

05 웹페이지의 작성

05 Web Page Creation

차례

5.1 홈페이지와 HTML 문서

5.1.1 HTML의 역사

5.1.2 HTML 문서의 구조와 텍스트 관련 태그

5.1.3 하이퍼링크 및 멀티미디어 포함하기

5.1.4 다양한 형식의 내용 구성하기

5.2 동적 웹페이지

5.2.1 동적 HTML의 특징

5.2.2 스타일시트의 개념 및 CSS

5.2.3 자바스크립트(JavaScript) 및 DOM의 작동원리

5.3 서버 연동 페이지

5.3.1 CGI 원리와 Form 요소

5.3.2 CGI 프로그래밍

5.3.3 웹서버 프로그램 PHP

5.3.4 웹서버 프로그램 ASP와 JSP

5.4 웹문서 교환 표준 XML

5.4.1 XML의 개요

5.4.2 XML 문서 작성 원리

5.5 새로운 문서 표준 HTML5

5.5.1 HTML5의 탄생 배경 및 특징

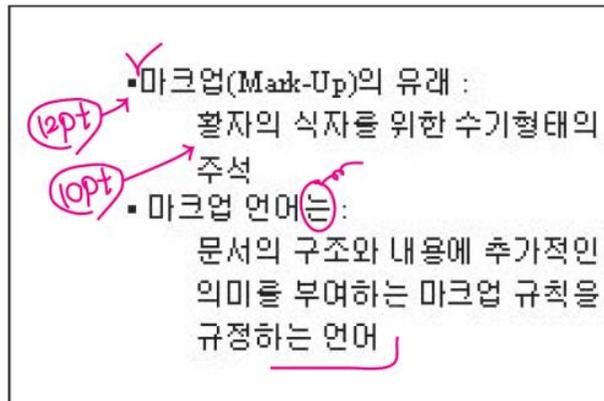
5.5.2 HTML5에서 변경된 엘리먼트

5.5.3 애플리케이션 개발을 위한 다양한 API

5.5.4 모바일 웹

HTML의 역사_1

- HTML(HyperText Markup Language)
 - 팀 버너스리가 웹(WWW; World Wide Web) 방식 제안
 - 웹에서 하이퍼텍스트를 구현하기 위해 HTML 언어 제시
 - 웹 페이지를 위한 마크업 언어
 - 활자인쇄 시대의 마크-업에서 유래
 - 문서의 속성을 설정하기 위한 마크업을 태그(Tag)로 표시
 - HTML 언어는 SGML 표준에 따라 정의
 - 확장자는 *.htm, *.html



```
<li><font size="12pt">마크업(Mark-Up)의 유래 :</font>
<br><font size="10pt">활자의 식자를 위한 수기 형태의 주석</font></li>
<li><font size="12pt">마크업 언어 :</font>
<br><font size="10pt">문서의 구조와 내용에 추가적
```

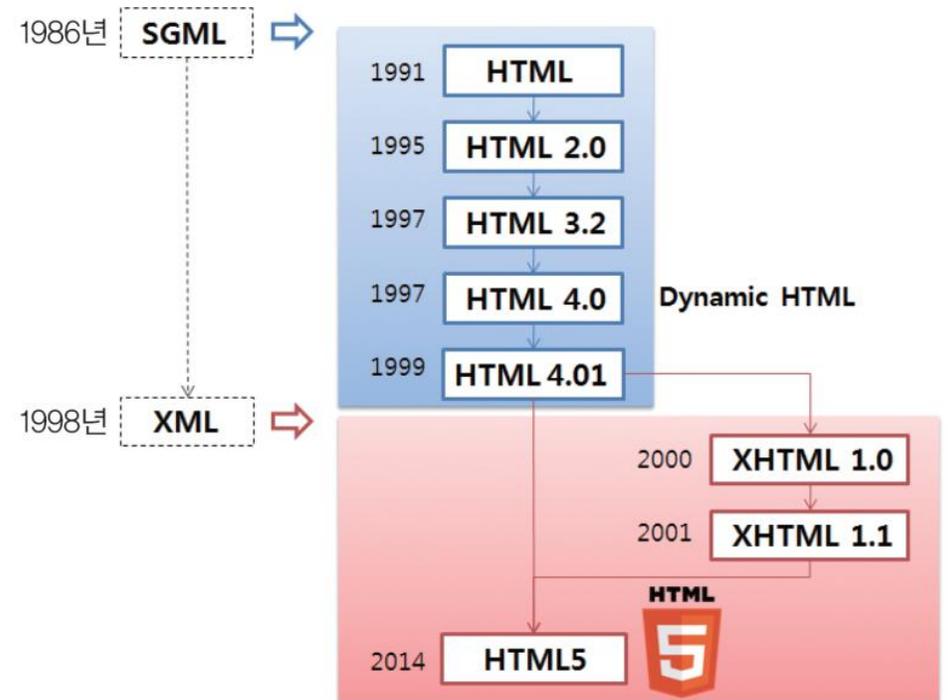
HTML의 역사_2

- 마크업 언어?
 - 태그 등을 이용하여 데이터의 구조를 명기하는 언어
 - SGML, HTML, XML, XHTML, SVG, MathML 등
- SGML(Standard Generalized Markup Language)
 - 다양한 전자문서의 구조/내용 기술을 위한 방법의 국제 표준
 - 1986년 ISO에서 국제표준으로 제정
 - 가장 큰 특징은 메타언어
 - 문서의 내용을 기술하는 것이 아니라 문서의 유형을 만드는 역할
 - HTML은 SGML로 정의된 문서 형식 중의 하나

HTML의 역사_3

■ HTML의 발전

- 간단하여 배우기 쉽고 사용 편리하여 인터넷의 대중화에 큰 기여
- 1994, HTML 2.0, 웹컨소시엄(W3C) 결성
- 1997, HTML 3.2 & HTML 4.0
 - 동적 HTML(Dynamic HTML) : HTML + CSS + JavaScript
- 1998, SGML을 간소화한 XML 제정
 - 문서나 자료의 교환이 위해 새로운 언어의 요구가 대두
 - 2000년 XHTML로 발전
- HTML5
 - 2006, 웹브라우저 개발사 WHATWG 결성
 - 2007, W3C와 WHATWG 공동 워킹그룹
 - 2014, HTML5 최종 표준안



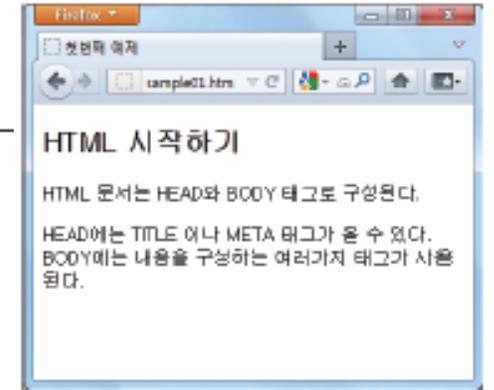
HTML 문서 구조와 텍스트 관련 태그_1

- HTML 문서의 요소(Element)
 - 시작태그, 내용, 끝태그로 구성
<태그이름 속성1="값" 속성2="값">문서의 내용</태그이름>

- HTML 문서의 구조

- <!DOCTYPE>
- <html>
- <head>
 - <title>,<meta> 등의 태그 포함
- <body>
 - 문서의 내용을 표현하는 부분

```
<!DOCTYPE html>
<html>
  <head>
    <title>첫번째 예제</title>
  </head>
  <body>
    <h2>HTML 시작하기</h2>
    <p>HTML 문서는 HEAD와 BODY 태그로 구성된다.</p>
    <p>HEAD에는 TITLE 이나 META 태그가 올 수 있다.<br>
    BODY에는 내용을 구성하는 여러가지 태그가 사용된다.
    </p>
    <!-- 주석문은 이렇게 작성 -->
  </body>
</html>
```

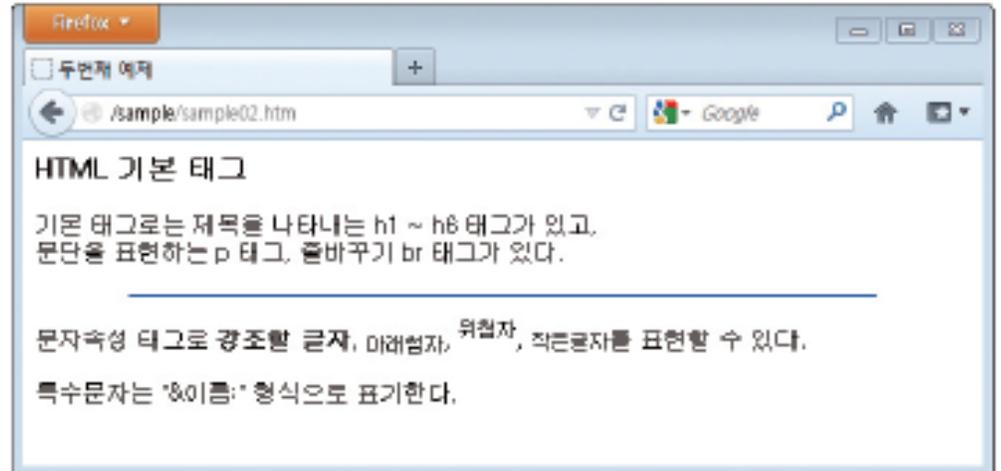


HTML 문서 구조와 텍스트 관련 태그_2

■ 텍스트 관련 태그

- 제목줄 <h1>~<h6>, 문단 <p>, 줄바꾸기
, 수평선 <hr ...>
- 문자속성 , <sub>, <sup>, <small>
- 특수문자 &이름;
 - 예, & < > “ (공백) 은 & < > " 로 표기

```
<html>
<head>
  <title>두번째 예제</title>
</head>
<body>
  <h3>HTML 기본 태그</h3>
  <p>기본 태그로는 제목을 나타내는 h1 ~ h6 태그가 있고,<br>
  문단을 표현하는 p 태그, 줄바꾸기 br 태그가 있다.</p>
  <hr width="80%" color="blue" align="center">
  <p>문자속성 태그로 <strong>강조할 글자</strong>, <sub>아래첨자</sub>,
  <sup>위첨자</sup>, <small>작은글자</small>를 표현할 수 있다.</p>
  <p>특수문자는 &quot;&amp;이름;&quot; 형식으로 표기한다.</p>
</body>
</html>
```



하이퍼링크 및 멀티미디어 포함하기_1

- 하이퍼링크 태그

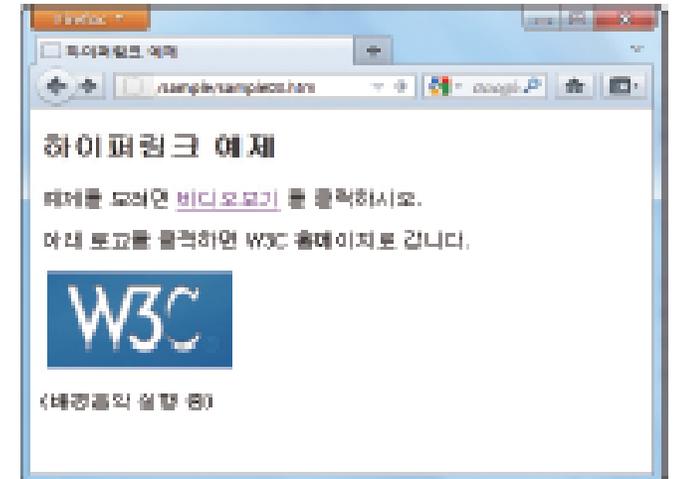
- ` ... `

- 멀티미디어 관련 태그

- `` : 이미지
- `<embed src="비디오/오디오 파일 주소">` : 비디오/오디오 파일
- `<video controls src="비디오 파일 주소">` : HTML5 비디오 파일
- `<audio controls src="오디오 파일 주소">` : HTML5 오디오 파일

하이퍼링크 및 멀티미디어 포함하기_2

```
<html>
  <head> <title>하이퍼링크 예제</title> </head>
  <body>
    <h2>하이퍼링크 예제</h2>
    <p>예제를 보려면 <a href="sample04.htm">비디오보기</a>를 클릭하십시오.</p>
    <p>아래 로고를 클릭하면 W3C 홈페이지로 갑니다.</p>
    <p>
      <a href="http://www.w3.org/">
        
      </a>
    </p>
    <embed src="starry_night.mp3" hidden="true">
      (배경음악 실행 중)
    </body>
</html>
```



하이퍼링크 및 멀티미디어 포함하기_3

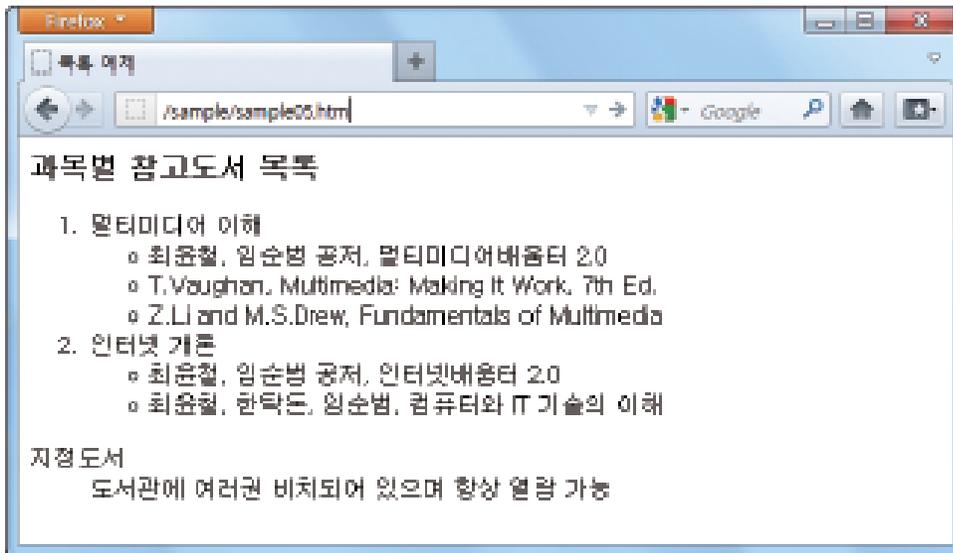
```
<html>
<head> <title>비디오 예제</title> </head>
<body bgcolor="#FFFF99">
  <h2>비디오 예제</h2>
  <video controls
    src="http://media.w3.org/2010/05/bunny/movie.ogv" width="400">
    웹브라우저가 HTML5 Video 태그를 지원하지 않습니다.
  </video>
  <br>원본출처: http://www.w3.org/wiki/HTML/Elements/video
</body>
</html>
```



다양한 형식의 내용 구성하기_1

■ 목록(List)

- 순서없는(Unordered) 목록: ... ,
- 순서있는(Ordered) 목록: ... ,
- 정의 목록(Definition List)
 - <dl> ... </dl>
 - <dt>...</dt>, <dd>...</dd>



```
<html>
<head> <title>목록 예제</title> </head>
<body>
<h3>과목별 참고도서 목록</h3>
<ol>
<i>멀티미디어 이해</i>
<ul>
<i>최윤철, 임순범 공저, 멀티미디어배움터 2.0</i>
<i>T.Vaughan, Multimedia: Making It Work, 7th Ed.</i>
<i>Z.Li and M.S.Drew, Fundamentals of Multimedia</i>
</ul>
<i>인터넷 개론</i>
<ul>
<i>최윤철, 임순범 공저, 인터넷배움터 2.0</i>
<i>최윤철, 한탁돈, 임순범, 컴퓨터와 IT 기술의 이해</i>
</ul>
</ol>
<dl>
<dt>지정도서</dt><dd>도서관에 여러권 비치되어 있으며 항상 열람 가능</dd>
</dl>
</body>
</html>
```

다양한 형식의 내용 구성하기_2

■ 테이블

- <table> 태그
- 각 줄: <tr>...</tr>
- 각 칸: <td>...</td>
- 칸 합치기: rowspan, colspan
- 제목/주석 <caption>

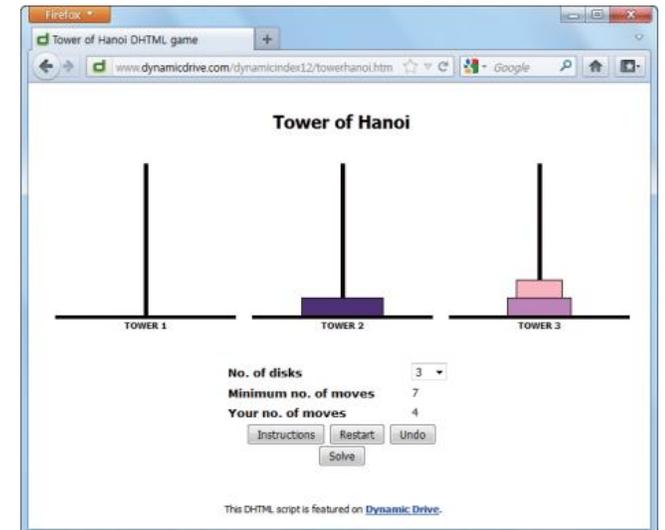
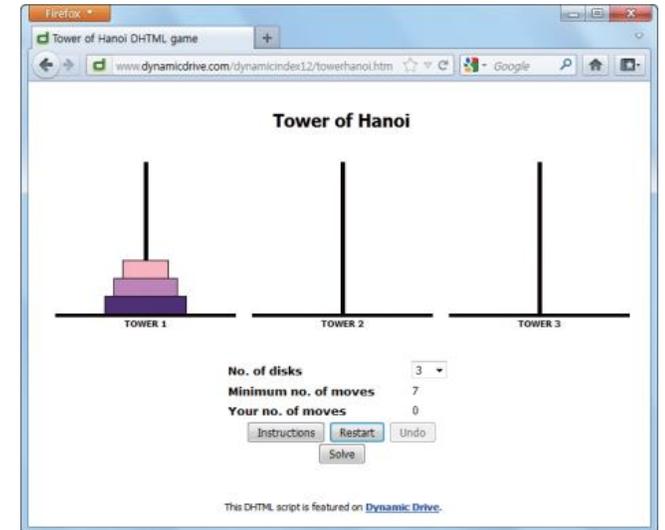
과목명	도서명
멀티미디어 이해	최윤철, 임순범 공저, 멀티미디어배움터 2.0
	T.Vaughan, Multimedia: Making It Work, 7th Ed.
	Z.Li and M.S.Drew, Fundamentals of Multimedia
인터넷 개론	최윤철, 임순범 공저, 인터넷배움터 2.0
	최윤철, 한탁돈, 임순범, 컴퓨터와 IT 기술의 이해

* 지정 도서는 도서관에 비치되어 있으며 항상 열람 가능

```
<html>
  <body>
    <h3>과목별 참고도서 목록</h3>
    <table border="1">
      <tr><th>과목명</th><th>도서명</th></tr>
      <tr><td rowspan="3">멀티미디어 이해</td>
        <td>최윤철, 임순범 공저, 멀티미디어배움터 2.0</td></tr>
      <tr><td>T.Vaughan, Multimedia: Making It Work, 7th Ed.</td></tr>
      <tr><td>Z.Li and M.S.Drew, Fundamentals of Multimedia</td></tr>
      <tr><td rowspan="2">인터넷 개론</td>
        <td>최윤철, 임순범 공저, 인터넷배움터 2.0</td></tr>
      <tr><td>최윤철, 한탁돈, 임순범, 컴퓨터와 IT 기술의 이해</td></tr>
      <caption align="bottom"><p align="left">* 지정도서는 도서관에 여러권 비치되어 있으며 항상 열람 가능</p></caption>
    </table>
  </body>
</html>
```

동적 HTML의 특징_1

- 동적 상호작용
 - 서버와의 상호작용
 - PHP, JSP, ASP, CGI 등을 이용
 - 사용자의 입력 => 서버에서 처리 후 => 다시 사용자에게 전송
 - 상당한 시간과 서버의 부담
 - 브라우저에서 간단한 상호작용
 - 동적 HTML : 웹브라우저 내에서 자체적으로 실행
 - 서버에 부담이 없고 이벤트에 즉각적인 반응
- 예제
 - 하노이탑 게임



동적 HTML의 특징_2



```
<html>
<head>
  <style type="text/css">
    h2 { display: block; border-style:double; border-color:red; padding: 5px; width: 70% }
    b { font-size: small; color: blue; }
    #example1 { display: block; margin: 3%; padding: 5%; background-color: limegreen }
  </style>
  <script type="text/javascript">
    function bgChange(bg) {
      ex=document.getElementById("example1");
      ex.style.background = bg;
      cn=document.getElementById("cname");
      cn.innerHTML= bg+" 선택";
    }
  </script>
</head>
<body>
  <h2>동적 HTML 예제</h2>
  <b>원하는 색상을 마우스로 클릭하시오</b>
  <table width="150" height="50" cellspacing="6">
    <tr align="center">
      <td onClick="bgChange('cyan') bgcolor="cyan">C</td>
      <td onClick="bgChange('magenta') bgcolor="magenta">M</td>
      <td onClick="bgChange('yellow') bgcolor="yellow">Y</td>
    </tr>
  </table>
  <div id="example1">
    CSS + JavaScript + DOM 예제
    <p> ... 배경 색상 보기 ... <span id="cname"> </span>
  </div>
</body>
</html>
```

스타일시트의 개념 및 CSS_1

- 스타일시트(Stylesheet)

- 웹문서 외형의 스타일을 좀 더 손쉽게 빠르게 설정하거나, 일반적인 HTML 태그로는 외형을 지정할 수 없었던 부분
- 웹문서 구성요소에 크기, 색상, 글자간격, 문단간격, 위치 등 지정

- CSS(Cascading Style Sheet)

- 기능의 복잡도에 따라 레벨1, 레벨2, 레벨3
- CSS3는 2005년 이후 모듈별로 구분하여 작성, 현재 개발 중
- CSS 스타일 정의
 - 선택자 스타일속성 : 값 ; 스타일속성 : 값; ...

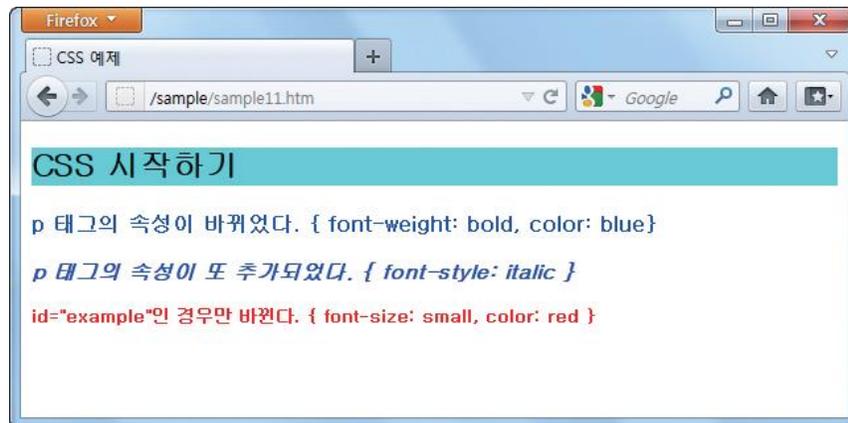
예) `h2 { display: block; border-style: double; border-color: red; }`

- 선택자에는 일반 태그, 혹은 id 이름(#로 시작)

스타일시트의 개념 및 CSS_2

■ CSS 정의 방법

- <style>태그 내
 - 선택자에 태그, 혹은 id (#이름)
- 인라인 스타일
 - 태그 내에 style 속성
- 외부 CSS 파일 연결
 - <link> 태그 이용



```
<html>
  <head>
    <style type="text/css">
      h2 { display: block; background-color: cyan}
      p { font-weight: bold; color: blue;}
      #example { font-size: small; color: red; }
    </style>
  </head>
  <body>
    <h2>CSS 시작하기</h2>
    <p>p 태그의 속성이 바뀌었다. { font-weight: bold, color: blue}</p>
    <p style="font-style:italic">p 태그의 속성이 또 추가되었다. {font-style: italic} </p>
    <p id="example">id="example"인 경우만 바뀐다. {size: small, color: red}</p>
  </body>
</html>
```

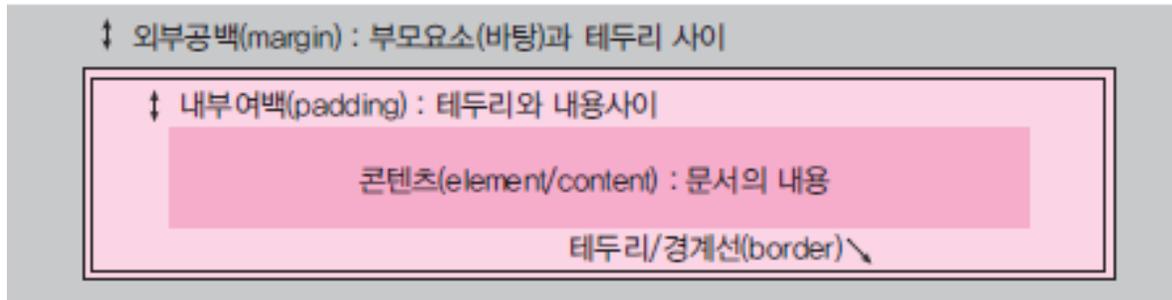
스타일시트의 개념 및 CSS_3

- 자주 사용되는 CSS 스타일 속성
 - 폰트 관련 속성: font-family, font-size, font-style, font-weight, ...
 - 텍스트 관련 속성: letter-spacing, word-spacing, vertical-align, text-align, text-indent, text-transform, text-decoration
 - 배경/색상 관련: color, background-color, background-image
 - 목록 관련: list-style-type, list-style-position, list-style-image
 - 박스모델 관련 : margin, padding, border-style, border-width, ...
- display 속성
 - block : 줄 바꾼다
 - in-line : 줄 안바꾼다
 - list-item : 목록 형태
 - none : 안 보여준다

스타일시트의 개념 및 CSS_4

■ 박스모델

- 외부공백(margin) 및 내부여백(padding)을 조절 가능
- 경계선(border) 지정도 가능



■ 레이어(Layer) 개념

- 웹문서의 구성요소 위치를 자유롭게 지정하기 위하여
- <div>, 요소에 id 혹은 class 이름 설정하여 스타일 지정
 - <div>요소는 원하는 부분을 블록으로
 - 요소는 인라인 형태로 설정

스타일시트의 개념 및 CSS_5

```
<html>
<head>
  <style type="text/css">
    #ex
    { position:absolute; left:40px; top:80px;
      width:440px; height:80px; border:2px solid; border-radius:25px; padding:6px;
      box-shadow: 10px 10px 2px #888888; background-color:yellow;
      transform: rotate(-20deg); column-count:2;
    }
    #word2
    { text-shadow: 3px 3px 5px #FF0000; }
  </style>
</head>
<body>
  <div id="ex">&nbsp;&nbsp;&nbsp;<b>CSS3 예제</b>
  <ul><li>경계선 라운드 처리</li>
    <li>배경 그림자 효과</li>
    <li>멀티 컬럼</li>
    <li>2D 변환: -20도 회전</li>
    <li><span id="word2">텍스트 그림자</span> 효과</li>
    <li>애니메이션도 가능</li>
  </ul>
</div>
</body>
</html>
```



CSS3에서 다양한 기능 추가

- 앞의 예제: 박스의 모서리 처리, 그림자 효과, 다단(Multi-column) 지정, 2D/3D 기하 변환, 텍스트의 그림자 등 효과 설정
- 박스 관련 속성: border-radius, box-shadow, border-image
- 폰트 지정 규칙: @font-face
- 문자 관련 속성: text-shadow, word-wrap, word-break, text-outline, text-justify
- 배경: background-size, background-origin, background-clip
- 2D/3D 변환 속성: translate(), rotate(), scale(), skew(), matrix()
- 다단 관련 속성: column-count, column-gap, column-rule, column-width
- 이 외 장면 변환(Transition), 애니메이션, 사용자 인터페이스 속성

자바스크립트 및 DOM의 작동 원리_1

- 자바스크립트의 역할

- 웹페이지 상에서 모든 사건을 처리 및 모든 문서객체의 제어
 - HTML 문서와 사용자 사이의 상호작용을 가능하게 하며
 - 멀티미디어 자료와 애니메이션을 효과적으로 표현
- 웹브라우저에서 실행

- 자바스크립트 언어

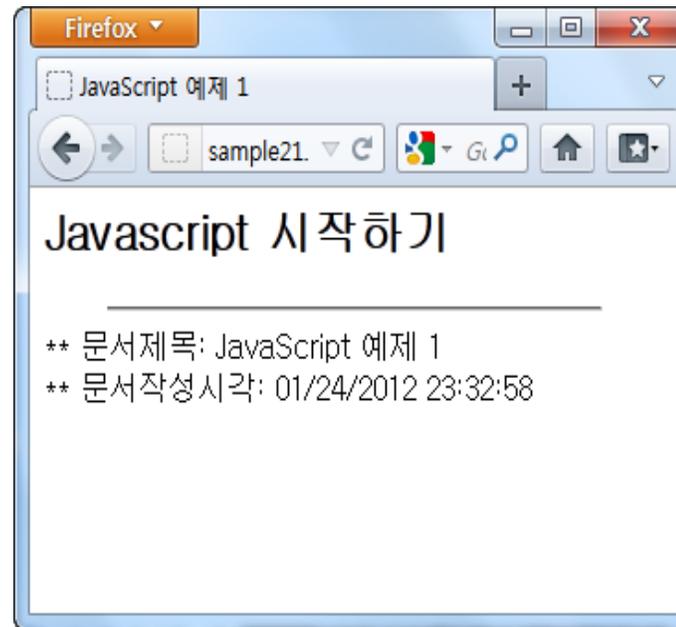
- 객체기반의(Object-based) 스크립트 언어
- 넷스케이프사가 개발, 유럽 표준은 ECMAScript라는 명칭
- C 및 자바 언어와 문법은 유사하지만 성격이 다른 스크립트 언어
- 윈도우, Mac과 같이 서로 다른 플랫폼에서도 동일한 실행이 가능

자바스크립트 및 DOM의 작동 원리_2

- 자바스크립트 코드 작성

- `<script type="text/javascript">` 자바스크립트 코드 `</script>`

```
<html>
  <head>
    <title>JavaScript 예제 1</title>
  </head>
  <body>
    <h2>Javascript 시작하기</h2>
    <script type="text/javascript">
      document.write("<HR width=80%>");
      document.write("** 문서제목: ");
      document.write(document.title + "<BR>");
      document.write("** 문서작성시각: ");
      document.write(document.lastModified);
    </script>
  </body>
</html>
```



자바스크립트 및 DOM의 작동 원리_3

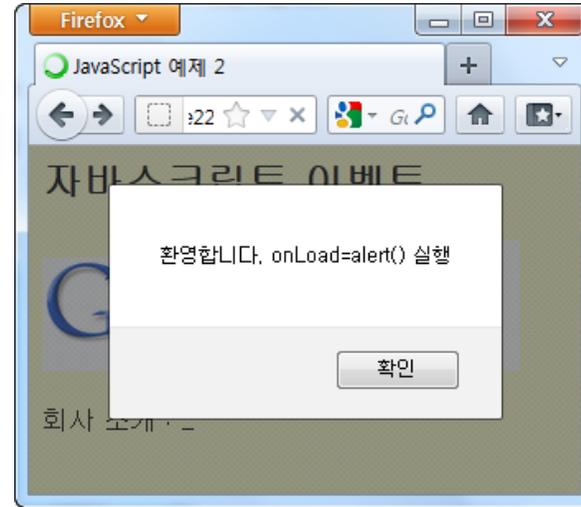
■ 자바스크립트 언어 문법

- 변수의 선언이 간단
 - 데이터 유형 자동 결정
- 연산자
 - (산술)+, -, *, /, ++, --, % (논리)&&, ||, ! (비교)==, !=, >, >=, <, <=
- 제어문 및 객체 제어
 - if 문, switch 문, for 문, while 문, for...in 문, with 문, new 연산자
- 이벤트 프로그램
 - 이벤트 종류: load, unload, click, mouseover, mouseout, error 등
 - <태그 이벤트처리기="자바스크립트 함수 혹은 프로그램">
예) <td onClick="bgChange('cyan')" bgcolor="cyan">C</td>
- 함수
 - 호출되기 이전에 정의

```
var char3 = '3', num2 = 2;  
var result1 = num2 + char3 // result1 = 5, plus 연산  
var result2 = char3 + num2; // result2 = '3'+2' = '32', string연산 (Concatenation)
```

자바스크립트 및 DOM의 작동 원리_4

```
<html>
<head>
  <title>JavaScript 예제 2</title>
  <script type="text/javascript">
    function writeAddr(obj, num) {
      switch (num) {
        case 1: addr="구글 http://www.google.com"; break;
        case 2: addr="애플 http://www.apple.com"; break;
        default: addr = "_";
      }
      ex=document.getElementById("intro"); ex.innerHTML = addr;
      obj.border = "3";
    }
    function clearAddr(obj) {
      ex=document.getElementById("intro"); ex.innerHTML = "_";
      obj.border = "0";
    }
  </script>
</head>
<body bgcolor="#FFFFCC" onLoad="alert('환영합니다.');">
  <h2>자바스크립트 이벤트</h2>
  <p>
    
    </p>
  <p>회사 소개 : <span id="intro">_</span></p>
</body>
</html>
```



자바스크립트 및 DOM의 작동 원리_5

■ 자바스크립트 객체(Object)

- 각 객체는 다양한 속성(Property)과 메소드(Method) 보유
- 시스템 내장 객체
 - document, window, history, location, Array, Math, Form, String 등
- 사용자 정의 객체
 - 사용자가 웹문서에서 정의한 모든 요소, id 지정하여 접근 가능

■ 문서객체모델(Document Object Model, DOM)

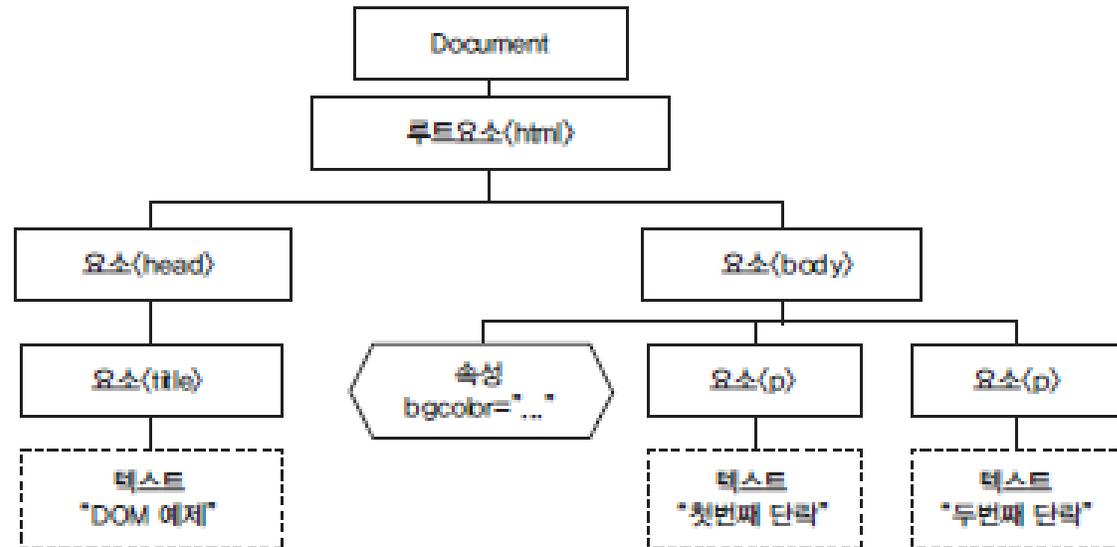
- 문서 내 모든 객체에 접근하고 수정할 수 있도록 API 규정
- HTML 문서 및 XML 문서에 적용 가능
- 자바스크립트 혹은 C/C++언어나 자바 등 다양한 언어 사용 가능
- 웹문서를 하나의 트리 형태로 표현하여 모든 객체를 노드로 표현

자바스크립트 및 DOM의 작동 원리_6

■ DOM 트리

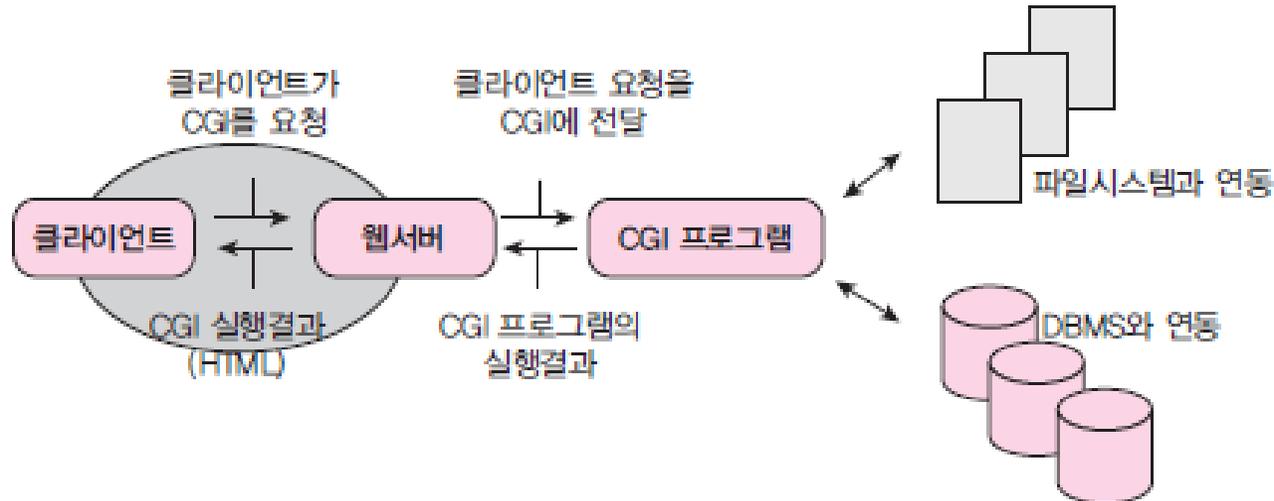
- 트리 운행 방식에 따라 모든 노드를 접근할 수 있도록 여러가지 속성 및 함수 API 제공
- 예제문서에서, 첫 번째 <p>단락에 접근하려면
 - document.documentElement.lastChild.firstChild

```
<html>
  <head>
    <title>DOM 예제</title>
  </head>
  <body bgcolor="yellow">
    <p>첫번째 단락</p>
    <p>두번째 단락</p>
  </body>
</html>
```



CGI 원리와 Form 요소_1

- CGI(Common Gateway Interface)
 - 서버에서 정보를 검색하거나 보안이 필요한 웹페이지 경우 필요
 - 웹브라우저, 웹서버, 응용프로그램 간의 인터페이스
 - 사용자가 입력하면 브라우저는 웹서버에 CGI를 통해 요청
 - 서버는 응용 프로그램으로 전달하여 정보를 처리
 - 처리결과는 HTML 페이지로 구성하여 사용자 웹브라우저로 전달



CGI 원리와 Form 요소_2

■ <form> 요소

- HTML 문서에서 서버 프로그램으로 데이터를 전달하는 데 사용

<form action="서버프로그램 주소" method="GET 또는 POST"> ...입력양식... </form>

- GET은 CGI 환경변수를 통해, POST는 표준 입력방법으로 데이터 전달

■ 입력 형식

<input type="입력필드형식" name="입력필드이름" value="초기값">

- 입력필드 형식: text(텍스트), password(암호), radio(라디오 버튼), checkbox(체크박스 선택), submit(전송버튼), reset(취소버튼), button(일반 버튼)

<textarea name="입력필드이름" rows="라인수" cols="폭">: 긴 텍스트

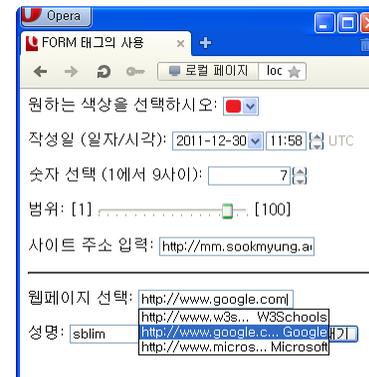
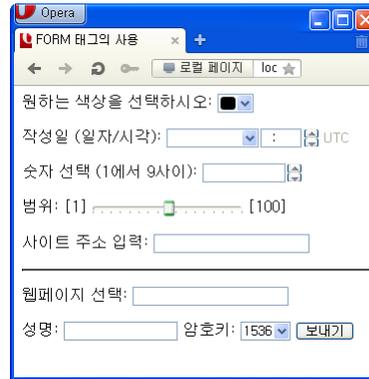
<select name="" size=""> <option>...항목...</option> ... </select>: 선택목록

CGI 원리와 Form 요소_4

- HTML5에서 개선된 입력 기능
 - 새로 추가된 form 입력 유형
 - type = datetime, date, month, week, time: 달력에서 날짜/시간 입력
 - type = number, range: 제한된 범위의 숫자로 입력, 스크롤바로 입력
 - type = color, file, search: 색상표 색상 선택, 파일보기에서 파일 선택
 - type = tel, url, email: 데이터의 형식이 맞는지 유효성 확인
 - <datalist> 요소
 - 스크롤 목록이 아닌 펼침 목록을 작성
 - <keygen> 요소
 - 암호키를 제공

CGI 원리와 Form 요소_5

```
<html>
<head> <title>HTML5 input 태그</title></head>
<body>
  <form method="get">
    <p>원하는 색상을 선택하시오: <input type="color" name="mycolor" /> </p>
    <p>작성일 (일자/시각): <input type="datetime" name="mydaytime" /> </p>
    <p>숫자선택(1에서 9사이): <input type="number" name="mynum" min="1" max="9"/></p>
    <p>범위: [1]<input type="range" name="point" min="1" max="100" />[100] </p>
    <p>사이트 주소 입력: <input type="url" name="addr" /> </p>
    <hr>
    <p>웹페이지 선택: <input type="url" list="url_list" name="link" /> </p>
    <datalist id="url_list">
      <option label="W3Schools" value="http://www.w3schools.com" />
      <option label="Google" value="http://www.google.com" />
      <option label="Microsoft" value="http://www.microsoft.com" />
    </datalist>
    <p>성명: <input type="text" name="myname" size="14"/>
      암호키: <keygen name="security" />
    <input type="submit" /> </p>
  </form>
</body>
</html>
```



CGI 프로그래밍_1

- CGI 프로그램

- 웹브라우저로부터의 사용자 요구를 받아들여 웹서버와 응용프로그램을 연결하는 프로그램
- Perl, C/C++, Python 등의 언어를 많이 사용

- HTML 문서에서 CGI 실행 방법

- <form> 요소에서 submit 버튼으로 전송
- <a> 요소의 href 속성을 이용
 - ...

예, ...

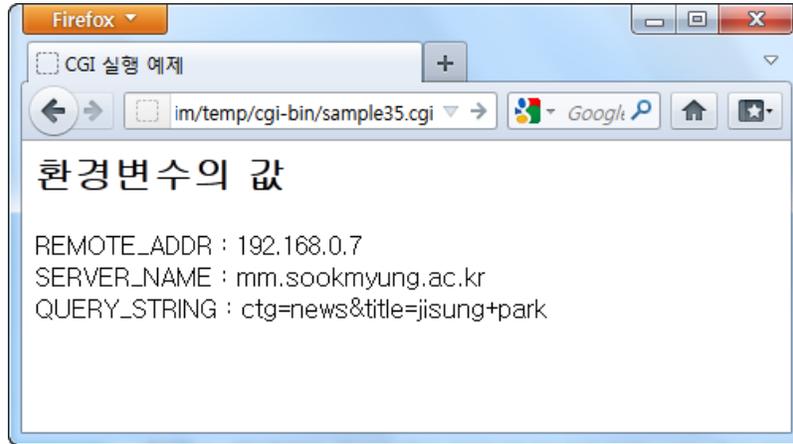
...

- 서버프로그램주소? 다음에 name=value 형식, 2개 이상은 &로 연결, 공백은 +로, 한글은 %뒤 16진수로 표현

CGI 프로그래밍_2

- CGI 프로그램 실행결과 전송
 - 우선 헤더정보 출력 “Content-type: text/html”
 - 해당 결과를 <html> 파일 형식으로 출력
- CGI 환경변수
 - CGI 프로그램 내에서 전달인자 전송 혹은 필요한 정보 수집
 - 환경변수 예
 - 사용자 IP 주소(REMOTE_ADDR), 서버 도메인 이름(SERVER_NAME), CGI 프로그램 이름(SCRIPT_NAME), 전달방식(REQUEST_METHOD), 데이터 길이(CONTENT_LENGTH), 전달 문자열(QUERY_STRING) 등

CGI 프로그래밍_3



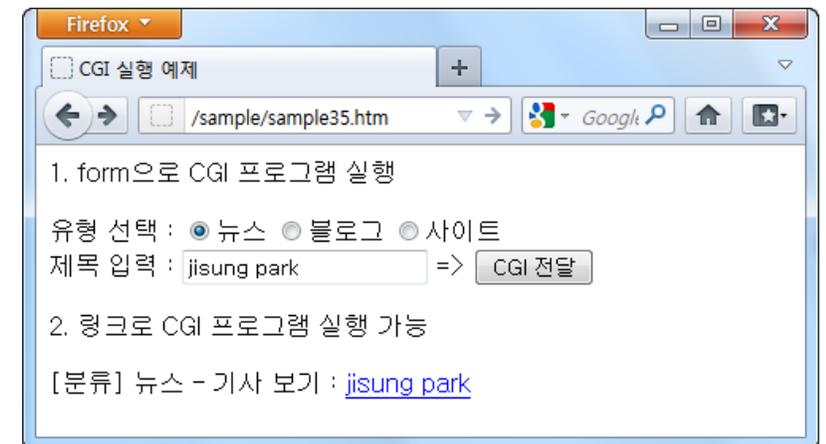
CGI 프로그램: sample35.c

▶ 컴파일 방법: `cc[gcc] -o sample35.cgi sample35.c`

```
#include <stdio.h>

main()
{
    printf("Content-type: text/html\n\n");
    printf("<html>");
    printf("<body>");
    printf("<h2>환경변수의 값</h2>");
    printf("REMOTE_ADDR : %s", getenv("REMOTE_ADDR"));
    printf("SERVER_NAME : %s", getenv("SERVER_NAME"));
    printf("QUERY_STRING : %s", getenv("QUERY_STRING"));
    printf("</body>");
    printf("</html>");
}
```

```
<html>
<head> <title>CGI 실행 예제</title> </head>
<body>
<p>1. form으로 CGI 프로그램 실행</p>
<form action="cgi-bin/sample35.cgi" method="get">
    유형 선택 : <input type="radio" name="ctg" value="news">뉴스
                <input type="radio" name="ctg" value="blog">블로그
                <input type="radio" name="ctg" value="site">사이트 <br>
    제목 입력 : <input type="text" name="title" value=""> <input type="submit" value="CGI 전달">
</form>
<p>2. 링크로 CGI 프로그램 실행 가능</p>
[분류] 뉴스 - 기사 보기 :
    <a href="cgi-bin/sample35.cgi?ctg=news&title=jisung+park">jisung park</a>
</body>
</html>
```

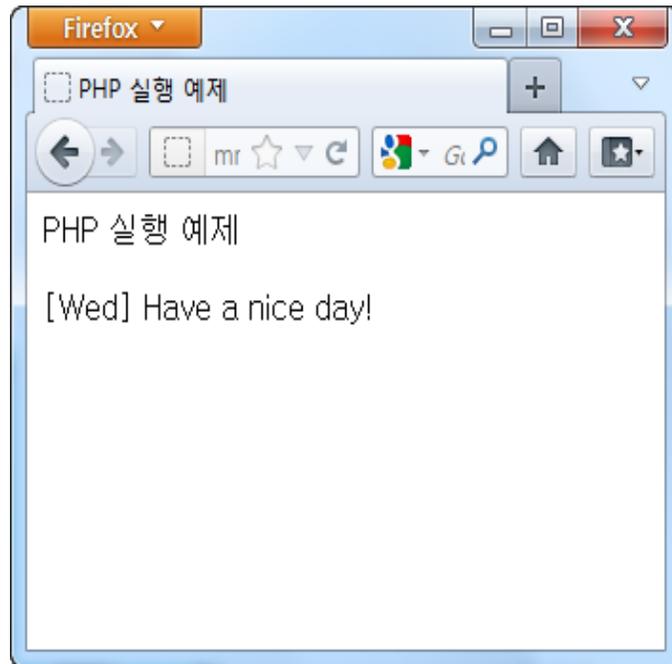


웹서버 프로그램 PHP_1

- PHP(Professional Hypertext Preprocessor)
 - 서버측에서 실행되는 프로그램
 - HTML 문서안에 PHP 스크립트 삽입, PHP 스크립트가 서버에서 처리
 - 그 결과물인 HTML 태그만이 클라이언트에서 보여줌
 - 사용자는 소스코드에 접근 못하므로 소스코드의 보안성 보장
 - Apache 서버에서 사용
 - 다양한 데이터베이스 서버에 대해 사용하기 간편한 인터페이스 제공
 - LAMP 환경: Linux 환경, Apache 서버, PHP와 MySQL을 연동
 - PHP 스크립트 코드
 - HTML 문서 내에서 `<?php 와 ?>` 사이에 위치
 - 스크립트 코드들은 서버에서 실행된 후 결과가 HTML 문서 내에 포함
 - 변수는 \$로 시작, 배열은 Array 함수를 이용, 출력은 echo 함수 이용

웹서버 프로그램 PHP_2

- <form> 입력 값의 사용
 - GET 메소드의 경우 : \$_GET["변수명"]
 - POST 메소드의 경우 : \$_POST["변수명"]



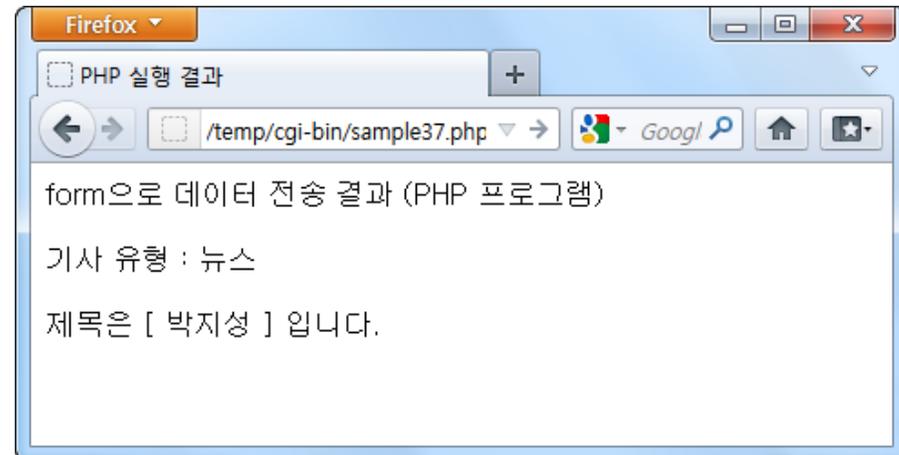
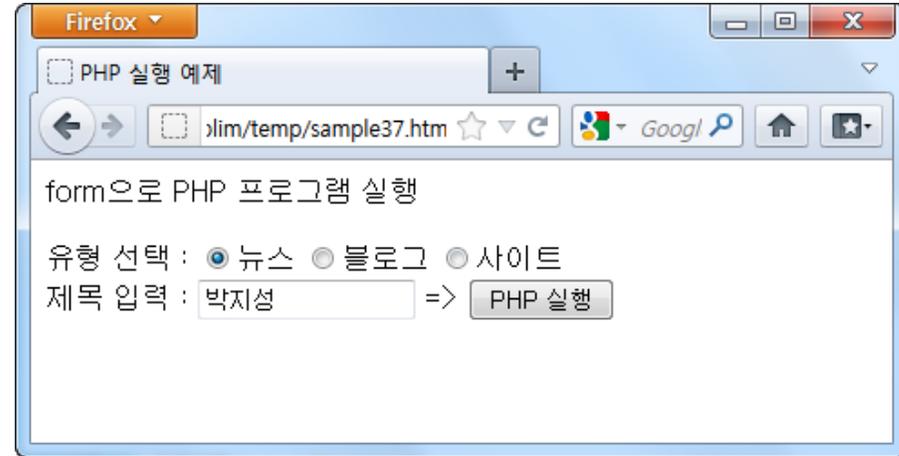
```
<html>
  <head> <title>PHP 실행 예제</title> </head>
  <body>
    <p>PHP 실행 예제</p>
    <?php
      $week=date("D");
      if ($week=="Fri")
        echo "[ $week ] Have a nice weekend!";
      elseif ($week=="Sun")
        echo "[ $week ] Have a nice Sunday!";
      else
        echo "[ $week ] Have a nice day!";
    ?>
  </body>
</html>
```

웹서버 프로그램 PHP_3

```
<html>
<head> <title>PHP 실행 예제</title> </head>
<body>
  <p> form으로 PHP 프로그램 실행</p>
  <form action="cgi-bin/sample37.php" method="GET">
    유형 선택 : <input type="radio" name="ctg" value="news">뉴스
              <input type="radio" name="ctg" value="blog">블로그
              <input type="radio" name="ctg" value="site">사이트 <br>
    제목 입력 : <input type="text" name="title" size="15">=&gt; <input type="submit"
value="PHP 실행">
  </form>
</body>
```

sample37.php

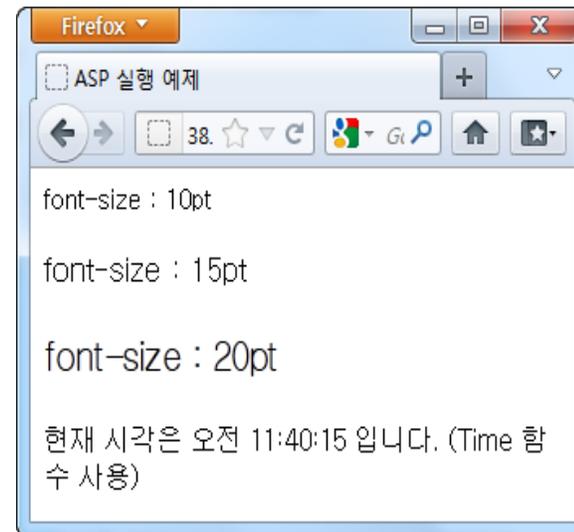
```
<html>
<head> <title>PHP 실행 결과</title> </head>
<body>
  <p>form으로 데이터 전송 결과 (PHP 프로그램)</p>
  <p>기사 유형 : <?php echo $_GET["ctg"]; ?> </p>
  <p>제목은 [ <?php echo $_GET["title"]; ?> ] 입니다.</p>
</body>
</html>
```



웹서버 프로그램 ASP와 JSP_1

- ASP(Active Server Page)
 - 윈도우 기반 웹서버 IIS(Internet Information Server) 필요
 - 비주얼베이직 스크립트(VBScript)나 자바스크립트 사용
 - HTML 문서 내에서 <% 와 %> 태그 사이에 위치
 - 서버에서 처리되어 HTML 태그로 변환 후 사용자에게 보여지는 부분
 - 출력은 Response.write로 수행, 혹은 축약형으로 <%= 표기 사용

```
<html>
<head> <title>ASP 실행 예제</title> </head>
<body>
  <% for i=1 to 3 %>
    <% psize = 10+i*4 %>
    <p style="font-size: <%= psize %>">
      font-size : <%= psize %>pt </p>
  <% next %>
  현재 시각은 <%= Time %>입니다. (Time 함수 사용)
</body>
</html>
```

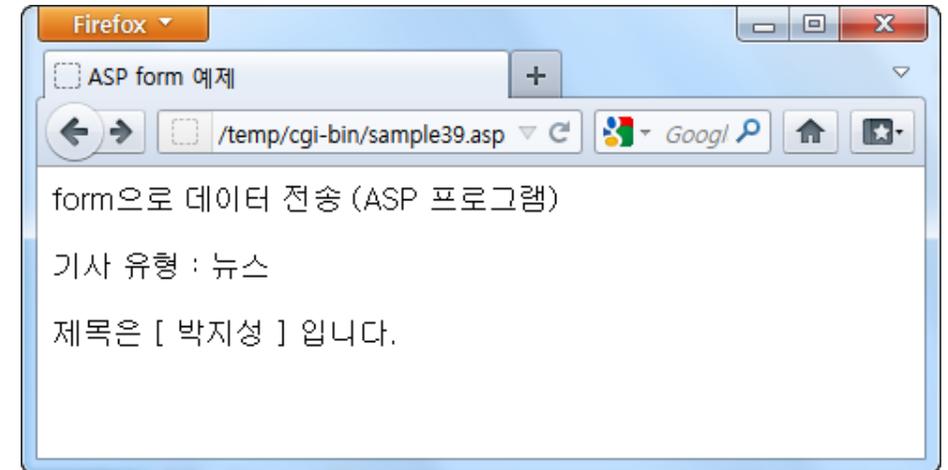
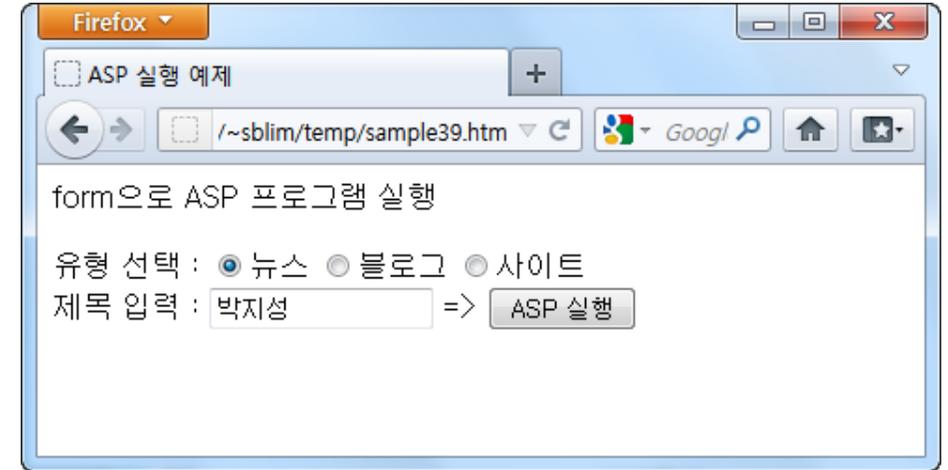


웹서버 프로그램 ASP와 JSP_2

```
<html>
<body>
<p> form으로 ASP 프로그램 실행</p>
<form action="cgi-bin/sample39.asp" method="GET">
  유형 선택 : <input type="radio" name="ctg" value="news">뉴스
             <input type="radio" name="ctg" value="blog">블로그
             <input type="radio" name="ctg" value="site">사이트 <br>
  제목 입력 : <input type="text" name="title" size="15">=&gt; <input type="submit"
value="ASP 실행">
</form>
```

sample39.asp

```
<html>
<head> <title>ASP form 예제</title> </head>
<body>
<p> form으로 데이터 전송 (ASP 프로그램)</p>
<p>기사 유형 : <%Request.form("ctg")%> </p>
<p>제목은 [ <%Request.form("dept")%> ] 입니다.</p>
</body>
</html>
```



웹서버 프로그램 ASP와 JSP_3

- JSP(Java Server Page)
 - ASP와 매우 유사한 점이 많은 서버측 스크립트
 - JSWDK(Java Server Web Development Kit)라는 자바 서버 설치하여 사용
 - 기존의 자바 프로그램과 연결이 가능
 - 자바 프로그램은 JDBC를 통해 데이터베이스를 액세스하고 SQL 문을 이용하여 데이터베이스를 검색

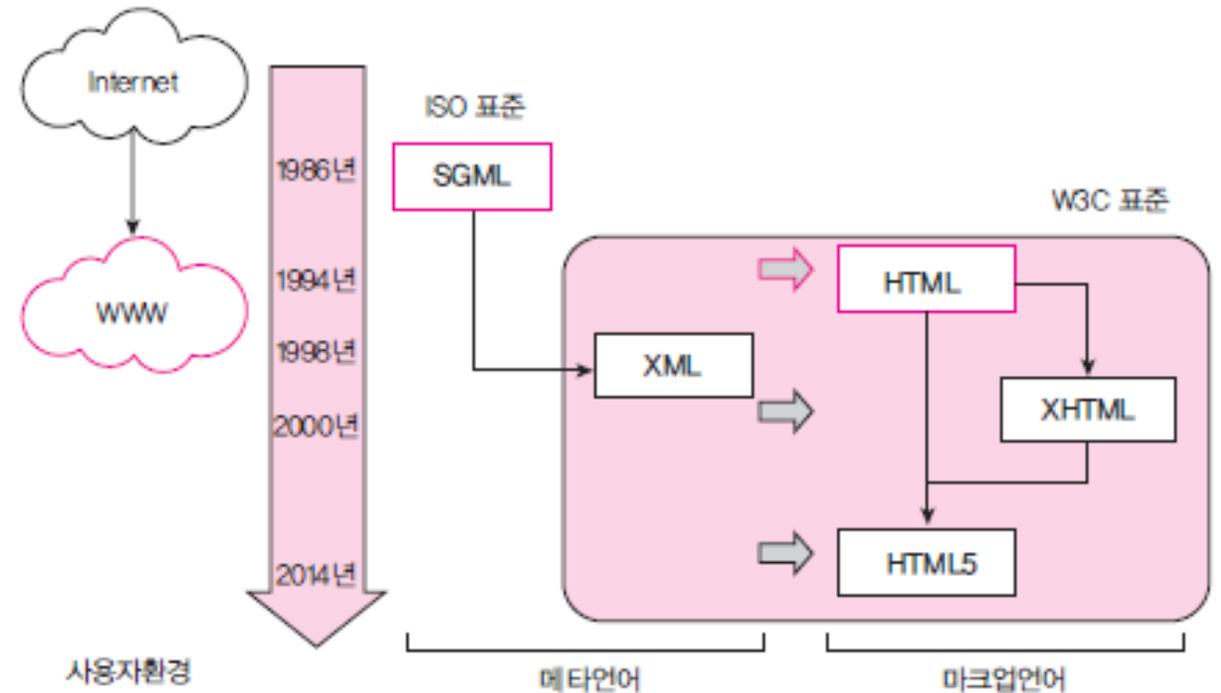
XML의 개요_1

- 새로운 마크업 언어의 필요성
 - 웹문서를 교환하거나 특히 원하는 정보만을 효율적으로 검색하기 위해서는 보다 강력한 태깅 언어가 필요
 - SGML(Standard Generalized Markup Language)의 후속 개념
- SGML의 특징
 - 마크업 언어를 정의하기 위한 메타언어(Meta Language)
 - 문서의 논리적 구조, 내용, 출력 스타일을 각각 분리하여 표현
- HTML
 - SGML로 태그를 정의하여 개발
 - 사용하기 간단하여 인터넷의 대중화에 큰 기여
 - 태그가 고정되어 기능이 제한적, 정교한 웹 페이지 출력 곤란

XML의 개요_2

■ XML(eXtensible Markup Language)의 출현

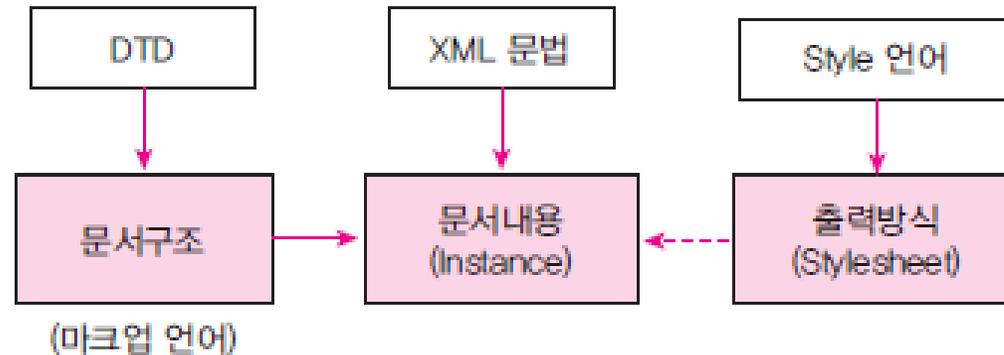
- SGML보다는 훨씬 간단하면서 SGML의 장점을 가진 마크업 언어
- HTML은 문서를 웹 환경에서 보여주기 위해, XML은 웹 환경에서 문서를 저장하고 원하는 정보를 검색/교환하기 위해
- XML의 발전 과정



XML 문서 작성 원리_1

- XML의 특징

- 문서의 구조 정의, 내용 작성, 스타일 지정을 분리하여 작성
 - 문서형 선언부(DTD: Data Type Definition)에서 구조 정의
 - 작성된 XML 문서: 실제문서(DI: Document Instance)
 - 출력 스타일: CSS(Cascading Style Sheet)나 XSL(eXtensible Stylesheet Language) 스타일시트 언어로 지정

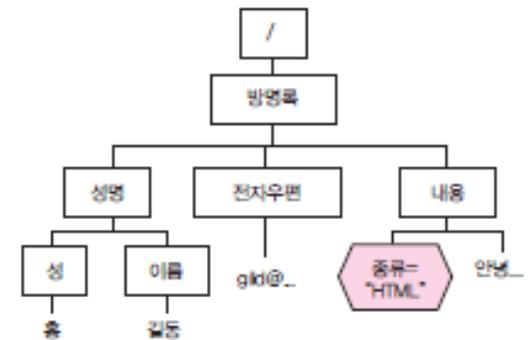


XML 문서 작성 원리_2

- DTD로 문서구조 정의하기
 - 문서 구조 정의 방법 : DTD, Schema, RelaxNG 등
 - 태그는 ELEMENT로 속성은 ATTLIST로 정의

```
<?xml version="1.0" encoding="EUC-KR"?>
<!DOCTYPE 방명록 [
  <!ELEMENT 방명록 (성명, 전자우편, 내용)>
  <!ELEMENT 성명 (성, 이름)>
  <!ELEMENT 성 (#PCDATA)>
  <!ELEMENT 이름 (#PCDATA)>
  <!ELEMENT 전자우편 (#PCDATA)>
  <!ELEMENT 내용 (#PCDATA)>
  <ATTLIST 내용 종류 (TEXT|HTML) "HTML"
]>
```

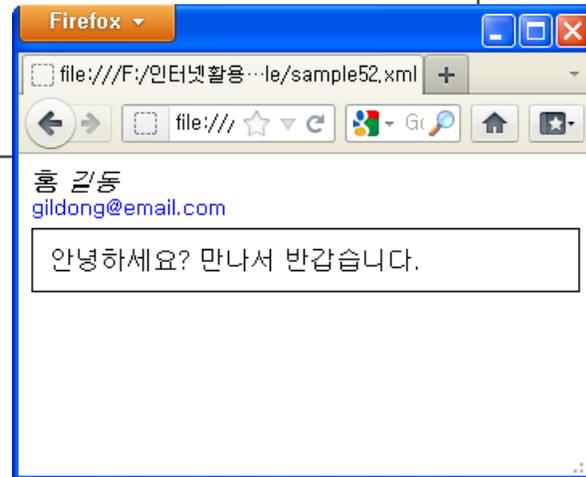
```
<?xml version="1.0" encoding="EUC-KR"?>
<방명록>
  <성명>
    <성>홍</성>
    <이름>길동</이름>
  </성명>
  <전자우편>gildong@email.com</전자우편>
  <내용 종류="HTML">안녕하세요? 만나서 반갑습니다.
  </내용>
</방명록>
```



XML 문서 작성 원리_3

- 문서의 출력을 위한 스타일시트 작성하기
 - XSL(eXtensible Stylesheet Language) 또는 CSS(Cascading Style Sheet)를 이용하여 문서의 스타일 지정

```
<?xml version="1.0" encoding="EUC-KR"?>
<?xml-stylesheet type="text/css" href="memo52.css"?>
<방명록>
  <성명> <성>홍</성> <이름>길동</이름> </성명>
  <전자우편>gildong@email.com</전자우편>
  <내용 종류="HTML">
    안녕하세요? 만나서 반갑습니다.
  </내용>
</방명록>
```

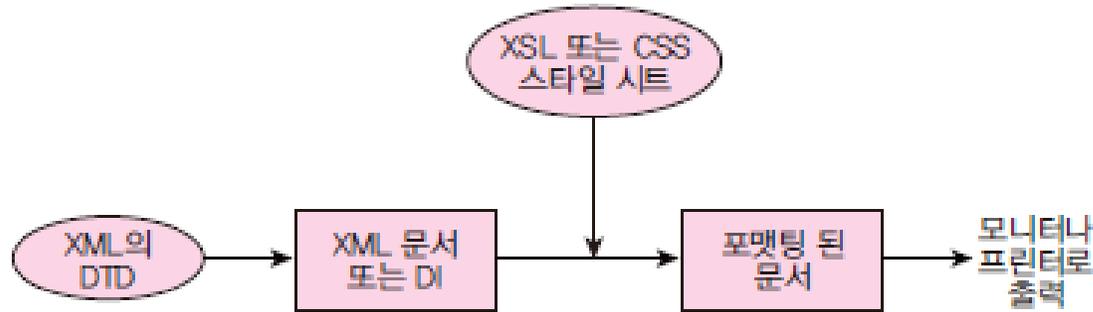


memo52.css

```
방명록 { display:block; margin: 8px; }
성명 { display:block; }
성 { font-weight:bold; }
이름 { font-style:italic; }
전자우편 { display:block; color:blue; font-size:small}
내용 { display:block; margin-top: 5px;
border-style: solid; border-width:1px; padding: 10px; }
```

XML 문서 작성 원리_4

- XML 문서의 출력과정



- 응용 프로그램 개발을 위한 API

- XML 문서를 접근, 변경, 처리를 원활히 수행하도록 API 제공
- 대표적인 API
 - DOM(Document Object Model)과 SAX(Simple API for XML) 방식

XML 문서의 활용_1

- 가장 활용되는 분야: 전자상거래와 전자문서
 - 전자상거래나 의미기반 검색 등에서의 데이터 교환 분야
 - 전자출판, 디지털 콘텐츠의 표현 및 교환을 위한 문서교환 분야
 - 최근 오피스 문서 docx, pptx

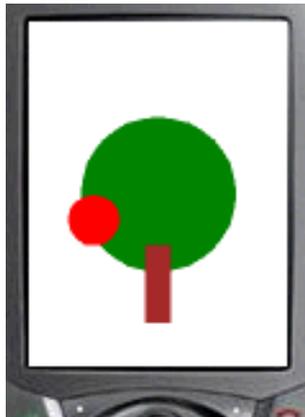
| | |
|--|---|
| <pre><html> <body> title : XML Bible (재고 12권) author : Gwyneth Paltrow pages : 652 price : (usd)39.99 title : XML 클래스 (재고 0권) author : 임순범 pages : 458 price : 19000원 ... </body> </html></pre> | <pre><?xml version="1.0" encoding="EUC-KR"?> <list> <book InStock="12"> <title>XML Bible</title> <author>Gwyneth Paltrow</author> <pages>652</pages> <price currency="usd">39.99</price> </book> <book InStock="0"> <title>XML 클래스</title> <author>임순범</author> <pages>458</pages> <price>19000</price> </book> ... </list></pre> |
|--|---|

XML 문서의 활용_2

- 새로운 마크업 언어 생성

- MathML : 수학기호 표현
- SVG(Scalable Vector Graphics) : 벡터 그래픽스
- SMIL(Synchronized Multimedia Integration Language) : 멀티미디어 정보의 동기화
- XHTML, HTML5

모바일 SVG 실행 예제



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1 Tiny//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11-tiny.dtd">

<svg width="100" height="100" version="1.1" baseProfile="tiny">
  <title>SVGT example</title>
  <circle cx="50" cy="50" r="30" fill="green" />
  <circle cx="25" cy="60" r="10" fill="red" />
  <rect x="45" y="70" width="10" height="30" fill="brown"/>
</svg>
```

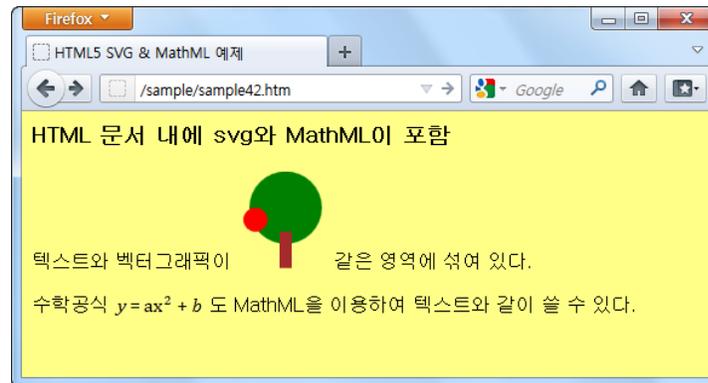
HTML5의 탄생 배경 및 특징

- 1997년 HTML 4.0 이후 큰 변화
 - XML 및 Ajax 등에 기반한 웹2.0 기술의 발전
 - XHTML의 발전은 미흡
 - 2006년 WHATWG에서 HTML 후속 버전의 개발 추진
 - W3C가 WHATWG과의 협력 하에 HTML5 워킹그룹 신설
- 리치 인터넷 애플리케이션(RIA) 개발
 - 리치 사용자 인터페이스(Rich User Interface) 구현이 필요
 - 어도비 플래시나 MS 실버라이트 등의 플러그인 이용
 - 최근에는 XML, CSS, 자바스크립트, Ajax 등 웹 표준 기술 사용



HTML5의 사양의 특징

- 마크업에는 주로 의미를 부여, 스타일은 가급적 CSS3 활용하도록
- 플러그인을 적게 사용하고 최대한 웹 표준을 적용하도록
 - SVG와 MathML 기본 지원, 자바스크립트와 HTTP도 지원
 - 리치 인터넷 애플리케이션(RIA)의 편리한 개발 위해 다양한 API 제공
 - 위치관련 및 오프라인 등 모바일 환경까지 고려한 API를 제공

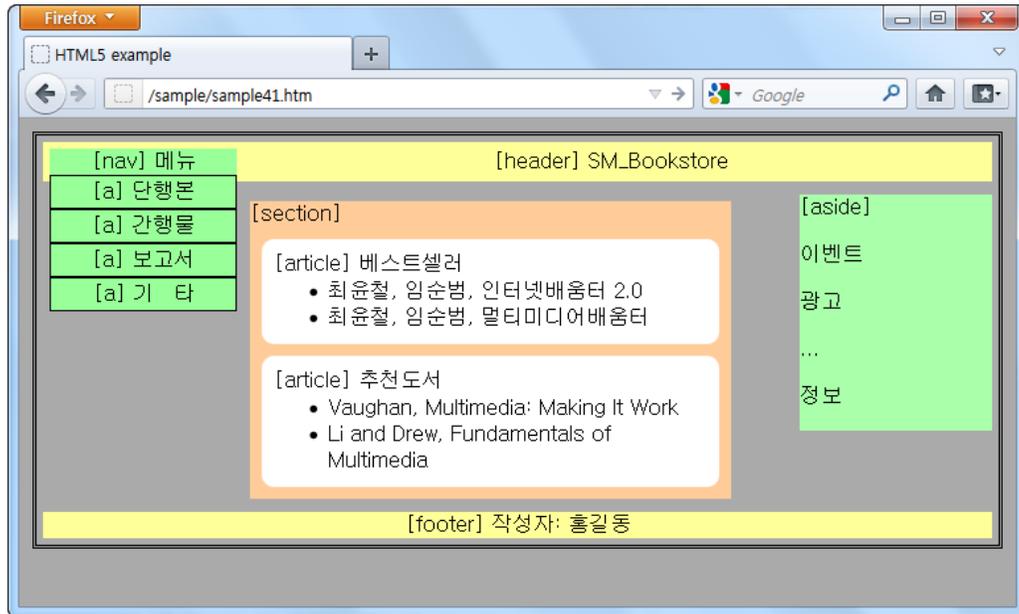


```
<html>
<head> <title> HTML5 SVG & MathML 예제 </title> </head>
<body bgcolor="#FFFF88">
  <h3>HTML 문서 내에 svg와 MathML이 포함</h3>
  텍스트와 벡터그래픽이
  <svg width="80" height="80">
    <circle cx="40" cy="30" r="30" fill="green" />
    <circle cx="15" cy="40" r="10" fill="red" />
    <rect x="35" y="50" width="10" height="30" fill="brown"/>
  </svg>같은 영역에 섞여 있다.
  <p>수학공식
  <math> <semantics>
    <mrow> <mi>y</mi> <mi>=</mi> <msup><mi>a</mi><mn>2</mn></msup>
    <mo>+</mo> <mi>b</mi> </mrow>
    <annotation>y=ax^{2}+b </annotation>
  </semantics>
  </math> 도 MathML을 이용하여 텍스트와 같이 쓸 수 있다. </p>
</body>
</html>
```

HTML5에서 변경된 엘리먼트_1

- 의미를 부여할 수 있는 구조적 마크업
 - 문서형 선언 `<!doctype html>`
 - 추가된 구조적 마크업
 - `<header>` 머리글 부분
 - `<footer>` 꼬리말 부분
 - `<nav>` 메뉴의 내비게이션
 - `<section>` 본문 내용
 - `<hgroup>` 제목줄
 - `<article>` 세부 내용
 - `<aside>` 별도의 내용을 분리된 위치에

HTML5에서 변경된 엘리먼트_2



```
<!doctype html>
<html>
<head>
  <title> HTML5 example </title>
  <style>
    body { background-color:#AAAAAA; border:double; border-width:medium;
margin:10px; }
    header { background-color:#FFFF99; text-align:center; margin:5px; padding:5px; }
    nav { display:block; width: 20%; background-color: #99FF99; float: left; }
    nav a { display:block; padding:2px; border:solid; border-width:thin; }
    section { display:block; width:50%; background-color:#FFCC99; float:left; margin:10px; }
    article { display:block; background-color:#FFFFFF; border-radius:9px; margin:9px; }
    aside { display:block; width:20%; background-color:#AAFFAA; float:right; margin:5px; }
    footer { clear:both; background-color:#FFFF99; text-align:center; margin:5px; }
    article ul { margin: 1px; }
  </style>
</head>
<body>
  <header>[header] SM_Bookstore
  <nav>[nav] 메뉴
    <a href="...">[a] 단행본</a> <a href="...">[a] 간행물</a>
  </nav>
</header>
<section> [section]
  <article>[article] 베스트셀러
    <ul> <li>최윤철, 임순범, 인터넷배움터 2.0</li>
      <li>최윤철, 임순범, 멀티미디어배움터</li> </ul>
  </article>
  <article>[article] 추천도서
    <ul> <li>Vaughan, Multimedia: Making It Work</li>
      <li>Li and Drew, Fundamentals of Multimedia</li> </ul>
  </article>
</section>
<aside>[aside]<p>이벤트</p><p>광고</p>... <p>정보</p></aside>
<footer>[footer] 작성자: 홍길동 </footer>
</body>
</html>
```

HTML5에서 변경된 엘리먼트_3

- 다양하고 편리한 웹 폼(WebForm) 입력 기능
 - <input> 태그에 추가된 type 속성
 - type = datetime, date, month, week, time: 달력에서 날짜/시간 입력
 - type = number, range : 제한된 범위의 숫자로 입력, 스크롤바 입력
 - type = color, file, search : 색상표에서 선택, 파일보기에서 파일 선택
 - type = tel, url, email : 데이터의 형식이 맞는지 유효성 확인
 - 콤보박스를 작성할 수 있는 <datalist>
 - 암호키를 선택하는 <keygen> 등의 태그
- 웹 미디어 기능의 강화
 - <video> 태그와 <audio> 태그 추가
 - <source> 태그를 사용하여 미디어 소스 파일을 지정
 - <canvas> 태그 추가: 래스터 그래픽스 기반 각종 도형 및 조작

HTML5에서 변경된 엘리먼트_4



```
<html>
<head> <title>HTML5 canvas 예제</title>
<script>
function myDraw() {
    var canvas = document.getElementById("myCanvas");
    var ctx = canvas.getContext("2d");

    ctx.strokeStyle = "rgb(220, 0,0)";
    ctx.strokeRect (5, 10, 900, 200); // 배경 사각형 테두리
    ctx.strokeText ("Vincent Van Gogh Memorial Gallery", 10,220,800); // 하단 갤러리 이름

    var gogh_img=new Array(4);
    gogh_img[0]="../pics/gogh01.jpg"; // 고흐 그림 파일 주소
    gogh_img[1]="../pics/gogh02.jpg";
    gogh_img[2]="../pics/gogh03.jpg";
    gogh_img[3]=" ";

    var frm=new Image();
    frm.src="../pics/frame.png"; // 사진액자 프레임
    for (var i=0;i<gogh_img.length;i++) {
        var img=new Image();
        img.src=gogh_img[i];
        if (img.src != " ") ctx.drawImage(img,54+i*220,56); // 고흐 그림 배치
        ctx.drawImage(frm,30+i*220,30); // 액자 배치
    }
    ctx.fillStyle = "rgba(255,255,0,0.5)";
    ctx.fillRect (690,30,190,161); // 마지막 불투명 덧칠
}
</script>
</head>
<body onload="myDraw();" >
    Canvas 위에 바탕 사각형을 그리고, 그 위에 이미지 배치(마지막은 사각형을 반투명으로 덧칠)
    <canvas id="myCanvas" width="920" height="230"></canvas>
</body>
</html>
```

HTML5에서 변경된 엘리먼트_5

- 의미(Semantics)를 가지는 요소
 - 향후 검색엔진에서 이를 활용하면 많은 도움 예상
 - 비디오/오디오 포함한 그림은 <figure>, 캡션은 <figcaption>
 - <time>은 시간을 표현, <mark>는 별도로 표시한 콘텐츠를 표시
 - 방점 등 루비 텍스트를 표현 <ruby>, <rt>, <rp> 태그 사용
- 사용중단 엘리먼트
 - CSS로 완전 대체 가능한 스타일 기반 요소는 많이 배제
 - <basefont>, <center>, <big>, , <s>, <strike>, <tt>, <u> 등
 - 웹접근성에 많은 지장을 초래하고 있는 <frame>
 - 다른 엘리먼트와 사용법에서 혼란을 주는 <applet>, <acronym>

애플리케이션 개발을 위한 다양한 API_1

- 드래그앤 드롭(Drag & Drop) API
 - 웹브라우저 내에서 다양한 요소를 드래깅(Dragging)하는 API
 - 해당하는 엘리먼트에 드래그 속성을 `draggable="true"`로 설정
 - 드래깅 이벤트에는 `dragstart`, `drag`, `drop`, `dragend`, `dragover`, `dragenter`, `dragleave` 등
- 오프라인 웹애플리케이션(Offline Web Applications) API
 - 인터넷에 접속할 수 없을 때에도 계속해서 웹 애플리케이션이나 웹문서들을 사용할 수 있도록 하는 API
 - 관련 리소스를 애플리케이션 캐시(Application Cache)에 보관
 - 매니페스트(Manifest) 파일: 캐시에 저장할 리소스 목록을 나열
 - 자바스크립트 인터페이스: 캐시된 파일의 실행/관리를 위한 API 제공

애플리케이션 개발을 위한 다양한 API_2

- 웹 스토리지(Web Storage) API
 - 사용자 입력 데이터를 클라이언트 쪽의 로컬 스토리지에 저장
 - 키(Key)-값(Value)의 형태로 저장
 - 세션 스토리지(Session Storage): 브라우저 종료시 데이터 삭제
 - 로컬 스토리지(Local Storage): 세션이 끝나도 계속 데이터 저장
- 로컬 데이터베이스(Indexed Database) API
 - 클라이언트에 데이터베이스를 저장하자는 목적
 - 초기에는 웹SQL 개발을 추진, 최근에는 Indexed DB로 방향 변경
 - 오브젝트 스토어(ObjectStore)라는 저장 공간에 데이터를 저장하고 인덱스를 통하여 검색

애플리케이션 개발을 위한 다양한 API_3

- 파일(File) API
 - 로컬 파일에 접근하여 내용을 읽고 쓸 수 있는 기능을 제공
 - 파일 업로드, 파일의 이름, 경로, 크기 등 파일정보 제공
- 웹 워커(Web Workers) API
 - 여러 프로그램이 동시 실행되는 멀티스레드(Multi-thread) 기능
- 웹 소켓(Web Sockets) API
 - 서버와 브라우저 사이에 양방향 통신 채널을 제공해 주는 기술
 - 서버와 브라우저가 연결을 유지한 상태에서 양방향 통신

애플리케이션 개발을 위한 다양한 API_4

- 웹 메세징(Web Messaging) API
 - 여러 창에서 실행되는 애플리케이션 간에 메시지를 주고받게 해주는 기능
 - 브라우저 내에서 통신을 하는 효율적인 애플리케이션 개발 가능
- 지오로케이션(Geo-Location) API
 - 사용자 PC 혹은 모바일 단말기에서 현재 위치를 파악하는 기능
 - 모바일 기기의 경우 단말기에 부착된 GPS 센서로부터 정보 제공
 - 기본적으로 위도/경도 좌표가 포함, 이외에 정확도, 고도, 방향, 속도 등의 추가적인 데이터가 제공

모바일 웹

- 모바일 환경

- 일찍부터 HTML5를 지원하는 웹 브라우저를 탑재
- 화면의 미세한 크기 차이 외에는 동등하게 HTML5 실행이 가능

- 모바일 환경에서 중요한 HTML5 기능

- HTML은 못하였지만 몇 가지 새로운 기능이 모바일 환경을 고려
 - 플래시보다는 <video>, <audio>, <canvas>와 SVG 등 표준화된 방법
 - 다양한 입력 폼을 지원하여 사용자 인터페이스 개발 및 사용이 용이
 - 지오로케이션 API로 위치정보를 활용하는 애플리케이션 개발 가능
 - 오프라인 애플리케이션 API로 인터넷 접속이 끊기는 경우를 대비
 - 오프라인이 될 경우 로컬 스토리지, 웹 데이터베이스, 애플리케이션 캐시 등이 유용하게 활용