : #ffff00;
    width:100px ; height : 100px;
    line-height : 100px;
    text-align : center;
    animation-duration : 2s;
    animation-name : ani1;
  }

  @keyframes ani1 {
    from {
      width:100px;
      transform:rotate(0deg)
    }
    to {
      width:500px;
      transform:rotate(360deg)
    }
  }
</style>
```

animation

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>HTML5 연습</title>
  <style>
    #wrap #rec {
      background-color : #ffff00;
      width : 100px ; height : 100px;
      line-height : 100px; text-align : center;
      animation-duration : 2s;
      animation-name : ani1;
      animation-iteration-count : 10;
      animation-direction : alternate;
      animation-name : ani1;
    }
    @keyframes ani1 {
      from {
        width:100px; transform:rotate(0deg)
      }
      to {
        width:500px; transform:rotate(360deg)
      }
    }
  </style>
</head>
```

```
<body>
  <div id="wrap">
    <div id="rec">rec</div>
  </div>
</body>
</html>
```





@import

외부 css 파일을 불러올때 사용

- <link> 태그 대신 사용 가능
- <style> 태그 내부에 위치
- <style> 태그 내용 중 상단에 위치

<link rel='stylesheet' href='all.css'>

➔ @import url('all.css')

<link rel='stylesheet' href='screen.css' media='screen'>

➔ @import url('screen.css') screen

<link rel='stylesheet' href='print.css' media='print'>

➔ @import url('print.css') print



@import

link / @import 차이점

- link
 - 속도 빠름
 - 외부 css 파일이 여러 개인 경우 개수 만큼 link 선언
한 개로 사용하려면 한 개 파일 안에 모든 내용 작성
- @import
 - 속도 느림, 로딩되는 동안 스타일이 적용되지 않은
형태가 보일 수 있음
 - 여러 개의 외부 css 파일 선언 가능

@import

link / @import 차이점

```
<head>
  <meta charset="utf-8">
  <link rel='stylesheet' href='screen.css'>
  <link rel='stylesheet' href='print.css'>
</head>
```

```
<head>
  <meta charset="utf-8">
  <style>
    @import url('main.css');
  </style>
</head>
```

```
@import url('screen.css');
@import url('print.css');
```



@media

하나의 HTML 문서를 다양한 장치에서 서로 다르게 보여주는 기능

```
<link rel='stylesheet' href='screen.css' media='screen'> / @import url('screen.css') screen
```

```
<link rel='stylesheet' href='print.css' media='print'> / @import url('print.css') print
```



```
@media screen { ... }
```

```
@media print { ... }
```



@media

미디어 종류

속성	설명	속성	설명
all	모든 미디어 타입	<u>print</u>	인쇄기
aural	음성 합성 장치	projection	프로젝터
braille	점자 표시 장치	<u>screen</u>	컴퓨터 스크린
handheld	PDA, PMP 등	<u>tv</u>	TV
embossed	점자 표시 장치		



@media

1	2	3 700px x 279px
4	5	6

1 500px x 386px
2
3
4
5
6

@media

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>HTML5 연습</title>
  <style>
    .row {
      width : 100%;
      text-align : center;
      margin-bottom : 5px;
    }
    .item {
      display : inline-block;
      width : 30%; height : 50px;
      border : 1px solid black;
      border-radius : 2px;
      box-sizing : border-box;
    }
    @media screen and (max-width : 500px) {
      .item {
        width : 100%;
        margin-bottom : 5px;
      }
    }
  </style>
</head>
```

```
<body>
  <div class="row">
    <div class="item">1</div>
    <div class="item">2</div>
    <div class="item">3</div>
  </div>
  <div class="row">
    <div class="item">4</div>
    <div class="item">5</div>
    <div class="item">6</div>
  </div>
</body>
</html>
```