

# **HTML5+CSS3**

1. HTML5란 .....	3
2. HTML5로 기본적인 문서 만들기 .....	23
3. HTML5 비디오 요소 .....	44
4. 캔버스(Canvas) .....	55
5. 웹 폼 .....	86
6. CSS3 .....	107
7. CSS3 활용 .....	150
8. 다양한 API .....	184

# **1. HTML5란**



스마트폰

# HTML5



태블릿 PC



리눅스

# HTML5 Timline



W3C, HTML 표준을 업데이트하지 않을 것을 선언하고, XHTML을 웹표준으로 선언.

XHTML 문서와 호환되지 않는 HTML 문서가 많았고, XHTML 문서의 MIME 타입을 따르지 않는 브라우저가 대부분이었기 때문에 XHTML은 거의 보급되지 않았음

웹 애플리케이션이 기하급수적으로 늘어났음 → HTML 플랫폼으로 돌아갈 수 밖에 없는 환경

1998      2004      2005      2006      2007      2008      2009      2010

# HTML5 Timline



XHTML을 보급해야 한다는  
W3C와 양립하여, HTML을  
업데이해야 한다는 의견을  
가진 **WHATWG** 발족

**Web Hypertext Application  
Technology Working Group:**  
Apple, Mozilla, Opera 등이  
공동으로 발족한 워킹 그룹으로  
HTML을 진화시키고 그 성과를  
표준화 단체에 제출하는 것을  
목적으로 하였다.

1998      2004      2005      2006      2007      2008      2009      2010

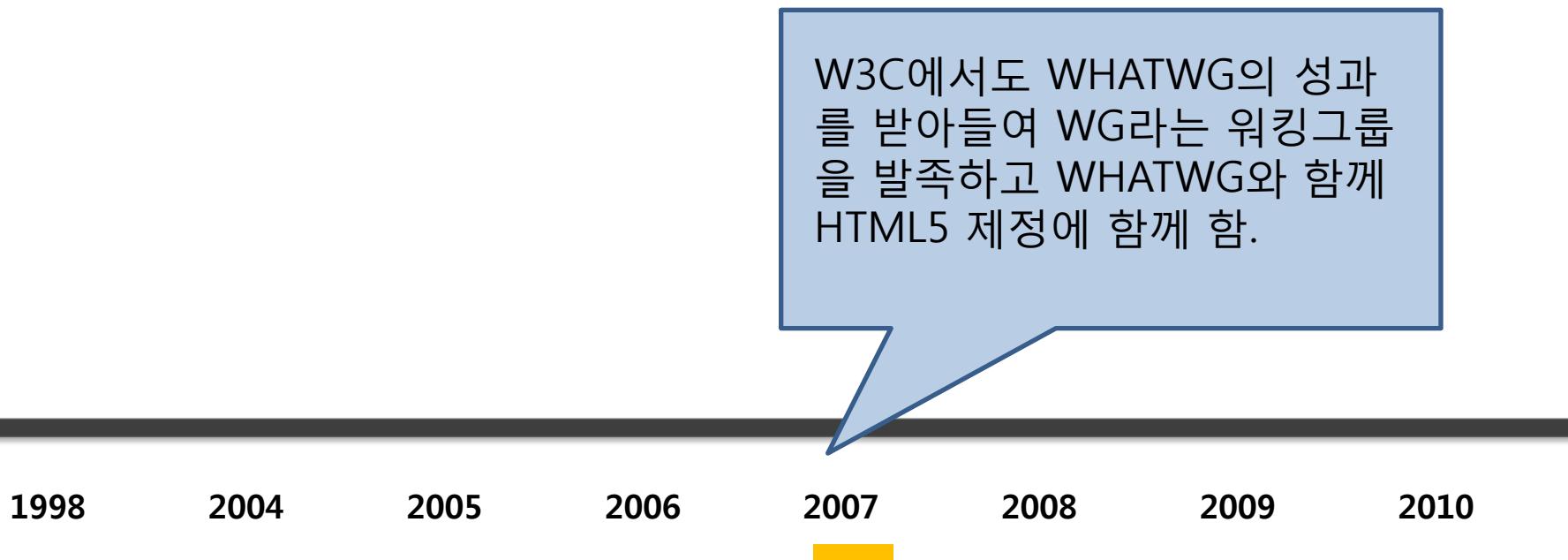


# HTML5 Timline

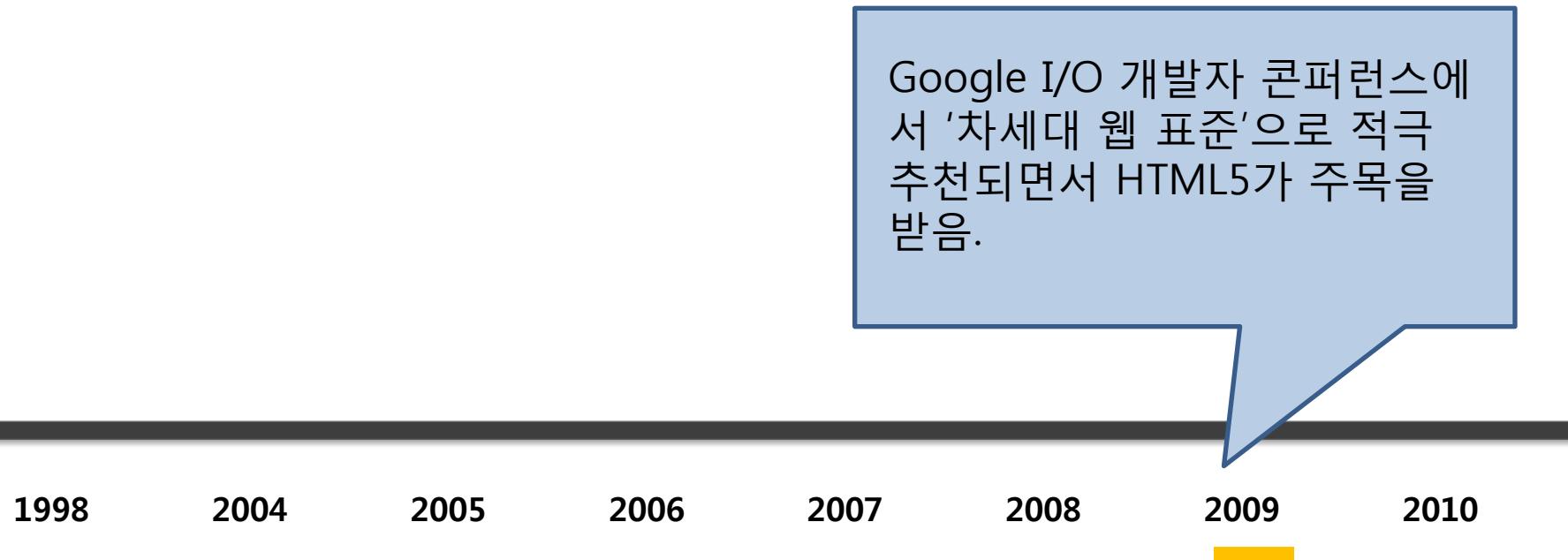


1998      2004      2005      2006      2007      2008      2009      2010

# HTML5 Timline



# HTML5 Timline



# HTML5 Timline



최종 권고는 완성을 의미.  
HTML5는 모든 기능을 다 갖추고 사용할 필요는 없음.  
지금부터 준비해야 함

HTML5 최종 심사 청구 2012 (예상)  
HTML5 W3C 권고 2022 (예상)

1998

2004

2005

2006

2007

2008

2009

2010

웹 브라우저 **플러그인으로부터 자유로워지는 웹**

좀 더 **포괄적이면서 유연한** 표준

플래시를 대체

스마트폰부터 적용

차세대 웹 표준 기술로서의 HTML5

- **section** : 일반적인 문서나 애플리케이션 영역을 표시합니다. 섹션의 제목을 나타내는 h1 ~ h6와 함께 사용할 수 있습니다.
- **article** : 뉴스 기사나 블로그 글 같은 독립적인 콘텐츠를 표시합니다.
- **aside** : 문서의 주요 부분을 표시하고 남는 사이드 바 콘텐츠를 표시합니다.
- **hgroup** : 머리말을 표시합니다.
- **header** : 헤더 부분으로, 사이트 소개나 내비게이션 등을 표시합니다.
- **footer** : 푸터 부분을 표시합니다. 사이트 제작자나 저작권 정보 등을 나타낼 때 주로 사용합니다.
- **nav** : 사이트 안의 내비게이션 요소를 표시합니다.
- **figure, figcaption** : 그림이나 비디오 같은 멀티미디어 콘텐츠에 캡션을 붙이려고 할 때 사용합니다.
- **audio, video** : 멀티미디어 컨텐츠를 표시합니다. 둘다 애플리케이션 개발자들이 사용자 인터페이스를 직접 만들 수 있는 API를 제공 합니다.

# HTML5의 새로운 태그(요소)



- embed : 플러그인 컨텐츠를 표시합니다.
- mark : 별도로 표시한 콘텐츠를 표시합니다.
- progress : 시간이 걸리는 작업 과정을 표시합니다.
- meter : 측정값을 표시합니다.
- time : 날짜나 시간을 표시합니다.
- ruby, rt, rp : 루비 언어를 표현합니다.
- canvas : 동적인 비트맵 그래픽을 표시합니다.
- command : 사용자 실행 명령어를 표시합니다.
- details : 사용자 요청 후에 얻은 콘트롤이나 부가 정보를 표시합니다.
- datalist : input에 대한 새 list와 함께 콤보박스를 만듭니다.
- keygen : 키의 페어를 생성하기 위한 컨트롤을 나타냅니다.
- output : 출력 결과를 나타냅니다.

- media : a와 area 요소에서 사용할 수 있습니다.
- ping : a와 area 요소에서 브라우저가 URI에 ping 정보를 보냈는지 사용자에게 알려줍니다. 개인정보 노출을 우려하는 사용자를 위해 끌 수 있는 기능도 제공합니다.
- hreflang, rel : area 요소에서 사용합니다.
- target : base 요소에서 사용합니다.
- value : li 요소에서 계속 지원됩니다.
- start : ol 요소에서 계속 지원됩니다.
- charset : meta 요소에서 문자 인코딩을 위해 지원합니다.
- autofocus : input (type이 hidden일 때를 제외), select, textarea, button 요소에 추가 됩니다.
- placeholder : input, textarea 요소에서 사용할 수 있습니다.
- form : input, output, select, textarea, button, fieldset 요소들이 한 개 이상의 폼 양식과 함께 사용합니다.
- required 속성은 : input (type 속성이 hidden, image, submit 같은 버튼일 때 제외), textarea에서 사용합니다.

- disabled : fieldset 요소에서 사용합니다.
- autocomplete, min, max, multiple, pattern, step 등 : input 요소에서 사용할 수 있습니다.
- novalidate : form 요소에서 사용합니다.
- formaction, formenctype, formmethod, formnovalidate, and formtarget : input, button, form 요소에서 사용합니다.
- type, label : menu 요소에서 사용합니다.
- scoped : style 요소에서 사용합니다.
- async : script 요소에서 사용합니다.
- menifest : html 요소에서 사용합니다.
- sizes : link 요소에서 사용합니다.
- reversed : ol 요소에서 사용합니다.
- seamless, sandbox : iframe 요소에서 사용합니다.

# HTML5의 변경 태그(요소)



- a : href 없이 사용하면 '작업용 링크'로 사용 합니다.
- address : 요소는 섹션 개념을 통해 새로운 영역을 표시합니다.
- b : 텍스트를 진하게 표시할 뿐만 아니라, 제품을 소개할 때의 제품명, 문서 초록에서의 키워드처럼 특별하게 중요하지는 않지만 진하게 표시하고자 할 때 사용합니다.
- hr : 단락 단위로 주제를 바꾸려고 할 때 사용합니다.
- i : 텍스트를 기울어지게 표시할 뿐만 아니라, 선박 이름, 머리 속의 생각, 다른 언어에서 숙어, 기술적인 용어, 생물종 표시처럼 소리나 분위기를 대체하는 문구 등에 사용합니다.
- label : 브라우저가 더 이상 라벨과 컨트롤 사이에 포커스 이동을 하면 안됩니다.
- menu : 실제 문서 메뉴 정보를 제공하는 데 사용합니다.
- small : 세부 주석이나 법적 인쇄 문서에서 작은 인쇄 정보를 표시할 때 사용합니다.
- strong : 단순히 진하게 표시하는 것보다 더 중요한 정보를 표시할 때 사용합니다.

- img의 border 속성 : 값이 "0"일 때만 사용하고 가급적 CSS를 사용합니다.
- script의 language 속성 : 값이 "JavaScript" (대소문자 구별)에만 사용하고 type 속성과 함께 쓰지 않고 생략할 수 있습니다.
- a의 name 속성 : id로 바꾸어 쓰기를 권장합니다.

# HTML5의 중단 태그(요소)



- basefont
- big
- center
- font
- s
- strike
- tt
- u
  
- frame
- frameset
- noframes

- basefont
  - big
  - center
  - font
  - s
  - strike
  - tt
  - u
  - frame
  - frameset
  - noframes
- acronym : 혼란스럽기 때문에 제외합니다. 축약어 abbr를 사용합니다.
  - applet : 이미 object로 대체되었습니다.
  - isindex : 이미 폼 양식을 통해 대체되었습니다.
  - dir : 이미 ul 요소를 통해 대체되었습니다.
  - noscript는 HTML 문법에서만 사용하고 XML 문법에서는 더 이상 사용하지 않습니다.

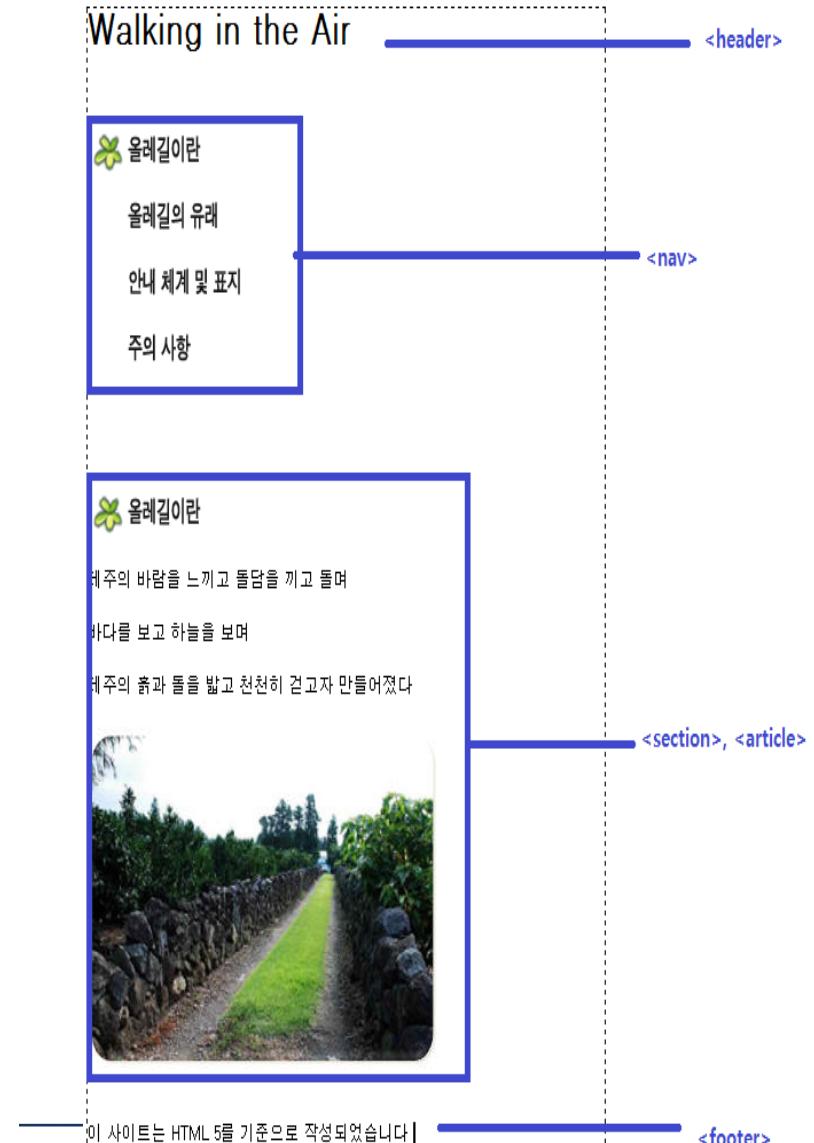
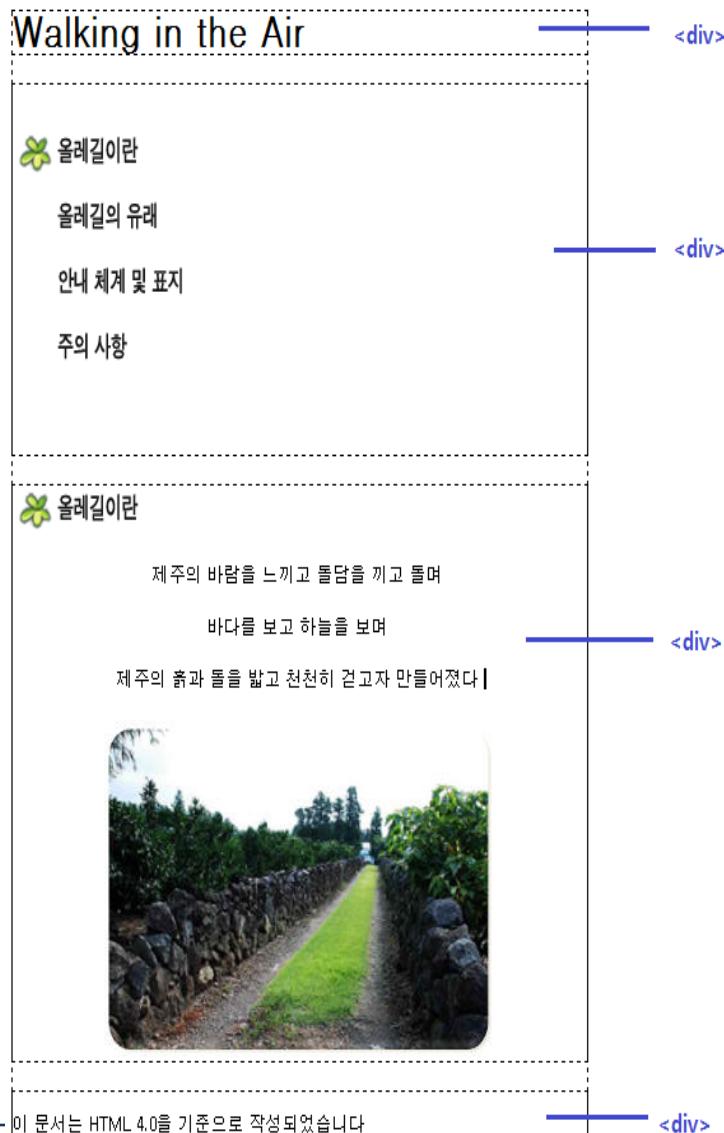
- align 속성: caption, iframe, img, input, object, legend, table, hr, div, h1, h2, h3, h4, h5, h6, p, col, colgroup, tbody, td, tfoot, th, thead 및 tr.
- alink, link, text and vlink 속성: body.
- background 속성: body.
- bgcolor 속성: table, tr, td, th 및 body.
- border 속성: table and object.
- cellpadding 및 cellspacing 속성: table.
- char 및 charoff 속성: col, colgroup, tbody, td, tfoot, th, thead 및 tr.
- clear 속성: br.
- compact 속성: dl, menu, ol 및 ul.
- frame 속성: table.
- frameborder 속성: iframe.

- height 속성: td 및 th.
- hspace 및 vspace 속성: img 및 object.
- marginheight 및 marginwidth 속성: iframe.
- noshade 속성: hr.
- nowrap 속성: td 및 th.
- rules 속성: table.
- scrolling 속성: iframe.
- size 속성: hr.
- type 속성: li, ol and ul.
- valign 속성: col, colgroup, tbody, td, tfoot, th, thead and tr.
- width 속성: hr, table, td, th, col, colgroup 및 pre.

- LocalStorage : localStorage를 이용하면 게시판이나 블로그에서 글을 작성할 때 사용자 컴퓨터에 임시 저장하는 기능을 쉽게 구현할 수 있습니다.
- Drag & Drop : 웹 페이지 상에서 원하는 항목을 드래그할 수 있는 API
- Geolocation : GPS가 내장된 모바일 기기 등에서 자신의 현재 위치를 알아낼 수 있는 API들이 추가됩니다.
- Canvas 2D API : canvas 요소와 그리기 API를 제공하여 그래픽 환경을 제공합니다.
- Web Workers : CPU 부하를 많이 차지하는 작업을 여러 워커(worker)로 나누어 작업할 수 있는 자바스크립트 API를 제공합니다.
- Indexed DB API : SQL 같은 별도의 언어를 배우지 않아도 되며 간단한 자바스크립트 API 만으로 데이터베이스 조작이 가능합니다.

## **2. HTML5로 기본적인 문서 만들기**

# HTML4 문서 vs HTML5 문서



# HTML4 문서 vs HTML5 문서



Walking in the Air

올레길이란  
올레길의 유래  
안내 체계 및 표지  
주의 사항

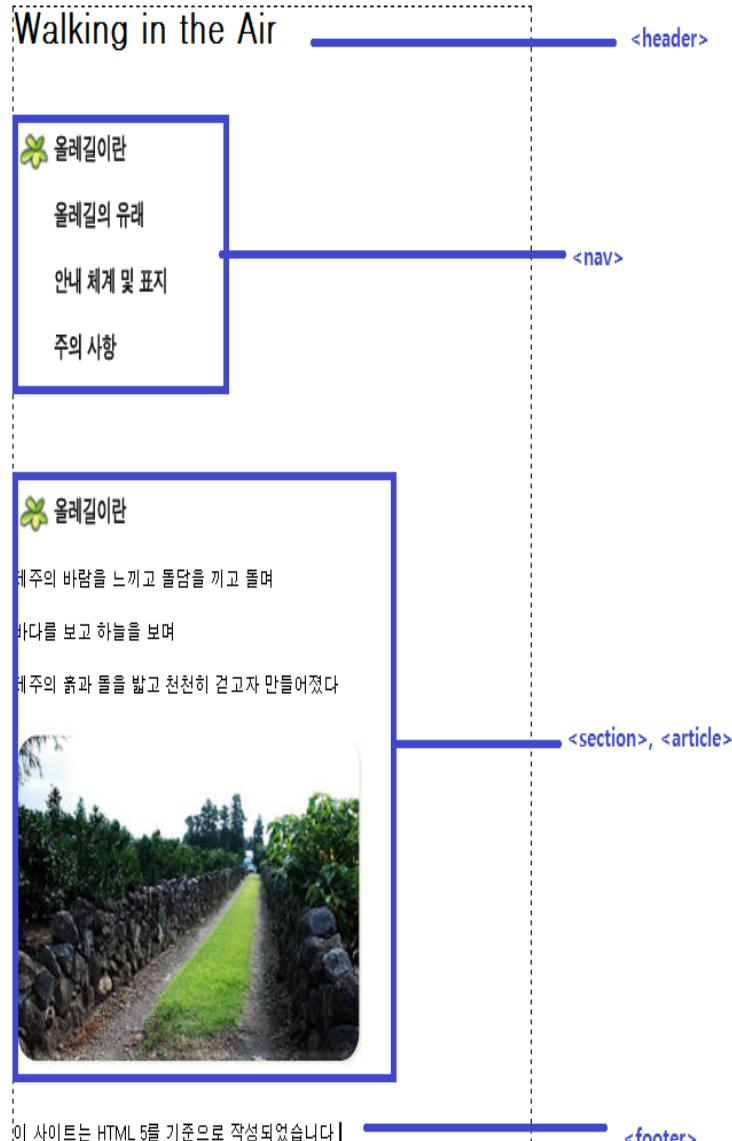
올레길이란  
제주의 바람을 느끼고 풀담을 끼고 둘며  
바다를 보고 하늘을 보며  
제주의 꽃과 돌을 밟고 천천히 걷고자 만들어졌다!



이 문서는 HTML 4.0을 기준으로 작성되었습니다.

```
<body>
<div id="header">
  <h1> walking in the air</h1>
</div>
<div id="nav">
  <ul>
    <li>올렛길이란</li>
    <li>올렛길의 유래</li>
    ...
  </ul>
</div>
<div id="sidebar">
  ...
</div>
<div id="content">
  
  제주의 바람을 느끼고..
</div>
<div id="footer">
  이 문서는 HTML 4를 기준으로 작성되었습니다.
</div>
</body>
</html>
```

# HTML4 문서 vs HTML5 문서



```
<body>
<header>
  <h1> walking in the air</h1>
</header>
<nav>
  <ul>
    <li>올렛길이란</li>
    <li>올렛길의 유래</li>
    ...
  </ul>
</nav>
<aside>
  ...
<section>
  <article>
    
    제주의 바람을 느끼고..
  </article>
</section>
<footer>
  이 문서는 HTML 5를 기준으로 작성되었습니다.
</footer>
</body>
```

DOCTYPE : 현재 문서의 유형을 정의 [html](#)

## HTML 4.01

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Strict//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

DOCTYPE : 현재 문서의 유형을 정의

## HTML 5

```
<!DOCTYPE html>
```

시맨틱 태그 : 태그 이름만 보고도 역할을 짐작할 수 있는 태그

## 왜 사용할까?

- 검색 엔진에서 쉽게 웹 문서 내용을 검색할 수 있다.
- 화면 판독기 같은 웹 보조 기구에서 웹 문서를 정확하게 알 수 있다.
- 어떤 장치에서나 똑같이 문서를 해석할 수 있다 → 웹 문서의 표준 시작

# HTML5의 시맨틱 태그 - <header>



<header> - 사이트 전체의 머리말 또는 포스트의 머리말

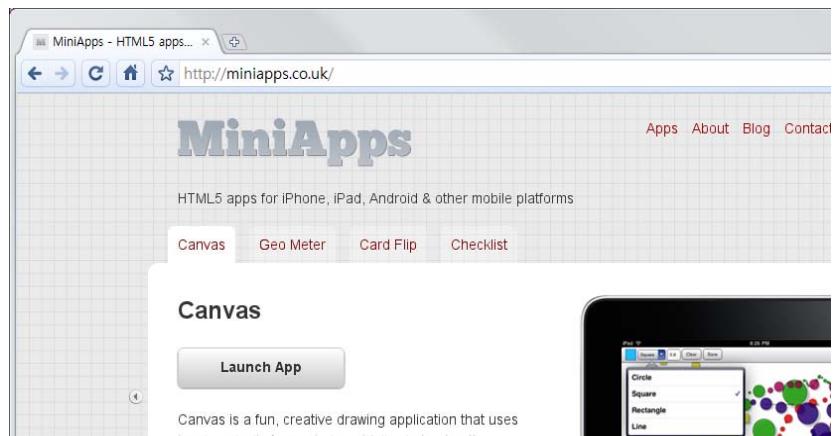


```
<header id="mainHeader">  
  <nav>  
    메뉴  
  </nav>  
  <h1>Designer Tweet </h1>  
</header>
```

# HTML5의 시맨틱 태그 - <hgroup>



<hgroup> - 제목과 그와 관련된 부제목을 묶어줌.



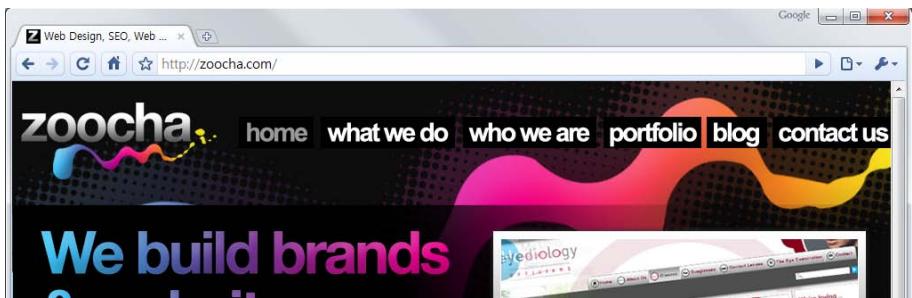
사이트 제목(MiniApps)과 부제목(HTML5...platforms)을 <hgroup>으로 묶게 되면 웹 페이지 전체 구조를 만들 때 문서에서 중요한 뼈대가 어떤 것인지를 쉽게 알 수 있고 그만큼 문서를 빠르고 정확하게 조직할 수 있음.

```
<header>
<nav> ... </nav>
<hgroup>
  <h1>MiniApps </h1>
  <h2>HTML5 apps for iPhone, iPad,  
Android & other mobile  
platforms</h2>
</hgroup>
</header>
```

# HTML5의 시맨틱 태그 - <nav>



<nav> – 문서 안에서, 혹은 다른 문서로 링크하기 위한 내비게이션 링크  
– 위치에 영향을 받지 않음



<nav> 요소가 <header> 요  
소 안에 포함되어 있음. 따라  
서 헤더 위치를 옮기게 되면  
헤더 안의 메인 메뉴 위치도  
옮겨지게 됨.

```
<header>
<a href="/"><img src.."></a>
<nav>
<ul>
<li><a href=..">Homepage</a></li>
<li><a href=..">What we do</a></li>
<li><a href=..">Who we are</a></li>
<li><a href=..">Portfolio</a></li>
<li><a href=..">Blog</a></li>
<li><a href=..">Contact</a></li>
</ul>
</nav>
</header>
```

# HTML5의 시맨틱 태그 - <nav>



<nav> – 문서 안에서, 혹은 다른 문서로 링크하기 위한 내비게이션 링크  
– 위치에 영향을 받지 않음

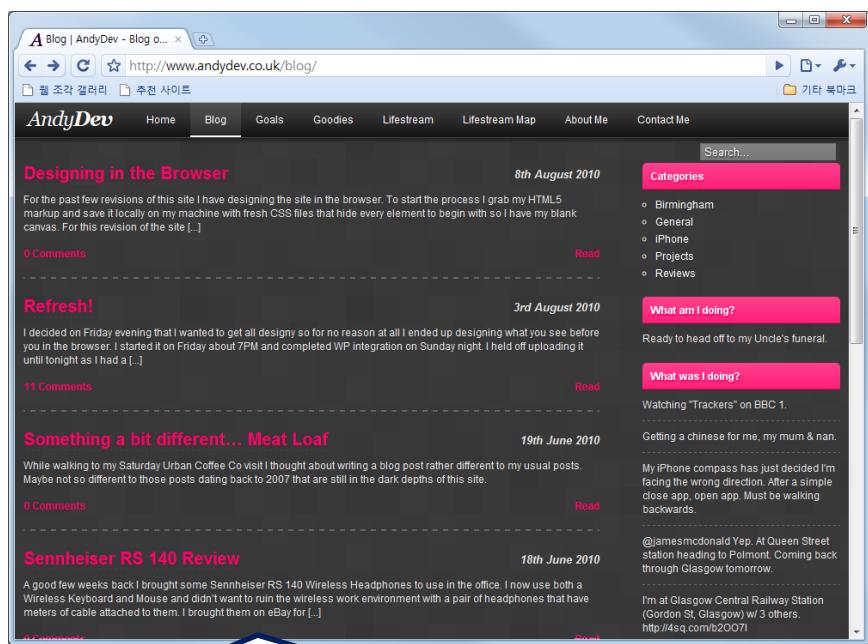
The screenshot shows the homepage of Project Fedena. At the top, there is a navigation bar with links: HOME, ABOUT, BLOG, FORUMS, DOWNLOAD, INSTALL, CONTRIBUTORS, and CONTACT. Below the navigation bar, there is a main content area with various sections like 'BRINGING A REVOLUTION IN THE FIELD OF EDUCATION', 'JOIN THE DEVELOPMENT TEAM', 'SEE THE LIST OF COMPANIES', 'GET INVOLVED', 'FEDENA CODE WARS', and 'TOP CONTRIBUTORS'. A large blue box highlights the text: '헤더가 사이트 왼쪽에 위치. <nav> 요소는 헤더와 떨어져 있음.'

```
<nav class="top-nav">
<a href="/"></a>
<ul>
<li><a href="/">Home</a></li>
<li><a href="/about">About</a></li>
<li><a href="/blog">Blog</a></li>
<li><a href="/forum">Forums</a></li>
<li><a href="/download">Download</a></li>
<li><a href="/install">Install</a></li>
<li><a href="/users">Contributors</a></li>
<li><a href="/contact">Contact</a></li>
</ul>
</nav>
```

# HTML5의 시맨틱 태그 - <article>



<article> – 웹 상의 실제 내용. 블로그의 포스트나 웹 사이트 내용, 코멘트 등.  
– 재배포 가능



하나의 <article> 태그 안에 포함된  
포스트 제목과 포스팅 날짜, 포스팅  
내용

```
<article class="blog clearfix">
  <h3>
    <a href="#">Designing in the Browser</a>
  </h3>
  <time datetime="2010-08-08T10:08:05+01:00" class="data">8th August 2010</time>
  <p>For the past few revisions of this site I have been designing the site in the browser. To start the process I grab my HTML5 markup and save it locally on my machine with fresh CSS files that hide every element to begin with so I have my blank canvas. For this revision of the site [...]

0 Comments Read



Refresh! 3rd August 2010



I decided on Friday evening that I wanted to get all design done so for no reason at all I ended up designing what you see before you in the browser. I started it on Friday about 7PM and completed WP integration on Sunday night. I held off uploading it until tonight as I had a [...]



11 Comments Read



Something a bit different... Meat Loaf 19th June 2010



While walking to my Saturday Urban Coffee Co visit I thought about writing a blog post rather different to my usual posts. Maybe not so different to those posts dating back to 2007 that are still in the dark depths of this site.



0 Comments Read



Sennheiser RS 140 Review 18th June 2010



A good few weeks back I bought some Sennheiser RS 140 Wireless Headphones to use in the office. I now use both a Wireless Keyboard and Mouse and didn't want to ruin the wireless work environment with a pair of headphones that have meters of cable attached to them. I brought them on eBay for [...]



0 Comments Read



@jamesmcdonald Yep. At Queen Street station heading to Polmont. Coming back through Glasgow tomorrow.



I'm at Glasgow Central Railway Station (Gordon St, Glasgow) w/ 3 others. http://4sq.com/b2007


```

# HTML5의 시맨틱 태그 - <section>



<section> – 제목을 붙여서 콘텐츠 그룹을 묶은 것.

The screenshot shows the Areya webservices website with red boxes highlighting different sections:

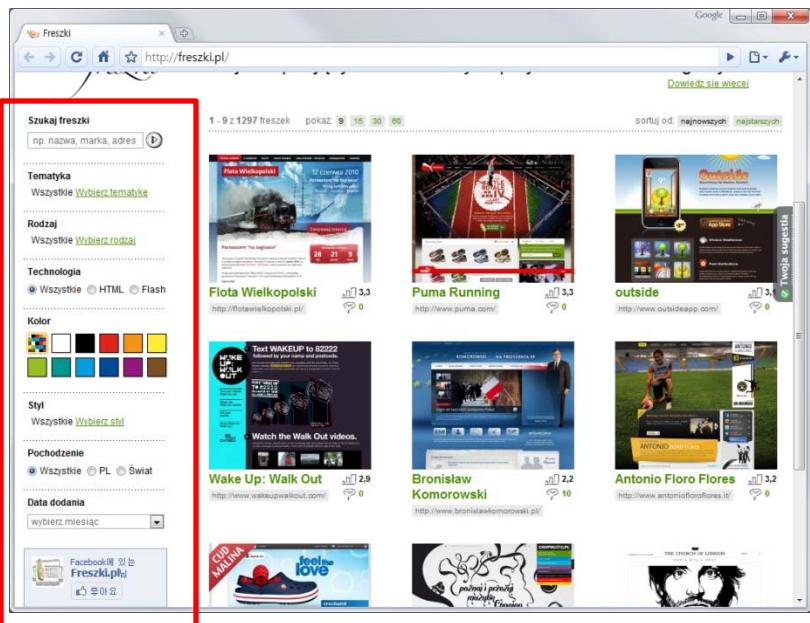
- 헤더 섹션**: The top navigation bar with links like home, portfolio, über uns, kontakt, FAQ, Support, Webmail, and Kundencenter.
- 본문 내용 섹션**: The main content area divided into four columns:
  - Webhosting**: Describes domain registration, security, and fast service.
  - Webdesign**: Describes HTML/CSS, design techniques, and how clients can easily update their websites.
  - Administration & Verwaltung**: Describes CMS, system management, and SEO optimization.
  - Werbung**: Describes advertising services.
- 프로젝트 소개 섹션**: A section titled "aktuelle Projekte" showing thumbnails of various projects like "zartesfleisch.com" and "trailerpark.info".
- 푸터 섹션**: The footer containing contact form fields for Name, Email, and Message, along with a "absenden" button.

```
<body>
<section class="wrapper clear section">
  <header class="header">
    ...
  </header>
</section>
<section class="section main clear">
  <article class="article"> ...</article>
  <article class="article"> ...</article>
  ...
</section>
<section class="wrapper clear section contact">
  <h2 class="left">aktuelle Projekte</h2>
  ...
</section>
<section class="wrapper clear section foot-section">
  <div class="contact-inner" id="contact"> ....
</div>
  ...
</section>
</body>
```

# HTML5의 시맨틱 태그 - <aside>



<aside> – 본문을 표시하고 남는 내용



<!-- sidebar -->

<aside>

```
<form action="" method="post"
id="freshListSearch">
<fieldset class="search">
<h3><label for="fSearch">Szukaj
freszki</label></h3>
<input type="text" name="txtsearch"
id="fSearch" value="np. nazwa, marka,
adres" />
<input type="hidden"
name="baseURL"
value="/najnowsze/h/n1n9/-/f/r/e/s/z/k/i/"
id="fBaseUrl" />
<button
type="submit">Szukaj</button>
.....
</aside>
<!-- sidebar -->
```

# HTML5의 시맨틱 태그 - <footer>



<footer> – 저작권 정보나 사이트 제작자 정보 표시



```
<!-- sidebar -->
<aside>
<form action="" method="post"
id="freshListSearch">
<fieldset class="search">
<h3><label for="fSearch">Szukaj  
freszki</label></h3>
<input type="text" name="txtsearch"
id="fSearch" value="np. nazwa, marka,  
adres" />
<input type="hidden"
name="baseURL"
value="/najnowsze/h/n1n9/-/f/r/e/s/z/k/i/"
id="fBaseUrl" />
<button
type="submit">Szukaj</button>
.....
</aside>
<!-- sidebar -->
```

# HTML5의 시맨틱 태그 - <figure> & <figcaption>



<figure> – 캡션을 붙이려고 하는 요소에 함께 사용함. 표나 그림, 소스 코드 등 어디에나 사용할 수 있음.

<figcaption> - <figure>를 사용한 대상에 캡션을 붙임.



<figure>

```

```

<figcaption>

요즘에는 보기 힘든 빨간색 우체통

```
</figcaption>
```

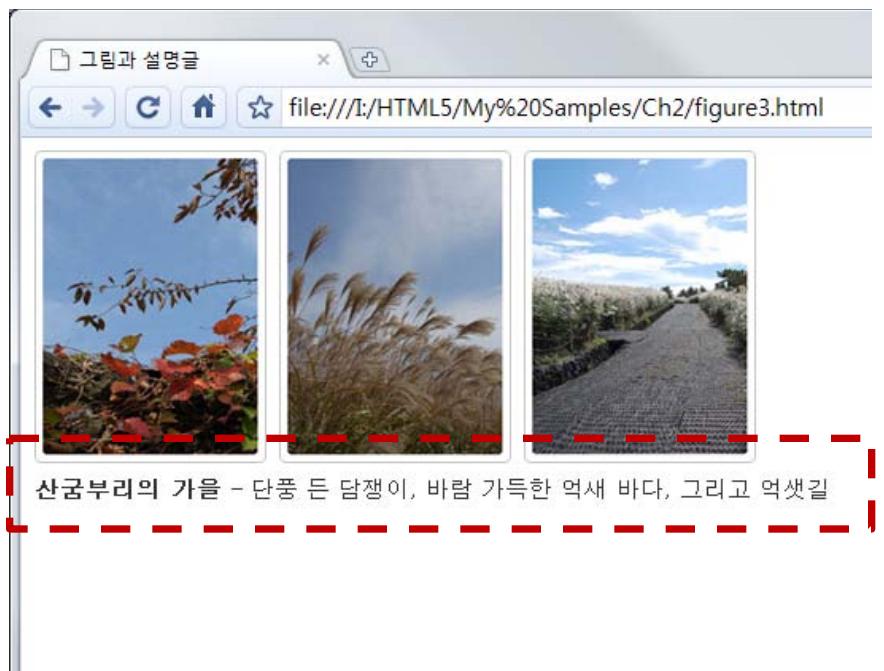
```
</figure>
```

# HTML5의 시맨틱 태그 - <figure> & <figcaption>



<figure> – 캡션을 붙이려고 하는 요소에 함께 사용함. 표나 그림, 소스 코드 등 어디에나 사용할 수 있음.

<figcaption> - <figure>를 사용한 대상에 캡션을 붙임.



## <figure>

```

```

```

```

```

```

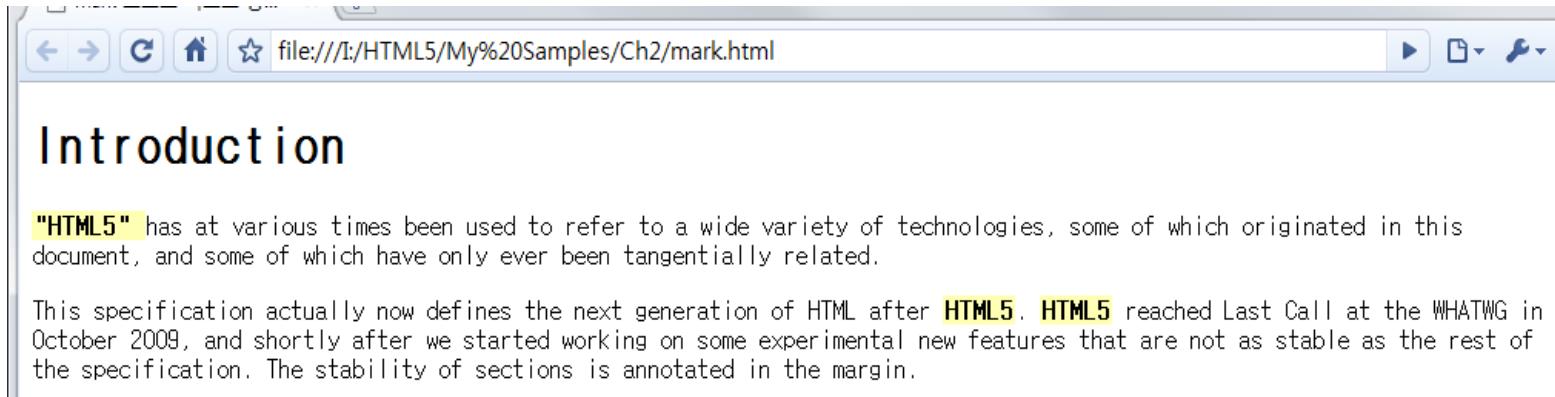
## <figcaption>

```
<b>산굼부리의 가을</b> - 단풍 든 담쟁이, 바람 가득한 억새 바다, 그리고 억새길
```

```
</figcaption>
```

```
</figure>
```

<mark> – css를 이용해 배경색이나 글자색을 지정해서 텍스트를 강조함



The screenshot shows a web browser window with the URL "file:///I:/HTML5/My%20Samples/Ch2/mark.html". The page content includes an 

## Introduction

 heading and two paragraphs. The first paragraph contains the text "'**HTML5**' has at various times been used to refer to a wide variety of technologies, some of which originated in this document, and some of which have only ever been tangentially related." The word "HTML5" is highlighted with a yellow background. The second paragraph continues with "This specification actually now defines the next generation of HTML after **HTML5**. **HTML5** reached Last Call at the WHATWG in October 2009, and shortly after we started working on some experimental new features that are not as stable as the rest of the specification. The stability of sections is annotated in the margin."

```
<head>
<style>
  mark{background-color:#FFFFB3;font-weight:bold;padding:1px}
</style>
</head>
<body>
  <h1>Introduction</h1>
  <p> <mark> "HTML5" </mark> has at various times ....</p>
  <p>... generation of HTML after <mark> HTML5</mark> . <mark> HTML5
</mark> reached Last Call at ...</p>
```

# HTML5의 시맨틱 태그 - <time>



<time> – 날짜와 시간 정보를 표현하는 마이크로포맷 태그



```
<section id="page">
  <header class="summary">
    <h1><a title="One day JavaScript
conference" class="url"
href="http://2009.full-frontal.org/">Full
Frontal</a></h1>
    <h2>자바스크립트 컨퍼런스 <time
datetime="2009-11-20T09:00">2009년 11월
20일</time></h2>
  </header>
  <section id="register">
    <a href="http://2010.full-
frontal.org/">2010년 컨퍼런스 일정 보기
    </a>
  </section>
```

# HTML5의 시맨틱 태그 – <address>



<address> – 웹 페이지의 연락처 정보 (제작자 이름, 웹 페이지 주소, 이메일 주소, 전화번호 등)



<footer>

<address>

by <a

href="http://http://www.webguru.pe.kr"

target="\_blank">webguru</a> ,

mailto:<a

href="mailto:funcom@gmail.com">funcom  
@gmail.com</a>

</address>

</footer>

## <i> : 중요하진 않지만 기울어진 텍스트

외국어나 분류학적인 용어, 기술 용어, 선박 이름, 대본에서의 무대 지문, 일부 악보 기록, 또는 생각이나 손으로 직접 쓴 텍스트 등 다른 텍스트와 구별되는 텍스트를 표현할 때 사용

## <b> : 중요하진 않지만 진한 텍스트

제품 소개 페이지에서의 제품 이름이나 문서 초록의 키워드처럼 특별히 중요하지는 않지만 진하게 표시하려고 할 경우에 사용

## <small> : 포커스를 받지 못하는 텍스트

저작권 문장이나 라이센스 정보처럼 주 포커스를 받지 못하는 텍스트 콘텐츠용으로 사용.  
단, <p> 요소를 사용한 단락이나 목록 같은 블록 단위 콘텐츠에서는 <small> 요소를 사용하면 안됨.

## <hr> : 단락의 주제를 바꿈

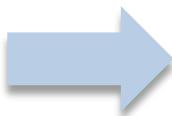
웹 페이지 안에서 주제가 바뀔 때 사용.

### **3. HTML5 비디오 요소**

- MPEG(\*.mp4, \*m4v) : 영상과 음성을 디지털 데이터로 전송하고 저장하는 규격중 하나. 최근엔 H.264 코덱이 MPEG-4의 part10 AVC로 규격화되었음.
- 플래시 비디오(\*.flv) : 플래시 플레이어에서만 재생할 수 있는 동영상.
- Ogg(\*.ogv) : 공개 소스 파일 형식. 무료로 사용할 수 있기 때문에 여러 미디어 플레이어에서 사용 중.
- AVI(\*.avi) : 윈도우 운영체제에서 오디오와 비디오를 합쳐 지원하는 파일 형식. 화질이 뛰어난 반면 용량이 크기 때문에 실시간 영상으로는 적합하지 않음.
- WebM(\*.webm) : 구글이 발표한 오픈 소스 방식의 동영상 파일.



인코딩



.mp4, .mov, .avi  
etc...

디코딩



- H.264 - 매우 높은 압축률을 가지면서 고화질의 영상 지원
  - 주로 .mp4 파일에 많이 사용됨. 사파리, 크롬 브라우저에서 지원
  - 유료 코덱이지만 HTML5 동영상 재생 용도로 사용할 경우에 한해 무료 사용 가능
- 오그 테오라 – 공개 코덱(무료 코덱). \*.ogv 파일에 주로 사용됨.
  - 파이어폭스 등에서 지원
- VP8 – 공개 코덱. \*.webm 파일에서 주로 사용됨

- **MP3 (MPEG-1 AUDIO Layer3)**

- MP3는 오디오 압축 형식으로, PCM 오디오 데이터 중에서 실제 사람이 들을 수 없는 부분이나 필요없는 부분을 버리고 다시 인코딩하는 방식
- 음질을 좋게 하려면 파일 크기가 커짐
- MP3는 1991년에 표준화되었고 특허권이 등록되어 있음.

- **AAC(Advanced Audio Coding)**      有.      X

- 애플의 아이폰, 아이팟, 아이튠즈에서 사용되는 기본 오디오 형식
- 1997년에 표준화되었고 특허권이 등록되어 있음.

- **Vorbis**

- 오픈 소스이며 누구나 무료로 사용할 수 있는 오디오 코덱
- 확장자는 .ogg나 .oga를 사용
- 재생할 수 있는 플레이어가 적고 인코딩 시간이 더 걸린다는 단점
- 무료로 사용할 수 있기 때문에 PC 게임 등에 많이 사용됨.

# 각 브라우저별 지원하는 비디오 코덱



브라우저	버전	비디오 코덱
파이어폭스	3.6	ogg/theora
	4	ogg/theora, WebM, H.264**
사파리	3	H.264
IE	9	H.264, WebM*
	10	H.264
크롬	9.0	ogg/theora, H.264
	10.0	ogg/theora, WebM, H.264*
오페라	10.5	ogg/theora
	11.1	ogg/theora, WebM

\* : WebM의 경우 하드웨어 가속을 필요로 하기 때문에, 현재까지 모바일 기기에서 완벽하게 지원지 못함.

\*\* : 공식적인 지원이 아니라 플러그인 형태로 사용할 수 있음

# <video> 태그



속성	설명
<b>src</b>	동영상 파일의 경로
<b>poster</b>	동영상이 화면에 나타나지 않을 때 대신 표시할 그림
<b>preload</b>	재생 단추를 눌렀을 때 동영상이 재생되게
<b>autoplay</b>	동영상 자동 재생
<b>loop</b>	반복 재생 횟수
<b>controls</b>	동영상 화면에 컨트롤 막대 추가
<b>width</b>	동영상 화면의 너비
<b>height</b>	동영상 화면의 높이

# <video> 태그



- 동영상의 경우에도 소스 파일의 경로와 위치만 지정하면 쉽게 삽입할 수 있음
- 단, HTML5의 경우 동영상은 자동으로 재생하지 않기 때문에 동영상 화면에 컨트롤 막대를 추가하여 재생 단추를 표시하거나 자동 재생 되도록 해야 함.

Ex)

```
<video src="ThankYou.mp4" width="720" height="480" autoplay>
```

- poster 이미지는 동영상의 크기와 같은 크기의 이미지를 사용하는 것이 좋음.

Ex)

```
<video src="carrot.mp4" width="480" height="320" autoplay  
poster="poster.jpg">
```

- controls 속성은 동영상 화면에 컨트롤 막대를 표시할지 여부를 지정 컨트롤 막대의 형태나 바로가기 메뉴는 웹 브라우저에 따라 달라질 수 있음.

Ex)

```
<video src="ThankYou.mp4" width="480" height="320" autoplay controls>
```

## ① MIME 유형 설정

동영상 파일을 볼 수 있도록 리눅스/아파치 서버 상에서 MIME 유형을 선언해야 합니다. 서버 관리자에게 다음과 같이 설정해 달라고 요청하면 됩니다.

video/ogg .ogv

video/mp4 .mp4

video/webm .webm

## ② <video> 태그를 사용해 동영상 파일의 크기와 기본 속성을 지정합니다.

```
<video width="480" height="320" autoplay controls poster="without.jpg">  
</video>
```

# <video> 태그 – 모든 코덱을 고려해 작성하려면



③ 코덱에 따라 달라지는 동영상 파일은 <source> 태그를 사용해서 연결합니다.

```
<video width="480" height="320" autoplay controls poster="without.jpg">
  <!-- ogv는 예전 파이어폭스와 오페라 -->
  <source src="without.ogv" type="video/ogg">
  <!-- mp4는 사파리, IE9, 아이폰, 아이패드, 안드로이드, 윈도우 7 폰 -->
  <source src="without.mp4" type="video/mp4">
  <!-- WebM는 파이어폭스4, 오페라, 크롬 -->
  <source src="without.webm" type="video/webm">
</video>
```

# <audio> 태그

- <audio> 태그를 사용함으로써 웹 브라우저 자체에서 음악을 재생할 수 있음.
- <video> 태그의 속성과 거의 같음.

브라우저	버전	오디오 코덱
파이어폭스	3.6	ogg/vorbis, wav
	4	ogg/vorbis, wav
사파리	3	mp3, wav, AAC
IE	9	mp3, AAC
	10	mp3, AAC
크롬	10.0	ogg/vorbis, mp3, wav, AAC
오페라	10.5	ogg/vorbis, wav
	11.1	ogg/vorbis, wav

# <audio> 태그



- <audio> 태그를 사용함으로써 웹 브라우저 자체에서 음악을 재생할 수 있음.
- <video> 태그의 속성과 거의 같음.

```
<audio controls autoplay>  
  <source src="old_melody.ogg">  
  <source src="old_melody.mp3">  
</audio>
```

## **4. 캔버스(CANVAS)**

# <canvas> 태그



그래픽이 그려질 영역

속성	설명
id	한 페이지에 여러 개의 캔버스를 그렸을 때 캔버스 구별
width / height	캔버스 크기 지정

```
<!DOCTYPE HTML>  
<html>  
<head>  
<script type="application/javascript">  
function draw_m( ) {  
    var m_canvas = document.getElementById("m");  
    var m_ctx = m_canvas.getContext("2d");  
    m_ctx.fillRect (10, 10, 100, 100);  
    m_ctx.fillStyle = "rgb(200,0,0)";  
}  
</script>  
</head>  
<body onload="draw_m( )">  
    <canvas id="m"  
width="500" height="300"></canvas>  
</body>
```

도형을 그리는 함수 정의

ID가 m인 캔버스 객체를 불러와 m\_canvas라는 변수에 저장

드로잉 컨텍스트의 인스턴스 만듬

사각형 그림

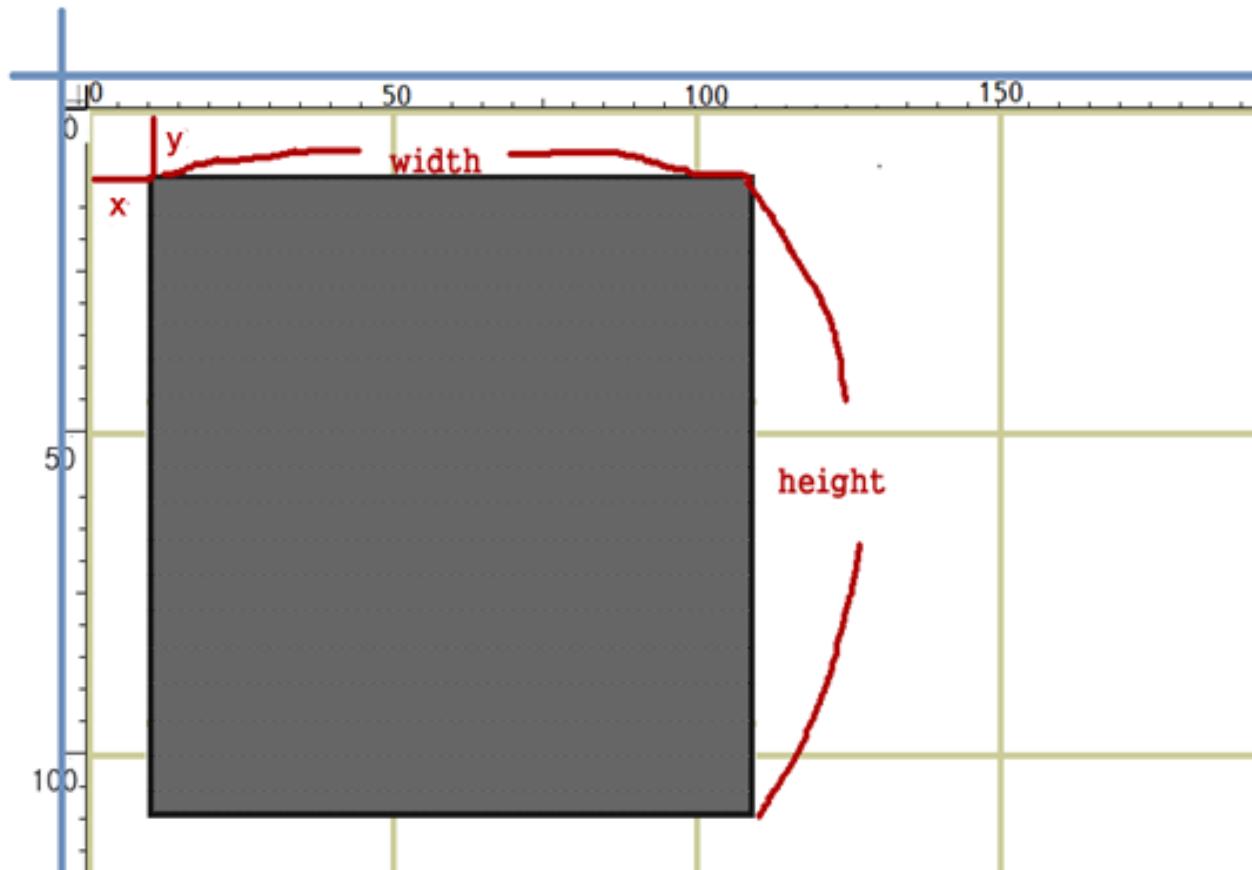
사각형을 채울 스타일 정의

onload 이벤트와 함께 함수 실행

# fillRect() 함수와 좌표

**fillRect (x, y, width, height);**

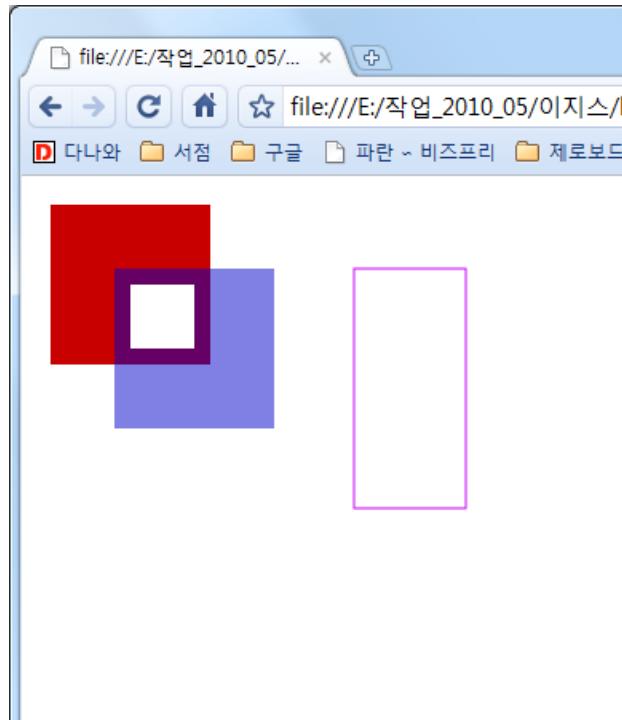
→ (x,y)라는 위치에서 시작하여 너비 width, 높이 height인 사각형



```
<script type="application/javascript">
function draw_m() {
    var canvas = document.getElementById("m");
    if(canvas.getContext) {
        var ctx = canvas.getContext("2d");
        ctx.fillRect (10, 10, 100, 100);
        ctx.fillStyle = "rgb(200,0,0)";
    }
    else {
        alert("사파리 브라우저나 파이어폭스 1.5 이상의 브라우저에서만 사이트 내용을 제대로 볼 수 있습니다.")
    }
}
```

# 사각형 그리기

- **fillRect(x, y, width, height)** : 색이 칠해진 사각형 그림
- **strokeRect(x, y, width, height)** : 테두리만 있는 사각형 그림
- **clearRect(x, y, width, height)** : 특정 영역을 지우고 완전히 투명하게 만듦



```
function draw_m() {  
    var canvas = document.getElementById("m");  
    if(canvas.getContext) {  
        var ctx = canvas.getContext("2d");  
        ctx.fillStyle = "rgb(200,0,0)";  
        ctx.fillRect (10, 10, 100, 100);  
        ctx.fillStyle = "rgba(0,0,200, 0.5)";  
        ctx.fillRect (50, 50, 100, 100);  
        ctx.clearRect(60, 60, 40, 40);  
        ctx.strokeStyle="rgb(200,0,250)";  
        ctx.strokeRect(200, 50, 70, 150);  
    }  
}
```

**beginPath()** : 경로 시작

**closePath()** : 경로 종료

**stroke()** : 경로 테두리 그리기

**fill()** : 도형 채우기

**moveto(x,y)** : (x,y) 위치로 시작점을 옮김

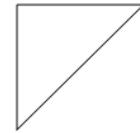
**lineto(x,y)** : x에서 y까지 직선을 그림

<삼각형을 채울 때>



```
ctx.beginPath();
ctx.moveTo(50,50);
ctx.lineTo(150,50);
ctx.lineTo(50,150);
ctx.fill();
```

<삼각형의 테두리만 그릴 때>

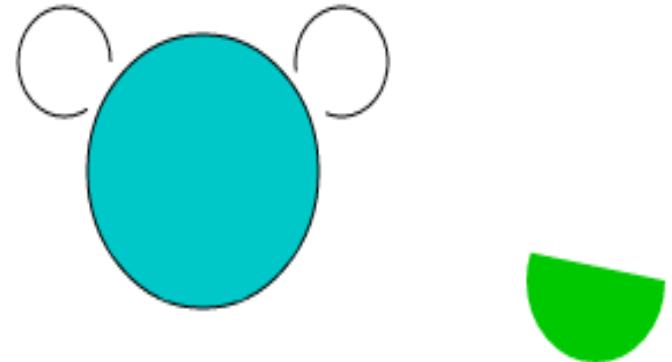


```
ctx.beginPath();
ctx.moveTo(50,50);
ctx.lineTo(150,50);
ctx.lineTo(50,150);
ctx.lineTo(50,50);
ctx.closePath();
ctx.stroke();
```

## **arc(x, y, r, startAngle, endAngle, anticlockwise)**

- (x,y)에서 시작하여 반시계방향(anticlockwise)으로 반지름(r)만큼의 원을 그림.
- startAngle / endAngle : 호의 시작점과 끝점을 각도로 표시  
각도는 라디안으로 표시 | var radians = (Math.PI/180)\*degrees
- Anticlockwise : 반시계 방향으로 그릴지 여부. true 또는 false로 지정

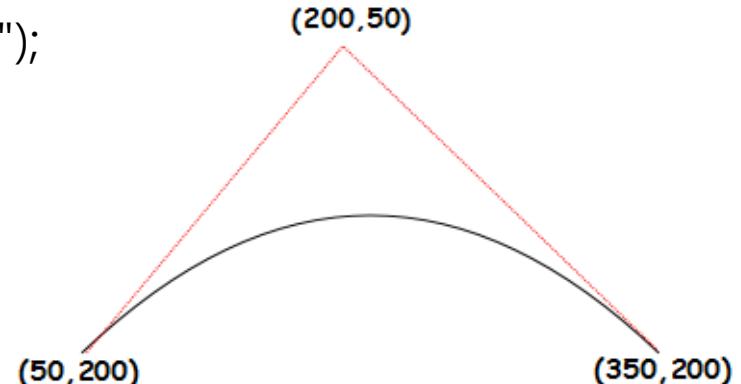
```
function draw_m() {
    var canvas = document.getElementById("m");
    if(canvas.getContext){
        var ctx = canvas.getContext('2d');
        ctx.beginPath();
        ctx.arc(70, 70, 20, 0,(Math.PI/180)*60,true);
        ctx.stroke();
        ctx.beginPath();
        ctx.arc(130, 110, 50, 0, Math.PI*2,true);
        ctx.fillStyle="rgb(0,200,200)";
        ctx.fill();
        ctx.stroke();
        ctx.beginPath();
        ctx.arc(190, 70, 20, (Math.PI/180)*110,(Math.PI/180)*170,true);
        ctx.stroke();
        ctx.beginPath();
        ctx.arc(300, 150, 30, 0,(Math.PI/180)*200,false);
        ctx.fillStyle="rgb(0,200,0)";
        ctx.fill();
    } else {
        alert("사파리 브라우저나 파이어폭스 1.5 이상의 브라우저에서만 사이트 내용을 제대로 볼 수 있습니다.")
    }
}
```



## quadraticCurveTo(cp1x, cp1y, x, y)

한 개의 조절점(cp1x, cp1y)을 이용해 (x,y)까지의 곡선을 그림

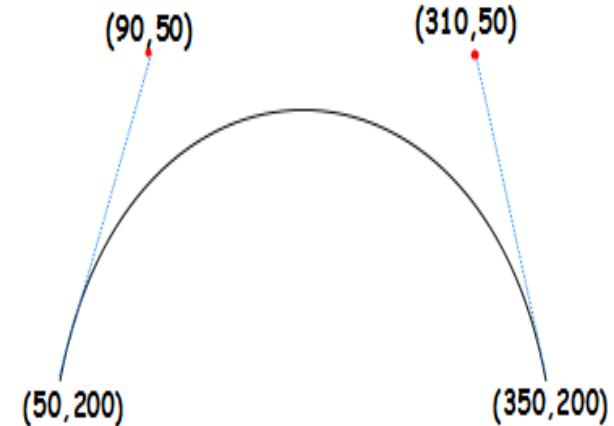
```
function draw_m() {  
    var canvas = document.getElementById("m");  
    if(canvas.getContext){  
        var ctx = canvas.getContext('2d');  
        ctx.beginPath();  
        ctx.moveTo(50,200);  
        ctx.quadraticCurveTo(200,50,350,200);  
        ctx.stroke();  
    }  
}
```



## bezierCurveTo(cp1x, cp1y, cp2x, cp2y, x, y)

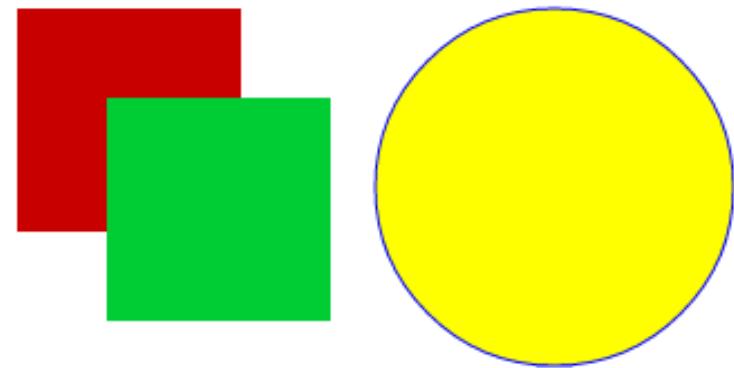
두 개의 조절점(cp1x, cp1y)와 (cp2x, cp2y)를 이용해 (x,y)까지의 곡선을 그림

```
function draw_m() {  
    var canvas = document.getElementById("m");  
    if(canvas.getContext){  
        var ctx = canvas.getContext('2d');  
        ctx.beginPath();  
        ctx.moveTo(50,200);  
        ctx.bezierCurveTo(90,50,310,50,350,200)  
        ctx.stroke();  
    }  
}
```



**fillStyle= color,      strokeStyle = color**

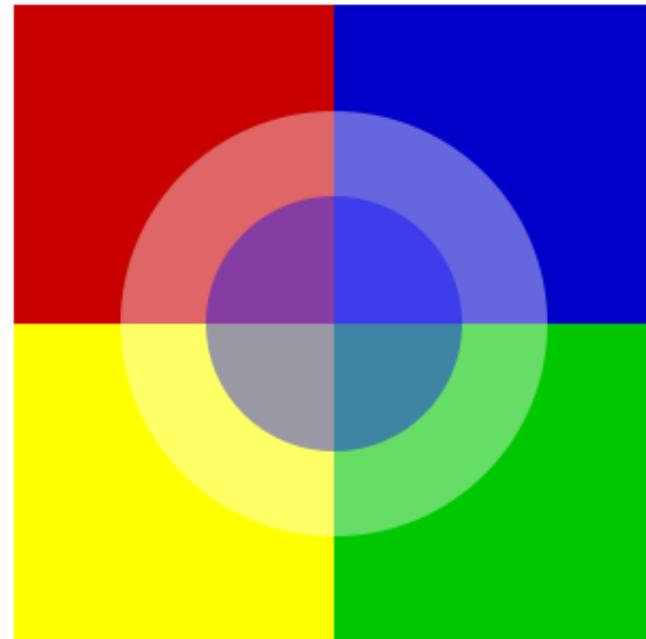
```
var m_canvas = document.getElementById("m");
if(m_canvas.getContext) {
    var m_ctx = m_canvas.getContext("2d");
    m_ctx.fillStyle = "rgb(200,0,0)";
    m_ctx.fillRect (10, 10, 100, 100);
    m_ctx.fillStyle = "#0c3";
    m_ctx.fillRect (50, 50, 100, 100);
    m_ctx.beginPath();
    m_ctx.arc(250,90,80,0,Math.PI*2,true);
    m_ctx.fillStyle="yellow";
    m_ctx.strokeStyle="blue";
    m_ctx.fill();
    m_ctx.stroke();
}
```



## globalAlpha = value

사용할 수 있는 값 : 0.0 ~ 1.0, 기본값 1.0

```
var m_canvas = document.getElementById("m");
if(m_canvas.getContext) {
    var m_ctx = m_canvas.getContext("2d");
    m_ctx.fillStyle = "rgb(200,0,0)";
    m_ctx.fillRect (10, 10, 150, 150);
    m_ctx.fillStyle = "rgb(0,0,200)";
    m_ctx.fillRect (160, 10, 150, 150);
    m_ctx.fillStyle = "rgb(0,200,0)";
    m_ctx.fillRect (160, 160, 150, 150);
    m_ctx.fillStyle = "yellow";
    m_ctx.fillRect (10, 160, 150, 150);
    m_ctx.globalAlpha=0.4;
    m_ctx.beginPath();
    m_ctx.arc(160,160,100,0,Math.PI*2,true);
    m_ctx.fillStyle="white";
    m_ctx.fill();
    m_ctx.beginPath();
    m_ctx.arc(160,160,60,0,Math.PI*2,true);
    m_ctx.fillStyle="blue";
    m_ctx.fill(); }
```



**globalAlpha = value**

사용할 수 있는 값 : 0.0 ~ 1.0, 기본값 1.0

```
var m_canvas = document.getElementById("m");
if(m_canvas.getContext) {
    var m_ctx = m_canvas.getContext("2d");

    m_ctx.fillStyle = "rgba(255,51,0,1.0)";
    m_ctx.fillRect (10, 10, 150, 30);
    m_ctx.fillStyle = "rgba(255,51,0,0.8)";
    m_ctx.fillRect (10, 40, 150, 30);
    m_ctx.fillStyle = "rgba(255,51,0,0.6)";
    m_ctx.fillRect (10, 70, 150, 30);
    m_ctx.fillStyle = "rgba(255,51,0,0.4)";
    m_ctx.fillRect (10, 100, 150, 30);
    m_ctx.fillStyle = "rgba(255,51,0,0.2)";
    m_ctx.fillRect (10, 130, 150, 30);
}
```



① 그라디언트 객체를 만듬

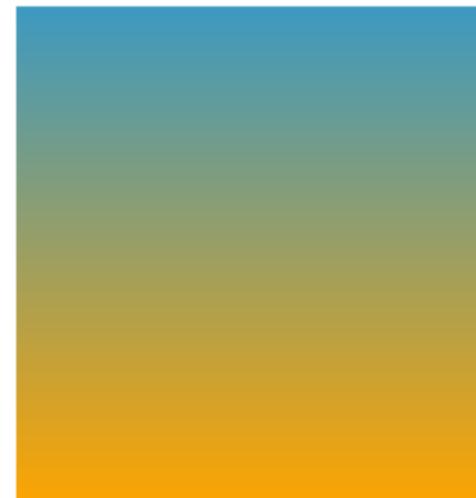
`createLinearGradient(x1, y1, x2, y2)`

`createRadialGradient(x1, y1, r1, x2, y2, r2)`

② 객체에 원하는만큼 색을 할당

`addColorStop(position, color)`

```
if(m_canvas.getContext) {  
    var ctx = m_canvas.getContext("2d");  
    var grad = ctx.createLinearGradient(0,0,0,200);  
    grad.addColorStop(0, '#39c');  
    grad.addColorStop(1, 'orange');  
    ctx.fillStyle=grad;  
    ctx.fillRect(10,10,200,190);  
}
```



# 스타일 적용하기 – 그라디언트 2



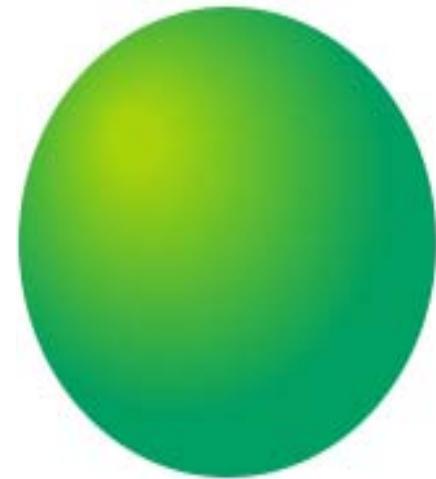
```
if(m_canvas.getContext) {  
    var ctx = m_canvas.getContext("2d");  
    var grad = ctx.createLinearGradient(0,0,0,200);  
  
    grad.addColorStop(0, '#39c');  
    grad.addColorStop(0.5, '#fff');  
    grad.addColorStop(1, 'orange');  
  
    ctx.fillStyle=grad;  
    ctx.fillRect(10,10,200,190);  
}
```



# 스타일 적용하기 – 그라디언트 3



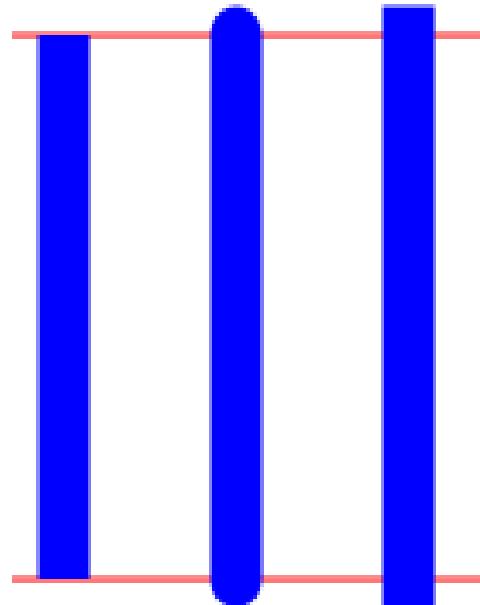
```
if(m_canvas.getContext) {  
    var ctx = m_canvas.getContext("2d");  
    var radgrad = ctx.createRadialGradient(45,45,10,62,60,80);  
  
    radgrad.addColorStop(0, '#A7D30C');  
    radgrad.addColorStop(1, 'rgb(1,159,98)');  
  
    ctx.fillStyle = radgrad;  
    ctx.arc(80,80,80,0,Math.PI*2,true);  
    ctx.fill();  
}
```



- **lineWidth = value** : 선 두께 (사용할 수 있는 값은 양수, 기본값 1.0)
- **lineCap = type** : 선의 끝 모양 (사용할 수 있는 값 butt, round, square, 기본값 butt)
- **lineJoin = type** : 선과 선의 만남 (사용할 수 있는 round, bevel, miter. 기본값 miter)
- **miterLimit = value** : 연결 부분이 잘리는 크기

# 스타일 적용하기 – 선 스타일

```
var lineCap=['butt', 'round', 'square'];
var lineJoin=['round', 'bevel', 'miter'];
ctx.strokeStyle="#f00";
ctx.beginPath();
ctx.moveTo(10,10);
ctx.lineTo(150,10);
ctx.moveTo(10,150);
ctx.lineTo(150,150);
ctx.stroke();
ctx.strokeStyle="blue";
for (var i=0;i<lineCap.length;i++) {
    ctx.lineWidth=15;
    ctx.lineCap=lineCap[i];
    ctx.beginPath();
    ctx.moveTo(25+50*i,10);
    ctx.lineTo(25+50*i,150);
    ctx.stroke();
}
```



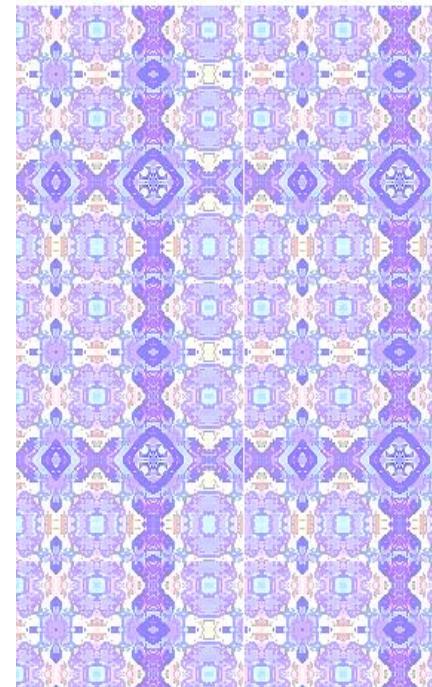
# 스타일 적용하기 – 선 스타일

```
var lineCap=['butt', 'round', 'square'];
var lineJoin=['round', 'bevel', 'miter'];
ctx.strokeStyle="#f00";
ctx.beginPath();
ctx.moveTo(10,10);
ctx.lineTo(150,10);
ctx.moveTo(10,150);
ctx.lineTo(150,150);
ctx.stroke();
ctx.strokeStyle="green";
for (var j=0;j<lineJoin.length;j++) {
    ctx.lineWidth=20;
    ctx.lineJoin = lineJoin[j];
    ctx.beginPath();
    ctx.moveTo(180,60*j+50);
    ctx.lineTo(220,60*j+15);
    ctx.lineTo(260,60*j+50);
    ctx.stroke();
}
```



## createPattern(image, type)

```
if(canvas.getContext){  
    var ctx = canvas.getContext('2d');  
  
    var img = new Image();  
    img.src = 'images/pattern.png';  
    img.onload = function(){  
        var ptn = ctx.createPattern(img,'repeat');  
        ctx.fillStyle = ptn;  
        ctx.fillRect(0,0,300,500);  
    }  
}
```



**shadowOffsetX = float** : 객체로부터 그림자가 얼마나 떨어져 있는가.  
양수값은 오른쪽에 그림자.

**shadowOffsetY = float** : 객체로부터 그림자가 얼마나 떨어져 있는가.  
양수값은 아래에 그림자.

**shadowBlur = float** : 그림자가 얼마나 흐릿한가. 기본값은 0

**shadowColor = color** : 그림자 색상.

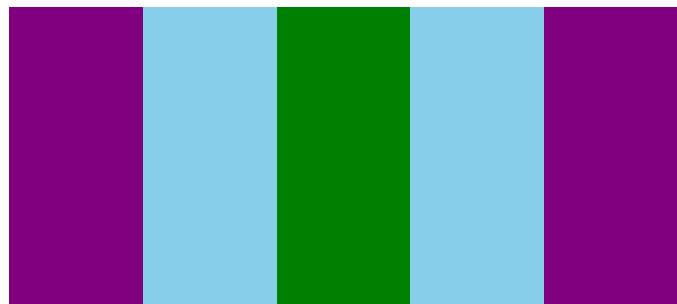
```
if(canvas.getContext){  
    var ctx = canvas.getContext('2d');  
    ctx.shadowOffsetX = 20;  
    ctx.shadowOffsetY = 20;  
    ctx.shadowBlur = 2;  
    ctx.shadowColor="#CCC";  
    ctx.fillRect(10,10,200,200);  
}
```



# 드로잉 상태 저장하기

**save(), restore()**

```
if(canvas.getContext){  
    var ctx = canvas.getContext('2d');  
  
    ctx.fillStyle="purple";  
    ctx.fillRect(10,10,50,100);  
    ctx.save();  
    ctx.fillStyle = "skyblue";  
    ctx.fillRect (60,10,50,100);  
    ctx.save();  
    ctx.fillStyle = "green";  
    ctx.fillRect (110,10,50,100);  
    ctx.restore();  
    ctx.fillRect (160,10,50,100);  
    ctx.restore();  
    ctx.fillRect (210,10,50,100);  
}
```



# 이미지 표시하기

- ① 이미지 객체를 만듬
- ② **drawImage()** 메서드로 객체에 이미지를 그림

```
if(canvas.getContext){  
    var ctx = canvas.getContext('2d');  
    var img = new Image();  
    img.onload=function(){  
        ctx.drawImage(img,10,10)  
    }  
    img.src='images/flower.jpg'
```



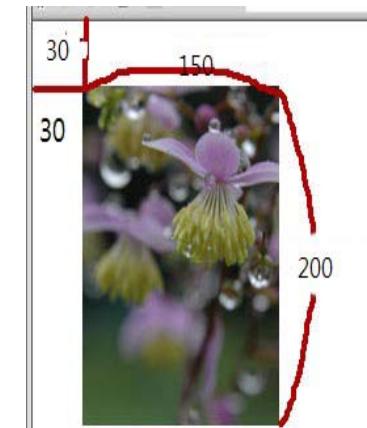
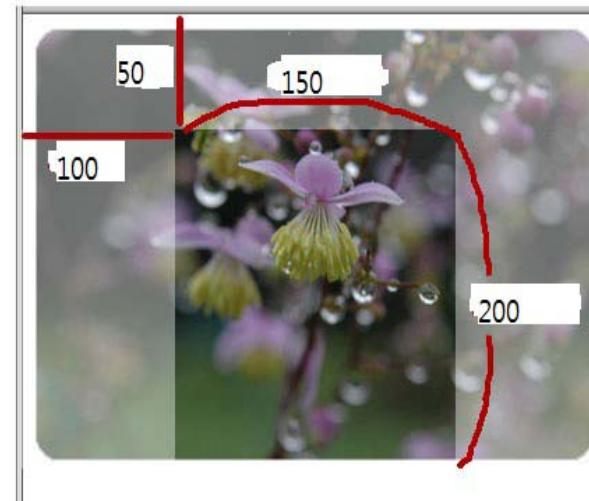
```
var img2=new Image();  
img2.onload=function(){  
    ctx.drawImage(img2,10,320,200,140);  
}  
img2.src='images/flower.jpg';  
}
```



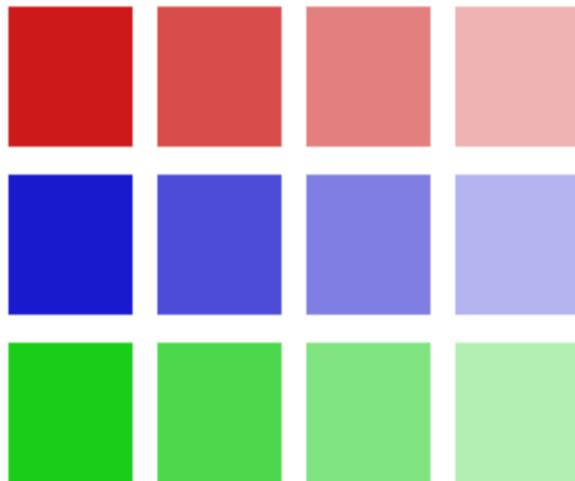
# 이미지 슬라이스하기

**drawImage(image, sx, sy, sWidth, sHeight, dx, dy, dWidth, dHeight)**

```
if(canvas.getContext){  
    var ctx = canvas.getContext('2d');  
    var img = new Image();  
    img.onload=function(){  
        ctx.drawImage(img,100,50,150,200,30,30,150,200);  
    }  
    img.src='images/flower.jpg';  
}
```



## translate(x,y) : 위치 옮기기



```
if(m_canvas.getContext) {  
    var ctx = m_canvas.getContext('2d');  
    for(i=0;i<4;i++){  
        ctx.save();  
        ctx.fillStyle="rgb(200,0,0)";  
        ctx.globalAlpha=0.9-0.2*i;  
        ctx.translate(60*i,0);  
        ctx.fillRect(0,0,50,50);  
        ctx.restore(); }  
    for(i=0;i<4;i++){  
        ctx.save();  
        ctx.fillStyle="rgb(0,0,200)";  
        ctx.globalAlpha=0.9-0.2*i;  
        ctx.translate(60*i,60);  
        ctx.fillRect(0,0,50,50);  
        ctx.restore(); }  
    for(i=0;i<4;i++){  
        ctx.save();  
        ctx.fillStyle="rgb(0,200,0)";  
        ctx.globalAlpha=0.9-0.2*i;  
        ctx.translate(60*i,120);  
        ctx.fillRect(0,0,50,50);  
        ctx.restore(); }
```

**rotate(각도)** : 회전시키기



```
if(m_canvas.getContext) {  
    var ctx = m_canvas.getContext('2d');  
    ctx.translate(200,200);  
    ctx.save();  
    ctx.fillStyle="purple";  
    for (i=0;i<10;i++){  
        ctx.globalAlpha = 1-(0.1*i)  
        ctx.rotate(Math.PI*2/10);  
        ctx.fillRect(10,10,150,50);  
    }  
}
```

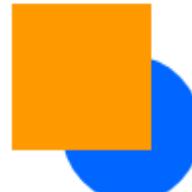
**scale(x, y) :** 확대/축소



```
if(m_canvas.getContext) {  
    var ctx = m_canvas.getContext('2d');  
    ctx.fillStyle="orange";  
    ctx.save();  
    ctx.fillRect(0,0,50,50);  
    ctx.translate(0,60);  
    ctx.scale(1.0, 0.5);  
    ctx.fillRect(0,0,50,50);  
    ctx.restore();  
    ctx.translate(0,100);  
    ctx.scale(2.0, 2.0);  
    ctx.fillRect(0,0,50,50);  
}
```

## globalCompositeOperation 메서드

globalCompositeOperation의 결과 화면은 브라우저마다 조금씩 다릅니다.

					
Source-over	Source-in	Source-out	Source-atop	Destination-over	Destination-in
					이 결과 화면은 파이어폭스 3.6.6 의 결과입니다.
Destination-out	Destination-atop	Lighter	Darker	copy	

## clip()

```
if(canvas.getContext){  
    var ctx = canvas.getContext('2d');  
    var img = new Image();  
    img.onload=function(){  
        ctx.drawImage(img,10,10)  
    }  
    img.src='images/flower.jpg'  
    ctx.beginPath();  
    ctx.arc(200,110,100,0,Math.PI*2,true);  
    ctx.clip();  
}
```



## **5. 웹 품**

# <input>에서 사용할 수 있는 유형들



HTML4에서 사용하는 유형

유형	설명	코드
텍스트	텍스트 입력	<input type="text">
패스워드	비밀번호 입력. 화면에 ●로 표시됨	<input type="password">
체크박스	클릭할 때마다 on/off	<input type="checkbox">
라디오 버튼	그룹 중에서 한가지 선택	<input type="radio">
파일	파일 선택	<input type="file">
전송 버튼	폼에 입력한 값 전송	<input type="submit">

# <input>에서 사용할 수 있는 유형들



HTML5에서 추가된 유형

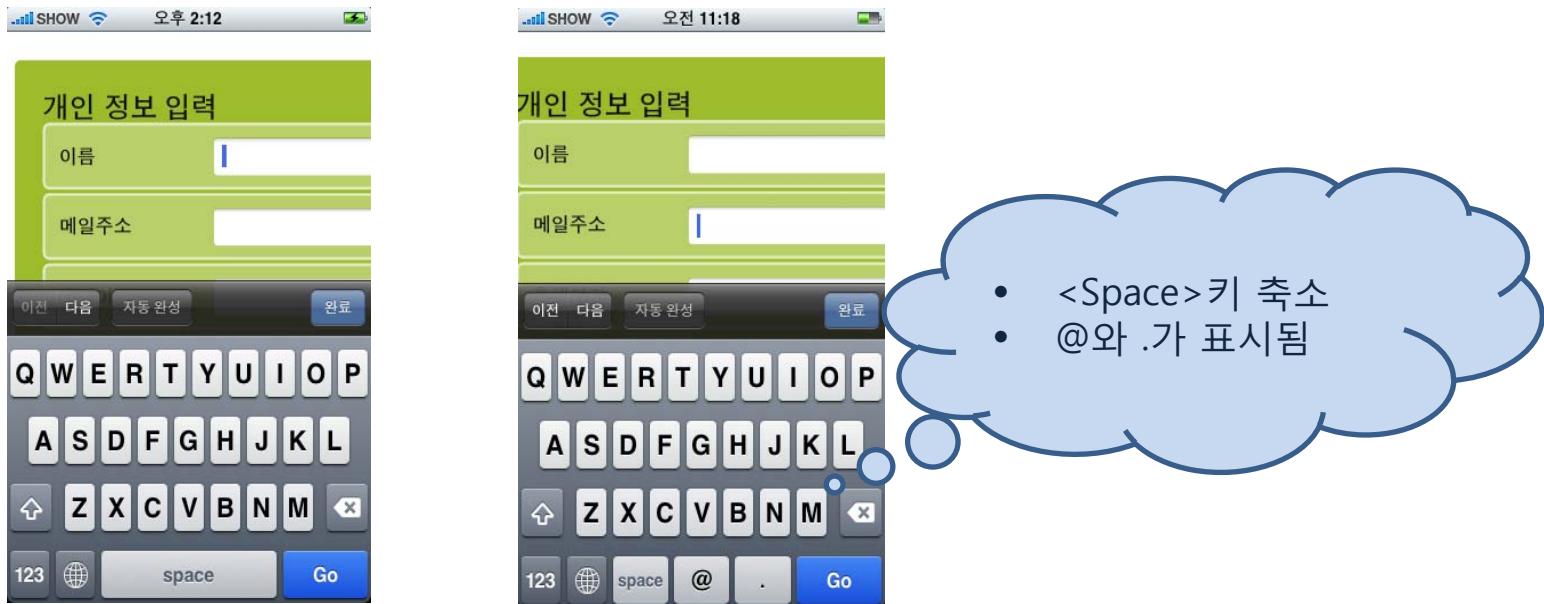
유형	설명	코드
이메일 주소	이메일 주소 입력	<input type="email">
웹 주소	웹 사이트 주소 입력	<input type="url">
숫자(스핀상자)	숫자를 ▲/▼로 눌러 선택	<input type="number">
숫자(슬라이드더)	슬라이드 막대로 지정 범위의 숫자 선택	<input type="range">
날짜/시간 등	달력에서 날짜 선택 스핀 박스에서 시간 선택	<input type="date"> <input type="datetime">
검색 상자	검색 상자 삽입	<input type="search">
색상	웹 상에서 색상 선택	<input type="color">

## **<input> 태그의 새로운 유형들**

현재까지는 <input> 태그의 새로운 유형들을  
일부 브라우저에서만 지원

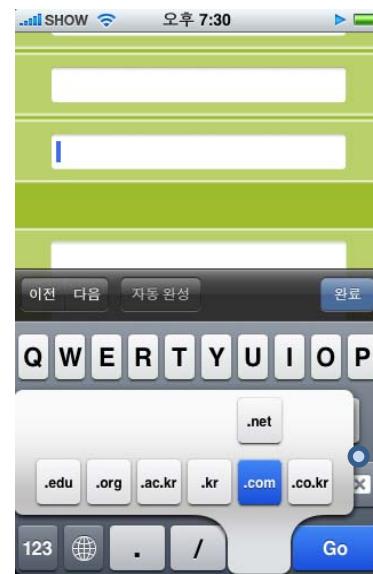
<input type="email">

- 텍스트 필드에 입력하던 것을 이메일 주소 필드로 세분화
- 사용자가 입력하는 단계에서 입력한 값이 이메일 주소인지 아닌지 확인
- '오페라' 브라우저에서 가장 정확하게 지원함



## <input type="url">

- 텍스트 필드에 입력하던 것을 웹 주소 필드로 세분화
- 사용자가 입력하는 단계에서 입력한 값이 이메일 주소인지 아닌지 확인
- '오페라' 브라우저에서 가장 정확하게 지원함



• <Space> 키 없어짐  
• .와 /가 표시됨  
• 도메인 쉽게 선택 가능

# 스핀 박스로 표시하는 숫자 필드



```
<input type="number" [min="최소값" max="최대값"  
step="간격" value="초기값"] >
```

- type="number"만 필수 항목
- step : 짹수나 홀수처럼 특정 숫자로  
입력을 제한하려고 할 때 숫자의 간격을  
지정. 기본값 1

# 스핀 박스로 표시하는 숫자 필드



Opera screenshot showing a form with a numeric input field. The input field has a value of 1. The page title is "file:///I:/HTML5/My%20Samples/Ch5/TMP1If7ol6.htm".

Chrome screenshot showing a form with a numeric input field. The input field has a value of 1. The page title is "file:///I:/HTML5/My%20Samples/Ch5/TMP1If7ol6.htm".

```
<form>
  <fieldset>
    <legend>주문 내용</legend>
    <ol>
      <li>
        <label for="prod">상품명 </label>
        <input id="prod" type="text" size="25">
      </li>
      <li>
        <label for="num">주문개수 </label>
        <input id="num" type="number"
min="1" max="3" value="1">
      </li>
    </ol>
  </fieldset>
  <button type="submit" value="등록"></button>
</form>
```

# 슬라이드 막대로 표시하는 숫자 필드



```
<input type="range" [min="최소값" max="최대값"  
step="간격" value="초기값"] >
```

- type="number"만 필수 항목
- step : 짹수나 홀수처럼 특정 숫자로 입력을 제한하려고 할 때 숫자의 간격을 지정. 기본값 1

# 슬라이드 막대로 표시하는 숫자 필드



크롬

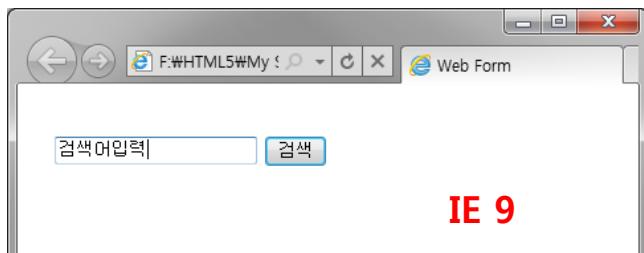
```
<form>
  <fieldset>
    <legend>주문 내용</legend>
    <ol>
      <li>
        <label for="prod">상품명</label>
        <input id="prod" type="text" size="25">
      </li>
      <li>
        <label for="num">주문개수</label>
        <input id="num" type="number" min="1" max="3" value="1">
      </li>
    </ol>
  </fieldset>
  <button type="submit" value="등록"></button>
</form>
```

유형	소스	설명
날짜	<input type="date">	연도와 월, 날짜
월	<input type="month">	연도와 월
주	<input type="week">	연도와 주
시간	<input type="time">	시간
날짜와 시간	<input type="datetime">	날짜 선택, 시간 표시(UTC 시간대)
날짜와 시간 (로컬)	<input type="datetime-local">	날짜 선택, 시간 표시(로컬)

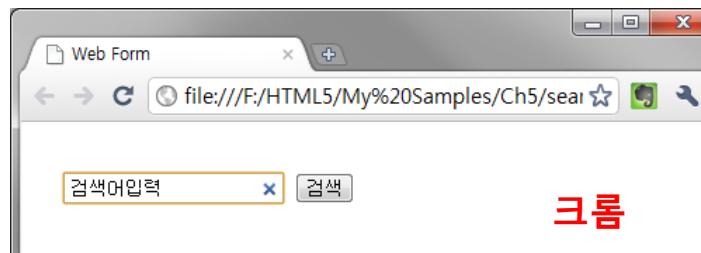
유형	소스	설명
날짜	<input type="date">	원하는 배송일 <input type="text" value="2010-06-16"/> ▾
월	<input type="month">	원하는 배송일 <input type="text" value="2010-06"/> ▾
주	<input type="week">	원하는 배송일 <input type="text" value="2010-W24"/> ▾
시간	<input type="time">	원하는 배송일 <input type="text" value="15:15"/> 
날짜와 시간	<input type="datetime">	원하는 배송일 <input type="text" value="2011-01-30"/> ▾ <input type="text" value="12:00"/>  UTC
날짜와 시간 (로컬)	<input type="datetime-local">	원하는 배송일 <input type="text" value="2011-01-30"/> ▾ <input type="text" value="12:00"/> 

<input type="search">

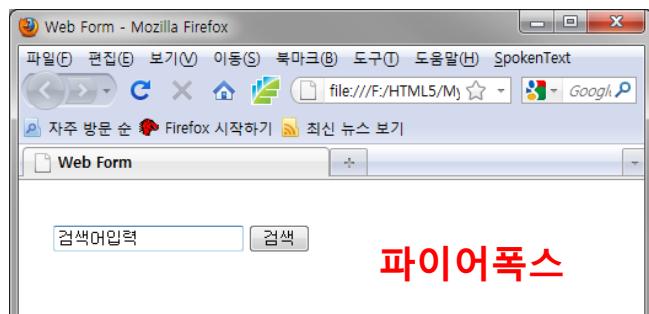
검색 상자를 삽입하기 위한 필드



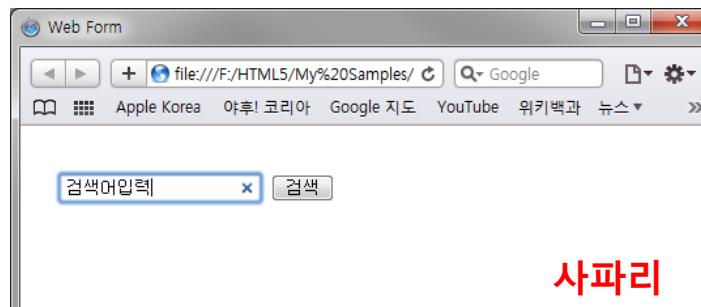
IE 9



크롬



파이어폭스



사파리

<input type="color">

색상을 선택할 수 있는 색상 필드

IE 9/크롬/사파리/파이어폭스

A screenshot of a web browser window showing three input fields:

- 상품명: A text input field.
- 원하는 색상: An input field with a color picker icon.
- 주문개수: A text input field with the value "1".

오페라

A screenshot of the Opera browser showing a color picker dialog box. The dialog box contains:

- 원하는 색상: A color selection area with a preview swatch.
- 주문개수: A text input field with the value "1".
- #000000: A hex color code field.
- 기타...: A button for other options.

## **<input> 태그의 새로운 속성들**

자바스크립트로 처리해야 했던 부분을  
HTML5에서는 간단하게 속성으로 처리할 수 있음

# 자동 완성 제어 - autocomplete

사용할 수 있는 값 : on, off (기본값 : on)

폼에서 개인 정보를 입력할 때 사용자가 (브라우저에서) 자동 완성 기능을 꾼  
지 않더라도, 사이트 문서에서 직접 자동 완성을 끌 수 있음.

**autocomplete="on"**

개인 정보 입력

- 이름
- 메일주소   
f...@naver.com  
l...@naver.com  
m...@naver.com  
n...@naver.com  
w...@naver.com
- 전화번호

**autocomplete="off"**

개인 정보 입력

- 이름
- 메일주소   
f...@naver.com  
l...@naver.com  
m...@naver.com  
n...@naver.com  
w...@naver.com
- 전화번호

# 입력 필드에 커서 표시 – autofocus



페이지를 불러오자 마자 입력 필드 안에 마우스 커서 표시

```
<li>
  <label for="name">이름</label>
  <input id=name type="text" placeholder="8자 미만 공백없이" autofocus>
</li>
<li>
  <label for=email>메일주소</label>
  <input id="email" type="email" placeholder="exa
</li>
```

결과보기

개인 정보 입력

- 이름
- 메일주소
- 전화번호

```
<datalist>
  <option value="서버 값" label="사용자를 위한 레이블">
  <option value="서버 값" label="사용자를 위한 레이블">
  ...
</datalist>
```

```
<li>
  <label for="prodid">색상</label>
  <input id="prod" type="text" list="mycolor">
  <datalist id="mycolor">
    <option value="black" label="검정색">
    <option value="white" label="흰색">
    <option value="skyblue" label="하늘색">
  </datalist>
</li>
```

**결과보기**

- 색상 |
  - 주문 |
- |         |     |
|---------|-----|
| black   | 검정색 |
| white   | 흰색  |
| skyblue | 하늘색 |

# 입력값 제한 – min, max, maxLength, step



min : 필드의 최소값

max : 필드의 최대값

maxLength : 입력할 수 있는 최대 문자 개수

step : 범위 내의 숫자 간격

# 힌트 표시 - placeholder



필드 안에 적당한 힌트 내용을 표시하고 있다가 클릭하면 내용이 사라짐.

```
<li>
  <label for="name">이름</label>
  <input id=name type="text" placeholder="8자 미만 공백없이">
</li>
<li>
  <label for=email>메일주소</label>
  <input id=email type="email" placeholder="example@domain.com">
</li>
<li>
  <label for=phone>전화번호</label>
  <input id=phone name=phone placeholder="02)1111-1111">
</li>
```

결과보기

개인 정보 입력

- 이름
- 메일주소
- 전화번호

# 필수 입력 필드 체크 - required



: submit 가.



required="required" 라고 하거나 그냥 required라고만 해도 됨.  
hidden이나 image, button, submit, reset 유형에서는 사용할 수 없음.

```
<li>
  <label for="name">이름</label>
  <input id=name type="text" placeholder="8자 미만 공백없이" autofocus required>
</li>
<li>
  <label for=email>메일주소</label>
  <input id=email type="email" placeholder="example@domain.com">
</li>
<li>
  <label for="phone">전화번호</label>
  <input id=phone name="phone" placeholder="02)1234-5678">
</li>
```

결과보기

개인 정보 입력

- 이름
- 메일주소
- 전화번호  값을 지정해야 합니다.

# ***6. CSS3***

# CSS3란?

## HTML5

로그인 정보

- 아이디
- 비밀번호
- 비밀번호 확인
- 회원 등급  [준회원]

개인 정보

- 이름  [5자미만 공백없이]
- 메일 주소  [abcd@domain.com]
- 연락처

코딩 숙련도(상,중,하)

- 

**제출**

## CSS3

Step 1: 로그인 정보

아이디

비밀번호

비밀번호 확인

회원 등급  [준회원]

Step 2: 개인 정보

이름  [5자미만 공백없이]

메일 주소  [abcd@domain.com]

연락처

Step 3: 코딩 숙련도(상,중,하)

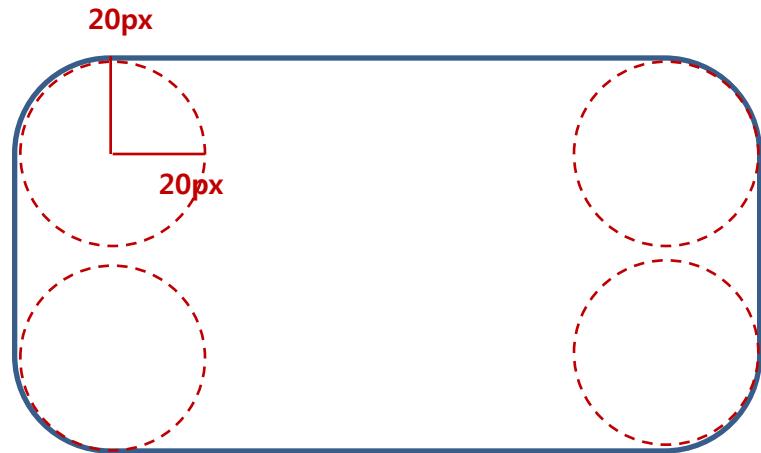
**제출**

아직까지는 브라우저들이 **새로운 CSS3 프로퍼티를 부분적으로만 구현**할 수 있기 때문에 대부분의 CSS3 프로퍼티를 사용하려면 프로퍼티 앞에 **브라우저를 식별할 수 있는 접두사**를 붙여서 사용해야

-webkit-	사파리, 크롬 등
-moz-	모질라, 파이어폭스 등
-o-	오페라
-mx- - ms -	인터넷 익스플로러

# 박스 모델의 테두리 border-radius

```
div.rounded {  
    background-color: #666;  
    color: #fff;  
    line-height: 20px;  
    width: 400px;  
    padding: 10px;  
    -webkit-border-radius: 20px;  
    -moz-border-radius: 20px;  
}
```

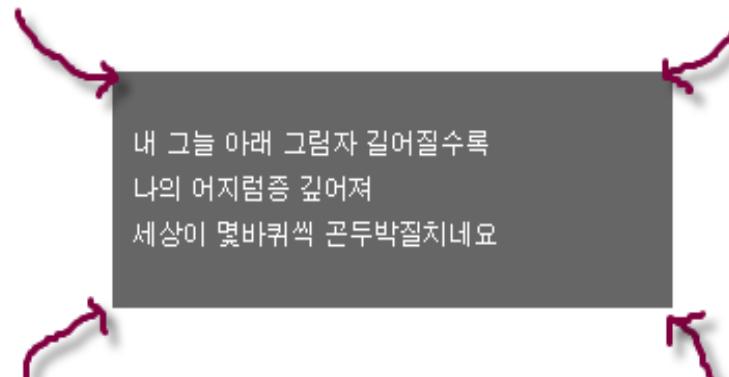


# 박스 모델의 테두리 border-radius



-moz-border-radius-topleft

-webkit-border-top-left-radius



-moz-border-radius-topright

-webkit-border-top-right-radius

-moz-border-radius-bottomleft

-webkit-border-bottom-left-radius

-moz-border-radius-bottomright

-webkit-border-bottom-right-radius

# 박스 모델의 테두리 border-radius

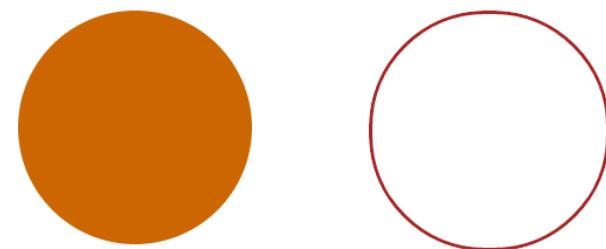


```
div.rounded {  
background-color: #FFFF99;  
line-height: 20px;  
width: 250px;  
padding: 10px;  
border: thin double #C90;  
-webkit-border-bottom-left-radius:15px;  
-moz-border-radius-bottomleft:15px;  
-webkit-border-top-right-radius:15px;  
-moz-border-radius-topright:15px;  
}
```

내 그늘 아래 그림자 길어질수록  
나의 어지럼증 깊어져  
세상이 몇바퀴씩 곤두박질치네요

# border-radius 응용하기 – 원 그리기

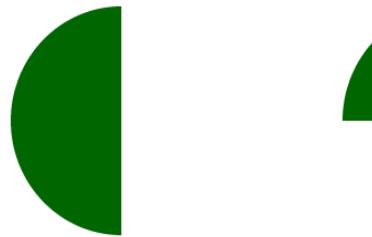
```
<style type="text/css">
.circle1 {
    background-color: #C60;
    width: 200px;
    height: 200px;
    -webkit-border-radius:100px;
    -moz-border-radius:100px;
}
.circle2 {
    background-color: #fff;
    border: 3px #a72525 solid;
    width: 200px;
    height: 200px;
    -webkit-border-radius:100px;
    -moz-border-radius:100px;
}
</style>
```



# border-radius 응용하기 – 반원 그리기

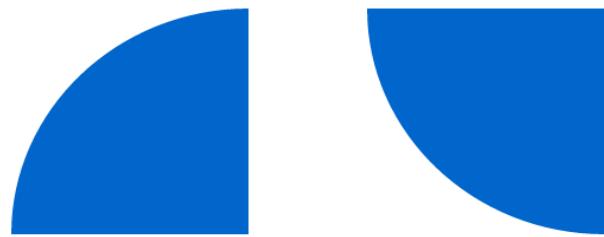


```
<style type="text/css">
.half1 {
    position:absolute;
    left:100px;
    top:50px;
    background-color: #060;
    width: 100px;
    height: 200px;
    -webkit-border-radius:200px 0 0 200px;
    -moz-border-radius:200px 0 0 200px;
}
.half2 {
    position:absolute;
    left:400px;
    top:50px;
    background-color: #060;
    width: 200px;
    height: 100px;
    -webkit-border-radius:200px 200px 0 0;
    -moz-border-radius:200px 200px 0 0;
}
</style>
```



# border-radius 응용하기 – 1/4원 그리기

```
<style type="text/css">
.quater1 {
    background-color: #06C;
    width: 200px;
    height: 200px;
    -webkit-border-radius:200px 0 0 0;
    -moz-border-radius:200px 0 0 0;
}
.quater2 {
    background-color: #06c;
    width: 200px;
    height: 200px;
    -webkit-border-radius:0 0 0 200px;
    -moz-border-radius:0 0 0 200px;
}
</style>
```



```
.grad {  
width:300px;  
padding:10px;  
border:8px solid #000;  
-moz-border-top-colors: #900 #C60 #F60 #F90 #FC0;  
-moz-border-right-colors: #900 #C60 #F60 #F90 #FC0;  
-moz-border-bottom-colors: #900 #C60 #F60 #F90 #FC0;  
-moz-border-left-colors : #900 #C60 #F60 #F90 #FC0;  
}
```

내 그늘 아래 그림자 길어질수록  
나의 어지럼증 깊어져  
세상이 몇바퀴씩 곤두박질치네요



border-image-source : 테두리에 꾸밀 이미지 지정

border-image-slice : 이미지를 9개 영역으로 쪼개기.

top, right, bottom, left를 기준으로 한 옵셋 거리 지정

border-image-repeat : 이미지 배치(반복) 방법 지정

stretch	영역만큼 이미지를 늘림
repeat	영역만큼 타일 형식으로 반복
round	타일 형식으로 반복하되 마지막에 이미지가 잘릴 경우 이미지가 잘리지 않도록 온전하게 채움

border-image : <border-image-source> || <border-images-slice> ||  
<border-image-repeat>

# 테두리에 이미지 꾸미기



```
<style type="text/css">
.mybox {
    border-width:0 12px 0 12px;
    width:300px;
    padding:20px;
    -moz-border-image:url("whiteButton.png") 0 12 0 12 stretch stretch;
    -webkit-border-image:url("whiteButton.png") 0 12 0 12 stretch stretch;
    border-image:url("whiteButton.png") 0 12 0 12 stretch stretch;
}
</style>
</head>
<body>
<div class="mybox"> 이미지를 사용해 테두리 꾸미기, .
</body>
```

<whiteButton.png>



결과보기

이미지를 사용해 테두리 꾸미기



```
<style type="text/css">  
.resize {  
    width:200px;  
    height:50px;  
    padding:5px;  
    border:1px solid #06F;  
    overflow:hidden;  
    resize:both;  
}  
</style>  
</head>  
<body>  
<div class="resize"> .... </div>  
</body>
```

horizontal	요소의 너비만 조절
vertical	요소의 높이만 조절
both	너비와 높이 모두 조절

CSS3 속성 중 resize 속성을 이용하면 요소의 크기를 사용자가 임의대로 조절할 수 있습니다. CSS3 속성 //

CSS3 속성 중 resize 속성을 이용하면 요소의 크기를 사용자가 임의대로 조절할 수 있습니다. CSS3 속성 중 resize 속성을 이용하면 요소의 크기를 사용자가 임의대로 조절할 수 있습니다.

**box-shadow** 속성은 모두 4가지 값을 가지는데 최소 3개의 값을 지정해야 함

가로 오프셋	양수값은 오른쪽, 음수값은 왼쪽에 그림자 생김
세로 오프셋	양수값은 아래쪽, 음수값은 위쪽에 그림자 생김
blur radius	그림자의 번지는 정도. 0이 최소값.
그림자 색상	16진수나 RGB값, 색상 이름 모두 사용 가능

```
<style type="text/css">
a img:hover {
    box-shadow: 0 5px 10px rgba(0,0,0,0.6);
    -moz-box-shadow: 0 5px 10px rgba(0,0,0,0.6);
    -webkit-box-shadow: 0 5px 10px rgba(0,0,0,0.6);
}
</style>
```



# 텍스트에 그림자 효과 추가하기



text-shadow 속성을 사용해서 텍스트 그림자를 표시할 수 있음.  
box-shadow와 마찬가지로 모두 4개의 값을 사용함

```
<style type="text/css">  
.text1 {  
    color:#06F;  
    text-shadow:3px 3px 5px #000;  
}  
.text2{  
    color:#C30;  
    text-shadow:3px -3px 5px #000;  
}  
</style>
```

**HTML5 & CSS3**

**HTML5 & CSS3**

- 배경보다 짙은 글자색과 흰색이나 밝은 회색의 그림자색
- 가로 오프셋은 0, 세로 오프셋은 1px 정도로 작게
- 영문 글자에 어울림

```
<style type="text/css">  
  .text1 {  
    color:#000;  
    text-shadow:0px 1px 1px #fff;  
  }  
  
  .text2{  
    color:#fff;  
    text-shadow:0px 1px 1px #000;  
  }  
  
</style>
```



The image shows two examples of text with CSS3 text shadows. The top example, "HTML5 & CSS3", is in a dark gray box and uses a black text color with a white text shadow. The bottom example, also "HTML5 & CSS3", is in a white box and uses a white text color with a black text shadow.

**HTML5 & CSS3**

**HTML5 & CSS3**

# 텍스트에 그림자 효과 응용



```
<style type="text/css">
body {
    background:#066;
    font-family: Arial Black;
    font-size:50px;
}
.text1 {
    color:#fff;
    text-shadow:0px 1px 8px #fff;
}
.text2 {
    color:#fff;
    text-shadow:1px -3px 8px #6F0;
}
.text3 {
    color:#000;
    text-shadow:1px 1px 4px #fff;
}
</style>
```

**HTML5 & CSS3**

**HTML5 & CSS3**

**HTML5 & CSS3**

```
<style type="text/css">  
body {  
    background:#000;  
    font-family: Arial Black;  
    font-size:50px;  
}  
.text3 {  
    color:#000;  
    text-shadow:0 0 4px #ccc,  
                0 -5px 4px #ff3,  
                2px -10px 6px #fd3,  
               -2px -15px 11px #f80,  
                2px -19px 18px #f20;  
}  
</style>
```



## 1) 웹킷 브라우저(사파리, 크롬)

예) background:-webkit-gradient(linear, left bottom, left top, color-stop(0, #F60), color-stop(1, #F00));



## 2) 파이어폭스

예) background: -moz-linear-gradient(top, #f60, #f00);



# 선형 그라데이션

```
<style type="text/css">  
.grad1{  
    position:relative;  
    width:150px;  
    height:150px;  
    background: #06f;  
    background: -webkit-gradient(linear, left top, left bottom, frc  
background: -moz-linear-gradient(top,#06f, #fff);  
}  
.grad2{  
    position:relative;  
    width:150px;  
    height:150px;  
    background: #06f;  
    background: -webkit-gradient(linear, left bottom, right top, f  
background: -moz-linear-gradient(bottom 45deg, #06f, #fff);  
}  
</style>
```

색상만 지정했을 때



결과보기

방향을 지정했을 때



## 1) 웹킷 브라우저(사파리, 크롬)

예) background-image: -webkit-gradient(radial, center center, 10, center center, 90, from(#F60), to(#F00));



## 2) 파이어폭스

예) background-image: -moz-radial-gradient(center 45deg, circle closest-corner, #F60 10%, #F00 70%);

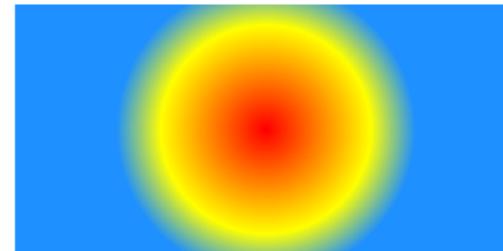


# 원형 그라데이션

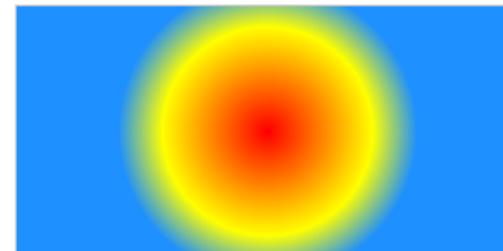


```
<style type="text/css">  
.grad1{  
    background: -webkit-gradient(radial, center center, 10, center center, 90, from(red),  
to(#1E90FF), color-stop(70%, yellow));  
    background: -moz-radial-gradient(circle, red, yellow, #1E90FF);  
}  
</style>
```

원형 그라데이션



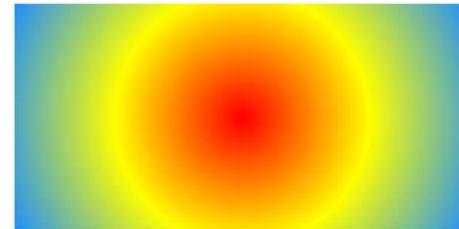
(파이어폭스의 경우) 타원, closest-side



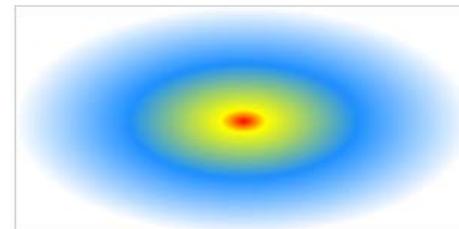
# 원형 그라데이션

```
<style type="text/css">  
.grad2{  
background: -webkit-gradient(radial, center center, 10, center center, 90, from(red),  
to(#1E90FF), color-stop(70%, yellow));  
background:-moz-radial-gradient(ellipse closest-side, red, yellow 10%, #1E90FF  
50%, white);  
}  
</style>
```

원형 그라데이션



(파이어폭스의 경우) 타원, closest-side



# 배경 이미지 크기를 자유롭게

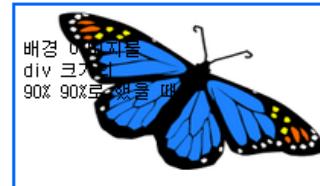
- background-size 속성 사용
- 이미지 크기를 백분율이나 픽셀 크기로 지정
- 백분율로 지정할 때는 배경 이미지가 적용되는 요소의 크기를 기준으로 함.

```
<style type="text/css">
#bg1{
    background:url(bg.png);
    background-size:90% 90%;
    -moz-background-size:90% 90%;
    -webkit-background-size:90% 90%;
    ...
}
#bg2{
    background:url(bg.png);
    background-size:100px 80px;
    -moz-background-size:100px 80px;
    -webkit-background-size:100px 80px;
    ...
}
</style>

<div id="bg1">
    배경 이미지를 div 크기의 90% 90%로 했을 때 </div>
<div id="bg2"> 배경 이미지를 150px 120px 로 했을 때
    </div>
```



배경 이미지 원본



배경 이미지를  
div 크기의  
90% 90%로 했을 때



배경 이미지를  
150px 120px 로 했을 때

## CSS3 이전

```
<style type="text/css">  
    .noopa {  
        background-color:#fff;  
        color:#000;  
    }  
    .opa {  
        background-color:#fff;  
        opacity:0.5;  
        color:#000;  
    }  
</style>
```

HTML5



HTML5



## CSS3의 rgba 이용

```
<style type="text/css">  
  .noopa {  
    background-color:#fff;  
    color:#000;  
  }  
  .opa1 {  
    background-color:rgba(255,255,255,0.5);  
    color:rgb(0,0,0);  
  }  
  .opa2 {  
    background-color:rgb(255,255,255);  
    color:rgba(0,0,0,0.5);  
  }  
</style>
```



## 회전

rotate(각도)

지정한 각도만큼 회전시킴. 각도값은 도(degree), 또는 라디안 값

```
<style type="text/css">  
.myphoto{  
    transform: rotate(15deg);  
    -webkit-transform: rotate(15deg);  
    -moz-transform: rotate(15deg);  
}  
</style>
```



## 확대/축소

scale(sx, sy)	가로로 sx만큼, 세로로 sy만큼 확대
scaleX(sx)	가로로 sx만큼 확대
scaleY(sy)	세로로 sy만큼 확대

```
<style type="text/css">  
.myphoto{  
    transform: scale(1.5, 1.5);  
    -webkit-transform: scale(1.5, 1.5);  
    -moz-transform: scale(1.5, 1.5);  
}  
</style>
```



## 왜곡

skew(각도, 각도)	첫번째 각도는 x 축상의 왜곡 각도, 두번째 각도는 y축 상의 왜곡 각도.
skewX(각도)	x 축을 따라 주어진 각도만큼 왜곡시킴
skewY(각도)	y 축을 따라 주어진 각도만큼 왜곡시킴

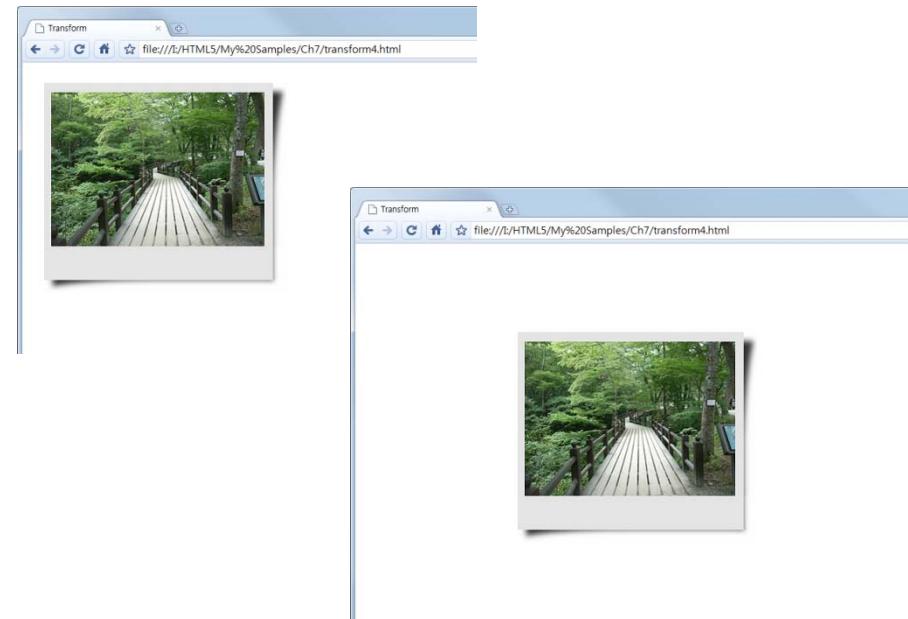
```
<style type="text/css">  
.myphoto{  
    transform: skew(10deg, 10deg);  
    -webkit-transform: skew(10deg, 10deg);  
    -moz-transform: skew(10deg, 10deg);  
}  
</style>
```



## 이동

translate(tx,ty)	x 축으로 tx만큼, y 축으로 ty만큼 이동
translateX(tx)	x 축으로 tx만큼 이동
translateY(ty)	y 축으로 ty만큼 이동

```
<style type="text/css">  
 .myphoto{  
 position:absolute;  
 left:30px;  
 top:30px;  
 transform: translate(200px, 100px);  
 -webkit-transform:translate(200px, 100px);  
 -moz-transform:translate(200px, 100px);  
 }  
</style>
```



# 트랜지션 - transition



-webkit-transition-property	애니메이션을 적용할 요소의 속성 지정
-webkit-transition-duration	트랜지션이 끝날 때까지 걸리는 시간 (sec 단위)
-webkit-transition-timing-function	트랜지션 형태 선택 (ease, linear, ease-in, ease-out, ease-in-out, cubic-beizer)
-webkit-transition	위의 세 가지 속성을 하나의 속성으로 줄여서 씀.

# 트랜지션 - transition



```
<style type="text/css">
a {
    background: #fff;
    color: #aa0000;
    -webkit-transition: all 1s linear;
}
a:hover {
    background: #220077;
    color: #fff;
}
</style>
</head>
<body>
<ul>
<li><a href="#classic">고전주의</a></li>
<li><a href="#romantic">낭만주의</a></li>
<li><a href="#realitic">사실주의</a></li>
<li><a href="#symbolic">상징주의</a></li>
</ul>
...
</body>
```

고전주의

낭만주의

사실주의

상징주의

고전주의

낭만주의

사실주의

상징주의

고전주의

낭만주의

사실주의

상징주의

메뉴 위로 마우스 포인터를 올려놓으면 메뉴 배경색과 글자색이 서서히 바뀜.  
마우스 포인터를 치우면 서서히 원래 스타일로 되돌아 옴.

# 새로운 CSS3 셀렉터

항목[속성^="값"]	"값"으로 시작하는 속성이 있는 항목 선택
항목[속성\$="값"]	"값"으로 끝나는 속성이 있는 항목 선택
항목[속성*= "값"]	"값"이 포함되는 속성이 있는 항목 선택

```
a[href$=".pdf"] {  
    padding-right:13px;  
    background:#FFF url(icon_pdf.png) center right no-repeat;  
}  
  
a[href$=".hwp"] {  
    padding-right:13px;  
    background:#FFF url(icon_hwp.png) center right no-repeat;  
}  
  
a[href$=".zip"] {  
    padding-right:13px;  
    background:#FFF url(icon_zip.png) center right no-repeat;  
}
```

# 슈도 클래스 :not, :target



:not	지정한 항목을 제외한 모든 항목 선택
:target	타켓이 되는 부분에 원하는 표시를 할 수 있음

```
*:not(a) { color : yellow; }
```

링크를 제외한 모든 항목에 적용

```
<style type="text/css">
p.target{
border : 2px solid #900;
background:#FFC;
border-radius:5px;
-webkit-border-radius:5px;
-moz-border-radius:5px;
padding:5px 8px;
}
</style>
```

```
<ul>
<li><a href="#classic">고전주의</a></li>
<li><a href="#romantic">낭만주의</a></li>
<li><a href="#realitic">사실주의</a></li>
<li><a href="#symbolic">상징주의</a></li>
</ul>
<p id="classic">고전주의는 ...</p>
<p id="romantic">낭만주의는 ...</p>
<p id="realitic">19세기 후반부터는 ...</p>
<p id="symbolic">상징주의는 ...</p>
```

The screenshot displays two browser windows side-by-side. Both windows show the same HTML content: a list of literary movements and their descriptions. The top window, titled 'Pseudo-class target', shows the first link ('고전주의') in yellow, while the others are black. The bottom window, also titled 'Pseudo-class target', shows all links in black, indicating they are not selected by the ':not(a)' selector.

## CSS2부터 사용

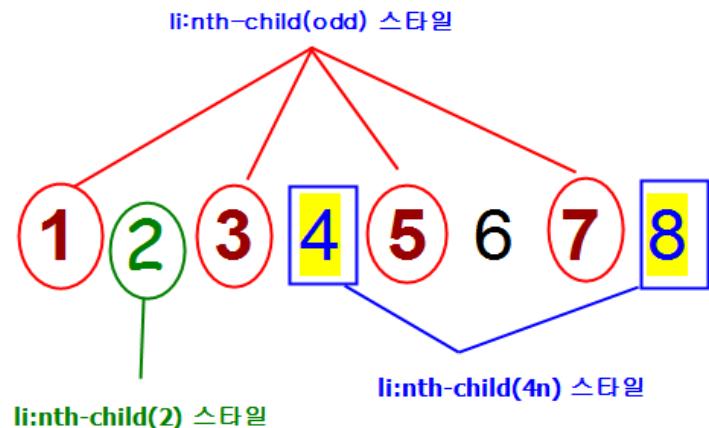
:root	문서의 루트 선택(웹 문서의 경우 <html> 선택)
:last-child	컨테이너 항목의 마지막 자식 선택
:first-of-type	해당 유형의 첫번째 항목 선택
:last-of-type	해당 유형의 마지막 항목 선택
:only-child	컨테이너 항목의 유일한 자식 항목 선택
:only-of-type	해당 유형을 가진 컨테이너 항목의 유일한 자식 항목 선택
:empty	빈 항목 선택

## CSS3부터 사용

:nth-child(n)	항목 중에서 n번째 자식 선택
:nth-last-child(n)	끝에서부터 세어 n번째 자식 선택
:nth-of-type(n)	유형이 같이 n번째 자식 선택
:nth-last-of-type(n)	끝에서부터 세어 유형이 같은 n번째 항목 선택

# 구조적 슈도 클래스

```
<style type="text/css">
li {
    list-style-type:none;
    margin-right:30px;
    float:left;
    font-size:48px;
    font-family:Arial, Helvetica, sans-serif;
}
li:nth-child(odd) {
    color : #900;
    font-weight:bold;
}
li:nth-child(2) {
    color:green;
    font-family: Comic Sans Ms ;
}
li:nth-child(4n) {
    color:blue;
    background-color:yellow;
}
</style>
```



# RGB와 RGBA

```
<head>
<style type="text/css">
    div { width:300px;}
    div.c1 {background:rgb(150,20,20); height:30px;}
    div.c2 {background:rgb(250,200,25); height:30px;}
    div.c3 {background:rgb(25,50,200); height:30px;}
</style>
</head>
<body>
    <div class="c1"></div>
    <div class="c2"></div>
    <div class="c3"></div>
</body>
```



# RGB와 RGBA

```
<head>
<style type="text/css">
div {width:300px;}
div.a1 {background:rgba(225,70,10,1.0); height:30px;}
div.a2 {background:rgba(225,70,10,0.8); height:30px;}
div.a3 {background:rgba(225,70,10,0.6); height:30px;}
div.a4 {background:rgba(225,70,10,0.4); height:30px;}
div.a5 {background:rgba(225,70,100,0.2); height:30px;}
</style>
</head>
<body>
<div class="a1"></div>
<div class="a2"></div>
<div class="a3"></div>
<div class="a4"></div>
<div class="a5"></div>
</body>
```



- **Hue(색상)** : 색의 3속성 중 색상. 색상환의 각도를 기준으로 색상 표현. 0과 360은 빨강을, 120 정도면 초록을, 240 정도면 파랑을 나타내며 그 사이의 값을 이용해 그 외의 색들을 표시.
- **Saturation(채도)** : 색의 3속성 중 채도를 나타내며 %로 표시. 아무 것도 섞지 않은 상태를 채도가 높다고 하며 0%면 회색톤으로, 100%면 순색으로 표시됨.
- **Lightness(밝기)** : %로 표시하며 0은 어둡고 50%는 중간 정도, 100%는 아주 밝음.

# HSL과 HSLA

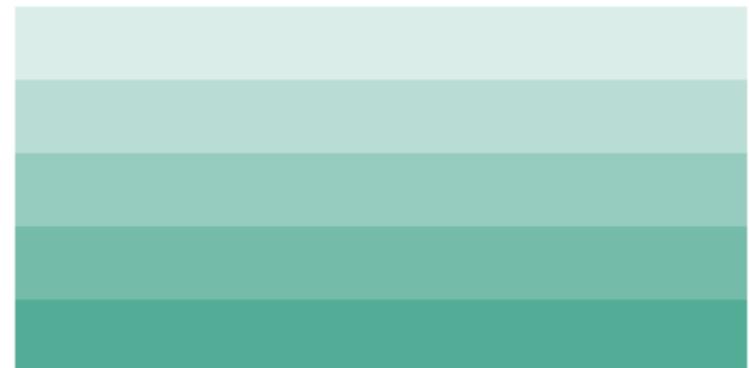
```
<head>
<style type="text/css">
    div { width:300px; }

    div.c1 {background:hsl(320, 100%, 25%); height:30px;}
    div.c2 {background:hsl(320, 100%, 50%); height:30px;}
    div.c3 {background:hsl(320, 100%, 75%); height:30px;}
</style>
</head>
<body>
    <div class="c1"></div>
    <div class="c2"></div>
    <div class="c3"></div>
</body>
```



# HSL과 HSLA

```
<head>  
  
<style type="text/css">  
  
div { width:300px; }  
  
div.h1 {background:hsla(165,35%,50%,0.2); height:30px;}  
div.h2 {background:hsla(165,35%,50%,0.4); height:30px;}  
div.h3 {background:hsla(165,35%,50%,0.6); height:30px;}  
div.h4 {background:hsla(165,35%,50%,0.8); height:30px;}  
div.h5 {background:hsla(165,35%,50%,1.0); height:30px;}  
  
</style>  
</head>  
  
<body>  
  
  <div class="h1"></div>  
  <div class="h2"></div>  
  <div class="h3"></div>  
  <div class="h4"></div>  
  <div class="h5"></div>  
  
</body>
```



```
@font-face {  
    font-family: 글꼴 이름;  
    src :url(글꼴 파일 경로) format(파일 유형);  
}
```

```
<style>  
@font-face {  
    font-family:Blackout, Arial Black;  
    src:url('Blackout_2am.ttf') format('truetype');  
}  
.normal_text { /* 따로 글꼴 지정하지 않았을 때 */  
    font-size:80px;  
    color:#900;  
}  
.text1{  
    font-family:Blackout;  
    font-size:80px;  
    color: #006;  
}  
</style>  
  
<body>  
    <div class="normal_text">ABCDEF</div>  
    <div class="text1">ABCDEF</div>  
</body>
```

ABCDEF

ABCDEF



column-count	단의 개수
column-width	단의 너비
column-gap	단 사이 간격
column-rule	단 사이 테두리

```
.mcol{
    -moz-column-count:3;
    -moz-column-gap:20px;
    -moz-column-rule:1px solid #ddccb5;
    -webkit-column-count:3;
    -webkit-column-gap:20px;
    -webkit-column-rule:1px solid #ddccb5;
}
```

Untitled Document file:///I:/HTML5/My%20Samples/Ch7/multi\_columns.html

**허브란?**

라틴어의 “푸른 풀”을 의미하는 Herba에서 출발하며, “잎, 줄기와 뿌리 등이 식용, 약용에 TM이거나 향기나 향미가 이용되는 식물의 총체”라고 할 수 있다. 즉, 잎, 줄기, 뿌리, 꽃 등을 허브의 의미에 포함하며 그 성분이 식품이나 음료속에 보존용 항신료 태는 건강증진제로서 첨가되는 식물과 식품, 음료 외에 제품에 향수, 헤장, 세정의 효과를 기대하여 이용되는 식물의 전부라고 할 수 있다. 현재 국내에서 일반적으로 알고 있는 허브에 대한 개념은 외국에서 도입된 식물에 한정되어 생각하는 경우가 대부분이나 사실은 이미 수천년 전부터 우리 조상들도 생활 전반에 걸쳐 많은 곳에서 이용하여 왔다. 즉, 산마초를, 예를들면 쑥, 냉이, 씹바퀴, 꼴취 등을 불황식단에 반찬으로 이용된 것

과 한방 처방에 들어가 있는 모든 식물들도 넓게는 모두 허브의 범주에 넣을 수 있는 것이다.

- 허브는 약용식물이다
- 허브는 향기식물이다
- 허브는 채소이다
- 허브는 향신료이다
- 허브는 미인을 만든다

**허브의 종류**

- 바질(Basil) : 두통, 신경과민, 구내염, 강장효과, 건위, 진정, 설운, 불면증과 열을 잘 나오게 하는 효능이 있고, 폴립을 방지하여 늦게까지 공부하는 수험생에게 좋다. 또한 신장의 활동을 촉진시키며 벌레 물린데의 살균효과가 있다.
- 개모마일(Chamomile) : 목욕제로 쓰면 심신의 긴장을 풀어주며 전신 미용에 효과가 크다. 불면증에도

**허브 향기 가득한 곳**

- [포천 허브아일랜드](#)
- [대구 허브힐즈](#)
- [평창 허브나라](#)
- [원영 허브랜드](#)
- [연천 허브빌리지](#)

## 7. CSS3 활용

## **7.1 서브 메뉴가 펼쳐지는 내비게이션 만들기**

```
<ul class="navi">
<li><a href="#"> Home </a></li>
<li><a href="#"> 웹 개발</a>
<ul>
  <li><a href="#"> 웹 디자인 </a></li>
  <li><a href="#"> HTML5 + CSS3</a></li>
  <li><a href="#"> JQuery</a></li>
  <li><a href="#"> 웹 + 모바일</a></li>
</ul>
</li>
<li><a href="#"> 게시판 </a>
<ul>
  <li><a href="#"> Q & A </a></li>
  <li><a href="#"> 방명록 </a></li>
</ul>
</li>
<li><a href="#"> 프로필 </a></li>
</ul>
```

- [Home](#)
- [웹 개발](#)
  - [웹 디자인](#)
  - [HTML5 + CSS3](#)
  - [JQuery](#)
  - [웹 + 모바일](#)
- [게시판](#)
  - [Q & A](#)
  - [방명록](#)
- [프로필](#)

# CSS 적용

```
<style type="text/css">
.navi {
    list-style-type: none;
    height: 40px;
    padding: 0;
    margin: 0;
}

.navi li {
    float: left;
    position: relative;
    padding: 0;
    line-height: 40px;
    background:#600;
}

.navi li a {
    display: block;
    font-family:"굴림";
    font-weight:600;
    font-size:14px;
    padding: 5px 25px;
    color: #fff;
    text-decoration: none;
}
```

```
.navi li a:hover {
    color:#F90;
}

.navi li ul {
    opacity: 0;
    position: absolute;
    left: 0;
    width: 180px;
    background: #D8AFAF;
    list-style-type: none;
    padding: 0;
    margin: 0;
}

.navi li:hover ul {
    opacity: 1;
}

.navi li ul li {
    float: none;
    position: static;
    height: 0;
    line-height: 0;
    background: none;
}
```



# CSS 적용



```
.navi li:hover ul li {  
    height: 30px;  
    line-height: 30px;  
    padding: 5px 0;  
}  
  
.navi li ul li a {  
    background: #D8AFAF;  
    font-family: "굴림";  
    font-weight: 600;  
    font-size: 14px;  
    color: #900;  
}  
  
.navi li ul li a:hover {  
    background: #900;  
}  
  
</style>
```

A screenshot of a website's header navigation bar. The bar has a dark red background and white text. It includes links for Home, 웹 개발 (Web Development), 게시판 (Board), and 프로필 (Profile). Below the main menu, there is a sidebar with a light red background containing links for 웹 디자인 (Web Design), HTML5 + CSS3, JQuery, and 웹 + 모바일 (Web + Mobile).

# 트랜지션 효과 추가



```
.navi li:hover ul li {  
    height: 30px;  
    line-height: 30px;  
    padding: 5px 0;  
}  
.navi li ul li a {  
    background: #D8AFAF;  
    font-family: "굴림";  
    font-weight: 600;  
    font-size: 14px;  
    color: #900;  
}  
.navi li ul li a:hover {  
    background: #900;  
}  
  
.navi li { -webkit-transition: all 0.2s; }  
.navi li a { -webkit-transition: all 0.5s; }  
.navi li ul { -webkit-transition: all 1s; }  
.navi li ul li { -webkit-transition: height 0.5s; }  
  
</style>
```



크롬으로 확인하기.  
트랜지션 효과로 인해 서브 메뉴가 부드럽게 펼쳐짐.

## **7.2 가로 템 메뉴 만들기**

# 기본 HTML 소스



```
<div id="menu">
<ul class="navi">
    <li><a href="#"> Home </a></li>
    <li><a href="#"> 웹 개발</a>
        <ul>
            <li><a href="#"> 웹 디자인
        </a></li>
            <li><a href="#"> HTML5 +
CSS3</a></li>
                <li><a href="#"> JQuery</a></li>
                <li><a href="#"> 웹 + 모바일</a></li>
        </ul>
    </li>
    <li><a href="#"> 게시판 </a>
        <ul>
            <li><a href="#"> Q & A </a></li>
            <li><a href="#"> 방명록</a></li>
        </ul>
    </li>
    <li><a href="#"> 프로필 </a></li>
</ul>
</div>
```

- [Home](#)
- [웹 개발](#)
  - [웹 디자인](#)
  - [HTML5 + CSS3](#)
  - [JQuery](#)
  - [웹 + 모바일](#)
- [게시판](#)
  - [Q & A](#)
  - [방명록](#)
- [프로필](#)

# CSS 적용

```
<style type="text/css">
#menu {
    width: 960px;
    height: 40px;
    clear: both;
}

ul.navi {
    float: left;
    width: 960px;
    margin: 0;
    padding: 0;
    list-style: none;
    background: #39F;
    -moz-border-radius-topright: 10px;
    -webkit-border-top-right-radius: 10px;
    -moz-border-radius-topleft: 10px;
    -webkit-border-top-left-radius: 10px;
}
```

```
ul.navi li {
    display: inline;
}

ul.navi li a {
    float: left;
    font: bold 14px 굴림;
    line-height: 40px;
    color: #fff;
    text-decoration: none;
    text-shadow: 1px 1px 1px #880000;
    margin: 0;
    padding: 0 30px;
    background: #39F;
    -moz-border-radius-topright: 10px;
    -webkit-border-top-right-radius: 10px;
    -moz-border-radius-topleft: 10px;
    -webkit-border-top-left-radius: 10px;
}
```



# CSS 적용



```
ul.navi .current a, ul.navi li:hover > a {  
    color: #fff;  
    text-decoration: none;  
    text-shadow: 1px 1px 1px #330000;  
    background: #36F;  
    -moz-border-radius-topright: 10px;  
    -webkit-border-top-right-radius: 10px;  
    -moz-border-radius-topleft: 10px;  
    -webkit-border-top-left-radius: 10px;  
}  
  
ul.navi ul { display: none; }  
  
ul.navi li:hover > ul {  
    position: absolute;  
    display: block;  
    width: 920px; height: 45px;  
    margin: 40px 0 0 0; background: #33F;  
    -moz-border-radius-bottomright: 10px;  
    -webkit-border-bottom-right-radius: 10px;  
    -moz-border-radius-bottomleft: 10px;  
    -webkit-border-bottom-left-radius: 10px;  
}
```

```
ul.navi li:hover > ul li a {  
    float: left;  
    font: bold 14px 굴림;  
    line-height: 45px;  
    color: #fff;  
    text-decoration: none;  
    text-shadow: 1px 1px 1px #110000;  
    margin: 0;  
    padding: 0 30px 0 0;  
    background: #33F;  
}  
  
ul.navi li:hover > ul li a:hover {  
    color: #120000;  
    text-decoration: none;  
    text-shadow: none;  
}
```



## **7.3 그라데이션을 이용한 메뉴 만들기**

# 그라데이션을 사용한 메뉴의 예 - 애플닷컴



# 기본 HTML 소스



```
<div>
  <ul class="navi">
    <li class="home"><a href="#">Home</a></li>
    <li id="html"><a href="#">HTML5</a></li>
    <li id="css"><a href="#">CSS3</a></li>
    <li id="jquery"><a href="#">JQuery</a></li>
    <li id="mobile"><a href="#">Mobile</a></li>
    <li id="profile"><a href="#">Profile</a></li>
  </ul>
</div>
```

- [Home](#)
- [HTML5](#)
- [CSS3](#)
- [JQuery](#)
- [Mobile](#)
- [Profile](#)

```
ul.navi li {  
    list-style-type: none;  
    display: block;  
    width: 8em;  
    height: 3em;  
    float: left;  
    text-align: center;  
    font-family: "Lucida Grande", sans-serif;  
    border-right: 1px #818181 solid;  
    -webkit-box-shadow: 1px 1px 1px #bbb;  
    -moz-box-shadow: 1px 1px 1px #bbb;  
    box-shadow: 1px 1px 1px #bbb;  
    background-color: #c9c9c9;  
    background: -webkit-gradient(linear, left top, left bottom, from(#c9c9c9), to(#848484), color-stop(0.6, #a1a1a1));  
    background: -moz-linear-gradient(top, #c9c9c9, #a1a1a1, #848484);  
}
```

```
ul.navi li a {  
    display: block;  
    padding: .8em .5em .5em .5em;  
    text-decoration: none;  
    color: #292929;  
    text-shadow: 1px 1px 0px #cccccc;  
}  
  
ul.navi li:hover{  
    background-color: #333333;  
    background: -webkit-gradient(linear, left top, left bottom, from(#333333), to(#4c4c4c), color-  
stop(0.6, #474747));  
    background: -moz-linear-gradient(top, #333333, #474747, #4c4c4c);  
}  
  
ul.navi li:hover > a{  
    color:#FFF; /* 글자색 */  
    text-shadow:0px 1px 0px #fff; /* 텍스트 그림자 */  
}
```

결과보기



# CSS 적용 – 메뉴의 양쪽 끝 둥글게 처리



```
/* 첫번째 항목의 스타일 */
ul.navi li:first-child {
    -webkit-border-top-left-radius: .5em; /* 왼쪽 상단 코너를 부드럽게 */
    -moz-border-radius-topleft: .5em;
    border-top-left-radius: .5em;
    -webkit-border-bottom-left-radius: .5em; /* 왼쪽 하단 코너를 부드럽게 */
    -moz-border-radius-bottomleft: .5em;
    border-bottom-left-radius: .5em;
}

/* 마지막 항목의 스타일 */
ul.navi li:last-child {
    -webkit-border-top-right-radius: .5em; /* 오른쪽 상단 코너를 부드럽게 */
    -moz-border-radius-topright: .5em;
    border-top-right-radius: .5em;
}
```

결과보기



## **7.4 사진 갤러리 만들기**

# 기본 HTML 소스



```
<div id="gallery">
  <ul>
    <li> 
       </li>
    <li> 
       </li>
    <li> 
       </li>
    <li> 
       </li>
    <li> 
       </li>
    <li> 
       </li>
  </ul>
</div>
```



```
<style>

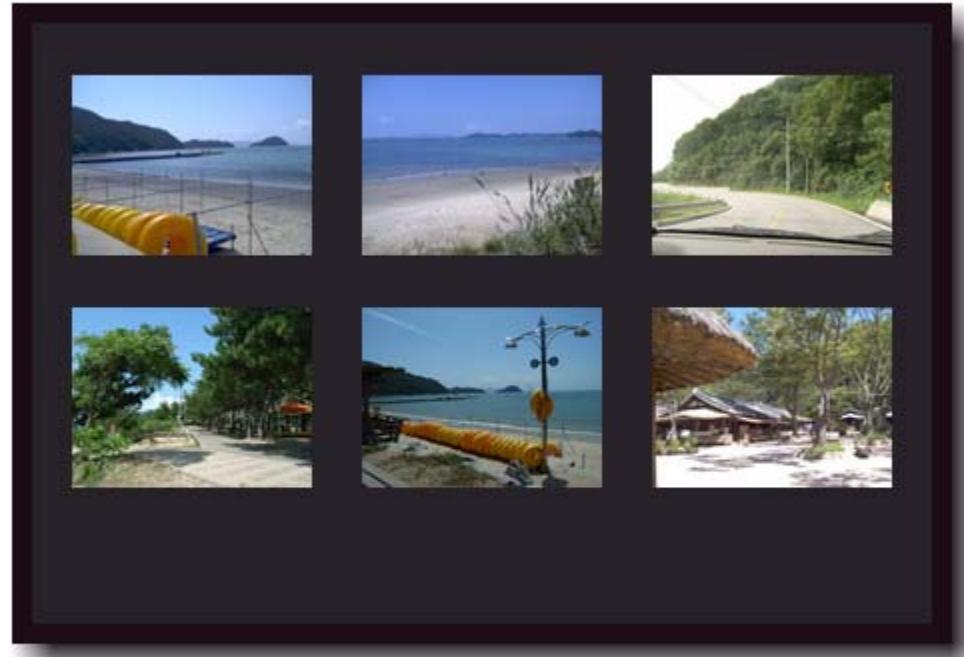
/* 썸네일 이미지를 올려놓을 갤러리 스타일 */
#gallery {
    border: 10px solid #1D0C16; /* 굵은 테두리 */
    height: 300px; /* 높이 */
    width: 450px; /* 너비 */
    margin-left: auto; /* 여백 자동 */
    margin-right: auto;
    overflow: visible; /* 오버플로일 경우 visible로 */
    background: #272229; /* 배경색 */
    -webkit-box-shadow: #272229 10px 10px 20px; /* 박스 그림자 */
    -moz-box-shadow: #272229 10px 10px 20px;
    filter: progid:DXImageTransform.Microsoft.Shadow(color='#272229',
Direction=135, Strength=10);
    box-shadow: #272229 10px 10px 20px;
}

/* 썸네일 목록 전체 스타일 */
#gallery ul{
    margin-left:-30px; /* 갤러리 가운데에 콘텐츠가 올 수 있도록 */
}
```

```
/* 썸네일 각 항목에 대한 스타일 */
#gallery ul li {
    list-style:none; /* 불릿 없앰 */
    display:inline-table; /* 테이블 스타일로 배치*/
    padding:10px; /* 안여백은 각각 10px로 */
}

/* 평소 큰 이미지 .pic 에 대한 스타일 */
#gallery ul li .pic{
    -webkit-transition: all 0.6s ease-in-out; /* 웹킷 브라우저에서 트랜지션 효과가 나타나도록 */
    opacity:0; /* 페이드아웃 효과 */
    visibility:hidden; /* 처음에는 보이지 않게 */
    position:absolute; /* 고정 위치 */
    margin-top:10px; /* 상단 여백 */
    margin-left:-20px; /* 왼쪽으로 20픽셀 이동해서 */
    border:1px solid black; /* 검은색 테두리 그림 */
    -webkit-box-shadow:#272229 2px 2px 10px; /* 박스 그림자 */
    -moz-box-shadow:#272229 2px 2px 10px; /* 박스 그림자 */
    filter:progid:DXImageTransform.Microsoft.Shadow(color='#272229', Direction=135,
Strength=5); /* 박스 그림자 */
    box-shadow:#272229 2px 2px 10px; /* 박스 그림자 */
}

</style>
```



예제: [samples\gallery3.html](#)  
(크롬/사파리 브라우저에서 확인할 것)

# 마우스 오버 효과 추가



```
#gallery ul li .small:hover{  
    cursor:pointer;  
}  
  
/* 썸네일 이미지 위로 마우스 오버했을 때 큰 이미지의 스타일 */  
#gallery ul li:hover .pic {  
    width:320px; /* 너비 */  
    height:240px; /* 높이 */  
    opacity:1; /* 불투명하게 */  
    visibility:visible; /* 화면에 보임 */  
    float:right; /* 오른쪽으로 플로팅 */  
}
```



예제: [samples\gallery3.html](#)  
(크롬/사파리 브라우저에서 확인할 것)

## **7.5 폴라로이드 느낌의 사진 만들기**

# 기본 HTML 소스



```
<body>
  
</body>
```



스타일을 적용할 수 있도록 `<div>` 태그로 감쌈

```
<body>
  <div class="polaroid">
    
  </div>
</body>
```



```
<style type="text/css">
.polaroid{
    position:relative;
    left:100px;
    top:50px;
    width:300px;
    padding:10px 10px 20px 10px;
    border: 1px solid #bfbfbf;
    background-color:#fff;
    -webkit-box-shadow: 2px 2px 3px rgba(135,139,144,0.4);
    -moz-box-shadow: 2px 2px 3px rgba(135.139.144,0.4);
    box-shadow: 2px 2px 3px rgba(135.139.133.0.4);
}
</style>
</head>

<body>
<div class="polaroid rotate">
    
    <p class="caption">여름이 깊어가고 있었다 </p>
</div>
</body>
```



여름이 깊어가고 있었다

```
.caption {  
    text-align:center;  
    font-family:"바탕";  
    font-weight:600;  
    font-size:16px;  
    color:#060;  
}  
  
.rotate{  
    -webkit-transform : rotate(9deg);  
    -moz-transform : rotate(9deg);  
    transform : rotate(9deg);  
}
```



## **7.6 클릭하면 펼쳐지는 아코디언 메뉴 만들기**

```

<div class="accordionMenu">
    <h1>아코디언 메뉴</h1>
    <div id="html5" class="menuSection">
        <h2><a href="#html5">HTML5</a></h2>
        <p>... </p>
    </div>
    <div id="css3" class="menuSection">
        <h2><a href="#css3">CSS3</a></h2>
        <p>...</p>
    </div>
    <div id="api" class="menuSection">
        <h2><a href="#api">API</a></h2>
        <p>...</p>
    </div>
</div>

```



# CSS 적용



```
<head>
<style>
.accordionMenu{
    font : 12px/20px 돋움;
    color:#424242;
    background:#fff;
    padding:10px;
    width:500px;
    margin:0;
}
.accordionMenu h1 {
    margin:0;
    font-size:20px;
    text-shadow:2px 2px 2px #a
}
.accordionMenu h2{
    margin:5px 0;
    paddion:0;
}
```

```
.accordionMenu h2 a{
    font-size:15px;
    display:block;
    font-weight:normal;
    color:#424242;
    text-shadow:2px 2px 2px #aeaeae;
    text-decoration:none;
    margin:0;
    padding:10px;
    background:#8f8f8f;
```

결과보기

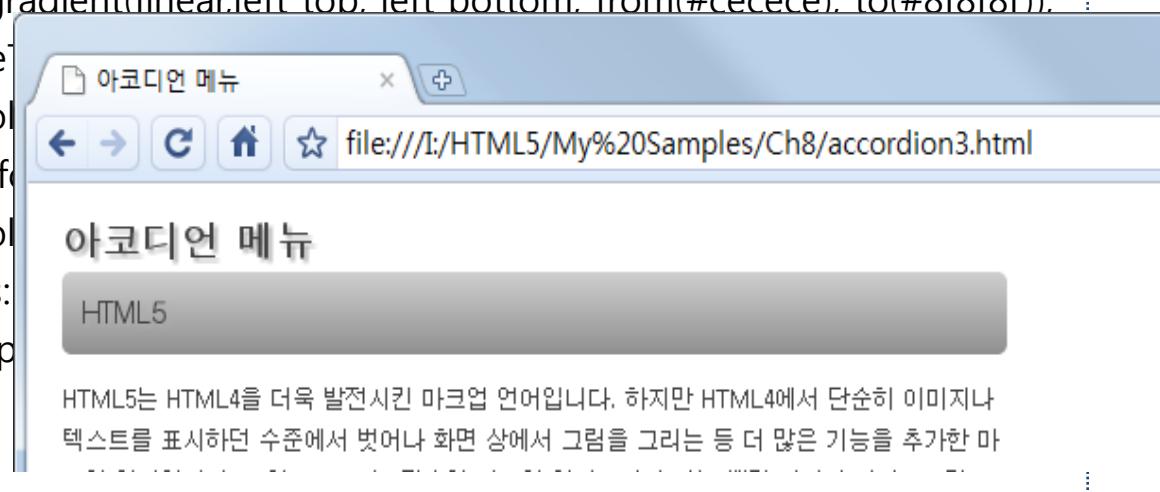


# CSS 적용 – 메뉴 탭 스타일에 그라데이션, 테두리



```
.accordionMenu h2 a{  
    font-size:15px;  
    display:block;  
    font-weight:normal;  
    color:#424242;  
    text-shadow:2px 2px 2px #aeaeae;  
    text-decoration:none;  
    margin:0;  
    padding:10px;  
    background:-moz-linear-gradient(top,#cecece,#8f8f8f);  
    background:-webkit-gradient(linear,left top, left bottom, from(#cecece), to(#8f8f8f));  
    filter: progid:DXImage  
(startColorstr=#ffcecece, endColorstr=#8f8f8f);  
-ms-filter: "progid:DXImageTransfo  
(startColorstr=#ffcecece, endColorstr=#8f8f8f);  
    -webkit-border-radius:  
    -moz-border-radius:5px;  
    border-radius:5px;  
}
```

결과보기



# CSS 적용 – 마우스 오버 스타일



```
.accordionMenu :target h2 a,  
.accordionMenu h2 a:focus,  
.accordionMenu h2 a:hover,  
.accordionMenu h2 :active {  
background:#2288dd;  
background:-moz-linear-gradient(top,#6bb2ff,#2288dd);  
background:-webkit-gradient(linear,left top, left bottom, from(#6bb2ff), to(#2288dd));  
-ms-filter: "progid:DXImageTransform.Microsoft.gradient(startColorstr=#ff6bb2ff,  
endColorstr=#ff2288dd)";  
color:#fff;  
}
```

결과보기

아코디언 메뉴

HTML5

HTML5는 HTML4를 더욱 발전시킨 마크업 언어입니다. 하지만 HTML4에서 단순히 이미지나 텍스트를 표시하던 수준에서 벗어나 화면 상에서 그림을 그리는 등 더 많은 기능을 추가한 마크업 언어입니다. 또한 HTML5는 단순히 마크업 언어로서의 기능 뿐만 아니라 자바스크립트를 사용해 프로그램을 만들 수 있는 API까지 포함하고 있습니다.

CSS3

CSS는 웹 디자인이나 웹 개발에 있어서 핵심적인 요소입니다. 특히 현재 개발 중인 CSS3는 기존의 CSS에서는 할 수 없던 여러 가지 효과를 쉽게 만들 수 있고 CSS3만으로도 간단한 애니메이션을 만들 수도 있고 멋진 내비게이션을 만들 수 있기 때문에 더욱 주목을 받고 있습니다.

# CSS 적용 – 트랜지션 효과



```
.accordionMenu p {  
padding:0 10px;  
margin:0;  
height:0;  
overflow:hidden;  
-moz-transition:height 0.5s ease-in;  
-webkit-transition:height 0.5s ease-in;  
-o-transition:height 0.5s ease-in;  
transition:height 0.5s ease-in;  
}  
  
.accordionMenu :target p{  
overflow: auto;  
height:100px;  
}
```

결과보기

## 아코디언 메뉴

HTML5

CSS3

CSS는 웹 디자인이나 웹 개발에 있어서 핵심적인 요소입니다. 특히 현재 개발 중인 CSS3는 기존의 CSS에서는 할 수 없던 여러 가지 효과를 쉽게 만들 수 있고 CSS3만으로도 간단한 애니메이션을 만들 수도 있고 멋진 내비게이션을 만들 수 있기 때문에 더욱 주목을 받고 있습니다.

API

## **8. 다양한 API**

# SVG(Scalable Vector Graphic)



웹 페이지에 직접 그림을 그림

CSS로 스타일을 지정할 수도 있고 동적인 비헤이비어도 추가할 수 있음.

```
<svg width="500" height="500">
  <rect x="10" y="10" width="100" height="100" fill="orange">
    <g transform="translate(70 21) rotate(-30)">
      <rect x="100" y="100" width="200" height="100" fill="green">
    </g>
  </svg>
```

# SVG(Scalable Vector Graphic)



웹 페이지에 직접 그림을 그림

CSS로 스타일을 지정할 수도 있고 동적인 비헤이비어도 추가할 수 있음.

svg 루트

```
<svg width="500" height="500">  
  <rect x="10" y="10" width="100" height="100" fill="orange">  
    <g transform="translate(70 21) rotate(-30)">  
      <rect x="100" y="100" width="200" height="100" fill="green">  
    </g>  
</svg>
```

# SVG(Scalable Vector Graphic)



웹 페이지에 직접 그림을 그림

CSS로 스타일을 지정할 수도 있고 동적인 비헤이비어도 추가할 수 있음.

그래픽 요소

```
<svg width="500" height="500">  
  <rect x="10" y="10" width="100" height="100" fill="orange">  
    <g transform="translate(70 21) rotate(-30)">  
      <rect x="100" y="100" width="200" height="100" fill="green">  
    </g>  
</svg>
```

- 도형이나 텍스트를 그리기 위한 요소.
- <rect>, <circle>, <ellipse>, <line>, <polyline>, <polygon>, <text>, <tspan>, <tref> 등

# SVG(Scalable Vector Graphic)



웹 페이지에 직접 그림을 그림

CSS로 스타일을 지정할 수도 있고 동적인 비헤이비어도 추가할 수 있음.

```
<svg width="500" height="500">
  <rect x="10" y="10" width="100" height="100" fill="orange">
    <g transform="translate(70 21) rotate(-30)">
      <rect x="100" y="100" width="200" height="100" fill="green">
    </g>
  </svg>
```

컨테이너 요소

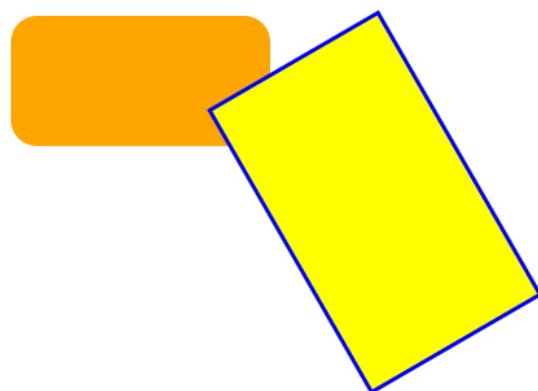
- <g>, <defs>, <use>, <desc> 등

# SVG – 사각형 그리기 <rect>

<b>x, y</b>	사각형의 시작 위치.
<b>width, height</b>	사각형의 너비와 높이
<b>rx, ry</b>	모서리가 둥근 사각형일 경우 좌표에서의 원의 반지름

```
<svg width="500" height="500">  
  <rect x="10" y="10" width="100" height="100"  
        fill="orange" rx="20px"></rect>  
  <g transform="translate(70 21) rotate(-30)">  
    <rect x="100" y="100" width="200" height="100"  
          fill="yellow" stroke="blue" stroke-width="3"></rect>  
  </g>  
</svg>
```

결과보기

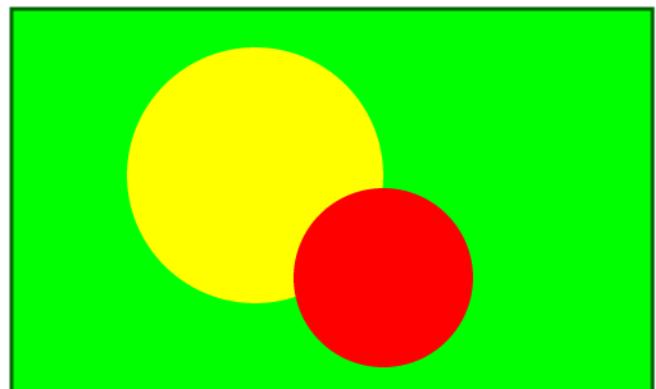


# SVG – 원 그리기 <circle>

<b>cx, cy</b>	원 중점의 좌표
<b>r</b>	원의 반지름

```
<svg width="600" height="500">  
  <rect x="10" y="10" width="500" height="300"  
        fill="lime" stroke="darkgreen" stroke-width="3">  
  </rect>  
  <circle cx="200" cy="140" r="100" fill="yellow"></circle>  
  <circle cx="300" cy="220" r="70" fill="red"></circle>  
</svg>
```

결과보기

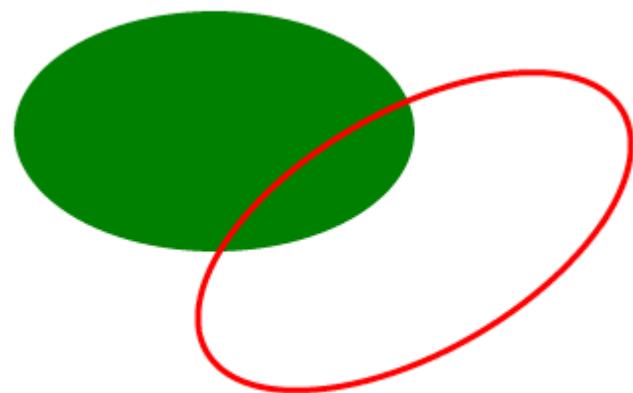


# SVG – 타원 그리기 <ellipse>

<b>cx, cy</b>	타원 중점의 좌표
<b>rx, ry</b>	타원의 가로축 반지름과 세로축 반지름.

```
<svg width="600" height="500">  
<ellipse cx="150" cy="100" rx="100" ry="60"  
    fill="green"></ellipse>  
<g transform="translate(250 150) rotate(60)">  
    <ellipse rx="60" ry="120" fill="none" stroke="red"  
        stroke-width="3"></ellipse>  
</g>  
</svg>
```

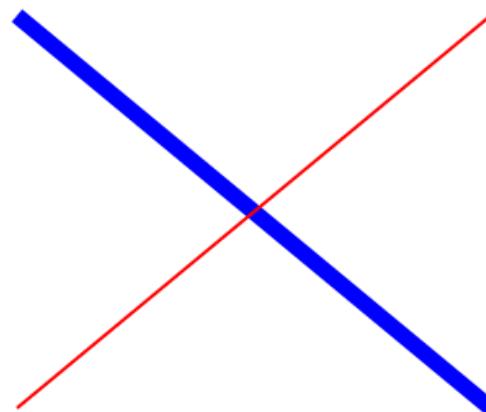
결과보기



x1, y1	시작점 좌표
x2, y2	끝점 좌표

```
<svg width="600" height="500">
  <line x1="10" y1="10" x2="300" y2="250" stroke="blue"
stroke-width="10" />
  <line x1="300" y1="10" x2="10" y2="250" stroke="red"
stroke-width="2"/>
</svg>
```

결과보기



웹 상에 있는 요소를 끌어옮길 수 있게 함

## dataTransfer 객체

프로퍼티	dropEffect	드래그가 끝날 때 사용할 효과 지정
	effectAllowed	드래그하는 동안 사용할 효과 지정
메서드	clearData(type)	해당 유형의 데이터 삭제
	getData(type)	해당 유형의 데이터 가져옴
	setData(type, data)	주어진 data 값으로 해당 유형의 데이터를 설정

## draggable 속성

객체를 드래그할 수 있는지 지정

ex) <div id="drag1" **draggable="true"**

## 드래그 데이터

드래그 이벤트가 발생했을 때 드래그되는 대상  
데이터 유형과 값, 두 가지 정보가 들어있음

text/plain 또는 text	텍스트
text/uri-list	사이트 주소. 주소가 하나일 때는 URL이라고 쓰기도 함
file	다른 애플리케이션 파일

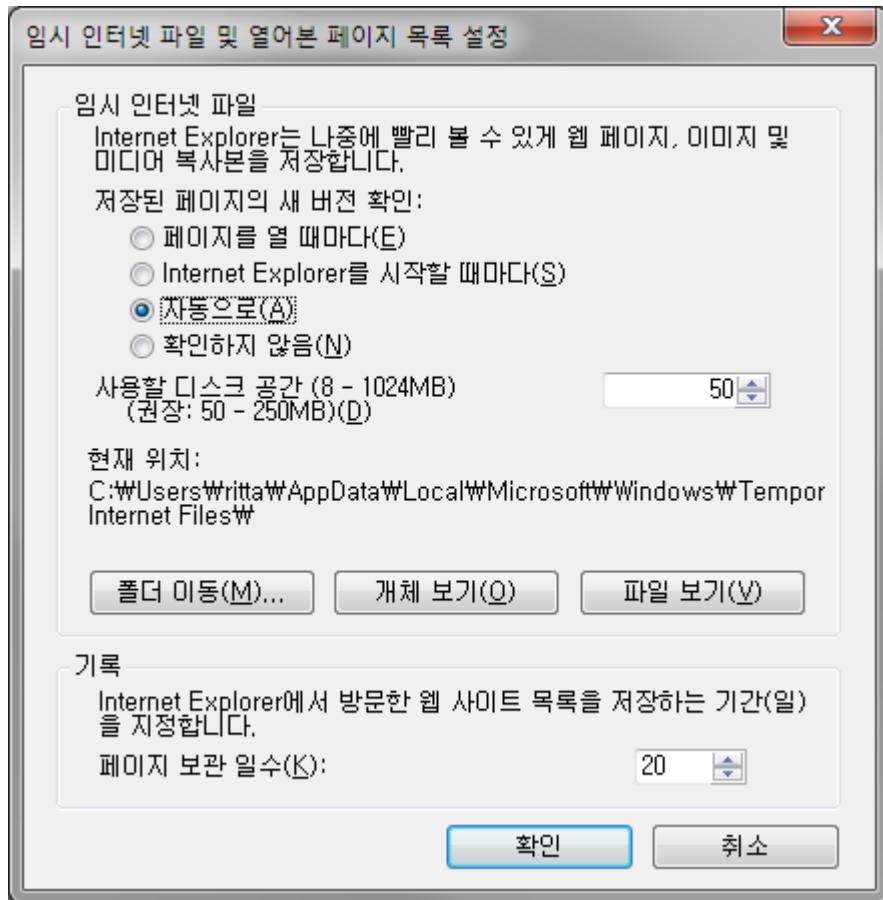
## 드래그 이펙트

dropEffect 프로퍼티와 effectAllowed 프로퍼티에서 사용할 효과

copy	객체를 드롭 대상 위치로 복사	move	객체를 드롭 대상 위치로 이동
link	원래 위치와 드롭 대상 위치 사이에 링크를 만듦		

## 드래그 이벤트

dragstart	사용자가 객체를 드래그하려고 할 때
drag	마우스가 움직일 때
dragenter	드롭할 위치로 옮겨질 때. 해당 위치로 드롭할 것인지 여부를 결정해야 함.
dragover	드래그되는 요소가 드롭 대상 위치 위로 옮겨질 때.
dragleave	드래그되는 요소가 드롭 대상 영역을 벗어나서 이동할 때.
drop	드롭 대상 영역 위에 성공적으로 드롭되었을 때.
dragend	성공하든 실패하든 드래그가 끝났을 때.



## IE의 임시 인터넷 파일

한번 접속했던 사이트의 이미지,  
미디어 복사본 저장

애플리케이션 캐시와 비슷한 개념

오프라인 사용이 불가능

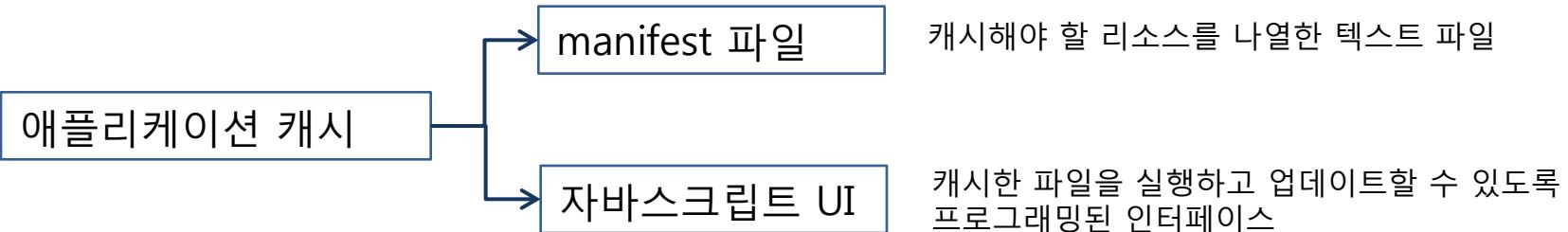
웹에 접속할 수 없는 상황에서도 애플리케이션 실행되어야.

## <장점>

**오프라인 브라우징** : 오프라인 상태에서도 사용자가 사이트를 둘러볼 수 있다.

**속도** : 캐시된 리소스들이 로컬에 있기 때문에 좀 더 빠르게 로딩할 수 있다.

**서버 로딩 줄임** : 리소스 변경이 있을 경우에만 서버에서 리소스를 다운로드하면 된다.



## <manifest 파일>

CACHE MANIFEST

# VERSION 10

NETWORK:

<http://robertnyman.com/html5/offline/offline.html>

CACHE:

offline.html

base.css

FALLBACK:

online.css offline.css

## <manifest 파일>

CACHE MANIFEST

# VERSION 10

NETWORK:

<http://robertnyman.com/html5/offline/offline.html>

온라인일 때 액세스할 파일

CACHE:

offline.html

base.css

FALLBACK:

online.css offline.css

## <manifest 파일>

CACHE MANIFEST

# VERSION 10

NETWORK:

<http://robertnyman.com/html5/offline/offline.html>

CACHE:

offline.html

base.css

오프라인일 때 액세스할 파일

FALLBACK:

online.css offline.css

## <manifest 파일>

CACHE MANIFEST

# VERSION 10

NETWORK:

<http://robertnyman.com/html5/offline/offline.html>

CACHE:

offline.html

base.css

FALLBACK:

online.css offline.css

지정한 파일이 없을 경우 사용  
할 파일

## <html manifest="offline.manifest">

- 애플리케이션 캐시가 만들어지면 페이지가 한번 로딩된 후로는 리소스가 자동으로 캐시에 저장됨.
- manifest 파일이 수정되었을 경우에만 서버로부터 파일을 가져옴.
- (IE 임시 인터넷 파일은 사이트에 접속할 때마다 임시 인터넷 파일을 갱신함)

서버에서 MIME 타입을 추가해야 함

## 브라우저 쿠키(cookie)의 단점

크기가 작다. (최대 크기 약 4KB 정도)

같은 사이트에서 둘 이상의 탭을 열면 둘 이상의 트랜잭션을 추적하기 어렵다.

보안 문제를 일으키기 쉽다.



## 데이터 저장 및 검색

저장      sessionStorage.setItem(key, value);    localStorage.setItem(key, value);

ex) sessionStorage.setItem(1, '이것은 예제 문장입니다');

검색      var item = sessionStorage.getItem(key);

var item = localStorage.getItem(key);

ex) var item=sessionStorage.getItem(1);  
alert(item);

## 데이터 삭제

삭제      sessionStorage.removeItem(key);    localStorage.removeItem(key);

모두 삭제 sessionStroage.clear();        localStroage.clear();

속성	설명
storageArea	스토리지 유형(세션인지 로컬인지) 표시
key	변경된 키
oldValue	키의 이전 값
newValue	키의 새로운 값
url	키가 변경된 페이지의 URL

# 웹 스토리지 – 스토리지 이벤트의 속성들



```
function savethestuffSession(){  
    var lala = document.getElementById("first");  
    var thevalue = lala.value;  
    sessionStorage.setItem(1, thevalue);  
    getthestuffSession();  
}  
  
function getthestuffSession(){  
    var data;  
    var thediv = document.getElementById("displaysession");  
    data = sessionStorage.getItem(1);  
    if (data){  
        thediv.innerHTML = "저장된 값: "+data;  
    }  
}
```

결과보기

## 세션 스토리지(Session Storage)

HTML5 is powerful

저장된 값:HTML5 is powerful

## 로컬 스토리지(Local Storage)

CSS3 is fantastic

저장된 값:CSS3 is fantastic

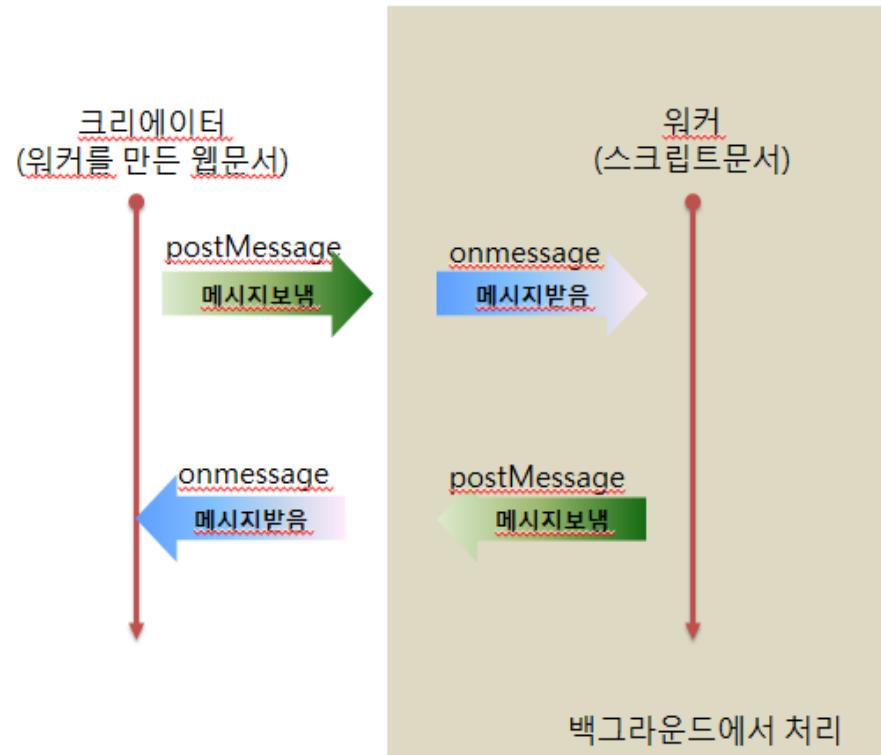
## 세션 스토리지(Session Storage)

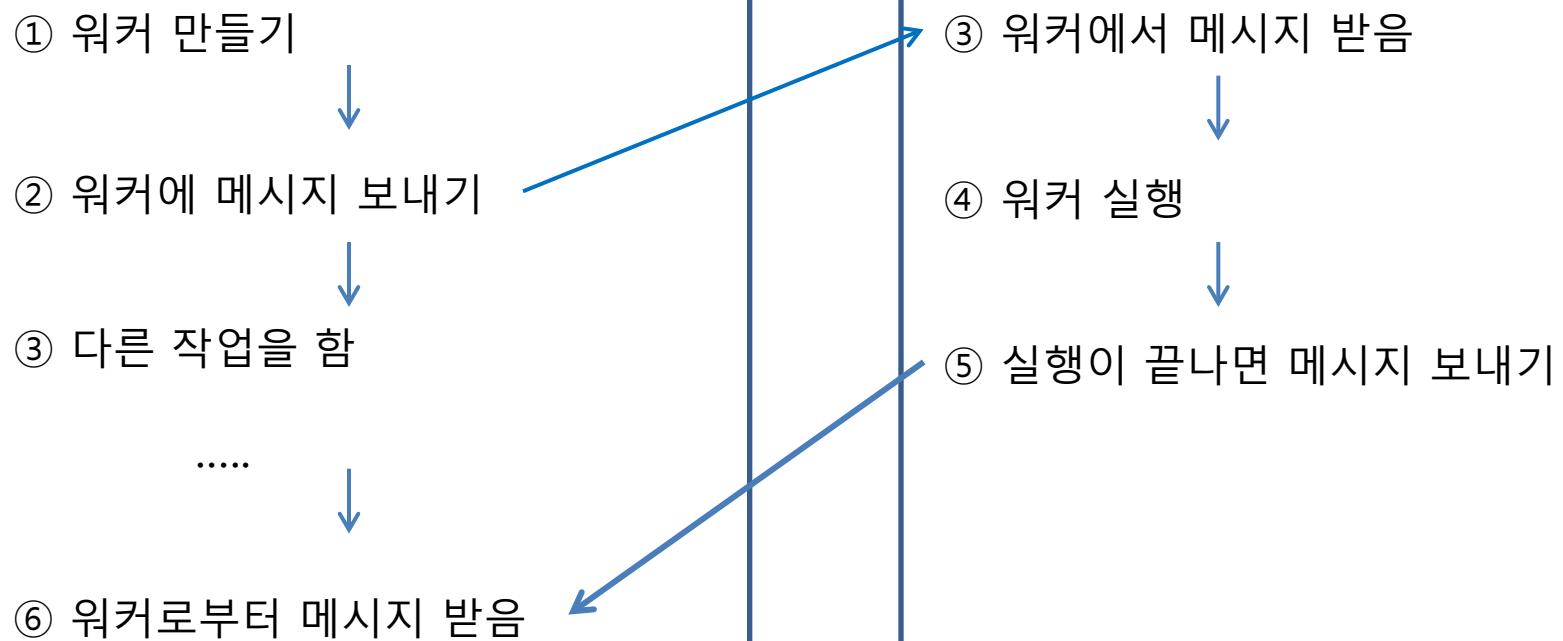
비어있음

## 로컬 스토리지(Local Storage)

저장된 값:CSS3 is fantastic

## 오래 걸리는 스레드를 실행하는 동안, 백그라운드 스레드에서 스크립트 실행





## ① 지오로케이션 서비스를 사용할 수 있는 브라우저인지 체크

```
if (navigator.geolocation) {  
    /* 지오로케이션 API 소스 작성 */  
}  
else {  
    alert("이 브라우저에서는 지오로케이션 서비스를 사용할 수 없습니다.");  
}
```

## ② 현재 위치 확인 – `getCurrentPosition` 메서드

## ③ 현재 위치 계속 확인 – `watchPosition` 메서드

## 사용자 위치 설명 – position 객체 이용

timestamp : 위치를 가져온 시간

coords : nsIDOMGeoPositionCoords 객체를 이용해 좌표 표시

### nsIDOMGeoPositionCoords 객체의 속성

latitude	사용자 위치의 위도
longitude	사용자 위치의 경도
altitude	사용자 위치의 표고. 표고를 지원하지 않는 장비에서는 0
accuracy	위치 정보의 오차
altitudeAccuracy	표고 정보의 오차. 표고를 지원하지 않는 장비에서는 0
heading	사용자의 진행 방향
speed	사용자가 움직이는 속도. m/s로 표시

```
<head>
  <script type="text/javascript" src="http://apis.daum.net/maps/maps2.js?apikey="*****"></script>
</head>
<body>
  <div id="map" style="width:400px; height:400px"></div>
  <script language="Javascript">
    var map;
    if (navigator.geolocation) {
      navigator.geolocation.getCurrentPosition(
        function(position) {
          lat=position.coords.latitude;
          lng=position.coords.longitude;

          map = new DMap("map");
          map.setCenter(new DLatLng(lat, lng));

          var indexMapControl = new DIndexMapControl();
          var zoomControl = new DZoomControl();
          ...
          zoomControl.setVAlign("middle");

          m = new DMark(new DLatLng(lat, lng));
          map.addOverlay(m);
        });
    }
  </script>
```

```
<head>
<script type="text/javascript" src="http://apis.daum.net/maps/maps2.js?apikey="*****"></script>
</head>
<body>
<div id="map" style="width:400px; height:400px"></div>
<script language="Javascript">
var map;
if (navigator.geolocation) {
    navigator.geolocation.getCurrentPosition(
        function(position) {
            lat=position.coords.latitude;
            lng=position.coords.longitude;

            map = new DMap("map");
            map.setCenter(new DLatLng(lat, lng));

            var indexMapControl = new DIndexMapControl();
            var zoomControl = new DZoomControl();
            ...
            zoomControl.setVAlign("middle");

            m = new DMark(new DLatLng(lat, lng));
            map.addOverlay(m);
        });
}
</script>
```

다음 API 등록키. 개인별로 값이 모두 다름

```
<head>
  <script type="text/javascript" src="http://apis.daum.net/maps/maps2.js?apikey="*****"></script>
</head>
<body>
  <div id="map" style="width:400px; height:400px"></div>
  <script language="Javascript">
    var map;
    if (navigator.geolocation) {
      navigator.geolocation.getCurrentPosition(
        function(position) {
          lat=position.coords.latitude;
          lng=position.coords.longitude;
          ...
          zoomControl.setVAlign("middle");
          m = new DMark(new DLatLng(lat, lng));
          map.addOverlay(m);
        });
    }
  </script>
```

현재 위치 알아내기

```
<head>
  <script type="text/javascript" src="http://apis.daum.net/maps/maps2.js?apikey="*****"></script>
</head>
<body>
  <div id="map" style="width:400px; height:400px"></div>
  <script language="Javascript">
    var map;
    if (navigator.geolocation) {
      navigator.geolocation.getCurrentPosition(
        function(position) {
          lat=position.coords.latitude;
          lng=position.coords.longitude;

          map = new DMap("map");
          map.setCenter(new DLatLng(lat, lng));

          var indexMapControl = new DIndexMapControl();
          var zoomControl = new DZoomControl();
          ...
          zoomControl.setVAlign("middle");

          m = new DMark(new DLatLng(lat, lng));
          map.addOverlay(m);
        });
    }
  </script>
```

현재 위치를 기준으로 map 만듬

```
<head>
  <script type="text/javascript" src="http://apis.daum.net/maps/maps2.js?apikey="*****"></script>
</head>
<body>
  <div id="map" style="width:400px; height:400px"></div>
  <script language="Javascript">
    var map;
    if (navigator.geolocation) {
      navigator.geolocation.getCurrentPosition(
        function(position) {
          lat=position.coords.latitude;
          lng=position.coords.longitude;

          map = new DMap("map");
          map.setCenter(new DLatLng(lat, lng));

          var indexMapControl = new DIndexMapControl();
          var zoomControl = new DZoomControl();
          ...
          zoomControl.setVAlign("middle");

          m = new DMark(new DLatLng(lat, lng));
          map.addOverlay(m);
        });
    }
  </script>
```

확대/축소 등 UI 컨트롤 설정

```
<head>
  <script type="text/javascript" src="http://apis.daum.net/maps/maps2.js?apikey="*****"></script>
</head>
<body>
  <div id="map" style="width:400px; height:400px"></div>
  <script language="Javascript">
    var map;
    if (navigator.geolocation) {
      navigator.geolocation.getCurrentPosition(
        function(position) {
          lat=position.coords.latitude;
          lng=position.coords.longitude;

          map = new DMap("map");
          map.setCenter(new DLatLng(lat, lng));

          var indexMapControl = new DIndexMapControl();
          var zoomControl = new DZoomControl();
          ...
          zoomControl.setVAlign("middle");

          m = new DMark(new DLatLng(lat, lng));
          map.addOverlay(m);
        });
    }
  </script>
```

현재 위치에 표시