

HTML 표준화 문서를 기반으로 하는 지침서



속이 깊은

HTML5 & CSS3

김명진 지음

09강

CSS3 Part-6

고급 기능

(다단, 변형, 트랜지션, 마키, 애니메이션)

학습 목표

CSS3에서는 새롭고 유용한 고급 기능들이 많이 추가되었다. 인쇄물에서의 뉴스 및 잡지의 경우와 같이 텍스트를 여러 단으로 표시할 수 있도록 하였고, 웹 요소를 옮기거나 회전시키는 등의 변형 기능을 추가하였다. 또한 특정 스타일에서 다른 스타일로 변경할 수 있는 트랜지션 기능과 콘텐츠를 동적으로 표현할 수 있는 마키 및 애니메이션 기능이 추가되었다. 그리고 요즘 인기와 관심이 많아지고 있는 반응형 웹을 만들 수 있는 미디어 쿼리의 많은 기능들이 추가되었다. 이번 장에서는 이렇게 새롭고 유용한 고급 기능들에 대하여 살펴보도록 하겠다.

Section

1

다단(Multi-column) 레이아웃

2

변형(Transform)

3

트랜지션(Transition)

4

콘텐츠 움직임(Marquee)

5

애니메이션(Animation)

6

미디어 쿼리(Media Query)



표준화 문서	CSS Multi-column Layout Module - http://www.w3.org/TR/css3-multicol/
표준화 단계	W3C Candidate Recommendation, (2011-04-12)

다단 문서의 개수 및 폭 지정하기

- **column-count** 속성은 다단을 구성할 때 단의 개수를 지정
- **column-width** 속성은 단의 폭을 지정
- **columns** 속성은 단의 개수와 폭을 한번에 지정하기 위한 대표 속성

속성	column-width	column-count	CSS3			
속성 값	<length> auto					
속성 특징	초기값	auto	적용	대체되지 않은 블록-레벨 요소들, 테이블 셀 및 인라인-블록 요소들 제외(테이블 요소들)	상속	X
JavaScript	object.style.columnWidth = "200px"		object.style.columnCount = "3"			
사용 예제	div { column-width: 200px }		div { column-count: 3 }			

속성	columns					CSS3
속성 값	<'column-width'> <'column-count'>					
속성 특징	초기값	개별 속성 참조	적용	대체되지 않은 블록-레벨 요소들, 테이블 셀 및 인라인-블록 요소들 제외(테이블 요소들)	상속	X
JavaScript	object.style.columns = "200px 3"					
사용 예제	div { columns: 200px 3 }					



다단 문서의 개수 및 폭 지정하기



속성

column-width, column-count, columns

CSS3

속성 값	설명
<i>auto</i>	기본 값으로, 단의 개수 및 폭의 크기를 자동으로 계산하도록 지정한다. column-width 속성 값을 auto로 지정한 후에 다단을 구성하려면 column-count 속성을 지정해야 하고, column-count 속성 값을 auto로 지정한 후에 다단을 구성하려면 column-width 속성을 지정해야 한다.
<i>length</i>	단의 개수 및 폭의 크기를 길이 값으로 지정할 수 있다. <ul style="list-style-type: none"> ✓ 단의 폭을 지정하게 되면 단의 개수는 요소의 폭에 따라 자동으로 계산된다. ✓ 단의 개수가 지정되면 단의 폭은 요소의 폭에 따라 자동으로 계산되기 때문에 요소의 폭이 증가하더라도 단의 개수는 변하지 않고 열의 너비만 변경된다.

예제

CSS3_09-01_ColumnWidth.html

Cascading Style Sheet Level 3

HTML이 HTML4.01에서 HTML5로 업그레이드가 되는 동안 CSS도 CSS2에서 CSS3로 업그레이드가 되면서 다양한 속성들이 추가되었다. 요즘은 모바일 기기 및 태블릿 PC 등의 다양한 디바이스들이 출시되고 있고 관련 산업이 발전되면서 이러한 디바이스들에 초점이 옮겨지고

있는 상황이다. 이러한 각종 디바이스들에 적응적으로 레이아웃을 입히기 위해서는 반드시 스타일시트의 도움을 받아야 하는 것은 누구나 알고 있는 사실이다. 특히, 기존 HTML4.01에서는 HTML 문서의 디자인을 위한 몇몇 요소들이 존재하였지만, HTML5에서는 이러한 요소들이 대부분 제거되었기 때문에 CSS3의 중요성이 높아졌다고 할 수 있다. CSS Level 2까지는 하나의 문서에 모든 기능들에 대한 내용들을 정의하였지만, CSS Level 3부터

는 기능에 따라 모듈로 분리하여 정의하고 있다. 이는 웹의 기능이 발전함에 따라 HTML도 진화를 거듭하여 HTML5로의 표준화를 앞세우고 있으며, 이를 기점으로 동적 애플리케이션 및 웹을 표현하는 디바이스들이 다양해짐에 따라서 웹의 표현도 더 다양해질 필요성이 확대되었다. 이로 인하여 웹의 표현 범위를 넓히기 위해 CSS의 기능들이 모듈별로 개발되고 있으며, 그 양도 CSS2.1과는 비교할 수 없을 만큼 방대함을 확인할 수 있다.

Cascading Style Sheet Level 3

HTML이 HTML4.01에서 HTML5로 업그레이드가 되는 동안 CSS도 CSS2에서 CSS3로 업그레이드가 되면서 다양한 속성들이 추가되었다. 요즘은 모바일 기기 및 태블릿 PC 등의 다양한 디바이스들이 출시되고 있고 관련 산업이 발전되면서 이러한 디바이스들에 초점이 옮겨지고 있는 상황이다. 이러한 각종 디바이스들에 적응적으로 레이아웃을 입히기 위해서는 반드시 스타일시트의 도움을 받아야 하는 것은 누구나 알고 있는 사실이다. 특히, 기존 HTML4.01에서는 HTML 문서의 디자인을 위

한 몇몇 요소들이 존재하였지만, HTML5에서는 이러한 요소들이 대부분 제거되었기 때문에 CSS3의 중요성이 높아졌다고 할 수 있다. CSS Level 2까지는 하나의 문서에 모든 기능들에 대한 내용들을 정의하였지만, CSS Level 3부터는 기능에 따라 모듈로 분리하여 정의하고 있다. 이는 웹의 기능이 발전함에 따라 HTML도 진화를 거듭하여 HTML5로의 표준화를 앞세우고 있으며, 이를 기점으로 동적 애플리케이션 및 웹을 표현하는 디바이스들이 다양해짐에 따라서 웹의 표현도 더 다양해질 필요성이 확대되었다. 이로 인하여 웹의 표현 범위를 넓히기 위해 CSS의 기능들이 모듈별로 개발되고 있으며, 그 양도 CSS2.1과는 비교할 수 없을 만큼 방대함을 확인할 수 있다.

column-width: 100px;

column-width: 200px;



다단 문서의 개수 및 폭 지정하기



속성

column-width, column-count, columns

CSS3

예제

CSS3_09-01_ColumnCount.html

Cascading Style Sheet Level 3

HTML이 HTML4.01에서 HTML5로 업그레이드가 되는 동안 CSS도 CSS2에서 CSS3로 업그레이드가 되면서 다양한 속성들이 추가되었다. 요즘은 모바일 기기 및 태블릿 PC 등의 다양한 디바이스들이 출시되고 있고 관련 산업이 발전되면서 이러한 디바이스들에 초점이 옮겨지고 있는 상황이다. 이러한 각종 디바이스들에 적응적으로 레이아웃을 만들기 위해서는 반드시 스타일시트의 도움을 받아야 하는 것은 누구나 알고 있는 사실이다. 특히, 기존 HTML4.01에서는 HTML 문서의 디자인을 위한 몇몇 요소들이 존재하였지만, HTML5에서는 이러한 요소들이 대부분 제거되었기 때문에 CSS3의 중요성이 높아졌다고 할 수 있다. CSS Level 2까지는 하나의 문서에 모든 기능들에 대한 내용들을 정의하였지만, CSS Level 3부터는 기능에 따라 모듈로 분리하여 정의하고 있다. 이는 웹의 기능이 발전함에 따라 HTML도 진화를 거듭하여 HTML5로의 표준화를 앞게 두고 있으며, 이를 기점으로 동적 애플리케이션 및 웹을 표현하는 디바이스들이 다양해짐에 따라서 웹의 표현도 더 다양해질 필요성이 확대되었다. 이로 인하여 웹의 표현 범위를 넓히기 위해 CSS의 기능들이 모듈별로 개발되고 있으며, 그 양도 CSS2.1과는 비교할 수 없을 만큼 방대함을 확인할 수 있다.

Cascading Style Sheet Level 3

HTML이 HTML4.01에서 HTML5로 업그레이드가 되는 동안 CSS도 CSS2에서 CSS3로 업그레이드가 되면서 다양한 속성들이 추가되었다. 요즘은 모바일 기기 및 태블릿 PC 등의 다양한 디바이스들이 출시되고 있고 관련 산업이 발전되면서 이러한 디바이스들에 초점이 옮겨지고 있는 상황이다. 이러한 각종 디바이스들에 적응적으로 레이아웃을 만들기 위해서는 반드시 스타일시트의 도움을 받아야 하는 것은 누구나 알고 있는 사실이다. 특히, 기존 HTML4.01에서는 HTML 문

서의 디자인을 위한 몇몇 요소들이 존재하였지만, HTML5에서는 이러한 요소들이 대부분 제거되었기 때문에 CSS3의 중요성이 높아졌다고 할 수 있다. CSS Level 2까지는 하나의 문서에 모든 기능들에 대한 내용들을 정의하였지만, CSS Level 3부터는 기능에 따라 모듈로 분리하여 정의하고 있다. 이는 웹의 기능이 발전함에 따라 HTML도 진화를 거듭하여 HTML5로의 표준화를 앞게 두고 있으며, 이를 기점으로 동적 애플리케이션 및 웹을 표현하는 디바이스들이 다양해짐에 따라서 웹의 표현도 더 다양해질 필요성이 확대되었다. 이로 인하여 웹의 표현 범위를 넓히기 위해 CSS의 기능들이 모듈별로 개발되고 있으며, 그 양도 CSS2.1과는 비교할 수 없을 만큼 방대함을 확인할 수 있다.

column-count: 1

Cascading Style Sheet Level 3

HTML이 HTML4.01에서 HTML5로 업그레이드가 되는 동안 CSS도 CSS2에서 CSS3로 업그레이드가 되면서 다양한 속성들이 추가되었다. 요즘은 모바일 기기 및 태블릿 PC 등의 다양한 디바이스들이 출시되고 있고 관련 산업이 발전되면서 이러한 디바이스들에 초점이 옮겨지고 있는 상황이다. 이러한 각종 디바이스들에 적응적으로 레이아웃을 만들기 위해서는 반드시 스타일시트의 도움을 받아야 하는 것은 누구나 알고 있는 사실이다. 특히, 기존 HTML4.01에서는 HTML 문서의 디자인을 위한 몇몇 요소들이 존재하였지만, HTML5에서는 이러한 요소들이 대부분 제거되었기 때문에 CSS3의 중요성이 높아졌다고 할 수 있다. CSS Level 2까지는 하나의 문서에 모든 기능들에 대한 내용들을 정의하였지만, CSS Level 3부터는 기능에

거지고 있는 상황이다. 이러한 각종 디바이스들에 적응적으로 레이아웃을 만들기 위해서는 반드시 스타일시트의 도움을 받아야 하는 것은 누구나 알고 있는 사실이다. 특히, 기존 HTML4.01에서는 HTML 문서의 디자인을 위한 몇몇 요소들이 존재하였지만, HTML5에서는 이러한 요소들이 대부분 제거되었기 때문에 CSS3의 중요성이 높아졌다고 할 수 있다. CSS Level 2까지는 하나의 문서에 모든 기능들에 대한 내용들을 정의하였지만, CSS Level 3부터는 기능에

따라 모듈로 분리하여 정의하고 있다. 이는 웹의 기능이 발전함에 따라 HTML도 진화를 거듭하여 HTML5로의 표준화를 앞게 두고 있으며, 이를 기점으로 동적 애플리케이션 및 웹을 표현하는 디바이스들이 다양해짐에 따라서 웹의 표현도 더 다양해질 필요성이 확대되었다. 이로 인하여 웹의 표현 범위를 넓히기 위해 CSS의 기능들이 모듈별로 개발되고 있으며, 그 양도 CSS2.1과는 비교할 수 없을 만큼 방대함을 확인할 수 있다.

Cascading Style Sheet Level 3

HTML이 HTML4.01에서 HTML5로 업그레이드가 되는 동안 CSS도 CSS2에서 CSS3로 업그레이드가 되면서 다양한 속성들이 추가되었다. 요즘은 모바일 기기

및 태블릿 PC 등의 다양한 디바이스들이 출시되고 있고 관련 산업이 발전되면서 이러한 디바이스들에 초점이 옮겨지고 있는 상황이다. 이러한 각종 디바이스들에 적응적으로 레이아웃을 만들기 위해서는 반드시 스타일시트의 도움을 받아야 하는 것은 누구나 알고 있는 사실이다. 특히, 기존 HTML4.01에서는 HTML 문서의 디자인을 위한 몇몇 요

소들이 존재하였지만, HTML5에서는 이러한 요소들이 대부분 제거되었기 때문에 CSS3의 중요성이 높아졌다고 할 수 있다. CSS Level 2까지는 하나의 문서에 모든 기능들에 대한 내용들을 정의하였지만, CSS Level 3부터는 기능에 따라 모듈로 분리하여 정의하고 있다. 이는 웹의 기능이 발전함에 따라 HTML도 진화를 거듭하여 HTML5로

의 표준화를 앞게 두고 있으며, 이를 기점으로 동적 애플리케이션 및 웹을 표현하는 디바이스들이 다양해짐에 따라서 웹의 표현도 더 다양해질 필요성이 확대되었다. 이로 인하여 웹의 표현 범위를 넓히기 위해 CSS의 기능들이 모듈별로 개발되고 있으며, 그 양도 CSS2.1과는 비교할 수 없을 만큼 방대함을 확인할 수 있다.

column-count: 3

column-count: 4



열의 간격 및 구분선 스타일 지정



속성

column-gap – 단과 단 사이의 간격을 지정

CSS3

속성 값	<length> normal				
속성 특징	초기값	normal	적용	다단 요소들	상속
JavaScript	<code>object.style.columnGap = "50px"</code>				
사용 예제	<code>div { column-gap: 50px }</code>				

속성 값	설명				
<i>normal</i>	기본 값으로, 브라우저에 의해서 지정되는 값으로, 대한 자세한 값은 아직 결정되지 않았지만, 1em 값만이 제안된 상태이다.				
<i>length</i>	'길이+단위' 값을 이용하여 단 사이의 간격을 지정할 수 있으며, 음수 값은 지정할 수 없다.				



속성

column-rule-style – 단과 단 사이의 구분선 형태를 지정

CSS3

속성 값	<border-style>				
속성 특징	초기값	none	적용	다단 요소들	상속
JavaScript	<code>object.style.columnRuleStyle = "dotted"</code>				
사용 예제	<code>div { column-rule-style: dotted }</code>				



열의 간격 및 구분선 스타일 지정

✓ 속성	column-rule-width – 단과 단 사이의 구분선 굵기를 지정	CSS3
------	-----------------------------------------	------

속성 값	<border-width>				
속성 특징	초기값	medium	적용	다단 요소들	상속
JavaScript	<i>object.style.columnRuleWidth</i> = "tick"				
사용 예제	div { column-rule-width : thick }				

✓ 속성	column-rule – 단과 단 사이의 구분 형태를 지정	CSS3
------	----------------------------------	------

속성 값	<'column-rule-width'> <'column-rule-style'> [<'column-rule-color'> transparent]				
속성 특징	초기값	개별 속성 참조	적용	다단 요소들	상속
JavaScript	<i>object.style.columnRule</i> = "#f00 double 3px"				
사용 예제	div { column-rule : f00 double 3px }				



열의 간격 및 구분선 스타일 지정



속성

column-rule - 단과 단 사이의 구분 형태를 지정

CSS3

예제

CSS3_09-01_ColumnRule.html

Cascading Style Sheet Level 3

HTML01 HTML4.01에서 HTML5로 업그레이드가 되는 동안 CSS도 CSS2에서 CSS3로 업그레이드가 되면서 다양한 속성들이 추가되었다. 요즘은 모바일 기기 및 태블릿 PC등의 다양한 디바이스들이 출시되고 있고 관련 산업이 발전되면서 이러한 디바이스들에 초점이 옮겨지고 있는 상황이다. 이러한 각종 디바이스들에 적응적으로 레이아웃을 입히기 위해서는 반

드시 스타일시트의 도움을 받아야 하는 것은 누구나 알고 있는 사실이다. 특히, 기존 HTML4.01에서는 HTML 문서의 디자인을 위한 몇몇 요소들이 존재하였지만, HTML5에서는 이러한 요소들이 대부분 제거되었기 때문에 CSS3의 중요성이 높아졌다고 할 수 있다. CSS Level 2까지는 하나의 문서에 모든 기능들에 대한 내용들을 정의하였지만, CSS Level 3부터는 기능에 따라 모듈로 분리하여 정의하고 있다. 이는 웹의 기능이 발전함에 따라 HTML도 진화를 거듭하여 HTML5로의 표준화를 앞메 두고 있으며, 이를 기점으로 동적 애플리케이션 및 웹을

표현하는 디바이스들이 다양해짐에 따라서 웹의 표현도 더 다양해질 필요성이 확대되었다. 이로 인하여 웹의 표현 범위를 넓히기 위해 CSS의 기능들이 모듈별로 개발되고 있으며, 그 양도 CSS2.1과는 비교할 수 없을 만큼 방대함을 확인할 수 있다.

Cascading Style Sheet Level 3

HTML01 HTML4.01에서 HTML5로 업그레이드가 되는 동안 CSS도 CSS2에서 CSS3로 업그레이드가 되면서 다양한 속성들이 추가되었다. 요즘은 모바일 기기 및 태블릿 PC등의 다양한 디바이스들이 출시되고 있고 관련 산업이 발전되면서 이러한 디바이스들에 초점이 옮겨지고 있는 상황이다. 이러한 각종 디바이스들에 적응적으로 레이아웃을 입히기 위해서는 반

드시 스타일시트의 도움을 받아야 하는 것은 누구나 알고 있는 사실이다. 특히, 기존 HTML4.01에서는 HTML 문서의 디자인을 위한 몇몇 요소들이 존재하였지만, HTML5에서는 이러한 요소들이 대부분 제거되었기 때문에 CSS3의 중요성이 높아졌다고 할 수 있다. CSS Level 2까지는 하나의 문서에 모든 기능들에 대한 내용들을 정의하였지만, CSS Level 3부터는 기능에 따라 모듈로 분리하여 정의하고 있다. 이는 웹의 기능이 발전함에 따라 HTML도 진화를 거듭하여 HTML5로의 표준화를 앞메 두고 있으며, 이를 기점으로 동적 애플리케이션 및 웹을

표현하는 디바이스들이 다양해짐에 따라서 웹의 표현도 더 다양해질 필요성이 확대되었다. 이로 인하여 웹의 표현 범위를 넓히기 위해 CSS의 기능들이 모듈별로 개발되고 있으며, 그 양도 CSS2.1과는 비교할 수 없을 만큼 방대함을 확인할 수 있다.

column-rule: 3px dotted aqua

Cascading Style Sheet Level 3

HTML01 HTML4.01에서 HTML5로 업그레이드가 되는 동안 CSS도 CSS2에서 CSS3로 업그레이드가 되면서 다양한 속성들이 추가되었다. 요즘은 모바일 기기 및 태블릿 PC등의 다양한 디바이스들이 출시되고 있고 관련 산업이 발전되면서 이러한 디바이스들에 초점이 옮겨지고 있는 상황이다. 이러한 각종 디바이스들에 적응적으로 레이아웃을 입히기 위해서는 반

드시 스타일시트의 도움을 받아야 하는 것은 누구나 알고 있는 사실이다. 특히, 기존 HTML4.01에서는 HTML 문서의 디자인을 위한 몇몇 요소들이 존재하였지만, HTML5에서는 이러한 요소들이 대부분 제거되었기 때문에 CSS3의 중요성이 높아졌다고 할 수 있다. CSS Level 2까지는 하나의 문서에 모든 기능들에 대한 내용들을 정의하였지만, CSS Level 3부터는 기능에 따라 모듈로 분리하여 정의하고 있다. 이는 웹의 기능이 발전함에 따라 HTML도 진화를 거듭하여 HTML5로의 표준화를 앞메 두고 있으며, 이를 기점으로 동적 애플리케이션 및 웹을

표현하는 디바이스들이 다양해짐에 따라서 웹의 표현도 더 다양해질 필요성이 확대되었다. 이로 인하여 웹의 표현 범위를 넓히기 위해 CSS의 기능들이 모듈별로 개발되고 있으며, 그 양도 CSS2.1과는 비교할 수 없을 만큼 방대함을 확인할 수 있다.

column-rule: 3px dashed red

Cascading Style Sheet Level 3

HTML01 HTML4.01에서 HTML5로 업그레이드가 되는 동안 CSS도 CSS2에서 CSS3로 업그레이드가 되면서 다양한 속성들이 추가되었다. 요즘은 모바일 기기 및 태블릿 PC등의 다양한 디바이스들이 출시되고 있고 관련 산업이 발전되면서 이러한 디바이스들에 초점이 옮겨지고 있는 상황이다. 이러한 각종 디바이스들에 적응적으로 레이아웃을 입히기 위해서는 반

드시 스타일시트의 도움을 받아야 하는 것은 누구나 알고 있는 사실이다. 특히, 기존 HTML4.01에서는 HTML 문서의 디자인을 위한 몇몇 요소들이 존재하였지만, HTML5에서는 이러한 요소들이 대부분 제거되었기 때문에 CSS3의 중요성이 높아졌다고 할 수 있다. CSS Level 2까지는 하나의 문서에 모든 기능들에 대한 내용들을 정의하였지만, CSS Level 3부터는 기능에 따라 모듈로 분리하여 정의하고 있다. 이는 웹의 기능이 발전함에 따라 HTML도 진화를 거듭하여 HTML5로의 표준화를 앞메 두고 있으며, 이를 기점으로 동적 애플리케이션 및 웹을

표현하는 디바이스들이 다양해짐에 따라서 웹의 표현도 더 다양해질 필요성이 확대되었다. 이로 인하여 웹의 표현 범위를 넓히기 위해 CSS의 기능들이 모듈별로 개발되고 있으며, 그 양도 CSS2.1과는 비교할 수 없을 만큼 방대함을 확인할 수 있다.

column-rule: 3px double blue

column-rule: 3px ridge green



다단 나누기

- **break-before** 속성은 특정 요소 앞에서 단을 분할하도록 지정
- **break-after** 속성은 특정 요소 뒤에서 단을 분할하도록 지정
- **break-inside** 속성은 해당 요소 안에서의 단을 강제로 분할할지의 여부를 지정

✓ 속성	break-before, break-after					CSS3
속성 값	auto always avoid left right page column avoid-page avoid-column					
속성 특징	초기값	auto	적용	블록-레벨 요소들	상속	X
JavaScript	<i>object.style.breakBefore</i> = "always"		<i>object.style.breakAfter</i> = "column"			
사용 예제	div { break-before : always }		div { break-after : column }			

✓ 속성	break-inside					CSS3
속성 값	auto avoid avoid-page avoid-column					
속성 특징	초기값	auto	적용	블록-레벨 요소들	상속	X
JavaScript	<i>object.style.breakInside</i> = "avoid-column"					
사용 예제	div { break-inside : avoid-column }					



다단 나누기



속성

break-before, break-after, break-inside

CSS3

속성 값	설명
<i>auto</i>	다단 분할을 강제하거나 금지하지 않도록 지정한다.
<i>always</i>	항상 요소(생성된 박스) 앞에서 페이지를 분할하도록 지정한다.
<i>avoid</i>	요소(생성된 박스) 앞에서 페이지 또는 단을 분할하지 않도록 지정한다.
<i>left</i>	요소(생성된 박스) 앞에서 페이지 또는 단을 분할하도록 지정한다. 다음에 오는 단은 왼쪽에 배치된다.
<i>right</i>	요소(생성된 박스) 앞에서 페이지 또는 단을 분할하도록 지정한다. 다음에 오는 단은 오른쪽에 배치된다.
<i>page</i>	요소(생성된 박스) 앞(break-before)이나 뒤(break-after)에서 항상 페이지를 분할하도록 지정한다.
<i>column</i>	요 요소(생성된 박스) 앞(break-before)이나 뒤(break-after)에서 항상 단을 분할하도록 지정한다.
<i>avoid-page</i>	요소(생성된 박스) 앞(break-before)이나 뒤(break-after), 또는 내부(break-inside)에서 페이지를 분할하지 않도록 지정한다.
<i>avoid-column</i>	요소(생성된 박스) 앞(break-before)이나 뒤(break-after), 또는 내부(break-inside)에서 단을 분할하지 않도록 지정한다.



■ 단의 확장



속성

column-span – 얼마나 많은 단을 차지하는지를 지정

CSS3

속성 값	none all					
속성 특징	초기값	none	적용	블록-레벨 요소들, 제외(float 및 absolute 로 위치한 요소들)	상속	X
JavaScript	object.style.columnSpan = "all"					
사용 예제	div { column-span: all }					

속성 값	설명
none	해당 요소가 여러 단에 확장해서 나타나지 않도록 지정한다.
all	해당 요소가 모든 단에 확장해서 나타나도록 지정한다. 단의 일부만 확장할 수는 없다.

예제	CSS3_09-01_ColumnSpan.html	
<p>Cascading Style Sheet Level 3</p> <p>HTML이 HTML4.01에서 HTML5로 업그레이드가 되는 동안 CSS도 CSS2에서 CSS3로 업그레이드가 되면서 다양한 속성들이 추가되었다. 요즘은 모바일 기기 및 태블릿 PC 등의 다양한 디바이스들이 출시되고 있고 관련 산업이 발전되면서 이러한 디바이스들에 초점이 옮겨지고 있는 상황이다. 이러한 각종 디바이스들에 적응적으로 레이아웃을 입히기 위해서는 반드시 스타일시트의 도움을 받아야 하는 것은 누구나 알고 있는 사실이다. 특히, 기존 HTML4.01에서는 HTML 문서의 디자인을 위한 몇몇 요소들이 존재하였지만, HTML5에서는 이러한 요소들이 대부분 제거되었기 때문에 CSS3의 중요성이 높아졌다. CSS3의 중요성이 높아졌고, 그 양도 CSS2.1과는 비교할 수 없을 만큼 방대함을 확인할 수 있다.</p> <p>표현하는 디바이스들이 다양해짐에 따라서 웹의 표현도 더 다양해질 필요성이 확대되었다. 이로 인하여 웹의 표현 범위를 넓히기 위해 CSS의 기능들이 모듈별로 개발되고 있으며, 그 양도 CSS2.1과는 비교할 수 없을 만큼 방대함을 확인할 수 있다.</p>	<p>Cascading Style Sheet Level 3</p> <p>HTML이 HTML4.01에서 HTML5로 업그레이드가 되는 동안 CSS도 CSS2에서 CSS3로 업그레이드가 되면서 다양한 속성들이 추가되었다. 요즘은 모바일 기기 및 태블릿 PC 등의 다양한 디바이스들이 출시되고 있고 관련 산업이 발전되면서 이러한 디바이스들에 초점이 옮겨지고 있는 상황이다. 이러한 각종 디바이스들에 적응적으로 레이아웃을 입히기 위해서는 반드시 스타일시트의 도움을 받아야 하는 것은 누구나 알고 있는 사실이다. 특히, 기존 HTML4.01에서는 HTML 문서의 디자인을 위한 몇몇 요소들이 존재하였지만, HTML5에서는 이러한 요소들이 대부분 제거되었기 때문에 CSS3의 중요성이 높아졌다. CSS3의 중요성이 높아졌고, 그 양도 CSS2.1과는 비교할 수 없을 만큼 방대함을 확인할 수 있다.</p> <p>표현하는 디바이스들이 다양해짐에 따라서 웹의 표현도 더 다양해질 필요성이 확대되었다. 이로 인하여 웹의 표현 범위를 넓히기 위해 CSS의 기능들이 모듈별로 개발되고 있으며, 그 양도 CSS2.1과는 비교할 수 없을 만큼 방대함을 확인할 수 있다.</p>	<p>Cascading Style Sheet Level 3</p> <p>HTML이 HTML4.01에서 HTML5로 업그레이드가 되는 동안 CSS도 CSS2에서 CSS3로 업그레이드가 되면서 다양한 속성들이 추가되었다. 요즘은 모바일 기기 및 태블릿 PC 등의 다양한 디바이스들이 출시되고 있고 관련 산업이 발전되면서 이러한 디바이스들에 초점이 옮겨지고 있는 상황이다. 이러한 각종 디바이스들에 적응적으로 레이아웃을 입히기 위해서는 반드시 스타일시트의 도움을 받아야 하는 것은 누구나 알고 있는 사실이다. 특히, 기존 HTML4.01에서는 HTML 문서의 디자인을 위한 몇몇 요소들이 존재하였지만, HTML5에서는 이러한 요소들이 대부분 제거되었기 때문에 CSS3의 중요성이 높아졌다. CSS3의 중요성이 높아졌고, 그 양도 CSS2.1과는 비교할 수 없을 만큼 방대함을 확인할 수 있다.</p> <p>표현하는 디바이스들이 다양해짐에 따라서 웹의 표현도 더 다양해질 필요성이 확대되었다. 이로 인하여 웹의 표현 범위를 넓히기 위해 CSS의 기능들이 모듈별로 개발되고 있으며, 그 양도 CSS2.1과는 비교할 수 없을 만큼 방대함을 확인할 수 있다.</p>
	column-span: none	column-span: all



■ 단의 콘텐츠 균형 조정



속성

column-fill - 다단 영역의 콘텐츠 흐름의 영향을 받는 방식 지정

CSS3

속성 값	auto balance				
속성 특징	초기값	balance	적용	다단 요소들	상속
JavaScript	<code>object.style.columnFill = "auto"</code>				
사용 예제	<code>div { column-fill: auto }</code>				

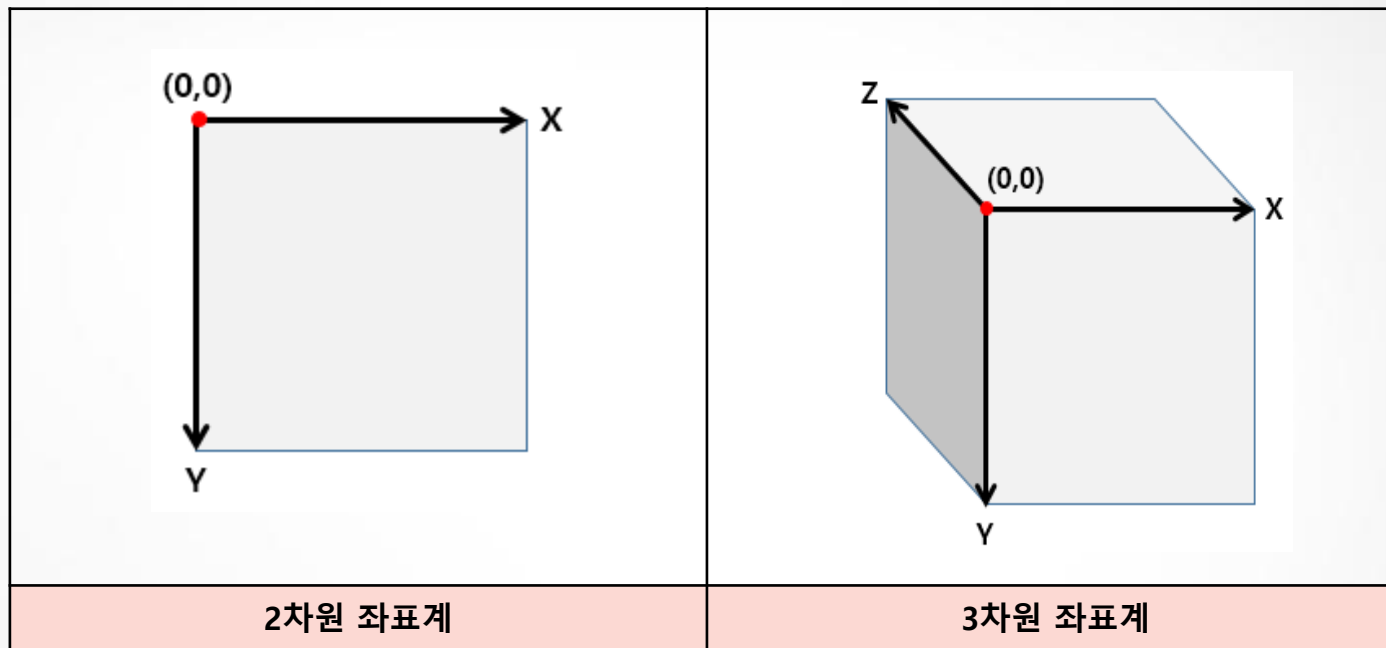
속성 값	설명
<i>balance</i>	콘텐츠가 단의 크기에 따라서 균형을 이루도록 지정한다.
<i>auto</i>	콘텐츠가 단의 순서대로 채워지도록 지정한다. 각 단의 길이는 다를 수 있다.

예제	CSS3_09-01_ColumnFill.html	
<p>Cascading Style Sheet Level 3</p> <p>HTML이 HTML4.01에서 HTML5로 업그레이드가 되는 동안 CSS도 CSS2에서 CSS3로 업그레이드가 되면서 다양한 속성들이 추가되었다. 요즘은 모바일 기기 및 태블릿 PC 등의 다양한 디바이스들이 출시되고 있고 관련 산업이 발전되면서 이러한 디바이스들에 초점이 옮겨지고</p> <p>있는 상황이다. 이러한 각종 디바이스들에 적응적으로 레이아웃을 입히기 위해서는 반드시 스타일시트의 도움을 받아야 하는 것은 누구나 알고 있는 사실이다. 특히, 기존 HTML4.01에서는 HTML 문서의 디자인을 위한 몇몇 요소들이 존재하였지만, HTML5에서는 이러한 요소들이 대부분 제거되었기 때문에 CSS3의 중요성이 높아졌다고 할 수 있다. CSS Level 2까지는 하나의 문서에 모든 기능들에 대한 내용들을 정의하였지만, CSS Level 3부터</p> <p>는 기능에 따라 모듈로 분리하여 정의하고 있다. 이는 웹의 기능이 발전함에 따라 HTML5로의 진화를 앞두고 있는 사실이다. 특히, 기존 HTML4.01에서는 HTML 문서의 디자인을 위한 몇몇 요소들이 존재하였지만, HTML5에서는 이러한 요소들이 대부분 제거되었기 때문에 CSS3의 중요성이 높아졌다고 할 수 있다. CSS Level 2까지는 하나의 문서에 모든 기능들에 대한 내용들을 정의하였지만, CSS Level 3부터</p>	<p>Cascading Style Sheet Level 3</p> <p>HTML이 HTML4.01에서 HTML5로 업그레이드가 되는 동안 CSS도 CSS2에서 CSS3로 업그레이드가 되면서 다양한 속성들이 추가되었다. 요즘은 모바일 기기 및 태블릿 PC 등의 다양한 디바이스들이 출시되고 있고 관련 산업이 발전되면서 이러한 디바이스들에 초점이 옮겨지고</p> <p>있는 상황이다. 이러한 각종 디바이스들에 적응적으로 레이아웃을 입히기 위해서는 반드시 스타일시트의 도움을 받아야 하는 것은 누구나 알고 있는 사실이다. 특히, 기존 HTML4.01에서는 HTML 문서의 디자인을 위한 몇몇 요소들이 존재하였지만, HTML5에서는 이러한 요소들이 대부분 제거되었기 때문에 CSS3의 중요성이 높아졌다고 할 수 있다. CSS Level 2까지는 하나의 문서에 모든 기능들에 대한 내용들을 정의하였지만, CSS Level 3부터</p> <p>는 기능에 따라 모듈로 분리하여 정의하고 있다. 이는 웹의 기능이 발전함에 따라 HTML5로의 진화를 앞두고 있는 사실이다. 특히, 기존 HTML4.01에서는 HTML 문서의 디자인을 위한 몇몇 요소들이 존재하였지만, HTML5에서는 이러한 요소들이 대부분 제거되었기 때문에 CSS3의 중요성이 높아졌다고 할 수 있다. CSS Level 2까지는 하나의 문서에 모든 기능들에 대한 내용들을 정의하였지만, CSS Level 3부터</p>	<p>의 기능이 발전함에 따라 HTML5로의 진화를 앞두고 있는 사실이다. 특히, 기존 HTML4.01에서는 HTML 문서의 디자인을 위한 몇몇 요소들이 존재하였지만, HTML5에서는 이러한 요소들이 대부분 제거되었기 때문에 CSS3의 중요성이 높아졌다고 할 수 있다. CSS Level 2까지는 하나의 문서에 모든 기능들에 대한 내용들을 정의하였지만, CSS Level 3부터</p> <p>는 기능에 따라 모듈로 분리하여 정의하고 있다. 이는 웹의 기능이 발전함에 따라 HTML5로의 진화를 앞두고 있는 사실이다. 특히, 기존 HTML4.01에서는 HTML 문서의 디자인을 위한 몇몇 요소들이 존재하였지만, HTML5에서는 이러한 요소들이 대부분 제거되었기 때문에 CSS3의 중요성이 높아졌다고 할 수 있다. CSS Level 2까지는 하나의 문서에 모든 기능들에 대한 내용들을 정의하였지만, CSS Level 3부터</p>
	column-fill: balance	column-fill: auto



표준화 문서	CSS Transforms Module Level 1 - http://dev.w3.org/csswg/css-transforms/
표준화 단계	W3C Editor's Draft, (2014-05-28)

좌표 시스템



- X축은 좌우로 값이 변한다
- Y축은 위아래로 변한다.
- Z축은 앞뒤로 값이 변하기 때문에 보고 있는 사람의 쪽으로 가까울수록 값이 커지게 되고, 멀어질수록 값이 작아지게 된다.



■ 변형 함수



속성

transform – 요소 박스의 변형 지정하기

CSS3

속성 값	none <transform-list>				<transform-list> = <transform-function> +	
속성 특징	초기값	none	적용	변형 가능한 요소들	상속	X
Java Script	<code>object.style.transform = "translate(10px, 20px)"</code> <code>object.style.transform = "skew(20deg, 30deg)"</code>				<code>object.style.transform = "scale(1.3, 1.2)"</code> <code>object.style.transform = "rotate(180deg)"</code> <code>object.style.transform = "matrix()"</code>	
사용 예제	<div <b="" {="">transform: translate(10px, 20px) }</div> <div <b="" {="">transform: scale(1.3, 1.2) }</div> <div <b="" {="">transform: rotate(180deg) }</div>				<div <b="" {="">transform: skew(20deg, 30deg) }</div> <div <b="" {="">transform: matrix() }</div>	



■ 변형 함수



속성

transform – 요소 박스의 변형 지정하기

CSS3

transform 변형 함수 값	기능
translate(dx, dy)	요소 박스를 지정한 거리만큼 수평과 수직으로 이동 한다.
translate3d(dx, dy, dz)	요소 박스를 지정한 거리만큼 수평과 수직, 앞뒤로 이동 한다.
translateX(dx)	요소 박스를 지정한 거리만큼 수평으로 이동 한다.
translateY(dy)	요소 박스를 지정한 거리만큼 수직으로 이동 한다.
translateZ(dz)	요소 박스를 지정한 거리만큼 앞뒤로 이동 한다.
scale(sx, sy)	요소 박스를 지정한 크기만큼 수평과 수직으로 확대 또는 축소 한다.
scaleX(sx)	요소 박스를 지정한 크기만큼 수평으로 확대 또는 축소 한다.
scaleY(sy)	요소 박스를 지정한 크기만큼 수직으로 확대 또는 축소 한다.
scaleZ(sz)	요소 박스를 지정한 크기만큼 앞뒤로 확대 또는 축소 한다.
rotate(angle)	요소 박스를 지정한 각도만큼 회전 한다.
rotate3d(rx, ry, rz, angle)	요소 박스를 지정한 각도만큼 수평과 수직, 앞뒤로 회전 한다.
rotateX(angle)	요소 박스를 지정한 각도만큼 수평으로 회전 한다.
rotateY(angle)	요소 박스를 지정한 각도만큼 수직으로 회전 한다.
rotateZ(angle)	요소 박스를 지정한 각도만큼 앞뒤로 회전 한다.
skew(x_angle, y_angle)	요소 박스를 지정한 각도만큼 수평과 수직으로 왜곡 한다.
skewX(angle)	요소 박스를 지정한 각도만큼 수평으로 왜곡 한다.
skewY(angle)	요소 박스를 지정한 각도만큼 수직으로 왜곡 한다.
matrix(n, n, n, n, n, n)	요소 박스를 6개의 매트릭스를 사용하여 2D 변형 을 한다.
matrix3d(n, n, n, n, n, n, n, n, n, n, n, n, n, n, n)	요소 박스를 4x4 매트릭스의 16개 값을 이용하여 3D 변형 을 한다.



■ 변형 함수



속성

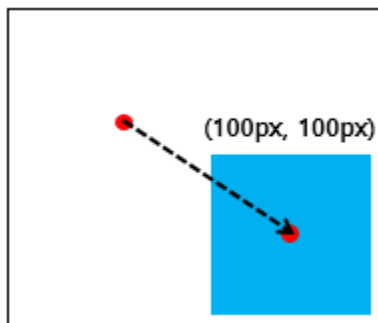
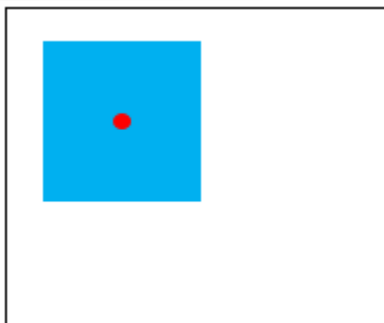
transform – 요소 박스의 변형 지정하기

CSS3



translate() 함수 – 요소 박스 이동하기

- 요소 박스를 x축(dx), y축(dy), z축(dz)으로 지정한 거리만큼 이동



예제

CSS3_09-02_Transform_Translate.html



transform 속성은 요소 박스의 변형을 지정하는 속성으로, 변형하고자 하는 형태별로 함수 타입의 속성 값을 이용한다. 이 때, 함수의 속성 값은 공백으로 구분하여 여러 개의 속성 값을 지정할 수 있다.



transform 속성은 요소 박스의 변형을 지정하는 속성으로, 변형하고자 하는 형태별로 함수 타입의 속성 값을 이용한다. 이 때, 함수의 속성 값은 공백으로 구분하여 여러 개의 속성 값을 지정할 수 있다.

transform: **translate**(20px, 10px)transform: **translate**(-20px, -10px)



변형 함수



속성


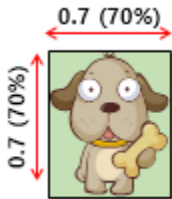
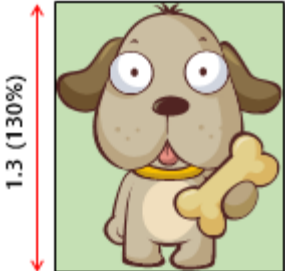

transform – 요소 박스의 변형 지정하기

CSS3



scale() 함수 – 요소 박스 확대/축소하기

- 요소 박스를 지정한 x축(dx), y축(dy), z축(dz) 만큼 확대 및 축소

			
본래 크기	70% 크기로 축소	130% 크기로 확대	가로(X축) 크기만 130% 확대

예제

CSS3_09-02_Transform_Scale.html



transform 속성은 요소 박스의 변형을 지정하는 속성입니다. 변형하고자 하는 요소를 transform 함수 타입의 속성 값을 이용한다. 이 때, 함수의 속성 값은 공백으로 구분하여 여러 개의 속성 값을 지정할 수 있다.

transform: **scale**(1.3, 1.2)transform: **scale**(0.7, 0.5)



■ 변형 함수



속성

transform – 요소 박스의 변형 지정하기

CSS3

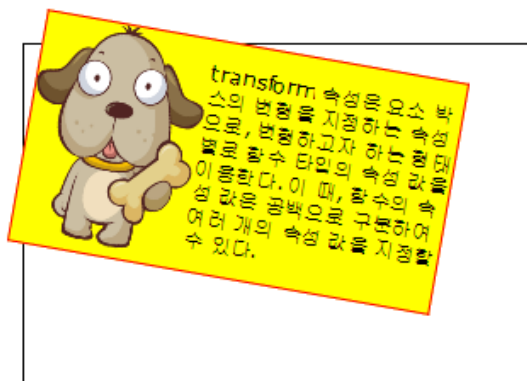


rotate() 함수 – 요소 박스 회전하기

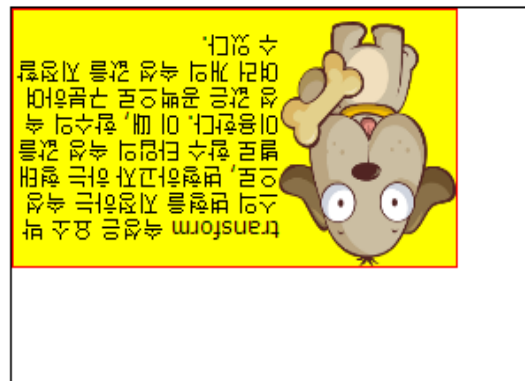
- 요소 박스를 지정한 x축(dx), y축(dy), z축(dz)을 기준으로 지정한 각도만큼 회전

예제

CSS3_09-02_Transform_Rotate.html



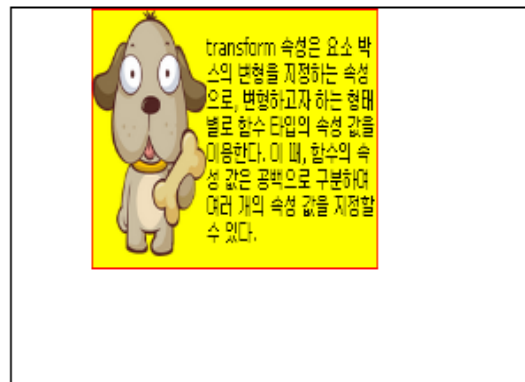
transform: rotate(10deg)



transform: rotate(180deg)



transform: rotateX(50deg)



transform: rotateY(50deg)



■ 변형 함수



속성

transform – 요소 박스의 변형 지정하기

CSS3

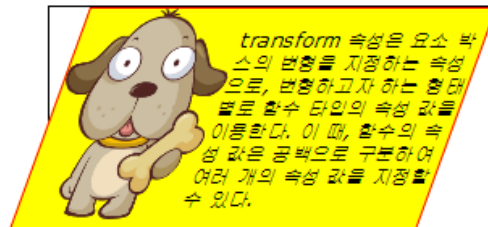
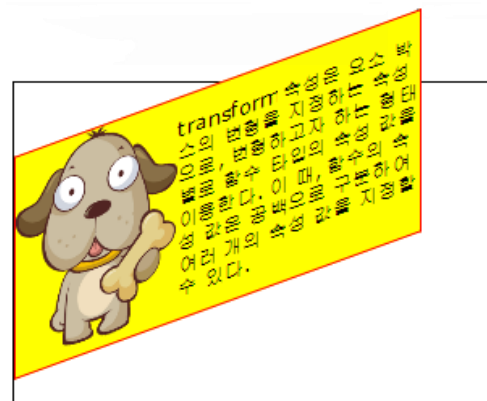
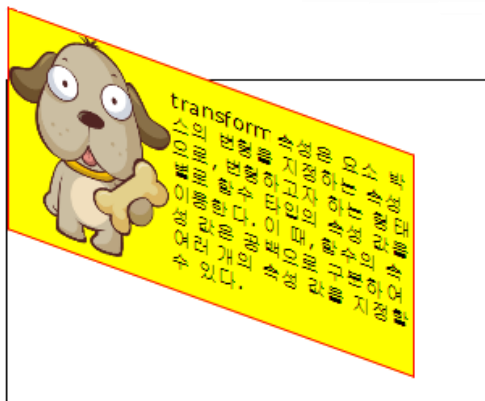


skew() 함수 – 요소 박스 기울이기

- x축(dx), y축(dy)을 기준으로 지정한 각도만큼 요소 박스를 기울이기

예제

CSS3_09-02_Transform_Skew.html

transform: **skewX**(20deg)transform: **skewX**(160deg)transform: **skewY**(20deg)transform: **skewY**(160deg)



■ 변형 함수



속성

transform – 요소 박스의 변형 지정하기

CSS3

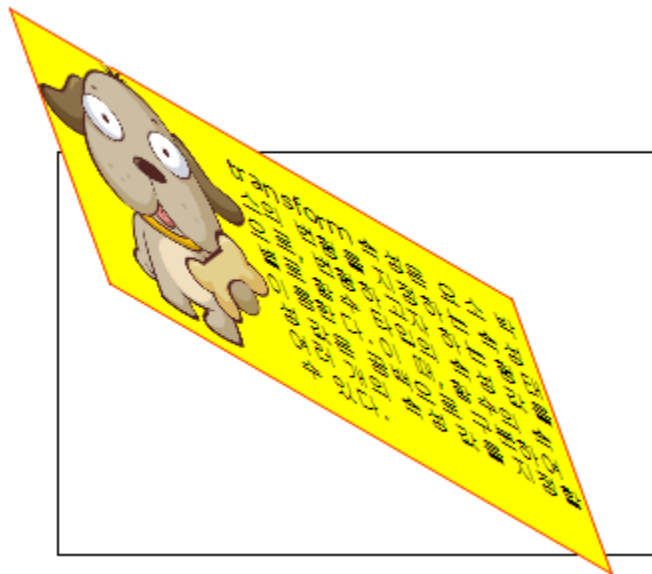
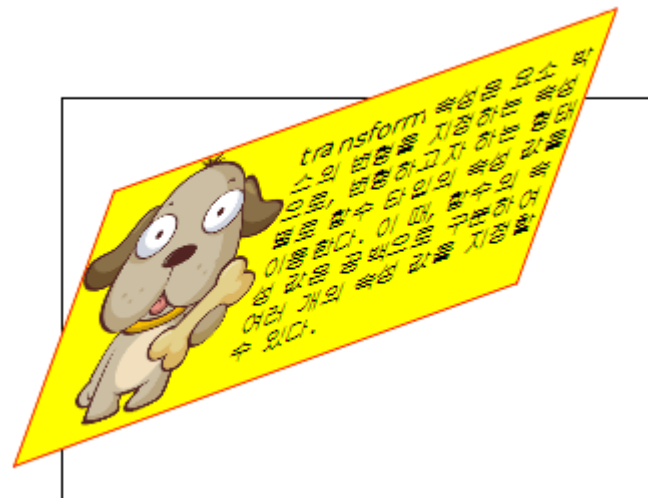


skew() 함수 – 요소 박스 기울이기

- x축(dx), y축(dy)을 기준으로 지정한 각도만큼 요소 박스를 기울이기

예제

CSS3_09-02_Transform_Skew.html

transform: **skew**(20deg, 30deg)transform: **skew**(160deg, 160deg)



■ 변형 함수



속성

transform – 요소 박스의 변형 지정하기

CSS3



matrix() 함수 – 요소 박스 변형하기

- 요소 박스의 이동, 확대 및 축소, 회전, 기울이기 등을 복합적으로 한번에 지정

예제

CSS3_09-02_Transform_Matrix.html

transform: **matrix**(0.866,0.5,-0.5,0.866,0,0)transform: **matrix**(0.586,0.8,-0.8,0.586,10,0)



■ 변형 기준 및 원근감



속성

transform-origin – 변형 기준 지정하기

CSS3

속성 값	<percentage> <length> left center right top bottom					
속성 특징	초기값	50% 50%	적용	변형 가능한 요소들	상속	X
JavaScript	<i>object.style.transformOrigin</i> = "20% 30%"					
사용 예제	div { transform-origin : left top } /* 기준점을 왼쪽 상단으로 지정 */					

속성 값	설명
percentage	변형의 기준이 되는 위치를 박스 가로 폭에 대한 상대적인(100분율) 거리로(offset) 지정한다
length	변형의 기준이 되는 위치를 박스 좌측 상단으로부터의 떨어진 고정된 거리를 지정한다
top	세로 위치 0%인 지점을 기준으로 지정한다.
right	가로 위치 100%인 지점을 기준으로 지정한다.
bottom	세로 위치 100%인 지점을 기준으로 지정한다.
left	가로 위치 0%인 지점을 기준으로 지정한다.
center	가로 및 세로 위치의 가운데 50%인 지점을 기준으로 한다.



■ 변형 기준 및 원근감



속성

transform-origin - 변형 기준 지정하기

CSS3

예제

CSS3_09-02_TransformOrigin.html



transform-origin: left top



transform-origin: right top



transform-origin: left bottom



transform-origin: right bottom



■ 변형 기준 및 원근감



속성

transform-style – 3차원 좌표 전달하기

CSS3

- 3차원 공간에서 변형이 적용된 3차원 좌표를 하위 요소에게 전달할 것인지를 결정

속성 값	flat preserve-3d					
속성 특징	초기값	flat	적용	변형 가능한 요소들	상속	X
JavaScript	<code>object.style.transformStyle = "preserve-3d"</code>					
사용 예제	<code>div { transform-style: preserve-3d } /* 기준점을 왼쪽 상단으로 지정 */</code>					

속성 값	설명
<i>flat</i>	하위 요소들의 좌표 값들은 부모 요소의 3차원(3D) 좌표와 관계없이 2차원(2D) 좌표 영역에 표시되도록 지정한다.
<i>preserve-3d</i>	하위 요소들의 좌표 값들은 부모 요소의 3차원 좌표 값을 전달받아 표시하도록 지정한다.



■ 변형 기준 및 원근감

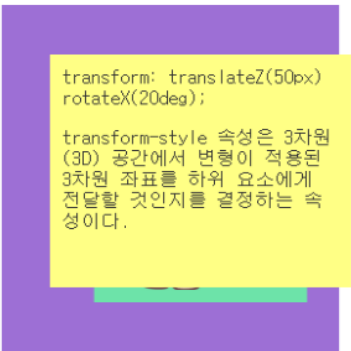
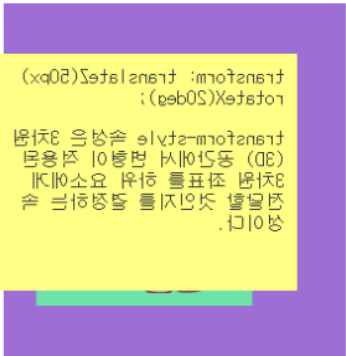
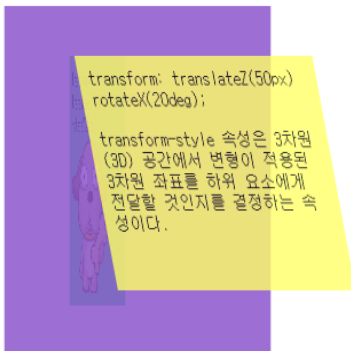



속성

transform-style – 3차원 좌표 전달하기

CSS3

- 3차원 공간에서 변형이 적용된 3차원 좌표를 하위 요소에게 전달할 것인지를 결정

예제	CSS3_09-02_TransformStyle.html
 <p>transform: translateZ(50px); rotateX(20deg);</p> <p>transform-style 속성은 3차원 (3D) 공간에서 변형이 적용된 3차원 좌표를 하위 요소에게 전달할 것인지를 결정하는 속성이다.</p>	 <p>(xq02)S2at1anet :mroTnert :(eeb0S)Xetator</p> <p>트랜스 에이션 3D-트랜스 용 100% 100% (0%) 100% 100% 트랜스 속 3D-트랜스 3D-트랜스 100%</p>
transform-style: flat	transform-style: flat
 <p>transform: translateZ(50px); rotateX(20deg);</p> <p>transform-style 속성은 3차원 (3D) 공간에서 변형이 적용된 3차원 좌표를 하위 요소에게 전달할 것인지를 결정하는 속성이다.</p>	 <p>:mroTnert (xq02)S2at1anet :(eeb0S)Xetator</p> <p>트랜스 에이션 3D-트랜스 용 100% 100% (0%) 100% 100% 트랜스 속 3D-트랜스 3D-트랜스 100%</p>
transform-style: preserve-3d	transform-style: preserve-3d



■ 변형 기준 및 원근감



속성

perspective – 원근감 지정하기

CSS3

- 3차원 공간에서 해당 요소와 관측 점과의 거리를 조절해서(사용자가 있는 방향으로 잡아당기거나 반대 방향으로 밀어내서) 원근감을 지정

속성 값	none <length>					
속성 특징	초기값	none	적용	변형 가능한 요소들	상속	X
JavaScript	<code>object.style.perspective = "200px"</code>					
사용 예제	<code>div { perspective: 200px } /* 기준점을 왼쪽 상단으로 지정 */</code>					

속성 값	설명
<i>none</i>	원근감이 없는 변형을 지정하도록 하기 때문에 모든 개체는 캔버스에 평탄하게 나타난다.
<i>length</i>	요소가 투영(projection)되는 중심까지의 거리를 지정하는 것으로, 사용자가 있는 쪽으로 얼마나 이동하는지를 픽셀 크기로 지정한다. 속성 값은 0보다 커야 하며, 값이 클수록 사용자로부터 멀게 보인다.



■ 변형 기준 및 원근감

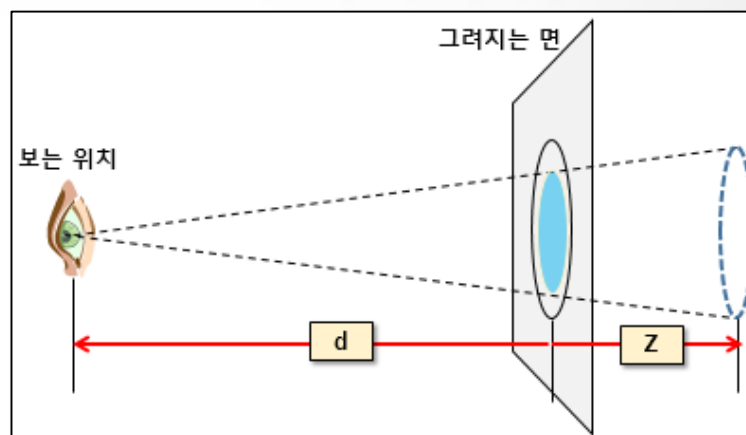
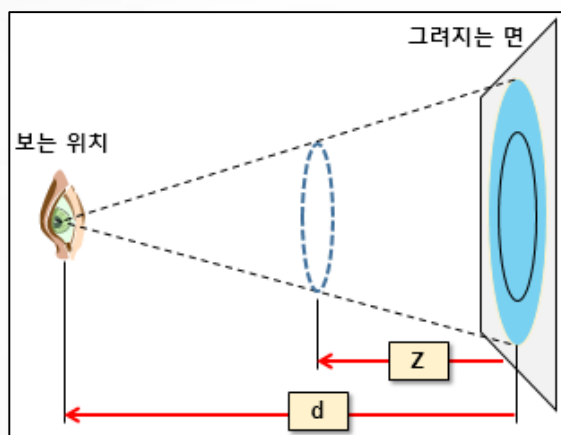


속성

perspective – 원근감 지정하기

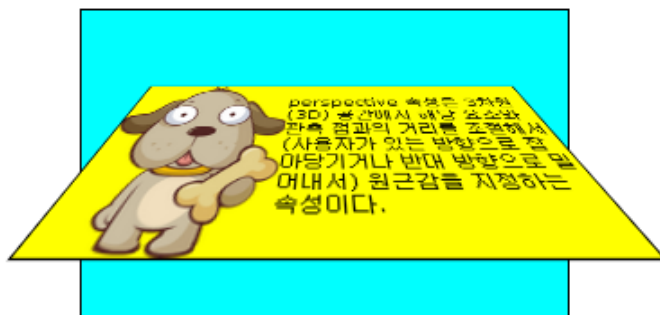
CSS3

- 3차원 공간에서 해당 요소와 관측 점과의 거리를 조절해서(사용자가 있는 방향으로 잡아당기거나 반대 방향으로 밀어내서) 원근감을 지정

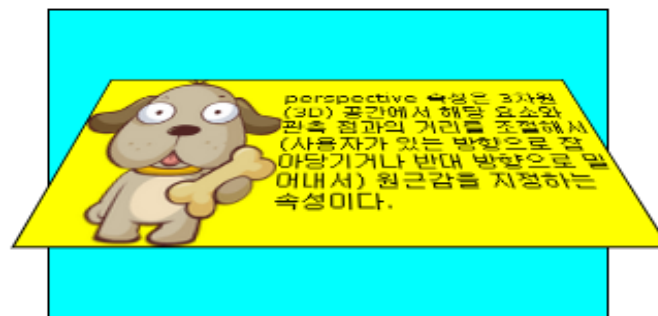


예제

CSS3_09-02_Perspective.html



perspective: 160px



perspective: 250px



■ 변형 기준 및 원근감



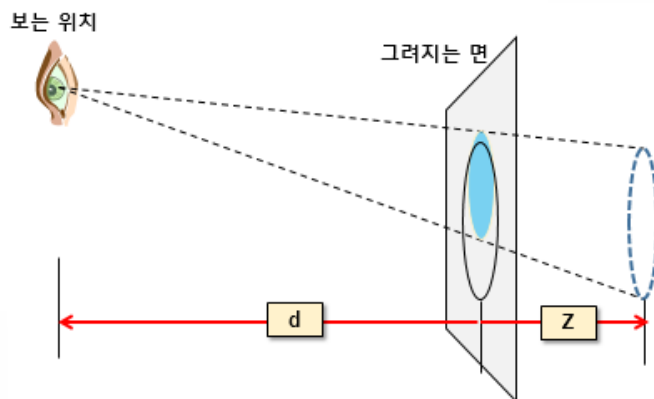
속성

perspective-origin – 원근감 방향 지정하기

CSS3

■ 3차원 공간에서 원근감의 방향을 지정

속성 값	<percentage> <length> left center right top bottom				
속성 특징	초기값	50% 50%	적용	변형 가능한 요소들	상속 X
JavaScript	<code>object.style.perspectiveOrigin = "200px"</code>				
사용 예제	<code>div { perspective-origin: 200px } /* 기준점을 왼쪽 상단으로 지정 */</code>				

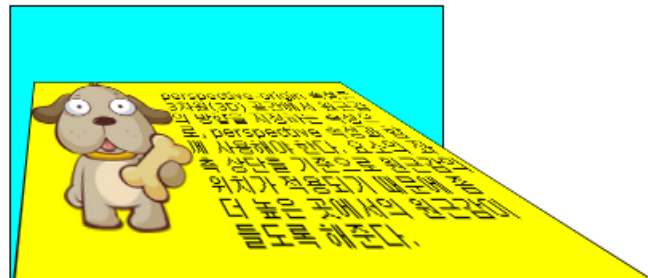


예제

CSS3_09-02_TransformOrigin.html



perspective-origin: 50% 50%



perspective-origin: 10% 50%



■ 변형 기준 및 원근감



속성

backface-visibility – 요소의 뒷면 표시하기

CSS3

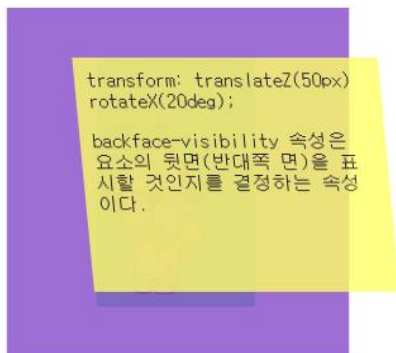
■ 요소의 뒷면(반대쪽 면)을 표시할 것인지를 결정

속성 값	visible hidden				
속성 특징	초기값	visible	적용	변형 가능한 요소들	상속 X
JavaScript	<code>object.style.backfaceVisibility = "hidden"</code>				
사용 예제	div { backface-visibility : hidden } /* 기준점을 왼쪽 상단으로 지정 */				

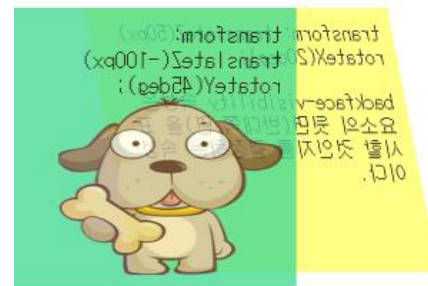
속성 값	설명
<i>visible</i>	요소의 뒷면을 표시하도록 지정한다.
<i>hidden</i>	요소의 뒷면을 표시하지 않도록 지정한다.

예제

CSS3_09-02_BackfaceVisibility.html



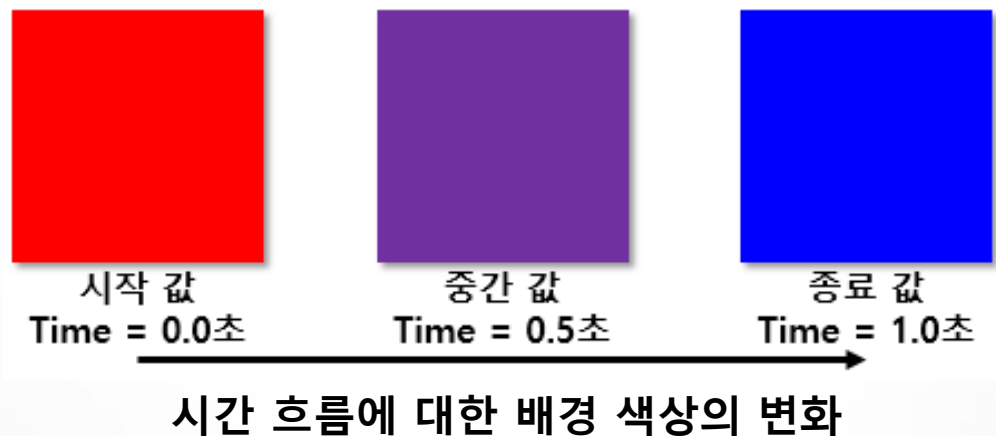
backface-visibility: visible




backface-visibility: hidden

표준화 문서	CSS Transitions - http://dev.w3.org/csswg/css-transitions/
표준화 단계	W3C Editor's Draft, (2014-01-29)

- **트랜지션(Transition)이란, 어떠한 상태가 다른 상태에 영향을 주는 것을 의미하는 것으로, CSS3에서의 트랜지션은 특정 스타일에서 다른 스타일로 바뀌는 것을 말한다.**
 - 애니메이션이 가능한 CSS 속성들만 트랜지션의 적용이 가능하다.
 - 트랜지션은 이전 값에서 시간이 지남에 따라 새로운 속성 값으로의 계산된 값이기 때문에 바로 표시되는 효과를 볼 수 있다.



트랜지션 속성 지정

	속성	transition-property – 요소 속성 변환하기	CSS3
-----------------------------------------------------------------------------------	----	----------------------------------	------

- 요소에 지정된 속성을 변환(transition) 하고자 할 때 사용

속성 값	none <single-transition-property> [',' <single-transition-property>]* <single-transition-property> = all <IDENT>					
속성 특징	초기값	all	적용	모든 요소들 가상 요소(:before 와 :after)	상속	X
JavaScript	<i>object.style.transitionProperty</i> = "width"					
사용 예제	div { transition-property : width }					

속성 값	설명
none	어떤 속성도 트랜지션을 일으키지 않도록 지정한다.
all	요소에 지정된 모든 속성이 트랜지션을 일으키도록 지정한다.
IDENT	트랜지션 효과를 적용할 속성 이름을 지정한다. 예를 들면, 배경색만 바꿀 것인지, 폭이나 높이를 바꿀 것인지를 선택하면 된다.

트랜지션 속성 지정



속성

transition-property – 요소 속성 변환하기

CSS3

예제

CSS3_09-03_TransitionProperty.html

```
img { width: 90%; height: 90% }
div { width: 150px; height: 150px; }
div > #Box1 { transition-property: width; transition-duration: 2s; }
div > #Box2 { transition-property: height; transition-duration: 2s; }
div > #Box3 { transition-property: all; transition-duration: 2s; }
#Box1:hover { width: 300px; }
#Box2:hover { height: 300px; }
#Box3:hover { width: 300px; height: 300px; padding: 30px; border-radius: 50px; background: coral; }
```



마우스 오버시 폭이
늘어납니다.

transition-property: width

⇒
트랜지션



마우스 오버시 폭이 늘어납니다.

transition-property: width



마우스 오버시 폭, 높
이, 배경색, 테두리가
변합니다.

transition-property: all


⇒
트랜지션



마우스 오버시 폭, 높이, 배경색, 테두리가
변합니다.


transition-property: all

트랜지션 속성 지정

	속성	transition-duration – 트랜지션 시간 지정하기	CSS3
-----------------------------------------------------------------------------------	----	------------------------------------	------

트랜지션이 진행되는 시간을 지정


속성 값	<time> [, <time>]*				
속성 특징	초기값	0s	적용	모든 요소들 가상 요소(:before 와 :after)	상속 X
JavaScript	<code>object.style.transitionDuration = "2s"</code>				
사용 예제	<code>div { transition-duration: 2s }</code>				

	속성	transition-timing-function – 트랜지션 진행 시간 지정하기	CSS3
-----------------------------------------------------------------------------------	----	----------------------------------------------	------

트랜지션이 진행되는 속도의 형태를 지정

속성 값	<single-transition-timing-function> [', <single-transition-timing-function>]* <single-transition-timing-function> = ease linear ease-in ease-out ease-in-out step-start step-end steps(<integer>[, [start end]]?) cubic-bezier(<number>, <number>, <number>, <number>)				
속성 특징	초기값	ease	적용	모든 요소들 가상 요소(:before 와 :after)	상속 X
JavaScript	<code>object.style.transitionTimingFunction = "linear"</code>				
사용 예제	<code>div { transition-timing-function: linear }</code>				

트랜지션 속성 지정

	속성	transition-timing-function – 트랜지션 진행 시간 지정하기	CSS3
-----------------------------------------------------------------------------------	----	----------------------------------------------	------

속성 값	설명
<i>ease</i>	(기본 값) 진행 속도가 처음에는 천천히 시작해서 점점 빠르게 진행되다가 후반부에는 다시 느려지는 형태이다. cubic-bezier(0.25, 0.1, 0.25, 1)과 같다.
<i>linear</i>	처음부터 끝까지 동일한 속도로 트랜지션을 진행한다. cubic-bezier(0, 0, 1, 1)과 같다.
<i>ease-in</i>	트랜지션의 시작을 느리게 해서 후반부로 갈수록 진행 속도가 빨라진다. cubic-bezier(0.42, 0, 1, 1)와 같다.
<i>ease-out</i>	트랜지션의 시작을 빠르게 해서 후반부로 갈수록 진행 속도가 느려진다. cubic-bezier(0, 0, 0.58, 1)와 같다.
<i>ease-in-out</i>	트랜지션의 시작을 느리게 시작해서 느리게 끝난다. 전반부의 형태는 'ease-in'과 같고 후반부는 "ease-out"과 같다.
<i>step-start</i>	트랜지션이 진행되는 구간의 시작점에서 속성 값이 변경된다. steps(1, start)와 같다.
<i>step-end</i>	트랜지션이 진행되는 구간의 종료점에서 속성 값이 변경된다. steps(1, end)와 같다.
<i>steps</i>	트랜지션이 진행되는 구간을 임의의 단계로 구분해서 각 단계의 시작점 또는 종료점에서 속성 값을 변경한다. steps(<integer>[, [start end]]?)와 같이 2개의 매개변수를 이용하는 형식에서, 첫 번째 i nteger는 함수의 간격을 의미하며 0이상의 값을 지정해야 한다. 두 번째 값은 값은 구간 내에서 변화가 발생하는 지점을 나타내는 것으로, 시작(start) 또는 종료(end) 값이 된다.
<i>cubic-bezier()</i> 함수	베지에 곡선의 시작점과 종료점의 제어점을 직접 지정해서 트랜지션의 속도를 지정한다. cubic-bezi er(<number>, <number>, <number>, <number>)와 같이 4개의 값을 이용하여 베지에 곡선의 제 어점인 P ₁ 와 P ₂ 의 좌표(x1, y1, x2, y2)를 지정한다. 좌표 x값은 0~1사이의 값이어야 하지만, y값은 1 이상의 값이 될 수 있다.

트랜지션 속성 지정



속성

transition-timing-function – 트랜지션 진행 시간 지정하기

CSS3

예제

CSS3_09-03_TransitionTimingFunction.html

아래 박스에 마우스 오버시 트랜지션이 동작합니다.

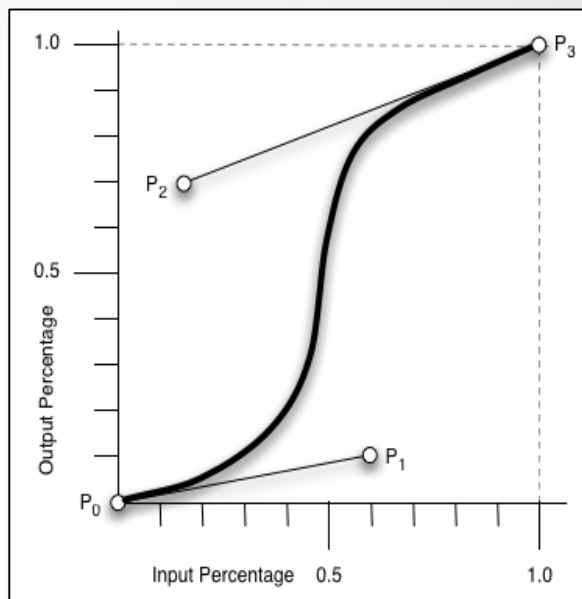
linear

ease


ease-in

ease-out

ease-in-out



트랜지션 속성 지정


	속성	transition-delay – 트랜지션 지연 시간 지정하기	CSS3
-----------------------------------------------------------------------------------	----	------------------------------------	------

- 트랜지션이 시작되기 전에 지연되는 시간을 지정

속성 값	<time> [, <time>]*				
속성 특징	초기값	0s	적용	모든 요소들 가상 요소(:before 와 :after)	상속 X
JavaScript	<code>object.style.transitionDelay = "1s"</code>				
사용 예제	<code>div { transition-delay: 1s }</code>				

예제	CSS3_09-03_TransitionDelay.html
<p>아래 박스에 마우스 오버시 트랜지션이 동작합니다.</p> <div> <div>linear(delay 1초)</div> <div>ease(delay 2초)</div> <div>ease-in(delay 3초)</div> <div>ease-out(delay 1초)</div> <div>ease-in-out(delay 1초)</div> </div>	


트랜지션 지정

	속성	transition – 트랜지션 지정하기	CSS3
-----------------------------------------------------------------------------------	----	------------------------	------

- 지금까지 살펴본 트랜지션 관련 속성들을 한번에 지정할 수 있는 대표 속성

속성 값	<single-transition> [' ' <single-transition>]* < single-transition > = [none <single-transition-property>] <transition-duration> <single-transition-timing-function> <transition-delay >					
속성 특징	초기값	개별 속성 참조	적용	모든 요소들 가상 요소(:before 와 :after)	상속	X
JavaScript	<code>object.style.transition = "all 2s"</code>					
사용 예제	div { transition : all 2s }					

트랜지션 지정

	속성	transition – 트랜지션 지정하기	CSS3
-----------------------------------------------------------------------------------	----	------------------------	------

트랜지션을 사용할 수 있는 속성과 속성 값 목록

속성 이름	속성 값	속성 이름	속성 값
background-color	color	margin-right	length
background-position	percentage, length	margin-top	length
border-bottom-color	color	max-height	percentage, length
border-bottom-width	length	max-width	percentage, length
border-left-color	color	min-height	percentage, length
border-left-width	length	min-width	percentage, length
border-right-color	color	opacity	number
border-right-width	length	outline-color	color
border-spacing	length	outline-width	length
border-top-color	color	padding-bottom	length
border-top-width	length	padding-left	length
bottom	percentage, length	padding-right	length
clip	rectangle	padding-top	length
color	color	right	percentage, length
font-size	percentage, length	text-indent	percentage, length
font-weight	font weight	text-shadow	shadow
height	percentage, length	top	percentage, length
left	percentage, length	vertical-align	percentage, length
letter-spacing	length	visibility	visibility
line-height	number, length	width	percentage, length
margin-bottom	length	word-spacing	percentage, length
margin-left	length	z-index	

표준화 문서	CSS Marquee Module Level 3 - http://www.w3.org/TR/css3-marquee/
표준화 단계	W3C Candidate Recommendation, (2008-12-05)

- 트랜지션이 한 스타일에서 다른 스타일로 변경될 때 진행 시간을 지정해서 속성 값이 부드럽게 변경되도록 한 것이라면,
- 마키 속성은 시간의 흐름에 따라서 박스의 콘텐츠를 지정한 특성에 맞게 이동시키도록 정의한 것


마키 스크롤 방법 지정

<div> <div></div> <div>속성</div> </div>	overflow-style – 마키 기능의 스크롤 방법 지정	CSS2/CSS3
----------------------------------------	-----------------------------------	-----------

속성 값	auto marquee-line marquee-block				
속성 특징	초기값	auto	적용	overflow와 동일	상속 yes
JavaScript	<code>object.style.overflowStyle = "auto"</code>				
사용 예제	<code>div { overflow-style: auto }</code>				

속성 값	설명
<i>auto</i>	브라우저가 스크롤링 도구를 선택하도록 한다. 브라우저마다 다를 수 있지만, 일반적으로 마키와 스크롤바는 동일한 동작을 보인다.
marquee-line	수평으로의 스크롤링(콘텐츠가 왼쪽이나 오른쪽으로 넘치는 경우)이 되도록 지정.
marquee-block	수직으로의 스크롤링(콘텐츠가 박스의 위쪽이나 아래쪽으로 넘치는 경우)이 되도록 지정

마키 스타일 지정



속성

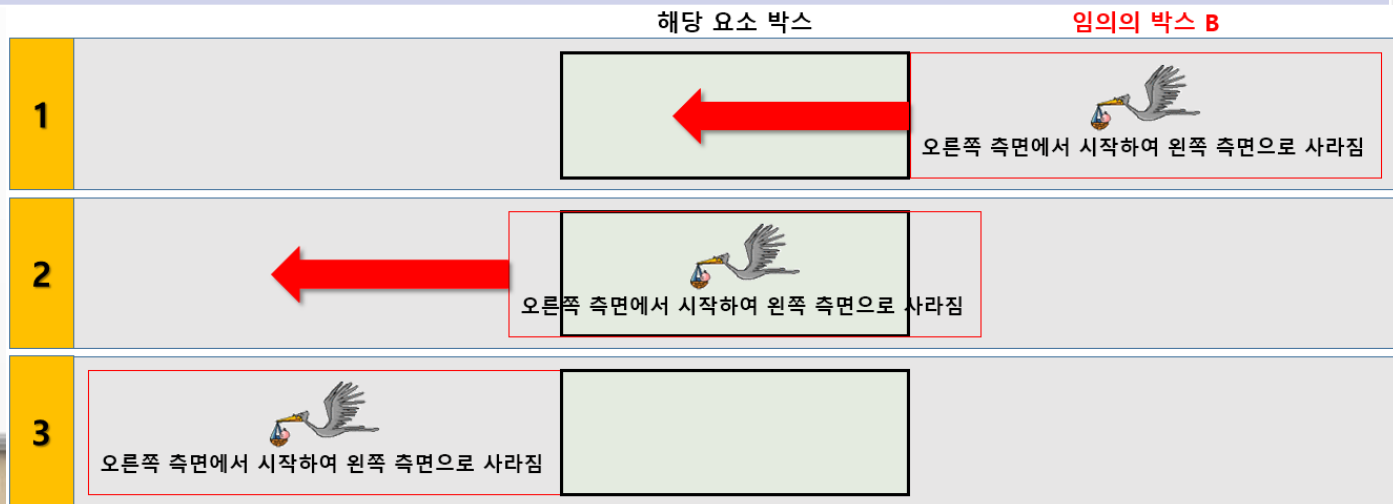
marquee-style - 마키 스타일 지정하기

CSS3

- 요소의 모든 내용을 임의의 박스로 생성한 후에 해당 요소 박스의 바깥으로부터 스크롤링하여 움직이도록 지정

속성 값	scroll slide alternate					
속성 특징	초기값	scroll	적용	overflow와 동일	상속	X
JavaScript	<code>object.style.marqueeStyle = "slide"</code>					
사용 예제	<code>div { marquee-style: slide }</code>					

속성 값	설명
<i>scroll</i>	임의의 박스 B(모든 콘텐츠)가 요소 박스 ①오른쪽 측면으로 시작하여 ②요소 박스로 움직이고 ③요소 박스의 왼쪽 측면으로 보이지 않을 때까지 스크롤된다. 이러한 스크롤은 반복적으로 이루어질 수 있다.



■ 마키 스타일 지정

<div>✓</div> 속성	marquee-style - 마키 스타일 지정하기	CSS3
-----------------	-----------------------------	------

속성 값	설명
slide	임의의 박스 B(모든 콘텐츠)가 요소 박스 ①오른쪽 측면으로 시작하여 ②요소 박스로 움직이고 ③요소 박스의 오른쪽 측면과 임의의 박스B의 오른쪽 측면이 일치할 때까지 스크롤된다. 콘텐츠의 일부는 요소 박스의 왼쪽으로 넘침(Overflow)이 일어난다.





마키 스타일 지정



속성

marquee-style - 마키 스타일 지정하기

CSS3

속성 값	설명
alternate	임의의 박스 B(모든 콘텐츠)가 요소 박스 ①오른쪽 측면으로 시작하여 ②요소 박스로 움직이고 ③요소 박스의 오른쪽 측면과 임의의 박스B의 오른쪽 측면이 일치할 때까지 첫 번째로 스크롤이 마무리된다. ④다시 반대 방향으로 요소 박스로 스크롤되고 ⑤임의의 박스B의 왼쪽 측면과 요소 박스 왼쪽 측면이 일치할 때까지 두 번째로 스크롤이 마무리가 된다. 콘텐츠의 일부는 요소 박스의 오른쪽으로 넘침(Overflow)이 일어난다. 이러한 스크롤은 반복적으로 이루어질 수 있다.




■ 마키 특성 지정

<div>✓</div> 속성	marquee-play-count – 요소 내용의 움직임 횟수 지정	CSS3
-----------------	---------------------------------------	------

속성 값	<non-negative-integer> infinite					
속성 특징	초기값	1	적용	overflow와 동일	상속	X
JavaScript	<i>object.style.marqueePlayCount</i> = "infinite"					
사용 예제	div { marquee-play-count : infinite }					

속성 값	설명
non-negative-integer	움직임의 횟수를 양수로 지정한다. 음수 값은 지정할 수 없다.
infinite	움직임의 횟수를 무한 반복으로 지정한다.

마키 특성 지정



속성

marquee-direction – 요소 내용의 움직이는 방향을 지정

CSS3

속성 값	forward reverse				
속성 특징	초기값	reverse	적용	overflow와 동일	상속 X
JavaScript	<code>object.style.marqueeDirection = "forward"</code>				
사용 예제	<code>div { marquee-direction: forward }</code>				

block-progression	direction	reverse	forward
tb (위쪽 → 아래쪽)	ltr (왼쪽 → 오른쪽)	left (왼쪽)	right (오른쪽)
	rtl (오른쪽 → 왼쪽)	right (오른쪽)	left (왼쪽)
rl / lr (오른쪽 ↔ 왼쪽)	ltr (왼쪽 → 오른쪽)	up (위쪽)	down (아래쪽)
	rtl (오른쪽 → 왼쪽)	down (아래쪽)	up (위쪽)



오른쪽에서 왼쪽으로 이동



reverse




왼쪽에서 오른쪽으로 이동




forward

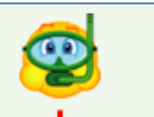
block-progression: tb; direction: ltr




위쪽



reverse



아래쪽



forward

block-progression: rl/lr; direction: ltr



마키 특성 지정



속성

marquee-speed - 요소 내용의 움직임 속도 지정

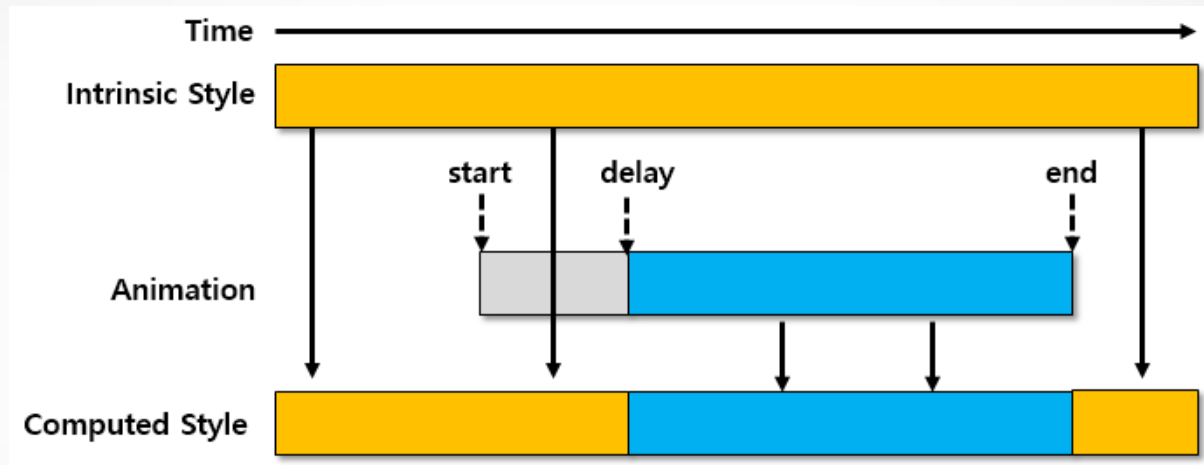
CSS3

속성 값	slow normal fast				
속성 특징	초기값	normal	적용	overflow와 동일	상속
JavaScript	<code>object.style.marqueeSpeed = "fast"</code>				
사용 예제	<code>div { marquee-speed: fast }</code>				

속성 값	설명
slow	콘텐츠 움직임의 속도를 느리게 지정한다.
normal	콘텐츠 움직임의 속도를 보통으로 지정한다.
fast	콘텐츠 움직임의 속도를 빠르게 지정한다.

표준화 문서	CSS Animation Level 1 - http://dev.w3.org/csswg/css-animations/
표준화 단계	W3C Editor's Draft, (2014-05-28)

■ 애니메이션 효과 적용 시점




- 애니메이션에서의 키프레임(keyframe)은 애니메이션을 구현할 때 지정된 애니메이션의 동작이 시작되는 특정 시점으로, 애니메이션 중간 중간에 요소(객체)의 위치와 크기 등이 자동으로 구현된다.
- @keyframes는 animation 속성에 적용할 키프레임을 생성하기 위한 규칙

```

@keyframes 애니메이션 이름 {
  from { [ CSS 속성: 값; ] }
  [ percentage ] { [ CSS 속성: 값; ] }
  to { [ CSS 속성: 값; ] }
}
    
```

애니메이션 효과 적용 시점

예제	Animation.html
<pre>#myDIV1 { width: 100px; height: 100px; margin: 20px; border-radius:10px; animation: myDIV1 5s infinite linear; }</pre>	<pre>@-webkit-keyframes myDIV1 { from { background-color: blue; } 50% { transform: scale(1.3, 1.3); } to { background-color: red;} }</pre>
<div>  </div>	

애니메이션 속성

<div> <div></div> <div>속성</div> </div>	animation-name – 애니메이션 이름 지정하기	CSS3
----------------------------------------	--------------------------------	------

- @keyframes 규칙으로 생성한 애니메이션 이름을 지정하여 해당 애니메이션이 실행되도록 하는 속성

속성 값	<single-animation-name># < <i>single-animation-name</i> > = none <custom-ident>				
속성 특징	초기값	none	적용	모든 요소들 제외(table-column-group, table-column)	상속
JavaScript	object.style.animationName= "myName"				
사용 예제	div { animation-name: myName }				

속성 값	설명
<i>none</i>	키프레임이 지정되지 않은 것으로 처리되기 때문에 애니메이션 효과가 없다.
custom-ident	지정된 이름으로 키프레임을 사용하도록 지정한다. 애니메이션이 여러 개인 경우에는 애니메이션 이름을 콤마(,)로 구분하여 지정할 수 있다.

애니메이션 속성

<div> <div></div> <div>속성</div> </div>	animation-duration – 애니메이션 진행 시간 지정하기	CSS3
----------------------------------------	---------------------------------------	------

진행되는 애니메이션의 진행 시간을 지정

속성 값	<time>#				
속성 특징	초기값	0s	적용	모든 요소들 제외(table-column-group, table-column)	상속
JavaScript	<i>object.style.animationDuration</i> = "5s"				
사용 예제	div { animation-duration : 5s }				

<div> <div></div> <div>속성</div> </div>	animation-timing-function – 애니메이션 진행 속도 지정하기	CSS3
----------------------------------------	----------------------------------------------	------

애니메이션의 진행 속도의 변화를 지정

속성 값	ease linear ease-in ease-out ease-in-out step-start step-end steps(<integer>[, [start end]]?) cubic-bezier(<number>, <number>, <number>, <number>)				
속성 특징	초기값	ease	적용	모든 요소들 제외(table-column-group, table-column)	상속
JavaScript	<i>object.style.animationTimingFunction</i> = "ease-in-out"				
사용 예제	div { animation-timing-function : ease-in-out }				

애니메이션 속성

<div> <div></div> <div>속성</div> </div>	animation-iteration-count – 애니메이션 반복 횟수 지정하기	CSS3
----------------------------------------	----------------------------------------------	------

속성 값	infinite <number>				
속성 특징	초기값	1	적용	모든 요소들 제외(table-column-group, table-column)	상속
JavaScript	<i>object.style.animationIterationCount</i> = "infinite"				
사용 예제	div { animation-iteration-count : infinite }				

속성 값	설명
infinite	애니메이션이 무한 반복되도록 지정한다.
number	애니메이션이 반복될 횟수를 지정한다. 음수 값은 허용되지 않으며, 정수가 아닌 실수 값을 지정하면 마지막 사이클 도중에 종료된다.

애니메이션 속성

<div> <div></div> <div>속성</div> </div>	animation-direction – 애니메이션 진행 방향 지정하기	CSS3
----------------------------------------	----------------------------------------	------

속성 값	normal reverse alternate alternate-reverse				
속성 특징	초기값	normal	적용	모든 요소들 제외(table-column-group, table-column)	상속
JavaScript	<code>object.style.animationDirection= "alternate"</code>				
사용 예제	div { animation-direction : alternate }				

속성 값	설명
<i>normal</i>	애니메이션의 진행 방향을 순방향으로 지정한다.
<i>reverse</i>	애니메이션의 진행 방향을 역방향으로 지정한다.
<i>alternate</i>	애니메이션의 진행 방향을 양방향으로 지정한다. 만일 애니메이션 반복 횟수가 여러 번 지정된 경우에는 홀수 번째는 순방향으로 진행되고 짝수 번째는 역방향으로 진행된다.
<i>alternate-reverse</i>	애니메이션의 진행 방향을 양방향으로 지정한다. 만일 애니메이션 반복 횟수가 여러 번 지정된 경우에는 홀수 번째는 역방향으로 진행되고 짝수 번째는 순방향으로 진행된다.

애니메이션 속성

<div> <div></div> <div>속성</div> </div>	animation-play-state – 애니메이션 진행 상태 지정하기	CSS3
----------------------------------------	-----------------------------------------	------

속성 값	running paused				
속성 특징	초기값	running	적용	모든 요소들 제외(table-column-group, table-column)	상속
JavaScript	<code>object.style.animationPlayState= "paused"</code>				
사용 예제	div { animation-play-state : paused }				

속성 값	설명
<i>running</i>	애니메이션을 진행 상태로 지정하며, 이 값으로 지정되어 있는 동안은 애니메이션이 정상적으로 진행된다.
<i>paused</i>	애니메이션을 정지 상태로 지정하며, 이 값으로 지정되어 있는 동안은 애니메이션이 일시 정지되며 진행 시간 또한 일시 정지 된다.

애니메이션 속성

<div> <div></div> <div>속성</div> </div>	animation-fill-mode – 애니메이션 효과 여부 지정하기	CSS3
----------------------------------------	----------------------------------------	------

- 애니메이션의 실행 이전이나 이후에 애니메이션 효과의 표시 여부를 지정

속성 값	none forwards backwards both				
속성 특징	초기값	none	적용	모든 요소들 제외(table-column-group, table-column)	상속
JavaScript	<code>object.style.animationFillMode= "forwards"</code>				
사용 예제	div { animation-fill-mode : forwards }				

속성 값	설명
<i>none</i>	애니메이션의 효과가 없도록 지정
<i>forwards</i>	애니메이션 실행이 종료된 후에 마지막 키프레임에 선언된 CSS 속성을 표시하도록 지정
<i>backwards</i>	애니메이션이 시작하기 전 또는 지연 시간에 첫 번째 키프레임에 선언된 CSS 속성을 표시하도록 지정
<i>both</i>	양쪽 forwards 및 backwards 에 적용되도록 지정

애니메이션 속성

<div> <div></div> <div>속성</div> </div>	animation-delay – 애니메이션 지연 시간 지정하기	CSS3
----------------------------------------	------------------------------------	------

- 애니메이션이 시작할 때의 지연 시간을 지정


속성 값	<time>#				
속성 특징	초기값	0s	적용	모든 요소들 제외(table-column-group, table-column)	상속
JavaScript	<i>object.style.animationDelay</i> = "3s"				
사용 예제	div { animation-delay : 3s }				

<div> <div></div> <div>속성</div> </div>	animation – 애니메이션 효과 지정하기	CSS3
----------------------------------------	---------------------------	------

- 지금까지 살펴본 애니메이션 관련 속성을 한번에 지정

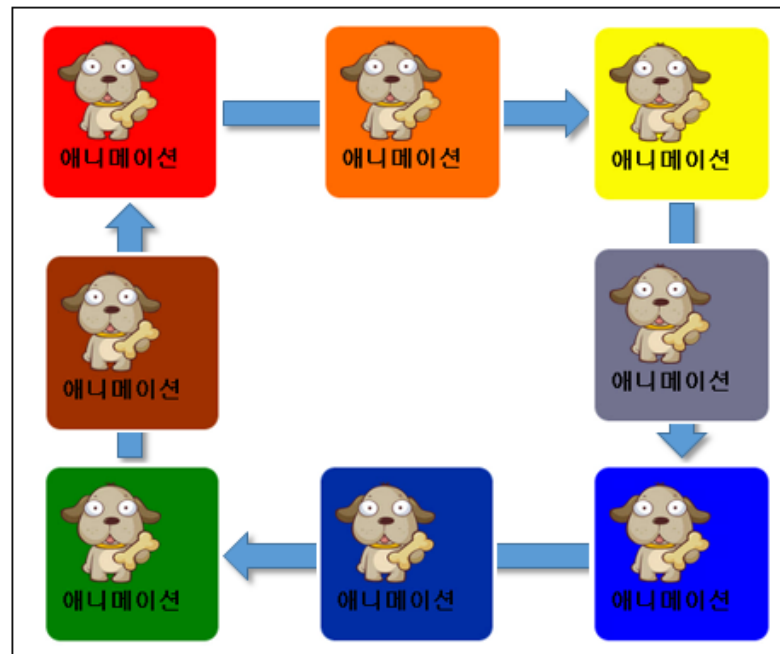
속성 값	<time> <single-timing-function> <time> <single-animation-iteration-count> <single-animation-direction> <single-animation-fill-mode> <single-animation-play-state> <single-animation-name>				
속성 특징	초기값	개별 속성값 참조	적용	모든 요소들 제외(table-column-group, table-column)	상속
JavaScript	<i>object.style.animation</i> = "myAnimation 5s ease-in 2s infinite alternate forwards"				
사용 예제	div { animation : myAnimation 5s ease-in 2s infinite alternate forwards }				

애니메이션 속성

예제	CSS3_09-05_Animation.html
<pre> #myDIV2 { width: 100px; height: 100px; margin: 20px; border-radius:10px; background:#92B901; color:#ffffff; position:relative; animation: myDIV2 5s infinite } </pre>	<pre> @-webkit-keyframes myDIV2 { 0% {transform: rotate(0deg); left:0px;} 25% {transform: rotate(20deg); left:0px;} 50% {transform: rotate(0deg); left:500px;} 55% {transform: rotate(0deg); left:500px;} 70% {transform: rotate(0deg); left:500px; background:#1ec7e6;} 100% {transform: rotate(-360deg); left:0px;} } </pre>
<div>  </div>	

애니메이션 속성

예제	CSS3_09-05_Animation.html
<pre>#myDIV3 { width: 100px; height: 100px; margin: 20px; border-radius:10px; background: red; position: relative; animation: myDIV3 5s infinite }</pre>	<pre>@-webkit-keyframes myDIV3 { from, to {background:red; left:0px; top:0px;} 25% {background:yellow; left:200px; top:0px;} 50% {background:blue; left:200px; top:200px;} 75% {background:green; left:0px; top:200px;} }</pre>



애니메이션 속성

예제

CSS3_09-05_Animation.html

```
#myDIV4 {
```

```
background: red; position: relative;
```

```
animation-name:myDIV4;
```

```
animation-duration:5s;
```

```
animation-timing-function:linear;
```

```
animation-delay:2s;
```

```
animation-iteration-count:infinite;
```

```
animation-direction:alternate;
```

```
animation-play-state:running;
```

```
}
```

```
@-webkit-keyframes myDIV4 {
```

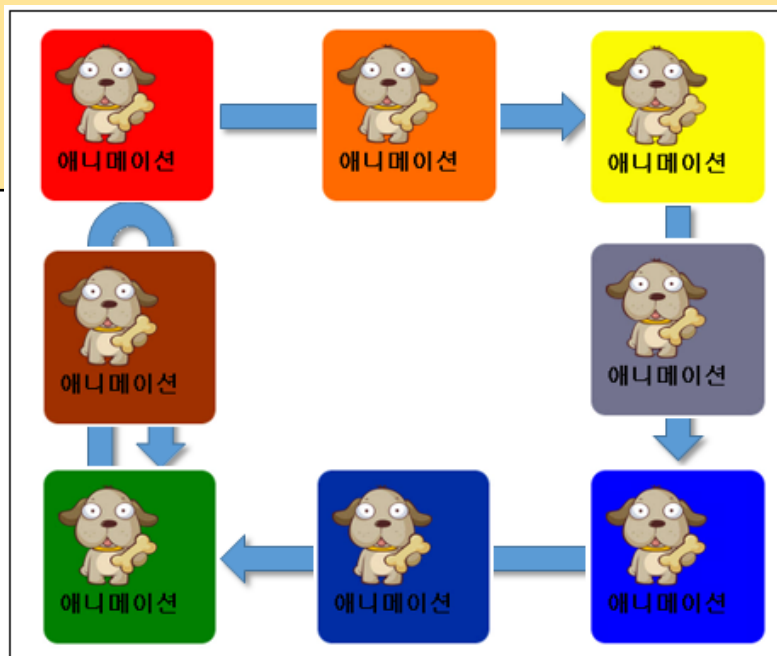
```
from, to {background:red; left:0px; top:0px;}
```

```
25% {background:yellow; left:200px; top:0px;}
```

```
50% {background:blue; left:200px; top:200px;}
```

```
75% {background:green; left:0px; top:200px;}
```

```
}
```





표준화 문서	Media Queries Level 3 - http://www.w3.org/TR/css3-mediaqueries/
표준화 단계	W3C Recommendation, (2012-06-19)

- 미디어 쿼리는 스타일 시트의 좀 더 정확한 표시를 허용함으로써 미디어 유형의 기능을 확장할 수 있기 때문에 반응형 웹의 구현할 때 아주 유용한 기능

■ 미디어 쿼리 사용의 배경

```
<link rel="stylesheet" type="text/css" media="screen" href="sans-serif.css">
<link rel="stylesheet" type="text/css" media="print" href="serif.css">
```

```
@media screen {
  * { font-family: sans-serif; }
}
```

미디어 타입	의도	미디어 타입	의도
all	모든 장치들	print	인쇄 장치
aural	음성 합성 장치(speech synthesizer)	projection	프로젝터 장치
braille	점자 표시 장치	screen	컴퓨터 화면 장치
embossed	양각(종이에 인쇄된 점자를 위한 의도)	tty	디스플레이 능력이 한정된, teletype
handheld	휴대 기기(전형적으로 작은 스크린)	ty	영상과 음성이 동시 출력(텔레비전 등)



■ 미디어 쿼리의 사용



link 요소를 이용하는 방법

- 스타일시트 파일을(sans-serif.css와 serif.css) 선택적으로 사용

다음은 특정 미디어(컬러 모니터)에 특정 스타일시트(example.css)를 적용하기 위한 예를 나타낸 것이다.

```
<link rel="stylesheet" media="screen and (color)" href="example.css" />
```

```
@import url(color.css) screen and (color);
```



style 요소의 media 속성 사용

- 미디어 유형에 따라서 선택적으로 스타일을 적용

다음은 media 속성에 따라서 style 구간을 선택적으로 적용하기 위한 예를 나타낸 것이다.

```
<style media="screen">
  p { font-family: sans-serif; font-size: 14px }
</style>
<style media="print">
  p { font-family: serif; font-size: 12px }
</style>
```



■ 미디어 쿼리의 사용



@media 규칙의 사용

■ 특정 미디어 유형에만 선택적으로 CSS를 적용

다음은 특정 미디어(프린터)에 특정 스타일을 적용하기 위한 예를 나타낸 것이다.

```
<style>
  @media screen { p { font-family: verdana, sans-serif; font-size: 14px } }
  @media print { p { font-size: 20px; color:red } }
  @media screen, print { p { font-weight: bold } }
</style>
```

미디어 유형이 키워드 'all'인 경우, 명시적으로 지정하지 않으면 all 이 되기 때문에, 다음의 예 2가지 모두 동일한 방법을 나타낸다.

```
@media all and (min-width: 500px) { ... }
```

```
@media (min-width: 500px) { ... }
```

```
@media (orientation: portrait) { ... }
```

```
@media all and (orientation: portrait) { ... }
```



■ 미디어 쿼리의 문법 및 조건

```
@media only screen and (min-width : 480px) { ... }
```

논리적
not 및 only

미디어
유형

논리적
OR/AND

미디어 기능(속성)

미디어 값



논리적 OR 와 AND

- 미디어 쿼리 문법에서, 콤마는 논리적 OR, 'and' 키워드는 논리적 AND의 의미

다음은 CSS에서의 @media 규칙을 사용하여 콤마로 구분된 목록에서의 미디어 쿼리의 예를 나타낸 것으로, 컬러 화면 또는 컬러 프로젝터 장치에 스타일을 적용한다.

```
@media screen and (color), projection and (color) { ... }
```

다음은 화면의 폭이 최소 320픽셀 이상이고 최대 폭이 800픽셀 이하인 경우에만 해당 CSS가 적용되도록 한 예를 나타낸 것이다.

```
@media screen and (min-width: 320px) and (max-width: 800px) { ... }
```




■ 미디어 쿼리의 문법 및 조건



논리적 NOT 및 only

- 키워드 'not'은 논리적 NOT을 표현, 'only' 키워드는 지정한 미디어 유형일 때만 CSS를 적용

다음에서 ①은 화면이 아닌 미디어 유형일 경우에 해당이 되고, ②는 미디어 유형이 화면일 경우에만 해당이 되는 예제를 나타낸 것이다.

- | | |
|---|----------------------------|
| ① | @media not screen { ... } |
| ② | @media only screen { ... } |



■ 미디어 기능(Media Features)



width

■ 출력 장치의 대상 표시 영역의 폭을 정의하는 기능

값	<length>	미디어 유형	시각과 촉각 미디어	min/max 접두사 사용	yes
---	----------	--------	------------	----------------	-----

미디어 유형이 프린터이고 25cm보다 넓은 출력일 때, CSS가 사용될 수 있음을 나타낸 예제이다.

link	<link rel="stylesheet" media="print and (min-width: 25cm)" href="http://..." />
style	<style type="text/css" media="print and (min-width: 25cm)" >
@media	@media print and (min-width: 25cm) { ... }

뷰포트(문서가 렌더링되는 화면이나 종이의 일부) 폭이 400픽셀 ~ 700픽셀 사이의 장치에서 CSS가 사용될 수 있음을 나타낸 예제이다.

link	<link rel="stylesheet" media="screen and (min-width: 400px) and (max-width: 700px)" href="http://..." />
style	<style type="text/css" media="screen and (min-width: 400px) and (max-width: 700px)" >
@media	@media screen and (min-width: 400px) and (max-width: 700px) { ... }



■ 미디어 기능(Media Features)



height

- 출력 장치의 대상 표시 영역의 높이를 정의하는 기능

값	<length>	미디어 유형	시각과 촉각 미디어	min/max 접두사 사용	yes
---	----------	--------	------------	----------------	-----



device-width

- 출력 장치의 디스플레이(렌더링되는 표면)의 폭을 정의하는 기능

값	<length>	미디어 유형	시각과 촉각 미디어	min/max 접두사 사용	yes
---	----------	--------	------------	----------------	-----

가로로 800픽셀을 표시하는 화면에서만 CSS가 사용될 수 있음을 나타낸 예제이다.

link	<code><link rel="stylesheet" media="screen and (device-width: 800px)" href="http://..." /></code>
style	<code><style type="text/css" media="screen and (device-width: 800px)"></code>
@media	<code>@media screen and (device-width: 800px) { ... }</code>



■ 미디어 기능(Media Features)



device-height

- 출력 장치의 디스플레이(렌더링되는 표면)의 높이를 정의하는 기능

값	<length>	미디어 유형	시각과 촉각 미디어	min/max 접두사 사용	yes
---	----------	--------	------------	----------------	-----

세로로 600픽셀을 표시하는 화면에서만 CSS가 사용될 수 있음을 나타낸 예제이다.

link	<link rel="stylesheet" media="screen and (device-height: 600px)" href="http://..." />				
style	<style type="text/css" media="screen and (device-height: 600px)">				
@media	@media screen and (device-height: 600px) { ... }				



orientation

- 화면의 가로 및 세로 방향에 따라서 미디어의 유형을 결정하는 기능

값	portrait landscape	미디어 유형	비트맵 미디어	min/max 접두사 사용	no
---	----------------------	--------	---------	----------------	----

portrait(세로 방향 길이가 가로 방향 길이보다 큰)인 모든 미디어에서 CSS가 사용될 수 있음을 나타낸 예제

link	<link rel="stylesheet" media="all and (orientation: portrait)" href="http://..." />				
style	<style type="text/css" media="all and (orientation: portrait)">				
@media	@media all and (orientation: portrait) { ... }				



■ 미디어 기능(Media Features)



aspect-ratio

- 'width' 미디어 기능의 값과 'height' 미디어 기능 값의 비율을 정의하는 기능

값	<ratio>	미디어 유형	비트맵 미디어	min/max 접두사 사용	yes
---	---------	--------	---------	----------------	-----



device-aspect-ratio

- 'device-width' 미디어 기능의 값과 'device-height' 미디어 기능 값의 비율을 정의하는 기능

값	<ratio>	미디어 유형	비트맵 미디어	min/max 접두사 사용	yes
---	---------	--------	---------	----------------	-----

화면의 가로/세로 픽셀 수가 16:9의 비율을 가진 장치에 CSS가 사용될 수 있음을 나타낸 예제이다.

link	<link rel="stylesheet" media="screen and (device-aspect-ratio: 16/9)" href="http://..." />
style	<style type="text/css" media="screen and (device-aspect-ratio: 16/9)">
@media	<pre>@media screen and (device-aspect-ratio: 16/9) { ... } @media screen and (device-aspect-ratio: 32/18) { ... } @media screen and (device-aspect-ratio: 1280/720) { ... } @media screen and (device-aspect-ratio: 2560/1440) { ... }</pre>



■ 미디어 기능(Media Features)



color

- 출력 장치의 색 성분에 할당되는 비트 수(색상의 깊이)를 나타내는 것으로, 색상 표시가 가능한 장치의 미디어 유형을 결정하는 기능

값	<integer>	미디어 유형	시각 미디어	min/max 접두사 사용	yes
---	-----------	--------	--------	----------------	-----

다음 2가지의 경우, 컬러 장치에 CSS가 사용될 수 있음을 나타낸 예제이다.

link	<code><link rel="stylesheet" media="all and (color)" href="http://..." /></code>
style	<code><style type="text/css" media="all and (color)"></code>
@media	<code>@media all and (color) { ... }</code> <code>@media all and (min-color: 1) { ... }</code>

다음은 색 성분당 2비트 이상인 컬러 장치에 CSS가 사용될 수 있음을 나타낸 예제이다.

@media	<code>@media all and (min-color: 2) { ... }</code>
--------	----------------------------------------------------



■ 미디어 기능(Media Features)



color-index

- 출력 장치의 색상 룩업 테이블에 들어있는 전체 수(색상 수)로 미디어 유형을 결정하는 기능

값	<integer>	미디어 유형	시각 미디어	min/max 접두사 사용	yes
---	-----------	--------	--------	----------------	-----

다음 2가지의 경우, 모든 컬러 인덱스 장치에 CSS가 사용될 수 있음을 나타낸 예제이다.

link	<code><link rel="stylesheet" media="all and (color)" href="http://..." /></code>
style	<code><style type="text/css" media="all and (color)"></code>
@media	<code>@media all and (color-index) { ... }</code> <code>@media all and (min-color-index: 1) { ... }</code>

다음은 화면이 256가지의 색상을 표시할 수 있는 경우에 CSS가 사용될 수 있음을 나타낸 예제이다.

@media	<code>@media screen and (min-color-index: 256) { ... }</code>
--------	---------------------------------------------------------------



■ 미디어 기능(Media Features)



monochrome

- 흑백 프레임 버퍼에 있는 각 픽셀에 할당되는 비트 수(흑백의 깊이, 음영)로 미디어 유형을 결정하는 기능

값	<integer>	미디어 유형	시각 미디어	min/max 접두사 사용	yes
---	-----------	--------	--------	----------------	-----

다음 2가지의 경우, 흑백 장치에 CSS가 사용될 수 있음을 나타낸 예제이다.

link	<code><link rel="stylesheet" media="all and (monochrome)" href="http://..." /></code>
style	<code><style type="text/css" media="all and (monochrome)"></code>
@media	<code>@media all and (monochrome) { ... }</code> <code>@media all and (min- monochrome: 1) { ... }</code>

다음은 픽셀당 2비트 이상인 흑백 장치에 CSS가 사용될 수 있음을 나타낸 예제이다.

@media	<code>@media all and (min- monochrome: 2) { ... }</code>
--------	----------------------------------------------------------



■ 미디어 기능(Media Features)



resolution

■ 출력 장치의 해상도(화소의 밀도)로 미디어의 유형을 결정하는 기능

- ✓ min-resolution : 밀도가 가장 작은 값과 지정된 값을 비교해야 한다.
- ✓ max-resolution : 밀도가 가장 큰 값과 지정된 값을 비교해야 한다.

값	<resolution>	미디어 유형	시각 미디어	min/max 접두사 사용	yes
---	--------------	--------	--------	----------------	-----

다음은 인치당 300도트 보다 큰 해상도를 가진 장치에서 CSS가 사용될 수 있음을 나타낸 예제이다.

link	<code><link rel="stylesheet" media="print and (min-resolution: 300dpi)" href="http://..." /></code>
style	<code><style type="text/css" media="print and (min-resolution: 300dpi)"></code>
@media	<code>@media print and (min-resolution: 300dpi) { ... }</code>



■ 미디어 기능(Media Features)



scan

- TV 미디어의 화면 주사 방식에 따라서 미디어의 유형을 결정하는 기능

값	progressive interlace	미디어 유형	텔레비전(TV)	min/max 점두사 사용	no
---	-------------------------	--------	----------	----------------	----

다음은 TV 화면의 주사 방식이 순차 방식일 경우에 CSS가 사용될 수 있음을 나타낸 예제이다.

link	<link rel="stylesheet" media="tv and (scan: progressive)" href="http://..." />
style	<style type="text/css" media="tv and (scan: progressive)">
@media	@media tv and (scan: progressive) { ... }



grid

- 출력 장치가 grid(점자 출력기와 같은 물리적 격자 형태)인지 비트맵인지를 결정하는 기능



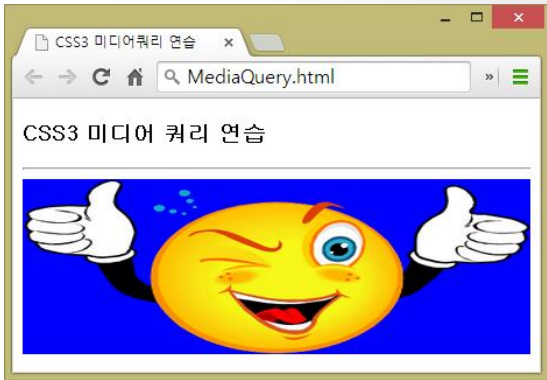
값	<integer>	미디어 유형	시각과 촉각 미디어	min/max 점두사 사용	no
---	-----------	--------	------------	----------------	----

다음은 대상 표시 영역의 폭이 최대 15em보다 작은 격자 기반의 휴대용 장치에 CSS가 사용될 수 있음을 나타낸 예제이다.

link	<link rel="stylesheet" media="handheld and (grid) and (max-width: 15em)" href="http://..." />
style	<style type="text/css" media="handheld and (grid) and (max-width: 15em)">
@media	@media handheld and (grid) and (max-width: 15em) { ... }



미디어 기능(Media Features)

예제	CSS3_09-06_MediaQuery.html		
Style	<div>div { width: 100%; height: 100% } @media screen and (max-width: 300px) { div { background-color: red } } @media screen and (min-width: 300px) and (max-width: 450px) { div { background-color: green } } @media screen and (min-width: 450px) { div { background-color: blue } }</div>		
Body	<div> </div>		
<div><div></div><div></div><div></div></div>			
width(150px 미만)		width(150px ~ 300px)	width(300px 이상)



다음 학습



- 1 캔버스(Canvas) 기본 내용
- 2 사각형 그리기
- 3 선 그리기