# 트랜지션(transition)

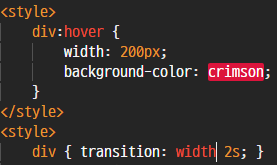
## 실습



공통 css 내용.

처음 상태의 div

### 기본 transition(전환)

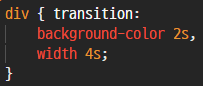


* 처음 상태와 마지막 상태를 지정했을 때, 중간 과정을 자동으로 생성해주는 속성.
* **주의사항:** 트랜지션은 반드시 **처음 상태에 적용**해야 한다.
  + 나중 상태에 트랜지션을 주면, 반쪽자리 모션이 된다.

(진행은 문제 없으나 되돌아올때 애니메이션이 사라짐.)

* 트랜지션의 기본 문법은 다음과 같다.
  + **{ transition: 변경할속성명 시간; }**
  + 어떤 속성(property)을 몇 초(duration) 동안 변경할건지 적는 것.
  + 이 때, 두 값의 순서는 상관이 없다.

### 병기 표기

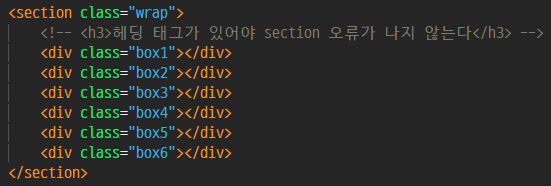
 두 개 이상을 한꺼번에 표기 가능. (쉼표 표기)

### all

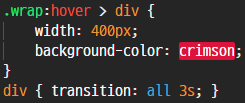


* all 은 ***변경 가능한* 모든 속성**을 뜻한다.

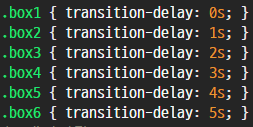
### 대기(delay)



#### section 은 헤딩 태그가 있어야 오류가 나지 않는다.

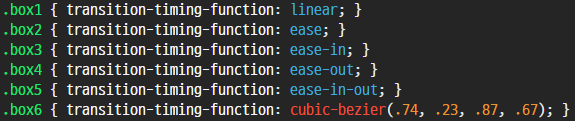


wrap 영역에 마우스 커서가 올라가면 div 속성이 바뀐다.



* delay는 **대기시간**을 뜻한다.
  + { transition-delay } 0.5s = 0.5초 = 500ms
  + 1s = 1000ms

### 중간속도 (가속도)



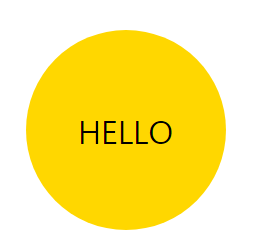
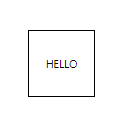
* linear 등속운동: 일정한 시간동안 일정한 값이 변하는 것
* ease 부드러운 효과
* ease - in 부드럽게 시작
* ease - out 부드럽게 마무리
* ease -in -out 부드럽게 시작해서 빠르게 가다가 부드럽게 마무리.
* cubic-bezier 사용자 정의 속성: 직접 가속도 효과를 만들 때 사용.
  + <https://cubic-bezier.com/>

### 통합 지정 (단축표기법)

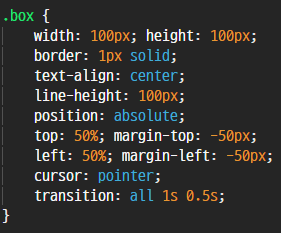


* 속성값은 순서없이 표기 가능하지만, 진행시간과 대기시간은 단위가 같으므로 먼저 작성된 내용이 진행시간, 나중에 작성한 내용이 대기시간이 된다.
* 따라서 해당 속성의 뜻은 3초동안 진행하고, 1초동안 대기하라는 뜻이 된다.

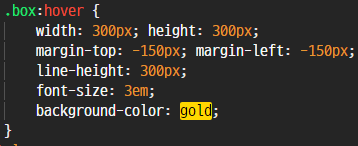
### 활용-1



* ghost button : 배경색이 투명하고 선두께가 얇은 형태의 버튼.

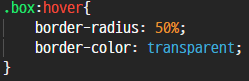
 원래의 상태

* position: absolute
*  relative<
*  html과 body의 높이를 브라우저 만큼 주기위해 height를 100%로 주었다. 주지 않으면 높이는 자식요소를 따라간다.
*  가운데 배치 공식

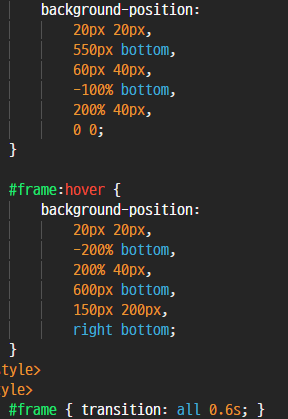


원래 요소가 margin으로 위치가 잡혀있기 때문에 그 자리에서 바뀌려면 여기서도 margin을 잡아줘야 한다. 글자도 가운데 두기 위해 line-height 와 글자 크기도 조정해주었다.

* 다만 다른 방법이 있다! { transform }

 원모양으로, 테두리는 투명하게.

### 활용-2



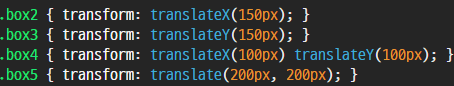
위치를 잡아놓고 전환효과를 준다. 우리나라에선 빠르게 주는 것이 좋다.

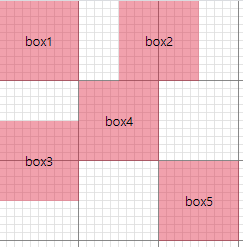
인터랙티브 웹페이지: 사용자의 반응을 이끌어내는 페이지.

## 트랜스폼(transform) 변형

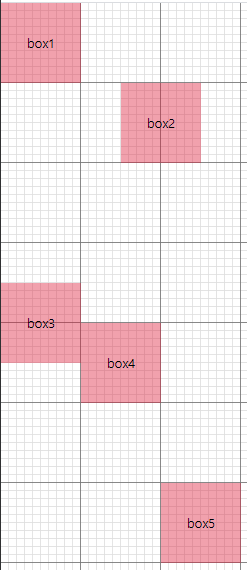
### 이동 (translate X/Y)





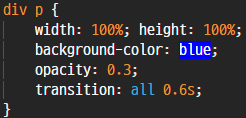
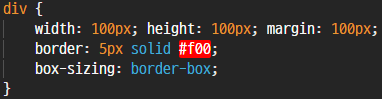


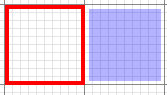


 relative는 원래있던 자리에서 움직이기 때문에 원래자리를 신경쓸것!

* position과 관계없이 적용된다.

### 이동2

 div의 하위 요소 p 태그 속성: div의 100%



p태그의 크기는 border크기를 뺀 95px이다.

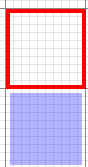


x축 (좌측>우측)으로 100px 가라.

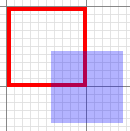
> 슬라이드 이미지의 아주 기본 형태

border은 프레임! overflow 속성으로 프레임을 벗어난 이미지는 숨긴다.



= 결과:  y축(상 > 하) 으로 100px



 대각선으로.

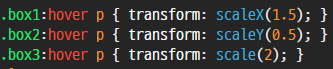
#### hover를 .box(부모요소)에 준 이유:

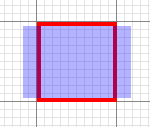
p태그에 hover를 줄 경우, 움직이면 호버영역에서 벗어나서 애니메이션이 끊길 수 있기 때문에, 안전하게 .box에 속성을 주는 것.

#### ★ css에서 띄어쓰기 주의

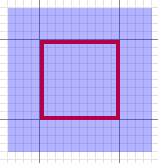
: .box1의 자손중에 :hover 를 가진 속성의 p태그를 찾는게 됨!

## 확대/축소 (scale)



.box1 : 

* 왜 양쪽 대칭으로 늘어날까?
* **transform의 특징: 기준좌표가 원점이기 때문.**

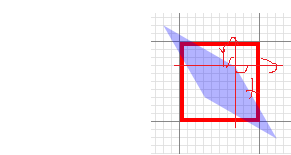
.box3 :  > 상품이미지에서 확대에 자주 쓰이는 방법

### 기울임



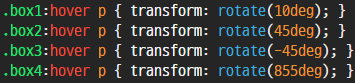
 프레임과 면사이의 흰 부분 이 20도라는 소리.

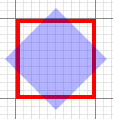




* 세로 y축. 가로 x축
* 기울임은 원본이 많이 변형되서 잘 쓰진 않음.

### 회전(rotate)

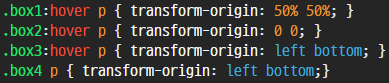
 + 시계방향 -반시계방향으로 돈다.

.box2 : 



* 빠르게 도는 걸 제어하는 방법: 시간 (0.6s) , 거리(각도크기:deg) 등.

### 회전중심 (-origin)



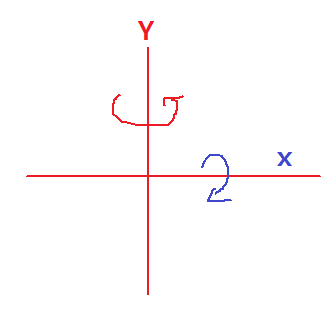
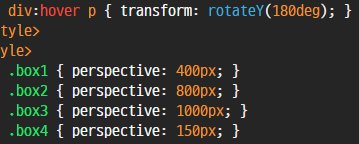
.box1 { transform-origin : 50% 50%; } = 기본값 (x축/y축)

.box4만 움직임이 자연스럽다.

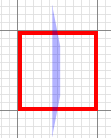
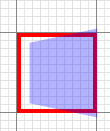
* transform-origin 은 움직이기 전에 속성값을 줘야한다.
* 모션이 부자연스러울 때는 속성값의 위치가 어디인지 확인해야 한다.

이거 헷갈리는디요

### 원근감 (perspective)

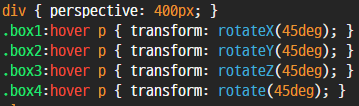


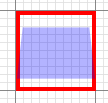
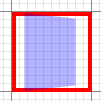
Y축을 기준으로 돈다는 것.

{ perspective: 100px } : 입체감이 더 크다.

=> **소실점의 거리가 가까울수록 입체감이 커진다**

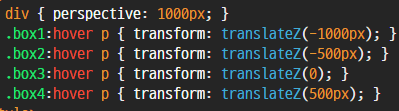
### 3D회전 (원근감)

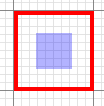
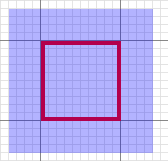


.box1 :  .box2 : 

rotateZ = rotate 동일하다. 때문에 거의 안쓴다.

### Z축이동



.box1 :  .box4: 

* **scale과 효과가 거의 동일하다!**

#### 과제: 멀티백그라운드를 이용한 배너광고 만들기

* 다중 배경 이미지와 트랜지션, 트랜스폼을 이용하여 인터랙티브 배너를 제작하시오!
* 1. 소스 이미지는 [images] 폴더에 정리한다.
* 2. CSS는 [css] 폴더에 'banner.css' 파일명으로 저장한다.
* 3. HTML은 'banner.html' 파일명으로 저장한다.
* 제출형식 : 이름.zip
* 제출마감 : 6/11



url의 경로를 쓸때,

 css 폴더에서 images 폴더로 들어가야 하므로

../ (상단의 부모 디렉터리를 뜻한다.)