**CSS3 Transitions**

Transition (CSS 속성에 에니메이션 효과를 주다) 개체에 적용된 스타일을 대상으로 주어진 시간과 조건으로 부드럽게 상태를 변화시키는 기술이다  
Transition 은 CSS3부터 새로 생겼습니다 따라서 웹 브라우저마다 지원사항이 다르고, 표현에 대한 부분에서도 몇 가지 문제점을 가지고 있습니다. 따라서 이 기능을 웹 사이트 작동에 지대한 영향을 끼치는 핵심 액션에 사용하는 것은 안전하지 않습니다.  
참조 : http://poiemaweb.com/css3-effect.html

## transition-property

변화가 일어날 속성을 명시합니다. 쉽표(,)로 구분하여 여러개의 속성을 동시에 지정할 수 있습니다.

## transition-duration

변화가 일어나는 시간 간격을 명시합니다. 기본값은 0이기 때문에 이 값을 표시를 해줘야 부드러운 변화 모습을 볼 수 있습니다. 음수값을 지정하면 0으로 인식됩니다

## transition-delay

변화가 시작되기 전까지의 시간을 정할 수 있습니다. 기본값은 0 입니다.

## Transition-timing-function

transition은 위 4가지 속성의 축약 속성(shorthand property)이다. 영어로 shorthand property 라고 지칭하니, 함축속성, 또는 대표속성이라고도 표현할 수 있을 것이다. 쉽게 말하면, transition 하나로 위 4가지를 다 표현할 수 있다는 얘기다.

예를 들어, 아래와 같은 코드가 있다면

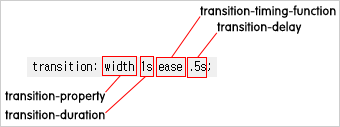
transition-property:width;

transition-duration: 1s;

transition-timing-function: ease;

transition-delay: .5s;

transition: width 1s ease .5s;



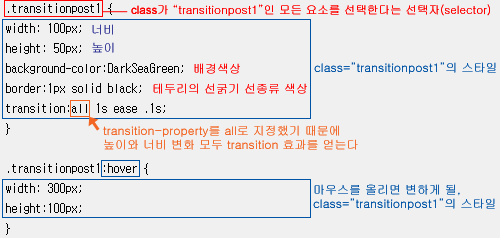
**transition-property**  
transition 구문의 첫번째 자리에 오는 transition-property 는 어떤 속성에 transition 효과를 줄지를 설정한다.  
  
transition-property의 값(value)

* none : transition 효과를 지정하지 않는다.
* all : 모든 속성이 transition 효과를 얻는다.
* property : transition 효과를 얻게 될 속성을 지정한다.

예제1

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title></title>  <style type="text/css">  div { width:100px; height: 100px; background-color: lightgreen;}  .transitionpost1 {  width: 100px;  height: 50px;  background-color:DarkSeaGreen;  border:1px solid black;  transition:all 1s ease .1s;  }    .transitionpost1:hover {  width: 300px;  height:100px;  } </style>  </head>  <body>  <div class="transitionpost1"></div>  </body>  </html> |

**이것의 설명은**



예제2)

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title></title>  <style type="text/css">  div { width:100px; height: 100px; background-color: lightgreen;}  .transitionpost2 {  width: 100px;  height: 50px;  background-color:DarkSeaGreen;  border:1px solid black;  **transition:width 1s ease .1s;**  }    .transitionpost2:hover {  width: 300px;  height:100px;  }  </style>  </head>  <body>  <div class="transitionpost2"></div>  </body>  </html>  <예2>는 다른 조건들을 <예1>과 동일하게 하되, transition-property를 all 이 아닌 width로 설정했다. <예2>의 도형은 **마우스 커서를 올렸을 때 똑같이 너비 300px, 높이 100px로 확장하지만**, **높이 변화는 transition의 효과를 얻지 못한다** |

예제3)

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title></title>  <style type="text/css">  div { width:100px; height: 100px; background-color: lightgreen;}  .transitionpost3 {  width: 100px;  height: 50px;  background-color:#8FBC8F;  border:1px solid black;  **transition:width 1s ease, background-color 13s ease;**  }    .transitionpost3:hover {  width: 300px;  height: 100px;  background-color:#000000;  }  </style>  </head>  <body>  <div class="transitionpost3"></div>  </body>  </html> |

예3 - 너비 변화는 1초, 배경색상 변화는 13초에 걸쳐 일어난다. transition 속성을 지정하지 않은 height(높이)는 변화는 하되, 점진적으로 변화하는 transition 효과는 얻지 못한다

**transition-duration 예제들**

**예제4**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title></title>  <style type="text/css">  div { width:100px; height: 100px; background-color: lightgreen;}  .transitionpost4 {  width: 100px;  height: 50px;  background-color:#8FBC8F;  border:1px solid black;  transition:width 11s ease, height 432ms ease, background-color .3s ease;  }    .transitionpost4:hover {  width: 300px;  height:100px;  background-color:#2580a2;  }  </style>  </head>  <body>  <div class="transitionpost4"></div>  </body>  </html> |

예4 - 너비 변화는 11초, 높이 변화는 432ms(밀리세컨드), 배경색상 변화는 0.3초에 걸쳐 일어난다.

**transition-timing-fuction :** transition 구문의 세번째 자리에 오는 transition-timing-functing은 변화가 일어나는 속도를 설정한다. 속도 커브의 함수를 설정한다는 것이 더 정확한 표현이겠지만, 어려운 말 보다는 실제 움직임을 보면서 이해하는 것이 편하다.

transition-timing-function의 값(value)으로는

**linear** : transition-timing-function을 linear로 설정하면 변화가 시작부터 종료까지 동일한 속도로 일어난다

**ease** : ease는 극초반은 느리게, 초반은 빠르게, 종료지점은 느리게 변화가 진행된다.

**ease-in**: ease-in은 시작지점의 변화가 천천히 진행된다.

**ease-out :** ease-out은 종료지점의 변화가 천천히 진행된다.

**ease-in-out** : ease-in-out은 시작지점과 종료지점의 변화가 천천히 진행된다.

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <style>  div {  width: 100px;  height: 100px;  background: red;  -webkit-transition: width 2s; /\* Safari \*/  transition: width 2s;  }  /\* For Safari 3.1 to 6.0 \*/  #div1 {-webkit-transition-timing-function: linear;}  #div2 {-webkit-transition-timing-function: ease;}  #div3 {-webkit-transition-timing-function: ease-in;}  #div4 {-webkit-transition-timing-function: ease-out;}  #div5 {-webkit-transition-timing-function: ease-in-out;}  /\* Standard syntax \*/  #div1 {transition-timing-function: linear;}  #div2 {transition-timing-function: ease;}  #div3 {transition-timing-function: ease-in;}  #div4 {transition-timing-function: ease-out;}  #div5 {transition-timing-function: ease-in-out;}  div:hover {  width: 300px;  }  </style>  </head>  <body>  <p><b>Note:</b> This example does not work in Internet Explorer 9 and earlier versions.</p>  <div id="div1">linear</div><br>  <div id="div2">ease</div><br>  <div id="div3">ease-in</div><br>  <div id="div4">ease-out</div><br>  <div id="div5">ease-in-out</div><br>  <p>Hover over the div elements above, to see the different speed curves.</p>  </body>  </html> |

**Transition-timing-function 소개사이트**

**https://matthewlein.com/ceaser/**

<http://mylko72.maru.net/jquerylab/study/css3-3d-perspective.html>

**transition-delay 예제들**

transition 구문의 마지막 네번째 자리에 오는 transition-delay 는 transition 효과가 언제 시작할지를 설정한다. 말 그대로, 딜레이(delay)를 설정한다.

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title></title>  <style type="text/css">  div { width:100px; height: 100px; background-color: lightgreen;}  .transitionpost8 {  width: 100px;  height: 50px;  background-color:DarkSeaGreen;  border:1px solid black;  transition:width 2s ease-in 1s;  }    .transitionpost8:hover {  width: 300px;  }  </style>  </head>  <body>  <div class="transitionpost8"></div>  </body>  </html> |

마우스 커서를 올리면 1초가 지난 후 2초에 걸쳐 너비가 확장된다.

**transition의 default 값**

transition의 default 값은 all 0 ease 0

transition 구문에서 생략된 값은 default 값으로 대체된다. transition 효과가 나타나기 위해서 반드시 주어져야 하는 값, 즉 생략해선 안되는 값은 transition-duration 이다. 0초에 걸친 점진적인 변화라는 것은 있을 수 없기 때문이다

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title></title>  <style type="text/css">  div { width:100px; height: 100px; background-color: lightgreen;}  .transitionpost9 {  width: 100px;  height: 50px;  background-color:Green;  border:1px solid black;  transition:2s;  }    .transitionpost9:hover {  width: 300px;  }  </style>  </head>  <body>  <div class="transitionpost9"></div>  </body>  </html> |

 transition의 값으로 '2s'만 입력했다. 생략된 값은 default 값으로 대체되므로, transition:2s;는 transition:all 2s ease 0; 같다.

**transition 효과를 얻을 수 있는 속성들(properties)**

꼭 그렇다고는 할 수 없지만, 보통 숫자값을 갖는 속성(property)들은 transition을 통해 점진적인 효과를 낼 수 있다 예 : opacity, border-radius, margin, color, font-size, padding 등등

transition 효과가 가능한 속성(property)의 목록은  
  
<http://www.w3.org/TR/css3-transitions/>

**그밖의 transition 효과들**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title></title>  <style type="text/css">  div { width:100px; height: 100px; background-color: lightgreen;}  .transitionpost10 {  opacity:1;  width: 100px;  height: 50px;  background-color:DarkSeaGreen;  border:1px solid black;  transition:**opacity** .5s ease-in;  }    .transitionpost10:hover {  opacity:0;  }  </style>  </head>  <body>  <div class="transitionpost10"></div>  </body>  </html> |

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title></title>  <style type="text/css">  div { width:100px; height: 100px; background-color: lightgreen;}  .transitionpost11 {  border-radius:0;  width: 100px;  height: 50px;  background-color:DarkSeaGreen;  border:1px solid black;  transition:**border-radius** .5s ease-in-out;  }    .transitionpost11:hover {  border-radius:50%;  }  </style>  </head>  <body>  <div class="transitionpost11"></div>  </body>  </html> |

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title></title>  <style type="text/css">  div { width:100px; height: 100px; background-color: lightgreen;}  .transitionpost12 {  margin-left:0;  width: 100px;  height: 50px;  background-color:DarkSeaGreen;  border:1px solid black;  transition:margin-left 1.5s ease-in-out;  }    .transitionpost12:hover {  margin-left:200px;  }  </style>  </head>  <body>  <div class="transitionpost12"></div>  </body>  </html> |

**복합적으로 사용**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title></title>  <style type="text/css">  div {text-align: center;}  .ex-02 {  display: inline-block;  margin: 0;  width: 80px;  height: 80px;  line-height:80px;  text-decoration: none;  font-size: 12px;  color: #999;  background-color: #f4f4f4;  border-width: 1px;  border-style: solid;  border-radius: 10%;  border-color:#999;  transition: width .2s ease,  height **.2s** ease, /\* .2는 0.2초입니다. \*/  line-height .2s ease,  color .2s ease,  font-size .2s ease,  border .2s ease,  background-color .2s ease,  border-radius .2s ease .4s;  }    .ex-02:hover {  width: 120px;  height: 120px;  line-height:120px;  color: #ffcccc;  font-size: 20px;  border-width: 5px;  border-color:#ff5e5d;  background-color: #fa8382;  border-radius: 50%;  }  </style>  </head>  <body>  <div class="ex-02">Transition</div>  </body>  </html> |

위와 비슷한 예제

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title></title>  <style type="text/css">  body {  color: #fff;  font: 600 14px/24px "Open Sans", "HelveticaNeue-Light", "Helvetica Neue Light", "Helvetica Neue", Helvetica, Arial, "Lucida Grande", Sans-Serif;  }  .box {  background: #2db34a;  border-radius: 6px;  cursor: pointer;  height: 95px;  line-height: 95px;  text-align: center;  transition: background .2s linear, border-radius 1s ease-in 1s;  width: 95px;  }  .box:hover {  background: #ff7b29;  border-radius: 50%;  }  </style>  </head>  <body>  <div class="box">Box</div>  </body>  </html> |

**perspective 설명**

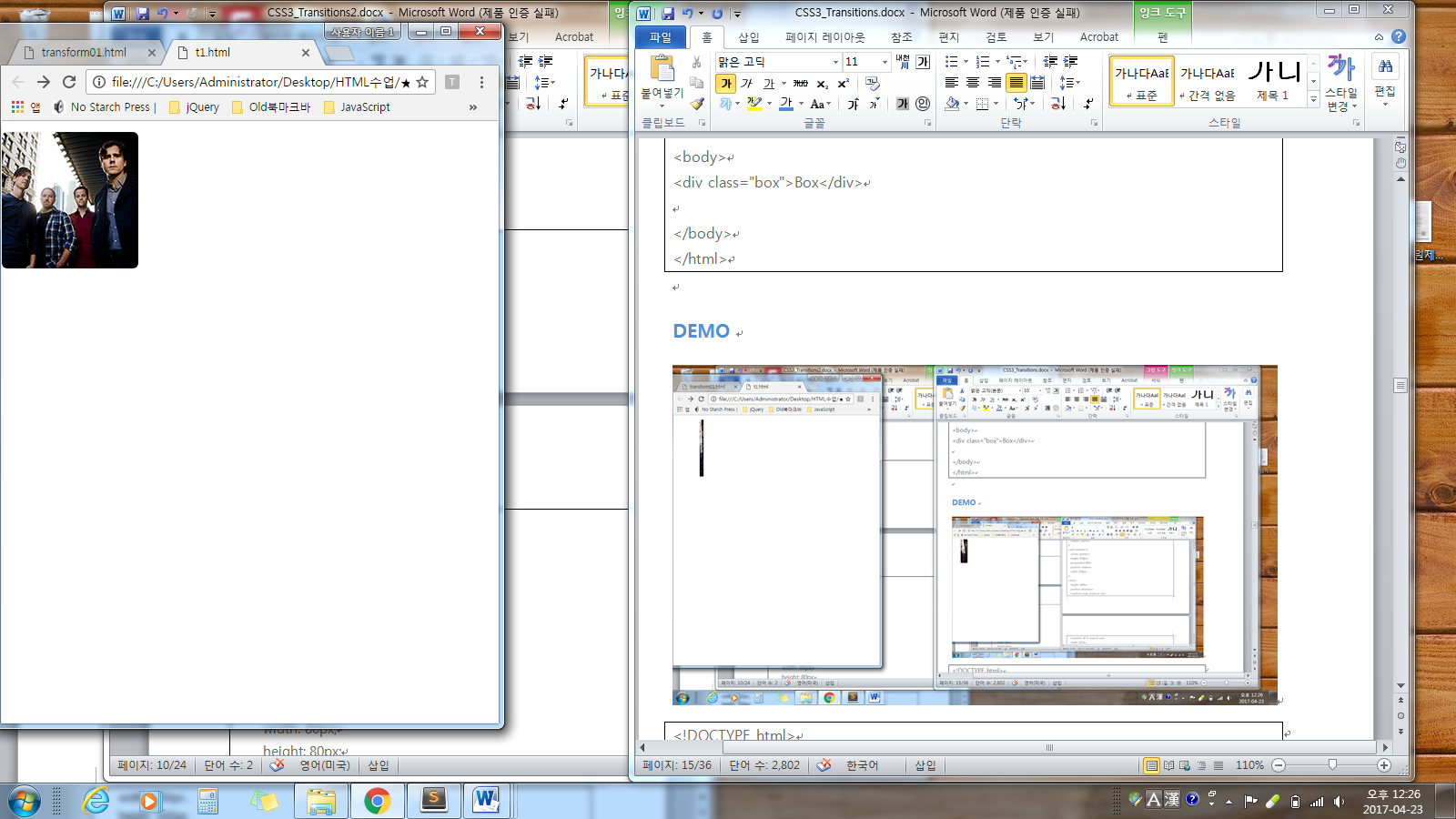
**https://webkit.org/blog-files/3d-transforms/perspective-by-example.html**

[**http://mylko72.maru.net/jquerylab/study/css3-3d-perspective.html**](http://mylko72.maru.net/jquerylab/study/css3-3d-perspective.html)

3D 환경을 만들기 위해서는 입체감을 부여하는데 반드시 필요한 perspective 속성을 활용해야 한다. 이 속성은 픽셀 기반의 길이 값을 수용한다. 이것은 z축을 따라 캔버스로부터 관찰자의 거리를 나타낸다.

perspective 속성은 사실상 전환된 요소의 부모에 적용된다. 즉 해당 부모의 모든 자식들이 동일한 관점으로 동일한 3D 공간에서 동작하게 된다. 그러나 만약 해당 요소에만 3D 전환을 적용하고 싶다면 그렇게 할 수도 있다.

**DEMO**



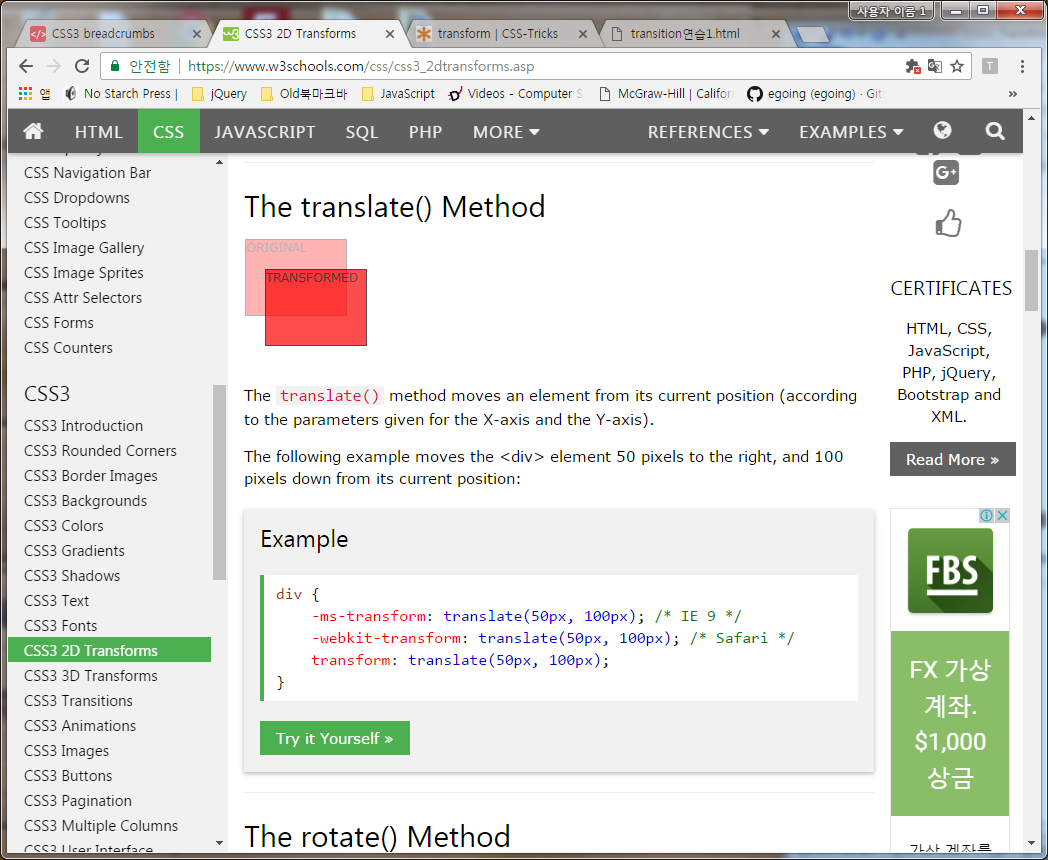
|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title></title>  <style type="text/css">  body {  font: 600 14px/24px "Open Sans", "HelveticaNeue-Light", "Helvetica Neue Light", "Helvetica Neue", Helvetica, Arial, "Lucida Grande", Sans-Serif;  margin: 12px 0;  }  .card-container {  cursor: pointer;  height: 150px;  perspective: 600;  position: relative;  width: 150px;  }  .card {  height: 100%;  position: absolute;  transform-style: preserve-3d;  transition: all 1s ease-in-out;  width: 100%;  }  .card:hover {  transform: rotateY(180deg);  }  .card .side {  backface-visibility: hidden;  border-radius: 6px;  height: 100%;  position: absolute;  overflow: hidden;  width: 100%;  }  .card .back {  background: #eaeaed;  color: #0087cc;  line-height: 150px;  text-align: center;  transform: rotateY(180deg);  }  </style>  </head>  <body>  <div class="card-container">  <div class="card">  <div class="side"><img src="https://s3-us-west-2.amazonaws.com/s.cdpn.io/29841/jimmy.jpg" alt="Jimmy Eat World"></div>  <div class="side back">Jimmy Eat World</div>  </div>  </div>  </body>  </html> |

## **CSS3 2D Transforms**

* translate()
* rotate()
* scale()
* skewX()
* skewY()
* matrix()
* **Transform :** codepen.io/vineethtr/full/XKKEgM / transform 은 블록속성이나 인라인블럭속성에서만 작동

## **The translate() Method**

The translate() method moves an element from its current position (according to the parameters given for the X-axis and the Y-axis).

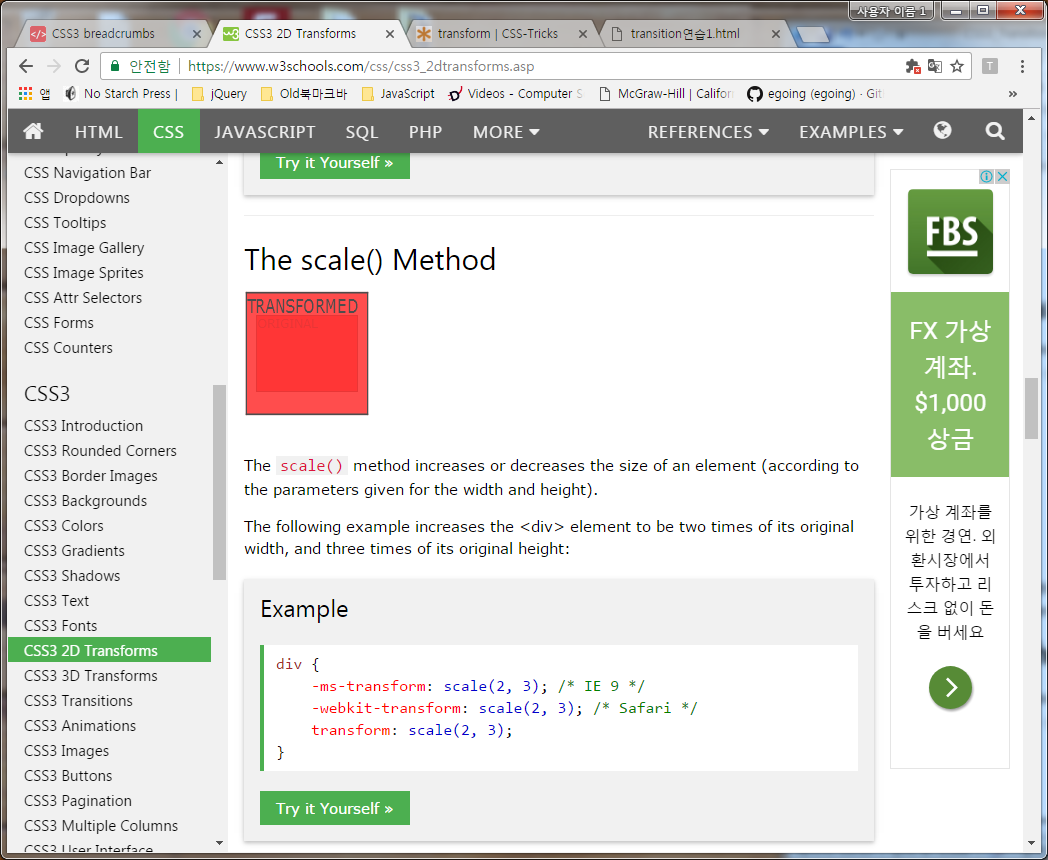


## **The rotate() Method**

## The rotate() method rotates an element clockwise or counter-clockwise according to a given degree.

## **The scale() Method**

The scale() method increases or decreases the size of an element (according to the parameters given for the width and height



## **The skewX() Method**

The skewX() method skews an element along the X-axis by the given angle

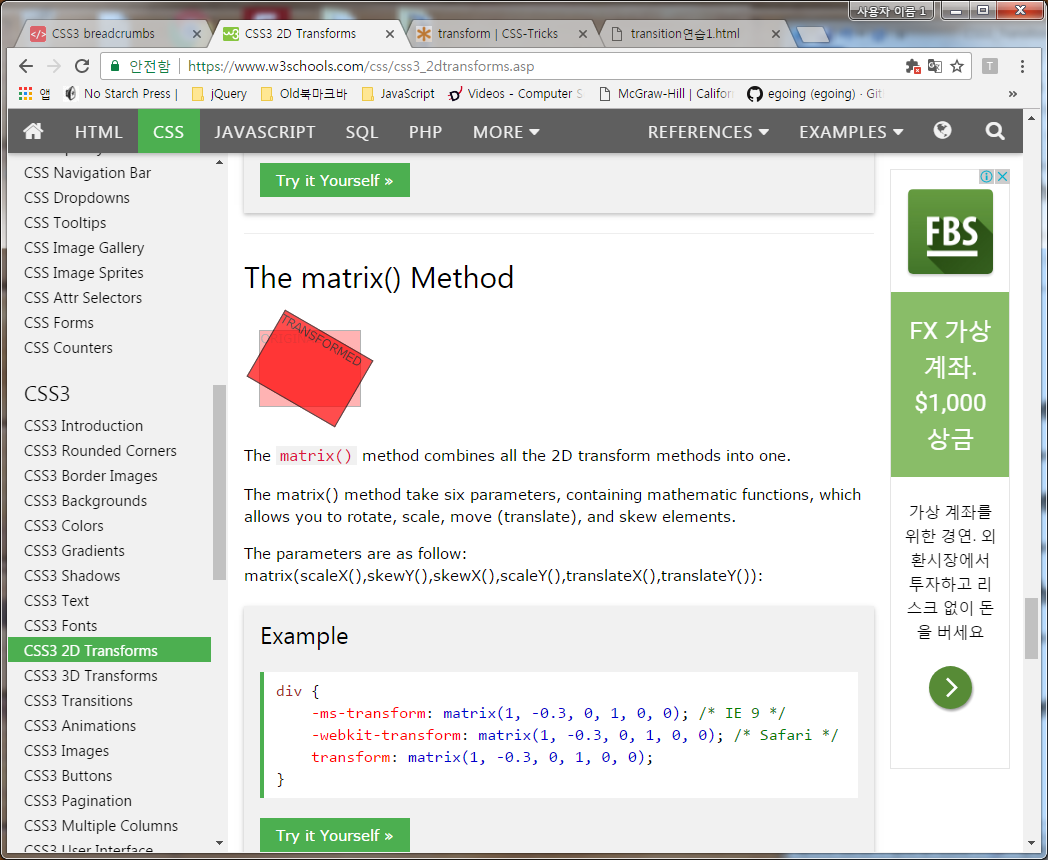
## **The skewY() Method**

The skewY() method skews an element along the Y-axis by the given angle.

## **The skew() Method**

The skew() method skews an element along the X and Y-axis by the given angles.

## **The matrix() Method**



The matrix() method combines all the 2D transform methods into one.

The matrix() method take six parameters, containing mathematic functions, which allows you to rotate, scale, move (translate), and skew elements.

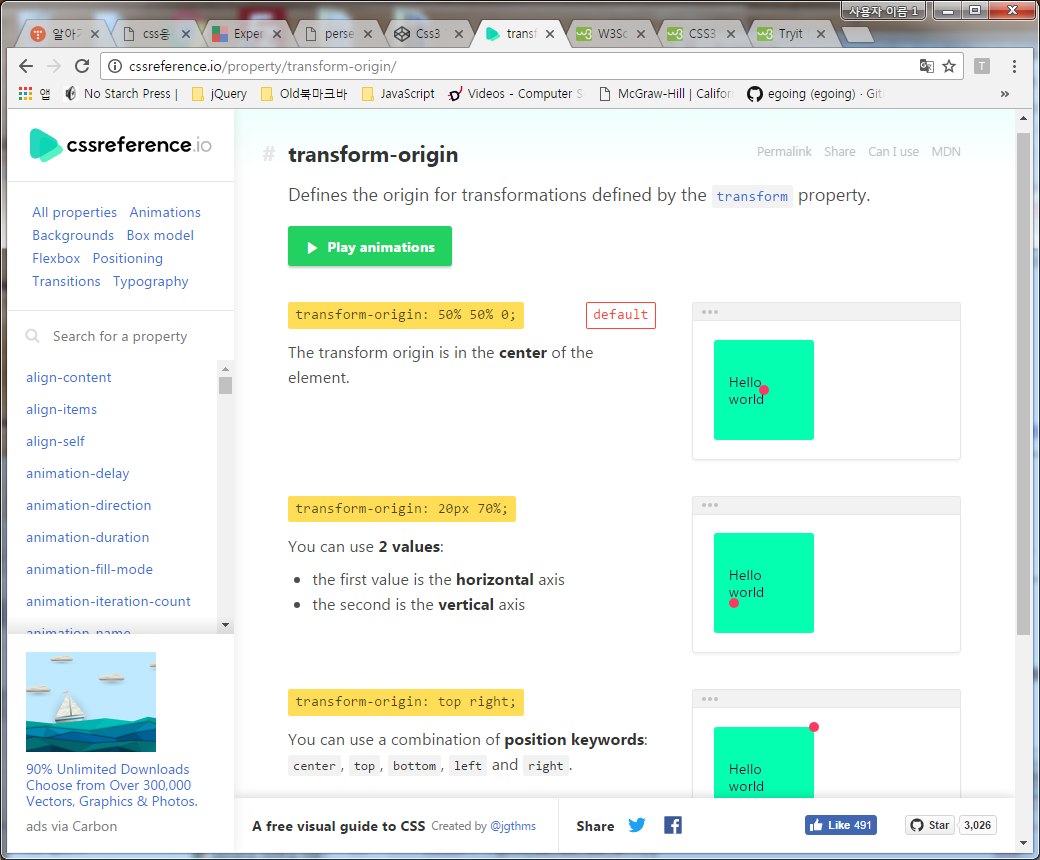
### **3D 이동**

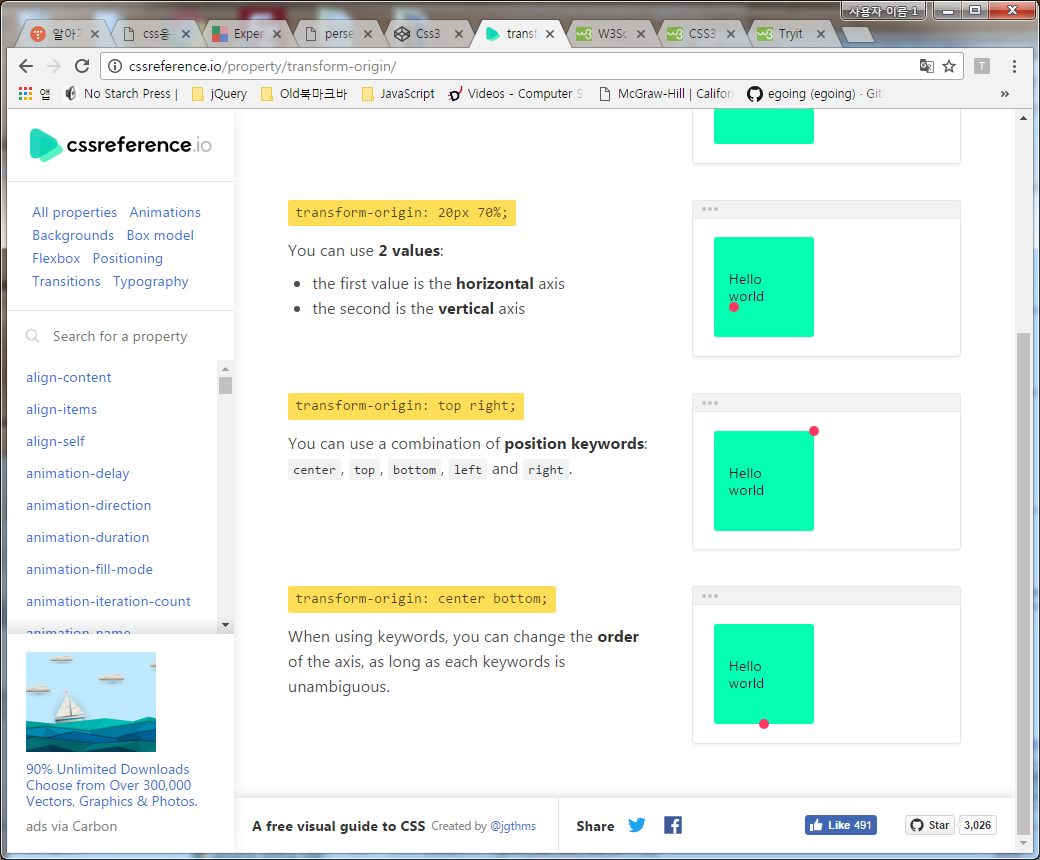
만약 요소를 움직일 때 perspective 값을 명시하지 않고 z축을 따라 100px을 옮긴다면 요소에 어떤 시각적인 변화도 일어나지 않는다. 거리를 가늠하는 관점이 없기 때문에 이런 움직임이 분명하게 드러나지 않는

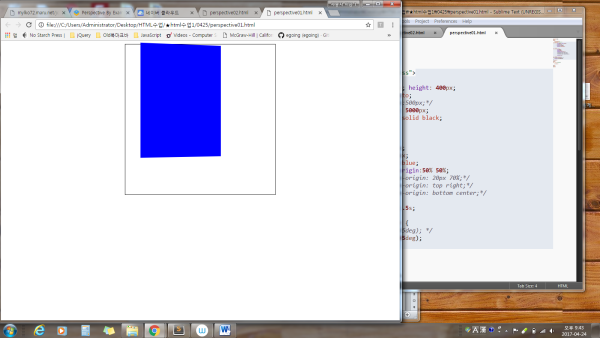
|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title></title>  <style type="text/css">  div.container {  width:300px;  margin:0 auto;  /\* perspective:500px;  perspective 값을 명시하지 않고 z축을 따라 100px을 옮긴다면 요소에 어떤 시각적인 변화도 일어나지 않는다. 거리를 가늠하는 관점이 없기 때문  \*/    }  div.container div {  width:300px;  height:300px;  background:blue;  transition:.5s;  }  div.container div:hover {  transform:translateZ(100px);  }  </style>  </head>  <body>  <div class="container">  <div></div>  </div>  </body>  </html> |

[**transform-origin**](https://developer.mozilla.org/ko/docs/Web/CSS/transform-origin)

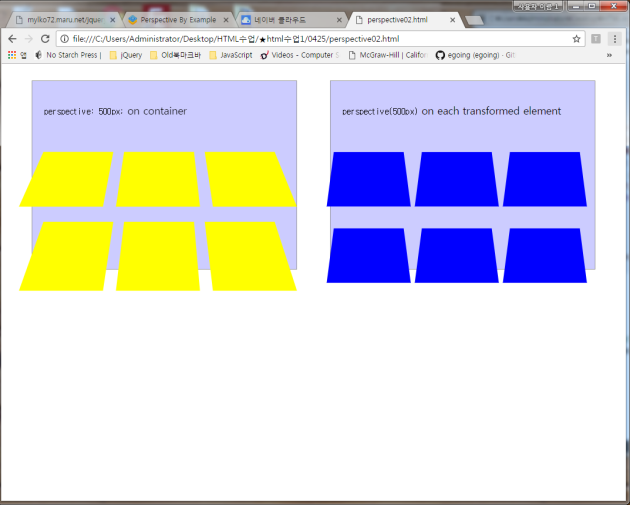
원점의 위치를 지정한다. 기본값은 엘리먼트의 왼쪽(left) 위(top)이지만, 수정이 가능하다. 회전(rotation), 확대/축소(scaling), 비틀기(skewing) 같이 한 점을 기준으로 수행되는 변환에 대해서 이 속성을 지정할 수 있다.







|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title></title>  <style type="text/css">  div.container {  width:400px; height: 400px;  margin:0 auto;  /\* perspective:500px;\*/  perspective: 5000px;  border:1px solid black;  }  div.container div {  width:300px;  height:300px;  background:blue;  transform-origin:50% 50%;  /\*transform-origin: 20px 70%;\*/  /\*transform-origin: top right;\*/  /\*transform-origin: bottom center;\*/    transition:.5s;  }  div.container div:hover {  /\* transform:rotateX(45deg); \*/  transform:rotateY(45deg);  }  </style>  </head>  <body>  <div class="container">  <div></div>  </div>  </body>  </html> |



|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title></title>  <style type="text/css">  body > div {  width: 960px;  margin: 0 auto;    }  body > div > div {  background: rgba(0, 0, 255, 0.2);  width: 450px;  height: 320px;  float: left;  border: 1px solid #999;  margin-top: 20px;  }  .perspective {  perspective: 500px;  }  .perspective div {  width: 130px;  height: 130px;  background: blue;  transform: rotateX(45deg);  float: left;  margin-right: 20px;  background-color: yellow;  }  .no-perspective {  float: right;  }  .no-perspective div {  width: 130px;  height: 130px;  background: blue;  transform: perspective(500px) rotateX(45deg);  float: left;  margin-right: 20px;  }  p {  padding: 20px;  font-size:1.1em;  }  </style>  </head>  <body>  <div>  <div class="perspective">  <p><code>perspective: 500px;</code> on container</p>  <div></div>  <div></div>  <div></div>  <div></div>  <div></div>  <div></div>  </div>  <div class="no-perspective">  <p><code>perspective(500px)</code> on each transformed element</p>  <div></div>  <div></div>  <div></div>  <div></div>  <div></div>  <div></div>  </div>  </div>  </body>  </html> |

## preserve-3d (관점 유지)

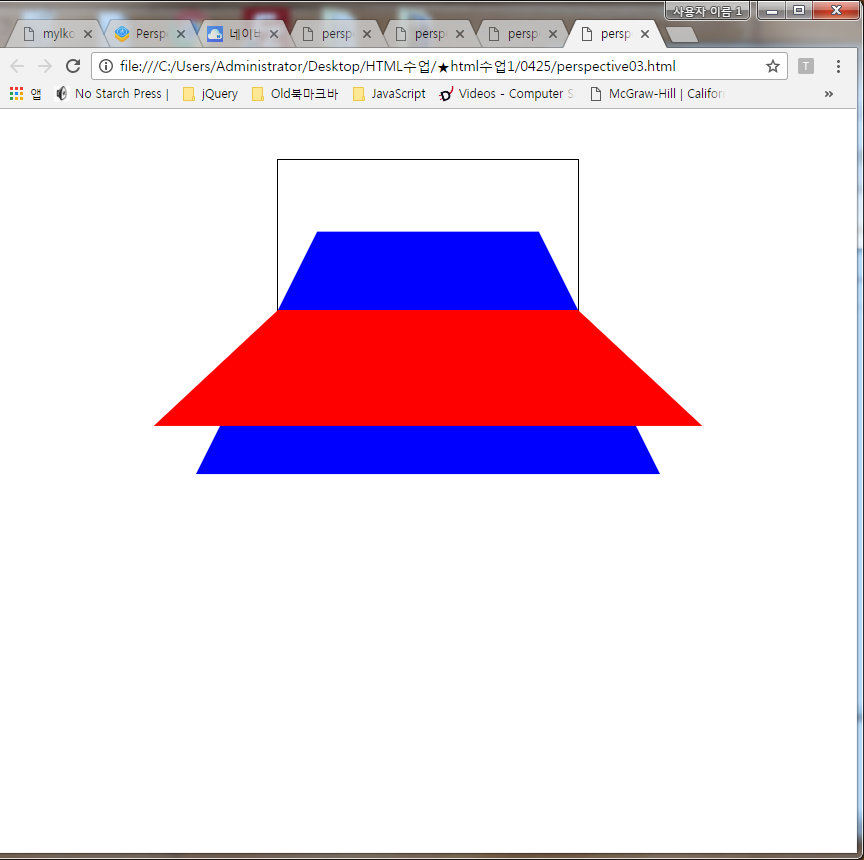
**설명**

perspective 속성은 상속되지 않는다. perspective 속성은 오직 직계 자식에만 적용된다. transform-style 속성은 요소가 2D에서 동작할 것인지 또는 3D에서 동작할 것인지를 명시할 수 있다. 기본값은 flat인데 이 속성에 preserve-3d 값을 적용하면 전환되는 요소의 자식요소가 그의 부모와 동일한 관점에서 그리고 동일한 3D 구현 상황에서 동작하게 한다.

다음은 전환되는 요소(child)가 자식(grandchild)을 갖고 있을 때 preserve-3d 값이 적용되지 않아 3D 회전이 발생하지 않은 예이다.

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title></title>  <style type="text/css">  .container {  perspective: 300px;  }  .child {  -webkit-transform: rotateX(45deg);  }  .grandchild {  -webkit-transform: rotateX(20deg);  }  </style>  </head>  <body>  <div class="container">  <div class="child">  <div class="grandchild">  </div>  </div>  </div>  </body>  </html> |

다음은 .grandchild 요소가 그의 부모와 동일한 관점에서 그리고 동일한 3D 구현 상황에서 동작하게 하기 위해서 그 부모에 preserve-3d 값을 추가한 예이다.



|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title></title>  <style type="text/css">  .container {  perspective: 300px;  width: 300px;  border:1px solid black;  margin:50px auto 50px;    }  .child {  background-color: blue;  transform: rotateX(45deg);  transform-style: preserve-3d;  width: 300px; height: 300px;    }  .grandchild {  background-color: red;  transform: rotateX(20deg);  width: 300px; height: 300px;    }  </style>  </head>  <body>  <div class="container">  <div class="child">  <div class="grandchild">  </div>  </div>  </div>  </body>  </html> |

## backface-visibility (뒷면 보기)

**설명**

backface-visibility 속성은 관찰자와 마주할 때 요소의 뒷면이 보여야 하는지 그렇지 않은지를 나타낼 수 있다. 기본값은 visible 이며 요소의 뒷면이 보이지 않도록 하기 위해서는 hidden 값을 사용한다.

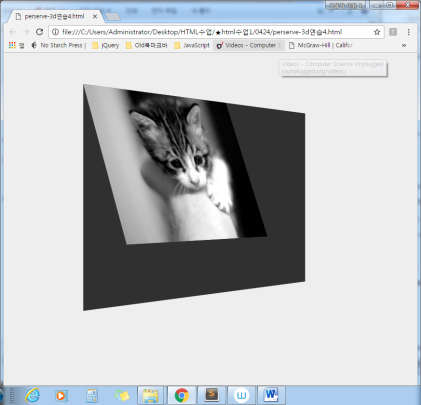
다음 예제에는 카드의 앞면과 뒷면을 보여주기 위해 두개의 이미지들을 준비하고 position:absolute로 설정하여 서로 위에 쌓아 놓는다. 그리고 .front 이미지는 y축을 기준으로 회전시켜서 관찰자에게 보이지 않게 한다. 문제는 마크업의 마지막에 .front 이미지가 명시되어 있어 .back 이미지 위로 나타나게 된다. 즉 관찰자는 뒤집혀진 .front 이미지의 뒷면을 보고 있게 된다.

|  |
| --- |
| <!DOCTYPE html> 작동가능한 코드는 아래에 따로 작성됨  <html>  <head>  <title></title>  <style type="text/css">  body > div {  width: 224px;  margin: 50px auto;  }  .container {  -webkit-perspective: 700px;  }  .card {  -webkit-transition: 2s;  -webkit-transform-style:preserve-3d;  }  .card img {  position: absolute;  /\*backface-visibility: hidden; 뒷면 이미지가 보이지 않게 된다\*/  }  .card .front {  -webkit-transform:rotateY(-180deg);  }  </style>  </head>  <body>  <div class="container" id="flip">  <figure class="card">  <img class="back" src="back.png" alt="Back Face">  <img class="front" src="sheep.png" alt="Queen of Hearts">  </figure>  </div>  </body>  </html> |

이것은 의도한 상황이 아니다. 이 상황을 해결하기 위해서 뒤집혀진 이미지의 뒷면이 더 이상 보이지 않도록 backface-visibility 속성을 사용할 수 있다.

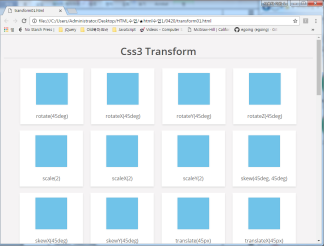
**완성된 코드는**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title></title>  <style type="text/css">  body > div {  width: 224px;  margin: 50px auto;  }  .container {  width: 224px;  height: 313px;  margin-top: 30px;  -webkit-perspective: 700px;  }  .card {  margin: 0;  -webkit-transition: 2s;  -webkit-transform-style:preserve-3d;  }  .container:hover .card {  -webkit-transform: rotateY(-180deg);  }  .card img {  position: absolute;  -webkit-backface-visibility: hidden;  }  .card .front {  -webkit-transform:rotateY(-180deg);  }  </style>  </head>  <body>  <div>  <div class="container" id="flip">  <figure class="card">  <img class="back" src="back.jpg" alt="Back Face">  <img class="front" src="q-hearts.jpg" alt="Queen of Hearts">  </figure>  </div>  </div>  </body>  </html> |



|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title></title>  <style type="text/css">  body {  background-color: #eee;  color: #555;  font-size: 1.1em;  font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif;  }  .container {  margin: 100px auto;  max-width: 700px;  /\* activate 3D space \*/  -webkit-perspective: 1800px;  perspective: 1800px;  }  .element {  width: 100%;  height: 400px;  background-color: rgba(0, 0, 0, 0.8);  -webkit-transform: rotateY(50deg);  transform: rotateY(50deg);  -webkit-transform-style: preserve-3d;  transform-style: preserve-3d;  }  .element img {  -webkit-transform-origin: 50% 0%;  -ms-transform-origin: 50% 0%;  transform-origin: 50% 0%;  /\* animate the image's transformation for demonstration purposes \*/  -webkit-animation: swing 3s linear infinite;  animation: swing 3s linear infinite;  }  @-webkit-keyframes swing {  0% {  -webkit-transform: rotateX(-30deg);  transform: rotateX(-30deg);  }  50% {  -webkit-transform: rotateX(30deg);  transform: rotateX(30deg);  }  100% {  -webkit-transform: rotateX(-30deg);  transform: rotateX(-30deg);  }  }  @keyframes swing {  0% {  -webkit-transform: rotateX(-30deg);  transform: rotateX(-30deg);  }  50% {  -webkit-transform: rotateX(30deg);  transform: rotateX(30deg);  }  100% {  -webkit-transform: rotateX(-30deg);  transform: rotateX(-30deg);  }  }  </style>  </head>  <body>  <div class="container">  <div class="element">  <img src="http://tympanus.net/codrops-playground/assets/images/cssref/properties/transform-style/transform-cat.jpg" alt="">  </div>  </div>  </body>  </html> |

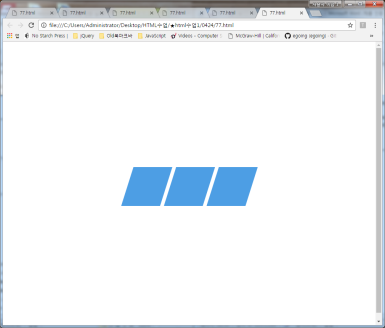
위 transform-origin 애니를 보려면 http://cssreference.io/property/transform-origin/



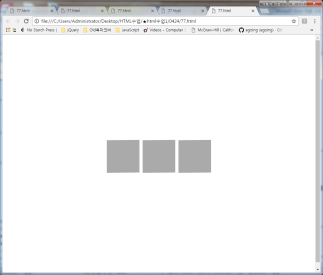
<https://codepen.io/vineethtr/pen/XKKEgM>

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title></title>  <style type="text/css">  @import url(https://fonts.googleapis.com/css?family=Open+Sans:400,300,400italic,600,700,800);  \*, \*:after, \*:before {  box-sizing: border-box;  }  body {  background: #F5F3F4;  margin: 0;  padding: 10px;  font-family: 'Open Sans', sans-serif;  text-align: center;  }  h1 {  color: #4c4c4c;  font-weight: 600;  border-bottom: 1px solid #ccc;  }  h2, h4 {  font-weight: 400;  color: #4d4d4d;  }  .card {  display: inline-block;  margin: 10px;  background: #fff;  padding: 15px;  min-width: 200px;  box-shadow: 0 3px 5px #ddd;  color: #555;  }  .card .box {  width: 100px;  height: 100px;  margin: auto;  background: #ddd;  cursor: pointer;  box-shadow: 0 0 5px #ccc inset;  }  .card .box .fill {  width: 100px;  height: 100px;  position: relative;  background: #03A9F4;  opacity: .5;  box-shadow: 0 0 5px #ccc;  -webkit-transition: 0.3s;  transition: 0.3s;  }  .card p {  margin: 25px 0 0;  }  .rotate:hover .fill {  -webkit-transform: rotate(45deg);  transform: rotate(45deg);  }  .rotateX:hover .fill {  -webkit-transform: rotateX(45deg);  transform: rotateX(45deg);  }  .rotateY:hover .fill {  -webkit-transform: rotateY(45deg);  transform: rotateY(45deg);  }  .rotateZ:hover .fill {  -webkit-transform: rotate(45deg);  transform: rotate(45deg);  }  .scale:hover .fill {  -webkit-transform: scale(2, 2);  transform: scale(2, 2);  }  .scaleX:hover .fill {  -webkit-transform: scaleX(2);  transform: scaleX(2);  }  .scaleY:hover .fill {  -webkit-transform: scaleY(2);  transform: scaleY(2);  }  .skew:hover .fill {  -webkit-transform: skew(45deg, 45deg);  transform: skew(45deg, 45deg);  }  .skewX:hover .fill {  -webkit-transform: skewX(45deg);  transform: skewX(45deg);  }  .skewY:hover .fill {  -webkit-transform: skewY(45deg);  transform: skewY(45deg);  }  .translate:hover .fill {  -webkit-transform: translate(45px, 1em);  transform: translate(45px, 1em);  }  .translateX:hover .fill {  -webkit-transform: translateX(45px);  transform: translateX(45px);  }  .translateY:hover .fill {  -webkit-transform: translateY(45px);  transform: translateY(45px);  }  .matrix:hover .fill {  -webkit-transform: matrix(2, 2, 0, 2, 45, 0);  transform: matrix(2, 2, 0, 2, 45, 0);  }  .perspective-100 .box {  -webkit-perspective: 100px;  perspective: 100px;  }  .perspective-200 .box {  -webkit-perspective: 200px;  perspective: 200px;  }  .to-100-0-0 {  -webkit-transform-origin: 100% 0 0;  transform-origin: 100% 0 0;  }  .to-0-100-0 {  -webkit-transform-origin: 0 100% 0;  transform-origin: 0 100% 0;  }    </style>  </head>  <body>  <h1>Css3 Transform</h1>  <!-- Rotate-->  <div class="card">  <div class="box rotate">  <div class="fill"></div>  </div>  <p>rotate(45deg) </p>  </div>  <div class="card">  <div class="box rotateX">  <div class="fill"></div>  </div>  <p>rotateX(45deg)</p>  </div>  <div class="card">  <div class="box rotateY">  <div class="fill"></div>  </div>  <p>rotateY(45deg)</p>  </div>  <div class="card">  <div class="box rotateZ">  <div class="fill"></div>  </div>  <p>rotateZ(45deg) </p>  </div>  <!-- scale-->  <div class="card">  <div class="box scale">  <div class="fill"></div>  </div>  <p>scale(2)</p>  </div>  <div class="card">  <div class="box scaleX">  <div class="fill"></div>  </div>  <p>scaleX(2) </p>  </div>  <div class="card">  <div class="box scaleY">  <div class="fill"></div>  </div>  <p>scaleY(2) </p>  </div>  <!-- skew-->  <div class="card">  <div class="box skew">  <div class="fill"></div>  </div>  <p>skew(45deg, 45deg) </p>  </div>  <div class="card">  <div class="box skewX">  <div class="fill"></div>  </div>  <p>skewX(45deg)</p>  </div>  <div class="card">  <div class="box skewY">  <div class="fill"></div>  </div>  <p>skewY(45deg)</p>  </div>  <!-- translate-->  <div class="card">  <div class="box translate">  <div class="fill"></div>  </div>  <p>translate(45px) </p>  </div>  <div class="card">  <div class="box translateX">  <div class="fill"></div>  </div>  <p>translateX(45px)</p>  </div>  <div class="card">  <div class="box translateY">  <div class="fill"></div>  </div>  <p>translateY(45px)</p>  </div>  <div class="card">  <div class="box matrix">  <div class="fill"></div>  </div>  <p> matrix(2, 2, 0, 2, 45, 0)</p>  </div>  <h4>Perspective : 100</h4>  <div class="perspective-100">  <div class="card">  <div class="box rotateX">  <div class="fill"></div>  </div>  <p>rotateX(90deg)</p>  </div>  <div class="card">  <div class="box rotateY">  <div class="fill"></div>  </div>  <p>rotateY(45deg)</p>  </div>  </div>  <h4>Perspective : 200</h4>  <div class="perspective-200">  <div class="card">  <div class="box rotateX">  <div class="fill"></div>  </div>  <p>rotateX(90deg)</p>  </div>  <div class="card">  <div class="box rotateY">  <div class="fill"></div>  </div>  <p>rotateY(45deg)</p>  </div>  </div>  <!-- transform origin-->  <h2>Transform origin</h2>  <div class="card">  <div class="box rotate">  <div class="fill to-100-0-0"></div>  </div>  <p>transform-origin : 100% 0 0 <br/>rotate(45deg)</p>  </div>  <div class="card">  <div class="box rotate">  <div class="fill to-0-100-0"></div>  </div>  <p>transform-origin : 0 100% 0<br/>rotate(45deg)</p>  </div>  <div class="card perspective-200">  <div class="box rotateX">  <div class="fill to-0-100-0"></div>  </div>  <p>transform-origin : 0 100% 0<br/>rotateX(45deg)</p>  </div>  <div class="card perspective-200">  <div class="box rotateX">  <div class="fill to-100-0-0"></div>  </div>  <p>transform-origin : 0 100% 0<br/>rotateX(45deg)</p>  </div>  <div class="card perspective-200">  <div class="box rotateY">  <div class="fill to-0-100-0"></div>  </div>  <p>transform-origin : 0 100% 0 <br/>rotateY(45deg)</p>  </div>  <div class="card perspective-200">  <div class="box rotateY">  <div class="fill to-100-0-0"></div>  </div>  <p>transform-origin : 100% 0 0<br/>rotateY(45deg)</p>  </div>  <div class="card">  <div class="box scale">  <div class="fill to-100-0-0"></div>  </div>  <p>transform-origin : 100% 0 0<br/>scale(2)</p>  </div>  <div class="card">  <div class="box scale">  <div class="fill to-0-100-0"></div>  </div>  <p>transform-origin : 0 100% 0<br/>scale(2)</p>  </div>  <div class="card">  <div class="box scaleX">  <div class="fill to-100-0-0"></div>  </div>  <p>transform-origin : 100% 0 0<br/>scaleX(2)</p>  </div>  <div class="card">  <div class="box scaleX">  <div class="fill to-0-100-0"></div>  </div>  <p>transform-origin : 0 100% 0<br/>scaleX(2)</p>  </div>  <div class="card">  <div class="box scaleY">  <div class="fill to-100-0-0"></div>  </div>  <p>transform-origin : 100% 0 0<br/>scaleY(2)</p>  </div>  <div class="card">  <div class="box scaleY">  <div class="fill to-0-100-0"></div>  </div>  <p>transform-origin : 0 100% 0<br/>scaleY(2)</p>  </div>  </body>  </html> |

**예제1**



|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title></title>  <style type="text/css">    .element {  display: inline-block;  background-color: #0074d9;  height: 100px;  width: 100px;  font-size: 1px;  padding: 1px;  color: white;  margin-right: 5px;  margin-left: 5px;  -webkit-animation: skew 3s infinite;  animation: skew 3s infinite;  -webkit-transform: skew(20deg);  transform: skew(20deg);  -webkit-animation-direction: alternate;  animation-direction: alternate;  opacity: .7;  }  @-webkit-keyframes skew {  0% {  -webkit-transform: skewX(20deg);  transform: skewX(20deg);  }  100% {  -webkit-transform: skewX(-20deg);  transform: skewX(-20deg);  }  }  @keyframes skew {  0% {  -webkit-transform: skewX(20deg);  transform: skewX(20deg);  }  100% {  -webkit-transform: skewX(-20deg);  transform: skewX(-20deg);  }  }  body, html {  height: 100%;  }  body {  display: -webkit-box;  display: -ms-flexbox;  display: flex;  -webkit-box-align: center;  -ms-flex-align: center;  align-items: center;  -webkit-box-pack: center;  -ms-flex-pack: center;  justify-content: center;  }  </style>  </head>  <body>  <div class="element">  </div>  <div class="element">  </div>  <div class="element">  </div>  </body>  </html> |



|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title></title>  <style type="text/css">  .element {  display: inline-block;  background-color: #aaaaaa;  height: 100px;  width: 100px;  -webkit-transform: skew(20deg);  transform: skew(20deg);  font-size: 1px;  padding: 1px;  color: white;  margin-right: 5px;  margin-left: 5px;  -webkit-animation: shake 2s infinite;  animation: shake 2s infinite;  -webkit-animation-direction: alternate;  animation-direction: alternate;  }  @-webkit-keyframes shake {  0% {  -webkit-transform: skewY(20deg);  transform: skewY(20deg);  }  100% {  -webkit-transform: skewY(-20deg);  transform: skewY(-20deg);  }  }  @keyframes shake {  0% {  -webkit-transform: skewY(20deg);  transform: skewY(20deg);  }  100% {  -webkit-transform: skewY(-20deg);  transform: skewY(-20deg);  }  }  body, html {  height: 100%;  }  body {  display: -webkit-box;  display: -ms-flexbox;  display: flex;  -webkit-box-align: center;  -ms-flex-align: center;  align-items: center;  -webkit-box-pack: center;  -ms-flex-pack: center;  justify-content: center;  }  </style>  </head>  <body>  <div class="element">  </div>  <div class="element">  </div>  <div class="element">  </div>  </body>  </html> |

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title></title>  <style type="text/css">  .element {  display: inline-block;  background-color: #0074d9;  height: 100px;  width: 100px;  font-size: 1px;  padding: 1px;  color: white;  margin-right: 5px;  margin-left: 5px;  -webkit-animation: roll 3s infinite;  animation: roll 3s infinite;  -webkit-transform: rotate(30deg);  transform: rotate(30deg);  opacity: .7;  }  @-webkit-keyframes roll {  0% {  -webkit-transform: rotate(0);  transform: rotate(0);  }  100% {  -webkit-transform: rotate(360deg);  transform: rotate(360deg);  }  }  @keyframes roll {  0% {  -webkit-transform: rotate(0);  transform: rotate(0);  }  100% {  -webkit-transform: rotate(360deg);  transform: rotate(360deg);  }  }  body, html {  height: 100%;  }  body {  display: -webkit-box;  display: -ms-flexbox;  display: flex;  -webkit-box-align: center;  -ms-flex-align: center;  align-items: center;  -webkit-box-pack: center;  -ms-flex-pack: center;  justify-content: center;  }  </style>  </head>  <body>  <div class="element">  </div>  <div class="element">  </div>  <div class="element">  </div>  </body>  </html> |

## **CSS3로 움직임을 만들때는 2가지 속성을 이용한다.**

**The @keyframes Rule**

**animation Property**

The animation property is a shorthand property for eight of the animation properties:

* animation-name
* animation-duration
* animation-timing-function
* animation-delay
* animation-iteration-count
* animation-direction
* animation-fill-mode
* animation-play-state ( 축약에 포함안됨)

**Note:** Always specify the animation-duration property, otherwise the duration is 0, and will never be played.

|  |  |
| --- | --- |
| **Default value:** | none 0 ease 0 1 normal none running |

## **Property Values**

[animation](https://developer.mozilla.org/ko/docs/Web/CSS/animation) 속성의 하위 속성은 다음과 같습니다.

 **animation-name** : 선언된 애니메이션 이름

 **animation-duration**: ms 단위로 실행되는 시간을 지정–––––––

 **animation-timing-function** : linear, ease 등의 시간에 따른 변화 가감속 속도를 지정

 **animation-iteration-count** : 애니메이션의 반복 효과 횟수 지정(기본값은 1로 지정되어 있음)

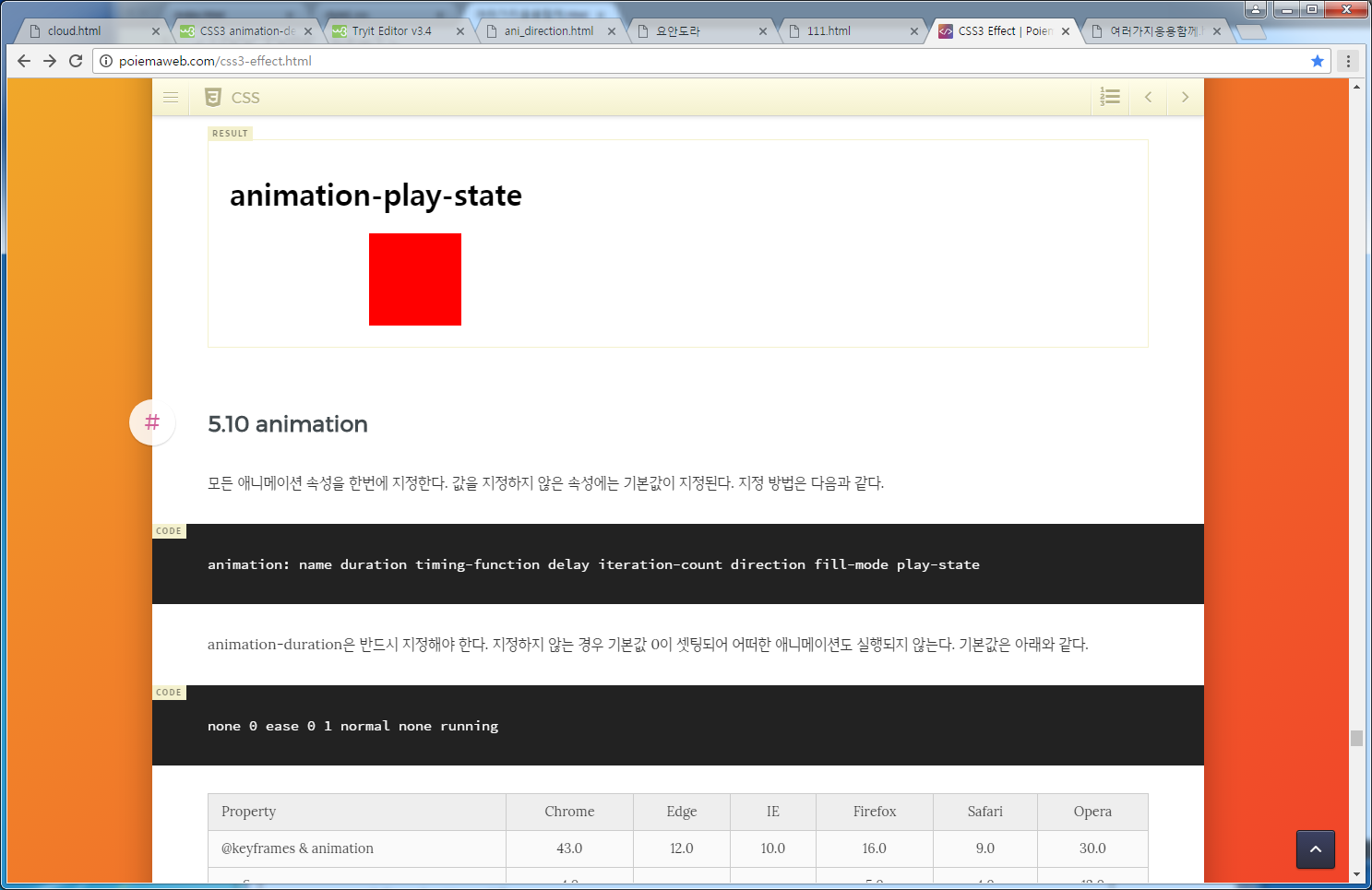
**animation-delay**

 **animation-direction** : 애니메이션이 반복되는 동안 실행되는 순서(기본은 normal, 반대라면 alternate) ( 앞뒤로 왔다 갔다 하게 할때 사용하면 좋다)

 **animation-delay** : 애니메이션 실행 시작 전에 실행 지연(유예) 시간을 지정, 음수의 값을 사용하면 그 시간만큼 애니메이션을 건너뛰고 진행합니다.

 **animation-fill-mode** : 애니메이션의 적용 후에 대상의 상태를 지정(backward, forward, both, none)

 **animation-play-state** : 애니메이션 실행 중에 중지되었을 때 계속 진행하거나 멈추게 할 것인지를 지정 ( 축약형에 사용할 수 없다 )  
  
animation



참조 : http://poiemaweb.com/css3-effect.html

|  |
| --- |
| html에서 ( **[animation-direction](https://developer.mozilla.org/ko/docs/Web/CSS/animation-direction" \o "The documentation about this has not yet been written; please consider contributing!)**)  <p>The Caterpillar and Alice looked at each other for some time in silence:  at last the Caterpillar took the hookah out of its mouth, and addressed  her in a languid, sleepy voice.</p>  css 에서  p {  animation-duration: 3s;  animation-name: slidein;  animation-iteration-count: infinite;  animation-direction: alternate;  }  @keyframes slidein {  from {  margin-left: 100%;  width: 300%;  }  to {  margin-left: 0%;  width: 100%;  }  } |

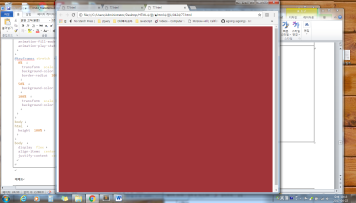
## **The @keyframes Rule**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <style>  div {  width: 100px;  height: 100px;  background-color: red;  -webkit-animation-name: example; /\* Safari 4.0 - 8.0 \*/  -webkit-animation-duration: 4s; /\* Safari 4.0 - 8.0 \*/  animation-name: **example;**  animation-duration: 4s;  }  /\* Standard syntax \*/  @keyframes **example** {  from {background-color: red;}  to {background-color: yellow;}  }  </style>  </head>  <body>  <p><b>Note:</b> This example does not work in Internet Explorer 9 and earlier versions.</p>  <div></div>  <p><b>Note:</b> When an animation is finished, it changes back to its original style.</p>  </body>  </html> |

위의 예제를 아래와 같이 바꿀 수 있다.

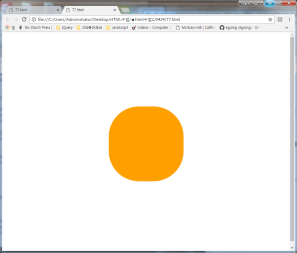
|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <style>  div {  width: 100px;  height: 100px;  background-color: red;  -webkit-animation-name: example; /\* Safari 4.0 - 8.0 \*/  -webkit-animation-duration: 4s; /\* Safari 4.0 - 8.0 \*/  animation-name: example;  animation-duration: 4s;  }  /\* Safari 4.0 - 8.0 \*/  @-webkit-keyframes example {  0% {background-color: red;}  25% {background-color: yellow;}  50% {background-color: blue;}  100% {background-color: green;}  }  /\* Standard syntax \*/  @keyframes example {  0% {background-color: red;}  25% {background-color: yellow;}  50% {background-color: blue;}  100% {background-color: green;}  }  </style>  </head>  <body>  <p><b>Note:</b> This example does not work in Internet Explorer 9 and earlier versions.</p>  <div></div>  </body>  </html> |

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <style>  div {  width: 100px;  height: 100px;  background-color: red;  position: relative;  -webkit-animation-name: example; /\* Safari 4.0 - 8.0 \*/  -webkit-animation-duration: 4s; /\* Safari 4.0 - 8.0 \*/  -webkit-animation-iteration-count: 3; /\* Safari 4.0 - 8.0 \*/  -webkit-animation-direction: reverse; /\* Safari 4.0 - 8.0 \*/  animation-name: example;  animation-duration: 4s;  animation-iteration-count: 3;  animation-direction: reverse;  }  /\* Safari 4.0 - 8.0 \*/  @-webkit-keyframes example {  0% {background-color:red; left:0px; top:0px;}  25% {background-color:yellow; left:200px; top:0px;}  50% {background-color:blue; left:200px; top:200px;}  75% {background-color:green; left:0px; top:200px;}  100% {background-color:red; left:0px; top:0px;}  }  /\* Standard syntax \*/  @keyframes example {  0% {background-color:red; left:0px; top:0px;}  25% {background-color:yellow; left:200px; top:0px;}  50% {background-color:blue; left:200px; top:200px;}  75% {background-color:green; left:0px; top:200px;}  100% {background-color:red; left:0px; top:0px;}  }  </style>  </head>  <body>  <p><b>Note:</b> This example does not work in Internet Explorer 9 and earlier versions.</p>  <div></div>  </body>  </html> |

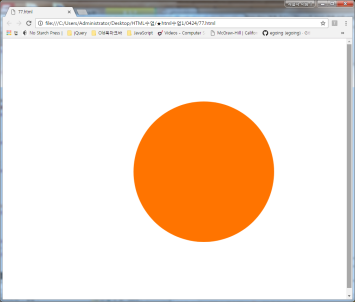
**예제1)** 

|  |
| --- |
| Html  <div class="element"></div> |
| Css  .element {  width: 100%;  height: 100%;  animation: pulse 5s infinite; }  @keyframes pulse {  0% {  background-color: #001F3F;  }  100% {  background-color: #FF4136;  } }  html, body {  height: 100%; } |

**예제2**

)

|  |
| --- |
| Html  <div class="element"></div> |
| Css  .element {  height: 250px;  width: 250px;  margin: 0 auto;  background-color: red;  animation-name: stretch;  animation-duration: 1.5s;   animation-timing-function: ease-out;   animation-delay: 0;  animation-direction: alternate;  animation-iteration-count: infinite;  animation-fill-mode: none;  animation-play-state: running; }  @keyframes stretch {  0% {  transform: scale(.3);  background-color: red;  border-radius: 100%;  }  50% {  background-color: orange;  }  100% {  transform: scale(1.5);  background-color: yellow;  } }  body, html {  height: 100%; }  body {  display: flex;  align-items: center;  justify-content: center; } |



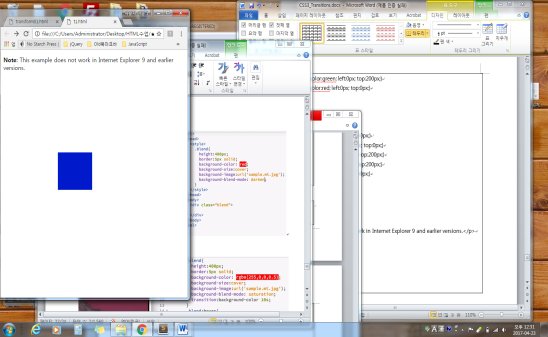
예제3)

|  |
| --- |
| Html  <div class="element"></div>  Css  .element {  height: 400px;  width: 400px;  background-color: red;  animation:   pulse 3s ease infinite alternate,   nudge 5s linear infinite alternate;  border-radius: 100%; }  @keyframes pulse {  0%, 100% {  background-color: red;  }  50% {  background-color: orange;  } }  @keyframes nudge {  0%, 100% {  transform: translate(0, 0);  }    50% {  transform: translate(150px, 0);  }    80% {  transform: translate(-150px, 0);  } }   html, body {  height: 100%; }  body {  display: flex;  align-items: center;  justify-content: center; } |

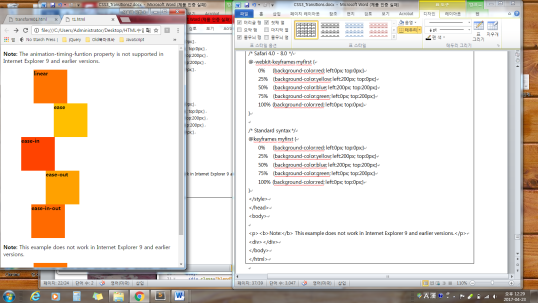
예제4)

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title></title>  <style type="text/css">  .photos img {  position: absolute;}  /\*.photos img {  -webkit-transition: all 0.5s ease-out;  -moz-transition: all 0.5s ease-out;  -o-transition: all 0.5s ease-out;  transition: all 0.5s ease-out;  }\*/  .photos img {  -webkit-transition: all 0.5s ease-out;  -moz-transition: all 0.5s ease-out;  -o-transition: all 0.5s ease-out;  transition: all 0.5s ease-out;  padding: 10px 10px 30px 10px;  background: white;  border: solid 1px black;  }  /\* .photos img:nth-of-type(1) {  left: 50px;  top: 50px;  }    .photos img:nth-of-type(2) {  left: 150px;  top: 100px;  }    .photos img:nth-of-type(3) {  left: 250px;  top: 50px;  }    .photos img:nth-of-type(4) {  left: 350px;  top: 150px;  }    .photos img:nth-of-type(5) {  left: 450px;  top: 50px;  }  \*/  .photos img:nth-of-type(1) {  left: 50px;  top: 50px;  -webkit-transform: rotate(5deg);  -moz-transform: rotate(5deg);  -o-transform: rotate(5deg);  transform: rotate(5deg);  }    .photos img:nth-of-type(2) {  left: 150px;  top: 100px;  -webkit-transform: rotate(-10deg);  -moz-transform: rotate(-10deg);  -o-transform: rotate(-10deg);  transform: rotate(-10deg);  }    .photos img:nth-of-type(3) {  left: 250px;  top: 50px;  -webkit-transform: rotate(7deg);  -moz-transform: rotate(7deg);  -o-transform: rotate(7deg);  transform: rotate(7deg);  }    .photos img:nth-of-type(4) {  left: 350px;  top: 150px;  -webkit-transform: rotate(-3deg);  -moz-transform: rotate(-3deg);  -o-transform: rotate(-3deg);  transform: rotate(-3deg);  }    .photos img:nth-of-type(5) {  left: 450px;  top: 50px;  -webkit-transform: rotate(2deg);  -moz-transform: rotate(2deg);  -o-transform: rotate(2deg);  transform: rotate(2deg);  }  /\*.photos img:hover {  -webkit-transform: scale(1.5);  -moz-transform: scale(1.5);  -o-transform: scale(1.5);  transform: scale(1.5);  z-index: 10;  }\*/  .photos img:hover {  -webkit-transform: rotate(380deg) scale(1.5);  -moz-transform: rotate(380deg) scale(1.5);  -o-transform: rotate(380deg) scale(1.5);  transform: rotate(380deg) scale(1.5);  z-index: 10;  }  </style>  </head>  <body>  <div class="photos">  <img src="137.png" />  <img src="137.png" />  <img src="137.png" />  <img src="137.png" />  <img src="137.png" />  </div>  </body>  </html> |

예제)



|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <style>  div {  width: 100px;  height: 100px;  background-color: red;  position: relative;  -webkit-animation-name: example; /\* Safari 4.0 - 8.0 \*/  -webkit-animation-duration: 4s; /\* Safari 4.0 - 8.0 \*/  -webkit-animation-iteration-count: 3; /\* Safari 4.0 - 8.0 \*/  -webkit-animation-direction: reverse; /\* Safari 4.0 - 8.0 \*/  animation-name: example;  animation-duration: 4s;  animation-iteration-count: 3;  animation-direction: reverse;  }  /\* Safari 4.0 - 8.0 \*/  @-webkit-keyframes example {  0% {background-color:red; left:0px; top:0px;}  25% {background-color:yellow; left:200px; top:0px;}  50% {background-color:blue; left:200px; top:200px;}  75% {background-color:green; left:0px; top:200px;}  100% {background-color:red; left:0px; top:0px;}  }  /\* Standard syntax \*/  @keyframes example {  0% {background-color:red; left:0px; top:0px;}  25% {background-color:yellow; left:200px; top:0px;}  50% {background-color:blue; left:200px; top:200px;}  75% {background-color:green; left:0px; top:200px;}  100% {background-color:red; left:0px; top:0px;}  }  </style>  </head>  <body>  <p><b>Note:</b> This example does not work in Internet Explorer 9 and earlier versions.</p>  <div></div>  </body>  </html> |

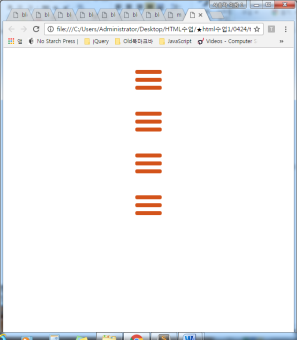


|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <style>  div {  width: 100px;  height: 50px;  background-color: red;  font-weight: bold;  position: relative;  -webkit-animation: mymove 5s infinite; /\* Safari 4.0 - 8.0 \*/  animation: mymove 5s infinite;  }  /\* Safari 4.0 - 8.0 \*/  #div1 {-webkit-animation-timing-function: linear;}  #div2 {-webkit-animation-timing-function: ease;}  #div3 {-webkit-animation-timing-function: ease-in;}  #div4 {-webkit-animation-timing-function: ease-out;}  #div5 {-webkit-animation-timing-function: ease-in-out;}  /\* Standard syntax \*/  #div1 {animation-timing-function: linear;}  #div2 {animation-timing-function: ease;}  #div3 {animation-timing-function: ease-in;}  #div4 {animation-timing-function: ease-out;}  #div5 {animation-timing-function: ease-in-out;}  /\* Safari 4.0 - 8.0 \*/  @-webkit-keyframes mymove {  from {left: 0px;}  to {left: 300px;}  }  /\* Standard syntax \*/  @keyframes mymove {  from {left: 0px;}  to {left: 300px;}  }  </style>  </head>  <body>  <p><strong>Note:</strong> The animation-timing-funtion property is not supported in Internet Explorer 9 and earlier versions.</p>  <div id="div1">linear</div>  <div id="div2">ease</div>  <div id="div3">ease-in</div>  <div id="div4">ease-out</div>  <div id="div5">ease-in-out</div>  </body>  </html>  <!DOCTYPE html>  <html>  <head>  <style>  div {  width: 100px;  height: 100px;  background-color: red;  position: relative;  -webkit-animation: myfirst 5s linear 2s infinite alternate; /\* Safari 4.0 - 8.0 \*/  animation: myfirst 5s linear 2s infinite alternate;  }  /\* Safari 4.0 - 8.0 \*/  @-webkit-keyframes myfirst {  0% {background-color:red; left:0px; top:0px;}  25% {background-color:yellow; left:200px; top:0px;}  50% {background-color:blue; left:200px; top:200px;}  75% {background-color:green; left:0px; top:200px;}  100% {background-color:red; left:0px; top:0px;}  }  /\* Standard syntax \*/  @keyframes myfirst {  0% {background-color:red; left:0px; top:0px;}  25% {background-color:yellow; left:200px; top:0px;}  50% {background-color:blue; left:200px; top:200px;}  75% {background-color:green; left:0px; top:200px;}  100% {background-color:red; left:0px; top:0px;}  }  </style>  </head>  <body>  <p><b>Note:</b> This example does not work in Internet Explorer 9 and earlier versions.</p>  <div></div>  </body>  </html> |

웹페이지 통째로 비틀기 예제 (확인필요)

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title></title>  <style type="text/css">  /\* body {  transform-origin: 50% 100%;  transform: rotateX(45deg);  }  \*/  /\*html {  perspective: 500px;  }\*/  /\*html {  perspective: 500px;  perspective-origin: 90% 50%;  }\*/  html, body {  width: 100%;  height: 100%;  min-width: 100%;  max-width: 100%;  padding: 0;  margin: 0;  overflow: hidden;  }    #Wrapper {  position: absolute;  width: 100%;  height: 100%;  overflow: scroll;  }  </style>  </head>  <body>  <!-- https://www.sitepoint.com/how-to-create-full-page-animations-using-css/ -->  <img src="137.png">  </body>  </html> |

**Transform**

 제이쿼리 들어감

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title></title>  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js"></script>  <script type="text/javascript">  $(document).ready(function(){  $('#nav-icon1,#nav-icon2,#nav-icon3,#nav-icon4').click(function(){  $(this).toggleClass('open');  });  });  </script>  <style type="text/css">  \* {  margin: 0;  padding: 0;  }  /\* Icon 1 \*/  #nav-icon1, #nav-icon2, #nav-icon3, #nav-icon4 {  width: 60px;  height: 45px;  position: relative;  margin: 50px auto;  -webkit-transform: rotate(0deg);  -moz-transform: rotate(0deg);  -o-transform: rotate(0deg);  transform: rotate(0deg);  -webkit-transition: .5s ease-in-out;  -moz-transition: .5s ease-in-out;  -o-transition: .5s ease-in-out;  transition: .5s ease-in-out;  cursor: pointer;  }  #nav-icon1 span, #nav-icon3 span, #nav-icon4 span {  display: block;  position: absolute;  height: 9px;  width: 100%;  background: #d3531a;  border-radius: 9px;  opacity: 1;  left: 0;  -webkit-transform: rotate(0deg);  -moz-transform: rotate(0deg);  -o-transform: rotate(0deg);  transform: rotate(0deg);  -webkit-transition: .25s ease-in-out;  -moz-transition: .25s ease-in-out;  -o-transition: .25s ease-in-out;  transition: .25s ease-in-out;  }  #nav-icon1 span:nth-child(1) {  top: 0px;  }  #nav-icon1 span:nth-child(2) {  top: 18px;  }  #nav-icon1 span:nth-child(3) {  top: 36px;  }  #nav-icon1.open span:nth-child(1) {  top: 18px;  -webkit-transform: rotate(135deg);  -moz-transform: rotate(135deg);  -o-transform: rotate(135deg);  transform: rotate(135deg);  }  #nav-icon1.open span:nth-child(2) {  opacity: 0;  left: -60px;  }  #nav-icon1.open span:nth-child(3) {  top: 18px;  -webkit-transform: rotate(-135deg);  -moz-transform: rotate(-135deg);  -o-transform: rotate(-135deg);  transform: rotate(-135deg);  }  /\* Icon 2 \*/  #nav-icon2 {  }  #nav-icon2 span {  display: block;  position: absolute;  height: 9px;  width: 50%;  background: #d3531a;  opacity: 1;  -webkit-transform: rotate(0deg);  -moz-transform: rotate(0deg);  -o-transform: rotate(0deg);  transform: rotate(0deg);  -webkit-transition: .25s ease-in-out;  -moz-transition: .25s ease-in-out;  -o-transition: .25s ease-in-out;  transition: .25s ease-in-out;  }  #nav-icon2 span:nth-child(even) {  left: 50%;  border-radius: 0 9px 9px 0;  }  #nav-icon2 span:nth-child(odd) {  left:0px;  border-radius: 9px 0 0 9px;  }  #nav-icon2 span:nth-child(1), #nav-icon2 span:nth-child(2) {  top: 0px;  }  #nav-icon2 span:nth-child(3), #nav-icon2 span:nth-child(4) {  top: 18px;  }  #nav-icon2 span:nth-child(5), #nav-icon2 span:nth-child(6) {  top: 36px;  }  #nav-icon2.open span:nth-child(1),#nav-icon2.open span:nth-child(6) {  -webkit-transform: rotate(45deg);  -moz-transform: rotate(45deg);  -o-transform: rotate(45deg);  transform: rotate(45deg);  }  #nav-icon2.open span:nth-child(2),#nav-icon2.open span:nth-child(5) {  -webkit-transform: rotate(-45deg);  -moz-transform: rotate(-45deg);  -o-transform: rotate(-45deg);  transform: rotate(-45deg);  }  #nav-icon2.open span:nth-child(1) {  left: 5px;  top: 7px;  }  #nav-icon2.open span:nth-child(2) {  left: calc(50% - 5px);  top: 7px;  }  #nav-icon2.open span:nth-child(3) {  left: -50%;  opacity: 0;  }  #nav-icon2.open span:nth-child(4) {  left: 100%;  opacity: 0;  }  #nav-icon2.open span:nth-child(5) {  left: 5px;  top: 29px;  }  #nav-icon2.open span:nth-child(6) {  left: calc(50% - 5px);  top: 29px;  }  /\* Icon 3 \*/  #nav-icon3 span:nth-child(1) {  top: 0px;  }  #nav-icon3 span:nth-child(2),#nav-icon3 span:nth-child(3) {  top: 18px;  }  #nav-icon3 span:nth-child(4) {  top: 36px;  }  #nav-icon3.open span:nth-child(1) {  top: 18px;  width: 0%;  left: 50%;  }  #nav-icon3.open span:nth-child(2) {  -webkit-transform: rotate(45deg);  -moz-transform: rotate(45deg);  -o-transform: rotate(45deg);  transform: rotate(45deg);  }  #nav-icon3.open span:nth-child(3) {  -webkit-transform: rotate(-45deg);  -moz-transform: rotate(-45deg);  -o-transform: rotate(-45deg);  transform: rotate(-45deg);  }  #nav-icon3.open span:nth-child(4) {  top: 18px;  width: 0%;  left: 50%;  }  /\* Icon 4 \*/  #nav-icon4 {  }  #nav-icon4 span:nth-child(1) {  top: 0px;  -webkit-transform-origin: left center;  -moz-transform-origin: left center;  -o-transform-origin: left center;  transform-origin: left center;  }  #nav-icon4 span:nth-child(2) {  top: 18px;  -webkit-transform-origin: left center;  -moz-transform-origin: left center;  -o-transform-origin: left center;  transform-origin: left center;  }  #nav-icon4 span:nth-child(3) {  top: 36px;  -webkit-transform-origin: left center;  -moz-transform-origin: left center;  -o-transform-origin: left center;  transform-origin: left center;  }  #nav-icon4.open span:nth-child(1) {  -webkit-transform: rotate(45deg);  -moz-transform: rotate(45deg);  -o-transform: rotate(45deg);  transform: rotate(45deg);  top: -3px;  left: 8px;  }  #nav-icon4.open span:nth-child(2) {  width: 0%;  opacity: 0;  }  #nav-icon4.open span:nth-child(3) {  -webkit-transform: rotate(-45deg);  -moz-transform: rotate(-45deg);  -o-transform: rotate(-45deg);  transform: rotate(-45deg);  top: 39px;  left: 8px;  }  </style>  </head>  <body>  <div id="nav-icon1">  <span></span>  <span></span>  <span></span>  </div>  <div id="nav-icon2">  <span></span>  <span></span>  <span></span>  <span></span>  <span></span>  <span></span>  </div>  <div id="nav-icon3">  <span></span>  <span></span>  <span></span>  <span></span>  </div>  <div id="nav-icon4">  <span></span>  <span></span>  <span></span>  </div>  </body>  </html> |

# CSS3 animation-fill-mode Property

- **animation-fill-mode**: 애니메이션 시작/끝 상태 제어

속성 : ( none | forwards | backwards | both )

**none** - 아무것도 지정되지 않은 상태

**forwards** - 애니매이션 키프레임이 100%에 도달했을때 그 상태를 유지

**backwards** - 애니메이션 스타일을 실제 애니메이션이 시작전에 미리 적용

**both** - forwards , backwards 둘다 적용함

출처: <http://mrb18.tistory.com/84> [The칸]

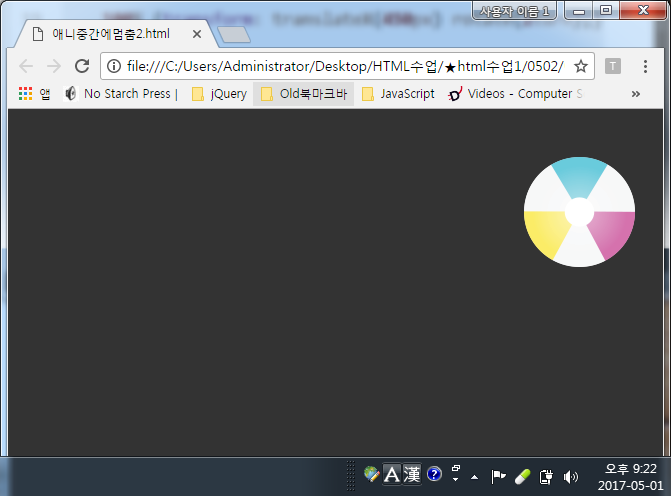
CSS Syntax : animation-fill-mode: none | forwards | backwards | both

|  |  |
| --- | --- |
| forwards | After the animation ends (determined by animation-iteration-count), the animation will apply the property values for the time the animation ended |

예제1)

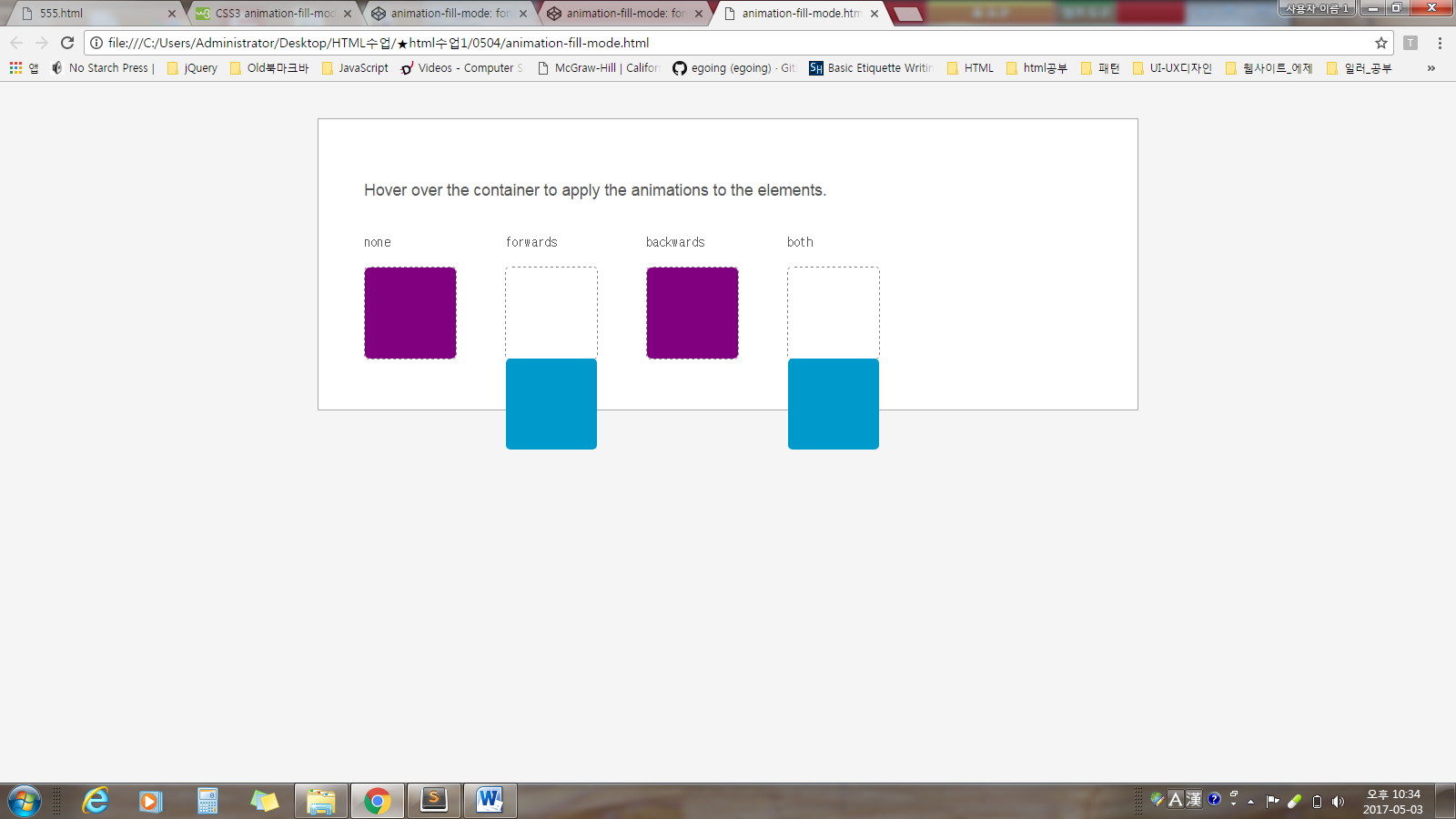
|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <style>  div {  width: 100px;  height: 100px;  background: red;  position: relative;  -webkit-animation: mymove 3s; /\* Safari 4.0 - 8.0 \*/  -webkit-animation-fill-mode: forwards; /\* Safari 4.0 - 8.0 \*/  animation: mymove 3s;  animation-fill-mode: backwards;  }  /\* Safari 4.0 - 8.0 \*/  @-webkit-keyframes mymove {  from {top: 0px;}  to {top: 200px;}  }  @keyframes mymove {  from {top: 0px;}  to {top: 200px;}  }  </style>  </head>  <body> |

예제 2)

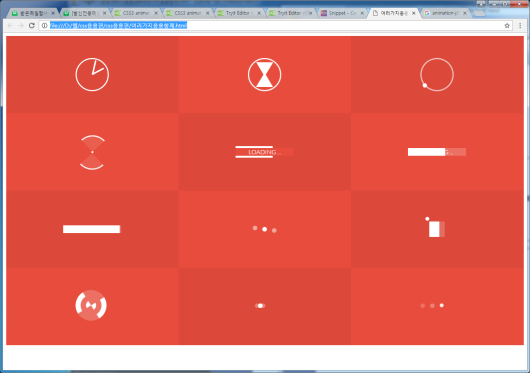


|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title></title>  <style type="text/css">  /\*using prefix free here so it's all nice and easier to read\*/  body {margin:3em 10%; background:#333;}    .ball {  animation-name: ballmove;  animation-duration: 2s;  animation-iteration-count:1;  animation-timing-function: ease-in-out;  animation-fill-mode:forwards /\* backwards, both \*/  }  @keyframes ballmove {  0% {transform: translateX(0) rotate(0);}  100% {transform: translateX(450px) rotate(2turn);} /\* 2번회전 \*/  }  </style>  </head>  <body>  <img class="ball" src="https://dl.dropbox.com/u/1143870/ball.png" alt="ball"width="111" height="111" />  </body>  </html> |

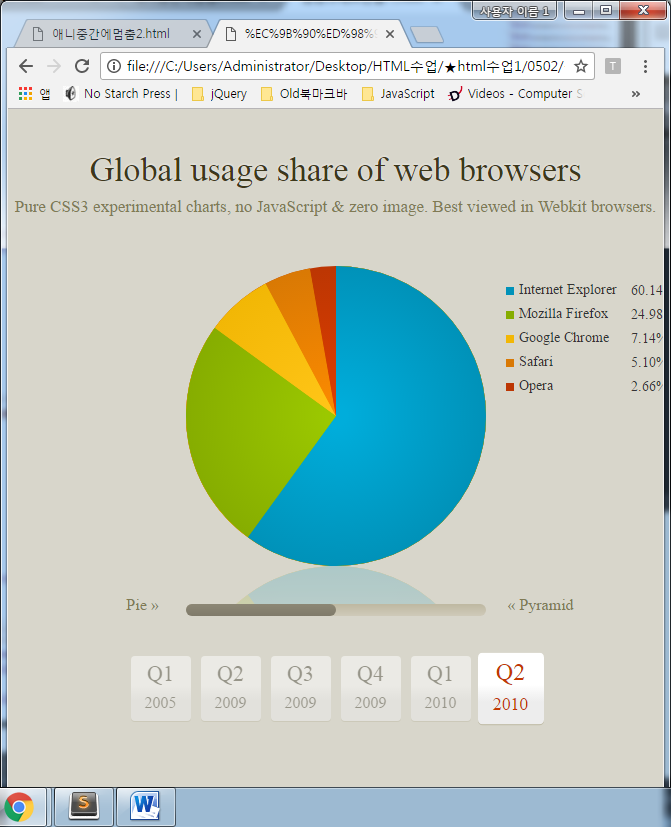
예제3)

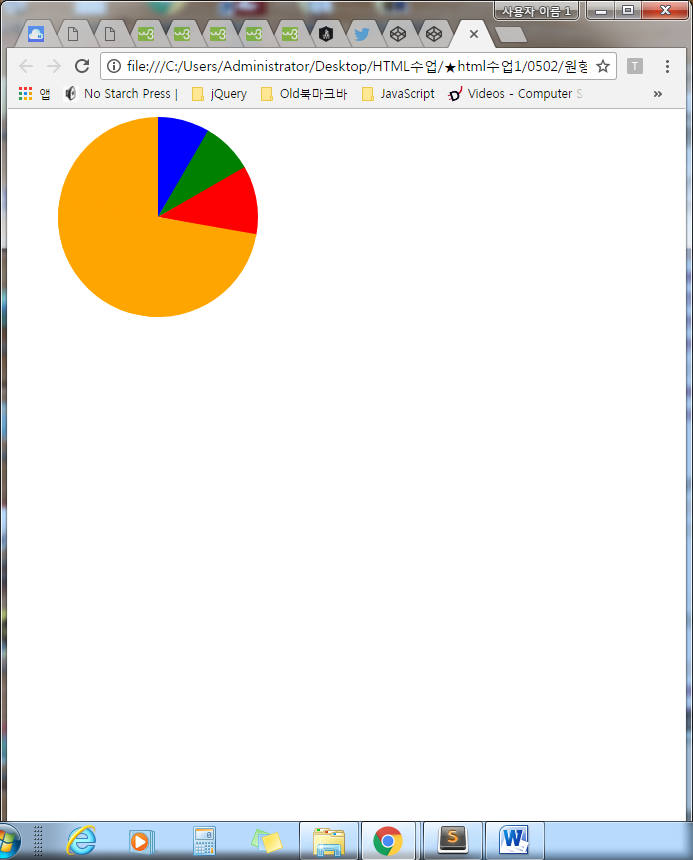


|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title></title>  <style type="text/css">  body {  background-color: #F5F5F5;  color: #555;  font-size: 1.1em;  font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif;  }  hr {  margin: 50px 0;  }  .container {  margin: 40px auto;  max-width: 800px;  background-color: white;  padding: 50px;  border: 1px solid #aaa;  }  .module {  width: 150px;  display: inline-block;  }  .wrapper {  width: 100px;  height: 100px;  border-radius: 5px;  border: 1px dashed grey;  display: inline-block;  }  .element {  width: 100px;  height: 100px;  border-radius: 5px;  background-color: purple;  -webkit-transform-origin: bottom left;  -ms-transform-origin: bottom left;  transform-origin: bottom left;  }  .container:hover .element {  -webkit-animation-name: rotate;  animation-name: rotate;  -webkit-animation-duration: 1s;  animation-duration: 1s;  -webkit-animation-timing-function: cubic-bezier(.18, 1.03, .5, 1.62);  animation-timing-function: cubic-bezier(.18, 1.03, .5, 1.62);  }  .element-2 {  -webkit-animation-fill-mode: forwards;  animation-fill-mode: forwards;  }  .element-3 {  -webkit-animation-delay: 1s;  animation-delay: 1s;  -webkit-animation-fill-mode: backwards;  animation-fill-mode: backwards;  }  .element-4 {  -webkit-animation-delay: 1s;  animation-delay: 1s;  -webkit-animation-fill-mode: both;  animation-fill-mode: both;  }  @-webkit-keyframes rotate {  0% {  background-color: orange;  -webkit-transform: rotate(0deg);  transform: rotate(0deg);  }  100% {  -webkit-transform: rotate(90deg);  transform: rotate(90deg);  background-color: #0099CC;  }  }  @keyframes rotate {  0% {  background-color: orange;  -webkit-transform: rotate(0deg);  transform: rotate(0deg);  }  100% {  -webkit-transform: rotate(90deg);  transform: rotate(90deg);  background-color: #0099CC;  }  }  </style>  </head>  <body>  <div class="container">  <p>  Hover over the container to apply the animations to the elements.  </p>  <div class="module">  <p><code>none</code></p>  <div class="wrapper">  <div class="element element-1"></div>  </div>  </div>  <div class="module">  <p><code>forwards</code></p>  <div class="wrapper">  <div class="element element-2"></div>  </div>  </div>  <div class="module">  <p><code>backwards</code></p>  <div class="wrapper">  <div class="element element-3"></div>  </div>  </div>  <div class="module">  <p><code>both</code></p>  <div class="wrapper">  <div class="element element-4"></div>  </div>  </div>  </div>  </body>  </html> |



|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title></title>  <style type="text/css">  @import url(http://fonts.googleapis.com/css?family=Lato:300);  .container{  text-align: center;  background-color: #e74c3c;  overflow: hidden;  }  .box:nth-child(2n-1){  background-color: rgba(0,0,0,0.05);  }  .box{  display: inline-block;  height: 200px;  width: 33.3%;  float:left;  position: relative;  /\*margin:0 -4px -5px -2px;\*/  transition: all .2s ease;  }  .box p{  color: #777;  font-family: Lato,"Helvetica Neue" ;  font-weight: 300;  position: absolute;  font-size: 20px;  width: 100%;  height: 25px;  text-align: center;  bottom: 0px;  margin: 0;  top:160px;  background-color: #fff;  opacity: 0;  text-transform: uppercase;  transition: all .2s ease;  }  .box:hover p{  bottom:0px;  top:175px;  opacity: 1;  transition: all .2s ease;  z-index: 2;  }  /\* MEDIA QUERIES \*/  @media (max-width: 700px){  .box{  width: 50%;  }  .box:nth-child(2n-1){  background-color: inherit;  }  .box:nth-child(4n),.box:nth-child(4n-3) {  background-color: rgba(0,0,0,0.05);  }  }  @media (max-width: 420px){  .box{  width: 100%;  }  .box:nth-child(4n),.box:nth-child(4n-3){  background-color: inherit;  }  .box:nth-child(2n-1){  background-color:rgba(0,0,0,0.05);  }  }  /\* -------------- Clock -------------- \*/  .clock{  border-radius: 60px;  border: 3px solid #fff;  height: 80px;  width: 80px;  position: relative;  top: 28%;  top: -webkit-calc(50% - 43px);  top: calc(50% - 43px);  left: 35%;  left: -webkit-calc(50% - 43px);  left: calc(50% - 43px);  }  .clock:after{  content: "";  position: absolute;  background-color: #fff;  top:2px;  left: 48%;  height: 38px;  width: 4px;  border-radius: 5px;  -webkit-transform-origin: 50% 97%;  transform-origin: 50% 97%;  -webkit-animation: grdAiguille 2s linear infinite;  animation: grdAiguille 2s linear infinite;  }  @-webkit-keyframes grdAiguille{  0%{-webkit-transform:rotate(0deg);}  100%{-webkit-transform:rotate(360deg);}  }  @keyframes grdAiguille{  0%{transform:rotate(0deg);}  100%{transform:rotate(360deg);}  }  .clock:before{  content: "";  position: absolute;  background-color: #fff;  top:6px;  left: 48%;  height: 35px;  width: 4px;  border-radius: 5px;  -webkit-transform-origin: 50% 94%;  transform-origin: 50% 94%;  -webkit-animation: ptAiguille 12s linear infinite;  animation: ptAiguille 12s linear infinite;  }  @-webkit-keyframes ptAiguille{  0%{-webkit-transform:rotate(0deg);}  100%{-webkit-transform:rotate(360deg);}  }  @keyframes ptAiguille{  0%{transform:rotate(0deg);}  100%{transform:rotate(360deg);}  }  /\* -------------- Sablier -------------- \*/  .hourglass{  position: relative;  height: 80px;  width: 80px;  top: 28%;  top: -webkit-calc(50% - 43px);  top: calc(50% - 43px);  left: 35%;  left: -webkit-calc(50% - 43px);  left: calc(50% - 43px);  border: 3px solid #fff;  border-radius: 80px;  -webkit-transform-origin: 50% 50%;  transform-origin: 50% 50%;  -webkit-animation: hourglass 2s ease-in-out infinite;  animation: hourglass 2s ease-in-out infinite;  }  .hourglass:before{  content: "";  position: absolute;  top:8px;  left: 18px;  width: 0px;  height: 0px;  border-style: solid;  border-width: 37px 22px 0 22px;  border-color: #fff transparent transparent transparent;  }  .hourglass:after{  content: "";  position: absolute;  top: 35px;  left: 18px;  width: 0px;  height: 0px;  border-style: solid;  border-width: 0 22px 37px 22px;  border-color: transparent transparent #fff transparent;  }  @-webkit-keyframes hourglass{  0%{-webkit-transform:rotate(0deg);}  100%{-webkit-transform:rotate(180deg);}  }  @keyframes hourglass{  0%{transform:rotate(0deg);}  100%{transform:rotate(180deg);}  }  /\* -------------- Loader1 -------------- \*/  .loader1{  position: relative;  height: 80px;  width: 80px;  border-radius: 80px;  border: 3px solid rgba(255,255,255, .7);  top: 28%;  top: -webkit-calc(50% - 43px);  top: calc(50% - 43px);  left: 35%;  left: -webkit-calc(50% - 43px);  left: calc(50% - 43px);  -webkit-transform-origin: 50% 50%;  transform-origin: 50% 50%;  -webkit-animation: loader1 3s linear infinite;  animation: loader1 3s linear infinite;  }  .loader1:after{  content: "";  position: absolute;  top: -5px;  left: 20px;  width: 11px;  height: 11px;  border-radius: 10px;  background-color: #fff;  }  @-webkit-keyframes loader1{  0%{-webkit-transform:rotate(0deg);}  100%{-webkit-transform:rotate(360deg);}  }  @keyframes loader1{  0%{transform:rotate(0deg);}  100%{transform:rotate(360deg);}  }  /\* -------------- loader2 -------------- \*/  .loader2{  position: relative;  width: 5px;  height: 5px;  top: 49%;  top: -webkit-calc(50% - 43px);  top: calc(50% - 2.5px);  left: 49%;  left: -webkit-calc(50% - 43px);  left: calc(50% - 2.5px);  background-color: #fff;  border-radius: 50px;  }  .loader2:before{  content: "";  position: absolute;  top: -38px;  border-top: 3px solid #fff;  border-right: 3px solid #fff;  border-radius: 0 50px 0px 0;  width: 40px;  height: 40px;  background-color: rgba(255, 255, 255, .1);  -webkit-transform-origin: 0% 100%;  transform-origin: 0% 100% ;  -webkit-animation: loader2 1.5s linear infinite;  animation: loader2 1.5s linear infinite;  }  .loader2:after{  content: "";  position: absolute;  top: 2px;  right: 2px;  border-bottom: 3px solid #fff;  border-left: 3px solid #fff;  border-radius: 0 0px 0px 50px;  width: 40px;  height: 40px;  background-color: rgba(255, 255, 255, .1);  -webkit-transform-origin: 100% 0%;  transform-origin: 100% 0% ;  -webkit-animation: loader2 1.5s linear infinite;  animation: loader2 1.5s linear infinite;  }  @-webkit-keyframes loader2{  0%{-webkit-transform:rotate(0deg);}  100%{-webkit-transform:rotate(360deg);}  }  @keyframes loader2{  0%{transform:rotate(0deg);}  100%{transform:rotate(360deg);}  }  /\* -------------- loader3 -------------- \*/  .loader3{  position: relative;  width: 150px;  height: 20px;  top: 45%;  top: -webkit-calc(50% - 10px);  top: calc(50% - 10px);  left: 25%;  left: -webkit-calc(50% - 75px);  left: calc(50% - 75px);  }  .loader3:after{  content: "LOADING ...";  color: #fff;  font-family: Lato,"Helvetica Neue" ;  font-weight: 200;  font-size: 16px;  position: absolute;  width: 100%;  height: 20px;  line-height: 20px;  left: 0;  top: 0;  background-color: #e74c3c;  z-index: 1;  }  .loader3:before{  content: "";  position: absolute;  background-color: #fff;  top: -5px;  left: 0px;  height: 30px;  width: 0px;  z-index: 0;  opacity: 1;  -webkit-transform-origin: 100% 0%;  transform-origin: 100% 0% ;  -webkit-animation: loader3 10s ease-in-out infinite;  animation: loader3 10s ease-in-out infinite;  }  @-webkit-keyframes loader3{  0%{width: 0px;}  70%{width: 100%; opacity: 1;}  90%{opacity: 0; width: 100%;}  100%{opacity: 0;width: 0px;}  }  @keyframes loader3{  0%{width: 0px;}  70%{width: 100%; opacity: 1;}  90%{opacity: 0; width: 100%;}  100%{opacity: 0;width: 0px;}  }  /\* -------------- loader4 -------------- \*/  .loader4{  position: relative;  width: 150px;  height: 20px;  top: 45%;  top: -webkit-calc(50% - 10px);  top: calc(50% - 10px);  left: 25%;  left: -webkit-calc(50% - 75px);  left: calc(50% - 75px);  background-color: rgba(255,255,255,0.2);  }  .loader4:before{  content: "";  position: absolute;  background-color: #fff;  top: 0px;  left: 0px;  height: 20px;  width: 0px;  z-index: 0;  opacity: 1;  -webkit-transform-origin: 100% 0%;  transform-origin: 100% 0% ;  -webkit-animation: loader4 10s ease-in-out infinite;  animation: loader4 10s ease-in-out infinite;  }  .loader4:after{  content: "LOADING ...";  color: #fff;  font-family: Lato,"Helvetica Neue" ;  font-weight: 200;  font-size: 16px;  position: absolute;  width: 100%;  height: 20px;  line-height: 20px;  left: 0;  top: 0;  }  @-webkit-keyframes loader4{  0%{width: 0px;}  70%{width: 100%; opacity: 1;}  90%{opacity: 0; width: 100%;}  100%{opacity: 0;width: 0px;}  }  @keyframes loader4{  0%{width: 0px;}  70%{width: 100%; opacity: 1;}  90%{opacity: 0; width: 100%;}  100%{opacity: 0;width: 0px;}  }  /\* -------------- loader5 -------------- \*/  .loader5{  position: relative;  width: 150px;  height: 20px;  top: 45%;  top: -webkit-calc(50% - 10px);  top: calc(50% - 10px);  left: 25%;  left: -webkit-calc(50% - 75px);  left: calc(50% - 75px);  background-color: rgba(255,255,255,0.2);  }  .loader5:after{  content: "LOADING ...";  color: #fff;  font-family: Lato,"Helvetica Neue" ;  font-weight: 200;  font-size: 16px;  position: absolute;  width: 100%;  height: 20px;  line-height: 20px;  left: 0;  top: 0;  }  .loader5:before{  content: "";  position: absolute;  background-color: #fff;  top: 0px;  height: 20px;  width: 0px;  z-index: 0;  -webkit-transform-origin: 100% 0%;  transform-origin: 100% 0% ;  -webkit-animation: loader5 7s ease-in-out infinite;  animation: loader5 7s ease-in-out infinite;  }  @-webkit-keyframes loader5{  0%{width: 0px; left: 0px}  48%{width: 100%; left: 0px}  50%{width: 100%; right: 0px}  52%{width: 100%; right: 0px}  100%{width: 0px; right: 0px}  }  @keyframes loader5{  0%{width: 0px; left: 0px}  48%{width: 100%; left: 0px}  50%{width: 100%; right: 0px}  52%{width: 100%; right: 0px}  100%{width: 0px; right: 0px}  }  /\* -------------- loader6 -------------- \*/  .loader6{  position: relative;  width: 12px;  height: 12px;  top: 46%;  top: -webkit-calc(50% - 6px);  top: calc(50% - 6px);  left: 46%;  left: -webkit-calc(50% - 6px);  left: calc(50% - 6px);    border-radius: 12px;  background-color: #fff;  -webkit-transform-origin: 50% 50%;  transform-origin: 50% 50% ;  -webkit-animation: loader6 1s ease-in-out infinite;  animation: loader6 1s ease-in-out infinite;  }  .loader6:before{  content: "";  position: absolute;  background-color: rgba(255, 255, 255, .5);  top: 0px;  left: -25px;  height: 12px;  width: 12px;  border-radius: 12px;  }  .loader6:after{  content: "";  position: absolute;  background-color: rgba(255, 255 ,255 ,.5);  top: 0px;  left: 25px;  height: 12px;  width: 12px;  border-radius: 12px;  }  @-webkit-keyframes loader6{  0%{-webkit-transform:rotate(0deg);}  50%{-webkit-transform:rotate(180deg);}  100%{-webkit-transform:rotate(180deg);}  }  @keyframes loader6{  0%{transform:rotate(0deg);}  50%{transform:rotate(180deg);}  100%{transform:rotate(180deg);}  }  /\* -------------- loader7 -------------- \*/  .loader7{  position: relative;  width: 40px;  height: 40px;  top: 40%;  top: -webkit-calc(50% - 20px);  top: calc(50% - 20px);  left: 43%;  left: -webkit-calc(50% - 20px);  left: calc(50% - 20px);  background-color: rgba(255, 255, 255, .2);  }  .loader7:before{  content: "";  position: absolute;  background-color: #fff;  height: 10px;  width: 10px;  border-radius: 10px;  -webkit-animation: loader7 2s ease-in-out infinite;  animation: loader7 2s ease-in-out infinite;  }  .loader7:after{  content: "";  position: absolute;  background-color: #fff;  top: 0px;  left: 0px;  height: 40px;  width: 0px;  z-index: 0;  opacity: 1;  -webkit-animation: loader72 10s ease-in-out infinite;  animation: loader72 10s ease-in-out infinite;  }  @-webkit-keyframes loader7{  0%{left: -12px; top: -12px;}  25%{left:42px; top:-12px;}  50%{left: 42px; top: 42px;}  75%{left: -12px; top: 42px;}  100%{left:-12px; top:-12px;}  }  @keyframes loader7{  0%{left: -12px; top: -12px;}  25%{left:42px; top:-12px;}  50%{left: 42px; top: 42px;}  75%{left: -12px; top: 42px;}  100%{left:-12px; top:-12px;}  }  @-webkit-keyframes loader72{  0%{width: 0px;}  70%{width: 40px; opacity: 1;}  90%{opacity: 0; width: 40px;}  100%{opacity: 0;width: 0px;}  }  @keyframes loader72{  0%{width: 0px;}  70%{width: 40px; opacity: 1;}  90%{opacity: 0; width: 40px;}  100%{opacity: 0;width: 0px;}  }  /\* -------------- loader8 -------------- \*/  .loader8{  position: relative;  width: 80px;  height: 80px;  top: 28%;  top: -webkit-calc(50% - 43px);  top: calc(50% - 43px);  left: 35%;  left: -webkit-calc(50% - 43px);  left: calc(50% - 43px);  border-radius: 50px;  background-color: rgba(255, 255, 255, .2);  border-width: 40px;  border-style: double;  border-color:transparent #fff;  -webkit-box-sizing:border-box;  -moz-box-sizing:border-box;  box-sizing:border-box;  -webkit-transform-origin: 50% 50%;  transform-origin: 50% 50% ;  -webkit-animation: loader8 2s linear infinite;  animation: loader8 2s linear infinite;    }  @-webkit-keyframes loader8{  0%{-webkit-transform:rotate(0deg);}  100%{-webkit-transform:rotate(360deg);}  }  @keyframes loader8{  0%{transform:rotate(0deg);}  100%{transform:rotate(360deg);}  }  /\* -------------- loader9 -------------- \*/  .loader9:before{  content: "";  position: absolute;  top: 0px;  height: 12px;  width: 12px;  border-radius: 12px;  -webkit-animation: loader9g 3s ease-in-out infinite;  animation: loader9g 3s ease-in-out infinite;  }  .loader9{  position: relative;  width: 12px;  height: 12px;  top: 46%;  left: 46%;  border-radius: 12px;  background-color: #fff;  }  .loader9:after{  content: "";  position: absolute;  top: 0px;  height: 12px;  width: 12px;  border-radius: 12px;  -webkit-animation: loader9d 3s ease-in-out infinite;  animation: loader9d 3s ease-in-out infinite;  }  @-webkit-keyframes loader9g{  0%{ left: -25px; background-color: rgba(255, 255, 255, .8); }  50%{ left: 0px; background-color: rgba(255, 255, 255, .1);}  100%{ left:-25px; background-color: rgba(255, 255, 255, .8); }  }  @keyframes loader9g{  0%{ left: -25px; background-color: rgba(255, 255, 255, .8); }  50%{ left: 0px; background-color: rgba(255, 255, 255, .1);}  100%{ left:-25px; background-color: rgba(255, 255, 255, .8); }  }  @-webkit-keyframes loader9d{  0%{ left: 25px; background-color: rgba(255, 255, 255, .8); }  50%{ left: 0px; background-color: rgba(255, 255, 255, .1);}  100%{ left:25px; background-color: rgba(255, 255, 255, .8); }  }  @keyframes loader9d{  0%{ left: 25px; background-color: rgba(255, 255, 255, .8); }  50%{ left: 0px; background-color: rgba(255, 255, 255, .1);}  100%{ left:25px; background-color: rgba(255, 255, 255, .8); }  }  /\* -------------- loader10 -------------- \*/  .loader10:before{  content: "";  position: absolute;  top: 0px;  left: -25px;  height: 12px;  width: 12px;  border-radius: 12px;  -webkit-animation: loader10g 3s ease-in-out infinite;  animation: loader10g 3s ease-in-out infinite;  }  .loader10{  position: relative;  width: 12px;  height: 12px;  top: 46%;  left: 46%;  border-radius: 12px;  -webkit-animation: loader10m 3s ease-in-out infinite;  animation: loader10m 3s ease-in-out infinite;  }  .loader10:after{  content: "";  position: absolute;  top: 0px;  left: 25px;  height: 10px;  width: 10px;  border-radius: 10px;  -webkit-animation: loader10d 3s ease-in-out infinite;  animation: loader10d 3s ease-in-out infinite;  }  @-webkit-keyframes loader10g{  0%{background-color: rgba(255, 255, 255, .2);}  25%{background-color: rgba(255, 255, 255, 1);}  50%{background-color: rgba(255, 255, 255, .2);}  75%{background-color: rgba(255, 255, 255, .2);}  100%{background-color: rgba(255, 255, 255, .2);}  }  @keyframes loader10g{  0%{background-color: rgba(255, 255, 255, .2);}  25%{background-color: rgba(255, 255, 255, 1);}  50%{background-color: rgba(255, 255, 255, .2);}  75%{background-color: rgba(255, 255, 255, .2);}  100%{background-color: rgba(255, 255, 255, .2);}  }  @-webkit-keyframes loader10m{  0%{background-color: rgba(255, 255, 255, .2);}  25%{background-color: rgba(255, 255, 255, .2);}  50%{background-color: rgba(255, 255, 255, 1);}  75%{background-color: rgba(255, 255, 255, .2);}  100%{background-color: rgba(255, 255, 255, .2);}  }  @keyframes loader10m{  0%{background-color: rgba(255, 255, 255, .2);}  25%{background-color: rgba(255, 255, 255, .2);}  50%{background-color: rgba(255, 255, 255, 1);}  75%{background-color: rgba(255, 255, 255, .2);}  100%{background-color: rgba(255, 255, 255, .2);}  }  @-webkit-keyframes loader10d{  0%{background-color: rgba(255, 255, 255, .2);}  25%{background-color: rgba(255, 255, 255, .2);}  50%{background-color: rgba(255, 255, 255, .2);}  75%{background-color: rgba(255, 255, 255, 1);}  100%{background-color: rgba(255, 255, 255, .2);}  }  @keyframes loader10d{  0%{background-color: rgba(255, 255, 255, .2);}  25%{background-color: rgba(255, 255, 255, .2);}  50%{background-color: rgba(255, 255, 255, .2);}  75%{background-color: rgba(255, 255, 255, 1);}  100%{background-color: rgba(255, 255, 255, .2);}  }  </style>  </head>  <body>  <div class="container">  <div class="box">  <div class="clock"></div>  <p>Clock</p>  </div>  <div class="box">  <div class="hourglass"></div>  <p>Hourglass</p>  </div>  <div class="box">  <div class="loader1"></div>  <p>Loader 1</p>  </div>  <div class="box">  <div class="loader2"></div>  <p>loader 2</p>  </div>  <div class="box">  <div class="loader3"></div>  <p>loader 3</p>  </div>  <div class="box">  <div class="loader4"></div>  <p>loader 4</p>  </div>  <div class="box">  <div class="loader5"></div>  <p>loader 5</p>  </div>  <div class="box">  <div class="loader6"></div>  <p>loader 6</p>  </div>  <div class="box">  <div class="loader7"></div>  <p>loader 7</p>  </div>  <div class="box">  <div class="loader8"></div>  <p>loader 8</p>  </div>  <div class="box">  <div class="loader9"></div>  <p>loader 9</p>  </div>  <div class="box">  <div class="loader10"></div>  <p>loader 10 </p>  </div>  </div>  </body>  </html> |

 원형그래프만들기 더조은폴더 > 0502 폴더에 있음



|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title></title>  <style type="text/css">  .pie {  position:absolute;  width:100px;  height:200px;  overflow:hidden;  left:150px;  -moz-transform-origin:left center;  -ms-transform-origin:left center;  -o-transform-origin:left center;  -webkit-transform-origin:left center;  transform-origin:left center;  }  /\*  unless the piece represents more than 50% of the whole chart.  then make it a square, and ensure the transform origin is  back in the center.  NOTE: since this is only ever a single piece, you could  move this to a piece specific rule and remove the extra class  \*/  .pie.big {  width:200px;  height:200px;  left:50px;  -moz-transform-origin:center center;  -ms-transform-origin:center center;  -o-transform-origin:center center;  -webkit-transform-origin:center center;  transform-origin:center center;  }  /\*  this is the actual visible part of the pie.  Give it the same dimensions as the regular piece.  Use border radius make it a half circle.  move transform origin to the middle of the right side.  Push it out to the left of the containing box.  \*/  .pie:BEFORE {  content:"";  position:absolute;  width:100px;  height:200px;  left:-100px;  border-radius:100px 0 0 100px;  transform-origin:right center;    }  /\* if it's part of a big piece, bring it back into the square \*/  .pie.big:BEFORE {  left:0px;  }  /\*  big pieces will also need a second semicircle, pointed in the  opposite direction to hide the first part behind.  \*/  .pie.big:AFTER {  content:"";  position:absolute;  width:100px;  height:200px;  left:100px;  border-radius:0 100px 100px 0;  }  /\*  add colour to each piece.  \*/  .pie:nth-of-type(1):BEFORE,  .pie:nth-of-type(1):AFTER {  background-color:blue;  }  .pie:nth-of-type(2):AFTER,  .pie:nth-of-type(2):BEFORE {  background-color:green;  }  .pie:nth-of-type(3):AFTER,  .pie:nth-of-type(3):BEFORE {  background-color:red;  }  .pie:nth-of-type(4):AFTER,  .pie:nth-of-type(4):BEFORE {  background-color:orange;  }  .pie[data-start="30"] {  transform:rotate(30deg);  }  .pie[data-start="60"] {  transform:rotate(60deg);  }  .pie[data-start="100"] {  transform:rotate(100deg);  }  .pie[data-value="30"]:BEFORE {  transform:rotate(31deg);  }  .pie[data-value="40"]:BEFORE {  transform:rotate(41deg);  }  .pie[data-value="260"]:BEFORE {  transform:rotate(260deg);  }  </style>  </head>  <body>  <div class="pie" data-start="0" data-value="30"></div>  <div class="pie highlight" data-start="30" data-value="30"></div>  <div class="pie" data-start="60" data-value="40"></div>  <div class="pie big" data-start="100" data-value="260"></div>  </body>  </html> |