

모바일 웹환경에 필요한 HTML5&CSS3

Lecturer Profiles



강사 : 강요천

주요 약력

경희대학교 졸
한국은행 시프템 개발참여
기타 다수 SI프로젝트 및 컨설팅 수행
삼성 SDS 근무
오라클 교육센터 외부강사
비트교육센터 PM

저서 : (열혈강의)객체중심 JAVA
상상력과 HTMLG CSS3 Java-Script로 빛는 모바일 웹



HTML5 & CSS3

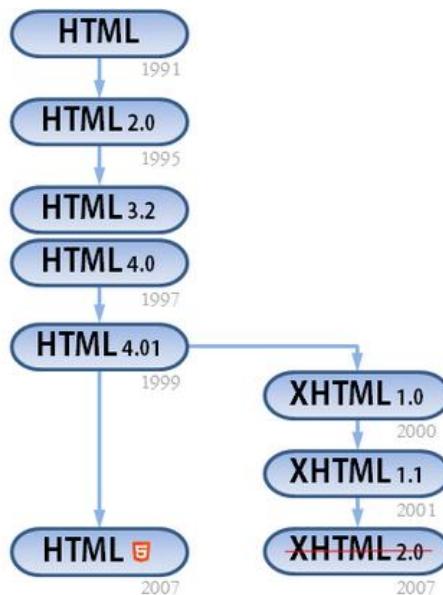
olc.oss.kr

강요천(cookie_00@naver.com)

목차

HTML5 & CSS3	1
HTML5 의 역사에 대한 해석	1
웹 기술의 진화	3
HTML5 의 구성 요소	6
HTML5 에서 가능한 일들	8
데이터의 확산과 시맨틱 태그	13
그래픽처리를 위한 API - <canvas>, SVG	16
WebGL with Three.js	22
Communication 관련 API	25
File API & Hardware	31
CSS3 와 새로운 웹 페이지	33
CSS3 의 기능적인 측면	41
웹 페이지에서 웹 앱으로	60
Touch Event	62
HTML5 와 위젯	64

HTML5 의 역사에 대한 해석



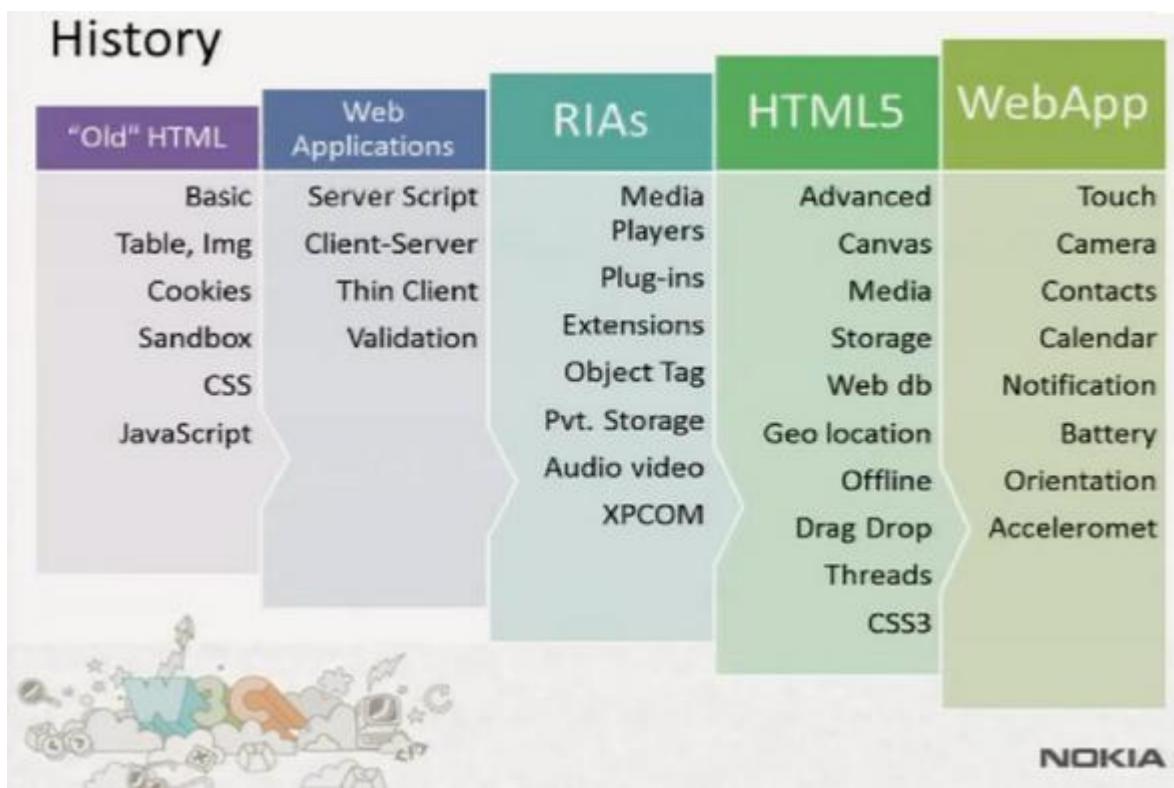
HTML5 이전의 버전이 1999년에 정의되었다는 사실을 생각해 볼 때, HTML5는 그저 단순한 태그나 처리에 대한 확장이 아니라, 새로운 패러다임을 수용하기 위한 움직임으로 해석할 필요가 있다.

HTML이 데이터 처리를 위한 마크업 언어였고, 이에 대한 엄격한 보완책으로 W3C에서는 XHTML을 표준안으로 제정하면서, 순수 마크업 방식이 HTML의 목표라고 생각되었던 시기가 있었다.

2000년대 중반에 RIA 관련 기술들이 많이 사용되면서 XHTML과 CSS, 자바스크립트의 간섭을 최소화하는 방식의 웹 프로그래밍이 유행하기 시작했고, 이런 변화에서 보면 HTML5라는 새로운 스펙이 아닌 XHTML과 같이 엄격한 마크업 스펙이 더 적합한 환경이다.

2000년대 중반이후에 HTML의 급격한 변화는 오히려 모바일 환경과 같은 외부 요인과 같이 다양한 환경에서 하나의 온전한 프로그래밍을 실행할 수 있는 플랫폼으로의 의미가 더 강해진 새로운 스펙을 장기적으로 수립하려고 하는 시도가 HTML5의 스펙

웹의 진화



(출처-<http://lazure2.wordpress.com/2011/11/26/1st-w3c-conference-for-web-developers-and-designers/>)

HTML5 가 가지고 있는 기술적인 API 들을 살펴보면 기존의 Application 들이 가져야만 했던 API 들이 상당 부분 포함되어 있는 것을 볼 수 있다.

특히 WebApp 의 영역에 있어서는 모바일 환경이나 N 스크린 환경을 고려하고 있는 것을 보여준다.

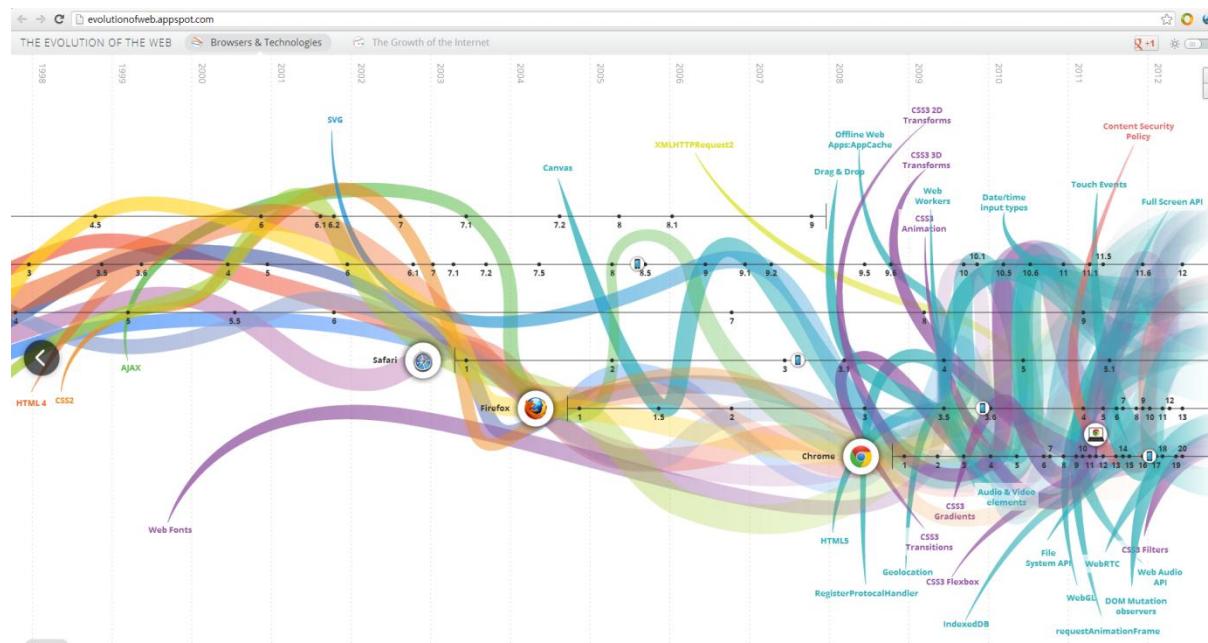
플랫폼 선점을 위한 경쟁

HTML5 에 대한 논의가 진행되던 시점이 IT 에서는 플랫폼에 대한 논의가 활발하던 시기라는 점을 생각해 볼 필요가 있다.

HTML5 스펙을 만족하는 브라우저나 디바이스에 대한 가능성과 기대가 가장 관련있던 브라우저의 벤더들이 적극적으로 HTML5 를 수용하는 계기가 됨. 이에 대해서 가장 늦게 대응하고 있는 브라우저는 IE (2011 년의 IE10 버전에서는 많은 부분이 개선 및 수용)

웹 기술의 진화

<http://evolutionofweb.appspot.com/>



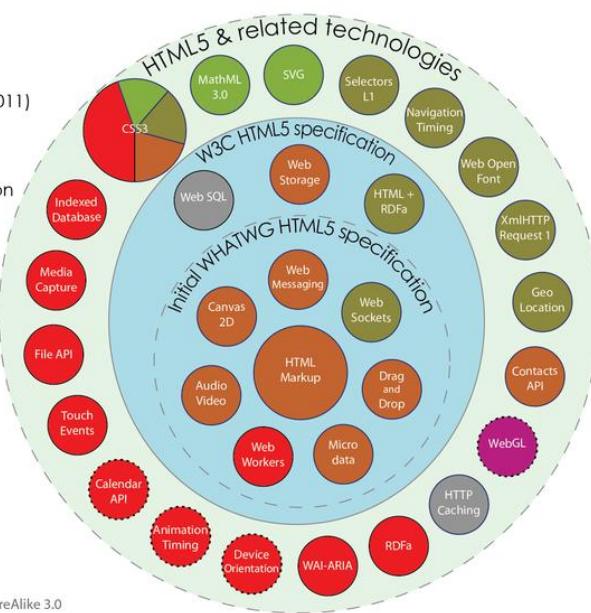
웹 기술이 시간이 지남에 따라 어떻게 변모해 왔는지를 HTML로 보여주는 화면, 현재 지원되는 모든 HTML5 표준 기술의 종류와 브라우저의 지원여부를 확인할 수 있음.

| HTML5에 대한 찬성과 반대

HTML5

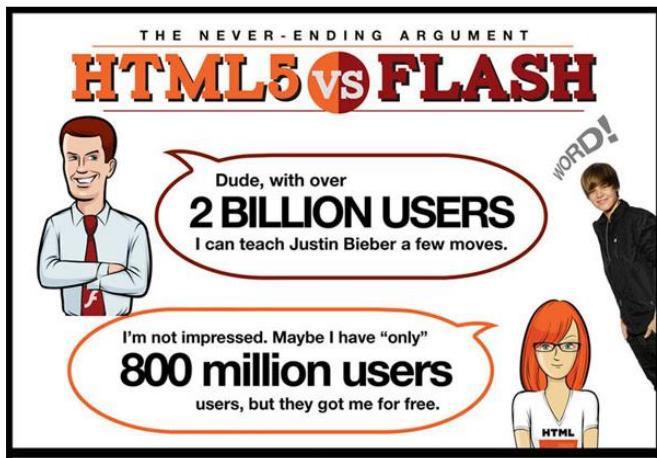
Taxonomy & Status (December 2011)

- W3C Recommendation
- Candidate Recommendation
- Last Call
- Working Draft
- Non-W3C Specifications
- Deprecated W3C APIs



By Sergey Mavrody 2011 | CC Attribution-ShareAlike 3.0

<http://www.wix.com/flash/infographic/html5-vs-flash.html>





Facebook의 CEO 마크 주커버그가 Disrupt 2012 컨퍼런스에 오랜만에 나와서 진행자와 이야기 하는 중에 HTML5에 대한 단상을 잠시 언급했습니다.

"When I'm introspective about the last few years
I think the biggest mistake that we made, as a company,
is betting too much on HTML5 as opposed to native...
because it just wasn't there. And it's not that HTML5 is bad.
I'm actually, on long-term, really excited about it.
One of the things that's interesting is we actually have more people on a daily basis using mobile Web Facebook
than we have using our iOS or Android apps combined. So mobile Web is a big thing for us."

HTML5 의 구성 요소



HTML5 에서의 마크업의 변화

마크업언어로의 HTML5: 좀 더 시맨틱한 의미를 전달할 수 있는 새로운 종류의 태그들을 지원하고, 검색엔진이나 자바스크립트의 객체에 맞는 개념을 표현할 수 있는 형태로 발전

<audio>와<video>등의 태그를 지원하면서 멀티미디어적인 부분이 강화됨.

<canvas>와 SVG 를 지원하기 때문에 높은 수준이나 낮은 수준의 그래픽처리 작업을 지원

CSS3 의 변화

2D, 3D 등에 대한 지원을 통해서 기존의 웹 페이지에서 불가능한 작업들이 가능하도록 지원, 이미지를 대신할 수 있는 새로운 속성들과, 애니메이션 등의 작업이 가능.

Flex box 모델 등을 통해서 보다 다양한 디자인 작업이 가능하도록 설계

모바일 환경에 대응할 수 있는 반응형 웹 제작이 가능하도록 지원.

자바스크립트의 변화

HTML5 가 되면서 겪는 가장 핵심적인 변화의 내용은 자바스크립트로 할 수 있는 API 들이 대폭으로 증가되었다는 점.

HTML5 의 플랫폼적인 모든 요소를 지원하기 위한 API 들이 제작되었고, 과거 이벤트와 URL 에 의존하는 방식이 아니라 하나의 Web App 의 제작이 가능하도록 온전한 하나의 Application 에 필요한 모든 기능을 자바스크립트를 통해서 지원.

WebWorker 등의 API 를 통해서 백그라운드 동작이 가능하도록 지원.

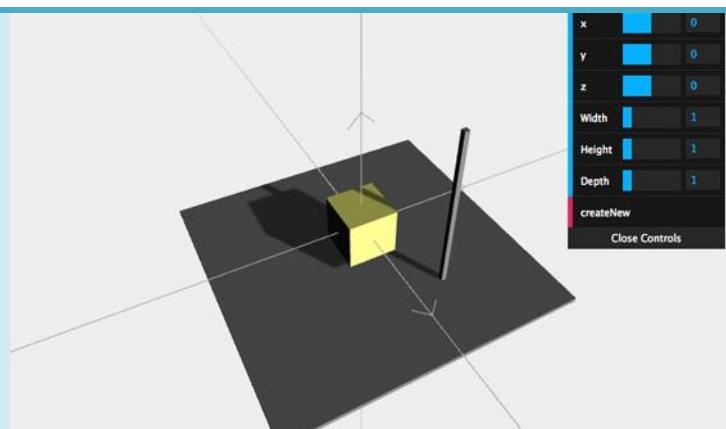
WebSocket 의 지원을 통해서 실시간 커뮤니케이션 지원 가능

XHR2 의 지원을 통한 크로스 브라우저의 접근 등의 작업이 가능

실제로 HTML5 에서 느껴지는 가장 큰 변화는 마크업적인 측면이 아니라, 자바스크립트적인 측면이 무척 강한 편

HTML5 에서 가능한 일들

WebGL 을 이용한 동적인 그래픽 처리



(출처- <http://fhtr.org/BasicsOfThreeJS/>)

Web Audio, Video 처리



File API 와 Drag And Drop 을 통한 처리

<http://www.inwebson.com/demo/html5-drag-and-drop-file-upload/>

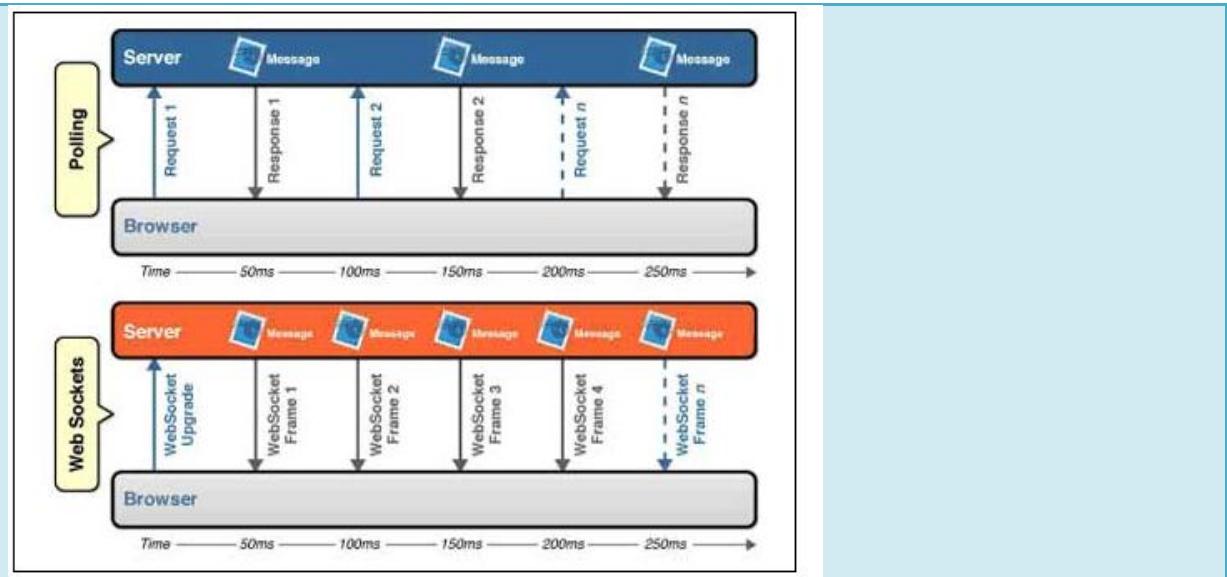


Notification 을 통한 사용자에 알림통보

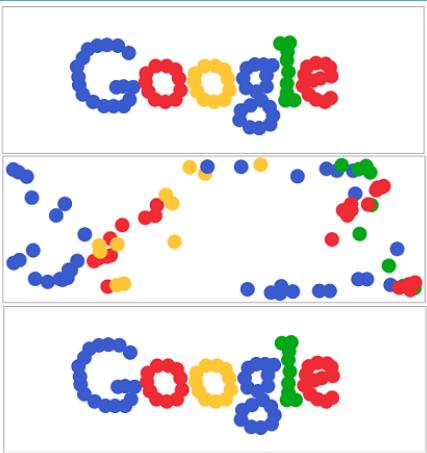
```
webkitNotifications.requestPermission(function() {  
    // Creates a normal notification  
    var notification = webkitNotifications.createNotification("http://mobilehtml5.org", "HTML5  
    notification", "Hi from the webpage");  
    notification.show();  
});
```

XHR2 를 이용한 다른 도메인간의 데이터 전송

WebSocket 을 통한 실시간 데이터의 처리



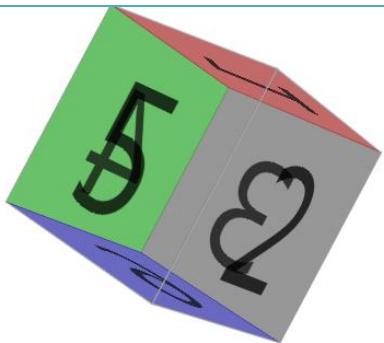
<canvas>와 SVG 를 이용하는 그래픽 처리



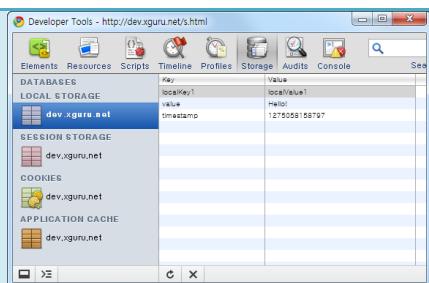
<http://www.jquery4u.com/html/html5-canvas-animation/>

WebWorker 를 통한 백그라운드 작업의 처리

CSS3 를 이용한 화면 전환 및 변형 처리



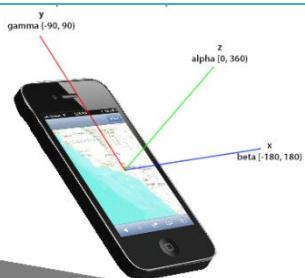
Web DB, Indexed DB, Storage 를 이용하는 데이터 처리



App Cache 를 이용하는 캐쉬 및 Offline 기능

Geolocation 을 이용한 위치 정보의 활용

Orientation 등의 디바이스 움직임을 이용하는 정보의 활용



Camera 와 연동하는 화면 처리 (WebRTC)

<http://my.opera.com/core/blog/2011/03/23/webcam-orientation-preview>

History API 를 이용하는 화면 히스토리 제어

데이터의 확산과 시맨틱 태그

Table → Div → Semantic Tags

기존 레이아웃의 문제점들에 대한 반성

- <table> 방식
 - 사용하기 쉽다는 장점, 복잡한 구성을 가지는 단점
 - 디자인적인 요소와 데이터적인 요소의 모호함
 - 다양한 미디어 환경에서의 적응이 불가능한 문제점
 - 참고(Why tables for layout is stupid <http://www.hotdesign.com/sebold/>)
- <div> 방식
 - 데이터적인 <div> 구분과 CSS를 이용하는 레이아웃의 배치를 활용하는 방식
 - 다양한 미디어 환경에 따른 CSS 선별 적용
 - 반복적인 <div>의 문제

```
01 <div id="menu">
02   <div class="selected">
03     <div class="graphicLeft">
04       <div class="graphicRight">
05         <a href="#">Home</a>
06       </div>
07     </div>
08   </div>
09   <div>
10     <div class="graphicLeft">
11       <div class="graphicRight">
12         <a href="#">About</a>
13       </div>
14     </div>
15   </div>
16   ...
17 </div>
```

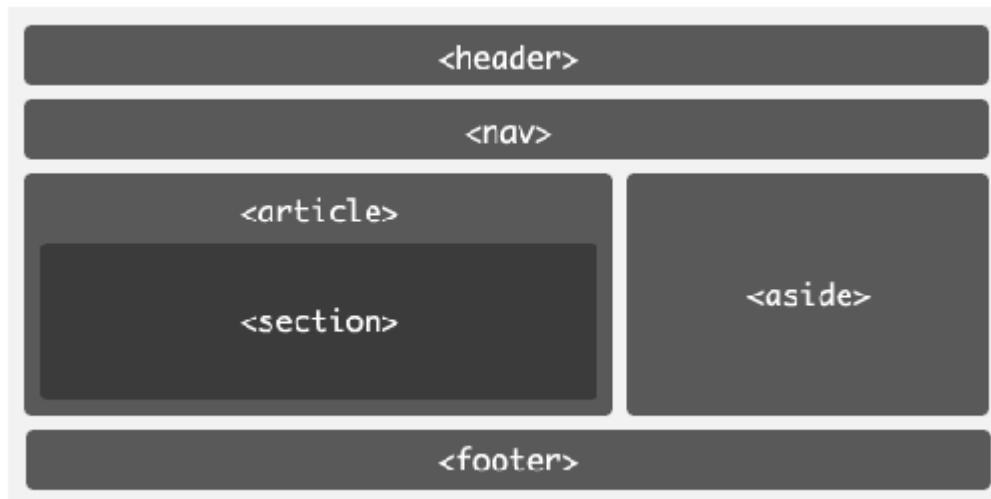
- Semantic 태그
 - <http://www.w3schools.com/tags/default.asp>
 - 구조에 따른 의미가 분리되므로, 검색엔진에게 유리하게 적용
 - 스크린 리더의 동작 시에 유리
 - 디바이스의 환경에 따른 적용 가능 – Multi column layout(CSS)

header

Hedaer content

footer	Footer content (end page)
section	web page section(area)
article	Independent content
aside	quotes
nav	navigation

HTML5 에서의 web page 가 원하는 구성



시맨틱태그의 의미

XML 과 달리 기존의 HTML 은 브라우저의 화면 처리와 무관하지 못한 상태에서 머물렀음.
ajax 사용 이후에 웹에서 필요한 것이 보다 XML 과 같이 구조적인 역할이 강조되는 상황에
대응할 수 있도록 설계 실제로 웹퍼블리셔들에게 보다는 검색엔진이나 데이터를 처리하는
입장에서 보다 의미를 가지는 내용이라고 볼 수 있음.

자바스크립트의 객체와 커스텀 데이터 속성

브라우저의 발전과 함께 자바스크립트의 영역이 점차 확대되어 가는 상황에서 태그 자체의 설정
만으로는 자바스크립트로 표현되는 객체를 제대로 표현하는데 한계를 가지게 됨.

자바스크립트의 경우에는 '객체'의 생성 자체가 다른 객체지향 프로그래밍 언어들과는 달리 Class 문법을 사용하지 않는 방식(prototype)을 사용하므로, 유연한 방식의 객체 처리가 가능.

객체의 순수한 데이터는 훼손하지 않고, 추가적인 정보를 추가하거나, 자바스크립트 객체가 가진 다양한 속성들을 반영하는데 있어서 사용 용도가 점차 증가하는 추세.

그래픽처리를 위한 API - <canvas>, SVG

고화질 처리용 SVG 와 저화질용<canvas>로 구분될 수 있음

- http://www.w3schools.com/html/html5_canvas.asp
- http://www.w3schools.com/html/html5_svg.asp

<canvas> 태그

- 2D 처리, 저화질의 그래픽 처리를 위한 태그
- 픽셀단위의 렌더링 처리, DOM 방식의 처리가 지원되지 않으므로, 화면의 수정시에는 redrawn 작업이 필요.
- Event Handler 를 지원하지 않음
- 작성된 이미지를 .png 나 .jpg 로 저장 가능

Browser	Details
Chrome	Supported in version 1.0 and greater
Firefox	Supported in version 1.5 and greater
Internet Explorer	Not supported
Opera	Supported in version 9.0 and greater
Safari	Supported in version 1.3 and greater

canvas 에 그려내는 간단한 원

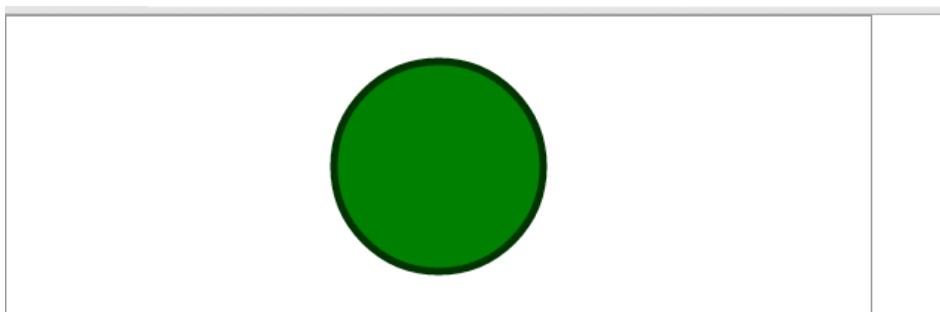
```
<!DOCTYPE HTML>
<html>
<head>
<style>
body{
margin:0px;
padding:0px;
}
#myCanvas{
border:1px solid #9C9898;
```

```

}
</style>
</head>
<body>
<canvas id="myCanvas" width="578" height="200"></canvas>
<script>
var canvas = document.getElementById('myCanvas');
var context = canvas.getContext('2d');
var centerX= canvas.width/ 2;
var centerY= canvas.height/ 2;
var radius = 70;

context.beginPath();
context.arc(centerX, centerY, radius, 0, 2 * Math.PI, false);
context.fillStyle= 'green';
context.fill();
context.lineWidth= 5;
context.strokeStyle= '#003300';
context.stroke();
</script>
</body>
</html>

```



회전

```

<!DOCTYPE HTML>
<html>
<head>
<style>
body{
margin:0px;
padding:0px;
}
#myCanvas{
border:1pxsolid#9C9898;
}

```

```

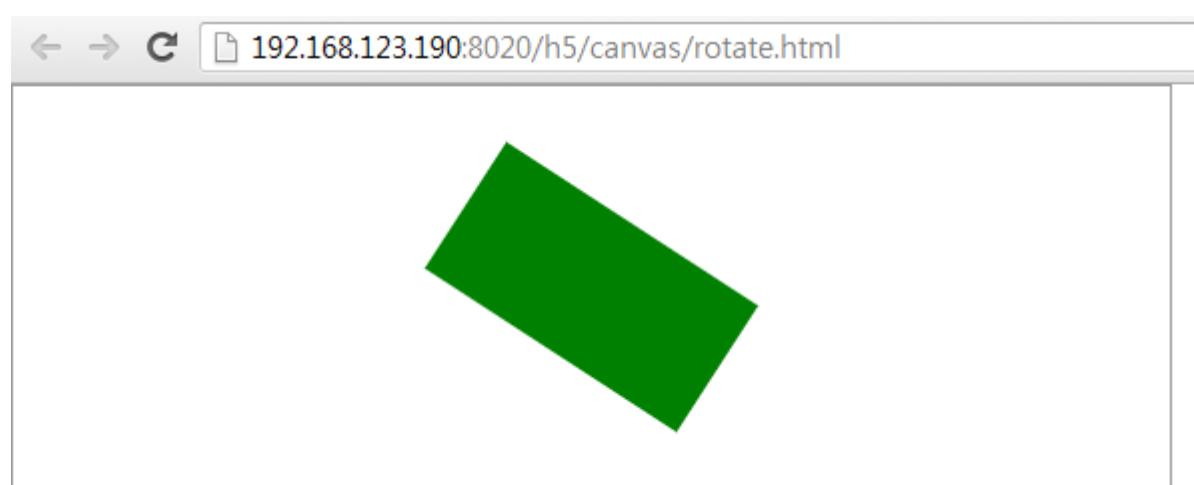
</style>
</head>
<body>
<canvas id="myCanvas" width="578" height="200"></canvas>

<script>
varcanvas = document.getElementById('myCanvas');
varcontext = canvas.getContext('2d');
varrectWidth= 150;
varrectHeight= 75;

context.translate(canvas.width/ 2, canvas.height /2);
context.rotate(10);

context.fillStyle= 'green';
context.fillRect(rectWidth/ -2, rectHeight /-2, rectWidth, rectHeight);
</script>
</body>
</html>

```



JPG 로 Canvas 내용 저장하기

```

<!DOCTYPE HTML>
<html>
<head>
<style>
body{
margin:0px;
padding:0px;

```

```

}

#canvasImg{
border:1pxsolid#9C9898;
}

</style>
</head>
<body>
<canvas id="myCanvas" width="578" height="200"
style="display:block;"></canvas>

<script>
var canvas = document.getElementById('myCanvas');
var context = canvas.getContext('2d');

// draw cloud
context.beginPath();
context.moveTo(170, 80);
context.bezierCurveTo(130, 100, 130, 150, 230, 150);
context.bezierCurveTo(250, 180, 320, 180, 340, 150);
context.bezierCurveTo(420, 150, 420, 120, 390, 100);
context.bezierCurveTo(430, 40, 370, 30, 340, 50);
context.bezierCurveTo(320, 5, 250, 20, 250, 50);
context.bezierCurveTo(200, 5, 150, 20, 170, 80);
context.closePath();
context.lineWidth= 5;
context.fillStyle= '#8ED6FF';
context.fill();
context.strokeStyle= '#0000ff';
context.stroke();

// save canvas image as data url (png format by default)
var dataURL= canvas.toDataURL();

// set canvasImg image src to dataURL
// so it can be saved as an image
document.getElementById('canvasImg').src= dataURL;

</script>
</body>
</html>

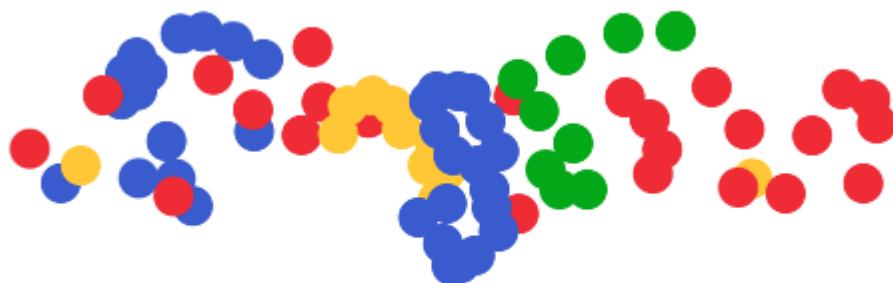
```

애니메이션 처리를 위한 새로운 프레임 제작

```
// request new frame
```

```
requestAnimationFrame(function() {
    animate(lastTime, myRectangle);
});
```

<http://jquery4u.com/animation/html5-canvas-animation/>



[TUTORIAL](#)

[DOWNLOAD](#)

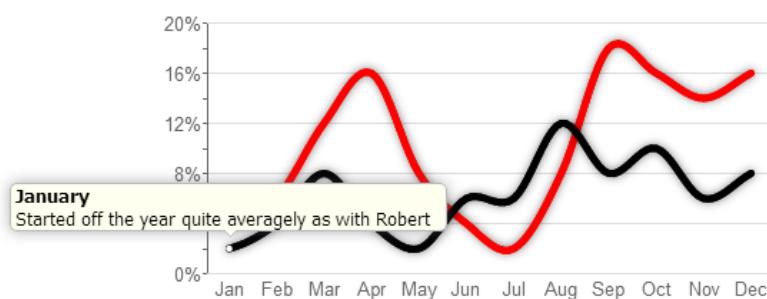


RGraph - Free HTML5/Javascript charts and graphs

RGraph produces interactive charts using the HTML5 canvas tag

[Home](#) [Download](#) [Examples](#) [Documentation](#) [Support](#) [License](#) [Blog](#) [About](#)

520 751

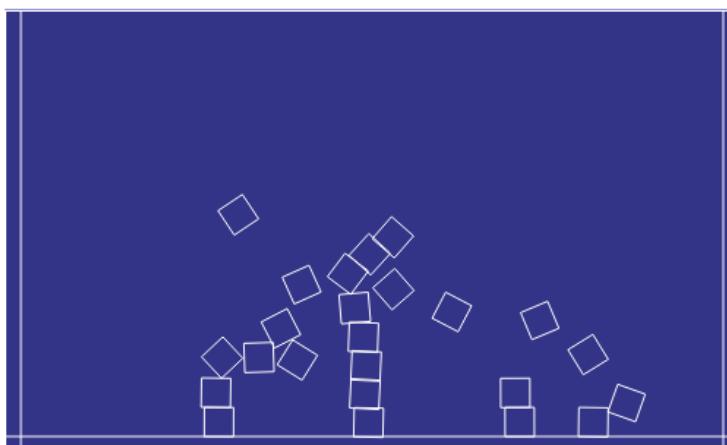


<svg>

- 높은 해상도의 작업을 처리해야 하는 경우에 주로 사용
- 이벤트 처리가 지원되기는 하지만, 각 구성 노드들이 따로 따로 렌더링 되기 때문에 많은 요소들이 있는 경우에는 속도의 저하의 우려가 있다(특히 모바일 환경)
- 각 요소에 이벤트 처리 등이 가능
- <http://www.html5rocks.com/en/tutorials/raphael/intro/>
- <http://jsdraw2dx.jsfiction.com/#examples>

물리엔진용 자바스크립트

<http://box2d.org/>



WebGL with Three.js

<http://mrdoob.com/131/clouds>

OpenGL ES 2.0 의 자바스크립트 버전

http://www.html5rocks.com/en/tutorials/webgl/webgl_fundamentals/ 등을 통해서 기본적인 예제를 볼 수 있으나 HTML5 이외에도 OpenGL에 대한 개념이 필요하므로, 좀 더 간단히 처리하기 위해서 Three.js 를 이용하는 것이 편리(JavaScript 3D library 라고 인식할 수 있다).



Three.js 를 이용하는 cube

```
<script>

    varcamera, scene, renderer;
    varmesh;

    init();
    animate();

    function init() {

        renderer = new THREE.WebGLRenderer();
        renderer.setSize( window.innerWidth,
        window.innerHeight );
```

```

document.body.appendChild( renderer.domElement );

        //

        camera = new THREE.PerspectiveCamera( 70,
window.innerWidth/ window.innerHeight, 1, 1000 );
        camera.position.z= 400;

        scene = new THREE.Scene();

        vargeometry = new THREE.CubeGeometry( 200, 200,
200 );

        vartexture =
THREE.ImageUtils.loadTexture( '../images/a_0.jpg' );
        texture.anisotropy= renderer.getMaxAnisotropy();

        varmaterial = new
THREE.MeshBasicMaterial( { map: texture } );

        mesh = new THREE.Mesh( geometry, material );
scene.add( mesh );

        //

        window.addEventListener( 'resize',
onWindowResize, false );

    }

    function onWindowResize() {

        camera.aspect= window.innerWidth/
window.innerHeight;
        camera.updateProjectionMatrix();

        renderer.setSize( window.innerWidth,
window.innerHeight );

    }

    function animate() {

        requestAnimationFrame( animate );

        mesh.rotation.x+= 0.005;
        mesh.rotation.y+= 0.01;

        renderer.render( scene, camera );

```

```
 }  
</script>
```

Communication 관련 API

WebSocket

HTTP로 양방향 실시간 통신을 할 수 있는 API. 실제로 HTTP 프로토콜메시지 규약을 WebSocket 규약으로 다시 재해석하도록 하는 방식으로 동작.

온라인 게임, 소셜분야에서 많이 활용되고 있음.

<http://demo.kaazing.com/livefeed/>

The screenshot shows a live feed interface from KaaZing. On the left, there's a 'Market Ticker' section with a graph showing price movements for stocks like KZNG, WSKT, and RTIM. Below the graph is a table of current stock prices. To the right of the market ticker is a news feed from The New York Times. Headlines include 'The Need for U.S.-Iran Talks', 'Cameron's Dangerous Gamble', 'Designers Dip Into Klimt's Well', and 'EU Employment Benchmark May Be Difficult to Meet'. Further down, there's a 'Currency Exchange' section showing rates for GBP/USD and EUR/USD. On the far right, there's a 'Twitter' section with a link to a political ticker and a 'Time-based Advertisement' for HTML5 featuring a woman looking at a screen.

WebSocket 기술의 스펙문서 - <http://tools.ietf.org/html/rfc6455>

```
GET /echo HTTP/1.1
Host: example.microsoft.com
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: dGhlIHNBsZSBr25jZQ==
Origin: http://microsoft.com
Sec-WebSocket-Version: 13
```

```
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
```

이름	유형	설명
WebSocket	객체	원격 호스트에 대한 양방향 채널을 제공합니다.
close	메서드	WebSocket을 닫습니다.
send	메서드	WebSocket을 사용하여 데이터를 서버로 보냅니다.
binaryType	속성	onmessage 에 수신되는 이진 데이터 형식입니다.
bufferedAmount	속성	send를 사용하여 대기 중인 데이터 바이트 수입니다.
extensions	속성	서버가 선택한 확장을 보고합니다.
onclose	속성	소켓을 닫을 때 호출되는 이벤트 처리기입니다.
onerror	속성	오류가 있을 때 호출되는 이벤트 처리기입니다.
onmessage	속성	메시지가 수신되었음을 알리는 이벤트 처리기입니다.
onopen	속성	WebSocket이 연결된 경우 호출되는 이벤트 처리기입니다.
protocol	속성	서버가 선택한 프로토콜을 보고합니다.
readyState	속성	WebSocket 연결 상태를 보고합니다.
url	속성	소켓의 현재 URL을 보고합니다.

Notifications

브라우저에게 일어나는 변화에 대해서 운영체제를 통해서 '알람'을 받게 하는 API

주로 WebWorker 의 작업의 완료나 WebSocket 으로 받아들이는 실시간 메시지 등의 처리와 같이 사용될 수 있다. 비동기화된 작업의 처리로 권장할만 하다.

다만 현재까지는 Webkit 브라우저들 중에서도 chrome 만이 지원한다.



```
// Retrieve tweets using jsonp.
var script = document.createElement("script");
script.src = 'http://twitter.com/statuses/user_timeline/' + username + '.json?' +
    'count=3&callback=fetchTweets';
document.body.appendChild(script);

function fetchTweets(data) {
    var tweet;
    var i = data.length;
    while (i--) {
        tweet = data[i];
        if (window.webkitNotifications.checkPermission() == 0) {
            window.webkitNotifications.createNotification(
                tweet.user.profile_image_url, tweet.user.name,
                tweet.text).show(); // note the show()
        } else {
            // Note that we can't call requestPermission from here as we are in the
            // callback function and not triggered just on user action
            console.log('You have to click on "Set notification permissions for this page" ' +
                'first to be able to receive notifications.');
        }
    }
}
```

XMLHttpRequest Level2 (XHR2)

- Cross-Origin XMLHttpRequest
 - 과거의 Ajax 의 same-origin domain 정책 문제를 해결할 수 있다.
 - Cross Origin Resource Sharing(CORS)를 허용
 -
- Progress events
 - 과거의 Ajax 에서는 single readystatechange event 만을 처리할 수 있었던 문제
 - 좀더 세분화되고, 상태를 체크할 수 있도록 개선

Progress Event Name

loadstart

progress

abort

error

load

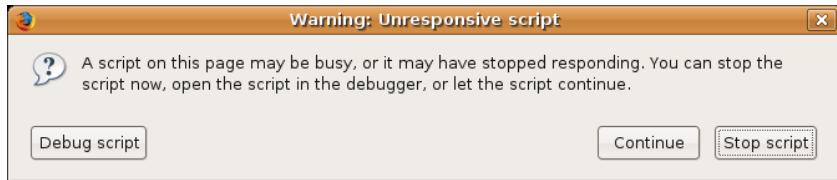
loadend

■

WebWorker

브라우저의 능력을 이용해서 백그라운드 작업을 실행하는 API, 비 동기화된 처리 작업을 하기 위해서 사용한다.

과거에는 자바스크립트 많은 리소스를 소모하는 경우에 차단되는 경고창이 겪는 문제를 해결하는데 도움을 준다.



백그라운드 counting

```
for (var i= 0; i<= 1000000000; i+= 1){  
    var j = i;  
}  
postMessage(j);
```

```
<!DOCTYPE HTML>  
<html>  
<head>  
<title>Big for loop</title>  
<script>  
var worker = new Worker('bigLoop.js');  
worker.onmessage= function (event) {  
    alert("Completed "+ event.data+ "iterations" );  
};  
  
function sayHello(){  
    alert("Hello sir...." );  
}  
</script>  
</head>  
<body>  
<input type="button" onclick="sayHello();" value="Say Hello" />  
</body>  
</html>
```

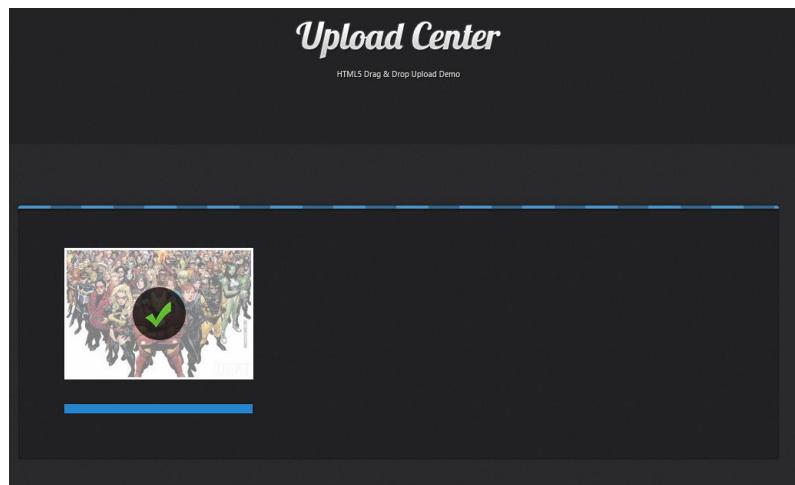

File API & Hardware

File API

브라우저가 local 의 파일 데이터를 읽어서 데이터를 추출할 수 있는 기능을 제공,

사용자의 PC 에서 일반 파일 복사처럼 Drag And Drop 으로 파일 데이터를 읽어서, 업로드나 화면에 Canvas 등으로 출력하는 방식으로 주로 사용

<http://demo.tutorialzine.com/2011/09/html5-file-upload-jquery-php/>



FileReader API 를 이용하는 데이터의 처리

```
var reader = new FileReader();

// Closure to capture the file information.
reader.onload= (function(theFile) {
return function(e) {
// Render thumbnail.
var span = document.createElement('span');
span.innerHTML= ['<img class="thumb" src="", e.target.result,
" title="", escape(theFile.name), "/>'].join('');
document.getElementById('list').insertBefore(span, null);
};

})(f);

// Read in the image file as a data URL.
reader.readAsDataURL(f);
```


CSS3 와 새로운 웹 페이지

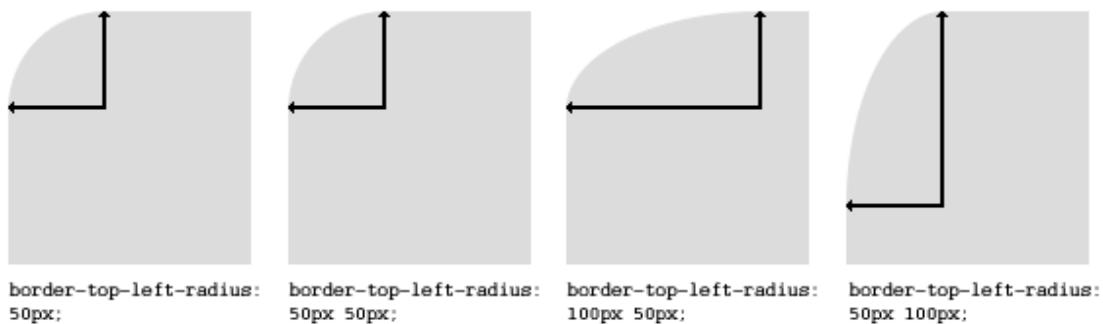
기존의 이미지나 플래시가 해야만 했던 부분의 상당부분을 CSS3 를 이용해서 작성할 수 있으면, 모바일 환경에서도 유용하게 사용할 수 있다.

- 외적인 측면에서의 주요 특징들
 - Rounded-corner 의 지원
 - Text Shadow
 - Border Image
 - Custom Fonts
 - Multiple Backgrounds
 - Color Gradients
- 기능적인 측면
 - Animation
 - Media Query
 - Transform
 - Transition
 - Multi-column

CSS3 를 적용하는데 있어서 반드시 조심해야 하는 사항들

- 브라우저마다 구현하거나 사용하는 설정이 다르다.
- 더 이상 CSS3 는 웹 퍼블리셔나 디자이너들만의 영역이 아니다.
- 자바스크립트로 해야 하는 작업과 CSS 로 해야하는 작업의 기준이 명확할 필요가 있다.

Rounded-Corner



```
border-radius: 4em;
```

is equivalent to

```
border-top-left-radius: 4em;  
border-top-right-radius: 4em;  
border-bottom-right-radius: 4em;  
border-bottom-left-radius: 4em;
```

and

```
border-radius: 2em 1em 4em / 0.5em 3em;
```

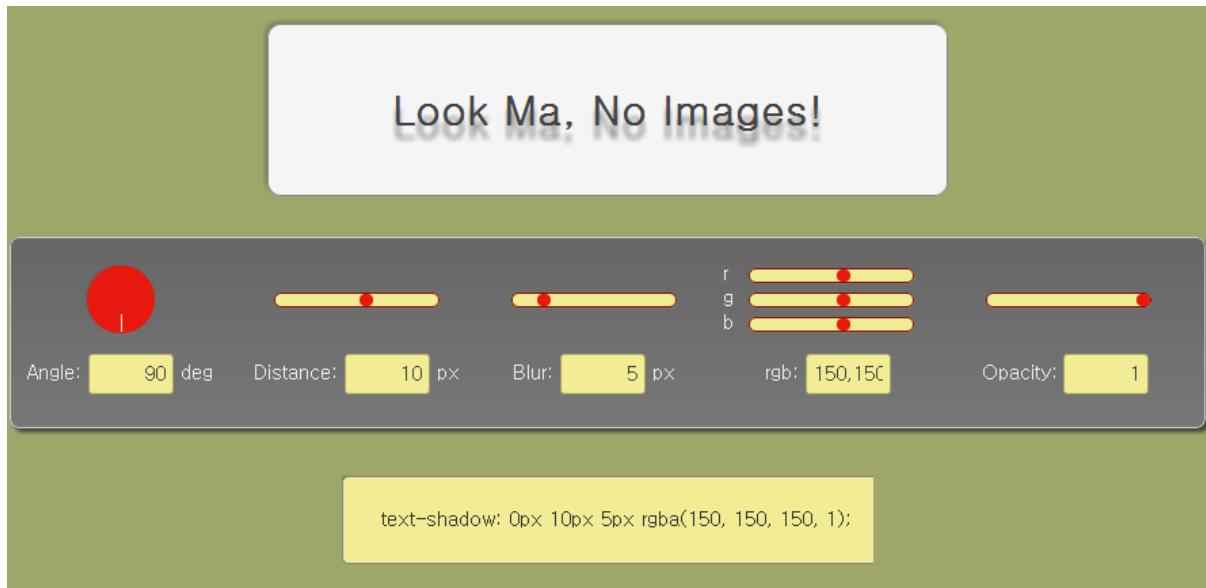
is equivalent to

```
border-top-left-radius: 2em 0.5em;  
border-top-right-radius: 1em 3em;  
border-bottom-right-radius: 4em 0.5em;  
border-bottom-left-radius: 1em 3em;
```

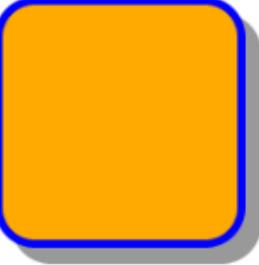
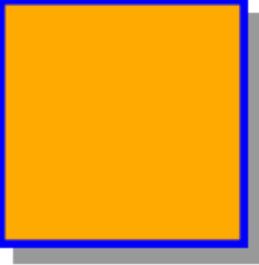
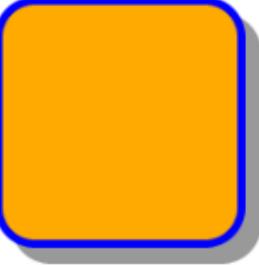
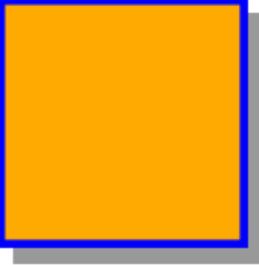
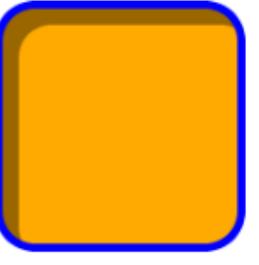
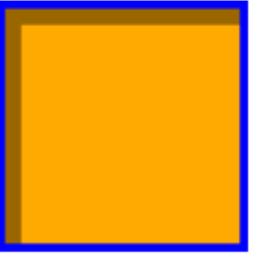
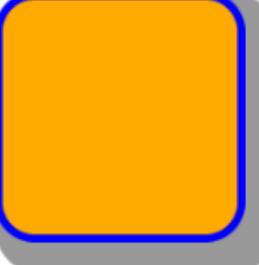
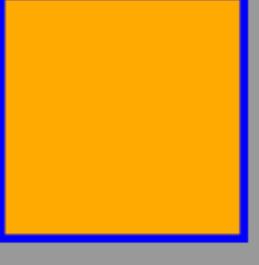
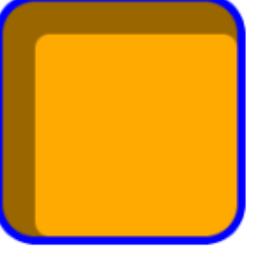
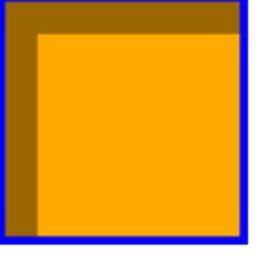
단순한 반지름을 지정하는 것 외에도 Corner Shaping이나 Corner Clipping 기술등이 포함되어 있다(<http://www.w3.org/TR/css3-background/#corner-shaping>).

Text Shadow

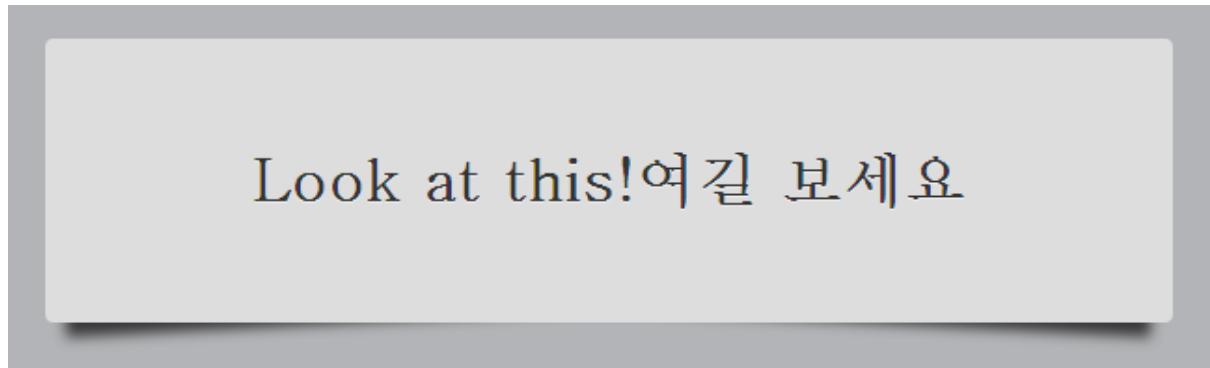
Box Shadow와 같이 결합되어 이미지로만 가능했던 효과를 처리할 수 있다.



Box Shadow 와의 결합

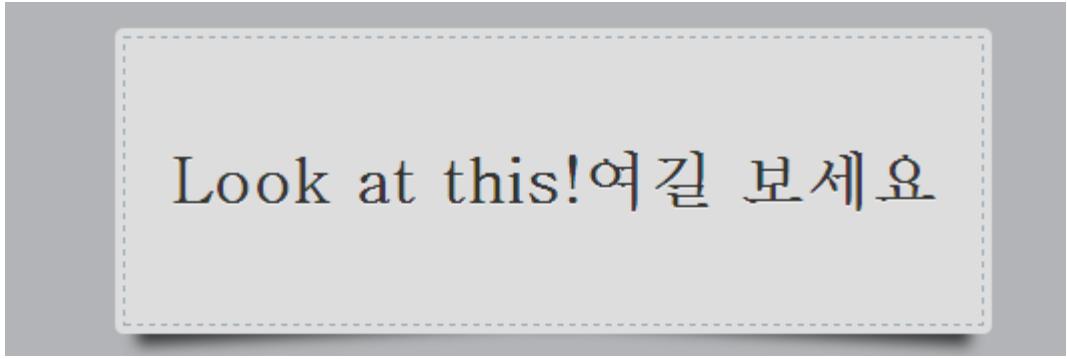
<pre>border:5px solid blue; background-color:orange; width: 144px; height: 144px;</pre>	<pre>border-radius: 20px;</pre> 	<pre>border-radius: 0;</pre> 
<pre>box-shadow: rgba(0,0,0,0.4) 10px 10px;</pre>		
<pre>box-shadow: rgba(0,0,0,0.4) 10px 10px inset</pre>		
<pre>box-shadow: rgba(0,0,0,0.4) 10px 10px 0 10px /* spread */</pre>		
<pre>box-shadow: rgba(0,0,0,0.4) 10px 10px 0 10px /* spread */ inset</pre>		

```
.sticker:before, .sticker:after {  
    z-index: -1;  
    position: absolute;  
    content: "";  
    bottom: 15px;  
    left: 10px;  
    width: 50%;  
    top: 80%;  
    max-width: 300px;  
    background: rgba(0, 0, 0, 0.7);  
    box-shadow: 0 15px 10px rgba(0, 0, 0, 0.7);  
    -webkit-transform: rotate(-3deg);  
}  
  
.sticker:after{  
    -webkit-transform: rotate(3deg);  
    right: 10px;  
    left: auto;  
}
```



Border Image

outline 을 이용하는 효과



Look at this!여길 보세요

```
.sticker
{
    outline: 1px dashed #98abb9;
    outline-offset: -5px;

    position: relative;
    width: 60%;
    background: #ddd;
    border-radius: 4px;
    border-radius: 4px;
    padding: 2em 1.5em;
    color: rgba(0,0,0, .8);
    text-shadow: 0 1px 0 #fff;
    line-height: 1.5;
    margin: 60px auto;
}
```

Multiple Backgrounds

여러 개의 이미지를 결합해서 하나의 이미지로 겹치도록 할 수 있다.

http://people.opera.com/zibin/background/multiple_background_image.html



Figure 8: Screenshot of a combined background image using multiple background images

```
background:  
url(rose.png) no-repeat 150px -20px,  
url(driedrose.png) no-repeat,  
url(fieldsky.jpg) no-repeat;
```

Multiple Backgrounds 는 아니지만 이미지나 요소의 위치를 마우스에 따라 조정해서 약간의 3D 효과를 만들어 낼 수도 있다.



Color Gradients

<http://gradients.glrzad.com/>을 이용하면 편리하다.

CSS3 Gradient Generator by: Damian Galarza

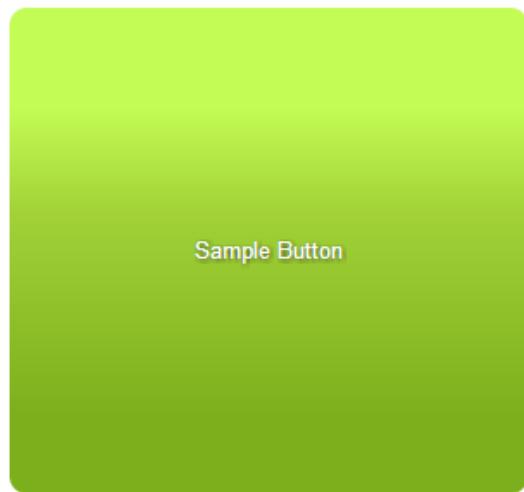
Color Swatches +

[Generate Random Gradient](#)
Select colors to make up your gradient.



Gradient Sample

The sample updates live while you're adjust your gradient.



CSS3 의 기능적인 측면

Media query

RWD(Responsive Web Design)시에 사용 <http://www.w3.org/TR/css3-mediaqueries/>

<http://tympanus.net/Tutorials/3DRestaurantMenu/>

iPhone5 를 구분할 수 있는 데이터 (<http://mobile.dzone.com/articles/new-iphone-5-and-ios-6>)

```
Mozilla/5.0 (iPhone; CPU iPhone OS 6_0 like Mac OS X) AppleWebKit/536.26 (KHTML, like Gecko) Version/6.0 Mobile/10A403 Safari/8536.25  
isPhone4inches = (window.screen.height==568);  
@media (device-height: 568px) and (-webkit-min-device-pixel-ratio: 2) {  
/* iPhone 5 or iPod Touch 5th generation */  
}
```

Media Query 에 사용하는 속성들

Feature	Value	Accepts min/max prefix	Description
width	length	Yes	The width of the targeted display area (ie: the browser window on most desktop and mobile devices). On desktop computers, if the user resizes the browser, the width changes as a result, while on most mobile browsers, the inability to resize the browser window means it will stay constant.
height	length	Yes	The height of the targeted display area (ie: the browser window on most desktop and mobile devices).
device-width	length	Yes	The width of the device's entire screen irrespective of the browser window's width. On a desktop with screen resolution 1024x768 for example, the device-width is always 1024. On an iPhone -including iPhone 4- the device-width is 320.
device-	length	Yes	The height of the device's entire screen. On

height			a desktop with screen resolution 1024x768 for example, the device-height is 768. On an iPhone- including iPhone 4, the device-height is 480.
orientation	"portrait" or "landscape"	no	The orientation of the device. When the device's "width" media is larger than its "height", the orientation value returned is "landscape", otherwise, it returns "portrait".
aspect-ratio	ratio	yes	Detects the ratio of value of the "width" media to the value of the "height" media, such as 2/1
device-aspect-ratio	ratio	yes	Detects the ratio of value of the "device-width" media to the value of the "device-height" media. On a device with screen resolution 1280x720, the following device-aspect-ratio values would all match it (multiples of 1280x720): @media screen and (device-aspect-ratio: 16/9) { } @media screen and (device-aspect-ratio: 1280/720) { } @media screen and (device-aspect-ratio: 2560/1440) { }
color	integer	yes	Number of bits per color component of the device. If the device is not a color device, the value is zero.
color-index	integer	yes	Number of entries in the color lookup table of the output device.
monochrome	integer	yes	Number of bits per pixel in a monochrome frame buffer. For non monochrome devices, this value is 0.
resolution	resolution	yes	The resolution of the output device, or the density of the pixels. Common units are "dpi" and "dpcm".
scan	"progressive" or "interlace"	no	The scanning process of "tv" output devices.
grid	1 or 0	no	Detects whether the output device is grid or bitmap. Grid based devices return a value of 1, all others a value of 0.

안드로이드 2.3 까지의 작은 화면(iPhone 3 포함)

```
/* ##### target mobile devices with max device width 480px ##### */
```

```
@media screen and (max-device-width: 480px){  
    /* some css here */  
}
```

iPhone 4+, iPad 3+

```
/* ##### CSS for high resolution devices in general ##### */  
@media screen and (-webkit-min-device-pixel-ratio: 1.5){  
    /* some css here */  
}  
  
/* ##### CSS for small screen high resolution devices such as iPhone 4 ##### */  
@media screen and (-webkit-min-device-pixel-ratio: 2) and (max-device-width: 480px){  
    /* some css here */  
}  
  
/* ##### CSS for large screen high resolution devices such as iPad 3 ##### */  
@media screen and (-webkit-min-device-pixel-ratio: 2) and (min-device-width: 768px){  
    /* some css here */  
}
```

다양한 화면 크기에 따른 media query 설정

```
/* ##### Mobile Phones Portrait ##### */  
@media screen and (max-device-width: 480px) and (orientation: portrait){  
    /* some CSS here */  
}  
  
/* ##### Mobile Phones Landscape ##### */  
@media screen and (max-device-width: 640px) and (orientation: landscape){  
    /* some CSS here */
```

```

}

/* ##### Mobile Phones Portrait or Landscape #####
@media screen and (max-device-width: 640px){
    /* some CSS here */
}

/* ##### iPhone 4+ Portrait or Landscape #####
@media screen and (max-device-width: 480px) and (-webkit-min-device-pixel-ratio: 2){
    /* some CSS here */
}

/* ##### Tablets Portrait or Landscape #####
@media screen and (min-device-width: 768px) and (max-device-width: 1024px){
    /* some CSS here */
}

/* ##### Desktops #####
@media screen and (min-width: 1024px){
    /* some CSS here */
}

```

```

<link rel="stylesheet" type="text/css" href="default.css" media="screen and (min-device-width:1024px)" />

<link rel="stylesheet" type="text/css" href="mobile.css" media="only screen and (max-device-width:640px)" />

```

Mobile Web

- [About](#)
- [History](#)
- [Consulting](#)
- [On Call](#)
- [Development](#)

Menu

About

The first access to the mobile web was commercially offered in Finland in 1996 on the Nokia 9000 Communicator phone via the Sonera and Radiolinja networks. This was access to the real internet. The first commercial launch of a mobile-specific browser-based web service was in 1999 in Japan when i-mode was launched by NTT DoCoMo.

Mobile Web

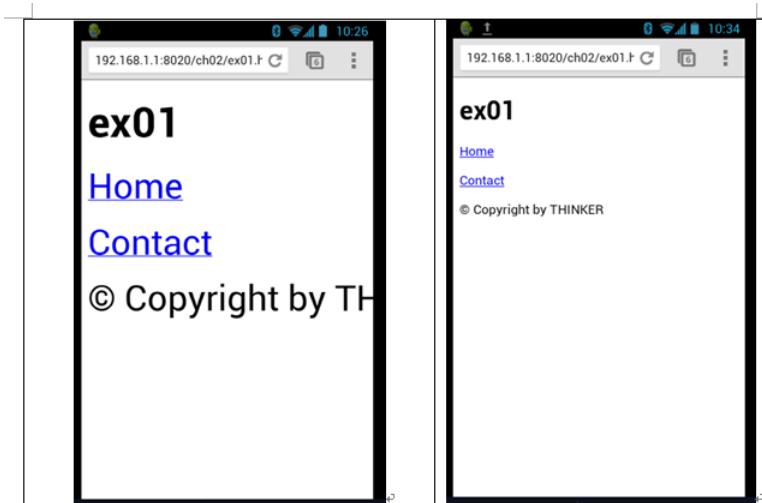
About

The first access to the mobile web was commercially offered in Finland in 1996 on the Nokia 9000 Communicator phone via the Sonera and Radiolinja networks. This was access to the real internet. The first commercial launch of a mobile-specific browser-based web service was in 1999 in Japan when i-mode was launched by NTT DoCoMo.



The Mobile Web refers to the use of Internet-

Viewport의 설정



Property	설명
width	뷰포트의넓이를의미한다.단위로는 'px'을이용 기본값은 980px, 최소값은 200px이고최대로 10,000 까지 지정할 수 있다. 'device-width'로설정하면디바이스의화면 크기를기준으로삼는다.
height	뷰포트의높이를의미한다.단위로는 'px'을이용 기본값으로는width를기준으로계산된값을활용한다.일반적으로개발자가신경쓰지않는다.
initial-scale	화면의배율

	0에서 10까지의 배율을 조정할 수 있다.
user-scalable	사용자가 화면을 확대할 수 있는지에 대한 결정 yes/no로 설정되고 기본값은 yes
minimum-scale	사용자가 zoom-out 할 때의 줄어드는 배율 기본으로 0.25 씩
maximum-scale	사용자가 zoom-in 할 때의 확대 비율
target-densitydpi	안드로이드에서만 사용되는 속성으로 dotsperinch 값을 의미. device-dpi 혹은 medium-dpi 값을 주로 사용

<http://www.paulund.co.uk/css3-animated-media-queries>

모바일화면 가로모드에서 글자 확대를 방지하기

뷰포트의 설정이나 다양한 상황 때문에 가로모드가 되면서 글자가 확대되는 경우가 발생하므로, 확대 방지를 아예 지정하는 것이 좋다.

* { -webkit-text-size-adjust: none; }

미디어 쿼리의 문제점과 대안

- CSS만 변경될 뿐 전송되는 데이터의 양 자체가 줄어드는 것은 아니다.
- 향후 유지보수상에 문제가 있을 수 있다.
- 리소스의 제약과 다양한 디바이스의 상황에서 미디어쿼리가 복잡해 질 수 있다.

대안으로 생각해 볼 수 있는 방식은 HTTP 프로토콜의 'user-agent' 정보를 활용해서 디바이스에 맞는 화면을 작성하는 '브라우저 스니핑(sniffing)기법'을 활용하는 것 (<http://www.quirksmode.org/js/detect.html>) – 그럼에도 불구하고 실제 디바이스의 정보인지 확인할 수가 없기 때문에 정확한 해결책은 될 수

없다(<http://stackoverflow.com/questions/661213/why-is-browser-sniffing-not-a-recommended-practice>).

```
GET / HTTP/1.1
Host: 192.168.123.146:8080
Connection: keep-alive
X-Purpose: instant
User-Agent: Mozilla/5.0 (Linux; Android 4.0.4; Galaxy Nexus Build/IMM76K)
AppleWebKit/535.19 (KHTML, like Gecko) Chrome/18.0.1025.166 Mobile
Safari/535.19
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate,sdch
Accept-Language: ko-KR,ko;q=0.8,en-US;q=0.6,en;q=0.4
Accept-Charset: windows-949,utf-8;q=0.7,*;q=0.3
```

스마트폰	User-Agent 정보
iPhone 4	Mozilla/5.0 (iPhone; U; CPU iPhone OS 4_0 like Mac OS X; en-us) AppleWebKit/532.9 (KHTML, like Gecko) Version/4.0.5 Mobile/8A293 Safari/6531.22.7
iPhone3	Mozilla/5.0 (iPhone; U; CPU iPhone OS 3_0 like Mac OS X; en-us) AppleWebKit/528.18 (KHTML, like Gecko) Version/4.0 Mobile/7A341 Safari/528.16
Samsung GalaxyS	Mozilla/5.0 (Linux;U;Android 2.1-update1; ko-kr; SHW-M105 Build/ECLAIR) AppleWebKit/530.17(KHTML, like Gecko) version/4.0 Mobile Safari/530.17
HTC Desire	Mozilla/5.0 (Linux; U; Android 2.1-update1; en-gb; HTC Desire Build/ERE27) AppleWebKit/530.17 (KHTML, like Gecko) Version/4.0 Mobile Safari/530.17
Windows 7 Phone	Mozilla/4.0 (compatible; MSIE 7.0; Windows Phone OS 7.0; Trident/3.1; IEMobile/7.0; LG; GW910)
iPad	Mozilla/5.0 (iPad; U; CPU OS 3_2 like Mac OS X; en-us) AppleWebKit/531.21.10 (KHTML, like Gecko) Version/4.0.4 Mobile/7B334b Safari/531.21.10

CSS3 의 transform(변형)/ transition(전이) 효과

CSS3 기능	지원 시작 브라우저
CSS 트랜지션	Safari 3.2: 13/11/2008 Firefox 4.0: Late 2010 Chrome 1.0: 02/09/2008 Opera 10.5: 02/03/2010 IE 10: 09/2011
CSS 애니메이션	Safari 4.0: 11/06/2008 Chrome 1.0: 02/09/2008 Firefox 5: 20/04/2011 IE 10: 09/2011
CSS 2D 트랜스폼	Safari 3.2: 13/11/2008 Firefox 3.5: 30/06/2009 Chrome 1.0: 02/09/2008 Opera 10.5: 02/03/2010 IE 9: 09/2010
CSS 3D 트랜스폼	Safari 4.0: 11/06/2008 Chrome: 28/08/2010 IE 10: 09/2011 Firefox: 27/10/2011

transform(트랜스폼)

- translation: 이동 효과
- scale: 확대, 축소 효과
- rotate: 회전 효과
- skew: 비틀기 효과

```
#img3d{
    -webkit-transform:translateX(100px)translateY(-100px) scale(0.5)
    rotate(45deg);
    -moz-transform:translateX(100px)translateY(-100px) scale(0.5)
    rotate(45deg);
    -ms-transform:translateX(100px)translateY(-100px) scale(0.5)
    rotate(45deg);
    -o-transform:translateX(100px)translateY(-100px) scale(0.5)
    rotate(45deg);
    transform:translateX(100px)translateY(-100px) scale(0.5)
}
```

```
rotate(45deg);  
}
```

Translate



Scale



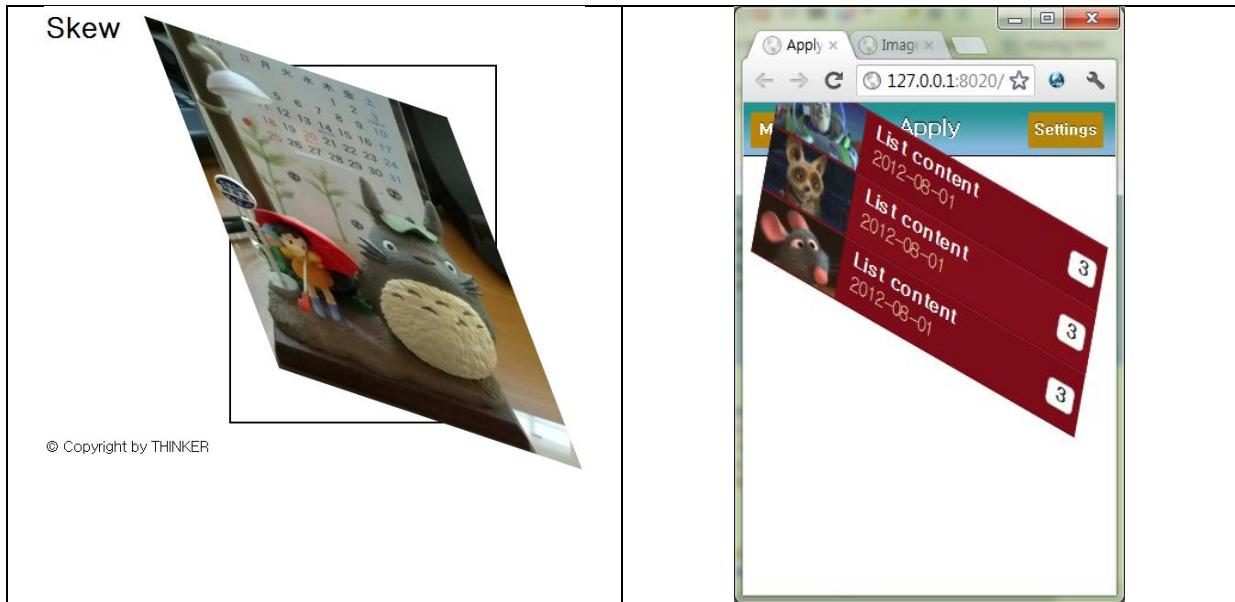
Rotate



Mix



© Copyright by THINKER

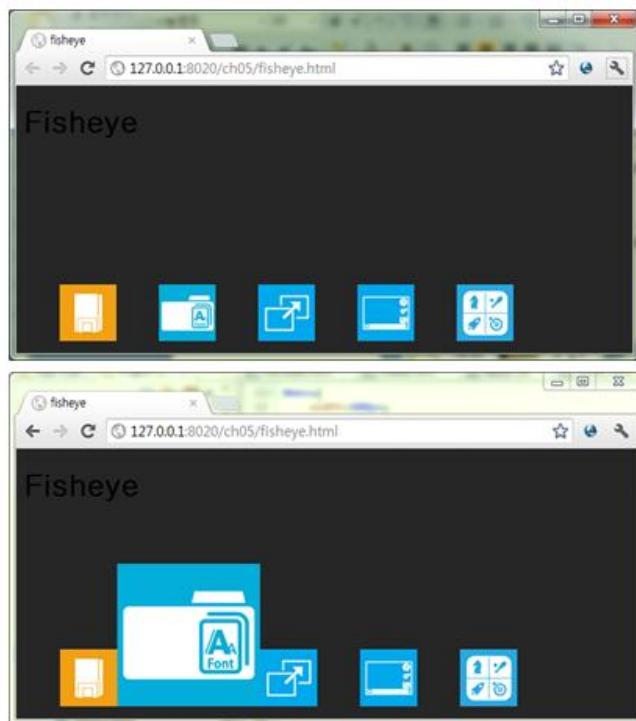


transition 지정을 통한 점진적인 효과

변형이 일어나는 시간이나, 반복횟수, 가속도 처리 방식 등을 지정할 때 사용한다.

```
#img3d{
    -webkit-transition-duration:2s;
    -moz-transition-duration:2s;
    -ms-transition-duration:2s;
    -o-transition-duration:2s;
    transition-duration:2s;
    /* 이하생략 */
}
-webkit-transition:1.0s;
-webkit-transition-timing-function:ease-in;
-webkit-transition-delay:0.5s;
```

마우스 호버효과와 트랜스폼/트랜지션 활용하기



css3 의 애니메이션 제작

플래시나 자바스크립트로 움직이는 애니메이션을 제작하는 대신에 CSS3 에는 keyframes 라는 프레임단위의 움직임을 통해서 애니메이션 효과를 구현할 수 있다.

(자바스크립트 방식과 CSS3 방식의 성능비교 -<http://dev.opera.com/articles/view/css3-vs-jquery-animations/>)

```
.show{
    -webkit-animation:fadeIn;
    -webkit-animation-duration:2s;
}

@-webkit-keyframes fadeIn{
    from{
        opacity:0;
    }
    to{
        opacity:1;
    }
}

.hide{
```

```

        -webkit-animation:fadeOut;
        -webkit-animation-duration:2s;
    }

@-webkit-keyframes fadeOut{
    from{
        opacity:1;
    }
    to{
        opacity:0;
    }
}

$('.page').bind('click', function( event ) {

    var target = $(this).data("target");

    console.log("target:"+ target);

    changePage( $(target) );

});

function changePage($next){

    var $before = $('.current');

    //다음화면이보이게하고, 나타나는효과를준다.
    $next.addClass('current show');

    //기존화면은현재보이고있으므로, 점점사라지는효과를준다.
    $before.addClass('hide');

    //새로운화면이다나타나면기존화면의 display속성을조정해서안보이게한다.
    $next.one("webkitAnimationEnd", function( ) {
        $before.removeClass("current hide");
        $next.removeClass('show');
    });
}

```

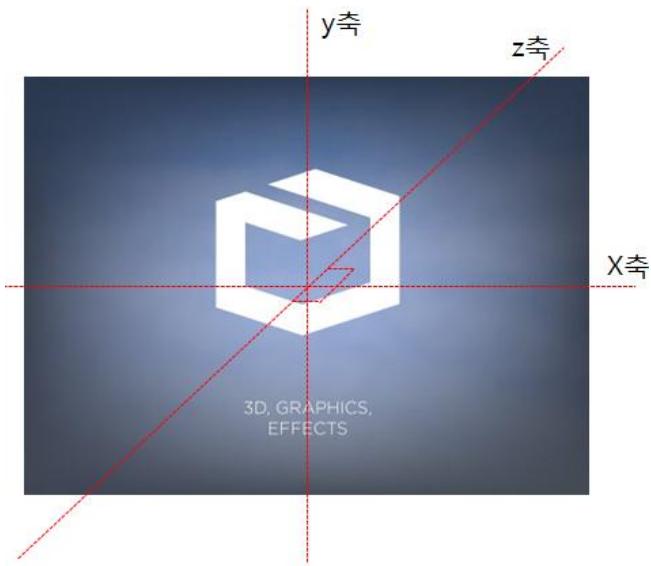
CSS3 의 애니메이션과 관련된 속성들

Value	Description
<i>animation-name</i>	Specifies the name of the keyframe you want to bind to the selector
<i>animation-duration</i>	Specifies how many seconds or milliseconds an animation takes to complete
<i>animation-timing-function</i>	Specifies the speed curve of the animation
<i>animation-delay</i>	Specifies a delay before the animation will start
<i>animation-iteration-count</i>	Specifies how many times an animation should be played
<i>animation-direction</i>	Specifies whether or not the animation should play in reverse on alternate cycles

<http://leaverou.github.com/animatable/>

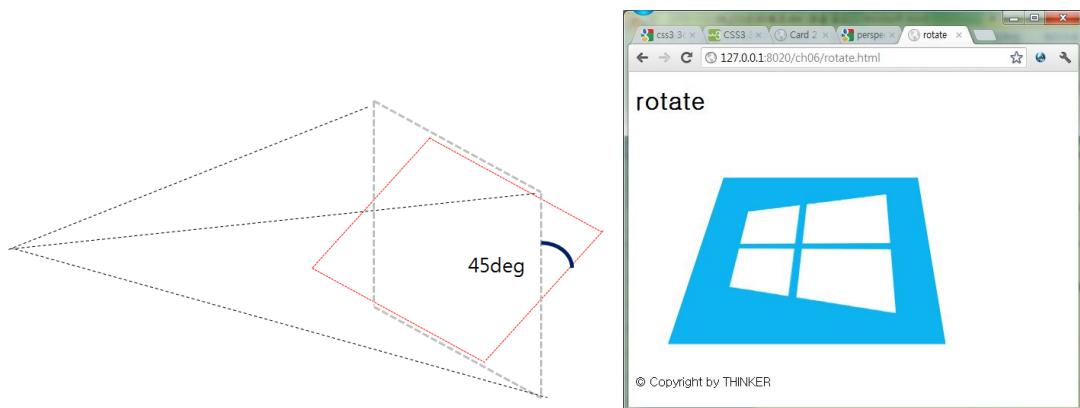
최근에는 <http://www.sencha.com/products/animator/demos/>와 같이 라이브러리나 프레임워크가 애니메이션을 지원하는 경우도 많아지는 추세

css3 의 3D 효과



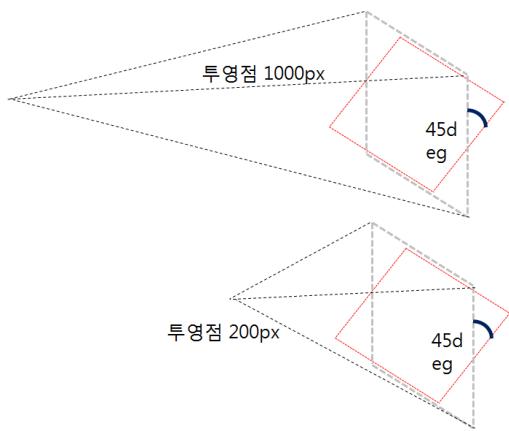
- rotate: x축, y축, z축으로 대상 요소를 회전
- translate: x축, y축, z축으로 대상 요소를 이동

모든 3D효과에서 가장 중요한 설정은 투영점을 어떻게 잡아주는가의 문제.

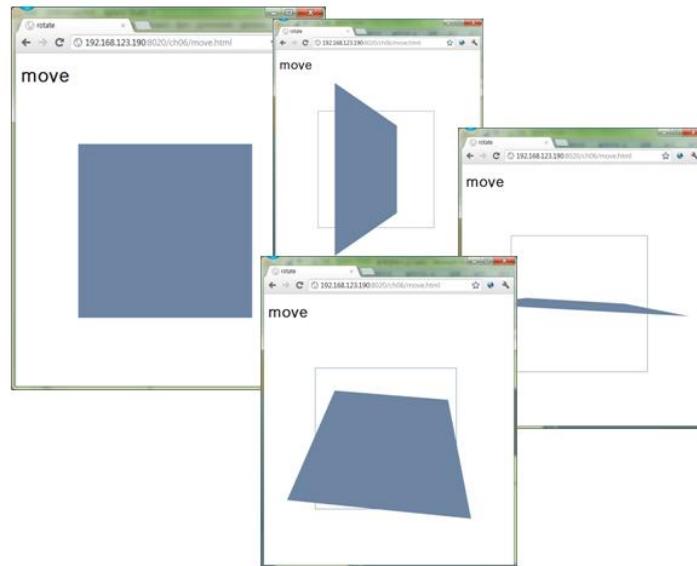


```
#wrapper{
    border:2px solid;
    width:300px;
    margin:0 auto;
    -webkit-transform-style: preserve-3d;
    -webkit-perspective:600px;
}
```

투영점의 설정에 따른 변화



키보드를 이용하는 3D 효과를 이용하는 회전하는 면 작성하기



```
#container{
    margin:100px;
    width:300px;
    height:300px;
    -webkit-perspective:400px;
    border:1px solid #6D84A2;
}

#wrapper{
    width:100%;
    height:100%;
    background-color:#6D84A2;
    -webkit-transform:translateZ(0px);
    -webkit-transform-style: preserve-3d;
    -webkit-transition-duration:2s;
}

<script>
var yDeg= 0;
var xDeg= 0;
$(document).ready(function() {

    $(document).bind("keydown", function(e) {
```

```

switch(e.keyCode){
    case 37: // left
        console.log("left");
        $("#wrapper").css("-webkit-transform", "rotateY("+ (yDeg-= 90)+"deg)");
        break;

    case 38: // up
        console.log("up");
        $("#wrapper").css("-webkit-transform", "rotateX("+(xDeg-= 90)+"deg)");
        break;

    case 39: // right
        console.log("right");
        $("#wrapper").css("-webkit-transform", "rotateY("+(yDeg+= 90)+"deg)");
        break;

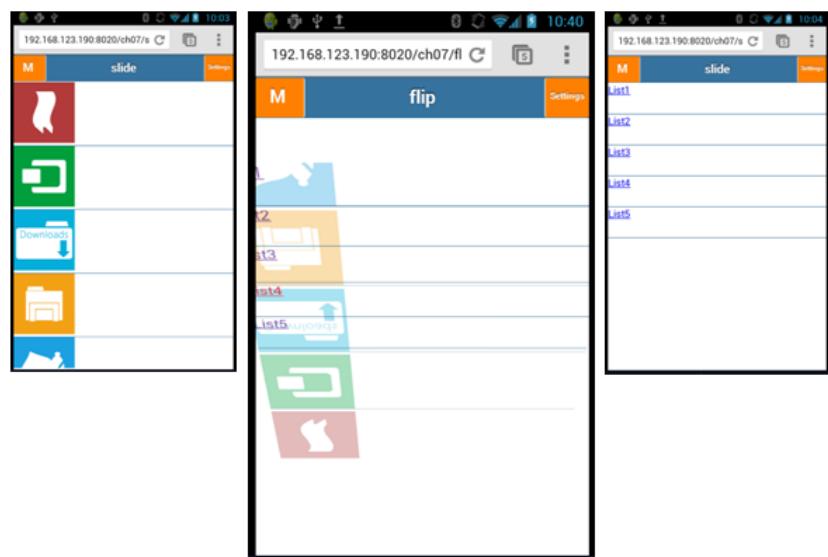
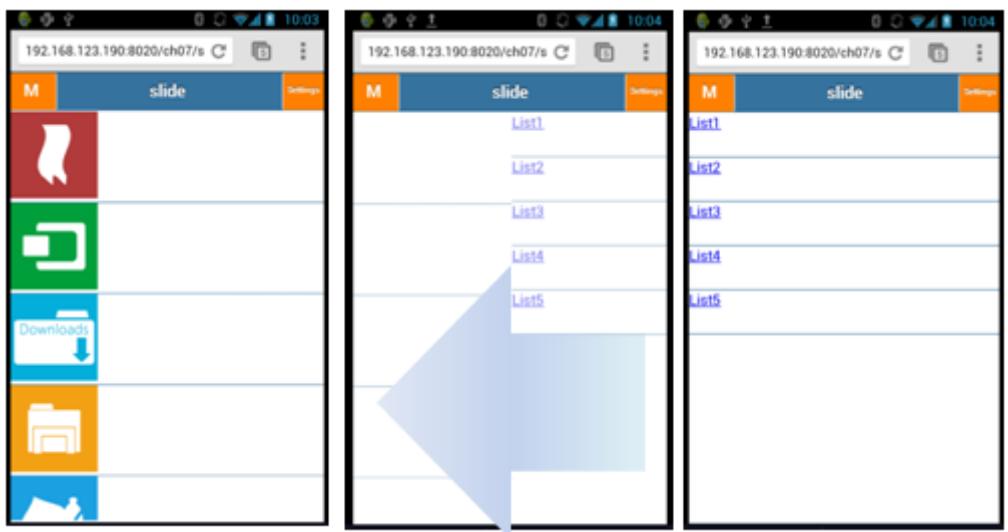
    case 40: // down
        console.log("down");
        $("#wrapper").css("-webkit-transform", "rotateX("+(xDeg+= 90)+"deg)");
        break;
    }
};

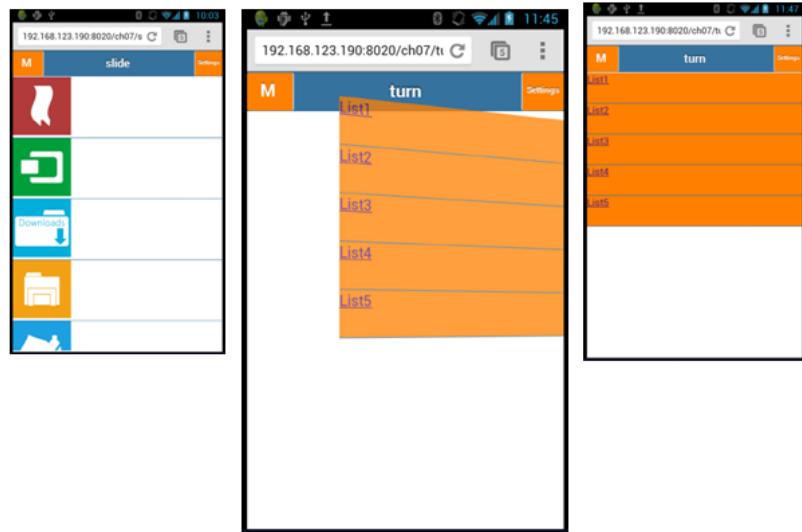
});
```

</script>

페이지의 이동효과

CSS3의 애니메이션과 3D 효과를 결합하면 모바일환경의 앱처럼 화면이 변경되는 효과를 제작할 수 있다.





impress.js

CSS3의 트랜스폼, 트랜지션 효과를 이용하여 프리젠테이션을 작성할 수 있는 간단한 자바스크립트 라이브러리 (prezi 와 유사한 효과를 웹으로 작성)

<http://www.paulund.co.uk/no-more-powerpoint-i-have-impress-js>

History 제어

HTML5 는 History API 를 이용해서 브라우저의 'back 버튼'의 동작을 제어할 수 있는 방안을 제공한다. 만일 HTML5 를 사용할 수 없는 상황이라면 history.js(<https://github.com/balupton/History.js/>) 를 이용해서 유사한 효과를 얻을 수 있다.

<https://github.com/balupton/history.js/wiki>Showcase>

```
$("#menu li a").bind("click", function(e){
    e.preventDefault();
    console.log("click:"+e.target.hash);
    window.history.pushState("", "", e.target.hash);
    changePage( $(e.target.hash) );
});
```

```
<script>
window.addEventListener("popstate", function(e) {
    console.log("popstate:"+ e);
}, false);
</script>
```

Offline Web Apps

HTML5 는 하나의 온전한 Application 처럼 동작할 수 있도록 offline 상황에서 cache 를 이용하는 방식으로 페이지나 데이터를 저장해서 사용할 수 있다.

```
<html manifest="example.appcache">
...
</html>
```

```
CACHE MANIFEST
index.html
stylesheet.css
images/logo.png
scripts/main.js
```

Storage 기능

HTML5 에서 지원하게 될 데이터 저장기능은 2 가지로 구분된다.

- Web Storage

sessionStorage	localStorage
Values persist only as long as the window or tab in which they were stored.	Values persist beyond window and browser lifetime.
Values are only visible within the window or tab that created them.	Values are shared across every window or tab running at the same origin.

- Web SQL database / Indexed database

Touch Event

데스크탑 환경과는 달리 모바일 환경에서는 click 이벤트 대신에 touch 기반의 이벤트로 처리.

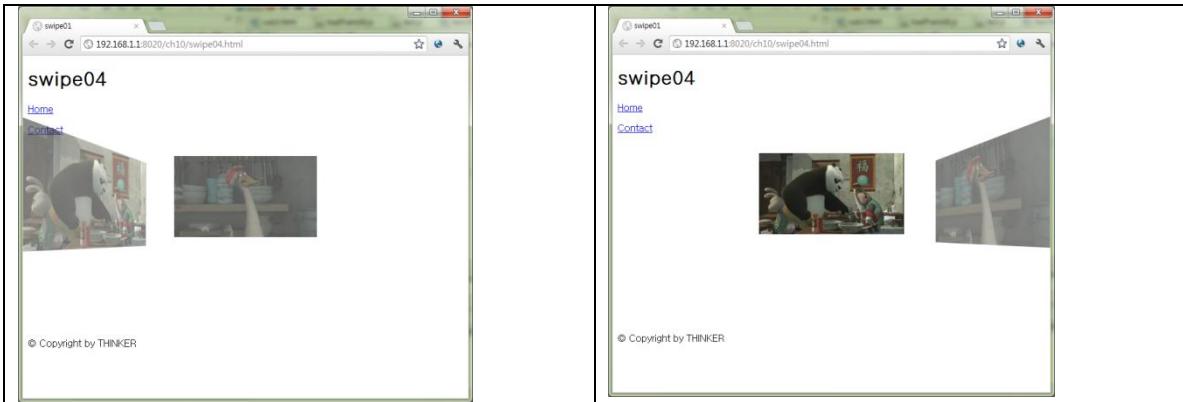
click 이벤트의 경우에는 기본적으로 200ms 의 시간을 소모

- touchstart → mousedown
- touchmove → mousemove
- touchend → mouseup

```
$(".main").bind("touchstart mousedown" , function (event ){  
    event.preventDefault();  
  
    var eventObj = checkDevice(event);  
  
    startX = eventObj.pageX;  
  
    //console.log("start: "+eventObj.pageX);  
  
});  
  
$(".main").bind("touchend mouseup" , function (event ){  
    event.preventDefault();  
  
    var eventObj = checkEndDevice(event);  
  
    //console.log("end: "+eventObj.pageX);  
  
    endX = eventObj.pageX;  
  
    var gap = startX - endX;  
  
    console.log(gap);  
    changeImg(gap);  
  
});  
  
$(".main").bind("touchmove mousemove", function(evnet ) {  
    event.preventDefault();  
  
    //console.log("move");
```

```
});
```

터치 기반의 이벤트를 이용해서 swipe 효과를 만들거나, drag 효과를 구현할 수 있다.



HTML5 와 위젯

HTML5 를 이용하는 개발의 생산성을 높여줄 수 있도록 위젯형태의 개발방식이나 프레임워크들이 점차 활성화되어가는 상황

<http://themefortress.com/reverie/>

REVERIE: VERSATILE HTML5 WORDPRESS FRAMEWORK

Reverie

Versatile HTML5 Responsive WordPress Framework
based on the Power of Foundation.

<http://demos.kendoui.com/>

The screenshot shows the homepage of the Kendo UI Demos website. At the top, there's a navigation bar with tabs for Overview, Web Demos, Mobile Demos, DataViz Demos, and ThemeBuilder. A red button on the right says "See Q3'12 BETA Demos". Below the navigation, there are three main demo sections: "Web Demos" (showing a desktop interface with multiple windows), "Mobile Demos" (showing three smartphones displaying different mobile applications), and "DataViz Demos" (showing a chart titled "STOCKHISTORY" with data for Google, Microsoft, Apple, and Amazon). Each section has a "Launch Demos" button at the bottom.

<http://www.sencha.com/products/touch/>

Log in / Register | Forum | App Gallery | Learn | Documentation | Cart 0

Sencha

Products | Support | Training | Company | Blog | Contact | Store

Home / Products

Sencha Touch Build Mobile Web Apps with HTML5

Buy Support | Download



The Best Framework Just Got Better
Build HTML5 mobile apps for iPhone, Android, and BlackBerry.

With over 50 built-in components, state management, and a built-in MVC system, Sencha Touch provides everything you need to create immersive mobile apps.

[View Examples](#) | [Watch Video](#)