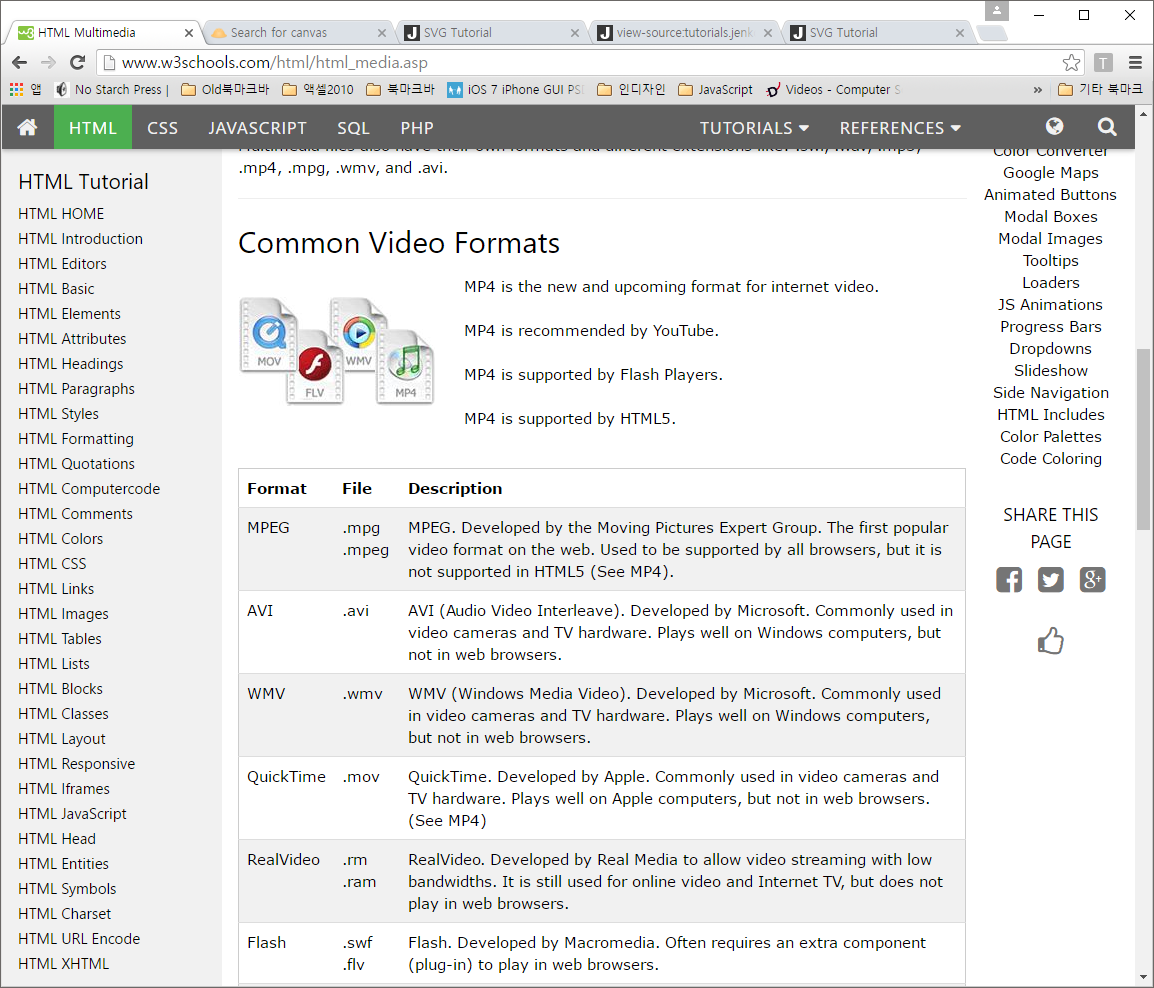
**HTML Multimedia**

멀티미디어는 sound, music, videos, movies, and animations

일반적으로 Video 포맷은 MP4를 사용하면 된다. MP4는 YouTube, Flash Player, HTML5 등에서 모두 작동한다.

**Video 포맷**

|  |  |  |
| --- | --- | --- |
| Format | File | Description |
| MPEG | .mpg .mpeg | MPEG. Developed by the Moving Pictures Expert Group. The first popular video format on the web. Used to be supported by all browsers, but it is not supported in HTML5 (See MP4). |
| AVI | .avi | AVI (Audio Video Interleave). Developed by Microsoft. Commonly used in video cameras and TV hardware. Plays well on Windows computers, but not in web browsers. |
| WMV | .wmv | WMV (Windows Media Video). Developed by Microsoft. Commonly used in video cameras and TV hardware. Plays well on Windows computers, but not in web browsers. |
| QuickTime | .mov | QuickTime. Developed by Apple. Commonly used in video cameras and TV hardware. Plays well on Apple computers, but not in web browsers. (See MP4) |
| RealVideo | .rm .ram | RealVideo. Developed by Real Media to allow video streaming with low bandwidths. It is still used for online video and Internet TV, but does not play in web browsers. |
| Flash | .swf .flv | Flash. Developed by Macromedia. Often requires an extra component (plug-in) to play in web browsers. |
| Ogg | .ogg | Theora Ogg. Developed by the Xiph.Org Foundation. Supported by HTML5. |
| WebM | .webm | WebM. Developed by the web giants, Mozilla, Opera, Adobe, and Google. Supported by HTML5. |
| MPEG-4 or MP4 | .mp4 | MP4. Developed by the Moving Pictures Expert Group. Based on QuickTime. Commonly used in newer video cameras and TV hardware. Supported by all HTML5 browsers. Recommended by YouTube. |

**Sound 포맷**

|  |  |  |
| --- | --- | --- |
| Format | File | Description |
| MIDI | .mid .midi | MIDI (Musical Instrument Digital Interface). Main format for all electronic music devices like synthesizers and PC sound cards. MIDI files do not contain sound, but digital notes that can be played by electronics. Plays well on all computers and music hardware, but not in web browsers. |
| RealAudio | .rm .ram | RealAudio. Developed by Real Media to allow streaming of audio with low bandwidths. Does not play in web browsers. |
| WMA | .wma | WMA (Windows Media Audio). Developed by Microsoft. Commonly used in music players. Plays well on Windows computers, but not in web browsers. |
| AAC | .aac | AAC (Advanced Audio Coding). Developed by Apple as the default format for iTunes. Plays well on Apple computers, but not in web browsers. |
| WAV | .wav | WAV. Developed by IBM and Microsoft. Plays well on Windows, Macintosh, and Linux operating systems. Supported by HTML5. |
| Ogg | .ogg | Ogg. Developed by the Xiph.Org Foundation. Supported by HTML5. |
| MP3 | .mp3 | MP3 files are actually the sound part of MPEG files. MP3 is the most popular format for music players. Combines good compression (small files) with high quality. Supported by all browsers. |
| MP4 | .mp4 | MP4 is a video format, but can also be used for audio. MP4 video is the upcoming video format on the internet. This leads to automatic support for MP4 audio by all browsers. |

**HTML Plug-ins**

Plug-in은 목적은 브라우저의 기능을 확대하기 위한 것이다. 플러그 인은 웹 브라우저의 기능을 확대하기 위한 표준화된 프로그램이다. 가장 많이 알려진것으로는 java applets 이다. Plug-in은 <object> <embed> 태그안에 사용한다.

<object width="400" height="50" data="bookmark.swf"></object>

<object data="cat.jpg"></object>

<embed width="400" height="50" src="bookmark.swf">

HTML5 파일 API 은 아래의 내용을 포함한다.

* [Core Objects of HTML5 File API](http://tutorials.jenkov.com/html5/file-api.html#core-objects-of-html5-file-api)
* [Selecting Files](http://tutorials.jenkov.com/html5/file-api.html#selecting-files)
  + [Selecting Multiple Files](http://tutorials.jenkov.com/html5/file-api.html#selecting-multiple-files)
* Selecting Files Via Drag and Drop
* Accessing Selected Files
* Loading Files With a FileReader
  + Creating a FileReader
  + Loading a File as Text
  + Loading a File as Text Slice
  + Loading a File as Data URL
  + Load a File as ArrayBuffer
* Monitoring Progress of File Loading
* Browser Support for HTML5 File API

|  |
| --- |
| 간단한 예제들 |
| <input type="file" onchange="readFiles(event)" > |
| <input type="file" onchange="readFiles(event)" multiple> |

**HTML5 History API**

* [The History Stack](http://tutorials.jenkov.com/html5/history-api.html)
* [HTML5 History API Security Restrictions](http://tutorials.jenkov.com/html5/history-api.html#html5-history-api-security-restrictions)
* [The history Object](http://tutorials.jenkov.com/html5/history-api.html#the-history-object)
* [History API Examples](http://tutorials.jenkov.com/html5/history-api.html#history-api-examples)
  + [back() and forward()](http://tutorials.jenkov.com/html5/history-api.html#back-and-forward)
  + [go()](http://tutorials.jenkov.com/html5/history-api.html#go)
  + [pushState()](http://tutorials.jenkov.com/html5/history-api.html#pushstate)
  + [replaceState()](http://tutorials.jenkov.com/html5/history-api.html#replacestate)
* [History Change Events](http://tutorials.jenkov.com/html5/history-api.html#history-change-events)
* [Using The History API in Practice](http://tutorials.jenkov.com/html5/history-api.html#using-the-history-api-in-practice)
  + [Configuring The Server](http://tutorials.jenkov.com/html5/history-api.html#configuring-the-server)
* [Browser Support For The HTML5 History API](http://tutorials.jenkov.com/html5/history-api.html#browser-support-for-the-html5-history-api)

**히스토리 object 기능들**

* back()
* forward()
* go(index)
* pushState(stateObject, title, url)
* replaceState(stateObject, title, url)

**<div>**

우리는 이제까지 어떤 특정한 목적을 갖고 있는 태그들을 보았다. <P>는 문단, <h1>은 헤딩 <a>는 링크를 위한 것이었다. 그렇다면 <div>는 자신의 특별한 것을 표현하지 않고 뒤따라오는 내용을 담는 일반적인 컨테이너이다

Html5에서 <header> <nav> <footer>등의 새로운 태그들이 <div>태그의 기능을 대치했다.

|  |
| --- |
| <div style="background-color:orange;text-align:center">  <p>Bright Widget</p>  </div>  <div style="background-color:olive;text-align:center">  <p>Dull widget</p>  </div> |
| <div hidden>  You can't see me!  </div> |

**Layouts**

웹사이트는 여러 개의 열 (column)안에 내용을 표현한다.

|  |  |
| --- | --- |
| <style>  #header {  background-color:black;  color:white;  text-align:center;  padding:5px; }  #nav {  line-height:30px;  background-color:#eeeeee;  height:300px;  width:100px;  float:left;  padding:5px; }  #section {  width:350px;  float:left;  padding:10px; }  #footer {  background-color:black;  color:white;  clear:both;  text-align:center;  padding:5px; }  </style> | <div id="header">  <h1>City Gallery</h1>  </div>  <div id="nav">  London<br>  Paris<br>  Tokyo  </div>  <div id="section">  <h1>London</h1>  <p>London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.</p>  <p>Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.</p>  </div>  <div id="footer">  Copyright © W3Schools.com  </div> |
| <style>  header {  background-color:black;  color:white;  text-align:center;  padding:5px; }  nav {  line-height:30px;  background-color:#eeeeee;  height:300px;  width:100px;  float:left;  padding:5px; }  section {  width:350px; float:left;  padding:10px; }  footer {  background-color:black;  color:white; clear:both;  text-align:center; padding:5px; }  </style> | <header>  <h1>City Gallery</h1>  </header>  <nav>  London<br>  Paris<br>  Tokyo  </nav>  <section>  <h1>London</h1>  <p>London is the capital city of England. It is the most populous city in the United Kingdom, ith a metropolitan area of over 13 million inhabitants.</p>  <p>Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.</p>  </section>  <footer>  Copyright © W3Schools.com  </footer> |

**CSS**

**C**ascading **S**tyle **S**heets 약자이다. CSS는 3개의 방법으로 HTML페이지에 추가된다.

* Inline - HTML 태그안에 직접 적는다
* Internal - <head> 태그안의 <style> 태그를 적고 그 안에 적는다
* External - **external CSS files (외부파일) 형식으로 사용한다**
* **Inline 방식**  
  형식은 style="property:value;"

실습예제

<body style="background-color:lightgrey;">

<h1 style="color:blue;">This is a heading</h1>

<p style="color:red;">This is a paragraph.</p>

<h1 style="font-family:verdana;">This is a heading</h1>

<p style="font-family:courier;">This is a paragraph.</p>

<h1 style="font-size:300%;">This is a heading</h1>

<p style="font-size:160%;">This is a paragraph.</p>

<h1 style="text-align:center;">Centered Heading</h1>

<p>This is a paragraph.</p>  
</body>

* **Internal 방식**

<!DOCTYPE html>

<html>

<head>

<style>

body {background-color:lightgrey;}

h1 {color:blue;}

p {color:green;}

</style>

</head>

<body>

<h1>This is a heading</h1>

<p>This is a paragraph.</p>

</body>

</html>

* **External 방식**

|  |  |
| --- | --- |
| <!DOCTYPE html> <html> <head>   <link rel="stylesheet" href="styles.css"> </head> <body>  <h1>This is a heading</h1> <p>This is a paragraph.</p>  </body> </html> | body {     background-color: lightgrey; }  h1 {     color: blue; }  p {     color:green; } |

styles.css 파일에 다음의 내용을 적고 저장한다

**<Form>**

사용자의 입력을 수집할 때 사용한다. 폼은 input, checkboxes, radio buttons, submit button 등

형식은

<form>  
.  
*form elements*  
.  
</form>

* 1. <Input> 은 가장 중요한 from 태그 속성이다.

|  |  |
| --- | --- |
| Type | 설명 |
| text | 일반적인 문자입력 |
| radio | 여러 개 중 하나만을 선택할 때 사용되는 radio button 입력) |
| submit | 전송버튼 ( 폼의 내용을 서버에게 보내는) |

|  |
| --- |
| <form> 실습하기  First name:<br>  <input type="text" name="firstname"><br>  Last name:<br>  <input type="text" name="lastname">  </form> |
| <form> 실습하기  <input type="radio" name="gender" value="male" checked> Male<br>  <input type="radio" name="gender" value="female"> Female<br>  <input type="radio" name="gender" value="other"> Other  </form> |
| <input type="radio" name="gender" value="male" checked> Male<br> 실습하기  <input type="radio" name="gender" value="female"> Female<br>  <input type="radio" name="gender" value="other"> Other <br>  First name:<br>  <input type="text" name="firstname" value="Mickey"><br>  Last name:<br>  <input type="text" name="lastname" value="Mouse"><br><br>  <input type="submit" value="Submit"><br>    <button type="button">Click me</button>  <button type="reset">Reset me</button>  <button type="submit">Submit me</button> |
| <form action="action\_page.php">  First name:<br>  <input type="text" name="firstname" value="Mickey"><br>  Last name:<br>  <input type="text" name="lastname" value="Mouse"><br><br>  <input type="submit" value="Submit">  </form> |

Action 속성은 해당 폼이 보내졌을 때 어떻게 처리되는 지를 정의하며 submit 속성에 의해 폼은 서버로 보내어진다. 또한 get 또는 post 속성이 있는데 폼이 보내어질 때 사용된다.

사용방법은

<form action="action\_page.php" method="get">

또는

<form action="action\_page.php" method="post">

기본값은 get 으로 설정되어 있으며 브라우저의 주소창에   
action\_page.php?firstname=Mickey&lastname=Mouse 와 같은 정보를 볼 수 있다. 데이터를 서버로 보낼 때 어떤 데이터가 어떤 데이터인지( what’s what) 를 쉽게 구별하기 위해 name 속성을 사용한다.

또한 관련있는 데이터끼리 묶을 수 있도록 <fielddset>이 있고 이 <fieldset>의 캡션을 위해 <legend> 요소가 있다.

|  |
| --- |
| fieldset and legend Elements Form |
| <fieldset >  <legend >Box Title</legend>  Here is some content  <p>Here is another paragraph</p>  </fieldset>  (아래의 스타일을 적용하면)  <body>  <fieldset style="border:#F00 1px solid;">  <legend style="color:#F00;">Box Title</legend>  Here is some content  <p>Here is another paragraph</p>  </fieldset>  </body> |
| <body>  <form style="width:440px;" action="<?php echo $\_SERVER['PHP\_SELF']; ?>" method="post">  <fieldset style="text-align:left; padding:24px;">  <legend>Subscribe to Our Newsletter &nbsp;</legend>  <br />  Name:<br />  <input name="name" type="text" maxlength="36" value="<?php echo $name; ?>" /><br />  Email:<br />  <input name="email" type="text" maxlength="36" value="<?php echo $email; ?>" /><br /><br />  <input name="mySubmitBtn" type="submit" value="Submit">  <?php echo $msg\_to\_user; ?>  </fieldset>  </form>  </body> |
| <?php  /\* -------------------------------------------------------------------  Script written by Adam Khoury @ www.developphp.com  January 1, 2010  Please retain this credit when displaying this code online  ---------------------------------------------------------------------- \*/  $name = "";  $email = "";  $msg\_to\_user = "";  if ($\_POST['name'] != "") {    include\_once "connect\_to\_mysql.php";    // Be sure to filter this data to deter SQL injection, filter before querying database  $name = $\_POST['name'];  $email = $\_POST['email'];    $sql = mysql\_query("SELECT \* FROM newsletter WHERE email='$email'");  $numRows = mysql\_num\_rows($sql);    if (!$email) {  $msg\_to\_user = '<br /><br /><h4><font color="FF0000">Please type an email address ' . $name . '.</font></h4>';  } else if ($numRows > 0) {    $msg\_to\_user = '<br /><br /><h4><font color="FF0000">' . $email . ' is already in the system.</font></h4>';  } else {  $sql\_insert = mysql\_query("INSERT INTO newsletter (name, email, dateTime)  VALUES('$name','$email',now() )") or die (mysql\_error());  $msg\_to\_user = '<br /><br /><h4><font color="0066FF">Thanks ' . $name . ', you have been added successfully.</font></h4>';  $name = "";  $email = "";  }  }  ?>  위의 2개의 내용을 한 페이지에 삽입하고 index.php 로 저장한다. |