

## 웹브라우저 초기 로딩시간 단축을 위한 HTML5 분석 및 연구

# HTML5 Analysis and Research for the Reduction of the Initial Load Time of a Web Browser

윤준수 · 박진태 · 황현서 · 표경수 · 문일영\*

한국기술교육대학교 컴퓨터공학과

Jun-soo Yun · Jin-tae Park · Hyun-seo Hwang · Gyung-soo Phyo · Il-young Moon\*

School of Computer Science and Engineering, Korea University Technology and Education, Chungcheongnam-do 330-708, Korea

### [요 약]

스마트폰을 비롯한 스마트 기기들의 출현으로 사용자들이 편리하게 이용할 수 있는 앱이 폭발적으로 증가하여 왔다. 그러나 다양한 스마트 기기 운영체제들의 등장은 앱 개발에 저해 요인으로 작용하고 있으며 이에 따라 다양한 플랫폼을 동시에 지원할 수 있는 HTML5에 대한 관심이 높아지고 있다. 따라서 브라우저를 이용한 HTML5는 크로스 플랫폼에 가장 획기적인 방법으로 사용되고 있다. 하지만 아직까지는 완벽한 호환성을 보이고 있지 못하며 브라우저의 환경에 따라 웹페이지의 초기 로딩시간이 차이가 나고 있다. 따라서 본 논문에서의 HTML5의 자바 스크립트와 CSS의 분석을 통해 브라우저별 초기 로딩시간을 많이 소요하는 원인을 파악하고 나아가 초기 로딩속도를 향상 시킬 수 있는 방안을 마련해 보고자 한다.

### [Abstract]

An app that users can conveniently use has been an explosive increase in the emergence of smart devices, including smart phones. However, the advent of various smart appliances operating systems, acting as an inhibitory factor to the development of the application. Accordingly, there has been a growing interest in HTML5 that can simultaneously support various platforms. HTML5 using the browser is used in the most innovative way of cross-platform. However, it does not show the complete compatibility for now. Depending on the browser of the environment, the difference between the initial load time of Web pages out. Therefore, to understand the cause of slowing down the browser-specific initial load time through the analysis of HTML5, JavaScript and CSS. Look for ways that can further improve the initial load rate.

**Key word** : HTML5, Javascript, Cascading style sheets, Initial load time, Browser.

<http://dx.doi.org/10.12673/jant.2015.19.5.440>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 17 September 2015; Revised 22 September 2015  
Accepted (Publication) 7 October 2015 (30 October 2015)

\*Corresponding Author; Il-Young Moon

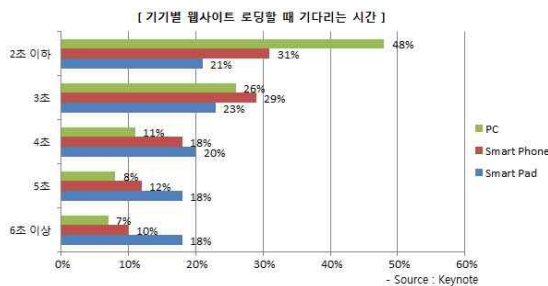
Tel: +82-10-2680-8221

E-mail: yuntn55@koreatech.ac.kr

## I. 서론

스마트폰의 등장은 우리 삶에 획기적인 변화를 가져왔다. 일상생활에서 의사소통 방식과 미디어 소비 방식을 변화시켰을 뿐만 아니라, 산업 구조의 지각 변동을 일으켰다[1]. 이제 사람들은 SNS 서비스를 통해 소식을 전하고 애플리케이션을 통해 광고를 한다. 스마트폰을 이용해 책과 신문을 읽고, 영화를 보고, 음악을 들을 수 있게 된 것이 불과 몇 년이라는 짧은 시간 안에 이루어졌다. 이와 더불어 스마트폰 앱 시장은 폭발적으로 증가하게 되었으며 그 종류를 수 백 가지가 된다[2]. 하지만 iOS, Android, Windows8 등 여러 스마트폰 플랫폼의 경쟁으로 인하여 스마트폰 앱 개발자들은 어려움을 겪고 있다. 각각의 플랫폼에 맞게 앱을 개발하여야 하며 그로인한 시간 및 인력 낭비가 심하게 되었다[3]. 이를 해결하기 위하여 크로스 플랫폼은 스마트폰에서도 관심이 높아지고 있으며 점차적으로 실생활에 적용되고 있다. 크로스 플랫폼은 실현 가능하도록 만든 건 HTML5이다. HTML5는 액티브 X, 플래시, 실버라이트 사용 없이도 동영상 재생이나 그래픽 구현이 가능하며 크로스플랫폼 호환성이 존재한다[4]. 또한 HTML5는 HTML, CSS, JavaScript로 구성되어 있으며 미디어 요소인 Audio, Video, Source, embed, track 등의 태그를 지원한다[5]. 이러한 HTML5의 발전은 플랫폼들의 경계를 없애는데 중요한 역할을 하였다. 하지만 아직까지도 각 플랫폼들 사이에 완벽한 호환성을 보이지 못하고 있다. 크롬에서 되는 기능이 익스플로러에서는 되지 않으며 각각의 플랫폼마다 접속 속도가 차이가 난다[6]. 특히 웹 브라우저의 초기 로딩 속도는 여러 플랫폼마다 다르게 측정되어 사용자들 사이에는 자신이 원하는 기능에 따라 플랫폼을 설치하여야 된다. 특히 스마트폰 환경에서는 낮은 CPU 와 낮은 Ram 그리고 느린 통신 속도로 인하여 웹페이지 초기 로딩 시간이 매우 느리게 측정된다[7]. 그림 1 과 같이 통계업체 Keynote에 따르면 대부분의 사람들은 2-3초 동안만 웹 페이지의 로딩을 기다리며 6초가 지나가면 웹 페이지의 접속을 포기한다. 느린 웹 페이지 초기 로딩속도로 인하여 웹 페이지의 접속을 포기하면 아마존과 구글과 같은 대형 웹 페이지에서는 막대한 양의 손실을 가져올 수 있다.

따라서 본 논문에서는 HTML5의 JavaScript와 CSS의 분석을



\* 출처 : keynote

그림 1. 기기별 웹사이트 로딩 시 기다리는 시간

Fig. 1. Device-specific waiting time when loading websites.

통해 브라우저별 초기 로딩시간을 많이 소요하는 원인을 파악하고 나아가 초기 로딩속도를 향상 시킬 수 있는 방안을 마련해 보고자 한다.

## II. 관련 연구

본HTML5 기술의 핵심적인 장점은 공개 표준 기반으로 이식성이 뛰어나고, 하이퍼링크 기반으로 동작하여 설치 및 업데이트 개념이 필요 없다는 것이다. 이식성은 HTML5 업계에서 큰 주목을 받는 가장 핵심적인 이유로 HTML5 앱을 만들면 어떤 디바이스에서든 실행이 가능함을 의미한다. 현재까지는 네이티브와 비교 시 느린 성능 이슈와 IOS, 안드로이드 등 서로 다른 플랫폼에서 브라우저나 웹뷰의 성능이 다른 성능 과현화가 많이 존재한다. 또한 브라우저나 웹뷰에서 지원하는 HTML5 기능이 서로 상이한 기능에 대한 이슈도 있다. 이러한 이슈들은 상당기간 유지될 것으로 보이나 구글, MS, 삼성전자, 애플, 등 세계의 주요 기업들의 노력으로 지속적으로 개선되고 있다.

### 2-1 W3C

W3C는 웹의 기반이 되는 HTML, 그래픽과 글꼴, 문서 포맷 등 사용자 환경영역을 정하는 일과 전자상거래 보안, 개인정보 유출방지, 음란물 차단, 전자서명, 인터넷의 특정 주소를 호출하는 표준규약인 HTTP와 차세대 규약인 HTTP-NG, TV와 웹의 융합에 따른 여러 기술적 문제 등 웹의 정보를 얻을 수 있도록 각 정보를 제공업체들에 다양한 기술적 지원을 권고한 일종의 가이드라인인 WAI 등의 프로젝트를 진행하고 있다. 또한 HTML5의 호환성을 높여 웹 페이지의 속도를 개선하는 연구를 진행 중이다.

### 2-2 구글 크롬

글로벌 웹 점유율에서 3분의1을 차지하고 있는 구글의 크롬 브라우저는 웹페이지의 로딩속도를 개선하기 위하여 어도비의 플래시 재생을 제한하는 기능을 탑재하였다. 어도비의 플래시는 인터넷 광고와 캔버스 앱으로 제작된 게임 등에 다양하게 사용되지만, 그만큼 프레임 수가 높아 전력 소모가 심하며 초기 로딩속도를 지연하는 원인으로 모바일 기기에서는 주로 지원하지 않는 경우가 늘어나고 있다. 구글의 결정으로 인해 플래시로 만들어진 광고 및 게임 등이 백그라운드 영역에서 돌아가지 않게끔 되어 디바이스의 전력상의 전원이나 로딩속도를 원활하게 구동 시키게 되었다. 어도비 플래시를 대체할 수 있는 기술로는 HTML5 등의 기술이 있다고는 하지만 아직 플래시 역할을 완벽하게 대체할 기술은 아니기에 구글에서도 지속적인 연구를 진행 중이다.

### 2-3 모질라 파이어폭스

모질라 파이어 폭스는 모질라 재단과 모질라 코퍼레이션이 개발하는 자유 소프트웨어 웹 브라우저로 윈도, 리눅스, OS X, 안드로이드를 지원한다. 모질라 파이어폭스는 웹 페이지를 렌더링하기 위해 게코 레이아웃 엔진을 사용하여 웹 표준을 구현하고 있다. 게코엔진은 넷스케이프 커뮤니케이션스에 의해 개발되었으나 초기에는 상당히 느린 속도였으며 W3C 표준에도 맞지 않았고 DHTML에 대한 지원이나 페이지를 새로 불러올 때의 페이지 요소 재정렬 면에서도 좋지 않는 성능을 보였다. 게코의 개발이 이어지면서 게코를 탑재한 다른 소프트웨어들이 나오기 시작하였고 아메리카 온라인에서는 콤퓨서브 7.0과 맥 OS X용 AOL에 게코를 탑재할 정도로 속도를 개선하였다.

### 2-4 오페라

오페라는 노르웨이, 오슬로의 오페라 소프트웨어가 개발하고 있으며 핵심 레이아웃 엔진인 프레스토를 장착하고 있다. 오페라는 스마트폰과 PDA를 위한 브라우저 분야에서 그들의 스몰스크린 렌더링 기술로 시장을 선도하고 있다. 또한 오페라는 다양한 기능을 통합해서 기본적으로 탑재하고 있으면서 같은 종류의 다른 프로그램에 비해 작고, 렌더링 속도가 가볍고 빠르다. 오페라 및 오페라 모바일에서는 웹 브라우저의 로딩속도를 감축하기 위하여 터보라는 기능을 제공하고 있는데, 이는 네트워크 연결이 느리거나 불안정한 환경에서 오페라 터보 기능을 토글키를 통해 활성화 시키면 웹페이지 서버측의 자료를 오페라 터버 서버에 우선 전송한 후 해당 정보를 오페라 터버 서버 상에서 압축하여 사용자에게 다시 전송하여 브라우저 사용자의 느린 네트워크 환경을 극복한다.

## III. 웹 브라우저 초기 로딩시간 개선 연구

웹 브라우저의 초기 로딩 시간 개선을 위하여 우선 웹 브라우저 별로 가장 많은 로딩 시간을 차지하는 속성을 찾아야 한다. 만약 가장 많은 로딩 시간을 차지하는 부분을 다른 언어로 대체하거나 약간의 기능을 변경하여 사용하게 된다면 초기 로딩 속도가 개선된다. 로딩 시간을 많이 차지하는 속성을 찾기 위하여 익스플로러, 크롬, 오페라, 사파리, 파이어폭스 등 총 5개의 브라우저를 통해 분석을 실시하였으며 모두 동일한 인터넷 환경과 측정 방법으로 속도를 측정하였다.

### 3-1 실험 환경

웹브라우저의 초기 로딩 속도를 정확하게 측정하기 위하여 아파치 HTTP 서버를 실험 PC에 설치하여 서버 PC를 만든 후 서버 PC에서 속도를 측정한다. 이렇게 하게 되면 인터넷 접속 속도에 따른 영향을 받지 않게 되고 정확한 측정이 이루어진다.

로딩시간 측정은 로딩이 시작되는 시간과 로딩이 완료되는 시간을 기록한다. 로딩완료시간에서 로딩시작시간을 빼주게 됨으로써 초기 로딩 시간을 측정한다.

실험을 진행할 웹 페이지는 최근에 가장 많이 사용되는 사이트들과 같은 기능의 웹페이지를 구성하여 로딩시간을 측정하고 점차 HTML5의 각 태그, JavaScript, CSS를 하나씩 제거하면서 가장 많이 시간이 감축되는 요소를 찾는다. 측정횟수는 각각 100회씩 실시하여 평균값을 이용한다.

### 3-2 HTML5 요소 분석

#### 1) HTML 속성, JavaScript, CSS 분석

HTML5의 속성은 <p>, <ul>, <link>, <input>, <label>, <img>, <section>, <img>, <video>, <div>등 으로 가장 많이 쓰이는 태그요소들을 제거하였다. 표 1은 5개의 브라우저별 각 요소를 제거 하였을 시 감소율을 나타낸 것으로 origin의 경우 원본 홈페이지로써 아무 요소를 제거하지 않고 측정한 결과 이다. 그림 2는 5개의 브라우저별 각 요소를 제거 하였을 시 측정된 시간을 나타낸 그래프 이다.

표 1. 브라우저 별 HTML 각 요소 제거 시 시간 감소율

Table 1. Time reduction rate when deleting each element of the browser by HTML.

element	reduction ratio				
	explorer	chrome	firefox	safari	opera
origin	0%	0%	0%	0%	0%
Script	94.94%	80.11%	81.92%	69.49%	87.51%
css	9.37%	3.09%	3.48%	35.76%	5.87%
p	8.54%	1.26%	5.87%	9.54%	3.21%
ul	10.04%	1.36%	5.33%	18.33%	0.77%
link	6.64%	5.36%	8.77%	42.97%	7.14%
input	2.81%	4.34%	8.9%	17.7%	7.86%
label	0.15%	3.33%	8.5%	17.11%	2.71%
section	1.92%	4.73%	6.2%	5.26%	3.32%
img	6.72%	5.38%	11.1%	9.09%	4.26%
video	4.85%	4.03%	9.57%	20.78%	7.19%
div	29.25%	3.82%	17.75%	48.19%	9.8%

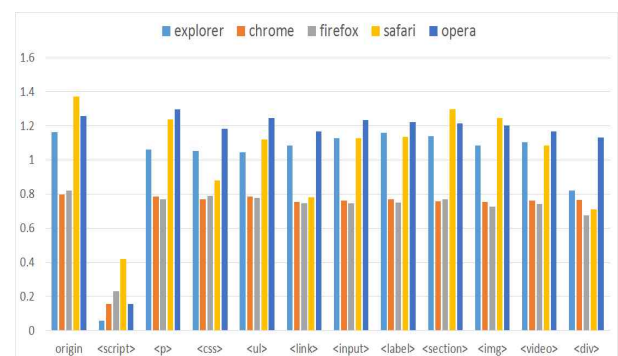


그림 2. 각 브라우저별 HTML 요소 제거 시 로딩시간

Fig. 2. Load time when deleting each browser-specific HTML elements.

각 브라우저마다 차이는 있지만 JavaScript를 제거하고 측정을 하였을 시 평균 82.79% 라는 감소율을 보였다. 이는 자바스크립트가 인터프리터 언어이기도 하는 동시에 많은 계산량을 가지고 있기 때문인 것으로 보인다. 또한 <div> 제거 할 때에도 많은 시간 감축을 보였으나 <div>를 제거 하게 된다면 원활한 UI를 제공해 주기 어렵기 때문에 <div>를 되도록 제거하지 않는 것이 좋다. CSS 역시 많은 시간 감축을 보였는데 CSS가 제거 되었을 경우도 UI 부분에 많은 영향을 미친다.

### 2) CSS의 각 속성 분석

웹 브라우저의 초기 로딩속도 저하에 많은 영향을 미치는 요소는 JavaScript, <div>, CSS 이다. JavaScript의 경우 제거를 하게 되면 웹사이트의 기능상에 문제가 생기며 <div>의 경우는 기본적인 태그 속성이라 더 이상 분석을 할 수 없다. 따라서 CSS의 각 속성들을 제거하여 추가적인 실험을 하였다. 실험 환경은 전과 동일하며 CSS의 제거 요소로는 background, margin, height, width, padding, position, border, outline, overflow, webkit-keyframes, webkit-animation, transition, float, display, color, text, box, z-index, font, cursor로 모두 웹사이트의 UI만 관여하는 속성들이다. 이 속성들을 제거한다고 하여도 웹페이지 기능에는 크게 문제되지 않는다. 표 2는 CSS의 각 요소를 제거 하였을 때의 시간 감소율을 나타낸 것이고 그림 3 는 요소 제거 시 로딩시간을 그래프로 나타낸 것이다.

표 2. 브라우저 별 CSS 각 요소 제거 시 시간 감소를  
Table 2. Time reduction rate when deleting each element of the browser by CSS.

element	reduction ratio				
	explorer	chrome	firefox	safari	opera
origin	0%	0%	0%	0%	0%
background	1.95%	3.27%	26.69%	44.72%	10.38%
margin	6.29%	6.3%	19.63%	6.79%	12.19%
height	10.9%	10.9%	17.27%	7.85%	4.51%
width	8.11%	11.51%	18.71%	19.09%	23.2%
padding	7.13%	6.18%	22.64%	7.31%	16.35%
position	7.27%	1.57%	79.9%	57.67%	6.86%
border	9.79%	7.75%	16.47%	11.42%	14.15%
outline	8.95%	9.45%	14.51%	7.82%	9.75%
overflow	11.88%	13.21%	16.64%	4.31%	19.69%
webkit-keyframes	5.45%	1.21%	14.95%	26.72%	22.76%
webkit-animation	7.27%	8.24%	17.93%	12.09%	16.98%
transition	3.49%	9.33%	19.96%	6.57%	6.86%
float	12.16%	9.57%	14.97%	7.86%	2.71%
display	8.11%	10.9%	14.05%	4.79%	6.41%
color	14.54%	1.57%	21.66%	11.75%	5.51%
text	7.13%	0.12%	20.42%	4.46%	12.37%
box	1.25%	12.72%	9.08%	3.26%	8.67%
z-index	6.43%	16.6%	16.03%	3.79%	23.12%
font	1.11%	0%	18.5%	1.72%	18.51%
cursor	4.05%	2.66%	19.19%	4.89%	18.51%

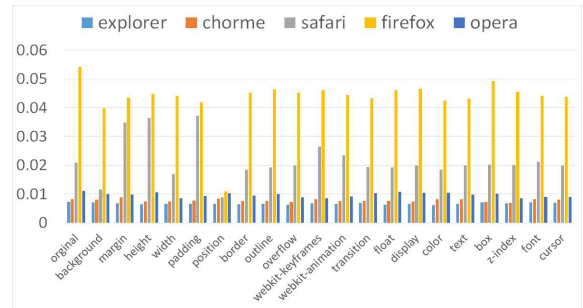


그림 3. 각 브라우저별 CSS 요소 제거 시 로딩시간  
Fig. 3. Load time when deleting each browser-specific CSS elements.

CSS 속성들의 시간 감축 역시 각 브라우저마다 차이는 있지만 HTML5 속성처럼 큰 감축을 보이는 속성은 없다. 다만 CSS의 webkit 속성을 제거 시 많은 감축을 보이고 있었다. webkit의 경우 웹 브라우저의 렌더링을 하는 레이아웃 엔진으로 주로 웹사이트의 움직임을 담당하는 부분이다. webkit을 이용하여 JavaScript의 움직임을 제어하게 된다면 웹페이지의 초기 로딩속도를 감축할 수 있을 것이다.

### 3-3 웹 페이지 초기 로딩속도 향상 연구

HTML5 각 요소들의 분석을 통하여 웹 브라우저의 초기 로딩 속도의 많은 영향을 끼치는 요소들을 파악했다. 그 요소 중 가장 큰 비중을 차지하는 요소는 JavaScript로써 JavaScript의 기능을 다른 언어로 구현하게 된다면 시간 감축 효과를 얻을 수 있다. 따라서 웹 페이지의 가장 많이 사용되는 기능 중 하나인 이미지 슬라이드를 JavaScript로만 구현한 한 개의 웹페이지와 CSS로만 구현된 한 개의 웹페이지를 통해 속도 측정을 진행 하였다. 또한 같은 방식으로 PC 환경이 아닌 저사양 모바일에서도 속도를 측정하였다. 그림 4의 위쪽 그림은 JavaScript로만 아래 그림은 CSS로만 구성된 이미지 슬라이드이다.

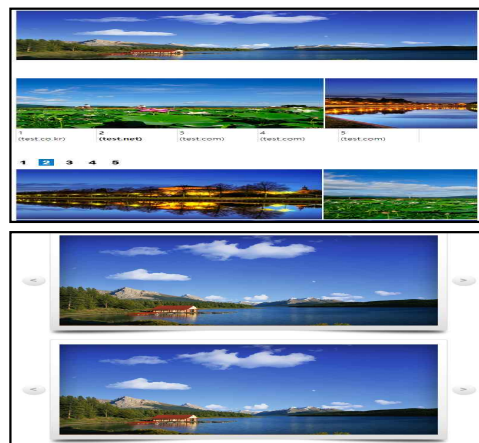
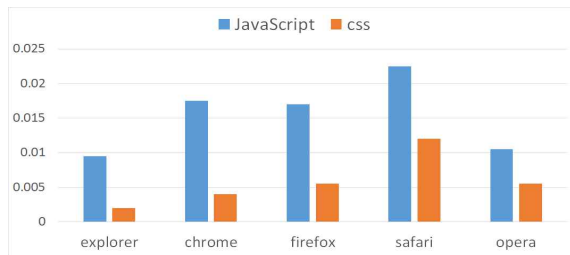


그림 4. JavaScript 와 CSS 로만 구현한 웹페이지  
Fig. 4. Web page that implements only JavaScript and CSS.

**표 3. PC 웹 브라우저에서의 JavaScript 와 CSS 속도 비교**  
**Table 3. JavaScript and CSS speed comparison from the Web browser of the PC.**

Browser	The average loading time (sec)		reduction ratio
	Image Slide composed in JavaScript	Image Slide composed in CSS	
explorer	0.0095	0.002	78.94%
chrome	0.0175	0.004	77.14%
firefox	0.017	0.0055	67.64%
safari	0.0225	0.012	46.66%
opera	0.0105	0.0055	47.61%

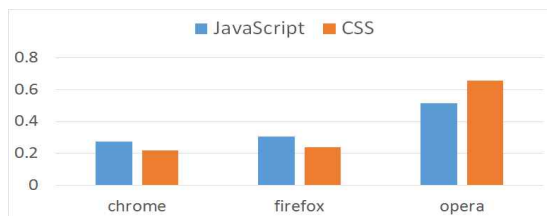


**그림 5. PC 웹 브라우저에서의 JavaScript 와 CSS 속도비교**  
**Fig. 5. JavaScript and CSS speed comparison from the Web browser of the PC.**

표 3은 웹 브라우저에서의 속도 비교 및 JavaScript로 구성된 이미지 슬라이드를 CSS로 구성하였을 때의 감소율을 나타낸다. 그림 5는 속도 비교를 그래프로 나타낸 것이다. 측정된 데이터를 바탕으로 평균 감소율은 63.6%가 측정되었다. 이는 웹 페이지에서 똑같은 이미지 슬라이드 기능을 수행하는데 있어서 JavaScript 보다 CSS가 더 빠르다는 것을 나타낸다. 뿐만 아니라 이미지 슬라이드의 전환효과도 CSS가 더 부드럽게 전환된다.

**표 4. 모바일 웹 브라우저에서의 JavaScript 와 CSS 속도 비교**  
**Table 4. JavaScript and CSS speed comparison from the Web browser of the mobile.**

Browser	The average loading time (sec)		reduction ratio
	Image Slide composed in JavaScript	Image Slide composed in CSS	
chrome	0.2725	0.218	20%
firefox	0.304	0.2395	21.21%
opera	0.5165	0.658	-27.39%



**그림 6. 모바일 웹 브라우저에서의 JavaScript 와 CSS 속도비교**  
**Fig. 6. JavaScript and CSS speed comparison from the Web browser of the mobile.**

표 4는 모바일 환경에서의 속도 비교 및 감소율을 나타내며 그림 6은 속도 비교를 그래프로 나타낸 것이다. 안드로이드 모바일 환경으로 측정을 하였기에 chrome, firefox, opera 등 3개의 브라우저로 측정을 하였다. 결과를 보면 opera를 제외한 나머지 브라우저에서 CSS로만 구성된 웹 페이지가 더 빠른 로딩 속도를 가지고 있는 것을 알 수 있다. opera의 경우 아직 안드로이드 환경에 적합한 환경을 제공하지 않아 속도를 측정할 시 데이터의 불안정성이 보였기에 예상과 다른 결과 값이 측정되었다.

**IV. 결 론**

스마트 폰의 획기적인 기능과 편리성으로 인해 스마트 폰 보급률이 기하급수적으로 늘어났다. 이와 더불어 스마트 폰 앱 시장은 폭발적인 증가를 보였다. 하지만 퍼편화된 플랫폼으로 인해 스마트폰 앱 개발은 힘들어 졌으며 앱 개발자들은 새로운 대안이 필요했다. 따라서 플랫폼에 상관없이 모든 플랫폼은 지원할 수 있는 HTML5이 각광받고 있으며 그 중요성 또한 높게 평가된다. 대부분의 웹 페이지가 HTML5로 만들어 졌을 만큼 HTML5는 널리 사용되고 있기에 웹 브라우저환경에서 HTML5를 이용한 웹 페이지의 초기 로딩시간 단축에 관한 연구가 세계 곳곳에서 활발히 진행 중이다. 본 논문에서는 HTML5의 분석을 통해 초기 로딩시간에 JavaScript가 가장 많은 영향을 미친다는 것을 알아내었다. JavaScript의 경우 HTML5에서 없어서는 안되는 중요한 연산 기능을 수행하기에 JavaScript를 제거하기 보다는 JavaScript를 대체할 수 있는 방안에 대해 연구를 진행하였다. 특히 JavaScript의 애니메이션 기능은 CSS로 대체하여 함으로써 웹 페이지의 초기 로딩 속도를 높일 수 있다. 웹 페이지의 애니메이션 효과는 JavaScript를 사용하지 않아도 CSS로 충분히 구현이 가능하다. 그 중 가장 널리 사용되는 이미지 슬라이드를 이용하여 JavaScript로만 구성된 이미지 슬라이드 웹페이지와 CSS로만 구성된 이미지 슬라이드 웹페이지를 비교 분석하였다. 속도 측정결과 CSS로만 구성된 이미지슬라이드 웹페이지가 더 빠르고 부드러운 애니메이션 효과를 만들어냈다. 즉 CSS로 애니메이션 효과를 만들게 되면 웹 페이지의 초기 로딩속도를 감축시킬 수 있다. 향후 본 연구를 바탕으로 범위를 넓혀 연구를 진행할 계획이다.

**참고 문헌**

[1] H. H. Choi and G. H. Kim, "A study on service migration in HTML5 based HTTP streaming environments," *The Journal of Korea Navigation Multimedia Society*, Vol. 14, No. 7, pp. 905-916, 2011.  
 [2] T. H. Kim and J. H. Kim, "Development of HTML5-based

lever learning webapp for cross-platform,” *The Journal of Korean Association of information Education*, Vol. 16, No. 2, pp. 189-199, 2012.

[3] H. G. Lee and J. H. Jin and S. C. Won and J. H. Cho and S. Y. Ryu and Y. S. Ko, “Scalable analysis framework for ECMA script,” *The Journal of Computing Science and Engineering*, Vol. 40, No. 5, pp. 283-289, 2013.

[4] W. S. Jung and E. J. Lee, “An optimization technique to improve readability of CSS,” *Journal of the Korea society of computer and information*, Vol. 15, No. 7, pp. 99-108, 2010.

[5] H. J. Lee and H. J. Yim and S. B. Lim, “3D stereoscopic representation of web content with CSS3 stylesheet,” *The*

*Journal of Computing Science and Engineering*, Vol. 20, No. 2, pp. 84-89, 2014.

[6] E. M. Lee and D. K. John, “The impact of HTML5 on web,” *The Journal of Communications of the Korean Institute of Information Scientists and Engineer*, Vol. 29, No. 6, pp. 55-60, 2011.

[7] B. H. Ahn and B. J. Kim, “HTML5: The current status of standards and the case studies,” *The Journal of Communications of the Korean Institute of Information Scientists and Engineer*, Vol. 30, No. 5, pp. 10-15, 2012.



**윤준수 (Jun-Soo Yun)**

2014년 8월 : 한국기술교육대학교 컴퓨터공학과 졸업 (공학사)  
 2014년 8월 ~ 현재 : 한국기술교육대학교 대학원 컴퓨터공학과 석사과정  
 ※관심분야 : WebOS, HTML5, BigData



**박진태 (Jin-Tae Park)**

2013년 8월 : 한국기술교육대학교 컴퓨터공학과 졸업 (공학사)  
 2015년 8월 : 한국기술교육대학교 대학원 컴퓨터공학과 (공학석사)  
 2015년 8월 ~ 현재 : 한국기술교육대학교 대학원 컴퓨터공학과 박사과정  
 ※관심분야 : WebOS, HTML5, JavaScript



**황현서 (Jun-Soo Yun)**

2014년 2월 : 한국기술교육대학교 컴퓨터공학과 졸업 (공학사)  
 2014년 2월 ~ 현재 : 한국기술교육대학교 대학원 컴퓨터공학과 석사과정  
 ※관심분야 : HTML5, BigData, iBeacon



**표경수 (Gyung-Soo Phyo)**

2015년 8월 : 한국기술교육대학교 컴퓨터공학과 졸업(공학사)  
 2015년 8월 ~ 현재 : 한국기술교육대학교 대학원 컴퓨터공학과 석사과정  
 ※관심분야 : WebOS, HTML5, Android



**문일영 (Il-Young Moon)**

2000년 2월 : 한국항공대학교 항공통신정보공학과 (공학사)  
 2002년 2월 : 한국항공대학교 대학원 항공통신정보공학부 (공학석사)  
 2005년 2월 : 한국항공대학교 대학원 정보통신공학과 졸업 (공학박사)  
 2004년 ~ 2005년 : 한국정보문화진흥원 선임연구원  
 2005년 3월 ~ 현재 : 한국기술교육대학교 컴퓨터공학부 부교수  
 ※관심분야 : 무선 인터넷 응용, 무선 인터넷, 모바일 IP