



Hanbit
RealTime
105

웹킷 CSS 바이블

실무편

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



웹킷 CSS 바이블 **실무편**

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

웹킷 CSS 바이블 실무 편

초판발행 2015년 6월 16일

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 978-89-6848-766-8 15000 / **정가** 9,900원

총괄 배용석 / **책임편집** 김창수 / **기획·편집** 김상민

디자인 표지/내지 여동일, 조판 최송실

마케팅 박상용 / **영업** 김형진, 김진불, 조유미

이 책에 대한 의견이나 오타자 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. <네이버와 함께하는 HTML5 개발 전략> 웹킷 CSS 강의 경력이 있으며, 기술을 공유하는 것을 좋아하지만, 부끄럼도 많은 수줍은 개발자입니다.

지은이_ 강영주

현재 NHN Technology Services에서 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 쪽 뻗은 도로, 바람을 좋아하는 "바이크 라이더" 김영호입니다. 현재 NHN Technology Services에서 프론트엔드 개발자로 일하고 있습니다. 네이버 2012 대선, 2014브라질월드컵, 건강, 게임 등 다양한 서비스를 담당해왔으며 현재는 네이버 지식쇼핑 개발을 맡고 있습니다. 웹 기술에 관심이 많고 호기심 많은 근성 있는 개발자입니다.

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 Services UIT에서 프론트엔드 개발자로 일하고 있습니다. 디자이너에서 직업을 바꾼 지 2년째인 새내기 개발자이지만, 새로운 기술들을 사용하는 데 큰 즐거움을 느낍니다.

지은이_ 정다운

현재 NHN Technology Services에서 프론트엔드 개발자로 일하고 있습니다. 전 직장에서 디자이너, 액션스크립터로서의 경험을 통해 웹에서 그려지는 UI/UX에 대해 관심을 두게 되었고, 지금도 웹 기술의 여러 방면에 관심을 가지고 배우려고 노력 중입니다. 이 외에도 그림을 그리거나 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 적극적으로 사용하면서 마크업 업무의 효율성이 높아졌다. 한 사례로 마크업 개발자의 업무 중에는 성능 최적화를 위해 CSS 파일의 크기를 줄이는 작업이 있다. CSS 파일의 크기를 줄이는 작업은 필요하지 않은 속성을 삭제하는 것이다. 다양한 브라우저에 대응하기 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 한 권의 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 웹은 다양한 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 앎으로써 미래에 대응할 수 있기를 바란다.

지은이_ 조은

이 책에서 다루는 내용

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

책의 구성

- chapter 1. 웹킷이란: 웹킷이 무엇이며 왜 탄생했는지, 어떻게 브라우저에서 동작하는지 등 기본적인 지식을 다룬다.
- chapter 2. 웹킷 CSS로 구현한 실용성 있는 UI: 웹킷 CSS 속성을 실무에 적용할 수 있는 응용 예제를 다룬다. 이 장을 통해 프로젝트에 어떤 형태로 웹킷 CSS 속성을 사용할 수 있는지 더 효과적으로 이해할 수 있을 것이다.

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.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,initial-
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>
    <scriptX/script>
  </body>
</html>
```

지원 환경

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

 15이상			iOS	안드로이드
○	○	○	○	○

- 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 점유율이 높고 이 책의 웹킷 CSS 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.

이 책은 여러 명의 공동 작업으로 제작했다. 같이 고민하고 시간을 내준 NHN Technology Service의 박춘권 님과 본 저자 외 공동 저자인 강영주, 김대현, 김영호, 김훈민, 민경환, 양주희, 정다운, 조은 님께 감사의 말을 전한다. 또한, 이 책이 나오기까지 여러모로 도움을 주신 이상훈 님과 김태훈 님, 김미영 님, 소지훈 님, 이성희 님께도 감사의 말을 전한다.



NHN Technology Service WebKit TF

한빛 eBook 리얼타임은 IT 개발자를 위한 eBook입니다.

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

한빛이 지금 생각하고 추구하는, 개발자를 위한 리얼타임 전자책은 이렇습니다.

1 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.

3 독자의 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.

4 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 드리하고자 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 소유권함을 표시한 문구가 있다면 위법하게 사용하고 있을 가능성이 큼니다. 이 경우 저작권법에 따라 불이익을 받으실 수 있습니다.

다양한 기기에 사용할 수 있습니다. 또한, 한빛미디어 사이트에서 구매하신 후에는 횡수에 관계없이 내려받을 수 있습니다.

한빛미디어 전자책은 인쇄, 검색, 복사하여 붙이기가 가능합니다.

전자책은 오타자 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.

chapter 1 웹킷이란 ——— 001

- 1.1 웹킷의 역사 ——— 001
 - 1.1.1 웹킷의 탄생 ——— 001
 - 1.1.2 오픈소스 웹킷의 오늘과 내일 ——— 003
- 1.2 웹킷 CSS ——— 005
- 1.3 렌더링 엔진 ——— 007
 - 1.3.1 렌더링이란? ——— 007
 - 1.3.2 렌더링 엔진 ——— 007
 - 1.3.3 브라우저 점유율 ——— 008
 - 1.3.4 웹 표준 지원율 ——— 009
- 1.4 웹킷이 활용되는 곳 ——— 010
 - 1.4.1 웹 브라우저 ——— 010
 - 1.4.2 웹 OS ——— 011
 - 1.4.3 왜 웹킷인가요 ——— 012
- 1.5 모바일 웹 이야기 ——— 012
 - 1.5.1 사례1: CSS 속성으로 구현한 그라데이션 배경 ——— 012
 - 1.5.2 사례2: CSS로 구현한 사이드 메뉴 ——— 015
 - 1.5.3 모바일 웹 시장의 현재 ——— 018
 - 1.5.4 웹킷 CSS를 공부한다는 것은 ——— 020

chapter 2 웹킷 CSS로 구현한 실용성 있는 UI ——— 023

- 2.1 유리에 반사되는 순위 키워드 만들기 ——— 023
 - 2.1.1 기본 HTML과 CSS ——— 023
 - 2.1.2 애니메이션 효과 ——— 030
 - 2.1.3 반사 효과 ——— 033
 - 2.1.4 자바스크립트 적용 ——— 035
 - 2.1.5 이 예제로 배운 점 ——— 037

2.2	하트 모양의 프로필 사진 만들기	037
2.2.1	기본 HTML과 CSS	038
2.2.2	프로필 이미지에 마스크 적용하기	046
2.2.3	이 예제로 배운 점	049
2.3	분포 그래프 그리기	049
2.3.1	기본 HTML과 CSS	050
2.3.2	지도 위에 원 이미지 올리기	054
2.3.3	원 이미지가 표시될 범위 설정	057
2.3.4	이 예제로 배운 점	059
2.4	화면 크기에 반응하는 배경 이미지와 메뉴 만들기	060
2.4.1	기본 HTML과 CSS	062
2.4.2	화면 크기에 반응하는 배경 이미지 설정	066
2.4.3	같은 크기로 늘어나고 줄어드는 메뉴 만들기	068
2.4.4	이 예제로 배운 점	071
2.5	세로 막대 그래프 만들기	071
2.5.1	기본 HTML과 CSS	071
2.5.2	그래프 그리기	074
2.5.3	이 예제로 배운 점	078
2.6	데이터 정렬하기	078
2.6.1	기본 HTML과 CSS	079
2.6.2	유연한 상자 설정	085
2.6.3	자바스크립트 적용	086
2.6.4	이 예제로 배운 점	088
2.7	플라로이드 사진첩 만들기	089
2.7.1	기본 HTML과 CSS	089
2.7.2	사진첩 효과 적용하기	092
2.7.3	선택한 사진을 확대하기	094
2.7.4	이 예제로 배운 점	096

웹킷이란

웹킷은 브라우저 렌더링 엔진이다. 현재 애플 사파리에서 웹킷 2를 사용하고 있으며, 크롬과 오페라에서는 웹킷의 개선판인 블링크^{Blink} 엔진을 사용하고 있다. 이 장에서는 웹킷과 웹킷 CSS가 무엇인지, 그리고 웹킷의 미래에 대해 다룬다. 그 첫 번째로 웹킷의 역사부터 시작해 보자.

1.1 웹킷의 역사

처음부터 웹킷이 세간의 주목을 받지는 못했다. 웹킷의 탄생부터 오픈소스가 되기 까지, 그리고 거기에서 더 나아가 지금과 같은 명성을 얻기까지 프로젝트 내부에 많은 갈등이 있었다. 이와 관련한 이야기는 아직도 사람들의 입에 오르내리고 있다. 웹킷의 어제와 오늘 그리고 내일을 들여다보자.

1.1.1 웹킷의 탄생

웹킷의 시작은 1998년으로 거슬러 올라간다. 당시 애플은 사파리에 탑재할 렌더링 엔진으로 KHTML을 눈여겨보고 있었다. KHTML은 오픈소스 커뮤니티인 KDE^{K Desktop Environment}가 유닉스와 리눅스에 사용하려 개발한 브라우저 렌더링 엔진이다. 2001년 6월 25일 애플은 KDE의 KHTML(레이아웃 엔진)과 KJS(자바스크립트 엔진)를 포크해^{fork} 사파리 개발을 위한 프로젝트를 시작한다. 이 프로젝트의 이름이 바로 웹킷이다. KHTML은 깔끔한 설계, 가독성 높은 코드를 자랑했

고 웹 표준도 준수했다. 게다가 누구나 수정해서 새로운 제품을 만들 수 있는 오픈 소스였기 때문에 자사의 브라우저를 만드는 데 시간을 절약하고 싶었던 애플에게는 최고의 토대였던 셈이다. 이후 애플은 웹킷을 OS X로 이식하면서 KHTML을 WebCore로, KJS는 JavaScriptCore로 이름을 바꾸었다.

프로젝트 시작 초기에 서로를 칭찬하는 애플과 KDE의 모습은 아름다워 보였다. 하지만 그 후 2년간 애플과 KDE 사이에는 갈등이 생겨났다. 오픈소스 커뮤니티는 개방과 협력, 자유, 열정을 원동력으로 삼는다. 커뮤니티 참여자가 이러한 특성을 이해하고 있어야 한다. 하지만 모든 커뮤니티 구성원이 오픈소스 문화를 잘 이해하고 있는 것은 아니다. 애플이 그랬다. 이윤을 추구하는 기업으로서 오픈소스 커뮤니티와는 다른 고유문화를 가진 애플에 오픈소스 도전은 결코 쉬운 일이 아니었다.

애플이 웹킷 프로젝트를 진행하는 동안 KDE도 KHTML 프로젝트를 계속 진행했다. 애플이 WebCore의 코드를 변경하면, KDE가 변경 사항을 KHTML에 반영했다. WebCore와 KHTML의 개발 플랫폼이 서로 다르므로 이 작업은 쉬운 일이 아니었다. 애플은 WebCore를 만들면서 자사 제품에 특화된 기능을 많이 넣고 싶었고, 실제로 KHTML의 많은 코드를 비밀리에 변경했다. 이 일 때문에 KHTML과 WebCore 사이에 코드 호환성 문제가 발생했다. 얽힌 데 얽힌 격으로 애플의 폐쇄적인 태도가 KDE를 더 어렵게 만들었다. 심지어 애플은 코드를 변경하면서 관련 문서를 제대로 제공하지 않았다. 애플이 변경한 코드를 KDE 개발자가 확인하려면 비공개 서명을 해야 하는 등 복잡한 절차를 거쳐야 했고, 웹킷의 버그 관리 데이터베이스는 외부에서 접근할 수 없었다. KDE 개발자들은 WebCore의 변경 내용을 KHTML에 반영하는 데 애를 먹었다.

KDE는 오픈소스 정신에 어긋나는 애플의 태도가 마음에 들지 않았다. KDE 개발자인 크리스티안 티브리나^{Cristian Tibrina}는 자신의 블로그에 사파리와 KHTML의

관계를 쓰라린 실패로 규정하는 글을 올렸다. 웹킷 탄생 초기 애플에게 우호적이었던 KDE 개발자도 하나둘 불만을 토로했다. 애플은 좋은 오픈소스 구성원이 되고자 노력했지만, KDE의 눈에 들기에는 부족했다. 이런 이유로 2003년 1월 샌프란시스코에서 열린 맥 월드 엑스포의 기조연설에서 스티브 잡스가 WebCore를 오픈소스로 선언했을 때 오픈소스 커뮤니티 진영의 반응은 그저 냉랭했다.

하지만 갈등은 시간이 지나면서 사그라졌다. 먼저 손을 내민 쪽은 애플이었다. 좋지 않은 분위기를 부담스러워 한 애플은 KDE에 상호 협력 관계 증진을 위한 논의를 할 것을 제안했다. 둘은 대화를 통해 서로의 시각차를 좁혀 가며 협업을 방해하는 요소를 하나둘 개선했다. 이후 웹킷 팀과 KHTML 개발자의 협업이 증가했고, 많은 KHTML 개발자가 웹킷 리뷰어, 커미터가 되었다. KDE가 캉커러Konqueror 4.1 이후 버전에 웹킷을 탑재하고 있다는 사실로 둘의 관계가 개선되었음을 짐작할 수 있다.

1.1.2 오픈소스 웹킷의 오늘과 내일

2005년 7월 월드와이드 개발자 컨퍼런스에서 애플은 드디어 웹킷을 완전 오픈소스로 만들겠다고 발표한다. 이 발표는 자사의 단말기에 탑재할 브라우저 엔진을 찾고 있던 모바일 제조사들의 큰 관심을 끌었다. 발표 4개월 후인 2005년 11월, 노키아는 웹킷으로 만든 브라우저를 탑재한 휴대전화 S60을 출시하면서 가장 빠르게 움직였다.

스마트폰과 HTML5의 등장은 웹킷의 인기에 불을 붙인다. 2007년 1월 애플이 아이폰을 출시하며 모바일 생태계를 뒤흔들었다. 여기에 HTML5가 등장하면서 웹과 모바일 플랫폼이 사업적으로 중요한 영역이 되자 많은 기업의 관심이 웹킷에 쏠렸다.

그 중 대표적인 기업인 구글은 웹킷이 성장하는 데 지대한 영향을 끼쳤다. 구글은

2007년 11월 웹킷을 기반으로 하는 기본 브라우저를 탑재한 안드로이드 운영체제를, 2008년 3월에는 웹킷을 이용해서 만든 브라우저인 크롬을 발표한다. 안드로이드는 스마트폰 운영체제 시장에서, 크롬은 브라우저 시장에서 각각 점유율을 높여가며 웹킷 성장의 견인차 구실을 했다. 구글은 여기에서 더 나아가 2009년 7월, 크롬 OS 프로젝트를 발표하며 마이크로소프트와 애플이 지배해 온 PC 운영체제 시장에 도전장을 던진다. 크롬 OS는 저사양 넷북에서 고사양 PC까지 아우르는 것을 목표로 하는 웹 OS로서 웹킷을 기반으로 한다. 브라우저를 벗어난 웹킷, 애플에 칼날을 겨누는 웹킷, 크롬 OS의 등장은 세간의 큰 관심을 끌었다.

웹킷은 이 밖에도 RIM의 블랙베리^{BlackBerry} 기본 브라우저, 어도비 크리에이티브 스위트 5, 삼성 타이젠, 아마존 킨들 이북 등 다양한 영역으로 활동 반경을 넓히며 꾸준히 성장한다. 2010년 5월에는 기존의 웹킷을 새롭게 설계해서 웹 콘텐츠와 애플리케이션 UI와 서로 다른 프로세스에서 처리하는 것을 목표로 하는 웹킷 2 프로젝트가 탄생했다. 이 프로젝트는 2014년 5월 현재 진행 중이다. 2005년에 처음 애플이 웹킷을 오픈소스로 발표했을 당시에 인터넷 익스플로러가 사실상 시장을 독점하고 있었고 웹킷을 탑재한 브라우저는 사파리 단 하나였다는 것을 생각하면 웹킷의 성공은 분명 대단한 일이다.

하지만 높은 인기가 항상 좋은 것은 아니며 때로는 독이 되기도 한다. 웹킷이 그렇다. 웹킷은 상업적인 목적을 가진 회사의 참여율이 월등히 높다는 점에서 다른 오픈소스와 다르다. 웹킷에 가장 많이 기여한 애플과 구글은 서로 다른 목표와 방향성을 가지고 있다. 이는 다른 참여 기업도 마찬가지다. 여러 이해관계를 웹킷이라는 하나의 그릇에 담으려다 보니 코드가 점점 복잡해졌다. 빌드 시스템이 증가했고 개발 속도는 느려졌다.

결국 2013년 4월 구글은 웹킷을 포크해 새로운 브라우저 렌더링 엔진인 블링크 개발을 선언하고, 크롬 28에 블링크를 탑재한다. 크로미움 프로젝트 공식 블로그

는 웹킷에서 블링크로 넘어간 이유를 웹킷이 여러 아키텍처를 지원하면서 웹킷과 크로미움 프로젝트의 복잡성이 증가했고, 이에 따라 집단 혁신 속도가 저하되었기 때문이라고 밝혔다.

웹킷의 양대 커미터 중 하나인 구글의 이탈은 웹킷 커뮤니티에 큰 충격을 주었다. 오페라 진영도 웹킷을 버리고 블링크에 합류했고, 삼성은 모질라 재단과 차세대 브라우저 엔진 서보^{Servo}의 공동 개발을 발표했다. 월간 커밋 수도 전에 비해 눈에 띄게 줄었다.

하지만 상황을 부정적으로 단정하기는 어렵다. 웹킷의 인기가 한풀 꺾인 것은 사실이지만 크롬을 지원하기 위해 웹킷에 추가했던 코드를 제거함으로써 포트를 줄인 점은 오히려 긍정적이다. 게다가 다른 포트를 지원하지 않는 블링크가 상업용 브라우저 엔진을 원하는 기업에게 웹킷의 대안이 될 수 있을지는 아직 미지수다.

앞으로 웹킷은 어떻게 될까? 미래는 알 수 없다. 분명 구글의 이탈은 충격적이다. 하지만 구글 참여 이전에도 웹킷은 잘 성장했고, 애플도 결코 만만치 않은 기업이 기 때문이다.

1.2 웹킷 CSS

웹킷 CSS는 아직 표준이 아닌 CSS 속성을 웹킷 렌더링 엔진에서 먼저 사용할 수 있도록 지원하는 CSS를 의미한다. 대부분의 문법은 다음과 같다.

`-webkit-속성 이름 : 속성 값`

'속성 이름:속성 값' 형태의 문법은 기존 CSS와 동일하나 앞에 '-webkit-'이라는 문구가 붙었다. 이는 웹킷 엔진에서 지원하는 속성을 먼저 사용하겠다는 것을 의미하며, 이를 제조사 접두어^{vendor prefix}라고 부른다.

한 기술을 표준으로 만드는 과정은 신중히 진행해야 하므로 기술이 나오고 표준이 되기까지 짧게는 1~2년, 길게는 10년 가까이 걸린다. 그러나 브라우저 제조사는 되도록 빨리 새로운 기술을 자신의 브라우저에 적용하려 한다. 그래서 새로운 기술이 표준이 아니라는 것과 현재 이 브라우저에서만 사용할 수 있다는 것을 동시에 명시하려고 속성 이름 앞에 제조사 접두어를 붙인다.

제조사 접두어를 활용하는 곳은 다양하지만, CSS에서 가장 활발하게 사용하고 있다. 예를 들어 CSS3의 transition 속성은 웹 표준이 아니지만, 대부분의 렌더링 엔진은 이 속성을 지원한다. transition 속성을 모든 브라우저에서 사용하려면 다음과 같이 코드를 작성한다(하나의 속성을 모든 브라우저에서 사용하도록 여러 제조사 접두어를 붙여 만든 CSS 코드다).

```
.transition {
  -webkit-transition: 3s all ease-in; // 웹킷, 블링크(사파리, 크롬, 오페라)
  -moz-transition: 3s all ease-in; // 게코(파이어폭스)
  -ms-transition: 3s all ease-in; // 트라이던트(인터넷 익스플로러)
  -o-transition: 3s all ease-in; // 프레스토(오페라 15 이하)
  transition: 3s all ease-in // 표준
}
```

표준 CSS 속성과 제조사 접두어를 붙인 CSS 속성을 함께 설정할 때는 제조사 접두어를 붙인 CSS 속성을 먼저 선언하고 표준 CSS 속성을 마지막에 선언하는 것이 좋다. 표준 CSS 속성을 지원하지 않는 브라우저에서는 먼저 선언된 제조사 접두어가 붙은 CSS 속성이 적용되고, 표준 CSS 속성을 지원하는 브라우저에서는 맨 마지막에 선언된 표준 CSS 속성이 적용된다. 코드의 양이 늘어나 낭비가 많은 코드라고 생각할 수 있지만 모든 브라우저에서 동일한 인터랙션을 구현하려면 어쩔 수 없는 선택이다.

1.3 렌더링 엔진

렌더링 엔진(rendering engine)은 웹 콘텐츠(HTML, XML 등)와 포맷(CSS, XML 등)을 가져와 화면에 보여 주는 역할을 하는 소프트웨어다. 일반적으로 모니터나 프린터에 보이는 콘텐츠 영역을 그린다. 렌더링 엔진을 공부하면 내가 만든 페이지를 브라우저가 어떻게 보여주는지, 다른 부분은 어떻게 작동하는지 이해할 수 있고, 그 이해를 바탕으로 더 나은 웹 페이지를 개발할 수 있다.

1.3.1 렌더링이란?

렌더링이란 컴퓨터 프로그램을 사용해 모델로부터 영상을 나타내는 과정을 말한다. 프론트엔드 개발에서 렌더링은 모델(HTML, XML, 이미지)과 포맷의 복합 정보를 눈으로 볼 수 있게 바꿔 보여 주는 것을 의미한다.

렌더링 방식은 매우 다양하나 보편적인 방식은 GPU 같은 렌더링 장치로 3차원 표현을 2차원 표현으로 만드는 것이다. GPU는 CPU가 복잡한 렌더링 계산을 수행할 때 도움을 주는 프로그램이다.

웹 브라우저가 렌더링을 진행할 때는 대체로 CPU에서 렌더링 계산을 수행한다. 하지만 CSS 기술이 발전하면서 3차원 기술이나 캔버스(canvas) 등 복잡한 연산을 요구하는 기술이 나오기 시작했다. 복잡한 연산으로 CPU에 과부하가 걸리는 경우에는 GPU의 도움을 받거나 하드웨어 가속기의 도움을 받기도 한다.

1.3.2 렌더링 엔진

우리가 사용하는 브라우저가 다양하듯 렌더링 엔진도 다양하다. 사람들은 그중에 마음에 드는 것을 선택해 사용한다. 렌더링 엔진의 수는 정말 많지만 전 세계적으로 보편화된 렌더링 엔진은 크게 4가지이며, 전 세계적으로 유명한 5개의 브라우저에서 사용하고 있다.

다음 표에서 주요 웹 브라우저에서 어떤 렌더링 엔진을 사용하는지 정리했다.

[표 1-1] 주요 웹 브라우저와 렌더링 엔진

웹 브라우저	렌더링 엔진	제조사
사파리	웹킷(WebKit)	애플
크롬	블링크(Blink)	구글
오페라	블링크(Blink)	오페라 소프트웨어
인터넷 익스플로러	트라이던트(Trident)	마이크로소프트
파이어폭스	게코(Gecko)	모질라 재단

앞의 표에서 블링크라는 생소한 렌더링 엔진을 볼 수 있다. 블링크는 크롬에서 웹킷을 기반으로 만든 렌더링 엔진이다. 엔진의 구조가 웹킷과 같고 웹킷이 지원하는 내용 대부분을 함께 지원한다. 이 책에서 소개하는 속성도 대부분 지원한다.

렌더링 엔진의 성능에 따라 웹 브라우저의 성능이 좌우될 정도로 렌더링 엔진은 웹 브라우저에서 중요하다. 같은 렌더링 엔진을 사용해도 브라우저의 최적화 수준에 따라 성능이 다른 경우도 있다.

1.3.3 브라우저 점유율

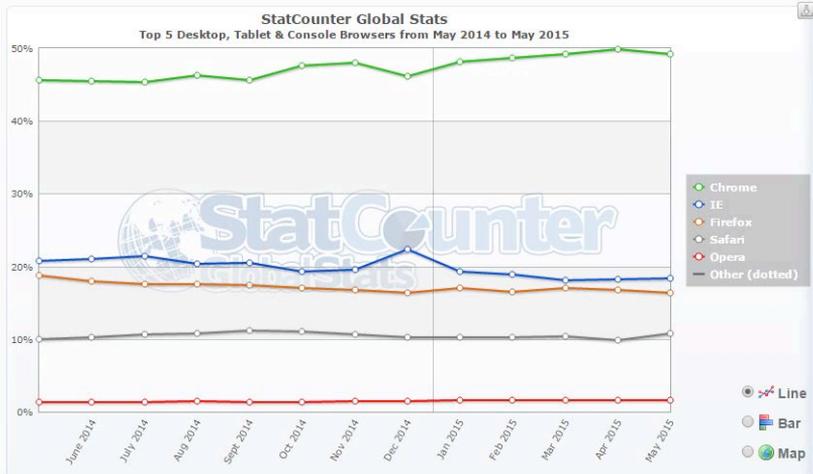
마이크로소프트의 인터넷 익스플로러는 전 세계에서 가장 많이 사용하는 브라우저다. 1998년 넷스케이프와의 브라우저 전쟁에서 승리한 뒤 전 세계 점유율 1위를 몇 년 전까지도 놓치지 않았으니 그 위상은 실로 대단하다 말할 수 있다.

그러나 당시 인터넷 익스플로러는 자신들만 지원하는 독자적인 기술을 계속 개발했고, 실력 있는 많은 개발자들이 웹 표준에서 벗어나는 인터넷 익스플로러의 행보를 걱정했다. 모질라 재단의 파이어폭스^{Firefox}가 오픈 웹을 지향하며 웹 표준을 안정적으로 준수하며 웹 브라우저 시장이 조금씩 바뀌었다. 이후 웹 표준을 더 완벽하게 준수하면서 성능도 우수한 구글의 크롬^{Chrome}이 출시되면서 인터넷 익스플

로리의 점유율은 하락세를 걷는다.

2014년 5월을 기준으로 전 세계 브라우저 점유율은 다음 그래프와 같다. 크롬의 점유율이 가장 높고 인터넷 익스플로러, 파이어폭스, 사파리, 오페라의 순서다.

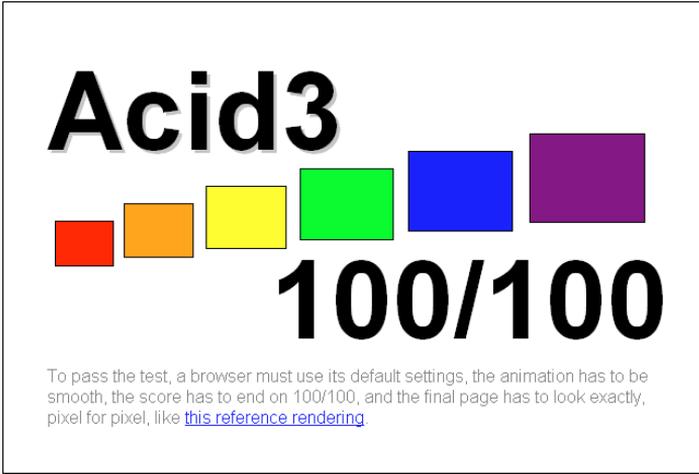
[그림 1-1] 전 세계 브라우저 점유율(출처: <http://gs.statcounter.com>)



앞으로 이 격차는 계속 벌어지겠지만, 브라우저의 점유율이 낮다는 것이 대응하지 않아도 된다는 것은 아니므로 실제로 웹 서비스를 개발할 때는 다른 브라우저도 반드시 살펴보고 개발해야 한다.

1.3.4 웹 표준 지원율

웹 브라우저의 웹 표준 지원율은 Acid 테스트로 확인한다. Acid 테스트는 웹 브라우저가 DOM과 자바스크립트에 관련된 정보를 얼마나 잘 준수하고 있고 이해하는지 알아보고 그 결과를 도표 형태로 보여 준다. Acid 테스트는 100개의 테스트와 참고 페이지와 비교를 통해 브라우저의 웹 표준 지원율을 수치로 알려 준다. 다음은 2014년 5월 기준 최신 버전의 사파리Safari를 Acid3 테스트로 확인한 결과다.



웹킷과 블링크 엔진은 웹 표준을 엄격히 준수하고 있고, 현재는 대부분의 렌더링 엔진이 웹 표준을 엄격히 준수한다. 웹 표준을 준수하면 모든 브라우저에서 문제 없이 비슷한 형태로 데이터 형태가 결정되며 이는 웹의 상호 운용성에 많은 도움을 준다.

1.4 웹킷이 활용되는 곳

지금까지 렌더링 엔진 자체를 살펴보았다면 웹킷 렌더링 엔진이 어디에서 쓰이고 활용되는지 알아보자.

1.4.1 웹 브라우저

웹킷은 브라우저에서 핵심적인 역할을 한다. 렌더링을 담당하며, 자바스크립트 해석 엔진과 상호 교류하면서 개발자가 만든 프로그램을 보여 주는 역할을 담당한다. 웹킷을 렌더링 엔진으로 사용하는 브라우저는 사파리다. 그 외에 웹킷과는 다르지만, 기원이 같은 블링크를 렌더링 엔진으로 사용하는 브라우저가 있다.

사파리

애플에서는 자신들의 주력 브라우저인 사파리에 웹킷을 탑재해 사용하고 있다. 사파리는 OS X과 iOS에서 가장 원활하게 작동하며 기본으로 제공하는 브라우저다.

블링크 엔진 브라우저

블링크 엔진은 웹킷과 다른 엔진이지만 기원이 같으므로 많은 부분에서 웹킷 CSS를 지원한다. 그런 의미로 블링크 엔진을 사용하는 브라우저를 함께 소개한다.

- 크롬: 크롬도 웹킷 엔진을 사용했지만, 크롬 28부터 웹킷의 소스를 바탕으로 만든 블링크 엔진을 사용한다. 두 엔진의 이름은 다르지만, 기원이 같으므로 많은 부분에서 공통점이 있다.
- 안드로이드 웹브라우저: 안드로이드 4.3까지의 웹뷰는 웹킷을 탑재해 사용했지만, 안드로이드 4.4부터 블링크 엔진을 사용한다.
- 오페라: 오페라 15부터 블링크 엔진을 사용한다.

1.4.2 웹 OS

웹이 꾸준히 발전하면서 웹을 기반으로 하는 OS가 제작되고 있다. 많은 웹 OS가 있지만 웹킷 엔진을 사용하면서 널리 알려진 웹 OS는 두 가지다.

삼성 타이젠

삼성에서 바다의 후속작으로 내놓은 웹 OS다. 크로미움^{chromium} 프로젝트를 기반으로 제작되었으며 공식적으로 출시는 되지 않았다. 하지만 다양한 곳에서 개발 행사를 개최해 많은 사람의 기대를 받는 웹 OS 중 하나다.

LG 스마트 TV

LG의 스마트 TV에 내장된 브라우저 또한 타이젠과 마찬가지로 크로미움을 기반으로 만들어졌다. 웹킷이 지원하는 속성을 대부분 지원한다. 스마트 TV라는 새로운 기기를 기반으로 한다는 점에서 많은 사람의 이목을 끌고 있다.

1.4.3 왜 웹킷인가요

많은 렌더링 엔진 중에서도 유달리 웹킷이 사람들의 시선을 끌게 된 이유는, 애플의 스티브 잡스가 플래시를 지원하지 않고 HTML5를 밀어주었기 때문이기도 하겠지만, 많은 사용자가 웹킷을 이용하고 그 사용자의 경험이 쌓였기 때문이라 생각한다. iOS에 기본으로 탑재된 사파리는 가장 쾌적한 모바일 웹 환경을 제공하고, 구글 크롬 또한 사람들에게 쾌적한 웹 환경을 제공했기 때문에 지금까지 발전할 수 있었다.

1.5 모바일 웹 이야기

웹킷의 성장에 가장 큰 영향을 준 것은 모바일 웹이다. 웹킷 기반 브라우저를 사용하는 애플과 구글의 모바일 기기가 대중적으로 보급되면서 CSS3가 확정되기 전에 웹킷 제조사 접두어를 붙여 모바일 웹에서는 마치 표준처럼 웹킷을 사용할 수 있게 되었다.

스마트폰 시대가 되면서 모바일 웹을 웹킷 CSS로 개발하는 것은 대세가 되고 있다. 몇 가지 사례를 살펴보자.

1.5.1 사례1: CSS 속성으로 구현한 그라데이션 배경

일반적으로 그라데이션 배경이나 이미지는 포토샵에서 만든 이미지를 사용했다. 그러나 gradient 속성이 발표된 후 CSS만으로도 그라데이션 효과를 표현할 수 있게 되었다. CSS 속성을 사용하면 디자인 리소스를 최소로 사용하고 CSS 코드만으로 구현할 수 있으므로 유지 보수가 쉽다. 그러나 gradient 속성은 아직 표준으로 확정되지 않아 대부분의 PC용 브라우저는 속성을 지원하지 않아 일부 PC 웹에서는 적용할 수 없다. 하지만 모바일 웹에서는 웹킷 제조사 접두어를 붙여 사용할 수 있다.

다음 그림에서 상품 섬네일 이미지 배경은 그라데이션으로 처리됐다. 모바일 웹에서만 제공하는 페이지라 웹킷 제조자 접두어를 붙인 `-webkit-linear-gradient` 속성으로 그라데이션을 구현할 수 있다.

[그림 1-3] CSS로 구현한 그라데이션 배경



이미지로 그라데이션을 구현하면 그래픽 작업에서 보이는 이미지와 모바일 기기에서 보이는 이미지의 느낌이 달라 작은 것이라도 수정해야 할 때마다 이미지를 직접 수정하고 반영해야 한다. 이미 서비스에 반영된 이미지라면 수정한 이미지를 적용하는 과정에서 사용자의 기기에 캐싱된 이미지도 고려해야 하고 이미지뿐만 아니라 CSS 파일도 수정해야 했다. 하지만 `-webkit-linear-gradient` 속성을 사용하면 수정 사항이 생겼을 때 CSS 파일만 수정하면 되므로 디자이너의 의견을 바로 반영해 확인할 수 있다. 이미지로 그라데이션 효과를 구현할 때보다 의사소통 비용도 줄일 수 있게 된다.

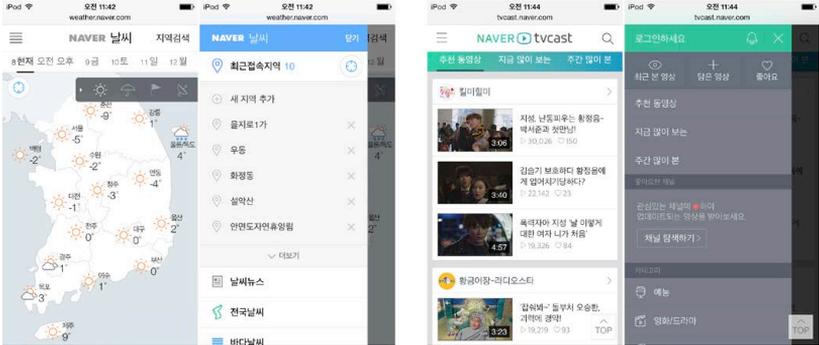
다음은 [그림 1-3]의 그레데이션 효과를 구현할 때 사용할 수 있는 코드다. 클래스 이름이 pd인 요소에 그레데이션 효과를 적용했다. pd 요소의 높이가 모바일 기기에 따라 달라지는 것을 고려해 ::before 가상 선택자와 ::after 가상 선택자로 요소의 위쪽과 아래쪽에 높이 100픽셀인 영역을 추가하고 그레데이션 효과를 적용했다. 이렇게 구현하면 요소의 위쪽 100픽셀과 요소의 아래쪽 100픽셀에 항상 그레데이션 효과가 나타난다.

```
<style>
    .pd {
        overflow: hidden;
        position: relative;
        height: 313px;
    }
    .pd::before, .pd::after {
        position: absolute;
        left: 0;
        right: 0;
        bottom: 0;
        z-index: 0;
        height: 100px;
        content: ' ';
    }
    .pd::before {
        top: 0;
        background: -webkit-linear-gradient(rgba(0,0,0,0.3), rgba(0,0,0,0) 40%);
    }
    .pd::after {
        bottom: 0;
        background: -webkit-linear-gradient(bottom, rgba(0,0,0,0.3),
        rgba(0,0,0,0) 40%);
    }
</style>
<div class="pd" >
    <div>
        ...
    </div>
</div>
```

1.5.2 사례2: CSS로 구현한 사이드 메뉴

최근에 모바일 웹에서 많이 사용하는 UX 기법 가운데 하나가 햄버거 버튼을 이용한 사이드 메뉴다. 햄버거를 연상시키는 세로줄이 세 개 있는 버튼을 누르면 화면의 다른 쪽에서 메뉴가 나타난다. 이때 나타나는 메뉴는 다음 그림처럼 세로로 긴 형태가 대부분이다.

[그림 1-4] 네이버 날씨 서비스와 TV 캐스트 서비스의 햄버거 버튼과 메뉴



만약 사이드 메뉴에 있는 메뉴의 개수가 상황에 따라 바뀌고 메뉴의 높이도 메뉴 개수에 따라 바뀌어야 한다면 flex 속성을 활용할 수 있다. flex 속성은 아직 표준이 아니라 모바일 서비스에 적용하려면 웹킷 제조사 점두어를 붙인 -webkit-flex를 사용한다.

다음 코드는 flex 속성으로 사이트 메뉴를 구현한 예제다. 클래스 이름이 menu 인 <nav> 요소 안에 <a> 요소가 3개 있다. <nav> 요소는 사이드 메뉴고 <a> 요소는 메뉴다. 사이드 메뉴의 높이는 화면 전체를 채워야 하고 메뉴의 높이는 똑같아야 한다. 사이드 메뉴인 <nav> 요소에 display:-webkit-flex와 height:100%를 적용하면 사이드 메뉴의 높이가 화면 전체를 채운다. 그리고 메뉴인 <a> 요소에 -webkit-flex:1을 적용해 메뉴의 높이가 똑같게 했다. 이렇게 웹킷 CSS 속성을 적용하면 사이드 메뉴의 하위에 메뉴가 추가돼도 메뉴의 높이는 항상 같아진다.

```
<style>
  html, body {
    margin: 0;
    padding: 0;
    height: 100%;
  }
  .side {
    height: 100%;
    background: rgba(0,0,0,0.5);
  }
  .menu a {
    width: 100px;
    background-color: #f0f0f0;
    border-top: solid 1px #666;
    padding-left: 10px;
    font-weight: bold;
    font-size: 14px;
    color: #000;
    text-decoration: none;
  }
  .menu a:first-child {
    border-top: 0;
  }
  .menu {
    display: -webkit-box;
    display: -webkit-flex;
    display: flex;
    -webkit-box-orient: vertical;
    -webkit-flex-direction: column;
    flex-direction: column;
    height: 100%;
  }
  .menu a {
    display: -webkit-box;
    display: -webkit-flex;
    display: flex;
    -webkit-box-flex: 1;
    -webkit-flex: 1;
    flex: 1;
    -webkit-box-align: center;
    align-items: center;
```

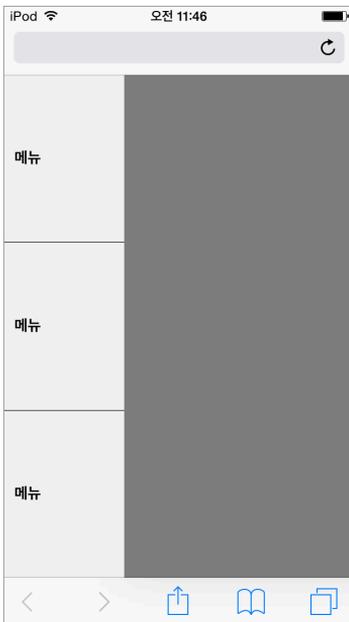
```

    }
</style>
<div class="side">
  <nav class="menu">
    <a href="#">span>메뉴</span></a>
    <a href="#">span>메뉴</span></a>
    <a href="#">span>메뉴 </span></a>
  </nav>
</div>

```

다음은 앞의 코드를 실행한 화면이다. 메뉴 3개가 같은 높이로 화면을 채우고 있다.

[그림 1-5] 메뉴가 3개인 사이드 메뉴



다음과 같이 CSS 코드를 수정하지 않고 메뉴만 더 추가해도 메뉴의 높이는 자동으로 같아진다.

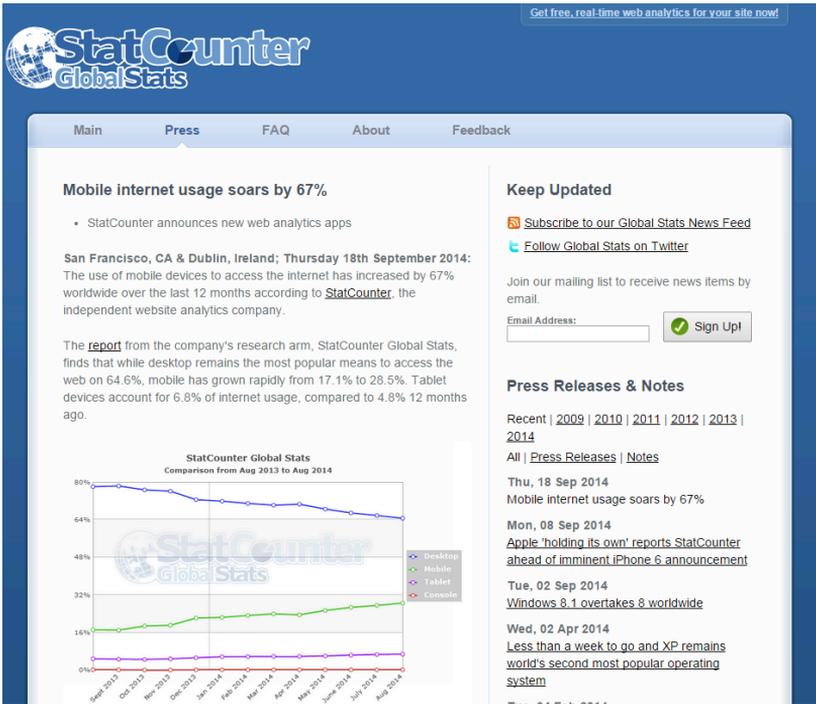
```
<nav class="menu">
  <a href="#">메뉴</a>
  <a href="#">메뉴</a>
  <a href="#">메뉴</a>
  <a href="#">메뉴</a>
  <a href="#">메뉴</a>
  <a href="#">메뉴</a>
</nav>
```

[그림 1-6] 메뉴가 6개로 늘어난 사이드 메뉴



1.5.3 모바일 웹 시장의 현재

웹킷 CSS 속성을 가장 많이 사용하는 모바일 브라우저의 사용률은 점점 증가하는 추세다. [스탯카운터 사이트](#)가 2014년 9월에 발표한 기사에 따르면 모바일 인터넷 사용자가 67% 증가했다.



지금은 모바일의 시대라고 해도 과언이 아니다. 일상생활에서 모바일 기기를 활용하는 사례도 늘어나고 모바일 웹 서비스도 다양해지고 있다. 모바일 웹 사용자가 증가했다는 것은 인터넷을 즐기는 사용자의 요구 사항이 변하고 있다는 것이다. 데스크톱이나 노트북으로만 인터넷 서비스를 사용하는 경험을 넘어 언제 어디서나 편하고 빠르게 인터넷 서비스를 사용하기 원한다. 최근 들어 필수 전자 제품으로 태블릿 PC와 스마트폰을 꼽는 사람도 있다.

그뿐 아니라 사용 연령대도 넓어지고 있다. 50, 60대 부모님 세대의 모바일 웹 사용률이 늘어나고 있다. 문자를 배우던 시대에서 모바일 인터넷과 앱을 사용하는 시

01 <http://goo.gl/xpkSvT>

대로 바뀌고 있다. 저자의 어머니는 계절마다 여행 다니시는 걸 좋아해 기차표를 예매할 때마다 저자에게 전화해 “기차 시간 좀 알아봐 줘” 하셨다. 그러나 지금은 지하철 노선도를 보거나 기차표를 예매하기 위해 인터넷 검색 서비스를 애용하고 있다. 인터넷의 사각지대였던 시니어 사용자의 반란이다. 이런 점에서 시니어 사용자의 행태와 요구 사항을 분석하고 사용성을 검증하는 사례와 시니어 대상 서비스가 늘고 있다. 다음은 서울시에서 시니어를 대상으로 만든 모바일 웹 사이트다.

[그림 1-8] 서울시 시니어 포털 50+ 모바일 서비스 화면(출처: senior.seoul.go.kr)



1.5.4 웹킷 CSS를 공부한다는 것은

이 책은 다가올 모바일 웹 시장을 준비하는 기반이 될 것이다. 앞에서 설명한 바와 같이 웹킷 엔진은 브라우저는 물론 스마트 TV와 게임기에도 사용하고 있어 웹 시장을 넘어 더 많은 곳에 진출할 기회가 생긴다는 것이다.

지금은 브라우저의 지원율이 높지 않거나 웹킷에서만 지원하는 속성이 대부분이

지만 이 책의 내용을 꾸준히 학습한다면 빠르게 발전해나가는 웹 시장에 적절히 대응할 수 있을 것이다.