

# 웹킷 CSS 바이블 때문

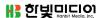
서미연, 강영주, 김대현, 김영호, 김훈민, 민경환, 양주희, 정다운, 조은 지음





# 웹킷 CSS 바이블 때문

서미연, 강영주, 김대현, 김영호, 김훈민, 민경환, 양주희, 정다운, 조은 지음



# 웹킷 CSS 바이블 레퍼런스 편

**초판발행** 2015년 8월 25일

지은이 서미연, 강영주, 김대현, 김영호, 김훈민, 민경환, 양주희, 정다운, 조은 / **펴낸이** 김태현 **펴낸곳** 한빛미디어(주) / **주소** 서울시 마포구 양화로 7길 83 한빛미디어(주) IT출판부 전화 02-325-5544 / **팩스** 02-336-7124 등록 1999년 6월 24일 제10-1779호 ISBN 978-89-6848-767-5 15000 / **정가** 35.000원

총괄 배용석 / 책임편집 김창수 / 기획·편집 김상민 디자인 표지/내지 여동일, 조판 최송실

**마케팅** 박상용 / **영업** 김형진, 김진불, 조유미

이 책에 대한 의견이나 오탈자 및 잘못된 내용에 대한 수정 정보는 한빛미디어(주)의 홈페이지나 아래 이메일로 알려주십시오. 한빛미디어 홈페이지 www.hanbit.co.kr / 이메일 ask@hanbit.co.kr

Published by HANBIT Media, Inc. Printed in Korea

Copyright © 2015 서미연 & HANBIT Media, Inc.

이 책의 저작권은 서미연과 한빛미디어(주)에 있습니다.

저작권법에 의해 보호를 받는 저작물이므로 무단 복제 및 무단 전재를 금합니다.

지금 하지 않으면 할 수 없는 일이 있습니다.

책으로 펴내고 싶은 아이디어나 원고를 메일(ebookwriter@hanbit.co.kr)로 보내주세요. 한빛미디어(주)는 여러분의 소중한 경험과 지식을 기다리고 있습니다.

### 지은이 서미연

웹디자이너로 시작한 지 10년.

시대의 트렌드에 따라 저의 관심사는 웹이라는 한 울타리 안에서 계속 변화해왔고 현재 NHN Technology Services에서 프론트엔드 개발자로 일하고 있습니다. 〈네이 바와 함께하는 HTML5 개발 전략〉웹킷 CSS 강의 경력이 있으며, 기술을 공유하는 것을 좋아하지만, 부끄럼도 많은 수줍은 개발자입니다.

### 지은이 강영주

현재 NHN Technol ogy Servic es에서 프론트엔드 개발자로 일하고 있습니다.

#### 지은이 김대현

NHN Technology Services에서 프론트엔드 개발자로 일하고 있습니다. 웹 기술에 관심이 많고 코딩과 삽질을 즐기는 평범한 개발자입니다.

#### 지은이 김영호

탁 트인 자연과 쭉 뻗은 도로, 바람을 좋아하는 "바이크 라이더" 김영호입니다. 현재 NHN Technology Services에서 프론트엔드 개발자로 일하고 있습니다. 네이버 2012 대선, 2014 브라질월드컵, 건강, 게임 등 다양한 서비스를 담당해왔으며 현재는 네이버 지식쇼핑 개발을 맡고 있습니다.

웹기술에 관심이 많고 호기심 많은 근성있는 개발자 입니다.

#### 지은이\_ **김훈민**

자바 개발자로 입문해서 현재는 NHN Technology Services 프론트엔드 개발팀에서 Ajax UI 개발자로 근무하고 있습니다. http://huns.me 블로그를 운영하고 있으며, 항상 "왜?"라는 질문을 던지려 애쓰는 중입니다.

## 지은이\_ 민경환

NHN Technology Services에서 프론트엔드 개발자로 근무하고 있습니다. 마크업, 자바스크립트 등 프론트엔드 개발을 사랑하며 브라우저에서 일어날 수 있는 모든 일에 관심 두고 있습니다. 네이버에서 스마트에디터, 포스트, 네이버 캐스트 등의 개발을 담당했습니다. 프로젝트에서 얻은 경험을 바탕으로 반응형 웹, 플렉서블 레이아웃 등 사내/외 기술 전파에 힘쓰고 있습니다. 평소엔 사소한 개발 주제로 심각하게 토론하는 것과 맥주 마시는 것을 즐기며 마음의 정리가 필요할 땐 홀로 여행을 떠납니다.

#### 지은이 양주희

현재 NHN Technology Services에서 프론트엔드 개발자로 일하고 있습니다. 디 자이너에서 직업을 바꾼 지 2년째인 새내기 개발자이지만, 새로운 기술들을 사용하는데 큰 즐거움을 느낍니다.

## 지은이\_ 정다운

현재 NHN Technology Services에서 프론트엔드 개발자로 일하고 있습니다. 전 직장에서 디자이너, 액션스크립터로서의 경험을 통해 웹에서 그려지는 UI/UX에 관해 관심을 두게 되었고, 지금도 웹 기술의 여러 방면에 관심을 가지고 배우려고 노력 중입니다. 이 외에도 그림을 그리기나 우쿨렐레에도 관심이 많은 개발자입니다.

# 지은이\_ **조은**

현재 NHN Technology Services에서 프론트엔드 개발자로 일하고 있습니다. 웹 표준 커뮤니티 클리어보스 리더를 맡고 있으며, 한국 웹 표준 프로젝트 멤버로도 활동하고 있습니다. 치킨을 좋아하며 사람을 만나는 걸 좋아하고 UX에도 관심이 있어, 기술과 인간이 어떻게 더 밀접한 관계를 맺을 수 있을까 늘 고민하고 있습니다.

웹 개발자로 일한 지 5년이 돼 간다. PC 웹의 UI 마크업 업무를 주로 담당했지만, 몇 해 전부터 스마트폰의 보급으로 모바일 웹 시장이 형성되면서 모바일 웹 마크업 업무도 점점 늘어나고 있다.

모바일 웹 기술에 대한 관심이 점점 높아지면서 웹킷 CSS 속성도 점점 많이 사용하는 추세다. PC 웹에서는 제대로 활용할 수 없었던 웹킷 CSS 속성을 모바일 웹에서는 적 극적으로 사용하면서 마크업 업무의 효율성이 높아졌다. 한 사례로 마크업 개발자의 업무 중에는 성능 최적화를 위해 CSS 파일의 크기를 줄이는 작업이 있다. CSS 파일의 크기를 줄이는 작업이 있다. CSS 파일의 크기를 줄이는 작업은 필요하지 않은 속성을 삭제하는 것이다. 다양한 브라우저에 대 응하기 위해 같은 효과를 내는 CSS를 여러 줄 사용해야 할 때가 많은데 이런 불필요한 CSS를 웹킷 CSS 속성으로 줄여서 해결한 경우가 많았다. 그 외에도 대표적인 브라우저에 모두 대응해야 하는 PC 웹 서비스의 특성 때문에 적용할 수 없던 웹킷 CSS 속성을 모바일 웹에서는 다양하게 시도해 적용해 볼 수 있었다.

웹킷 CSS 속성에 더 많은 관심을 두기 시작했고 웹킷 CSS 속성을 공부하기 시작했다.국내에는 아직 웹킷 CSS 속성에 관한 자료가 부족해 웹킷 CSS 속성 레퍼런스를 한권의 책으로 정리한다면 좋겠다는 생각에 이 책을 쓰기로 했다. 웹킷 CSS가 나에게 주었던 다양한 경험과 편리함이 독자에게도 전해지기를 희망한다.

지은이\_ 서미연

웹은 지난 20년간 많은 변화를 겪으며 세계에서 가장 많이 사용하며 사랑받는 기술이다. 기술이 발전함에 따라 사람들의 요구는 커졌고, 그런 요구를 충족하기 위해 웹은다양한 변화와 발전을 거듭했다.

웹킷은 웹의 변화와 발전에 가장 빠르게 대응하고 진화하는 엔진으로 웹킷을 기반으로 제작한 블링크 엔진을 사용하는 크롬 브라우저는 전 세계적으로 압도적인 점유율을 보이고 있으며, 모바일에서는 웹킷을 빼놓고는 이야기하기 힘들 정도로 웹킷은 많은 영향력을 가지고 있다. 웹킷 CSS를 안다는 것은 단순히 '웹킷에만 기술을 도입한다'라는 것이 아닌, 표준을 주도하는 엔진인 웹킷에 적용되는 기술을 먼저 앎으로써 빠르게 변화하는 웹 기술에 대한 빠른 대응 능력 및 선도능력을 지닐 수 있게 한다. 또한, 웹킷은 오픈소스를 기반으로 하고 있으므로 노드 JS와 함께 사용하여 웹 기반의 멀티플랫폼 네이티브 게임 제작, 스마트 TV나 Watch 등에 내장된 브라우저의 엔진을 웹킷으로 사용하는 곳에서 웹 기반의 애플리케이션을 제작하는 것도 가능하게 한다.

웹킷은 앞으로도 표준을 선도하는 엔진으로서 꾸준히 발전해 나갈 것이다. 언젠가는 웹킷이 아닌 다른 브라우저 또한 표준을 준수하며 발전해나갈 것이고, 웹킷 CSS를 앎 으로써 미래에 대응할 수 있기를 바란다.

# 이 책에서 다루는 내용 ---

이 책에서는 웹킷 엔진 기반의 브라우저에서 사용할 수 있는 웹킷 CSS 속성을 다룬다.

### 책의 구성

약 113가지의 최신 웹킷 CSS 속성을 다룬다. 이 장에서 웹킷 CSS 속성의 정의, 지원 환경, 구문, 예제, 참고 속성 등을 다룬다.

### 책의 구조

- 이 책에서 다루는 예제 코드는 HTML5 표기 방법으로 작성됐다. 예제에 사용한 소스 코드와 이미지는 다음 주소에서 내려받을 수 있다.
  - 소스 코드와 이미지 다운로드: https://github.com/webkittf/webkitcss

각 속성에 있는 예제 코드 주소로 PC나 모바일에서 웹킷 CSS 속성이 어떻게 동작하는지 확인할 수 있다. 예제 코드는 오픈소스인 jsbin.com에서 작성했기 때문에 예제 코드 주소 뒤에 '/edit'를 입력하면 직접 코드를 수정해 테스트할 수 있다.

- 예제 코드 링크: http://jsbin.com/-webkit-animation
- 편집 모드 링크: http://jsbin.com/-webkit-animation/edit

예제 코드는 QR 코드도 함께 제공해 무료 QR 코드 리더기를 스마트폰에 설치하면 QR 코드를 통해 모바일에서 쉽게 예제를 볼 수 있다.

#### 코드 구조

예제 코드는 기본 HTML5 구조에 해당하는 코드는 생략하고 〈style〉 태그와 〈div〉 태그, 〈script〉 태그가 있는 부분만 표시한다.

```
<html>
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width,initial-</pre>
scale=1.0,minimum-scale=1.0,maximum-scale=1.0">
        <title>-webkit-</title>
        <style>
             div {
                 font-size: 14px:
             }
        </style>
    </head>
    <body>
        <div>Hello_world!</div>
        ⟨script⟩⟨script⟩
    </body>
</html>
```

# 지원 환경

지원 환경은 웹킷 엔진을 기반으로 동작하는 브라우저를 기준으로 다음 표에 있는 브라우저만 확인했다.

15이상	6		iOS	안드로이드
0	0	0	0	0

• 예제에 있는 속성 적용 결과는 크롬 39에서 확인한 화면이다. 웹킷 엔진은 애플 사파리에 처음 적용됐지만 이 책에서 다루는 웹킷 CSS 속성 가운데 일부는 사파리에서 정상적으로 작동하지 않는다. 그래서 사파리보다 브라우저 점유율도 높고 이 책의 웹킷CSS 속성이 모두 정상적으로 작동하는 크롬에서 예제를 확인했다.

# 먼저 알아야 할 것 ----

HTML5와 CSS에 관한 전반적인 지식과 자바스크립트, jQuery에 관한 초급 정도의 지식이 있어야 이 책의 전반적인 내용을 이해할 수 있다. 독자의 프로젝트에 적용한다면 더 쉽게 이해할 수 있을 것이다.

# 한빛 eBook 리얼타임

한빛 eBook 리얼타임은 IT 개발자를 위한 eBook입니다.

요즘 IT 업계에는 하루가 멀다 하고 수많은 기술이 나타나고 사라져 갑니다. 인 터넷을 아무리 뒤져도 조금이나마 정리된 정보를 찾기도 쉽지 않습니다. 또한, 잘 정리되어 책으로 나오기까지는 오랜 시간이 걸립니다. 어떻게 하면 조금이라 도 더 유용한 정보를 빠르게 얻을 수 있을까요? 어떻게 하면 남보다 조금 더 빨 리 경험하고 습득한 지식을 공유하고 발전시켜 나갈 수 있을까요? 세상에는 수 많은 종이책이 있습니다. 그리고 그 종이책을 그대로 옮긴 전자책도 많습니다. 전자책에는 전자책에 적합한 콘텐츠와 전자책의 특성을 살린 형식이 있다고 생 각합니다.

한빛이 지금 생각하고 추구하는, 개발자를 위한 리얼타임 전자책은 이렇습니다.

# eBook First -

# 빠르게 변화하는 IT 기술에 대해 핵심적인 정보를 신속하게 제공합니다

500페이지 가까운 분량의 잘 정리된 도서(종이책)가 아니라, 핵심적인 내용을 빠르게 전달하기 위해 조금은 거칠지만 100페이지 내외의 전자책 전용으로 개발한 서비스입니다. 독자에게는 새로운 정보를 빨리 얻을 기회가 되고, 자신이 먼저 경험한 지식과 정보를 책으로 펴내고 싶지만 너무 바빠서 엄두를 못 내는 선배, 전문가, 고수 분에게는 좀 더 쉽게 집필할 수 있는 기회가 될 수 있으리라 생각합니다. 또한, 새로운 정보와 지식을 빠르게 전달하기 위해 O'Reilly의 전자책 번역 서비스도 하고 있습니다.

# 무료로 업데이트되는 전자책 전용 서비스입니다

2 종이책으로는 기술의 변화 속도를 따라잡기가 쉽지 않습니다. 책이 일정 분량 이상으로 집필되고 정리되어 나오는 동안 기술은 이미 변해 있습니다. 전자책으로 출간된 이후에도 버전 업을 통해 중요한 기술적 변화가 있거나 저자(역자)와 독자가 소통하면서 보완하여 발전된 노하우가 정리되면 구매하신 분께 무료로 업데이트해 드립니다.

목자의 편의를 위해 DRM-Free로 제공합니다

구매한 전자책을 다양한 IT 기기에서 자유롭게 활용할 수 있도록 DRM-Free PDF 포맷으로 제공합니다. 이는 독자 여러분과 한빛이 생각하고 추구하는 전자책을 만들어 나가기 위해 독자 여러분이 언제 어디서 어떤 기기를 사용하더라도 편리하게 전자책을 볼 수 있도록 하기 위함입니다.

✔ 전자책 환경을 고려한 최적의 형태와 디자인에 담고자 노력했습니다

종이책을 그대로 옮겨 놓아 가독성이 떨어지고 읽기 어려운 전자책이 아니라, 전자책의 환경에 가능한 한 최적화하여 쾌적한 경험을 드리고자 합니다. 링크 등의 기능을 적극적으로 이용할 수 있음은 물론이고 글자 크기나 행간, 여백 등을 전자책에 가장 최적화된 형태로 새롭게 디자인하였습니다.

앞으로도 독자 여러분의 충고에 귀 기울이며 지속해서 발전시켜 나가도록 하겠습니다.

지금 보시는 전자책에 소유 권한을 표시한 문구가 없거나 타인의 소유권한을 표시한 문구가 있다면 위법하게 사용하고 있을 가능성이 큽니다. 이 경우 저작권법에 따라 불이익을 받으실 수 있습니다.

다양한 기기에 사용할 수 있습니다. 또한, 한빛미디어 사이트에서 구매하신 후에는 횟수에 관계없이 내려받으실 수 있습니다.

한빛미디어 전자책은 인쇄, 검색, 복사하여 붙이기가 가능합니다.

전자책은 오탈자 교정이나 내용의 수정·보완이 이뤄지면 업데이트 관련 공지를 이메일로 알려 드리며, 구매하신 전자책의 수정본은 무료로 내려받으실 수 있습니다.

이런 특별한 권한은 한빛미디어 사이트에서 구매하신 독자에게만 제공되며, 다른 사람에게 양도나 이전은 허락되지 않습니다.

Part 1	Animation —— 001	
	-webkit-animation: 애니메이션을 설정한다 003	
	-webkit-animation-delay: 애니메이션 지연 시간을 설정한다 007	
	,	
	-webkit-animation-direction: 애니메이션 재생 방향을 설정한다 <b>01</b>	
	-webkit-animation-duration: 애니메이션 재생 시간을 설정한다 015	,
	-webkit-animation-fill-mode: 애니메이션 시작과 끝 스타일을 설정한다 01	7
	-webkit-animation-iteration-count: 애니메이션을 무한으로 재생한다 024	
	-webkit-animation-name: 애니메이션 이름을 설정한다 027	
	-webkit-animation-play-state: 애니메이션에 정지와 재생 기능을 추가한다	
	029	
	-webkit-animation-timing-function: 애니메이션 재생 속도를 제어한다	
	031	
	@-webkit-keyframes: 스크립트 없이 애니메이션을 재생한다	
Part 2	Transition —— 043	
	-webkit-transition: 요소에 마우스 포인터를 올렸을 때 애니메이션을 실행한다	
	-webkit-transition-delay: transition 애니메이션에서 지연 시간을 설정한다 <b>048</b>	
	,	,
	-webkit-transition-duration: transition 애니메이션에서 애니메이션 시간을 설정한다	
	050	
	-webkit-transition-property: transition 애니메이션에서 변경할 속성을 설정한다	
	052	
	-webkit-transition-timing-function: transition 애니메이션에서 재생 속도를 설정한다	라
	063	

# Part 3 Flexible — 067

-webkit-box-align: 예전 방식 레이아웃 상자의 자식 요소를 정렬한다	069
-webkit-box-direction: 예전 방식 레이아웃 상자의 자식 요소를 배치하는 방법을 설정	영한다
081	
-webkit-box-flex: 예전 방식 레이아웃 상자의 자식 요소에 flex 속성을 설정한다	087
-webkit-box-flex-group: 예전 방식 레이아웃 상자의 자식 요소를 그룹으로 묶는다	
094	
-webkit-box-lines: 예전 방식 레이아웃 상자의 줄바꿈을 설정한다	098
-webkit-box-ordinal-group: 예전 방식 레이이웃 상자의 자식 요소 순서를 설정한다	4
099	
-webkit-box-orient: 예전 방식 레이아웃 상자의 레이아웃 방향을 설정한다	102
-webkit-box-pack: 예전 방식 레이아웃 상자의 자식 요소 배치 방법을 설정한다	106
-webkit-flex: 유연한 상자의 레이아웃을 적용한다 <b>117</b>	
-webkit-flex-basis: 유연한 상자의 레이아웃 기준을 설정한다 <b>1</b>	22
-webkit-flex-direction: 유연한 상자에서 자식 요소의 배치 방향과 순서를 변경한다	
126	
-webkit-flex-flow: -webkit-flex-direction 속성과 -webkit-flex-wrap 속성을 하나로 연	설정한다
132	
-webkit-flex-grow: 유연한 상자에서 자식 요소가 부모 요소에서 맞춰 늘어나게 설정한	다
139	
-webkit-flex-shrink: 유연한 상자에서 자식 요소가 부모 요소에 맞춰 줄어들게 설정한	<b>라</b>
142	
-webkit-flex-wrap: 유연한 상자에서 줄바꿈을 설정한다 <b>146</b>	
-webkit-align-content: 유연한 상자의 자식 요소를 배치하는 방법을 설정한다	
151	
-webkit-align-items : 유연한 상자의 자식 요소를 정렬한다 <b>16</b>	32

	-webkit-align-self: 유연한 상자의 자식 요소를 정렬한다	178
	-webkit-justify-content: 유연한 상자에서 자식 요소의 정렬을 변경한다	185
	-webkit-order: 유연한 상자에서 자식 요소의 순서를 지정한다	197
Part <b>4</b>	Background ——— 199	
	-webkit-background-clip: 배경을 콘텐츠 영역에만 표시한다	201
	-webkit-background-composite: 배경 이미지와 배경색을 합성한다	204
	-webkit-background-origin: background-position 속성이 아닌 다른 속성으로 배경의 위	치를 변경한다
	209	
	-webkit-background-size: 반응형 웹에서 자동으로 배경 이미지의 크기를 조	절한다
	213	
Part 5	Box ——— 225	
	-webkit-box-reflect: CSS 속성만으로 반사효과를 만든다	227
	-webkit-box-shadow: CSS 속성으로 그림자를 넣는다	233
	-webkit-box-sizing: 반응형웹에서 요소의 너비를 측정하는 기준을 정의한다	238
Part 6	Border —— 241	
	-webkit-border-radius: CSS 속성만으로 둥근 모서리를 만든다	243
	-webkit-border-top-left-radius: CSS 속성만으로 둥근 모서리를 만든다	250
	-webkit-border-top-right-radius: CSS \( \frac{45}{25} \) 2 441 \( \frac{15}{25} \) 2 441 \( \frac{15}{25} \) 2 57 2 441 \( \frac{15}{25} \) 2 57 2 57 2 57 2 57 2 57 2 57 2 57 2 5	252
	-webkit-border-bottom-left-radius: CSS 속성만으로 등근 모서리를 만든다	252
	-webkit-border-bottom-right-radius: CSS속성만으로 등근 모시리를 만든다	256
	-webkit-border-image: 이미지로 태두리를 만든다 25(	
	Wooding Soldon initiago.	•

Part <b>7</b>	Table ——— 265	
	-webkit-border-horizontal-spacing: 테이블에서 셀의 좌우 여백을 설정한다	267
	-webkit-border-vertical-spacing: 테이블에서 셀의 위아래 여백을 설정한다	
	270	
Part 8	Columns —— 275	
	-webkit-columns: 다단 레이아웃을 구현한다 <b>277</b>	
	-webkit-column-count: 다단 레이아웃에서 열 개수를 지정한다	283
	-webkit-column-width: 다단 레이아웃에서 열의 너비를 설정한다	286
	-webkit-column-gap: 다단 레이아웃에서 열 사이 간격을 설정한다	289
	-webkit-column-rule: 다단 레이아웃에서 열을 구분하는 선을 설정한다	292
	-webkit-column-rule-color: 다단 레이아웃에서 열을 구분하는 선의 색깔을 설정한	다
	295	
	-webkit-column-rule-style: 다단 레이아웃에서 열을 구분하는 선의 스타일을 설정	한다
	298	
	-webkit-column-rule-width: 애니메이션을 설정한다 <b>301</b>	
	-webkit-column-break-after: 다단 레이아웃에서 요소 다음에 강제로 단을 나눈다	
	303	
	-webkit-column-break-before: 다단 레이아웃에서 요소 앞에서 강제로 단을 나눈	다
	307	
	-webkit-column-break-inside: 다단 레이아웃에서 요소의 단을 나누지 않게 설정	한다
	311	
	-webkit-column-fill: 다단 레이아웃에서 콘텐츠를 채우는 방법을 설정한다	315
	-webkit-column-span: 다단 레이아웃에서 요소가 여러 열에 걸쳐 나타나게 한다	316

Part 9	Gradient —— 319	
	-webkit-linear-gradient: 선형 그러데이션 효과를 적용한다	321
	-webkit-radial-gradient: 방사형 그러데이션 효과를 구현한다	330
	-webkit-repeating-linear-gradient: 선형 그러데이션 패턴을 만든다	338
	-webkit-repeating-radial-gradient: 방사형그러데이션 패턴을 만든다	341
Part 10	Margin ——— 345	
	-webkit-margin-collapse: 요소의 여백 병합 방법을 설정한다	347
	-webkit-margin-top-collapse: 요소의 위쪽 여백 병합 방법을 설정한다	351
	-webkit-margin-bottom-collapse: 요소의 아래 여백 병합 방법을 설정한다	356
Part <b>11</b>	Marquee ——— 361	
	-webkit-marquee: 전광판에 글자가 흘러가는 듯한 효과를 구현한다	363
	-webkit-marquee-direction: 마퀴 효과의 방향을 지정한다	366
	-webkit-marquee-increment: 마퀴 효과의 이동 거리를 지정한다	370
	-webkit-marquee-repetition: 마퀴 효과의 이동 횟수를 지정한다	372
	-webkit-marquee-speed: 마퀴 효과의 이동 속도를 지정한다	374
	-webkit-marquee-style: 마퀴 효과의 이동 스타일을 지정한다	376

Part 12	Mask 381	
	-webkit-mask: 포토샵의 마스크 기능을 CSS로 구현한다	383
	-webkit-mask-attachment: 마스크 영역이 뷰포트보다 클 때 스크롤 여부	를 설정한다
	388	
	-webkit-mask-box-image: 마스크 효과를 적용한다	393
	-webkit-mask-clip: 마스크 효과 적용 범위를 지정한다	400
	-webkit-mask-image: 마스크 효과를 적용할 이미지를 적용한다	405
	-webkit-mask-origin: 마스크 효과를 적용할 기준을 설정한다	409
	-webkit-mask-position: 마스크 효과에서 마스크 이미지 위치를 설정한다	413
	-webkit-mask-repeat: 마스크 효과에서 미스크 이미지의 반복 여부를 설정한다	416
	-webkit-mask-size: 마스크 효과에서 마스크 이미지의 크기를 설정한다	422
Part 13	3D —— 427	
	-webkit-perspective: 원근감을 조절한다 429	
	-webkit-perspective-origin: 소실점의 위치를 변경한다	432
	-webkit-transform: 요소의 형태를 변형한다 435	
	-webkit-transform-origin: 요소의 형태를 변형하는 기준점을 지정한다	441
	-webkit-transform-origin-x: 요소를 X축으로 변형할 때 X축 기준점을 지	정한다
	444	
	-webkit-transform-origin-y: 요소를 Y축으로 변형할 때 Y축 기준점을 지	정한다
	447	
	-webkit-transform-origin-z: 요소를 Z축으로 변형할 때 Z축 기준점을 자	정한다
	450	
	-webkit-transform-style: 애니메이션 요소를 3차원으로 렌더링한다	455

Part <b>14</b>	Text ——— 459		
	-webkit-text-fill-color: 텍스트에 색상을 지정한다	461	
	-webkit-text-security: 입력하는 문자를 마스크 문자로 가린다 -		463
	-webkit-text-size-adjust: 뷰포트에 따라 글자 크기를 변경한다		465
	-webkit-text-stroke: 글자에 윤곽선을 그린다	469	
	-webkit-text-stroke-color: 글자 윤곽선의 색상을 지정한다 -		471
	-webkit-text-stroke-width: 글자 윤곽선의 너비를 지정한다		472
Part 15	User —— 475		
	-webkit-user-drag: 요소를 마우스로 드래그할 수 있다	477	
	-webkit-user-modify: 요소 안에 있는 텍스트를 편집할 수 있다		481
	-webkit-user-select: 콘텐츠를 선택할 수 없게 된다	485	
Part 16	기타 487		
	-webkit-appearance: 폼 요소의 모양을 바꾼다	489	
	-webkit-backface-visibility: 요소의 뒷면을 보여 준다	492	2
	-webkit-calc: CSS 속성으로 사칙 연산을 한다	195	
	-webkit-cross-fade: 두 이미지를 합성한다	198	
	-webkit-filter: 포토샵의 필터 기능을 CSS로 적용한다	501	
	-webkit-font-kerning: 인쇄에서 적용하면 커닝을 웹에서 적용한다		511
	-webkit-nbsp-mode:  로 줄바꿈을 조정한다	514	
	-webkit-rtl-ordering: 텍스트 렌더링 방향을 지정한다	517	
	-webkit-tap-highlight-color: 모바일 브라우저에서 링크 요소를 터치힐	때 요소의 색깔을	지정한다
	520		

 -webkit-touch-callout: 모바일 브라우저에서 요소를 길게 눌렀을 때의 동작을 지정한다

 524

 -webkit-writing-mode: 글을 쓰는 방향을 바꾼다

527

# Part **17** 부록 — 531

 부록.1 길이 단위
 533

 부록.2 색깔표기
 533

 부록.3 선모양표기
 538

 부록.4 선너비표기
 540

 부록.5 저자 테스트 리스트
 541



# Part 1 Animation

속성명	-webkit-anima	tion					
요약 설명	애니메이션을 설정한다.						
정의	애니메이션 이름과	<sup>나</sup> 재생 횟수, 자	생 방향 등 여러 애	니메이션 속성을 혀	하나로 합쳐서 설정	한다.	
지원환경	0 15 이상	0	<b>3</b>	É	ı <del>``</del> ı		
	0	0	4.0 이상 〇	2.0 이상 〇	0		

# ■ 구문

-webkit-animation: <name> || <duration> || <timing-function> || <delay> || <iterationcount> || <direction> || <fiil-mode> || <play-state> [,...];

### name: 애니메이션 이름.

@-webkit-keyframes 속성에서 설정한 애니메이션의 이름이다. 이름을 설정해야 애니메이션을 재생할수 있다.

duration: 애니메이션을 한 번 재생하는 데 걸리는 시간.

- 0: 애니메이션을 재생하지 않는다(기본값).
- 숫자와 단위: 애니메이션을 한 번 재생하는 데 걸리는 시간이다. 초(s)와 밀리초(ms) 단위를 사용한다. 단위가 없거나 값이 음수면 애니메이션을 재생하지 않는다.

# timing-function: 애니메이션의 단계별 재생 속도.

- ease: 애니메이션이 시작해 서서히 빨라지다 중간부터 다시 서서히 느려져 종료된다(기본값).
- ease-in: 애니메이션이 시작해 점점 빨라진다.
- ease-out: 애니메이션이 빨리 시작되고 점점 느려진다.
- ease-in-out: 애니메이션이 서서히 시작하고 서서히 종료된다. 시작할 때의 속도는 ease-in 값을 적용했을 때와 같고, 종료할 때의 속도는 ease-out 값을 적용했을 때와 같다.
- linear: 애니메이션을 처음부터 끝까지 일정한 속도로 재생한다.
- cubic-bezier (x1,y1,x2,y2): 큐빅 베지어<sup>cubic-bezier</sup> 곡선의 좌표 값을 직접 입력해 애니메이션 의 단계별 재생 속도를 설정한다.

# delay: 애니메이션 시작을 지연할 시간.

- 0: 속성을 적용하자마자 애니메이션을 시작한다(기본값).
- now: 속성을 적용하자마자 애니메이션을 시작한다. 0으로 설정한 것과 같다. iOS 2.0부터 사용할 수 있다.

• 숫자와 단위: 설정한 시간이 지난 뒤에 애니메이션을 시작한다. 사용할 수 있는 단위는 초(s)와 밀리초 (ms)다. 값이 양수면 속성을 적용한 순간부터 시간을 계산해 애니메이션 재생을 지연한다. 값이 음수면 속성을 적용한 순간 바로 애니메이션을 실행하지만, 지정한 시간이 지난 뒤의 장면부터 애니메이션을 재생한다. 예를 들어, 값이 '-1s'면 1초가 지난 뒤의 장면부터 애니메이션을 재생한다.

iteration-count: 애니메이션을 재생하는 횟수.

- 숫자: 설정한 횟수만큼 애니메이션을 재생한다. 기본값은 1이다. 숫자가 소수면 애니메이션을 재생하는 도중에 첫 번째 프레임으로 돌아가 멈춘다. 숫자가 음수면 애니메이션을 재생하지 않는다.
- infinite: 애니메이션을 무한으로 반복한다.

direction: 애니메이션 재생 방향.

- @-webkit-keyframes 속성의 from에 설정한 스타일에서 to에 설정한 스타일로 재생하는 것이 순방향이고, to에 설정한 스타일에서 from에 설정한 스타일로 재생하는 것이 역방향이다.
- normal: 애니메이션을 순방향으로 재생한다(기본값). 재생이 한 번 끝나면 첫 번째 프레임부터 다시 시작한다.
- alternate: 순방향으로 애니메이션을 시작해 역방향과 순방향으로 번갈아 애니메이션을 재생한다. 홀수 번째로 재생할 때는 순방향으로 재생하고, 짝수 번째로 재생할 때는 역방향으로 재생한다.
- reverse: 애니메이션을 역방향으로 재생한다. 재생이 한 번 끝나면 마지막 프레임부터 다시 시작한다.
- alternate-reverse: 역방향으로 애니메이션을 시작해 순방향과 역방향으로 번갈아 애니메이션을 재생한다. 홀수 번째로 재생할 때는 역방향으로 재생하고. 짝수 번째로 재생할 때는 순방향으로 재생한다.

fill-mode: 애니메이션 재생의 시작과 끝에 적용할 스타일.

- none: 애니메이션을 재생하기 전이나 재생한 후에는 요소에 기본으로 설정한 스타일을 적용한다(기본 값). 애니메이션 효과는 애니메이션에 정의된 시간동안만 적용된다.
- forwards: 애니메이션 재생이 끝난 뒤에 마지막 프레임에 설정한 스타일을 적용한다.
- backwards: -webkit-animation-delay 속성으로 애니메이션 재생을 지연하는 동안에도 첫 번째 프레임에 설정한 스타일을 적용한다.
- both: -webkit-animation-delay 속성으로 애니메이션 재생을 지연하는 동안에도 첫 번째 프레임에 설정한 스타일을 적용하고, 애니메이션 재생이 끝난 뒤에는 마지막 프레임에 설정한 스타일을 적용한다.

play-state: 애니메이션 재생 여부.

• running: 애니메이션을 재생한다(기본값).

• paused: 애니메이션을 정지한다.

# 예제

다음 코드는 -webkit-animation 속성을 적용한 예제다. 예제를 실행하면 3초 동안 꽃 이미지가 한 바퀴 회전하는 애니메이션이 무한으로 재생되도록 -webkit-animation 속성을 설정했다.

### [예제 1-1] -webkit-animation 적용 예제

#### [그림 1-1] 무한으로 회전하는 꽃 이미지



속성값을 애니메이션 이름(flower)과 재생 속도(linear), 시간(3s), 반복 횟수(infinite)만 적용했지만 실제 브라우저는 지연 시간, 재생 방향, 스타일, 재생 여부를 다음과 같이 기본값으로 적용되다.

-webkit-animation: flower 3s linear Os infinite normal none running;

다음 링크와 QR 코드로 PC와 모바일에서 예제를 실행할 수 있다.

• 예제 실행 URL: http://jsbin.com/-webkit-animation

● 예제 실행 QR 코드 **■ 현장**■



속성명	-webkit-animation-delay					
요약 설명	애니메이션에 지연 시간을 설정한다.					
정의	애니메이션 시작을	애니메이션 시작을 얼마나 지연할지 설정한다.				
지원환경	0 15 이상	9	<b></b>	É	iii	
	0	0	4.0 이상 〇	2.0 이상 〇	0	

# ■ 구문

-webkit-animation-delay : \langle time \rangle[,...];

time: 애니메이션 지연 시간.

- 0: 속성을 적용하자마자 애니메이션을 시작한다(기본값).
- now: 속성을 적용하자마자 애니메이션을 시작한다. 0으로 설정한 것과 같다. iOS 2.0부터 사용할 수 있다.
- 숫자와 단위: 설정한 시간이 지난 후 애니메이션을 시작한다. 사용할 수 있는 단위는 초(s)와 밀리초 (ms)다. 값이 양수면 속성을 적용한 순간부터 시간을 계산해 애니메이션 재생을 지연한다. 값이 음수면 속성을 적용한 순간 바로 애니메이션을 실행하지만 지정한 시간이 지난 후의 장면부터 애니메이션을 재생한다. 예를 들어, 값이 '-1s'면 1초가 지난 후의 장면부터 애니메이션을 재생한다.

# 예제

다음 코드는 -webkit-animation-delay 속성을 적용한 예제다. 이 예제는 요소를 클릭하면 지정된 애니메이션을 재생하도록 스크립트가 적용되어 있다. -webkit-animation-delay 속성값을 0으로 설정하면 예제를 실행하고 사각형을 클릭했을 때 사각형이 화면 왼쪽에서 오른쪽으로 이동하는 애니메이션이 바로 재생된다.

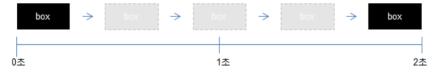
# [예제 1-2] -webkit-animation-delay:0 적용 예저

```
div {
    position: absolute;
    left: 100px;
    width: 100px;
    height: 50px;
    margin-left: -100px;
    background: #000;
    color: #fff;
```

```
-webkit-animation-timing-function: linear;
        -webkit-animation-duration: 2s;
        -webkit-animation-delay: 0;
    }
    @-webkit-kevframes test{
        from {
             left: 100px;
        }
        to {
            left: 300px;
        }
    }
</style>
<div>
    hox
</div>
<script>
    $(document).readv(function() {
        $('div').click(function() {
             $(this).css('-webkit-animation-name', 'test');
        });
    });
</script>
```

다음은 [예제 1-2]를 실행해 사각형이 이동하는 상태를 시간의 흐름에 따라 표현한 그림이다. 사각형이 왼쪽에서 오른쪽으로 2초 동안 이동하다.

[그림 1-2] -webkit-animation-delay:0 실행 화면



다음 코드는 -webkit-animation-delay 속성값을 now로 설정한 예제다. 예제를 실행하고 사각형을 클릭하면 사각형이 화면 왼쪽에서 오른쪽으로 이동하는 애니메이션이 바로 재생된다.

```
[예제 1-3] -webkit-animation-delay:now 석용 예저
```

```
<style>
   div {
```

```
-webkit-animation-timing-function: linear;
-webkit-animation-duration: 2s;
-webkit-animation-delay: now;
}
...
</style>
```

다음 코드는 -webkit-animation-delay 속성값을 2s로 설정한 예제다. 예제를 실행하고 사각형을 클릭하면 2초 후 애니메이션이 재생된다.

### [예제 1-4] -webkit-animation-delay:2s 적용 예저

```
div {
     ...
     -webkit-animation-name: test;
     -webkit-animation-duration: 2s;
     -webkit-animation-delay: 2s;
}
...
</style>
...
```

다음 코드는 -webkit-animation-delay 속성값을 음수인 -1s로 설정한 예제다. 예제를 실행하고 사각형을 클릭하면 사각형이 화면 가운데에서부터 오른쪽으로 이동하는 애니메이션 이 재생된다. 값을 음수로 설정했기 때문에 1초가 지난 후인, 화면 가운데 위치의 장면부터 애니메이션이 재생됐다.

#### [예제 1-5] -webkit-animation-delay:-1s 적용 예저

다음은 [예제 1-5]을 실행해 사각형이 이동하는 상태를 표현한 그림이다. 애니메이션이 재생되고 1초가 지난 위치에서부터 사각형이 이동하기 시작한다.

[그림 1-3] -webkit-animation-delay:-1s 실행 화면



다음 링크와 QR 코드로 PC와 모바일에서 예제를 확인할 수 있다.

- 예제 실행 URL: http://jsbin.com/-webkit-animation-delay

속성명	-webkit-animation-direction						
요약 설명	애니메이션에 재생 방향을 설정한다.						
정의	애니메이션의 재생 방향을 설정한다. @-webkit-keyframes 속성의 from에 설정한 스타일에서 to에 설정한 스타일로 재생하는 것이 순방향이고, to에 설정한 스타일에서 from에 설정한 스타일로 재생하는 것이 역방향이다.						
지원환경	0 15 이상	0	<b>6</b>	<b>É</b>	i		
	0	0	4.0 이상 〇	2.0 이상 〇	0		

# ■ 구문

-webkit-animation-direction : \( \def( derection \) [...];

direction: 애니메이션 재생 방향.

- normal: 애니메이션을 순방향으로 재생한다(기본값). 재생이 한 번 끝나면 첫 번째 프레임부터 다시 시작한다.
- alternate: 순방향으로 애니메이션을 시작해 역방향과 순방향으로 번갈아 애니메이션을 재생한다. 홀수 번째로 재생할 때는 순방향으로 재생하고, 짝수 번째로 재생할 때는 역방향으로 재생한다.
- reverse: 애니메이션을 역방향으로 재생한다. 재생이 한 번 끝나면 마지막 프레임부터 다시 시작한다.
- alternate-reverse: 역방향으로 애니메이션을 시작해 순방향과 역방향으로 번갈아 애니메이션을 재생한다. 홀수 번째로 재생할 때는 역방향으로 재생하고, 짝수 번째로 재생할 때는 순방향으로 재생한다.

#### NOTE

애니메이션이 역방향으로 재생될 때는 -webkit-animation-timing-function 속성과 관련된 효과도 반대로 적용된다. 예를 들어 서서히 시작했다 빠르게 종료하는 -webkit-animation-timing-function:ease-in을 적용한 애니메이션을 역방향으로 재생하면 빠르게 시작했다 서서히 종료하게 된다.

# 예제

다음 코드는 -webkit-animation-direction 속성을 적용한 예제다. 이 예제는 요소를 클릭하면 지정된 애니메이션을 재생하도록 스크립트가 적용되어 있다. 예제를 실행하고 사각형을 클릭하면 사각형이 화면 왼쪽에서 오른쪽으로 이동하는 애니메이션이 세 번 재생된다. -webkit-animation-direction 속성값을 normal로 설정했기 때문에 한 번 재생이 끝나면 사각형이 다시 화면 왼쪽에서 오른쪽으로 이동한다.

```
<style>
    div {
        position: absolute;
        left: 100px:
        width: 100px;
        height: 50px;
        margin-left: -100px;
        background: #000;
        color: #fff:
        -webkit-animation-name: test;
        -webkit-animation-duration: 2s:
        -webkit-animation-iteration-count: 3:
        -webkit-animation-timing-function :ease-in:
        -webkit-animation-direction: normal;
    }
    @-webkit-keyframes test{
        from {
             left: 100px;
        }
        to {
             left: 300px;
        }
    }
</style>
<div>
    box
</div>
<script>
    $(document).ready(function() {
        $('div').click(function() {
             $(this).css('-webkit-animation-name', 'test');
        });
    });
</script>
```

다음 코드는 사각형에 -webkit-animation-direction:alternate를 설정한 예제다. alternate 속성값은 순방향으로 애니메이션 재생을 시작해 역방향과 순방향으로 방향을 바꿔 가며 애니메이션을 재생한다. 예제를 실행하고 사각형을 클릭하면 처음에는 사각형이 순방

향인 화면 왼쪽에서 오른쪽으로 이동한다. 애니메이션이 두 번째로 재생될 때는 사각형이 역 방향인 오른쪽에서 왼쪽으로 이동한다. 세 번째로 재생될 때는 다시 순방향인 왼쪽에서 오른 쪽으로 이동한다.

#### [예제 1-7] -webkit-animation-direction:alternate 적용 예저

다음 코드는 사각형에 -webkit-animation-direction:reverse를 설정한 예제다. -webkit-animation-direction 속성값을 reverse로 설정했기 때문에 한 번 재생이 끝나면 사각형이 다시 화면 오른쪽에서 왼쪽으로 이동한다. 예제를 실행하고 사각형을 클릭하면 사각형이 화면 오른쪽에서 왼쪽으로 이동하는 애니메이션이 세 번 재생된다.

#### [예제 1-8] -webkit-animation-direction:reverse 전용 예제

```
div {
     ...
     -webkit-animation-iteration-count: 3;
     -webkit-animation-timing-function :ease-in;
     -webkit-animation-direction: reverse;
}
...
</style>
...
```

다음 코드는 사각형에 -webkit-animation-direction:alternate-reverse를 설정한 예제다. alternate-reverse 속성값은 역방향으로 애니메이션 재생을 시작해 순방향과 역 방향으로 방향을 바꿔 가며 애니메이션을 재생한다. 예제를 실행하고 사각형을 클릭하면 처음에는 사각형이 역방향인 화면 오른쪽에서 왼쪽으로 이동한다. 애니메이션이 두 번째로 재생될

때는 사각형이 순방향인 왼쪽에서 오른쪽으로 이동한다. 세 번째로 재생될 때는 다시 역방향 인 오른쪽에서 왼쪽으로 이동한다.

#### [예제 1-9] -webkit-animation-direction: alternate-reverse 적용 예제

```
div {
     ...
     -webkit-animation-iteration-count: 3;
     -webkit-animation-timing-function :ease-in;
     -webkit-animation-direction: alternate-reverse;
}
     ...
</style>
...>
```

다음 링크와 QR 코드로 PC와 모바일에서 예제를 확인할 수 있다.

- 예제 실행 URL: http://jsbin.com/-webkit-animation-direction
- 예제 실행 QR 코드 ■낮



속성명	-webkit-animation-duration					
요약 설명	애니메이션 재생 시간을 설정한다.					
정의	의 애니메이션을 처음부터 끝까지 한 번 재생하는 데 걸리는 시간을 설정한다. @ keyframes 속성의 from에 설정한 스타일에서 to에 설정한 스타일로 바뀌는 데 이다.					
지원환경	0 15 이상	9	<b>6</b>	Ć	i	
	0	0	4.0 이상 〇	2.0 이상 〇	0	

# ■ 구문

-webkit-animation-duration: <time> [,...];

time: 애니메이션을 한 번 재생하는 데 걸리는 시간.

- 0: 애니메이션을 재생하지 않는다(기본값).
- 숫자와 단위: 애니메이션을 한 번 재생하는 데 걸리는 시간이다. 사용할 수 있는 단위는 초(s)와 밀리초 (ms)다. 단위가 없거나 값이 음수면 애니메이션을 재생하지 않는다.

# 예제

다음 코드는 -webkit-animation-duration 속성을 적용한 예제다. 이 예제는 요소를 클릭하면 지정된 애니메이션을 재생하도록 스크립트가 적용되어 있다. 예제를 실행하고 사각형을 클릭하면 사각형이 화면 왼쪽에서 오른쪽으로 이동하도록 애니메이션을 설정했다. -webkit-animation-duration 속성값을 0으로 설정했기 때문에 사각형을 클릭해도 애니메이션이 재생되지 않는다. 값을 음수(-2s)로 바꿔서 설정해도 애니메이션은 재생되지 않는다.

#### [예제 1-10] -webkit-animation-duration:0 적용 예저

```
div {
    position: absolute;
    left: 100px;
    width: 100px;
    height: 50px;
    margin-left: -100px;
    background: #000;
    color: #fff;
    -webkit-animation-duration: 0;
    /* -webkit-animation-duration: -2s */ /* 값이 음수여도 애니메이션이 재생되지 않는다.
```

```
*/
    @-webkit-kevframes test{
         from {
             left: 100px;
         }
         to {
             left: 300px;
         }
    }
</style>
<div>
    box
</div>
<script>
    $(document).ready(function() {
         $('div').click(function() {
             $(this).css('-webkit-animation-name', 'test');
         });
    });
</script>
```

다음 코드는 -webkit-animation-duration 속성의 값을 2s로 설정한 예제다. 이 예제를 실행하고 사각형을 클릭하면, 사각형이 2초 동안 화면 왼쪽에서 오른쪽으로 이동하는 애니메이션이 재생되다.

## [예제 1-11] -webkit-animation-duration:2s 적용 예저

다음 링크와 QR 코드로 PC와 모바일에서 예제를 확인할 수 있다.

- 예제 실행 URL: http://jsbin.com/-webkit-animation-duration
- 예제 실행 QR 코드 **■**참



속성명	-webkit-animation-fill-mode						
요약 설명	애니메이션 시작과 끝에 스타일을 설정한다.						
정의	애니메이션의 시작과 끝에 어떤 스타일을 적용할지 설정한다. 애니메이션이 어떻게 나타나고 어떻게 사라질지 효과를 넣는 속성이다. 예를 들어 -webkit-animation-delay 속성으로 애니메이션 재생이 지연되고 있을 때에는, 지연되는 동안 애니메이션의 첫 프레임에 설정한 스타일을 볼 수 없다. 하지만 -webkit-animation-fill-mode 속성으로 backward나 both를 설정하면지연 시간에 상관없이 애니메이션의 첫 프레임에 설정한 스타일을 볼 수가 있다.						
지원환경	0 15 이상	9		<b>É</b>	iii		
	0	0	5.0 이상 〇	4.0 이상 〇	0		

# ■ 구문

-webkit-animation-fill-mode : \langle mode \rangle [,...];

mode: 애니메이션을 재생할 때 시작과 끝에 적용할 스타일의 유형.

- none: 애니메이션을 재생하기 전이나 재생한 후에는 요소에 기본으로 설정한 스타일을 적용한다(기본 값). 애니메이션 효과는 애니메이션에 정의된 시간 동안만 적용된다.
- forwards: 애니메이션 재생이 끝난 후에 마지막 프레임에 설정한 스타일을 적용한다.
- backwards: -webkit-animation-delay 속성으로 애니메이션 재생을 지연하는 동안에도 첫 번째 프레임에 설정한 스타일을 적용한다.
- both: -webkit-animation-delay 속성으로 애니메이션 재생을 지연하는 동안에도 첫 번째 프레임에 설정한 스타일을 적용하고, 애니메이션 재생이 끝난 후에는 마지막 프레임에 설정한 스타일을 적용한다.

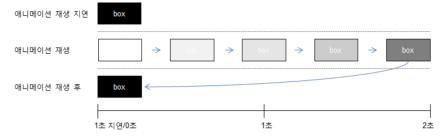
# ■ 예제

다음 코드는 -webkit-animation-fill-mode 속성을 적용한 예제다. 이 예제는 요소를 클릭하면 지정된 애니메이션을 재생하도록 스크립트가 적용되어 있다. 사각형의 기본 스타일은 배경색이 검정색이지만 애니메이션 시작은 흰색으로, 애니메이션 종료는 회색으로 스타일을 적용했다. 애니메이션에 1초의 지연 시간을 적용한 상태다. 예제를 실행하고 검정색 사각형을 클릭하면 1초 후에 사각형이 화면 왼쪽에서 오른쪽으로 이동하며 배경색이 흰색에서 회색으로 바뀐다. -webkit-animation-fill-mode 속성값을 none으로 설정했기 때문에 애니메이션이 지연되는 동안 흰색으로 변화되는 것은 보이지 않고 기본 스타일 색인 검정색으로 유지되다 지연이 끝난 후 배경색이 흰색에서 회색으로 바뀐다. 애니메이션이 끝난 후에는 다시 애니메이션을 설정하기 전 상태로 돌아온다.

```
<style>
    div {
        position: absolute;
        left: 100px;
        width: 100px;
        height: 50px;
        margin-left: -100px;
        background: #000;
        color: #fff;
        -webkit-animation-timing-function: linear;
        -webkit-animation-duration: 2s;
        -webkit-animation-delay: 1s;
        -webkit-animation-fill-mode: none:
    @-webkit-keyframes test{
        from {
             left: 100px;
             background-color: #fff;
        }
        to {
             left: 300px;
             background-color: #999;
        }
    }
</style>
<div>
    box
</div>
⟨script⟩
    $(document).ready(function() {
        $('div').click(function() {
             $(this).css('-webkit-animation-name', 'test');
        });
    });
</script>
```

다음은 [예제1-12]를 실행해 사각형의 스타일이 변화하는 과정을 단계별로 표현한 그림이다.

[그림 1-4] -webkit-animation-fill-mode:none 실행 화면



애니메이션 재생이 지연되는 동안에는 기본 스타일이 적용되어 사각형의 배경색은 검정색이다. 애니메이션이 재생되면 배경색은 흰색이 되고 사각형이 화면 오른쪽으로 이동한다. 사각형이 이동하는 동안 배경색도 흰색에서 회색으로 바뀐다. 애니메이션 재생이 끝나면 다시 기본 스타일이 적용되어 사각형은 화면 왼쪽에 위치하고 배경색은 검정색이 된다.

다음 코드는 -webkit-animation-fill-mode 속성값을 forwards로 설정한 예제다. 예제를 실행하고 애니메이션이 끝나면 마지막 프레임의 스타일이 사각형에 적용되어 사각형이 계속 화면 오른쪽에 위치하고 배경색도 계속 회색이 된다.

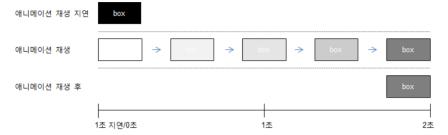
## [예제 1-13] -webkit-animation-fill-mode:forwards 적용 예저

```
<style>
    div {
        position: absolute;
        left: 100px;
        margin-left: -100px;
        background: #000;
        -webkit-animation-delay: 1s;
        -webkit-animation-fill-mode: forwards;
    @-webkit-keyframes test{
        from {
            left: 100px;
            background-color: #fff;
        to {
            left: 300px;
            background-color: #999;
    }
```

. . .

다음은 [예제 1-13]을 실행해 사각형의 스타일이 변화하는 과정을 단계별로 표현한 그림이다.

[그림 1-5] -webkit-animation-fill-mode:forwards 실행 화면



애니메이션 재생이 지연되는 동안에는 기본 스타일이 적용되어 사각형의 배경색은 검정색이다. 애니메이션 재생이 시작되면 배경색이 흰색이 되고 사각형이 화면 오른쪽으로 이동한다. 사각형이 이동하는 동안 배경색이 흰색에서 회색으로 바뀐다. 애니메이션 재생이 끝나면 마지막 프레임의 스타일이 적용되어 사각형은 화면 오른쪽에 위치하고 배경색은 회색이다.

다음 코드는 -webkit-animation-fill-mode 속성값을 backwards로 설정한 예제다. 예제를 실행하고 사각형을 클릭하면 첫 번째 프레임의 스타일이 적용되어 배경색이 흰색인 상태로 애니메이션이 지연된다.

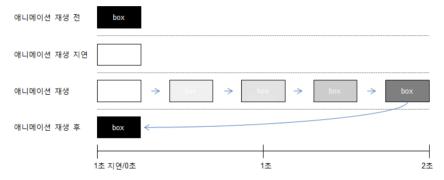
#### [예제 1-14] -webkit-animation-fill-mode:backwards 적용 예저

```
div {
    position: absolute;
    left: 100px;
    ...
    margin-left: -100px;
    background: #000;
    ...
    -webkit-animation-delay: 1s;
    -webkit-animation-fill-mode: backwards;
}
@-webkit-keyframes test{
    from {
        left: 100px;
    }
}
```

```
background-color: #fff;
}
to {
    left: 300px;
    background-color: #999;
}
</style>
...
```

다음은 [예제 1-14]를 실행해 사각형의 스타일이 변화하는 과정을 단계별로 표현한 그림이다.

[그림 1-6] -webkit-animation-fill-mode:backwards 실행 화면



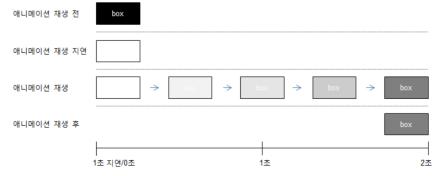
사각형을 클릭해 애니메이션 재생이 지연되면 첫 번째 프레임의 스타일이 적용되어 사각형의 배경색은 흰색이 된다. 애니메이션 재생이 시작되면 사각형이 화면 오른쪽으로 이동하면서 배경색이 회색으로 바뀐다. 애니메이션 재생이 끝나면 다시 기본 스타일이 적용되어 사각형은 화면 왼쪽에 위치하고 배경색은 검정색이 된다.

다음 코드는 -webkit-animation-fill-mode 속성값을 both로 설정한 예제다. -webkit-animation-fill-mode 속성값을 both로 설정하면 값을 forwards로 설정한 효과와 backwards로 설정한 효과가 모두 적용된다. 예제를 실행하고 사각형을 클릭하면 첫 번째 프레임의 스타일이 적용되어 배경색이 흰색인 상태로 애니메이션이 지연된다. 애니메이션 재생이 끝나면 마지막 프레임의 스타일이 적용되어 사각형은 화면 오른쪽에 위치하고 배경색은 회색이 된다.

```
<style>
    div {
        position: absolute;
        left: 100px;
        margin-left: -100px;
        background: #000;
        -webkit-animation-delay: 1s;
        -webkit-animation-fill-mode: both;
    @-webkit-keyframes test{
        from {
            left: 100px;
            background-color: #fff;
        }
        to {
            left: 300px;
            background-color: #999;
        }
    }
</style>
```

다음은 [예제 1-15]를 실행해 사각형의 스타일이 변화하는 과정을 단계별로 표현한 그림이다.

## [그림 1-7] -webkit-animation-fill-mode:both 실행 화면



사각형을 클릭해 애니메이션 재생이 지연되면 첫 번째 프레임의 스타일이 적용되어 사각형의 배경색은 흰색이 된다. 애니메이션 재생이 시작되면 사각형이 화면 오른쪽으로 이동하면서 배경색이 회색으로 바뀐다. 애니메이션 재생이 끝나면 마지막 프레임의 스타일이 적용되어 사각형은 화면 오른쪽에 위치하고 배경색은 회색이 된다.

#### NOTE

안드로이드 4.0.3에서 애니메이션이 실행될 때는 깜빡임 현상이 나타난다. left나 top, width, height 등 CSS2 속성값을 변경하는 애니메이션이 끝나고 요소의 기본 속성으로 돌아가는 순간에 깜빡인다.

애니메이션이 적용되기 전 상태로 돌아가지 못하도록 -webkit-animation-fill-mode 속성에 forwards나 backwards, both 중 아무 값이나 적용하면 문제가 해결되다.

다음 링크와 OR 코드로 PC와 모바일에서 예제를 확인할 수 있다.

• 예제 실행 URL: http://jsbin.com/-webkit-animation-fill-mode/

● 예제 실행 QR 코드 ■ ##X ■



속성명	webkit-animation-iteration-count						
요약 설명	애니메이션을 무힌	애니메이션을 무한으로 재생한다.					
정의	애니메이션을 재상	애니메이션을 재생할 횟수를 설정한다.					
지원환경	<b>○</b> 15 이상 <b>⑤ ⑥ ⑥</b>						
	0	0	4.0 이상 〇	2.0 이상 〇	0		

# ■ 구문

```
-webkit-animation-iteration-count : <number> [, ...];
```

number: 애니메이션을 재생하는 횟수.

- 숫자: 애니메이션을 설정한 횟수만큼 재생한다(기본값: 1). 숫자가 소수이면 애니메이션을 재생하는 도 중에 첫 번째 프레임으로 돌아가 멈춘다. 숫자가 음수이면 한 번만 재생한다.
- infinite: 애니메이션을 무한으로 반복한다.

## ■ 예제

다음 코드는 -webkit-animation-iteration-count 속성을 적용한 예제다. 이 예제는 요소를 클릭하면 지정된 애니메이션을 재생하도록 스크립트가 적용되어 있다. 예제를 실행하고 사각형을 클릭하면 사각형이 화면 왼쪽에서 오른쪽으로 이동한다. -webkit-animation-iteration-count 속성값을 1로 설정했기 때문에 애니메이션은 한 번만 재생된다.

#### [예세 1-16] -webkit-animation-iteration-count :1 석용 예세

```
div {
    position: absolute;
    left: 100px;
    width: 100px;
    height: 50px;
    margin-left: -100px;
    background: #000;
    color: #fff;
    -webkit-animation-timing-function: linear;
    -webkit-animation-duration: 2s;
    -webkit-animation-iteration-count: 1;
}
```

```
@-webkit-kevframes test{
         from {
             left: 100px;
        to {
             left: 300px;
    }
</style>
<div>
    hox
</div>
<script>
    $(document).readv(function() {
        $('div').click(function() {
             $(this).css('-webkit-animation-name', 'test');
        });
    });
</script>
```

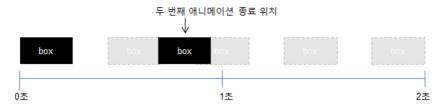
다음 코드는 -webkit-animation-iteration-count 속성값을 소수인 1.3으로 설정한 예제다. 예제를 실행하고 사각형을 클릭하면 애니메이션이 한 번 재생된 다음 전체 프레임의 3/10에 해당하는 프레임까지 더 재생되고 애니메이션이 끝난다.

#### [예제 1-17] -webkit-animation-iteration-count :1.3 적용 예저

```
div {
    div {
        ...
        -webkit-animation-timing-function: linear;
        -webkit-animation-duration: 2s;
        -webkit-animation-iteration-count: 1.3;
    }
    ...
</style>
    ...
```

다음은 [예제 1-17]을 실행해 사각형이 이동하는 상태를 시간의 흐름에 따라 표현한 그림이다. 전체 프레임의 3/10에 해당하는 프레임에서 애니메이션이 끝난다.

### [그림 1-8] -webkit-animation-interation-count :1.3 속성 실행 결과



다음 코드는 -webkit-animation-iteration-count 속성값을 infinite로 설정한 예제다. 예제를 실행하고 사각형을 클릭하면 애니메이션이 무한으로 재생된다.

## [예제 1-18] -webkit-animation-iteration-count : infinite 적용 예정

```
div {
     ...
     -webkit-animation-timing-function: linear;
     -webkit-animation-duration: 2s;
     -webkit-animation-iteration-count: infinite;
}
     ...
</style>
...
```

다음 링크와 QR 코드로 PC와 모바일에서 예제를 확인할 수 있다.

- 예제 실행 URL: http://jsbin.com/-webkit-animation-iteration-count
- 예제 실행 QR 코드 ■류



속성명	-webkit-animation-name						
요약 설명	애니메이션 이름을 설정한다.						
정의	재생할 @-webkit-keyframes 속성의 이름을 설정한다. 요소에 적용할 애니메이션을 선택하는 데 사용한다.						
지원환경	0 15 이상	6	<b>6</b>	É	i		
	0	0	4.0 이상 〇	2.0 이상 〇	0		

# ■ 구문

-webkit-animation-name : ⟨name⟩ [,...];

name: 애니메이션을 식별하는 문자열.

- none: 애니메이션을 재생하지 않는다. 값이 none이면 none이라는 이름의 @-webkit-keyframes 속 성이 있어도 애니메이션을 재생하지 않는다.
- 문자열: 재생할 애니메이션의 이름이다.

# 지 아 마 아니메이션 이름과 유효하지 않은 애니메이션 이름 다음 애니메이션 이름은 유효한 값이다. -webkit-animation-name:name; // 문자열로 시작하는 이름 -webkit-animation-name:-name; // 언더바(\_)로 시작하는 이름 -webkit-animation-name:-name; // 하이픈(-)으로 시작하는 이름 다음 애니메이션 이름은 유효하지 않은 값이다. -webkit-animation-name:1name; // 숫자로 시작하는 이름

-webkit-animation-name:@name; // 특수 문자로 시작하는 이름

# ■ 예제

다음 코드는 -webkit-animation-name 속성을 적용한 예제다. @-webkit-keyframes 속성에 설정한 이름인 test를 -webkit-animation-name 속성값으로 설정했다. 예제를 실행하면 test 이름을 갖는 @-webkit-keyframes을 실행한다.

```
<style>
    div {
        position: absolute;
        left: 100px;
        width: 100px;
        height: 50px;
        margin-left: -100px;
        background: #000;
        color: #fff;
        -webkit-animation-name: test;
        -webkit-animation-duration: 2s;
    @-webkit-keyframes test{
        from {
            left: 100px;
        }
        to {
            left: 300px;
        }
    }
</style>
<div>
    box
</div>
```

다음 링크와 QR 코드로 PC와 모바일에서 예제를 확인할 수 있다.

- 예제 실행 URL: http://jsbin.com/-webkit-animation-name
- 예제 실행 QR 코드



속성명	-webkit-animation-play-state						
요약 설명	애니메이션에 정지와 재생 기능을 추가한다.						
정의	애니메이션을 재생하거나 정지하도록 설정한다. paused 값을 적용하면 재생 중인 애니메이션 이 정지된다. running 값을 적용하면 정지한 애니메이션을 다시 재생한다. 애니메이션 재생이 멈추면 그 순간의 상태를 계속해서 나타난다. 다시 재생될 때에도 처음으로 돌아가지 않고 정 지된 상태에서 다시 재생을 시작한다.						
지원환경	0 15 이상	0	<b>6</b>	É	ı <del>``</del> ı		
	0	0	4.0 이상 〇	2.0 이상 〇	0		

## ■ 구문

-webkit-animation-play-state : ⟨play\_state⟩ [,...]

play\_state: 애니메이션 재생 여부.

• running: 애니메이션을 재생한다(기본값).

• paused: 애니메이션을 정지한다.

## 예제

다음 코드는 -webkit-animation-play-state 속성을 적용한 예제다. 예제를 실행하면 사각형이 화면 왼쪽과 오른쪽을 무한으로 반복하며 이동한다. 멈춤 버튼과 재생 버튼에는 각각 -webkit-animation-play-state 속성을 자바스크립트로 적용한 상태다. 멈춤 버튼을 클릭하면 -webkit-animation-play-state:paused를 적용해 애니메이션 재생을 정지한다. 재생 버튼을 클릭하면 -webkit-animation-play-state:running을 적용해 애니메이션을 재생한다.

## [예제 1-20] -webkit-animation-play-state 속성 활용 예저

```
<style>
   div {
      position: absolute;
      left: 100px;
      width: 100px;
      height: 50px;
      margin-left: -100px;
      background-color: #000;
      color: #fff;
```

```
-webkit-animation: test 2s linear infinite alternate;
    @-webkit-kevframes test{
        from{
            left:100px;
        }
        to{
            left:300px;
        }
    }
</style>
⟨button class="play"⟩
    재생
</button>
⟨button class="stop"⟩
    멈춤
</button>
<div>
    box
</div>
<script>
    $(document).readv(function() {
        $('.play').click(function() {
            // [재생] 버튼을 클릭하면 -webkit-animation-play-state:running 속성 적용
            $('div').css('-webkit-animation-play-state', 'running'); });
        $('.stop').click(function() {
            // [멈춤] 버튼을 클릭하면 -webkit-animation-play-state:paused 속성 적용
            $('div').css('-webkit-animation-play-state', 'paused');
        });
    });
</script>
```

다음 링크와 OR 코드로 PC와 모바일에서 예제를 확인할 수 있다.

• 예제 실행 URL: http://jsbin.com/-webkit-animation-play-state

● 예제 실행 QR 코드 **■** 🔭



속성명	-webkit-animation-timing-function						
요약 설명	애니메이션 재생 속도를 제어한다.						
정의	애니메이션을 재생하는 동안 단계별 재생 속도를 설정한다. 애니메이션의 프레임 사이의 재생 속도를 조절하는 속성이며, 애니메이션의 시작 부분과 종료 부분에 적용된다. 예를 들어 -webkit-animation-timing-function:ease-in-out 속성은 시작 부분의 재생 속도를 점점 빠르게 하고, 마지막 부분의 재생 속도를 점점 느리게 한다.						
지원환경	0 15 이상	0		Ć	<b>(</b>		
	0	0	4.0 이상 〇	2.0 이상 〇	0		

## ■ 구문

-webkit-animation-timing-function : \( \lambda \text{function} \rangle [,...]; \)

function: 애니메이션의 단계별 재생 속도.

• ease: 애니메이션이 서서히 시작해 빨라지다 중간부터 다시 느려져 서서히 종료된다(기본값).

• ease-in: 애니메이션이 서서히 시작해 점점 빨라진다.

• ease-out: 애니메이션이 빠르게 시작해 점점 느려진다.

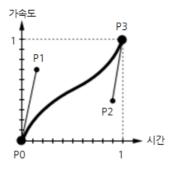
• ease-in-out: 애니메이션이 서서히 시작하고 서서히 종료된다. 시작할 때의 속도는 ease-in 값을 적용했을 때와 같고, 종료할 때의 속도는 ease-out 값을 적용했을 때와 같다.

• linear: 처음부터 끝까지 애니메이션을 일정한 속도로 재생한다.

• cubic-bezier (x1,y1,x2,y2): 큐빅 베지어<sup>cubic-bezier</sup> 곡선의 좌표 값을 직접 입력해 애니메이션 의 단계별 재생 속도를 설정한다.

큐빅 베지어 곡선: 큐빅 베지어 곡선은 네 개의 점으로 정의되는 3차원 곡선이다. 다음 그림에서 X축은 애니메이션 재생 시간을 의미하고, Y축은 진행 정도를 의미한다. P0은 초기 시간과 초기 애니메이션 상태(가속도)를 의미하며, P3는 마지막 시간과 최종 애니메이션 상태(가속도)를 의미한다. 이 점은 움직이지 않으며 고정이다. P1과 P2는 큐빅 베지어 곡선을 조절하는 점(핸들)이다. P1은 초기 시간과 초기 애니메이션의 곡선을 만드는 점으로 x1과 y1에 설정하는 좌표 값이다. P2는 마지막 시간과 최종 애니메이션 상태 곡선을 만드는 점으로 x2와 y2에 설정하는 좌표 값이다.

#### [그림 1-9] 큐빅 베지어 곡선



큐빅 베지어 값은 보통  $0 \sim 1$  값이지만 음수나 1보다 큰 값도 유효하다. 다음 코드의 경우 유효한 값이다.

```
      cubic-bezier(0.1, 0.75, 1.0, 0.1)
      // 0과 1 범위내 소수점

      cubic-bezier(0, 0, 1, 1)
      // 정수

      cubic-bezier(0.1, -0.6, 0.2, -1)
      // y축의 음수

      cubic-bezier(0.4, 1.5, 0.8, 5)
      // y축의 1.0보다 큰 값
```

잘못된 큐빅 베지어 곡선 값을 설정하면 기본값인 ease가 적용된다. -webkit-animation 속성에 [timing-function] 값을 잘못 지정하면 -webkit-animation 속성 자체 모두 무시하게 된다. 다음 코드의 경우 유효하지 않은 값이다.

```
      cubic-bezier(0, blue, 1,black)
      // 문자가 들어간 경우

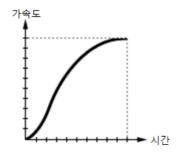
      cubic-bezier(0.3, 2.1)
      // 두 개의 값. 네 개의 값이어야 한다.

      cubic-bezier(-1.4, 0.3, -0.5, 2.1)
      // x축의 음수. x축은 0에서 1의 범위에 있어야 한다.

      cubic-bezier(2.45, 0.6, 4, 0.1)
      // x축의 1.0보다 큰 값. x축은 0에서 1의 범위에 있어야 한다.
```

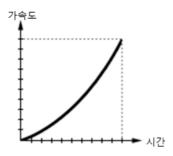
다음은 ease 값의 가속도 변화를 큐빅 베지어 곡선으로 표현한 그림이다.

[그림 1-10] ease 값의 큐빅 베지어 곡선: cubic-bezier(0.25, 0.1, 0.25, 1.0)



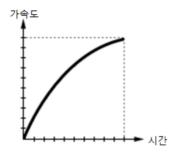
다음은 ease-in 값의 가속도 변화를 큐빅 베지어 곡선으로 표현한 그림이다.

[그림 1-11] ease-in 값의 큐빅 베지어 곡선: cubic-bezier (0.42, 0.0, 1.0, 1.0)



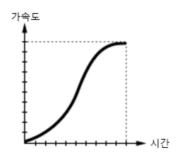
다음은 ease-out 값의 가속도 변화를 큐빅 베지어 곡선으로 표현한 그림이다.

[그림 1-12] ease-out 값의 큐빅 베지어 곡선: cubic-bezier(0.42, 0.0, 0.58, 1.0)



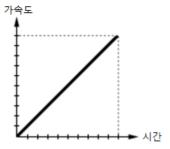
다음은 ease-in-out 값의 가속도 변화를 큐빅 베지어 곡선으로 표현한 그림이다.

[그림 1-13] ease-in-out 값의 큐빅 베지어 곡선: cubic-bezier (0.0, 0.0, 0.58, 1.0)



다음은 linear 값의 가속도 변화를 큐빅 베지어 곡선으로 표현한 그림이다.

[그림 1-14] linear 값의 큐빅 베지어 곡선: cubic-bezier (0.0, 0.0, 1.0, 1.0)



#### NOTE

http://cubic-bezier.com/ 사이트에서는 큐빅 베지어 곡선의 점을 직접 움직여 애니메이션 진행 속도를 확인할 수 있다. 큐빅 베지어 곡선으로 설정된 애니메이션 속도를 -webkit-animation-timing-function의 속성값인 ease, ease-in, ease-out, ease-in-out, linear 등과 비교할 수도 있다.

## ■ 예제

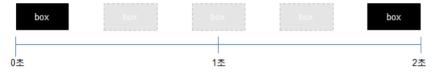
다음 코드는 -webkit-animation-timing-function 속성을 적용한 예제다. 예제를 실행하고 사각형을 클릭하면 사각형이 화면 왼쪽에서 오른쪽으로 이동한다. -webkitanimation-timing-function 속성값을 linear로 설정했기 때문에 사각형이 처음부터 끝 까지 일정한 속도로 이동한다.

## [예제 1-21] -webkit-animation-timing-function:linear 적용 예제

```
div {
    position: absolute;
    left: 100px;
    width: 100px;
    height: 50px;
    margin-left: -100px;
    background: #000;
    color: #fff;
    -webkit-animation-timing-function: linear;
    -webkit-animation-duration: 2s;
}
@-webkit-keyframes test{
```

다음은 [예제 1-21]을 실행해 사각형이 이동하는 과정을 시간의 흐름에 따라 표현한 그림이다. 사각형이 왼쪽에서 오른쪽으로 일정한 속도로 이동한다.

[그림 1-15] -webkit-animation-timing-function:linear 속성 실행 화면



다음 코드는 -webkit-animation-timing-function 속성값을 ease로 설정한 예제다. 예제를 실행하고 사각형을 클릭하면 사각형이 천천히 시작해 빠르게 이동한다. 중간부터 다시속도가 느려지면서 애니메이션이 천천히 끝난다.

#### [예제 1-22] -webkit-animation-timing-tunction:ease 석용 예제

```
<style>
    div {
        ...
        -webkit-animation-timing-function: ease;
        -webkit-animation-duration: 2s;
}
```

. . .

다음은 [예제 1-22]를 실행해 사각형이 이동하는 과정을 시간의 흐름에 따라 표현한 그림이다. 천천히 이동하기 시작해 빨라지다 중간 지점인 1초부터 다시 느려져 천천히 끝난다.

[그림 1-16] -webkit-animation-timing-function:ease 예제 실행 화면



다음 코드는 -webkit-animation-timing-function 속성값을 ease-in으로 설정한 예제다. 예제를 실행하고 사각형을 클릭하면 사각형이 천천히 이동하다 점점 빨라진다.

```
[예제 1-23] -webkit-animation-timing-function:ease-in 적용 예저
```

```
div {
    ...
    -webkit-animation-timing-function: ease-in;
    -webkit-animation-duration: 2s;
}
...
</style>
...
```

다음은 [예제 1-23]을 실행해 사각형이 이동하는 과정을 시간의 흐름에 따라 표현한 그림이다. 천천히 이동하기 시작해 점점 빨라진다.

[그림 1-17] -webkit-animation-timing-function:ease-in 속성 실행 화면

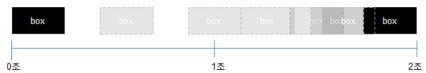


다음 코드는 -webkit-animation-timing-function 속성값을 ease-out으로 설정한 예제다. 예제를 실행하고 사각형을 클릭하면 사각형이 빠르게 이동하다 점점 느려진다.

```
div {
     ...
     -webkit-animation-timing-function: ease-out;
     -webkit-animation-duration: 2s;
}
...
</style>
...
```

다음은 [예제 1-24]을 실행해 사각형이 이동하는 과정을 시간의 흐름에 따라 표현한 그림이다. 빠르게 이동하다 점점 느려진다.

[그림 1-18] -webkit-animation-timing-function:ease-out 속성 실행 화면



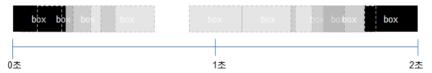
다음 코드는 -webkit-animation-timing-function 속성값을 ease-in-out으로 설정한 예제다. 예제를 실행하고 사각형을 클릭하면 사각형이 천천히 이동하기 시작하고 천천히 끝난다.

## [예제 1-25] -webkit-animation-timing-function:ease-in-out 적용 예제

```
div {
          ...
          -webkit-animation-timing-function: ease-in-out;
          -webkit-animation-duration: 2s;
    }
          ...
</style>
...
```

다음은 [예제 1-25]를 실행해 사각형이 이동하는 과정을 시간의 흐름에 따라 표현한 그림이다. 천천히 이동을 시작해 빨라지다 천천히 끝난다.

[그림 1-19] -webkit-animation-timing-function:ease-in-out 속성 실행 화면



### NOTE 참고. ease과 ease-in-out 비교

다음 링크와 QR 코드는 -webkit-animation-timing-function 속성값을 ease로 설정했을 때와 ease-in-out으로 설정했을 때 이동 속도를 비교할 수 있는 예제다. ease로 설정했을 때가 ease-in-out으로 설정했을 때보다 조금 더 빠르게 시작하고 중간 부분부터 느려지는 것을 확인할 수 있다.

- 예제 실행 URL: http://jsbin.com/-webkit-animation-timing-function2
- 예제 실행 OR 코드 **5** 및 ■



다음 코드는 큐빅 베지어 곡선을 설정해 -webkit-animation-timing-function 속성을 적용한 예제다. cubic-bezier (0,0,1,1)로 값을 설정했기 때문에 linear 값을 설정한 것과 같이 사각형이 일정한 속도로 이동한다.

## [예제 1-26] -webkit-animation-timing-function: cubic-bezier(0,0,1,1) 적용 예저

```
div {
     ...
     -webkit-animation-timing-function: cubic-bezier(0,0,1,1);
     -webkit-animation-duration: 2s;
}
     ...
</style>
...
```

다음 링크와 QR 코드로 PC와 모바일에서 예제를 확인할 수 있다.

- 예제 실행 URL: http://jsbin.com/-webkit-animation-timing-function
- 예제 실행 QR 코드



속성명	@-webkit-keyframes							
요약 설명	스크립트 없이 애니	스크립트 없이 애니메이션을 재생한다.						
정의		애니메이션을 재생할 각 프레임의 스타일을 설정한다. from 속성이나 0% 속성에 설정한 스타일에서 출발해 to 속성이나 100% 속성에 설정한 스타일로 점차 바뀌면서 애니메이션이 재생된다.						
지원환경	0 15 이상	<b>©</b>		É	i			
	0	0	4.0 이상 〇	0	0			

• 참고: 저자 테스트에서 사파리 5.1.7과 안드로이드 일부 버전에서 프레임 시간 비율을 퍼센트 단위(%)로 입력 시 적용되지 않았다(각 속성별 테스트 결과는 "부록.5 저자 테스트 리스트"를 참고한다).

# ■ 구문

animationname: 재생할 애니메이션 이름.

애니메이션을 적용할 요소의 -webkit-animation-name 속성값으로 설정한다.

keyframes-selector: 재생할 프레임별 시간 비율.

- 0%: 애니메이션의 시작 프레임이다.
- 100%: 애니메이션의 마지막 프레임이다.
- from: 애니메이션의 시작 프레임이다. 0%와 같다.
- to: 애니메이션의 마지막 프레임이다. 100%와 같다.

0%와 100% 사이에는 여러 개의 중간 값(%)을 설정해 프레임을 넣을 수 있다. 다음 코드는 중 가 값을 넣어 요소가 위에서 아래로 이동하게 하는 예제다.

```
@-webkit-keyframes mymove {
    0% {top: 0px;}
    25%{top:20px;}
    75%{top:175px;}
    100%{top: 200px;}
}
```

## 예제

다음 코드는 @-webkit-keyframes 속성을 적용한 예제다. 예제를 실행하면 동그라미가 화면 위에서 아래로 이동한다.

## [예제 1-27] @-webkit-keyframes 속성 적용 예제

```
<style>
    div {
        position: relative;
        top: 0;
        width: 100px;
        height: 100px;
        border-radius: 50px;
        background: #000;
        -webkit-animation: mymove 2s;
    @-webkit-kevframes mymove
        from {
            top: 0px;
        }/* 애니메이션 시작 */
            top: 200px;
        } /* 애니메이션 끝*/
    }
</style>
<div×/div>
```

top:0 값에 해당하는 화면에서 top:200px 값에 해당하는 위치까지 동그라미가 내려온다. 다음 링크와 QR 코드로 PC와 모바일에서 예제를 확인할 수 있다.

• 예제 실행 URL: http://jsbin.com/-webkit-keyframes

앞의 예제에 -webkit-animation 속성을 함께 적용해 공이 통통 튀는 애니메이션을 만들어 보겠다. 다음 코드와 같이 -webkit-animation-iteration-count 값과 -webkit-animation-timing-function 값, -webkit-animation-direction 값을 추가해 공이 통통 튀는 효과를 적용할 수 있다. -webkit-boder-radius 속성과 widht, height 속성을 적절히 이용하면 공이 땅에서 튕길 때 찌그러지는 효과도 적용할 수 있다.

```
<style>
    div {
        position: relative;
        top: 0;
        width: 100px;
        height: 100px;
        -webkit-border-radius: 50%;
        background: #000;
        -webkit-animation: ball 1s ease-in infinite alternate;
    @-webkit-keyframes ball
        from {
            top: 0px;
        }/* 애니메이션 시작 */
        to {
            top: 300px;
            width: 110px;
            height: 90px;
        } /* 애니메이션 끝*/
</style>
<div×/div>
```

다음 링크와 QR 코드로 PC와 모바일에서 예제를 확인할 수 있다.

- 예제 실행 URL: http://jsbin.com/-webkit-keyframes-ball
- 예제 실행 QR 코드



## NOTE 공이 튀는 효과 관련 속성

```
-webkit-animation
```

- -webkit-animation-iteration-count
- -webkit-animation-timing-funtion
- -webkit-animation-direction