

# 웹 프로그래밍 설계 (Web Programming Design)

高碩住 著



경북대학교 전자전기컴퓨터학부

# 서 문

이 교재는 웹 기반의 인터넷 프로그래밍 설계 과목의 강의 및 실습을 위한 자체 교재로 개발되었다. 최근 유무선 통신망에서 다양한 종류의 웹 기반 인터넷 서비스가 널리 보급됨에 따라 컴퓨터 및 정보통신 관련 전공자에게 웹 프로그래밍 능력은 필수처럼 여겨지고 있다. 이 교재에서는 먼저 인터넷 프로토콜 및 웹 프로그래밍 개용에 대하여 소개하고, 이어서 HTML, CSS, JavaScript 등의 유선 인터넷 기반 웹 프로그래밍 내용과 mobile HTML, WML 등의 모바일 인터넷 프로그래밍 내용을 기술하였다. 특히, 무선 (모바일) 웹 프로그래밍의 경우 관련 예제와 시뮬레이터 설치 및 사용법 등의 내용을 포함하여 스스로 학습할 수 있는 자료를 제공하고 있다.

학생들이 이 교재를 통해 유무선 웹 프로그래밍에 대한 이해에 조금이나마 도움이 되었으면 하고, 일상 생활에서 친숙해져 있는 WWW 웹 문서를 손 쉽게 직접 제작해 볼 수 있는 경험이 되기를 기원한다. 끝으로 이 책이 나오기까지 많은 도움을 준 경북대학교 전자전기컴퓨터학부 통신프로토콜연구실 대학원생들에게 감사의 마음을 전한다.

2009년 3월

경북대학교 전자전기컴퓨터학부

고 석 주

# 목 차

제 1 장. 인터넷 프로토콜.....	1
1.1 인터넷 서비스 .....	1
1.1.1 월드와이드웹 .....	1
1.1.2 홈페이지.....	2
1.1.3 블로그와 위키 .....	3
1.2 인터넷 주소와 도메인 이름 .....	4
1.2.1 인터넷 주소.....	4
1.2.2 도메인 이름과 DNS.....	5
1.2.3 웹주소.....	7
1.3 모바일 인터넷 .....	8
1.3.1 모바일 인터넷 서비스.....	8
1.3.2 모바일 인터넷 기술 .....	9
1.3.3 모바일 인터넷 주소 .....	10
1.4 다양한 인터넷 통신망 .....	11
1.4.1 인터넷.....	11
1.4.2 DSL.....	12
1.4.3 BcN .....	13
1.4.4 무선랜 .....	13
1.4.5 이동통신망 .....	14
1.4.6 와이브로.....	14
1.5 프로토콜 스택 .....	15
1.6 INTERNET PROTOCOL (IP).....	16
1.6.1 IP 주소의 표현.....	16
1.6.2 IP 패킷 전달 .....	18
1.6.3 패킷 캡슐화.....	19
1.6.4 인터넷 패킷 전달 .....	20
1.6.5 IP 헤더 포맷 .....	21
1.7 USER DATAGRAM PROTOCOL (UDP) .....	22

1.7.1 프로세스 통신 .....	22
1.7.2 포트번호.....	23
1.8 TRANSMISSION CONTROL PROTOCOL (TCP) .....	24
1.8.1 TCP 패킷 헤더 .....	25
1.8.2 TCP 연결 관리 .....	26
1.8.3 TCP 오류 및 흐름 제어.....	27
1.8.4 TCP 혼잡 제어.....	28
1.9 STREAM CONTROL TRANSMISSION PROTOCOL (SCTP).....	29
1.9.1 SCTP 멀티스트리밍 .....	29
1.9.2 SCTP 멀티홈잉.....	30
1.10 응용 계층 프로토콜 .....	31
1.10.1 클라이언트-서버 모델 .....	32
1.10.2 HTTP.....	33
1.10.3 FTP .....	33
1.10.4 SMTP.....	34
제 2 장. 웹 프로그래밍 개요.....	35
2.1 WWW과 HTTP 프로토콜 .....	35
2.1.1 WWW 구조.....	35
2.1.2 WWW 문서의 종류.....	36
2.1.3 HTTP 프로토콜.....	40
2.2 마크업 언어와 웹브라우저 .....	44
2.2.1 마크업 언어.....	44
2.2.2 웹브라우저의 종류.....	45
2.2.3 웹문서 제작 도구 .....	46
제 3 장. HTML .....	47
3.1 HTML 문서 구조 .....	47
3.1.1 기본 구조.....	47
3.1.2 태그(tag) .....	50
3.2 HTML 기본 태그.....	52

3.2.1 기본 태그.....	52
3.2.2 문자 서식 태그.....	56
3.2.3 단락 서식 태그.....	60
3.2.4 목록 태그.....	65
3.2.5 연결 태그.....	67
3.3 이미지 삽입하기.....	70
3.3.1 <IMG> 태그.....	70
3.3.2 <IMG> 태그의 속성.....	71
3.3.3 이미지로 문서 연결.....	74
3.4 테이블 만들기.....	75
3.4.1 테이블 기본 태그.....	75
3.4.2 테이블 서식 지정.....	80
3.4.2 테이블 정렬.....	84
3.5 프레임 만들기.....	85
3.5.1 프레임의 기본 구조.....	85
3.5.2 프레임의 서식 지정.....	88
3.5.3 타겟 프레임의 지정.....	89
3.6 입력 양식 만들기.....	92
3.6.1 입력 양식 태그.....	92
3.6.2 입력 양식 만들기 예제.....	94
3.6.3 문서 입력창 만들기.....	97
3.7 멀티미디어 활용하기.....	98
3.7.1 사운드 포함하기.....	98
3.7.2 동영상 포함하기.....	101
제 4 장. CSS.....	103
4.1 CSS 개요.....	103
4.2 스타일 시트의 사용법.....	105
4.2.1 내부 스타일 시트.....	105
4.2.2 외부 스타일 시트.....	107
4.2.3 라인 스타일 시트.....	108

4.3 스타일 정의 방법 .....	109
4.3.1 기본 방법 .....	109
4.3.2 CLASS 속성을 이용한 스타일 정의.....	110
4.3.3 ID 속성을 이용한 스타일 정의.....	112
4.4 스타일 정의를 위한 속성 .....	114
4.4.1 폰트 속성.....	114
4.4.2 텍스트 속성.....	115
4.4.3 색과 배경 속성.....	117
4.4.4 위치 속성.....	118
4.4.5 목록 속성.....	119
4.4.6 박스 스타일 속성 .....	121
제 5 장. 자바 스크립트 .....	123
5.1 자바 스크립트 소개 .....	123
5.2 자바스크립트 기초 .....	124
5.2.1 HTML과 자바스크립트.....	124
5.2.2 자바스트립트의 어휘 .....	125
5.2.3 자바스트립트의 자료형 .....	127
5.2.4 연산자(operator).....	127
5.2.5 문장(statement) .....	129
5.2.6 함수(function).....	130
5.2.7 이벤트 처리.....	132
5.3 자바 스크립트 객체 .....	134
5.3.1 사용자 정의 객체 .....	134
5.3.2 내장 객체 .....	136
5.3.3 표준 객체 .....	139
5.4 동적인 웹 문서 만들기 .....	141
5.4.1 프레임(frame) 객체 .....	142
5.4.2 레이어(layer) 객체 .....	144
5.4.3 쿠키(cookie) .....	146
5.4.4 자바 애플릿(applet) 코드와의 연동.....	148

5.5 자바스크립트 활용 .....	150
5.5.1 상태 바에서 움직이는 글자 .....	150
5.5.2 마우스를 올리면 커지는 글자 .....	151
5.5.3 달력 만들기 .....	152
제 6 장. ASP .....	154
6.1 윈도우즈 웹 서버 .....	154
6.1.1 IIS를 이용한 웹 서버 구축 .....	154
6.1.2 Access를 이용한 데이터베이스 구축 .....	158
6.1.3 ODBC 설정 .....	162
6.2 ASP 프로그래밍 .....	165
6.2.1 ASP 기초 .....	165
6.2.2 Visual Basic 스크립트 .....	166
6.2.3 ASP 객체 .....	183
6.3 SQL 프로그래밍 .....	199
6.3.1 Insert문 .....	199
6.3.2 Select문 .....	200
6.3.3 Update문 .....	205
6.4.4 Delete문 .....	206
제 7 장. 모바일 웹 프로그래밍 개요 .....	207
7.1 무선 인터넷의 이해 .....	207
7.2 무선 인터넷 모델 .....	210
7.3 휴대폰 종류 .....	211
7.4 모바일 플랫폼 .....	214
7.5 모바일 무선인터넷 시뮬레이터 .....	215
7.5.1 KTF ME1.3b 시뮬레이터 .....	215
7.5.2 Openwave 6.2.2 시뮬레이터 .....	217
7.5.3 UP 및 KUN 시뮬레이터 .....	218
제 8 장. MOBILE HTML .....	219

8.1 MHTML과 KUN (KTF UNIFIED NAVIGATOR) .....	219
8.1.1 개요 .....	219
8.1.2 mHTML 언어 구조 .....	220
8.2 MHTML 기본 태그 .....	227
8.3 그림과 링크, 목록 관련태그 .....	240
8.3.1 <img> 태그 .....	240
8.3.2 <A> 태그 .....	242
8.3.3 이미지 맵 태그 .....	244
8.3.4 목록 관련 태그 .....	247
8.4 표로 꾸며보는 폰페이지 .....	251
8.4.1 <table> 태그 .....	251
8.4.2 <frame> 태그 .....	253
8.5 MHTML용 입력양식 .....	255
8.5.1 <form> & <input> .....	255
8.5.2 목록선택 양식 .....	259
제 9 장. WML .....	262
9.1 WML 개요 .....	262
9.2 WML 기본 태그 .....	264
9.3 글과 그림 관련 태그 .....	273
9.4 링크 관련 태그 .....	279
9.5 입력양식 관련 태그 .....	286
9.6 WML 2.0 .....	288
9.6.1 WML 2.0 소개 .....	288
9.6.2 X-HTML 문서 작성 방법 .....	289
9.6.3 WML1.0을 WML2.0으로 변환하기 .....	293

# 제 1 장. 인터넷 프로토콜

## 1.1 인터넷 서비스

우리가 쉽게 접할 수 있는 주요 인터넷 서비스로는 다음이 있다.

- 월드와이드웹
- 인터넷 검색
- 인터넷 카페
- 온라인 게임
- 메신저
- 전자메일
- 홈페이지
- 블로그
- 위키

### 1.1.1 월드와이드웹

월드와이드웹(WWW; World Wide Web)이란 '세계 규모의 거미집 모양의 망'이라는 뜻으로, '하이퍼텍스트' 기능 혹은 'HTTP(HyperText Transfer Protocol)' 프로토콜을 사용하여 인터넷상에 분산되어 존재하는 온갖 종류의 정보를 통일된 방법으로 찾아볼 수 있게 하는 서비스 혹은 소프트웨어를 의미한다. 다음 그림은 WWW 의 예로써 '네이버' 웹사이트의 초기 화면을 보여준다.



<그림 1.1> 웹페이지

월드와이드웹은 줄여서 '웹(web)'이라고 부르며 최근 웹서비스가 주목 받고 있는 이유는, 문자 정보가 대부분이었던 과거의 통신수단과는 달리 WWW 에서는 문자, 화상, 음성 등에 등의 다양한 데이터 표현 및 전달이 가능하기 때문이다. WWW 의 주요 특징은 다음과 같다.

- (1) 웹문서는 웹서버(web server)라고 하는 컴퓨터에 '하이퍼텍스트' 형식으로 저장되어 있다.
- (2) 웹문서를 보기 위하여 우리는 웹브라우저(web browser)라는 소프트웨어가 필요하다.
- (3) 대표적인 웹브라우저로는 인터넷 익스플로러(Internet Explorer) 등이 있다.

하이퍼텍스트란 링크(link)에 의해서 상호연결된 텍스트 혹은 문서를 의미하며, 인터넷에 분산되어 있는 세계의 웹문서는 하이퍼텍스트로 연결될 수 있다. 전 세계의 하이퍼텍스트가 이리저리로 연결된 모습이 마치 거미가 집을 지은 것처럼 보이기 때문에 '월드와이드웹'이라는 이름이 붙여졌다. HTTP 는 하이퍼텍스트 웹문서를 전달하기 위해 사용되는 프로토콜(규칙)을 의미한다.

예) <http://www.empas.com/>

### 1.1.2 홈페이지

홈페이지란 개인이나 기관에서 자신의 홍보 및 소개를 위해 제작한 웹사이트 혹은 웹문서를 의미한다. 대부분의 학교, 기업, 정부기관 등에서 홍보를 위해 자체 홈페이지를 구축해 놓고 있다. 또한, 각 개인도 자신 소개 및 홍보를 위하여 '개인 홈페이지'를 가지고 있다. 다음 그림은 교육과학기술부의 홈페이지 화면이다.



<그림 1.2> 교육과학기술부 홈페이지

홈페이지 제작에 사용되는 프로그래밍 언어를 HTML(HyperText Markup Language)라한다. 홈페이지를 편리하게 저작하는 시각 프로그램인 소프트웨어로 '나모 웹 에디터' 등의 편집기가 있다. 실제 홈페이지 제작을 위해서는 HTML 외에도 JavaScript, ASP, JSP, PHP 등의 웹 프로그래밍 언어가 필요하며, 게시판 기능을 위해 MySQL 등을 이용한 데이터베이스 구축도 필요하다. 휴대 단말용' 무선인터넷 웹페이지도 제작할 수 있는데 m-HTML 이나 WML (Wireless Markup Language) 등의 웹 프로그래밍 언어를 사용한다.

### 1.1.3 블로그와 위키

블로그는 홈페이지와 유사한 것으로 '웹(web)'과 '항해일지'를 뜻하는 '로그(log)'의 합성어로 "인터넷 일기" 혹은 "인터넷 항해일지"를 의미한다. 블로그는 주인인 블로거가 저자이며 미디어 생성자이다. '한국형 블로그'를 '미니 홈페이지'라 부르기도 한다.

최근 인터넷 포털사이트에서 블로그 서비스를 제공하는 네이버블로그, 엠파스블로그, 파란블로그 등이 있다. 또한 최근 개인 홈페이지는 블로그 서비스로 제공되고 있으며, 휴대폰 등의 단말에서 제공되는 모블로그는 mobile 과 blog 의 합성어이다. 다음 그림은 파란블로그의 웹페이지 화면이다.



<그림 1.3> 블로그

위키란 최근에 등장한 인터넷 서비스로서 홈페이지 및 블로그와 유사한 성격을 가지나, 누구나 문서의 내용을 자유로이 추가, 변경 등의 편집을 할 수 있다는 측면에서 기존의 웹페이지와는 구별된다. 위키(WiKi)라는 용어는 하와이 원주민 말로 "빨리"라는 뜻을 지닌 "위키위키"에서 나온 말이며, "WiKi = What I Know Is" 표현의 줄임말이라는 견해도 있다. 위키페이지는 특히 과학기술 전문사이트에서 많이 사용되고 있으며, 유사한 주제를 연구하는 전문가들끼리 전문지식을 공유하고 최근 정보를 교환하는 토론의 장으로 널리 활용되고 있다.

다음 그림은 위키 관련 사이트인 '위키피디아'의 웹사이트이다.

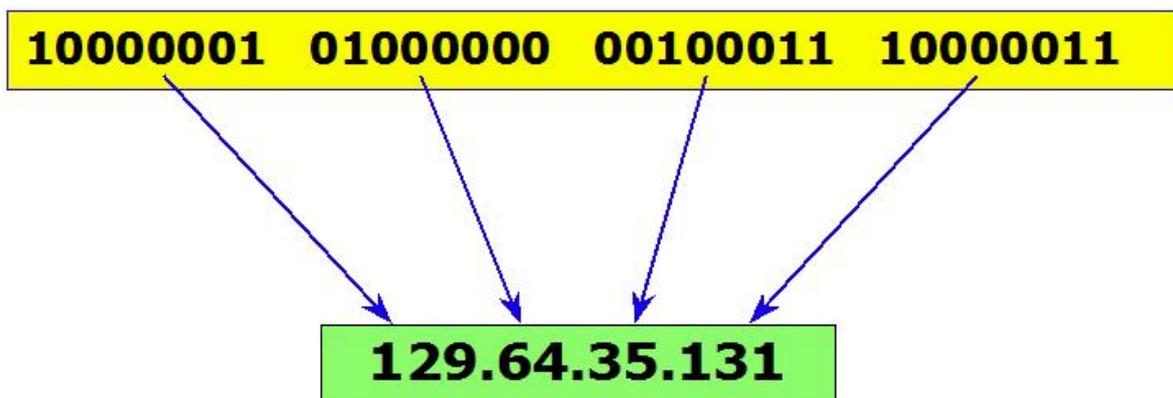


<그림 1.4> 위키사이트(<http://www.wikipedia.org/>)

## 1.2 인터넷 주소와 도메인 이름

### 1.2.1 인터넷 주소

인터넷 주소 혹은 IP 주소란 인터넷 상에서 내 컴퓨터를 식별하기 위한 주소이다. 즉, IP 주소를 통해 상대방의 컴퓨터에 인터넷 데이터를 보낼 수 있으며, 또한 상대방이 내 컴퓨터에 데이터를 전달할 수 있다. IP 주소는 "129.64.35.131"처럼 4개의 십진수와 "."를 사용하여 표현한다.



<그림 1.5> IP 주소의 표현

전 세계의 인터넷 주소 할당을 관리하는 기관은 미국에 있는 IANA(Internet Assigned Numbers Authority)이다. IANA 에서는 각 국가 혹은 단체에게 IP 주소를 할당 및 분배하는 총괄 업무를 관장하고 있다. 세계적으로 IP 주소할당을 효과적으로 추진하기 위하여 지역별로 IANA 의 역할을 대행하는 기관이 있는데 아시아-태평양 지역은 APNIC(Asia Pacific Network Information Center)에서 IP 주소 할당을 수행하고 있다. 또한 국가별로 주소를 할당해 주는 기관이 있는데 한국은 KRNIC 에서 주소 할당 업무를 수행하고 있다. IP 주소를 받고 싶은 개인이나 단체는 KRNIC 에 주소할당을 요청하거나 APNIC 이나 IANA 에 직접 주소를 요청할 수도 있다.



<그림 1.6> IP 주소할당 기관

### 1.2.2 도메인 이름과 DNS

도메인 이름이란 한마디로 컴퓨터의 이름이다. 즉, 인터넷에서 컴퓨터를 식별하기 위해 사용하는 컴퓨터의 고유 이름이다. 예를 들어, 웹이나 인터넷 서비스에서 사용하는 "[www.naver.com](http://www.naver.com)" 는 도메인 이름이다.

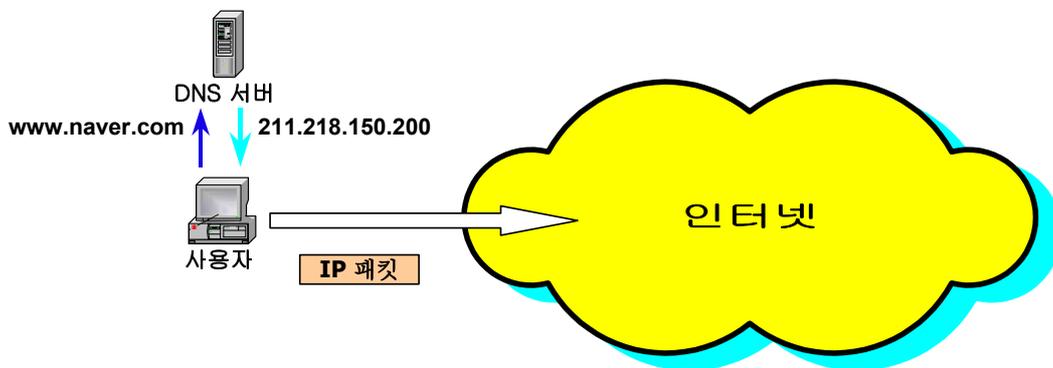
엄밀히 얘기하면 도메인 이름과 컴퓨터 이름은 다르다. 예를 들어, [www.naver.com](http://www.naver.com) 은 naver.com 도메인에 있는 웹서버(컴퓨터)의 이름이고, naver.com 은 도메인 이름이다. 즉, naver.com 도메인은 웹서버 ([www.naver.com](http://www.naver.com)) 이외에도 다른 많은 컴퓨터를 포함할 수 있다(예: mail.naver.com).

앞에서 <인터넷에서 컴퓨터를 식별하기 위한 용도로써> IP 주소를 얘기하였다. IP 주소처럼 도메인 이름도 컴퓨터의 식별자로 사용한다. 즉, 도메인 이름은 IP 주소로 대응한다. 예를 들어, "[www.naver.com](http://www.naver.com)" 도메인 이름은 "211.218.150.200"의 IP 주소를 갖는다.

그렇다면, 왜 IP 주소 외에 도메인 이름을 사용할까? IP 주소는 사람이 기억하고 사용하기에 불편하다. 예를 들어, 브라우저의 검색창에 "211.218.150.200" 이라는 IP 주소를 외워서 입력하기 보다는, "[www.naver.com](http://www.naver.com)" 이라는 웹서버의 이름을 입력하는 게 훨씬 더 편리하다.

도메인 이름 시스템 서버란 이미 언급한 "도메인이나 컴퓨터 이름을 IP 주소로 바꾸어 주는 서버"이다. 앞에서 "TCP/IP 인터넷 설정" 과정에서 'DNS(Domain Name System) 서버'의 주소를 설정하였다. 바로 여기에 설정한 DNS 서버가 검색창에 입력하는 컴퓨터 이름을 IP 주소로 바꾸어 준다.

다음 그림은 웹 브라우저를 사용하는 경우, DNS 와의 동작절차를 보인다.



<그림 1.7> DNS 서버

그림에서 웹 브라우저의 주소란에 "www.naver.com"이라 입력하면 컴퓨터는 미리 설정된 DNS 서버에게 조회하여 해당하는 IP 주소가 무엇인지 알아온다. DNS 서버가 해당하는 IP 주소를 웹 브라우저에게 알려주면 웹 브라우저는 '해당 IP 주소'를 사용하여 웹서버에게 인터넷 패킷을 전송한다.

보통은 각 학교나 회사 등의 기관별로 DNS 서버를 운영하고 있으며, 인터넷의 도메인 이름은 계층적으로 구성되어 있다. 예를 들어, [www.mest.go.kr](http://www.mest.go.kr)(교육과학기술부 홈페이지)의 경우, 'kr'은 한국(KOREA)을 의미하고, 'go'는 정부기관(government)을 'mest'는 교육과학기술부를 의미한다. '계층적'이란 의미는 'kr' 도메인 아래에 'go', 'co' (company), 'ac' (academy) 등의 하위 도메인이 있고, 또한, 'go' 도메인 아래에 'mest'(교육과학기술부) 등의 여러 하위 도메인이 있다는 것을 뜻한다.

### 1.2.3 웹주소

이제 도메인 이름을 사용하는 웹주소에 대하여 살펴본다. 웹주소란 URL(Uniform Resource Locator)이라고도 하며, 인터넷에서 특정 '웹문서의 위치'를 나타내는 주소이다. 웹주소는 대개 "컴퓨터(웹서버) 이름"과 "경로명"으로 구성된다. 다음 그림은 웹주소 형식이다.



<그림 1.8> 웹 주소 형식

예를 들어, "[www.naver.com/index.html](http://www.naver.com/index.html)"은 네이버 웹서버([www.naver.com](http://www.naver.com))에서 "index.html"이란 문서 경로를 나타내는 URL 이다. 즉, 웹브라우저의 주소창에 "[www.naver.com/index.html](http://www.naver.com/index.html)"라고 입력하면 해당 문서가 화면에 출력된다. 실제로는 "[www.naver.com](http://www.naver.com)"라고 입력해도 자동으로 "[www.naver.com/index.html](http://www.naver.com/index.html)" 문서를 참조하도록 되어 있다.

최근에는 '인터넷의 한글화' 추세에 따라 '한글' 도메인 이름도 사용하고 있다. 즉, 각 기관의 홈페이지 주소를 영문명 대신에 한글명으로 입력해도 해당 홈페이지에 접속할 수 있다. 다음 그림은 '청와대' 홈페이지의 영문 도메인명인 [www.president.or.kr](http://www.president.or.kr) 대신에 한글 도메인명인 '청와대'를 입력한 경우의 화면이다.



<그림 1.9> 한글 도메인 사용 예

해당 홈페이지를 한글 도메인명으로 접속하려면 해당 기관에서 관련 한글 도메인명 주소를 도메인 관리기관에 등록해야 한다. 국내에서 도메인명 등록은 '한국인터넷진흥원(NIDA)'을 통하여 이루어지고 있다. 관련 웹주소는 <http://www.nida.or.kr> 혹은 <http://domain.nic.or.kr> 이다.

### 1.3 모바일 인터넷

지금까지 유선인터넷에서 사용하는 여러 인터넷 서비스에 대하여 살펴보았다. 최근 휴대폰이나 PDA 등의 무선 휴대단말에서 인터넷을 사용하는 '모바일 인터넷' 혹은 '무선 인터넷' 서비스에 대하여 살펴본다.

#### 1.3.1 모바일 인터넷 서비스

모바일이나 무선 인터넷은 이동/무선 휴대 단말로 언제 어디서나 인터넷에 접속하여 다양한 정보검색과 전자상거래 등의 서비스 제공을 의미한다. 즉, 기존 인터넷 환경의 시간적 공간적 제약을 극복한 유비쿼터스 인터넷 서비스 환경이다. 휴대폰에서 음성전화 서비스만 제공하였으나, 모바일 인터넷 서비스로 인해 "유선과 무선 기술이 통합"되고 "PC 와 휴대폰 서비스가 융합"되었다.

모바일 인터넷에서는 기존 유선 인터넷에서 제공되던 웹서핑, 게임, 메신저 및 온라인 전자상거래를 포함하여 사실상 모든 종류의 인터넷 서비스를 무선 휴대단말기를 통하여 제공한다. 다음 그림은 휴대폰에서 사용할 수 있는 모바일 인터넷 서비스 화면의 예이다.



<그림 1.10> 모바일 인터넷 서비스

### 1.3.2 모바일 인터넷 기술

유선 인터넷과는 달리 모바일 인터넷은 단말기 화면의 크기가 작고, 네트워크 전송 속도가 느린 특징이 있다. 따라서, 유선인터넷에서 사용하던 HTTP, HTML 등의 기술을 무선 인터넷 환경에 그대로 적용하기는 어렵다. 그러므로 모바일 인터넷 서비스를 위해서는 전용 웹브라우저와 전용 마크업 언어가 필요하다.

모바일 인터넷용 웹브라우저에는 WAP(Wireless Application Protocol) 브라우저와 ME(Mobile Explorer)가 많이 사용된다. WAP 브라우저는 "WAP 포럼([www.wapforum.org](http://www.wapforum.org))"이라는 국제 표준기구에서 개발한 것이고, ME는 Microsoft사에서 모바일용으로 개발한 브라우저이다. 또한, 마크업언어 관점에서도 기존 유선 인터넷 환경에서 사용하던 HTML 대신에 WML(Wireless Markup Language) 혹은 m-HTML(mobile HTML) 등의 언어를 사용하여 웹문서가 제작되어야 한다.

다음은 국내 이동통신사업자들이 주로 사용하고 있는 브라우저와 마크업 언어이다.

이동통신사	대표 서비스명	브라우저	마크업 언어
SKT	NATE ( <a href="http://www.nate.com">www.nate.com</a> )	WAP	WML
KTF	매직앤 ( <a href="http://www.magicn.com">www.magicn.com</a> )	ME	m-HTML
LGT	EZ-i ( <a href="http://www.ez-i.co.kr">www.ez-i.co.kr</a> )	WAP	WML

모바일 인터넷 서비스를 이용하기 위해서는 휴대폰, PDA 등의 휴대단말기에 WAP, ME 등의 모바일 전용 브라우저가 탑재되어 있어야 하며, 또한 웹서버에 위치하는 각종 웹문서들도 WML, m-HTML 등의 전용 마크업 언어로 작성되어야 한다.

모바일 인터넷 서비스를 위한 무선 접속기술은 cdma2000 기술이 널리 사용되고 있으며, 최근에는 '와이브로(WiBro)'라 불리는 차세대 무선접속 기술이 등장하였다. 한국에서는 다양한 모바일 인터넷용 서비스 및 콘텐츠의 효율적인 개발을 위해 WIPI(Wireless Internet Platform for Interoperability)라는 무선인터넷 플랫폼 기술을 표준화하였다([www.wipi.or.kr](http://www.wipi.or.kr)).

### 1.3.3 모바일 인터넷 주소

한편, 휴대단말에서 무선인터넷을 사용할 수 있도록 모바일주소(WINC) 서비스도 제공되고 있다. 모바일주소란 무선인터넷 이용자들의 이용환경을 개선시키기 위하여 국가 인터넷 주소자원 관리기관인 한국인터넷진흥원(NIDA)에서 국내 이동통신사와 공동으로 복잡한 URL 대신 번호를 통해 무선인터넷에 접속하도록 실시하는 공공서비스이다.

예를 들어, 한국인터넷진흥원 도메인은 www.nida.or.kr 인데, 여기서 www 와 최상위도메인을 제거한 문자는 nida 이며, 각 글자에 해당하는 휴대폰 키패드의 숫자(번호)는 6432 이다. 또한 고유번호는 0 이므로, 한국인터넷진흥원의 모바일주소(WINC)는 6432#0 이다. 이처럼 도메인 문자들을 숫자로 대응시키는 과정에서 중복이 발생할 수 있으므로, 실제로는 좀 더 크고 다양한 WINC 번호가 할당된다.



<그림 1.11> 모바일 주소 예

#### <메모>

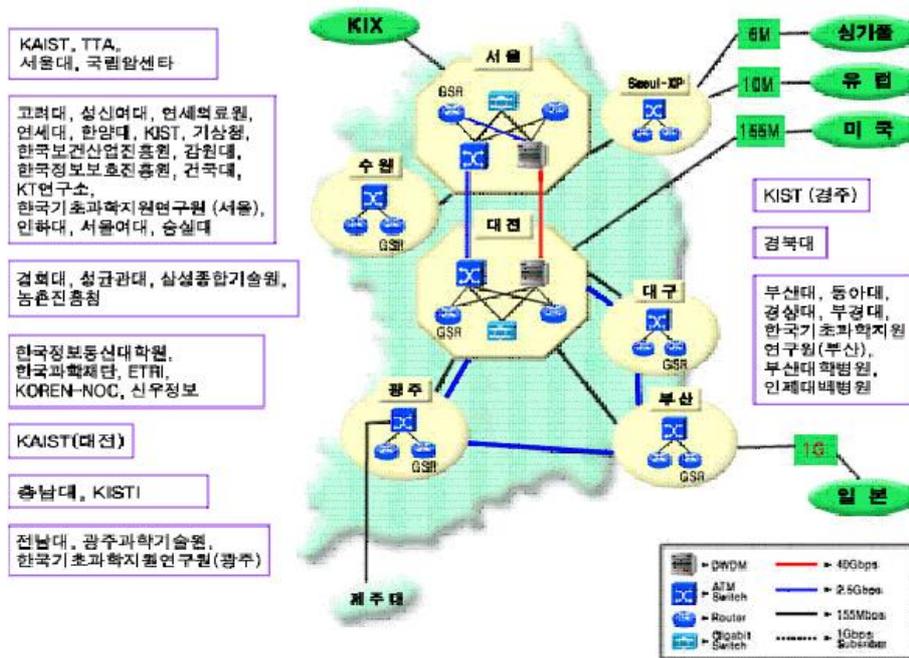
- WINC 는 Wireless Internet Numbers for Contents 의 약어로서, “무선인터넷 콘텐츠 접근번호체계”를 의미함
- 보다 상세한 WINC 사용법은 다음 홈페이지 참조 (<http://www.winc.or.kr/>)
- 실제로 휴대폰으로 버스 위치를 검색하는 “버스정보시스템(businfo)”에서 WINC 번호를 사용하고 있음

## 1.4 다양한 인터넷 통신망

인터넷 통신망 혹은 컴퓨터 네트워크란 말 그대로 컴퓨터로 구성되는 네트워크이며 사실상 인터넷을 의미한다. 즉, 인터넷 서비스는 인터넷이라는 컴퓨터 네트워크를 통해 이루어지는 컴퓨터 통신서비스인 셈이다. 여기서는 인터넷과 관련된 다양한 통신망에 대하여 살펴본다.

### 1.4.1 인터넷

인터넷은 1970년대 초에 미국 대학 및 연구소를 중심으로 컴퓨터간 데이터 통신을 위해 처음으로 인터넷이 개발되었다. 국내에 인터넷이 널리 보급된 시기는 1990년대에 WWW이 활성화되면서 본격적으로 인터넷이 사용되기 시작하였다. 다음 그림은 국내 인터넷 선도시험망 구성도이다.

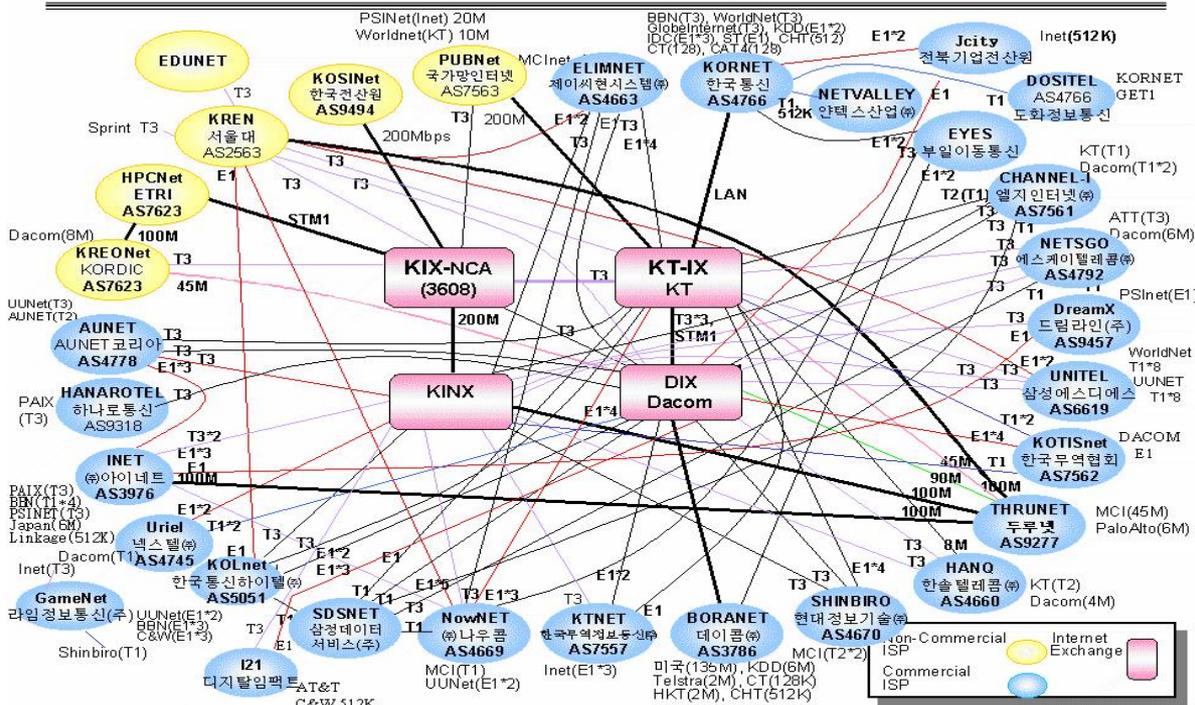


<그림 1.12> 국내 인터넷 선도시험망

실제 인터넷은 여러 가지 다양한 종류의 네트워크의 결합체이다. 인터넷서비스사업자(ISP)별로 소규모 인터넷이 구성되고, 이러한 소규모 인터넷이 국내 인터넷 백본망으로 결합되어 해외에 있는 인터넷과도 연결된다. 다음 그림은 실제 국내 인터넷 상용망 구성도를 이다.

# Internet Connectivity Map(Domestic)

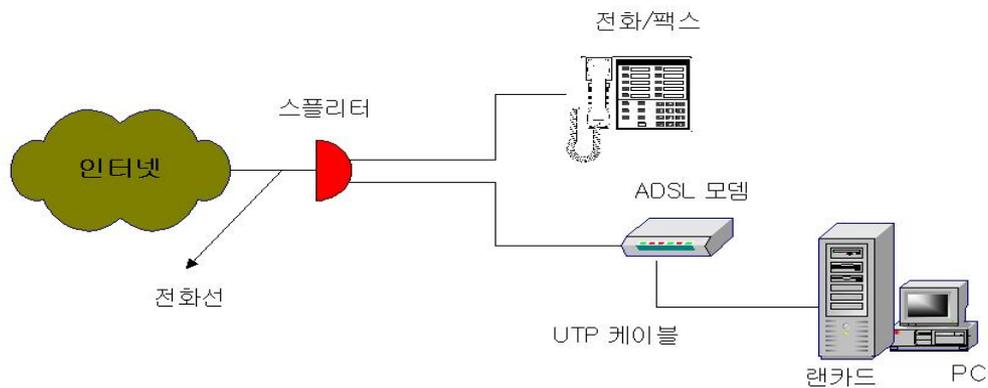
LAST UPDATED 99.6.30  
BY KRNIC



<그림 1.13> 국내 상용 인터넷망

## 1.4.2 DSL

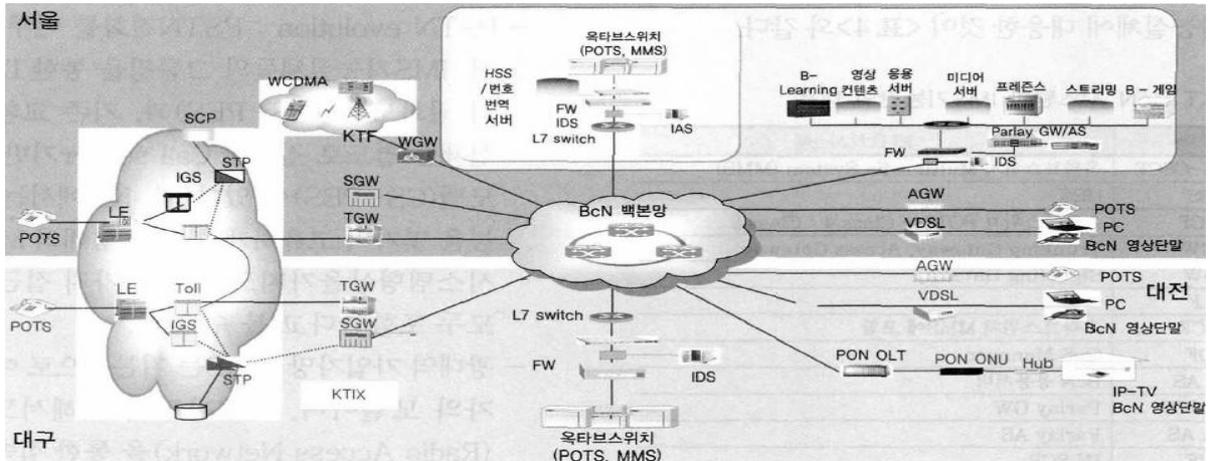
우리 가정에서 사용하는 PC 는 대개 DSL(Digital Subscriber Line)이라는 접속기술을 사용하여 인터넷에 접속하게 된다. 특히, ADSL(Asymmetric DSL) 접속기술이 널리 사용되고 있는데, 이는 각 가정의 PSTN 전화회선을 사용하여 인터넷에 접속할 때에 사용되는 기술이다. 각 가정에 유입되는 전화회선은 스플리터(splitter)를 통해 일반전화용과 ADSL 용으로 분리되고, PC 는 ADSL 모뎀을 사용하여 인터넷에 접속된다.



<그림 1.14> ADSL 기반 인터넷 접속

### 1.4.3 BcN

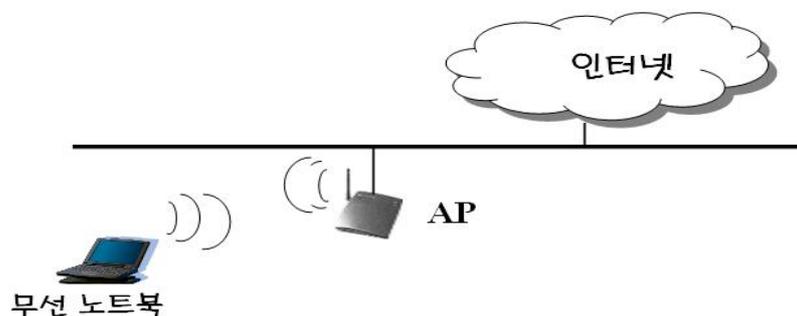
인터넷 사용자가 급증하고 또한 기존의 인터넷 서비스 뿐만 아니라 각종 이미지, 오디오, 비디오 등의 멀티미디어 데이터가 인터넷을 통해 제공됨에 따라, 인터넷 망으로 광대역 통신망으로 개선시킬 필요성이 대두 되었다. 이와 같은 인터넷망의 고도화 정책에 따라 국내에서는 차세대 통신망이라 불리는 BcN(Broadband convergence Network) 망기술을 개발하게 되었다. 다음 그림은 KT 에서 구축중인 BcN 망 구성도이다.



<그림 1.15> 국내 BcN 망 구성도

### 1.4.4 무선랜

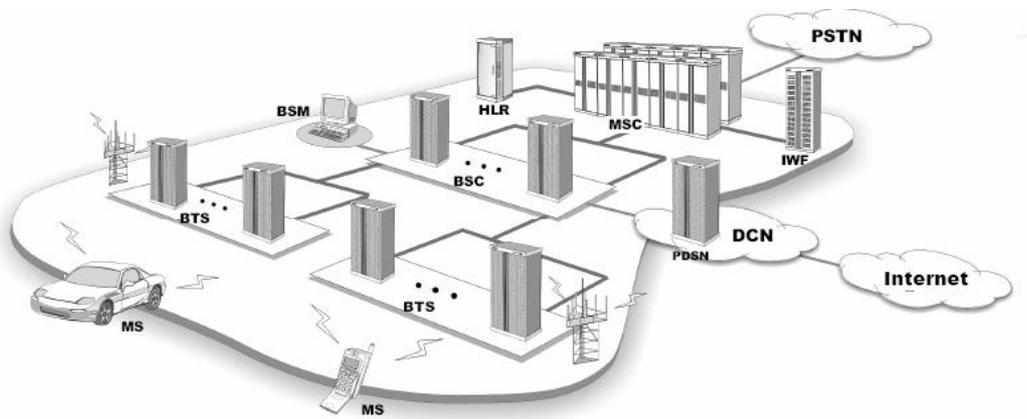
무선랜 혹은 WLAN 기술은 LAN 기술을 무선 영역으로 확대한 기술로써, 노트북 등의 이동 단말장치에 무선랜 카드를 장착하여 인터넷 접속에 사용되는 기술이다. 무선랜 접속을 위해서는 무선랜카드와 액세스 포인트라는 장비가 필요하다. 노트북에 무선랜카드를 장착하고, 주변에 설치된 AP를 통해 인터넷에 접속하게 된다. 다음 그림은 간략한 무선랜망 구성도이다.



<그림 1.16> 무선랜 망 구성도

### 1.4.5 이동통신망

우리가 사용하는 휴대폰은 이동통신망을 사용하여 데이터 통신을 수행하는데, 현재 사용되는 기술을 이동통신기술 혹은 IMT-2000(International Mobile Telecommunications - 2000) 기술이라 한다. 특히, IMT-2000 기술은 휴대폰에서 이동전화 뿐만 아니라 무선인터넷 서비스도 제공하고 있는데, 세부적인 무선접속기술에 따라 크게 CDMA(Code Division Multiple Access) 및 W-CDMA(Wideband CDMA) 기술로 분류할 수 있다. 다음 그림은 우리가 흔히 사용하는 CDMA2000 이동통신망의 구성도이다.



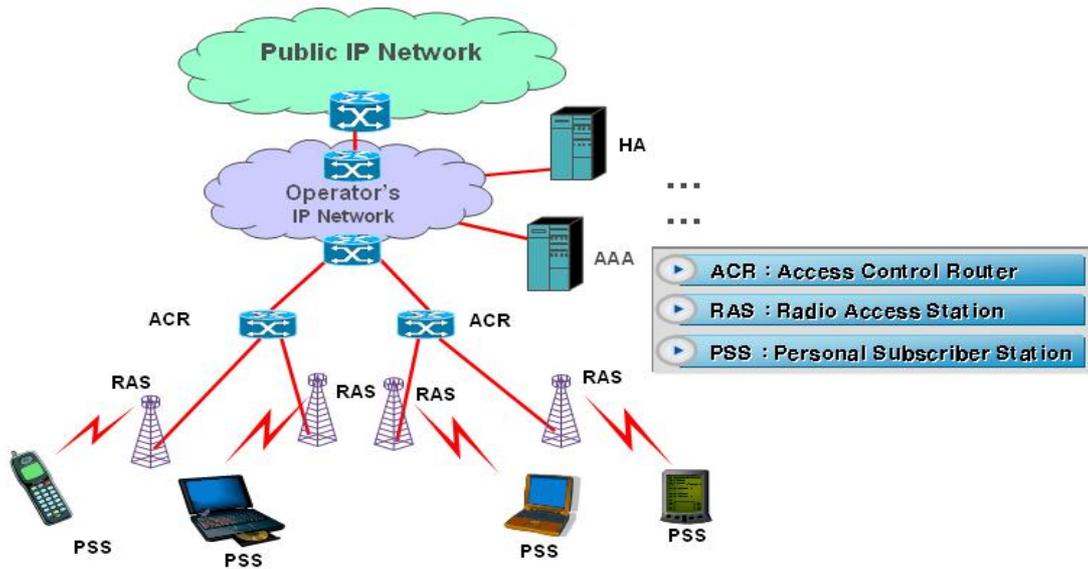
<그림 1.17> 이동통신망 구성도

### 1.4.6 와이브로

한편, 무선 이동통신망을 사용한 인터넷 접속의 광대역화 및 이동성 제공을 위해 새로이 등장한 무선접속기술이 있는데, 바로 와이브로(WiBro: Wireless Broadband) 기술이다. 와이브로 기술은 차세대 무선통신기술로 불리며, 기본 WLAN 커버리지를 1 km 대로 확대하였으며, 이동 중에도 유연한 무선 인터넷 서비스 제공을 주요 목표로 하고 있다.

기존 휴대폰의 이동통신기술이 음성전화서비스와 인터넷서비스를 동시에 제공하는 것과는 달리, 와이브로는 인터넷 서비스에 특화된 기술이며, 높은 데이터 수신 속도와 비교적 저렴한 이용요금을 특징으로 한다.

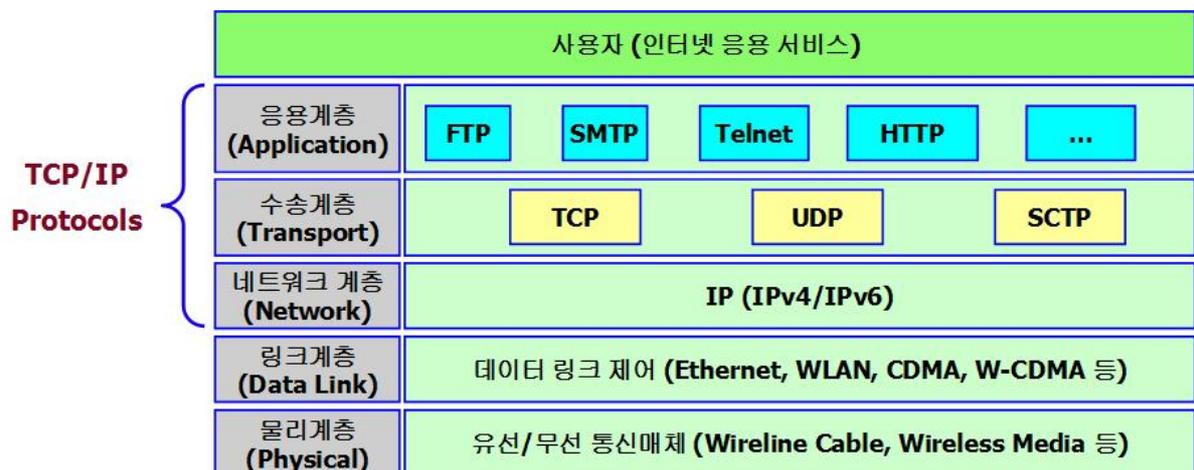
다음 그림은 와이브로 서비스를 위한 네트워크 구성도이다. 이동단말은 RAS 기지국에 연결되며, ACR 장비를 통해 인터넷에 접속된다.



<그림 1.18> 와이브로 통신망

## 1.5 프로토콜 스택

인터넷(Internet) 통신은 다음 그림처럼 'TCP/IP 프로토콜(protocol) 스택(stack)'에 해당하는 여러 가지 프로토콜들을 사용하여 이루어진다.



<그림 1.19> TCP/IP 프로토콜 스택

TCP/IP 프로토콜 스택은 기존의 OSI(Open System Interconnection) 7 계층 모델과는 달리 물리계층, 링크계층, 네트워크 계층, 수송계층 및 응용계층으로 나뉘어진다. 그림에서 응용계층은 OSI 모델의 계층 5(세션계층), 계층 5(표현계층), 계층 7(응용계층)의 기능을 모두 제공한다. 예를 들면, FTP는 파일전송과 관련된 세션제어, 표현 및 응용 기능 모두를 제공한다.

특히, 네트워크 계층의 IP 프로토콜과 수송계층의 여러 프로토콜, 그리고 응용계층의 여러 응용 프로토콜들을 TCP/IP 프로토콜 스택이라 부른다. 이와 같은 TCP/IP 프로토콜 스택은 하위 링크계층 및 물리계층에 다양한 종류의 프로토콜 혹은 접속기술을 지원한다.

<메모>

- 프로토콜이란 한 마디로 “통신규약 혹은 통신규칙”을 의미한다. 즉, 두 대의 컴퓨터가 서로 통신을 하기 위해서는 통신 프로토콜에 따라 동작을 해야 한다. 따라서 프로토콜은 모든 인터넷 장비들이 지켜져야 하며, 대개 표준화 기구에서 통신용 프로토콜 표준을 제정한다.
- TCP/IP 프로토콜의 표준화는 IETF(Internet Engineering Task Force) 표준화기구에서 표준으로 제정된다. (<http://www.ietf.org/> 참조)
- 반면에 하위 링크 계층 및 물리계층 기술은 다양한 인터넷 접속(access) 기술을 의미하며, CDMA (<http://www.3gpp.org/> 참조) 및 W-CDMA (<http://www.3gpp2.org/> 참조) 등의 이동통신기술 뿐만 아니라, IEEE 802 위원회에서 표준으로 제정된 WLAN, WiBro 등의 다양한 무선접속기술을 포함한다. (<http://www.ieee802.org/> 참조)

## 1.6 Internet Protocol (IP)

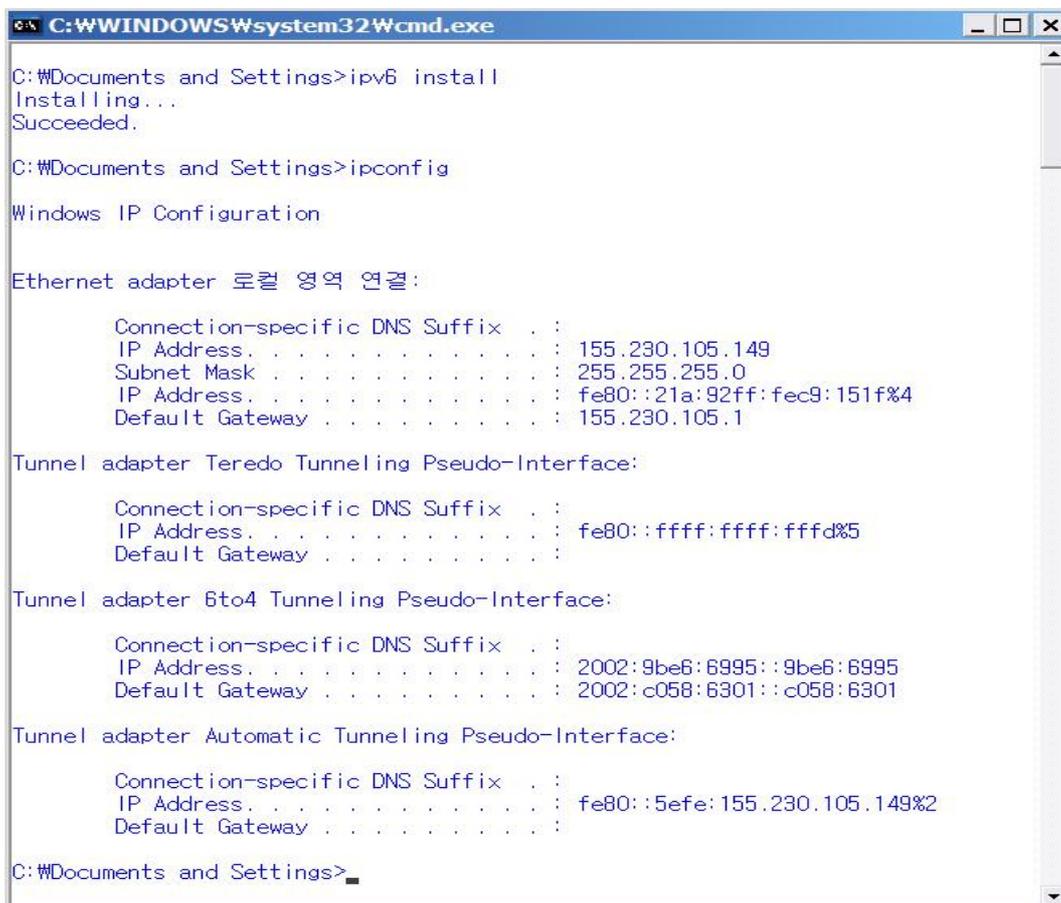
### 1.6.1 IP 주소의 표현

IP 프로토콜은 인터넷에서 컴퓨터간에 데이터 패킷(packet) 전달을 위해 사용되는데, 이를 위해서는 인터넷상의 특정 컴퓨터를 식별하기 위한 IP 주소(address)가 필요하다. 즉, IP 프로토콜은 해당 컴퓨터의 IP 주소를 참조하여 패킷을 최종 목적지에 전달한다.

IPv4(IP 버전 4)의 주소는 “129.64.35.131”처럼 4 개의 십진수로 표현되며, 이를 “dotted-decimal 표현방식”이라 한다. 각 숫자는 1byte(8bits)의 크기를 가지므로 “0~255” 사이의 숫자를 가지게 된다.

한편, IPv4 주소는 4 바이트(32 비트)의 크기를 가지는 반면에, IPv6 주소는 16 바이트(128 비트)의 크기를 갖는다. "3ffe:ffff::2085"처럼 IPv6 주소는 16 진수로 표현되고, 16 진수 4 개당 콜론(:)으로 구분하며, 콜론이 연속으로 2 개가 있는 경우에는 가운데에 모두 '0'이 있음을 의미한다.

다음 그림은 Window 에서 DOS 창을 통해 현재 사용 중인 IP 주소 정보를 파악하는 화면이다. Window XP 에서는 "ipv6 install" 명령을 통해 IPv6 기능을 사용할 수 있으며, 그림에서 해당 컴퓨터는 IPv4 주소와 함께 "2002:"로 시작하는 IPv6 주소를 사용하고 있음을 확인할 수 있다.



```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings>ipv6 install
Installing...
Succeeded.

C:\Documents and Settings>ipconfig

Windows IP Configuration

Ethernet adapter 로컬 영역 연결:

    Connection-specific DNS Suffix . . :
    IP Address. . . . . : 155.230.105.149
    Subnet Mask . . . . . : 255.255.255.0
    IP Address. . . . . : fe80::21a:92ff:fec9:151f%4
    Default Gateway . . . . . : 155.230.105.1

Tunnel adapter Teredo Tunneling Pseudo-Interface:

    Connection-specific DNS Suffix . . :
    IP Address. . . . . : fe80::ffff:ffff:fffd%5
    Default Gateway . . . . . :

Tunnel adapter 6to4 Tunneling Pseudo-Interface:

    Connection-specific DNS Suffix . . :
    IP Address. . . . . : 2002:9be6:6995::9be6:6995
    Default Gateway . . . . . : 2002:c058:6301::c058:6301

Tunnel adapter Automatic Tunneling Pseudo-Interface:

    Connection-specific DNS Suffix . . :
    IP Address. . . . . : fe80::5efe:155.230.105.149%2
    Default Gateway . . . . . :

C:\Documents and Settings>
```

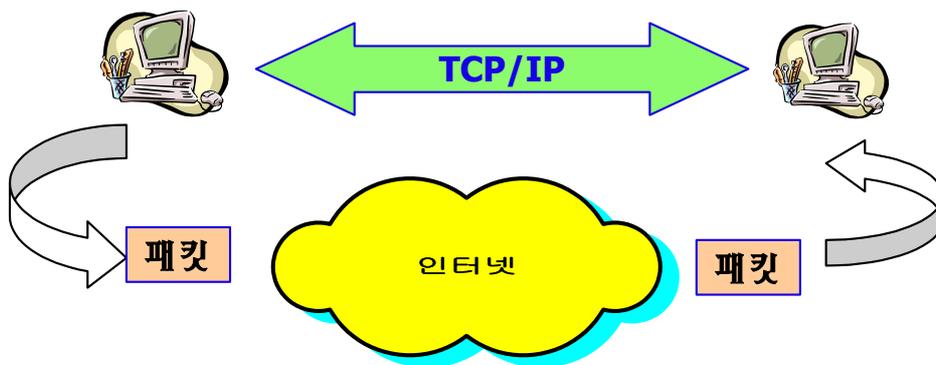
<그림 1.20> IP 주소 확인

## 1.6.2 IP 패킷 전달

TCP/IP 프로토콜의 주요 기능은 “인터넷을 통한 사용자 컴퓨터간 패킷(데이터) 전달”이다. 즉, 송신 컴퓨터의 데이터를 인터넷을 통하여 수신 컴퓨터에게 전달하는 것이다. TCP/IP 프로토콜에 의한 송수신 컴퓨터간 인터넷 패킷 전달과정을 정리하면 다음과 같다.

- (1) 송신자는 전송할 응용 데이터를 만들고, 수신 컴퓨터의 IP 주소를 파악한다.
- (2) IP 패킷 헤더에 수신자의 IP 주소를 첨가한 후, 헤더와 데이터로 구성되는 패킷을 만든다.
- (3) 패킷을 인터넷으로 전송한다.
- (4) 인터넷은 IP 라우팅 절차에 따라 수신자 컴퓨터에게 패킷을 전달한다.
- (5) 수신 컴퓨터는 패킷에서 응용 데이터를 추출하여 상위에 있는 수신자에게 전달한다.

다음 그림은 TCP/IP 프로토콜을 이용한 IP 패킷 전달 과정이다.



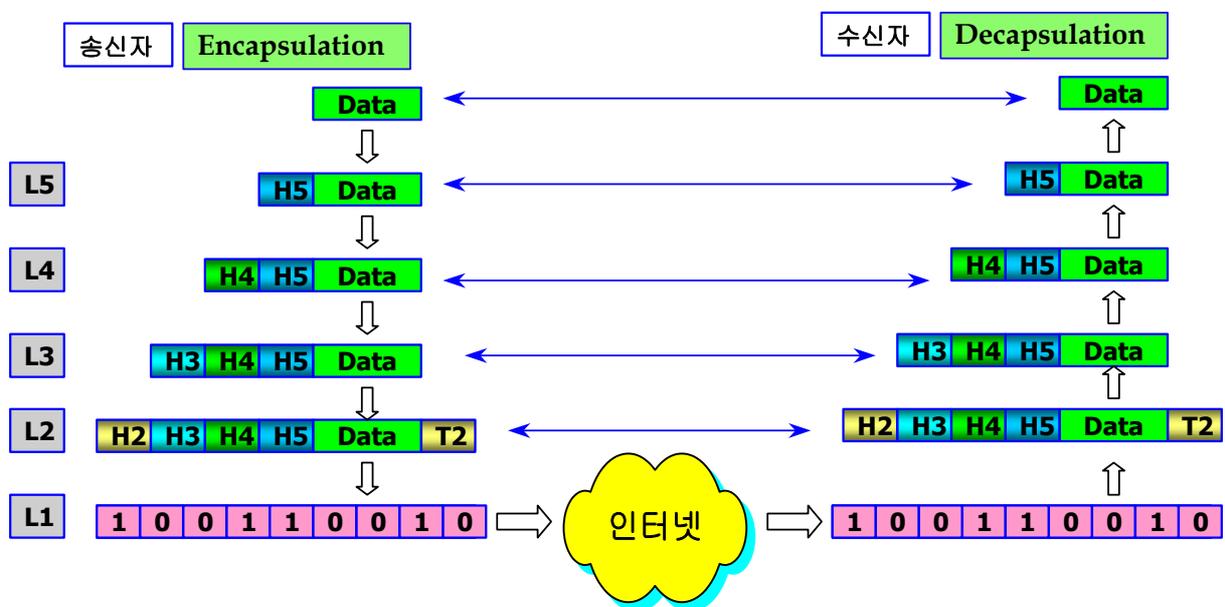
<그림 1.21> IP 패킷 전달

송신 컴퓨터에서 전송된 IP 패킷은 실제 인터넷 망에서 여러 개의 라우터를 거쳐 최종 수신 컴퓨터에 전달되는데, 이를 IP 라우팅이라 하며 IP 라우팅을 위해서 각 라우터들은 라우팅 테이블을 갖추고 있다.

### 1.6.3 패킷 캡슐화

TCP/IP 패킷 전달과정에서 중요한 점은 캡슐화(encapsulation)이다. 사용자의 데이터는 캡슐화 과정을 통해 인터넷 패킷으로 구성되고, 송신 컴퓨터를 떠나 인터넷을 향한다. 인터넷을 도착한 패킷이 수신 컴퓨터에 도달했을 때, 데이터는 역캡슐화(decapsulation)를 통해 수신자에게 전달된다.

다음 그림은 TCP/IP 인터넷 패킷 전달을 위한 캡슐화 및 역캡슐화 과정이다. 그림에서처럼, 송신 컴퓨터가 발생시킨 데이터는 TCP/IP 프로토콜 스택을 따라 내려오면서, 각 계층에 해당하는 헤더 혹은 트레일러가 첨가된다. 이러한 헤더들의 정보는 각 TCP/IP 계층에서 역할 수행을 위한 중요한 정보를 지니고 있다. 예를 들어, IP 헤더에는 IP 주소가 포함되어 인터넷 패킷전달에 중요한 정보로 사용된다.

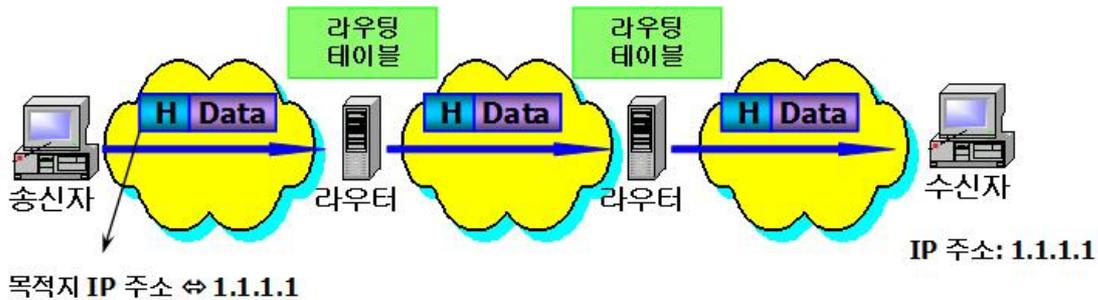


<그림 1.22> 인터넷 패킷 캡슐화 과정

송신 컴퓨터의 패킷은 물리계층에서 비트형태로 인터넷 망으로 전송되며, 인터넷 망에서는 라우팅 과정을 통해 수신 컴퓨터에게 해당 패킷을 전달한다. 수신 컴퓨터에 도착한 패킷은 다시 역캡슐화 과정을 통해 헤더를 제외한 데이터 부분만 상위계층으로 전달된다. 이와 같은 과정을 통해, 최종적으로 송신 사용자가 생성한 데이터가 수신 사용자에게 전달된다.

## 1.6.4 인터넷 패킷 전달

다음 그림은 IP 프로토콜에 의한 인터넷 패킷 전달과정을 보여준다.



<그림 1.23> 인터넷 패킷 전달

IP 패킷은 헤더(header)와 데이터(data) 부분으로 구성되는데, 헤더 부분에 송신자와 수신자의 IP 주소가 기록된다. IP 패킷전달은 인터넷 상의 라우터(router)에 의해서 이루어진다. 각 라우터는 '라우팅 테이블(routing table)'을 참조하여, 특정 패킷을 어디로 전달(forwarding)해야 할지를 결정한다. 이처럼 여러 라우터의 패킷 전달과정을 통해 최종적으로 IP 패킷이 수신컴퓨터에 도착하게 된다.

라우터들은 인접 라우터와 '라우팅 프로토콜'를 사용하여 라우팅 테이블 정보를 갱신한다. 이와 같은 라우팅 프로토콜에는 RIP(Routing Information Protocol), OSPF(Open Shortest Path First) 및 BGP(Border Gateway Protocol) 등의 프로토콜이 있다.

다음 그림은 "ping6" 프로그램을 사용한 IPv6 패킷 전달과정을 보여준다.

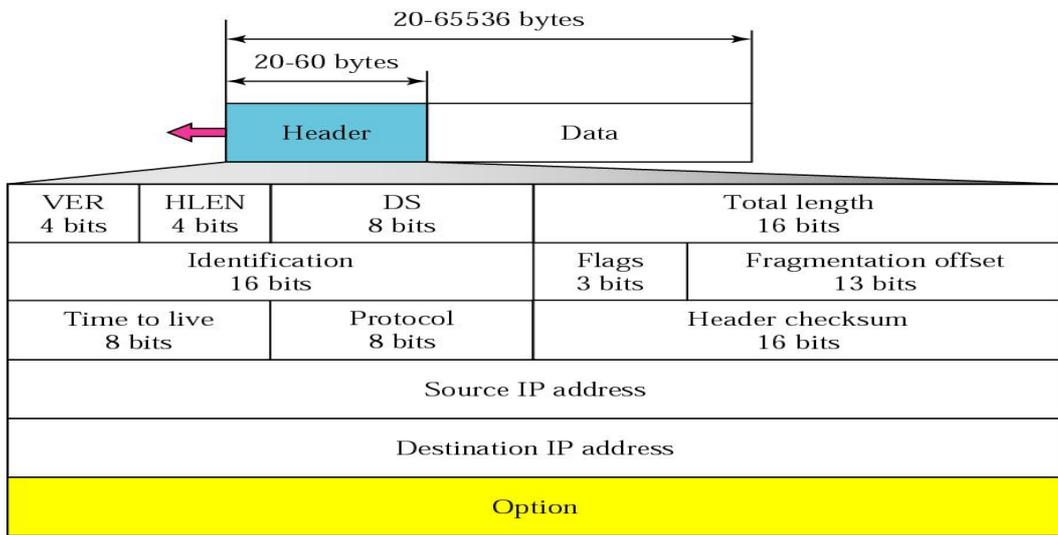
```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings>ping6 www.kame.net
Pinging www.kame.net [2001:200:0:8002:203:47ff:fea5:3085]
from 2002:9be6:6995::9be6:6995 with 32 bytes of data:
Reply from 2001:200:0:8002:203:47ff:fea5:3085: bytes=32 time=274ms
Reply from 2001:200:0:8002:203:47ff:fea5:3085: bytes=32 time=271ms
Reply from 2001:200:0:8002:203:47ff:fea5:3085: bytes=32 time=271ms
Reply from 2001:200:0:8002:203:47ff:fea5:3085: bytes=32 time=270ms
Ping statistics for 2001:200:0:8002:203:47ff:fea5:3085:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 270ms, Maximum = 274ms, Average = 271ms
C:\Documents and Settings>
```

<그림 1.24> ping6 사용 예

### 1.6.5 IP 헤더 포맷

IP 주소는 IP 패킷의 헤더부분에 기록되어 라우터의 패킷 전달과정에 참조된다. 그 밖에 IP 헤더는 여러 가지 다양한 정보를 포함하는데, 웹 프로그래밍의 이해를 위해 헤더 포맷(format)을 알아둘 필요가 있다.

다음 그림은 IPv4 헤더의 구조를 보여준다.



<그림 1.25> IPv4 헤더 구조

IPv4 헤더는 총 20 바이트로 구성되며, 필요에 따라 옵션(option)이 추가될 수 있다. 주요 필드에 대한 설명은 다음과 같다.

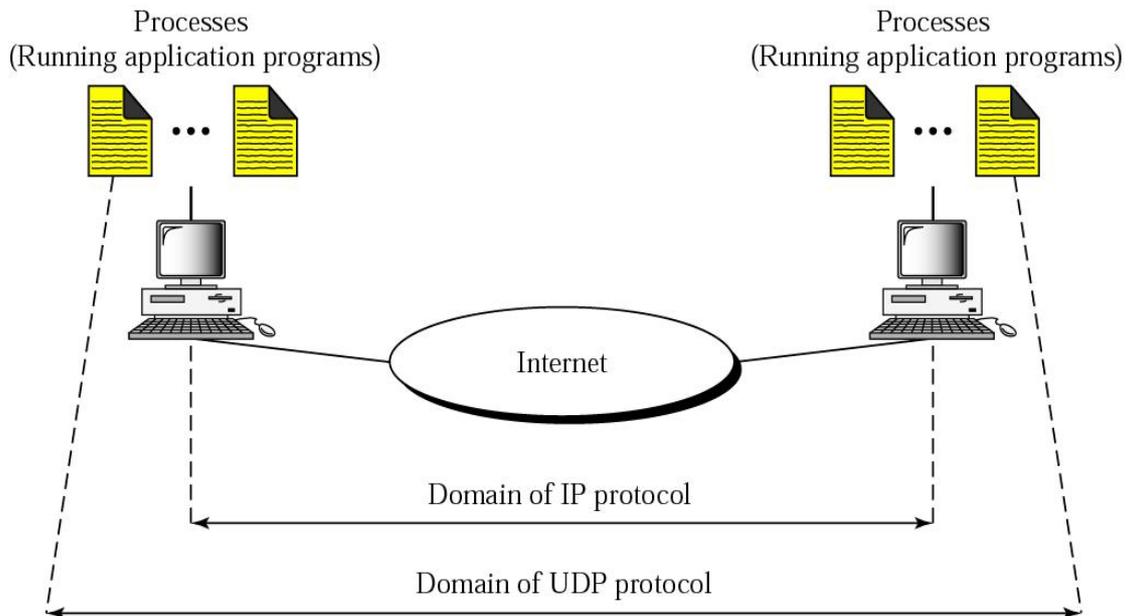
- VER: IPv4 혹은 IPv6 등의 IP 버전을 나타냄
- HLEN: 옵션을 포함한 IPv4 헤더의 길이를 표시함(20~60 바이트; 4 바이트 word 단위)
- DS: Differentiated Service 프로토콜이 사용되는 경우 해당 정보를 표시함
- Total Length: 헤더와 데이터를 포함한 IP 패킷의 전체 길이를 표시함
- Identification, Flags, Fragmentation offsets: IP 패킷이 전송 도중에 fragmentation 되었을 경우에 관련 정보를 표시함
- Time To Live (TTL): IP 패킷이 몇 개의 라우터를 경유하였는지를 표시함

- Protocol: IP 패킷의 데이터가 포함하는 프로토콜 정보를 표시함 (예: TCP, UDP 등)
- Header Checksum: 헤더가 전송 도중에 오류를 경험했는지 검사하기 위해 사용됨
- Source 및 Destination IP 주소: 송수신 컴퓨터의 IP 주소를 기록함

## 1.7 User Datagram Protocol (UDP)

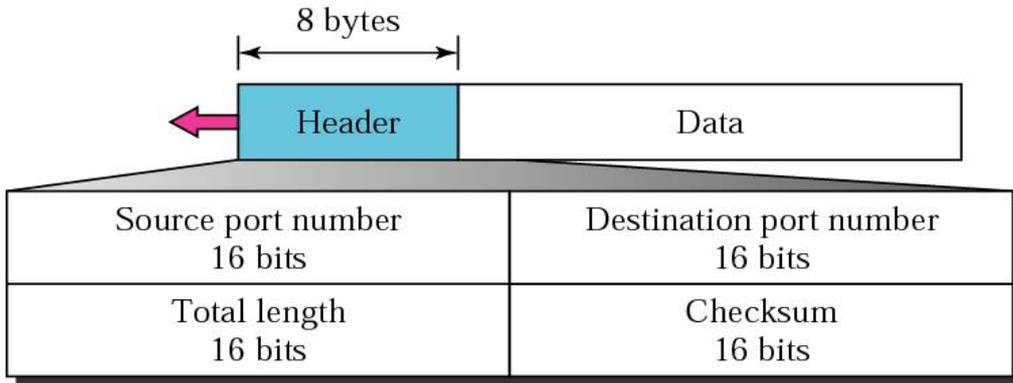
### 1.7.1 프로세스 통신

IP 프로토콜은 네트워크 계층에서 송신 단말과 라우터, 라우터와 라우터, 라우터와 수신 단말간에 IP 패킷을 전달하기 위해 사용되는 반면에, UDP, TCP, SCTP 프로토콜은 송신 및 수신 단말간에 종단간(end-to-end) 통신을 위해 사용된다. 이를 프로세스간(process-to-process) 통신이라 부른다.



<그림 1.26> 종단 프로세스 통신

UDP 의 주요 기능(사실상 유일한 기능)은 종단 응용 프로그램(혹은 프로세스)간 통신을 위해 포트(port) 번호를 제공하는 것이다. 그림 1-8 처럼 UDP 패킷 헤더는 8 바이트의 고정 길이를 가지며, 2 바이트 크기의 송수신 포트번호를 포함한다.

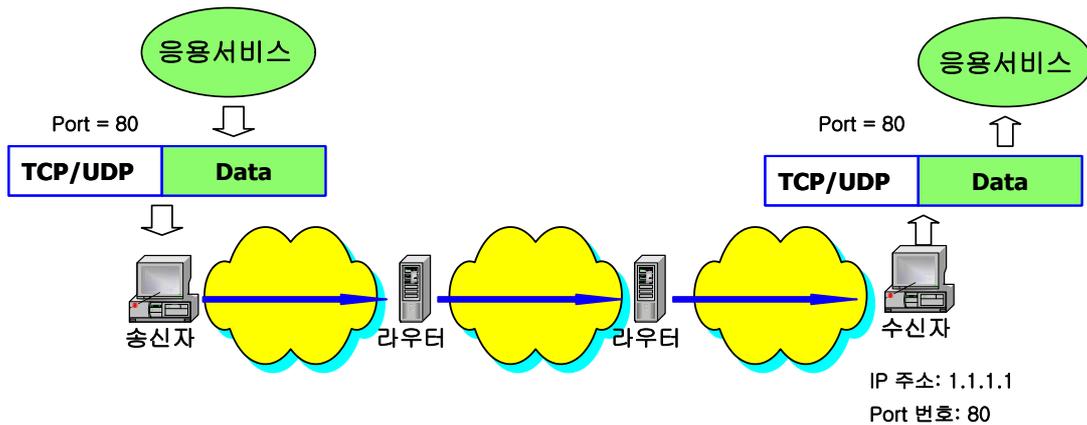


<그림 1.27> UDP 패킷 헤더 구조

### 1.7.2 포트번호

한편, UDP (TCP, SCTP 포함) 등의 수송계층 패킷의 헤더에는 2 바이트 크기의 포트번호가 포함되어 있으며, 이러한 포트번호는 컴퓨터에서 동작 중인 특정 응용서비스를 식별하는 데에 사용된다. 즉, 컴퓨터에는 웹서비스, 메일서비스 등의 다양한 응용서비스가 동시에 실행될 수 있는데, 그 중에 어떤 응용서비스에 대한 정보를 패킷이 포함하고 있는지를 나타내기 위해서 포트번호가 TCP, UDP 헤더에 포함되는 것이다.

다음 그림은 포트번호 80 번을 사용하는 웹서비스(WWW)에 대한 패킷전달 과정이다.



<그림 1.28> 포트 번호

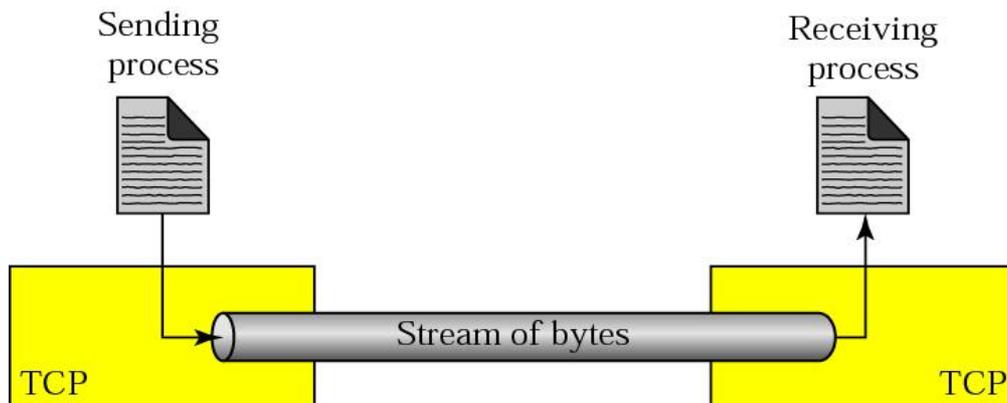
그림에서 송신자는 IP 주소 1.1.1.1 을 가지는 수신자(웹서버)에게 데이터를 전달하기 위해, TCP 헤더의 포트번호를 80 번으로 설정하여 패킷을 전송하고 있다. 해당 패킷이 라우팅 과정을 통해 수신컴퓨터에 도착하였을 때에, 수신컴퓨터는 포트번호가 80 번임을 확인하고 상위 웹 응용 프로그램에 데이터 내용을 전달한다.

포트번호는 이처럼 컴퓨터 도착한 패킷을 어떤 상위 응용 프로그램에 전달할지를 결정하기 위해 사용된다. 예를 들어, 메일서비스는 포트번호 25번을 사용하는 것처럼 응용서비스마다 다른 포트번호를 사용하며, 이러한 포트번호를 "well-known 포트번호"라 한다.

## 1.8 Transmission Control Protocol (TCP)

TCP는 가장 널리 사용되는 수송계층 프로토콜이다. UDP는 종단 프로세스간 통신을 위해 포트번호 정보만 제공하는 반면에, TCP는 연결(connection)관리, 오류(error) 제어 및 흐름(flow) 제어기능, 혼잡(congestion) 제어 기능 등의 상태(state)관리 기능을 제공한다.

UDP는 '메시지(message)' 단위의 데이터 전송을 수행하는 반면에, TCP는 일련의 바이트(byte) 스트림(stream)을 전송한다. 다음 그림은 종단 프로세스간에 TCP를 통해 연속적인 바이트 스트림을 전송하는 기능을 보여준다. 이처럼, 송수신 단말간에 비교적 많은 데이터를 전송하거나 오랜 시간동안 연결 설정이 필요할 때에 TCP를 사용한다.

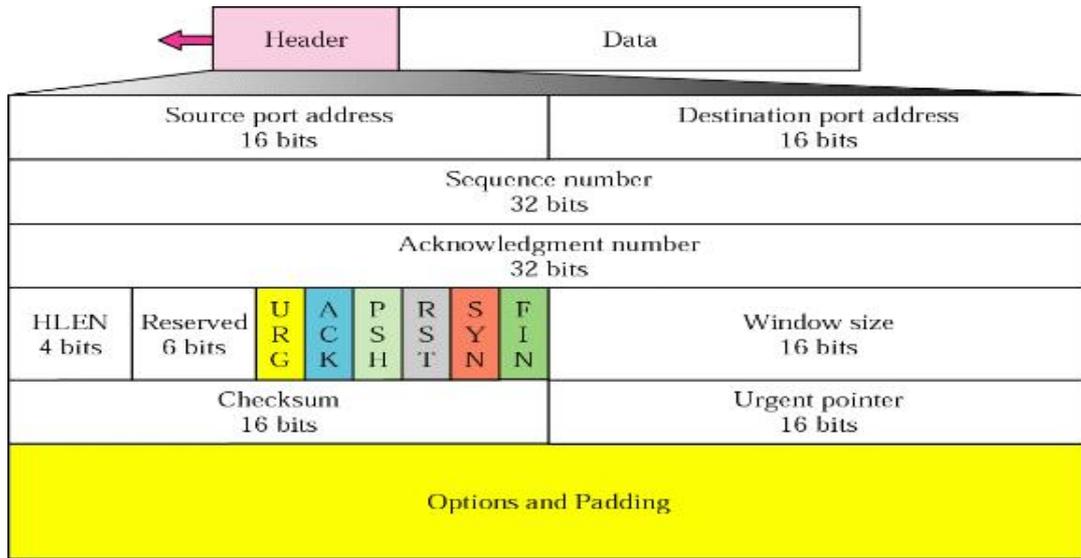


<그림 1.29> TCP 바이트 스트림 전송

위와 같은 바이트 스트림 전송을 위해 TCP는 UDP에 비해 많은 제어 기능을 필요로 한다. 아래에서는 TCP 제어 기능 및 이를 위한 헤더 구조를 살펴보기로 한다.

## 1.8.1 TCP 패킷 헤더

다음 그림은 TCP 헤더구조를 보여준다. 총 20 바이트의 기본 헤더에 필요에 따라 40 바이트까지 옵션이 추가될 수 있다.



<그림 1.30> TCP 헤더 구조

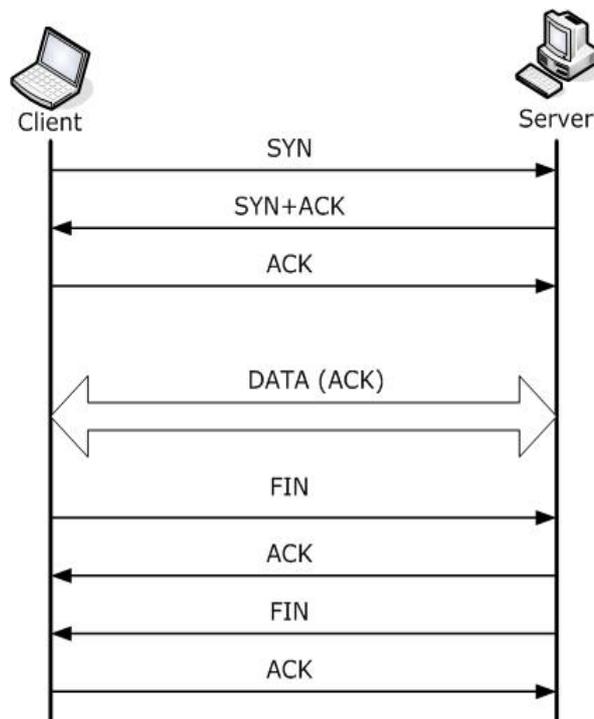
먼저, 첫줄의 4바이트는 UDP처럼 송수신 포트번호 기록을 위해 사용된다. 2번째와 3번째 줄은 TCP 오류, 흐름, 혼잡제어를 위해 사용되는 중요한 정보로서, 바이트 단위의 일련번호(sequence number)를 기록한다. 4번째 및 5번째 줄에는 각종 제어기능에 필요한 정보 및 헤더크기, 그리고 수신 컴퓨터의 버퍼(buffer) 용량을 나타내는 윈도우(window) 정보 등이 있다.

주요 필드(field)에 대한 설명은 다음과 같다.

- 송신(source) 및 수신(destination) 포트번호: 각 2 바이트(16비트)
- Sequence number: 송신자 입장에서, 해당 TCP 패킷의 데이터 부분에 포함된 첫 번째 데이터의 일련 번호(바이트 단위). 즉, 전체 데이터 중에 해당 패킷이 포함하는 데이터 정보를 나타냄
- Acknowledgment number: 수신자 입장에서, 송신자가 보낸 데이터 중에 어느 부분까지 성공적으로 수신하였는지를 관련 일련번호(바이트 단위)로 나타냄. TCP 오류제어에 사용됨
- HLEN: 옵션을 포함한 TCP 헤더의 크기 (20~60바이트; 4바이트 word 단위)

## 1.8.2 TCP 연결 관리

UDP와 달리 TCP은 연결관리 기능을 제공하며, 이를 통해 데이터 송수신 상태(state)를 지속적으로 관리한다. 아래 그림은 TCP 연결설정 및 연결해제 과정을 보여준다.



<그림 1.31> TCP 연결 관리

TCP 연결 관리는 크게 다음 3가지 과정으로 나뉜다.

### 1) 연결 설정(connection establishment)

연결을 원하는 client가 먼저 SYN(synchronization) 패킷을 전송하면, server는 이에 대한 ACK(acknowledgment) 정보와 자신의 SYN 정보를 담은 패킷을 전송한다. 이어서 client는 서버의 SYN에 대한 ACK를 전송함으로써 연결 설정을 완료한다. 이처럼 SYN, SYN+ACK, ACK 3개의 패킷을 교환하므로 TCP 연결과정을 3-way handshake 과정이라 부른다.

### 2) 데이터 송수신(data transfer)

연결설정 후에 client와 server는 데이터 패킷과 상응하는 ACK 패킷을 통해 데이터 전송을 수행

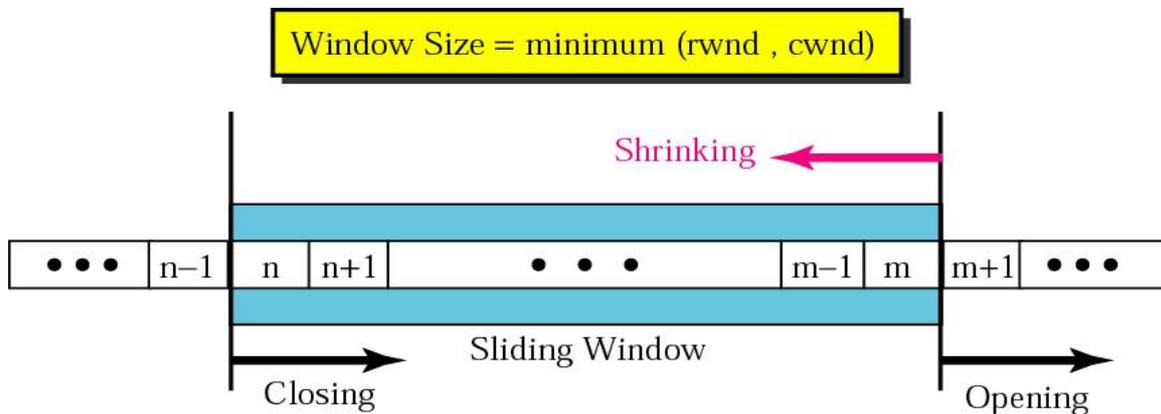
### 3) 연결 종료(connection shutdown)

연결 종료를 위해 client는 FIN(finish) 패킷을 전송하고 server는 ACK로 응답한다. 이어서, server가 FIN 패킷을 전송하면 client가 ACK로 응답하여 연결을 종료한다. 연결 종료 후에는 더 이상 데이터 전송을 수행할 수 없으며 관련된 모든 상태 정보가 메모리에서 해제된다. 이러한 과정을 4-way handshake 과정이라 부른다.

### 1.8.3 TCP 오류 및 흐름 제어

TCP는 UDP와 달리 송신한 패킷이 손실(loss)되었을 경우, 재전송(retransmission)을 통해 오류복구기능을 제공한다. 이러한 오류제어 기능은 수신자의 버퍼 용량을 고려하는 흐름(flow) 제어방식과 함께 수행되는 데, 이를 "sliding window" 기반 오류/흐름 제어방식이라 한다.

아래 그림은 sliding window 방식에 따른 송신버퍼에 있는 데이터(패킷)의 흐름을 보여준다.



<그림 1.32> TCP 오류 및 흐름제어를 위한 sliding window

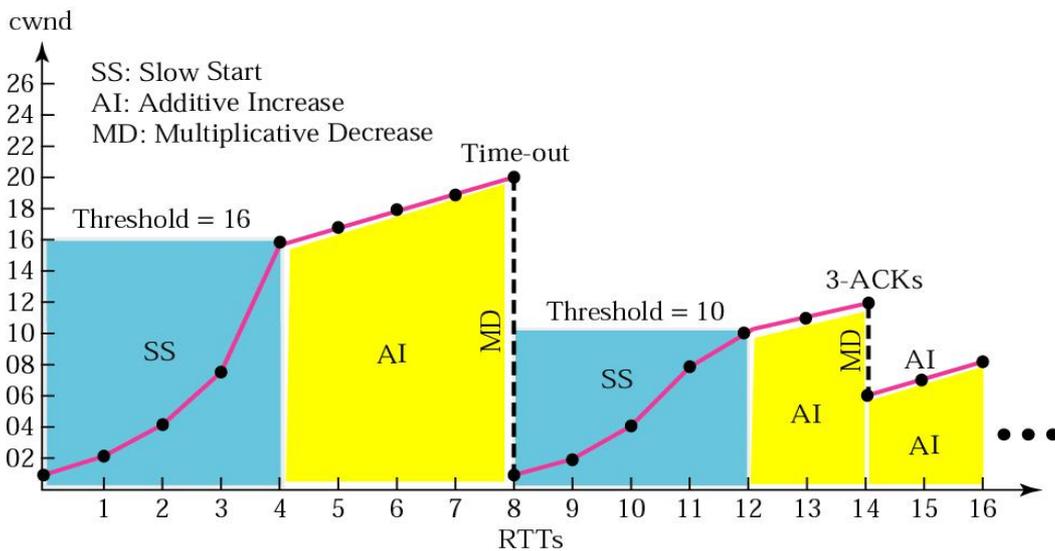
그림에서 n-1 번째 바이트까지의 데이터는 이미 전송이 완료된 상태이고, 송신자는 추가적으로 "n ~ m 번째" 데이터를 보낼 수 있으며, 이를 window 크기라 한다. 이러한 window 크기는 수신자의 버퍼용량(receiver window: rwnd)와 congestion window(cwnd) 크기의 최소값으로 결정된다. cwnd 크기는 다음에 기술하는 혼잡제어 알고리즘에 의하여 결정되는 값이다.

일단 송신된 데이터에 대하여 수신자로부터 해당 ACK 패킷이 (TCP 헤더의 ACK number를 이용) 도착하면, 전송이 완료된 것으로 간주하고 window를 전체적으로 오른쪽으로 이동시킨다(이를 'sliding'이라 함). window 크기에 따라 송신자는 새로운 데이터들을 더 보낼수 있다. 만약 제한된 시간 내에 ACK 패킷이 오지 않는 경우, 송신자는 해당 패킷을 '재전송'함으로써 오류 복구기능을 수행한다. 송신 Window의 크기는 혼잡제어 및 수신버퍼(rwnd) 용량에 따라 축소(shrinking) 될 수도 있다.

### 1.8.4 TCP 혼잡 제어

혼잡제어는 오류/흐름제어와 함께 TCP의 주요 특징 중의 하나이다. 오류/흐름제어는 종단 송수신자간에 손실 패킷의 재전송 및 수신버퍼 용량을 고려한 안전한 혹은 신뢰적인(reliable) 전송에 중점을 두는 반면에, 혼잡제어(congestion control)에서는 네트워크의 상태에 따라 데이터 송신량을 조절함으로써 전송의 효율성(throughput)을 높이는 방식이다.

구체적으로 TCP 혼잡제어에서는 혼잡윈도우(congestion window: cwnd) 값을 조절하게 되는데, 네트워크가 혼잡상태에 있을 때에는 cwnd를 낮추고, 정상적인 상태에 있을 때에는 cwnd를 높여주는 알고리즘이다. 다음 그림은 TCP 혼잡제어에 따른 cwnd 변화를 보여준다.



<그림 1.33> TCP 혼잡제어

혼잡제어 과정을 다음 3가지 모드(mode)로 나누어 cwnd 값을 달리 조절한다.

- SS(slow start) 모드: cwnd=1(여기서 1은 패킷 개수를 의미함)로 시작하고, 수신측에서 ACK가 도착하면 2배씩 cwnd를 증가시킨다.
- AI(additive increase) 모드: cwnd값이 threshold 값을 넘게 되면, AI 모드에 들어가며 이때는 ACK 패킷당 '1'씩 cwnd를 증가시킨다.
- MD(multiplicative decrease) 모드: 전송 도중에 손실(loss) 혹은 3개의 중복 ACK가 도착하는 경우, 네트워크 혼잡상태로 판단하고 cwnd 값을 급격히 감소시킨다.

TCP 혼잡제어에서는 이처럼 cwnd 값 조절을 통해, 최대한 안전하게 throughput을 높이고자 하며, 보다 상세한 혼잡제어에 대한 설명은 다른 문헌을 참고하기 바란다.

## 1.9 Stream Control Transmission Protocol (SCTP)

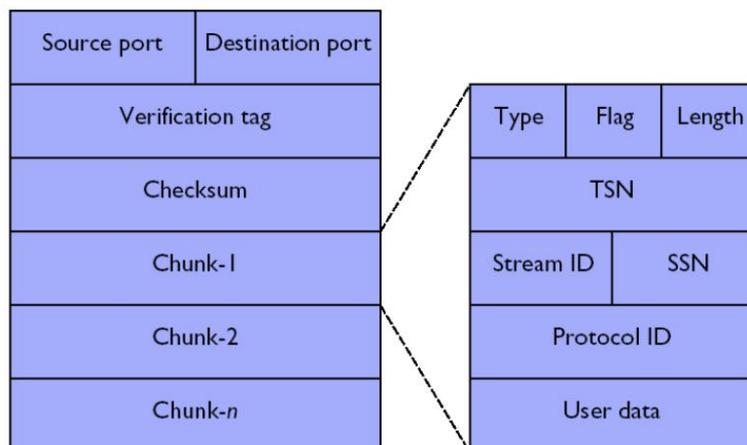
SCTP는 TCP, UDP에 이어서 3번째로 개발된 수송계층 프로토콜이다. 기본적인 프로토콜 특징은 TCP와 유사하며 다음과 같은 특징을 갖는다.

- TCP 3단계 연결설정 절차 대신에, SCTP는 4 단계 연결설정 절차를 따른다. 즉, 연결설정을 위해 client와 server간에 INIT, INIT-ACK, COOKIE-ECHO, COOKIE-ACK 4개의 패킷교환이 이루어진다. 이는 TCP의 "SYN storm" (정보보안의 DoS 공격에 해당하는 것으로 서버를 다운시키기 위해 공격자가 TCP SYN 패킷을 대량으로 발송하는 행위) 문제 해결을 위해, "cookie" 정보에 대한 확인 절차 위해 4단계가 필요하다.
- TCP 4단계 연결종료 절차 대신에, SCTP는 3단계 절차를 수행한다. 즉, SHUTDOWN, SHUTDOWN-ACK, SHUTDOWN-COMplete 패킷 교환을 통해 연결을 종료한다.
- TCP는 바이트 기반의 스트림 전송방식인데 비하여, SCTP는 UDP와 유사하게 메시지 기반의 전송 방식을 따른다. 이러한 메시지를 SCTP 청크(chunk)라 부른다. 즉, sequence number 등이 바이트 단위가 아닌 chunk 메시지 단위가 된다.

SCTP는 멀티스트리밍(multi-streaming) 기능과 멀티홈잉(multi-homing) 기능을 제공한다.

### 1.9.1 SCTP 멀티스트리밍

SCTP는 하나의 연결(association이라 부름)에서 여러 개의 스트림(stream) 데이터를 전송할 수 있다. 즉, 데이터가 여러 종류의 스트림으로 분류될 수 있는 경우에 각 스트림 데이터를 분리하여 전송 할 수 있다. 다음 그림은 SCTP 패킷 구조를 보여준다.



<그림 1.34> SCTP 패킷 구조와 멀티스트리밍

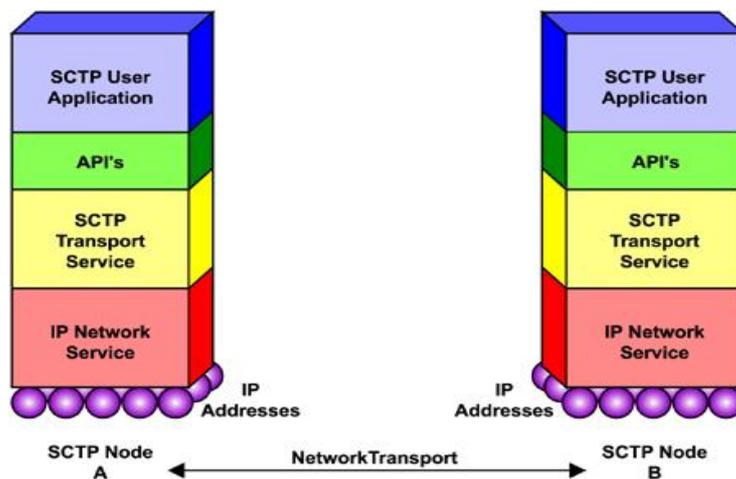
SCTP는 12바이트(포트번호: 4바이트, 검증태그: 4바이트, 체크섬: 4바이트)의 고정 헤더와 다수의 청크(제어 청크와 데이터 청크 포함)로 구성된다. 제어(control) 청크에는 INIT, INIT-ACK, COOKIE-ECHO, COOKIE-ACK 등의 연결설정에 사용되는 청크와, SHUTDOWN, SHUTDOWN-ACK, SHUTDOWN-COMPLETE 등의 연결종료에 사용되는 청크 및 그 외 HEARTBEAT 청크 등의 연결제어에 사용되는 다양한 청크가 있다.

데이터 전송을 위해서는 DATA 청크가 사용되고, DATA 청크는 16바이트의 데이터청크 헤더를 사용하고 다음 필드를 포함한다.

- TSN(Transmission Sequence Number): TCP의 sequence number에 해당되며, SCTP는 DATA 청크별로 TSN 값이 1씩 증가한다. TSN은 오류 및 혼잡제어 등에 사용된다.
- Stream ID 및 SSN(Stream Sequence Number): SCTP의 멀티스트리밍 특성을 지원하기 위한 것으로, 하나의 연결에 여러 스트림 데이터가 각각 별도의 Stream ID를 가질 수 있으며, SSN은 해당 Stream ID에서의 DATA 청크의 일련번호이다. 즉, Stream 별로 고유의 일련번호가 부여되어, 수신측에서 응용 프로세스에 전달된다.

### 1.9.2 SCTP 멀티호밍

멀티호밍(multi-homing)은 SCTP의 독특한 특징 중의 하나로서, 하나의 연결에서 IP 주소를 두 개 이상 사용할 수 있다. 즉, TCP, UDP 연결의 경우 하나의 IP 주소와 하나의 포트로 식별되는 데 비해, SCTP 연결은 하나의 포트와 두 개 이상의 IP 주소를 동시에 사용할 수 있다. 이러한 멀티호밍 특성을 통해 중단 단말간에 2개 이상의 경로(path)를 사용할 수 있으며, 하나의 경로가 장애(failure)를 일으키는 경우, 곧 바로 다른 경로를 통해 데이터 전송을 유지하게 해주는 기능을 제공한다.

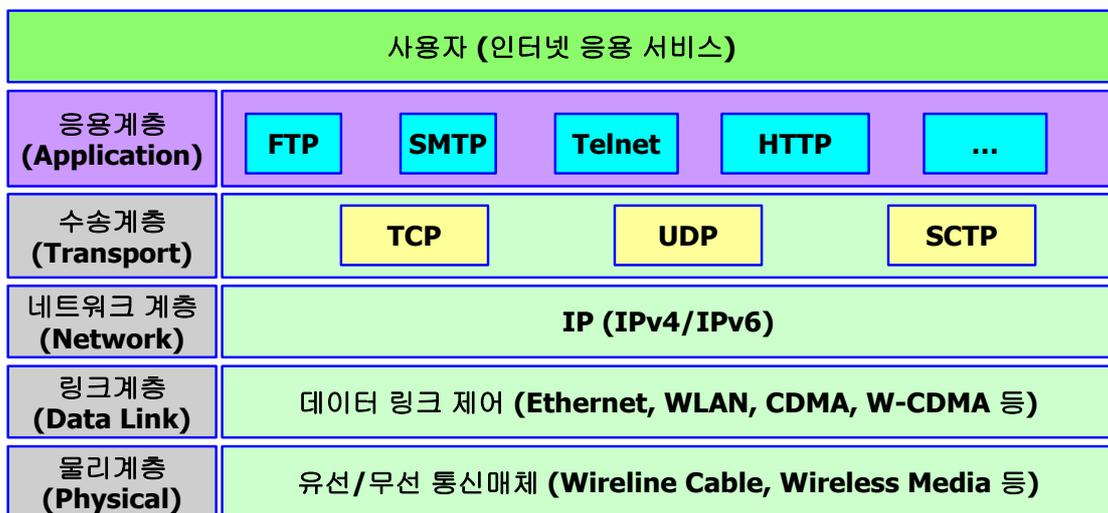


<그림 1.35> SCTP 멀티호밍

## 1.10 응용 계층 프로토콜

TCP/IP 패킷은 사용자에게 다양한 인터넷 응용 서비스를 제공하기 위한 데이터 정보를 포함한다. 각 데이터는 상위 응용서비스에 따라 다른 정보 혹은 다른 포맷으로 구성되는데, 이처럼 “응용서비스별로 어떻게 데이터를 구성하고 전송해야 하는지”를 정의한 프로토콜이 응용 계층 프로토콜이다. 즉, 각 응용서비스마다 고유한 응용 프로토콜이 정의된다.

다음 그림은 TCP/IP 프로토콜 스택에서의 응용 계층 프로토콜의 위치이다. 그림 8.23에서처럼, 응용 계층 프로토콜은 상위에 있는 응용서비스 사용자와 하위에 있는 수송계층 프로토콜과의 중간 인터페이스 기능을 제공한다.



<그림 1.36> TCP/IP 응용 프로토콜

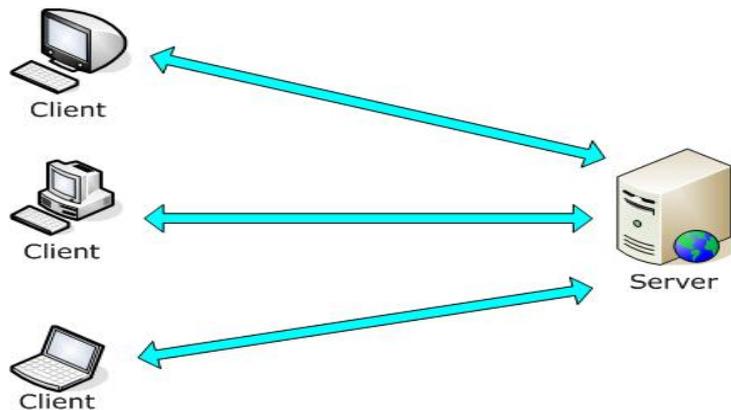
예를 들어, 우리가 지금까지 자주 접해온 인터넷 서비스들은 다음과 같이 상응하는 응용 계층 프로토콜을 기반으로 동작한다.

- 웹서비스(WWW): HTTP (HyperText Transfer Protocol)
- 메일서비스: SMTP (Simple Mail Transfer Protocol)
- 파일전송서비스: FTP (File Transfer Protocol)

### 1.10.1 클라이언트-서버 모델

우리가 지금까지 접해온 대부분의 인터넷 응용서비스들은 클라이언트-서버(Client-Server) 모델에 따라 동작한다. 예를 들어, WWW 웹서비스를 위해 웹클라이언트와 웹서버가 사용되며, 메일 서비스를 위해 메일 클라이언트와 메일 서버가 동작한다.

다음 그림은 client-server 모델의 개념을 보여준다.



<그림 1.37> Client-Server 모델

WWW 웹서비스에서 알 수 있듯이, client는 일반적인 서비스 사용자를 의미하고, server는 서비스를 제공하는 사업자를 의미한다. client는 필요할 때에 server에 접속하여(연결하여) 원하는 서비스를 받은 다음에 연결을 종료하는 반면에, server는 1년 365일 동안 불특정 다수의 client 접속을 대기하고 있어야 한다. 대개 server는 다수의 client를 상대하게 된다.

한편, C-S 모델에 대응되는 모델은 Peer-to-Peer (P-P) 모델이다. P-P 모델에서는 두 대의 단말이 동등한 입장에서 통신을 하게 되며, 사실상 각 단말이 client 역할을 하기도 하고 server 역할을 하기도 한다. 대개 먼저 접속을 요청하는 단말이 client 역할을 수행한다. P-P 모델의 전형적인 예제는 VoIP(Voice over IP) 전화 서비스이다.

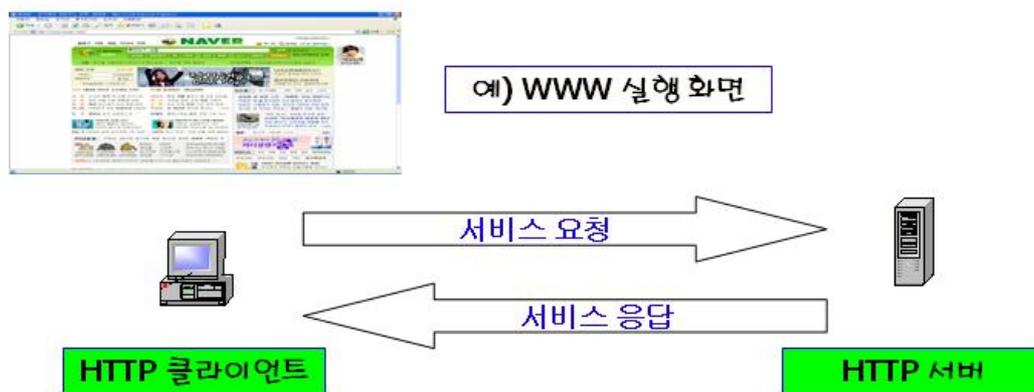


<그림 1.38> Peer-to-Peer 모델

### 1.10.2 HTTP

먼저 우리가 가장 많이 사용하는 WWW 혹은 웹서비스는 HTTP(HyperText Transfer Protocol)을 사용한다. HTTP 프로토콜은 말 그대로 하이퍼텍스트 정보를 전송하는 프로토콜이다.

다음 그림은 HTTP 프로토콜을 이용한 WWW 문서의 전달과정이다.



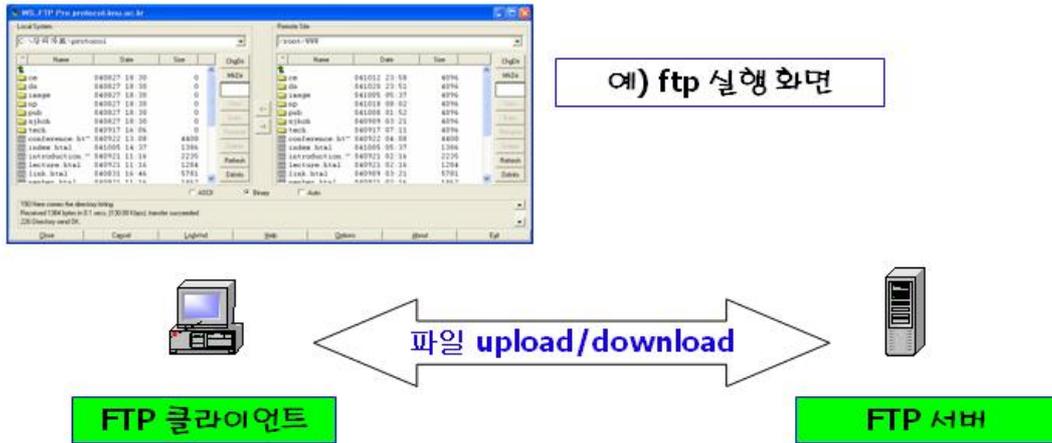
<그림 1.39> HTTP 기반 WWW

우리가 사용하는 인터넷 익스플로러에는 HTTP 클라이언트 프로그램이 동작한다. 또한 포털사이트 혹은 웹서버에는 HTTP 서버 프로그램이 동작하고 있다. 우리가 웹 브라우저를 사용하여 웹서버로부터 웹문서를 얻기까지의 과정을 정리하면 다음과 같다.

- 웹 브라우저는 HTTP 프로토콜을 사용하여 웹 서버에게 서비스(웹문서)를 요청한다. 요청 문서의 위치는 `http://www.naver.com/index.html`와 같이 URL 로 표시된다.
- 웹 서버는 해당 URL 문서를 HTTP 프로토콜을 사용하여 웹 브라우저에게 전송한다. HTTP 프로토콜의 데이터 부분에 해당 문서 정도가 포함된다.

### 1.10.3 FTP

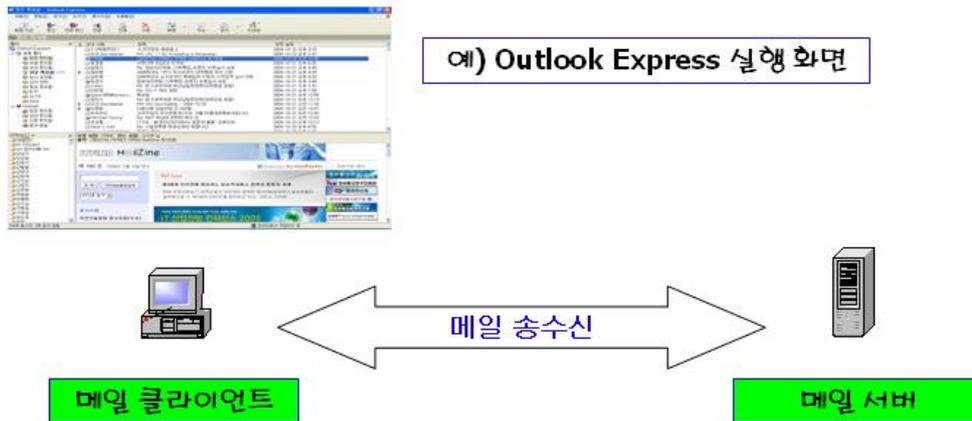
다음 그림은 FTP(File Transfer Protocol)의 클라이언트와 서버간의 동작 모습이다. FTP 클라이언트 프로그램을 사용하여 FTP 서버에 접속한 후 파일을 업로드나 다운로드하는 과정을 볼 수 있다. FTP는 20번 및 21번의 well-known 포트번호를 사용한다.



<그림 1.40> FTP

#### 1.10.4 SMTP

다음 그림은 SMTP(Simple Mail Transfer Protocol)의 동작 모습이다. SMTP 프로토콜은 메일 클라이언트가 서버에게 메일 전송을 요청할 때에 사용된다. 반면에 클라이언트가 자신의 메일을 서버로부터 가져오기 위해서는 POP(Post Office Protocol) 혹은 IMAP(Internet Message Access Protocol) 등의 별도 프로토콜을 사용한다



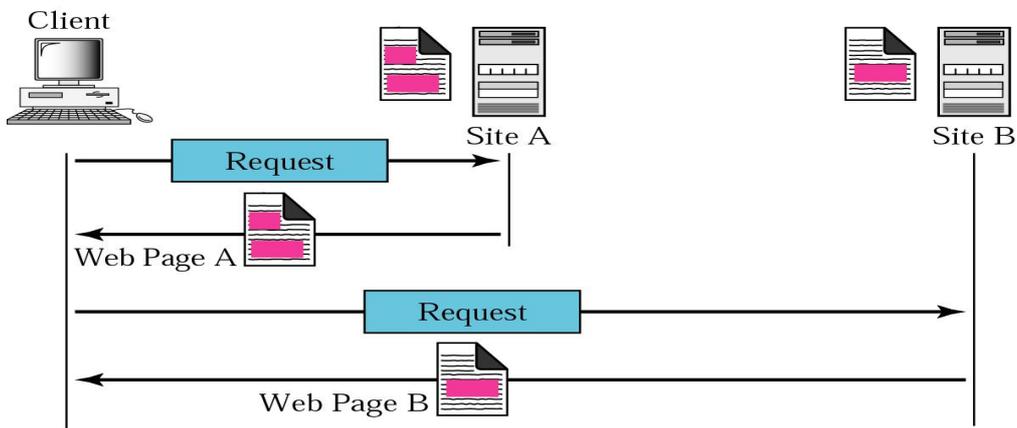
<그림 1.41> SMTP

## 제 2 장. 웹 프로그래밍 개요

### 2.1 WWW과 HTTP 프로토콜

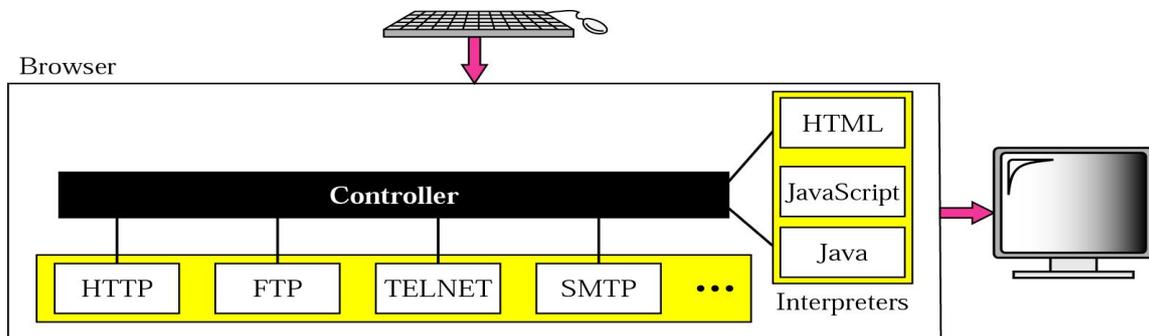
#### 2.1.1 WWW 구조

WWW 혹은 웹서비스는 Client-Server 구조를 따른다. 웹 Client는 웹서버에게 WWW 웹문서를 요청(Request)하면, 웹서버는 해당(URL) 웹문서로 응답한다. 웹 클라이언트는 여러 웹서버와 동시에 통신을 수행할 수 있다.



<그림 2.1> WWW 서비스 구조

웹 Client는 웹 브라우저(Browser)로 구현된다. 아래 그림과 같이 웹 브라우저는 HTTP, FTP, SMTP 등의 인터넷 응용 프로토콜을 사용하여 웹서버로부터 WWW 문서를 (HTML 등) 가져오고, 자바스크립트 등의 프로그래밍 기법(혹은 해석기)을 사용하여 사용자의 화면에 해당 결과물을 출력해 준다.



<그림 2.2> Web Browser

## 2.1.2 WWW 문서의 종류

WWW 문서는 크게 다음 3가지로 구분해 볼 수 있다.

- 1) 정적(static) 문서: 예) HTML
- 2) 서버측(server-site) 동적(dynamic) 문서: 예) JSP, ASP, PHP, CGI 등
- 3) 클라이언트측(client-site) 동적 문서: 예) JavaScript, CSS, 자바 애플릿(applet) 등

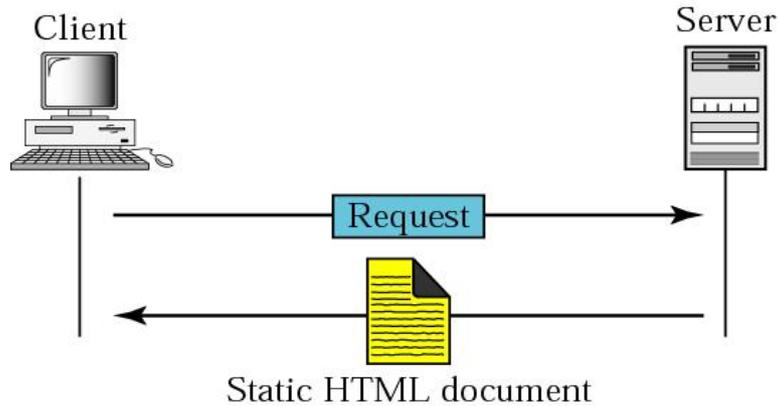
상기와 같이 웹문서는 “문서의 내용이 변경되는지 여부”에 따라 정적문서와 동적문서로 나뉘고, 동적문서의 경우 “최종적으로 문서의 내용이 결정되는 위치”에 따라 “서버측 동적문서”와 “클라이언트측 동적문서”로 구분해 볼 수 있다.

다음 표는 3가지 종류의 웹문서의 특징을 비교하고 있다.

<WWW 문서의 종류>

구분	정적 문서 (Static Document)	서버측 동적 문서	클라이언트측 동적 문서
대표 기술	HTML	ASP, JSP, PHP	Java Script, Java Applet, CSS
처리 방식	수동적, 고정적	사용자의 입력을 실시간 처리	사용자 환경에 따라 동적 처리
해석 위치	-	서버	클라이언트
활용 분야	내용이 변하지 않는 일반 웹문서	사용자의 입력이 요구되는 게시판, 설문조사, 회원관리 등	사용자의 환경에 따라 다른 웹문서를 출력하는 경우

다음 그림은 정적 웹문서의 동작원리를 보여준다. 서버에 있는 문서가 변경되지 않고 Client에 전달되고, 그대로 웹브라우저의 화면에 출력된다. HTML 문서가 대표적인 정적문서의 예이다.



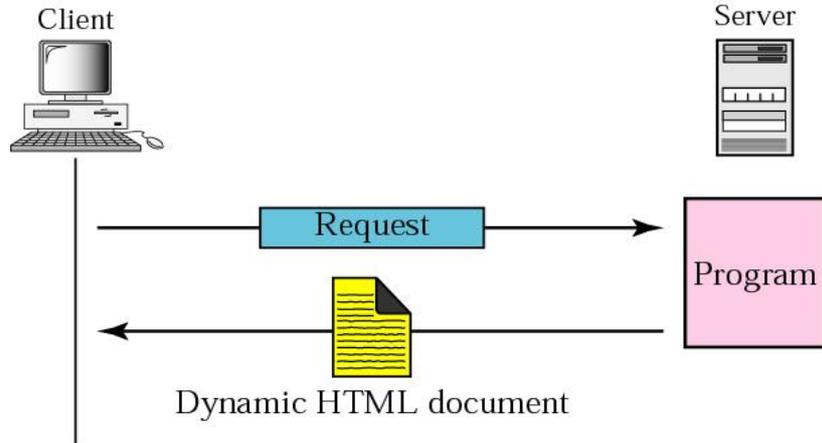
<그림 2.3> 정적 웹문서

서버측 동적문서의 경우, Client의 요청(Request)메시지가 도착하면, 서버에서 별도의 프로그램 혹은 스크립트(script)를 실행하여 웹문서를 생성한다. CGI 문서의 경우 서버에서 CGI 프로그램을 실행하고, 기타 ASP, JSP, PHP 등에서는 관련 스크립트를 실행하여 웹문서를 생성한다.

○ 메모

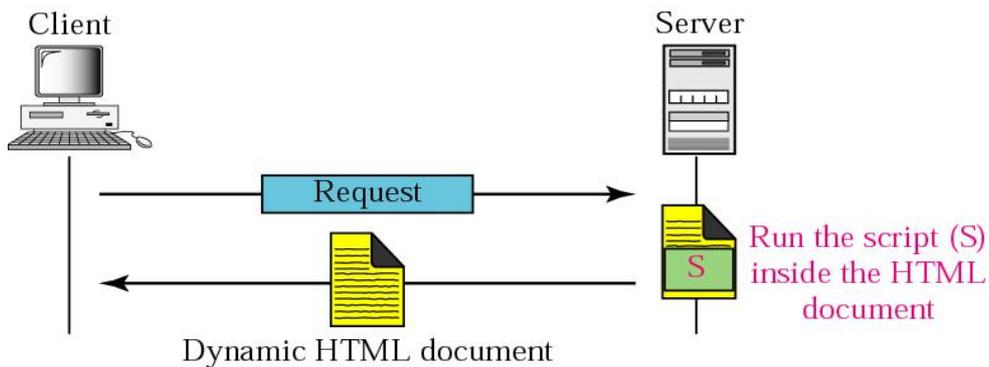
- 동적 문서의 경우, Client가 접속하는 시간마다 다른 웹문서가 생성될 수 있다. (예: 방문자 수를 보여주는 counter 프로그램)
- ASP(Active Server Page) 및 JSP(Java Server Page) 프로그래밍 언어에서 "서버 페이지(server page)"란 말은 "서버에서 웹문서를 생성시킴"을 의미한다.

다음 그림은 CGI 프로그래밍을 사용하는 "서버측 동적문서"의 예를 보여준다. 그림에서 웹서버는 CGI 프로그램을 실행시켜 웹문서를 생성하고, 생성된 문서를 client에 전송한다.



<그림 2.4> 서버측 동적문서 (프로그램 사용)

다음 그림은 스크립트를 사용하는 “서버측 동적문서”의 예를 보여준다, 그림에서, 프로그램 대신에 서버측 스크립트를 실행하여 웹문서를 생성하고 client에 전송한다. ASP, JSP, PHP 등이 이에 해당한다.

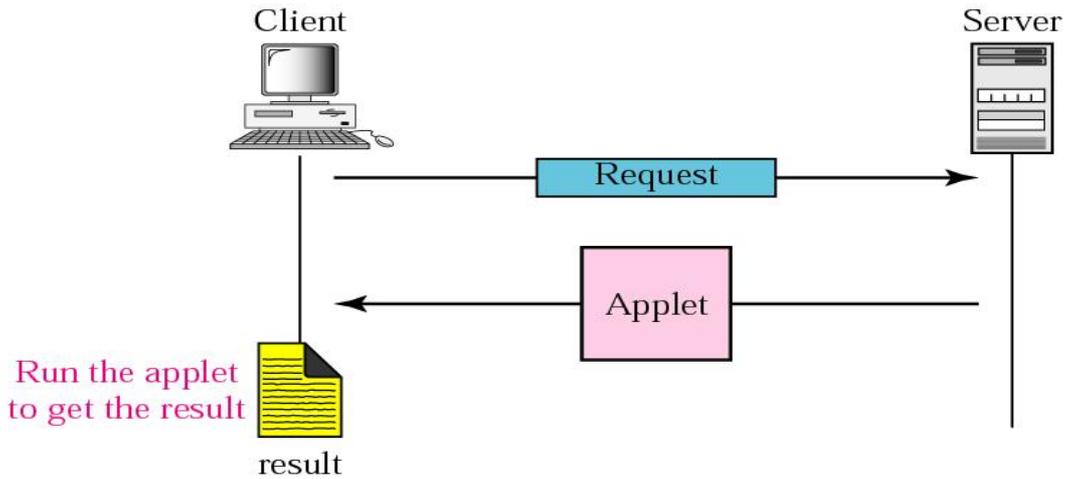


<그림 2.5> 서버측 동적문서 (스크립트 사용)

○ 메모

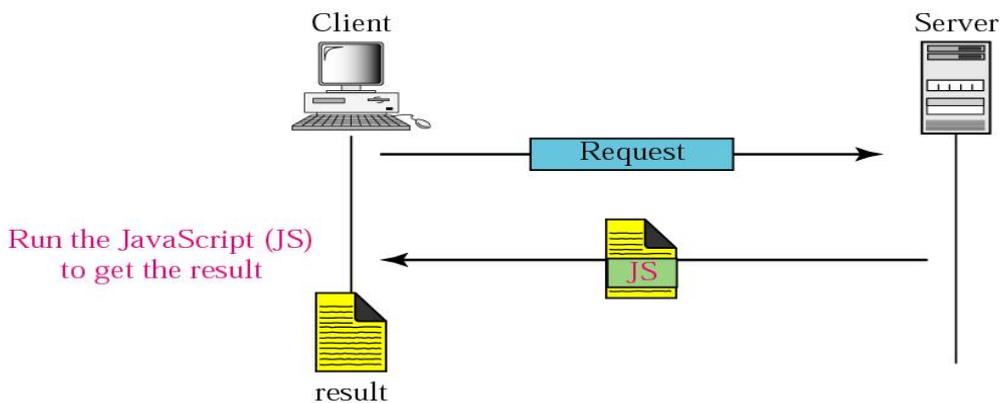
- 이때 사용되는 서버측 스크립트언어는 웹프로그래밍 언어에 따라 다를 수 있으며, ASP의 경우 Visual Basic Script, JSP의 경우 JavaScript를 사용한다.
- 앞서 얘기한 CGI 프로그래밍의 경우 “프로그램 실행”으로 서버에 부담(overload)을 많은 주는 단점이 있다. ASP, JSP 등의 스크립트 기법은 이러한 CGI 프로그래밍의 단점을 해결하기 위해 개발되었다.

다음은 Client측 동적문서에서 Java Applet 등의 프로그램을 사용하는 예제이다. 클라이언트의 요청에 의해서 웹서버는 Applet 등의 "프로그램"을 전송하고, 해당 프로그램이 클라이언트 측에서 실행되어 결과가 출력된다.



<그림 2.6> Client 측 동적문서 (프로그램 사용)

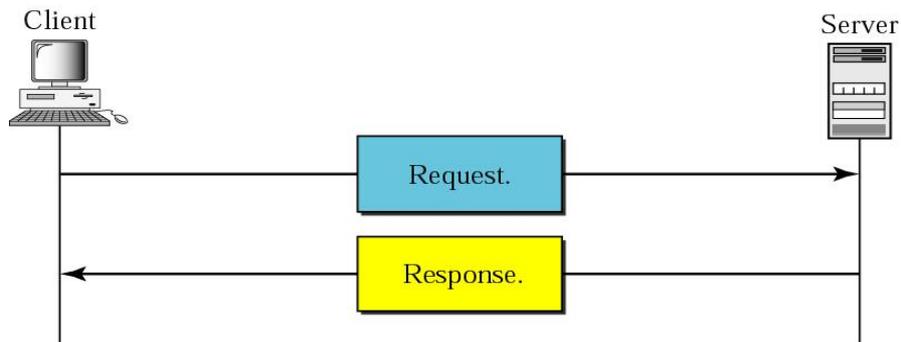
다음 그림은 "스크립트"를 사용하는 Client측 동적문서의 예이다. 클라이언트의 요청에 의해서 웹서버는 스크립트를 (예: Java Script, VB Script) 전송하고, 클라이언트는 해당 스크립트를 실행하여 웹문서를 화면에 출력한다.



<그림 2.7> Client 측 동적문서 (스크립트 사용)

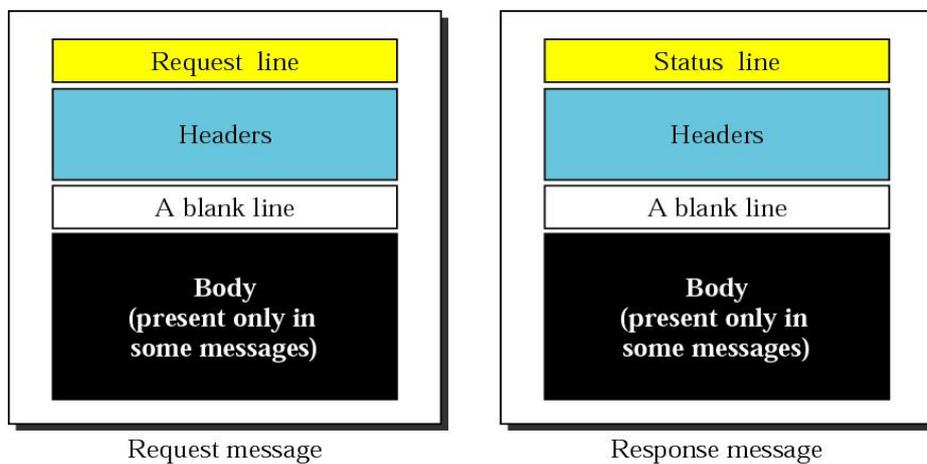
### 2.1.3 HTTP 프로토콜

웹문서는 Client와 서버사이에 HTTP 프로토콜을 사용하여 전달된다. Client는 Request(요청) 메시지를 통해 웹문서를 요청하고, 서버는 Response(응답) 메시지를 통해 해당 웹문서를 전달한다.



<그림 2.8> HTTP 동작 방식

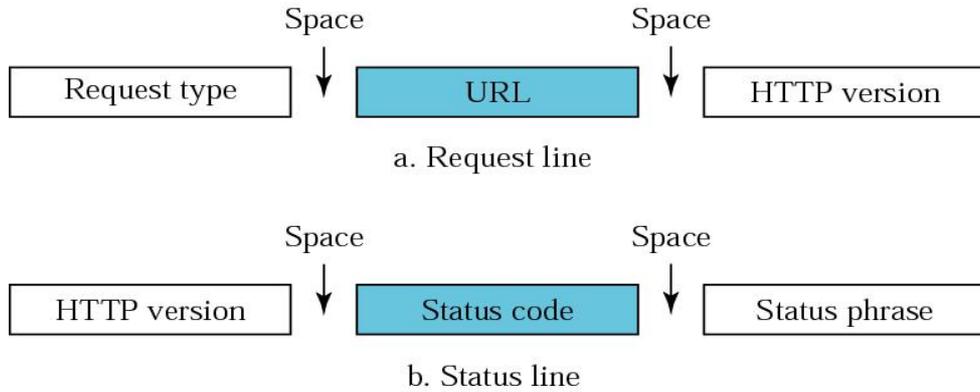
여기서 요청메시지와 응답메시지의 구성내용을 상세히 살펴보면 다음 그림과 같다.



<그림 2.9> HTTP 요청 및 응답메시지

그림에서 앞 부분은 HTTP 프로토콜 제어(control)에 필요한 정보이고, 실제 웹문서의 내용은 본문(Body) 부분에 포함된다.

상기 그림에서 첫 번째 줄의 정보는 다음 형식으로 채워진다. Request line의 URL은 서버에 있는 웹문서의 주소를 의미한다.



<그림 2.10> HTTP 메시지 헤더

Request line에서 Request Type은 Method(메소드)라 불리는 데, HTTP 요청 패킷의 종류 및 역할을 정의한다. 주로 사용되는 Request 메시지의 종류는 다음과 같다.

<HTTP Request (Method) 종류>

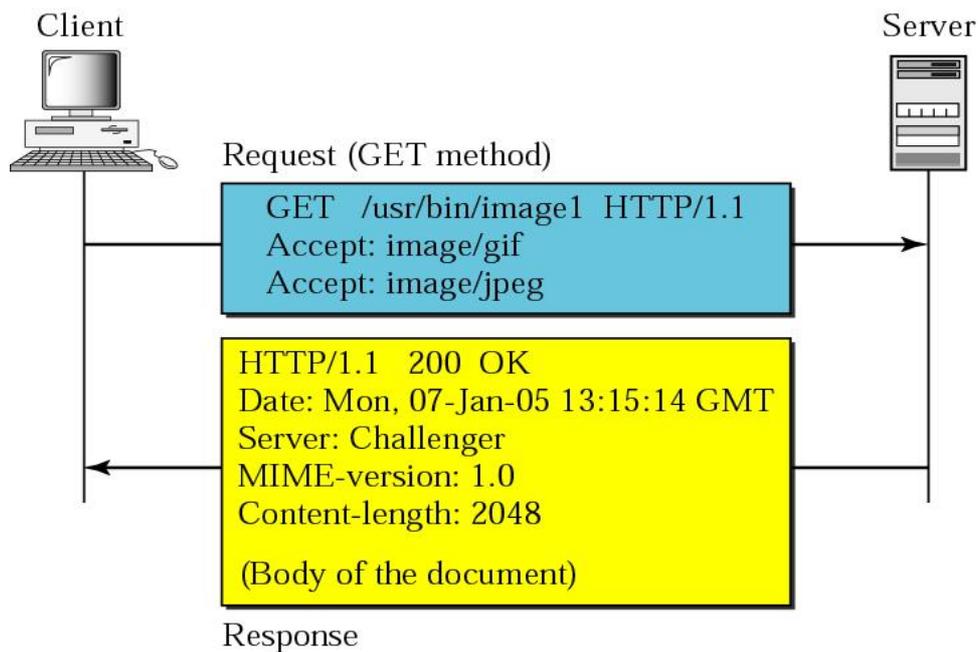
Method	의미
GET	서버에게 특정 웹문서를 요청함
POST	Client가 서버에게 어떤 정보를 보내는 경우에 사용함
OPTION	서버에서 지원되는 기능에 대하여 질문하는 경우에 사용함

한편, 서버가 응답하는 Response는 요청에 대한 상태(status) 코드를 포함하는데, 코드의 3자리의 숫자로 되어 있고, 첫 번째 숫자에 따라 다음의 의미를 갖는다.

<HTTP Response (Status Code) 종류>

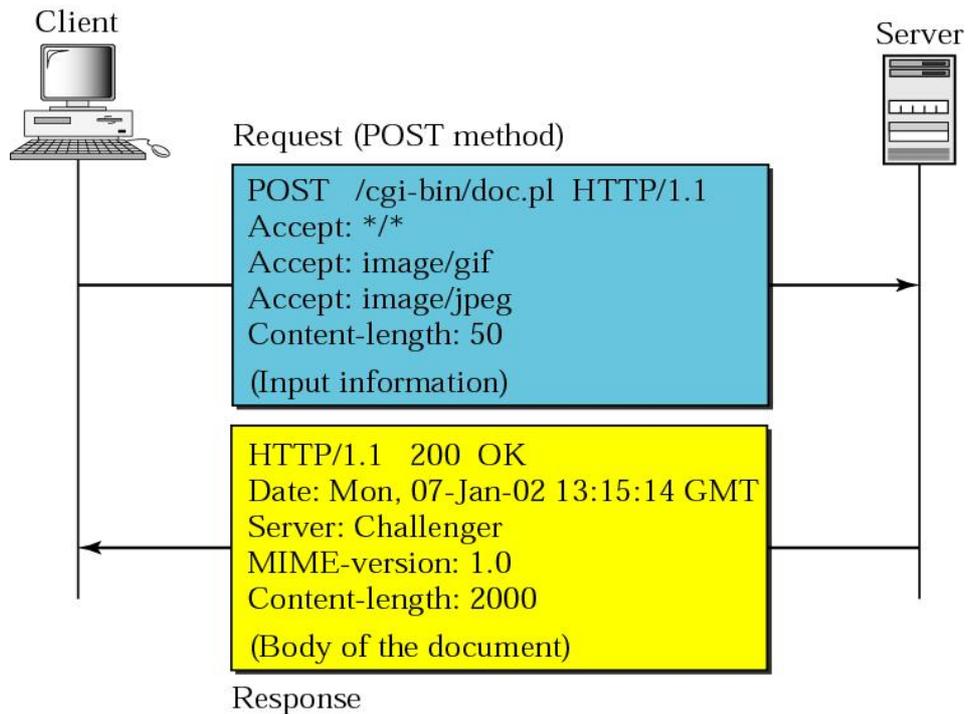
Code	의미
1XX	Informational
2XX	Success
3XX	Redirection
4XX	Client Error
5XX	Server Error

다음은 HTTP 통신의 한 예제로서 GET Method 를 사용하는 경우이다.



<그림 2.11> HTTP 통신 예제 (GET 방식)

다음은 HTTP 통신의 한 예제로서 POST Method를 사용하는 경우이다.



<그림 2.12> HTTP 통신 예제 (POST 방식)

○ 질문: 상기 2가지 예제에서 GET 방식과 POST 방식의 차이점은 무엇인가?

## 2.2 마크업 언어와 웹브라우저

### 2.2.1 마크업 언어

우리는 웹 문서 제작을 위한 마크업언어(Markup Language)로서 HTML을 주로 학습할 것이다. HTML을 비롯한 웹 문서제작 표준 언어의 변천사를 간략히 정리하면 다음과 같다.

(1) SGML(Standard Generalized Markup Language)

- 1986 년 ISO 에서 지정한 문서 처리 표준의 하나이며, 프로그래밍 언어는 아니다.

(2) HTML(Hyper Text Markup Language)

- 1989 년도에 CERN 에 있던 "팀 버너스 리"가 개발

- 웹에서 사용하는 표준 언어로 HTML 4.0 은 HTML 의 최신 버전이다.

(3) XML(eXtensible Markup Language)

- 1996 년 W3C 에서 제안하였으며 SGML 의 장점과 HTML 의 장점을 수용하여 만든 언어다.

(4) XHTML (Extensible Hypertext Markup Language)

- HTML+ XML (XML 기반의 HTML)

상기 사항을 표로 정리하면 다음과 같다.

<웹 문서제작 언어>

구분	HTML	SGML	XML
태그 사용	사용자 정의 불가능	사용자 정의 가능	사용자 정의 가능,
문서의 재사용	불가능	가능	가능
적용 분야	단순한 구조의 문서 및 내용이 길지 않은 문서	방대한 내용과 구조를 요하는 기술적인 문서	SGML 과 동일
문서 작성	간단하고 용이함. 논리 구조 작성의 어려움	매우 복잡함	SGML 을 단순화시켜 편리하게 작성 가능

## 2.2.2 웹브라우저의 종류

HTTP 통신에서 HTTP 클라이언트의 기능은 웹브라우저에 구현되어 있다. 우리는 대부분 MS사의 Internet Explorer를 사용하고 있으나, 그 밖에도 다음과 같은 웹브라우저가 있다.

- NCSA의 모자이크
- 오페라 소프트웨어(Opera Software)의 Opera
- AOL의 넷스케이프 네비게이터
- 선 마이크로 시스템의 Hot Java
- 모질라의 Firefox

특히, 최근에 모질라에서 배포된 Firefox 웹브라우저의 경우 다음 웹사이트에서 구할 수 있다.

정식버전 다운로드	<a href="http://www.mozilla.or.kr/ko/firefox/all.html">http://www.mozilla.or.kr/ko/firefox/all.html</a>
베타버전 다운로드	<a href="http://www.mozilla.or.kr/ko/firefox/all-beta.html">http://www.mozilla.or.kr/ko/firefox/all-beta.html</a> <a href="http://www.mozilla.com/en-US/firefox/all-beta.html">http://www.mozilla.com/en-US/firefox/all-beta.html</a>

Firefox의 최신 베타버전인 "Firefox 3.0b4" 한국어 버전의 경우, 다음 웹사이트에 관련 설치방법 및 사용법을 정리하였다.

- Firefox 웹 브라우저와 Web Developer 확장 기능 사용법 (CPL-TR-08-03)
- <http://protocol.knu.ac.kr/tech/CPL-TR-08-03-firefox.pdf>

### 2.2.3 웹문서 제작 도구

본격적으로 웹프로그래밍 언어에 대하여 학습하기 전에, 웹문서 제작에 필요한 도구(준비물)에 대하여 알아보자.

먼저, <웹 프로그래밍 언어>로는 다음 종류가 있고, 본 교재에서 주로 다룰 내용들이다.

- HTML
- CSS
- 자바스크립트
- CGI, ASP, PHP, JSP

웹문서의 모양을 좀 더 세련되도록 하기 위한 <웹 디자인> 분야로는 다음이 주로 사용된다.

- 포토샵
- 플래시

그 밖에 웹문서 작성을 위한 <웹 에디터>에는 다음 종류가 있다.

<<소스 위주의 웹 에디터>>

- 메모장
- 에디터플러스

<<편집 위주의 웹 에디터>>

- 나모
- 드림위버

# 제 3 장. HTML

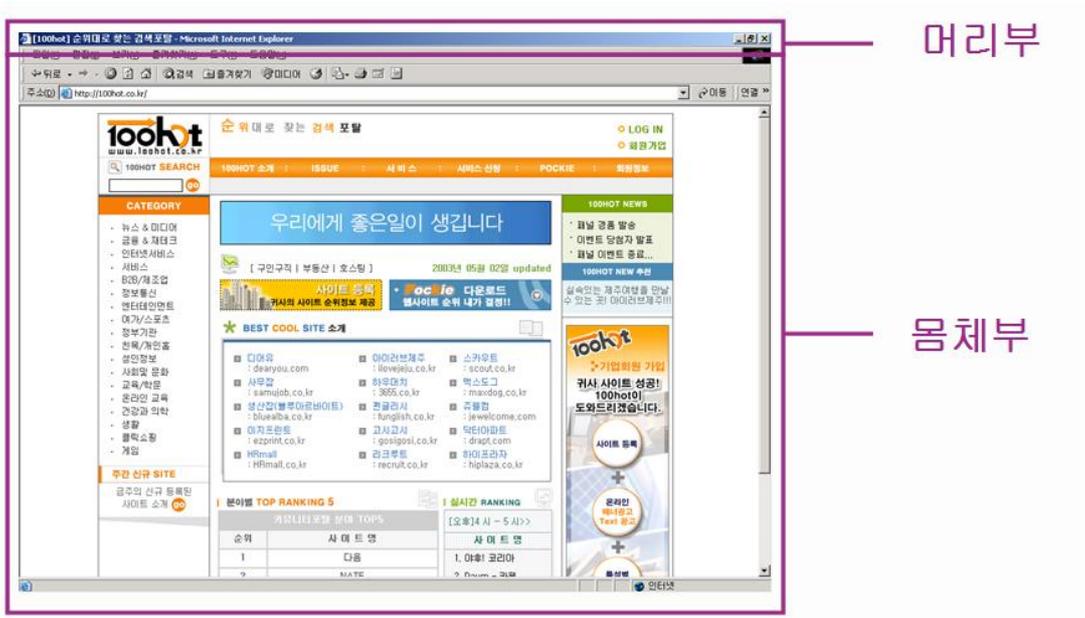
## 3.1 HTML 문서 구조

### 3.1.1 기본 구조

일반적인 HTML 문서의 기본 구조는 다음과 같다.

HTML 문서의 기본 구조		
<HTML>	_____	HTML 문서의 시작
<HEAD>	_____	HTML 문서의 머리 부분
<TITLE> 문서의 제목 </TITLE>	_____	HTML 문서의 제목
</HEAD>		
<BODY>	_____	HTML 문서의 몸통 부분
문서의 내용		
</BODY>		
</HTML>	_____	HTML 문서의 종료

상기 문서구조에 따라 브라우저에 출력되는 화면은 다음과 같다.



머리부의 <head> 태그 사이에 기록되는 문서 정보는 대개 다음을 포함한다.

- <title>: 홈페이지를 대표할 수 있는 문서 제목을 지정한다.
- <meta>: 다양한 속성을 가지고 있어 다양한 역할을 한다.
- <script>: 홈페이지를 역동적으로 변화시킨다.
- <style>: 문서 전체의 레이아웃을 결정하여 문서에 통일감을 준다.

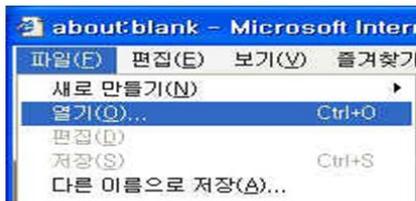
상기 태그의 상세한 사용법은 나중에 다시 학습하기로 하자.

먼저, 다음의 간단한 문서예제를 고려해보자.

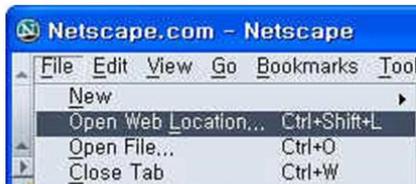
```
예제) test.html

<HTML>
  <HEAD>
    <TITLE>간단한 HTML 문서 작성 예제</TITLE>
  </HEAD>
  <BODY>
    HTML 문서는 머리와 몸통의 두 부분으로 구성되어
    있으며, 이를 정의하기 위하여 3개의 기본적인 태그가
    이용된다. 본 예제는 이러한 기본 태그를 사용하여 작성한
    것이다
  </BODY>
</HTML>
```

상기의 예제를 다음과 같이 브라우저에서 실행해보자.



익스플로러

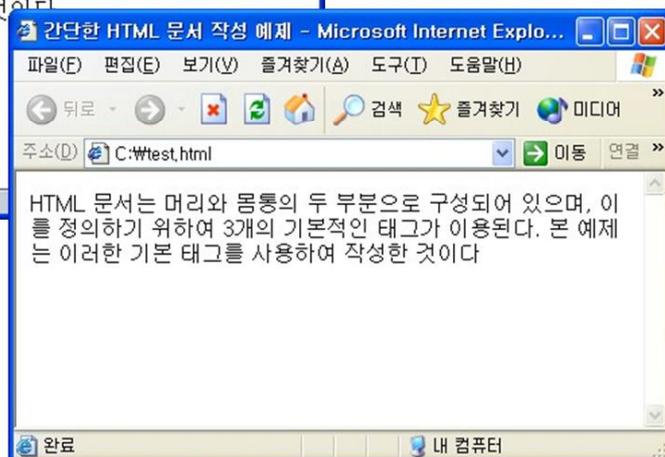


넷스케이프

상기 예제에 대한 브라우저 출력 결과는 다음과 같다.



넷스케이프



익스플로러

### 3.1.2 태그(tag)

HTML 에서 태그는 "문서의 구조와 형태를 표현하는 명령어"이며, "< >"를 사용하여 표현한다.

#### ○ 태그 사용 기본 원칙

1) 태그는 시작 태그와 종료 태그로 구성된다.

☞ 시작 태그 "<>"와 종료 태그 "</>"가 존재

☞ <태그명></태그명>과 같이 여는 태그와 닫는 태그가 쌍으로 사용된다.

2) 태그명과 "속성(attribute)"명, 속성명과 속성명 사이는 한 칸씩 띄운다.

☞ 그 외 공백을 두어서는 안 된다.

☞ 속성명과 값은 = 부호를 사용하여 표시한다.

☞ 값이 두 단어 이상일 경우는 ""로 묶어서 표시한다.

☞ 속성이 있는 태그가 있고 없는 태그가 있다.

☞ 속성을 반드시 하나 이상 사용해야 하는 태그도 있고 생략해도 되는 태그가 있다.

☞ 여는 태그만 독립적으로 존재하는 태그도 있다.

#### 태그의 형식

- <태그> ~ </태그>
- <태그 속성="값"> ~ </태그>
- <태그>

#### 태그의 예

- <H1> 태그의 기본 속성 </H1>
- <A HREF="test.html"> test 문서로 이동 </A>
- <BR>

3) 태그의 이름은 대소문자를 구분하지 않는다.

☞ <H1>, <h1>, <TITLE>, <title>

☞ 대문자 사용이 바람직

4) 여러 개의 공백문자들은 하나로 인식된다.

☞ 공백문자들은 하나를 제외하고 모두 무시

☞ 예제

**• 공백이 들어간 태그 사용**

<P> "HTML 문서에서는 공백을 무시하게 된다." </P>

**• 결과**

"HTML 문서에서는 공백을 무시하게 된다."

5) 태그를 중첩하여 사용할 수 있다

☞ 태그를 중첩해서 사용할 경우, 먼저 열어준 태그를 나중에 닫아준다.

☞ 예제

**• 순서가 올바른 태그 사용**

<CENTER><FONT SIZE="2"> HTML은 구조적인 언어이다.  
</FONT></ CENTER>

**• 순서가 올바르지 못한 태그 사용**

<CENTER><FONT SIZE="2"> 순서가 올바르지 못한 태그 사용  
</CENTER></FONT>

## 3.2 HTML 기본 태그

### 3.2.1 기본 태그

#### 가. 문서 구조 태그

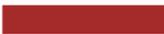
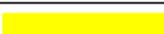
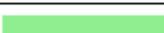
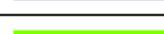
먼저, HTML 문서의 구조를 정의하기 위해 사용되는 태그에 대해 알아보자.

- <HTML>: HTML 문서의 시작을 의미하며, 동시에 HTML 문서 형식으로 작성되었음을 나타낸다.
- <TITLE>: HTML 문서의 제목을 브라우저의 제목 표시줄에 나타낸다.
- <HEAD>: 작성하려는 HTML 문서에 대한 설명을 나타낸다.
- <BODY>: HTML 문서의 실제 내용을 포함한다

<BODY> 태그의 속성으로는 다음을 지정할 수 있다.

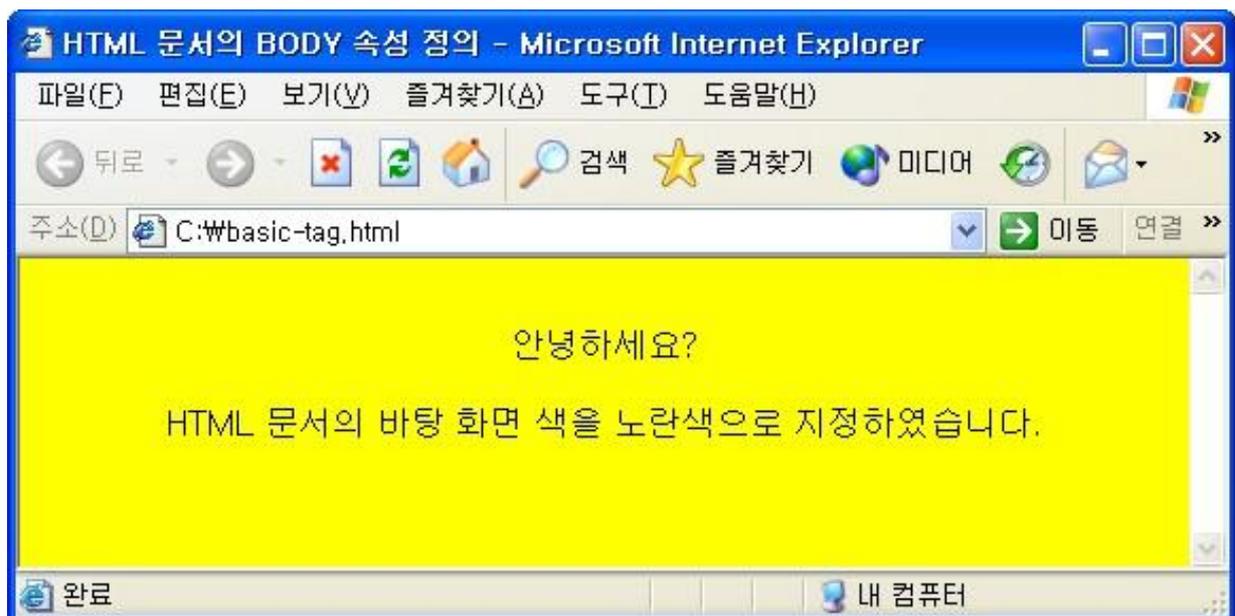
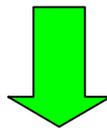
- BACKGROUND="이미지 파일명": 배경으로 사용될 이미지 파일을 지정한다.
- BGCOLOR="색상명" 또는 "RGB값": 배경색을 지정한다.
- TEXT="색상명" 또는 "RGB 값": 문자색을 지정한다.
- LINK="색상명" 또는 "RGB 값": 링크로 설정한 글자 색을 지정한다.
- ALINK="색상명" 또는 "RGB 값": 링크된 문자를 클릭할 때 변화되는 색을 지정한다.
- VLINK="색상명" 또는 "RGB 값": 링크를 사용한 후의 문자색을 지정한다.

색상 지정에 사용되는 '색상표'는 다음 그림과 같다.

색상	색상명	RGB 값
갈색 	BROWN	#A52A2A
감색 	NAVY	#000080
검정색 	BLACK	#000000
금색 	GOLD	#FFD700
노란색 	YELLOW	#FFFF00
밝은 녹색 	LIGHTGREEN	#90EE90
밝은 주황색 	LIGHTSALMON	#FFA07A
밝은 파란색 	LIGHTBLUE	#ADD8E6
밝은 회색 	LIGHTGRAY	#D3D3D3
보라색 	VIOLET	#EE82EE
분홍색 	PINK	#FFC0CB
붉은색 	RED	#FF0000
어두운 녹색 	DARKGREEN	#006400
어두운 붉은색 	DARKRED	#8B0000
어두운 파란색 	DARKBLUE	#00008B
연두색 	CHARTREUSE	#7FFF00
연보라색 	ORCHID	#DA70DB
올리브색 	OLIVE	#808000
은색 	SILVER	#C0C0C0
자주색 	PURPLE	#800080
주황색 	SALMON	#FA8072
짙은 분홍색 	MAGENTA	#FF00FF
초록색 	GREEN	#008000
파란색 	BLUE	#0000FF
하늘색 	AQUA	#00FFFF
하얀색 	WHITE	#FFFFFF
회색 	GRAY	#808080

상기의 기본 태그를 이용한 간단한 예제이다.

```
<HTML>
  <HEAD><TITLE> HTML 문서의 BODY 속성 정의 </TITLE></HEAD>
  <BODY BGCOLOR="YELLOW" >
    <CENTER>
      <BR> 안녕하세요? <P> HTML 문서의 바탕 화면 색을 노란색으로
        지정하였습니다.
      <!-- 바탕 화면색 지정 -->
    </CENTER>
  </BODY>
</HTML>
```



#### 나. 주석 태그

HTML 문서에 대한 부수적 설명 혹은 해석이 필요할 경우 사용되는 태그로써, 주석의 내용은 브라우저에서 출력되지 않는다.

**형 식 : <!-- 주석 내용 -->**

#### 다. <Hn> 제목 태그

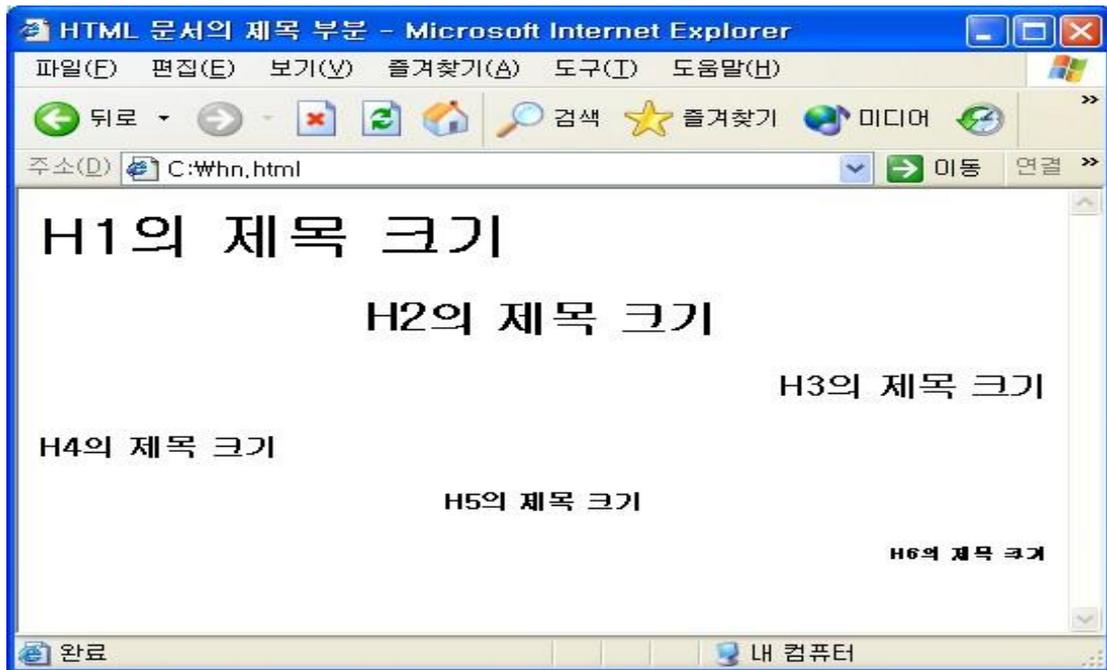
HTML 문서에 대한 부수적 설명 혹은 해석이 필요할 경우 사용되는 태그로써, 주석의 내용은 브라우저에서 출력되지 않는다.

**형 식 : <Hn ALIGN = "LEFT" 또는 "RIGHT" 또는 "CENTER">  
~ </Hn> n은 1부터 6까지의 숫자이다.**

다음은 제목 태그를 사용한 간단한 예제이다.

```
<HTML>
  <HEAD> <TITLE> HTML 문서의 제목 부분 </TITLE> </HEAD>
  <BODY>
    <H1 ALIGN = "LEFT"> H1의 제목 크기 </H1>
    <H2 ALIGN = "CENTER"> H2의 제목 크기 </H2>
    <H3 ALIGN = "RIGHT"> H3의 제목 크기 </H3>
    <H4 ALIGN = "LEFT" > H4의 제목 크기 </H4>
    <H5 ALIGN = "CENTER"> H5의 제목 크기 </H5>
    <H6 ALIGN = "RIGHT"> H6의 제목 크기 </H6>
  </BODY>
</HTML>
```

상기 예제의 결과화면은 다음과 같다.



### 3.2.2 문자 서식 태그

#### 가. 글꼴 태그

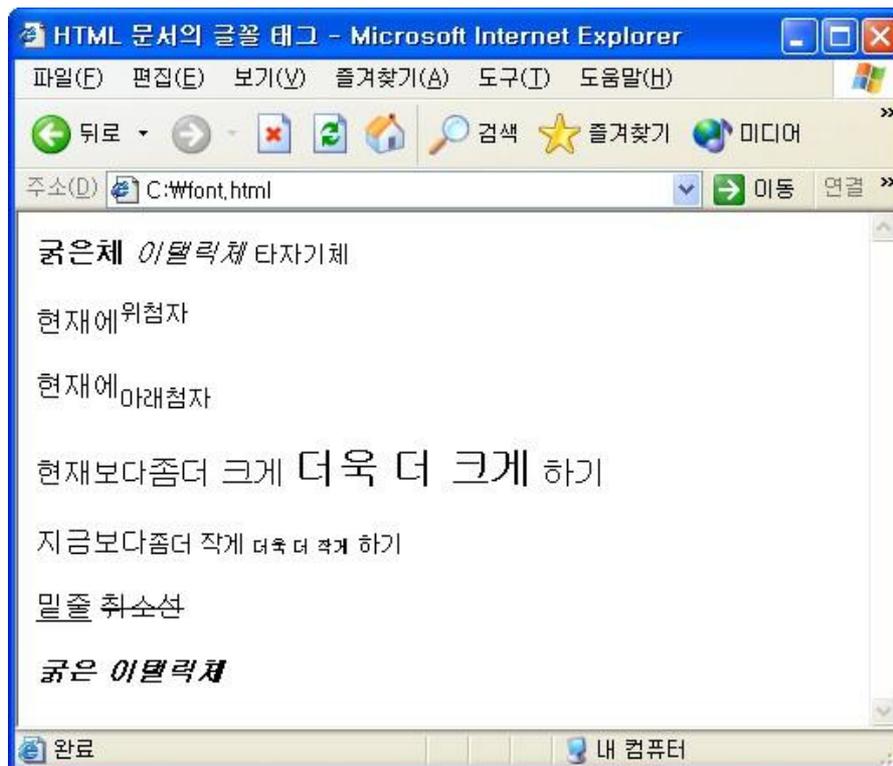
다음은 문자들의 크기 및 모양을 조절하기 위해 사용되는 태그이다.

<B>	: 굵은 문자(Bold)를 나타낸다.
<I>	: 이탤릭체(Italic)를 나타낸다.
<TT>	: 타자기체(TypewriterText)를 나타낸다.
<SUP>	: 윗첨자(SUPerscript)를 나타낸다.
<SUB>	: 아래첨자(SUBscript)를 나타낸다.
<BIG>	: 글자 크기를 현재 크기보다 크게 나타낸다.
<SMALL>	: 글자 크기를 현재 크기보다 작게 나타낸다.
<U>	: 밑줄(Underline)을 나타낸다.
<S>	: 취소선(Strike Through)을 나타낸다.

다음은 상기 태그에 대한 예제 문서이다.

```
<HTML>
<HEAD> <TITLE> HTML 문서의 글꼴 태그 </TITLE> </HEAD>
<BODY>
  <B>굵은체</B> <I>이탤릭체</I> <TT>타자기체</TT> <P>
  현재에<SUP>위첨자</SUP> <P>
  현재에<SUB>아래첨자</SUB> <P>
  현재보다<BIG>좀더 크게 <BIG>더욱 더 크게</BIG> 하기 </BIG> <P>
  지금보다<SMALL>좀더 작게 <SMALL>더욱 더 작게</SMALL>
  하기</SMALL> <P>
  <U>밑줄</U>   <S>취소선</S> <P>
  <B> <I>굵은 이탤릭체</I> </B>
</BODY>
</HTML>
```

상기 예제에 대한 출력 결과는 다음과 같다.



나. <FONT> 태그

앞서 기술한 <Hn> 태그가 문서의 제목 크기를 변경하는 태그인 반면에, <Font> 태그는 본문의 글자 크기를 조절하는 경우에 사용된다.

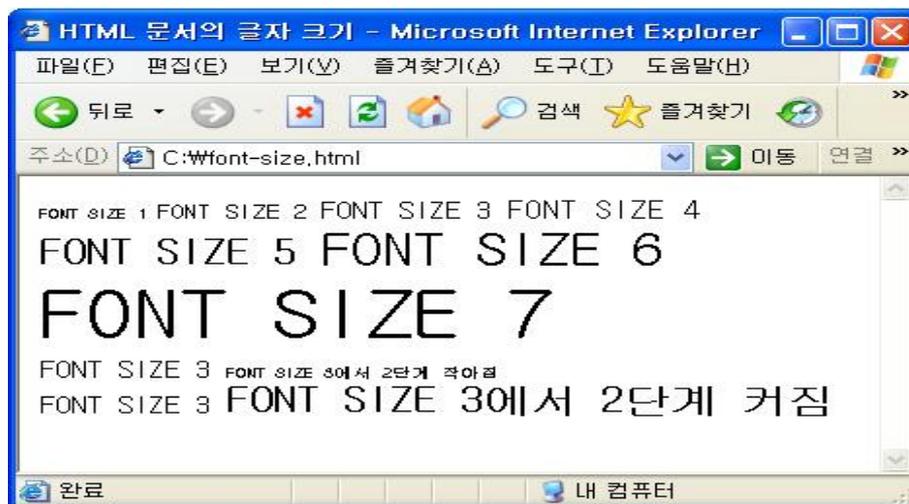
형 식 : <FONT SIZE = "n"> ~ </FONT>

- 1~7까지의 7단계로 구분되며, 1이 가장 작고 7이 가장 크다
- 상대적인 크기를 지정하는 경우에는 다음과 같이 사용한다.

형 식 : <FONT SIZE = "+n" 또는 "-n"> ~ </FONT>

다음은 <FONT> 태그의 사용 예제이다.

```
<HTML>
<HEAD> <TITLE> HTML 문서의 글자 크기 </TITLE> </HEAD>
<BODY>
  <FONT SIZE = "1"> FONT SIZE 1 </FONT>
  <FONT SIZE = "2"> FONT SIZE 2 </FONT>
  <FONT SIZE = "3"> FONT SIZE 3 </FONT>
  <FONT SIZE = "4"> FONT SIZE 4 </FONT> <BR>
  <FONT SIZE = "5"> FONT SIZE 5 </FONT>
  <FONT SIZE = "6"> FONT SIZE 6 </FONT>
  <FONT SIZE = "7"> FONT SIZE 7 </FONT> <BR>
  <FONT SIZE = "3"> FONT SIZE 3 </FONT>
  <FONT SIZE = "-2"> FONT SIZE 3에서 2단계 작아짐 </FONT> <BR>
  <FONT SIZE = "3"> FONT SIZE 3 </FONT>
  <FONT SIZE = "+2"> FONT SIZE 3에서 2단계 커짐 </FONT>
</BODY>
</HTML>
```



#### 다. 글자색 지정

글자색을 지정하는 경우에는 <FONT> 태그의 COLOR 속성을 사용한다.

**형 식 : <FONT COLOR= "색상명" 또는 "RGB 값"> ~ </FONT>**

다음은 글자색 지정에 대한 예제이다.

```
<HTML>
  <HEAD> <TITLE> HTML 문서의 글자 색 </TITLE> </HEAD>
  <BODY>
    <FONT COLOR = "BLUE" > BLUE </FONT> 파란색으로 표시 <BR>
    <FONT COLOR= "RED" > RED </FONT> 빨간색으로 표시
  </BODY>
</HTML>
```



#### 라. 기 타

문자의 깜빡거림 효과를 위해서는 <BLINK> 태그를 사용하고, 특수문자를 출력하고자 하는 경우에는 아래와 같이 별도로 지정한 기호를 사용한다.

**형 식 : <BLINK> ~ </BLINK>**

```
< : &lt;
> : &gt;
& : &amp;
" : &quot;
공백 : &nbsp;
```

### 3.2.3 단락 서식 태그

#### 가. 줄 바꿈 태그 <BR>

<BR> 태그는 "line break"를 의미하며 줄 바꿈 태그로 사용한다.

```
<HTML>
  <HEAD> <TITLE> HTML 문서의 줄바꿈 </TITLE> </HEAD>
  <BODY>
    줄바꿈 태그를 이용하면 <BR> 줄이 바뀝니다.<BR>
  </BODY>
</HTML>
```



#### 나. 단락 태그 <P>

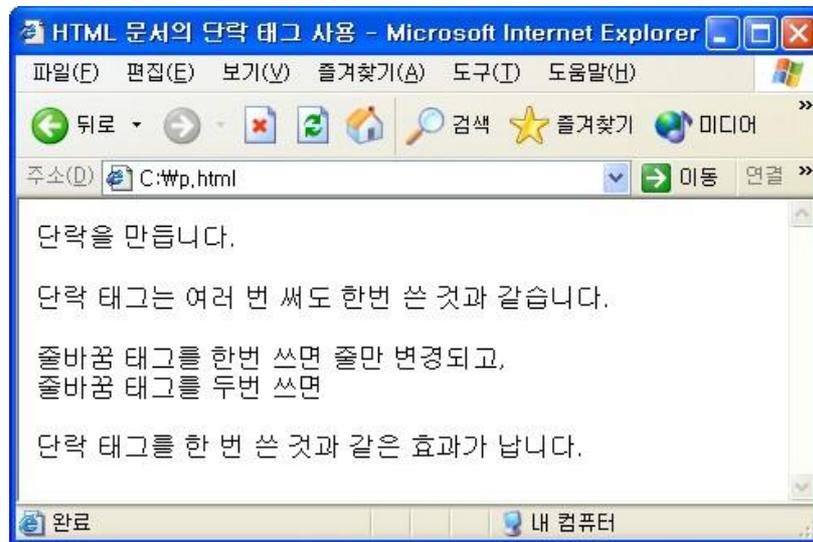
<P> 태그는 문단을 나누기 위한 태그이며 <BR> 태그와의 차이점은 다음과 같다.

- 줄바꿈 + 한 줄의 공백
- <BR> 태그를 두 번 사용한 결과
- 복수의 <P> 태그는 한 번 사용한 결과와 동일

```

<HTML>
<HEAD><TITLE> HTML 문서의 단락 태그 사용 </TITLE></HEAD>
<BODY>
  단락을 만듭니다.<P>
  단락 태그는 여러 번 써도 한번 쓴 것과 같습니다.<P> <P> <P>
  줄바꿈 태그를 한 번 쓰면 줄만 변경되고, <BR>
  줄바꿈 태그를 두 번 쓰면 <BR><BR> 단락 태그를 한 번 쓴 것과
  같은 효과가 납니다.
</BODY>
</HTML>

```



다. 꺾선 태그 <HR>

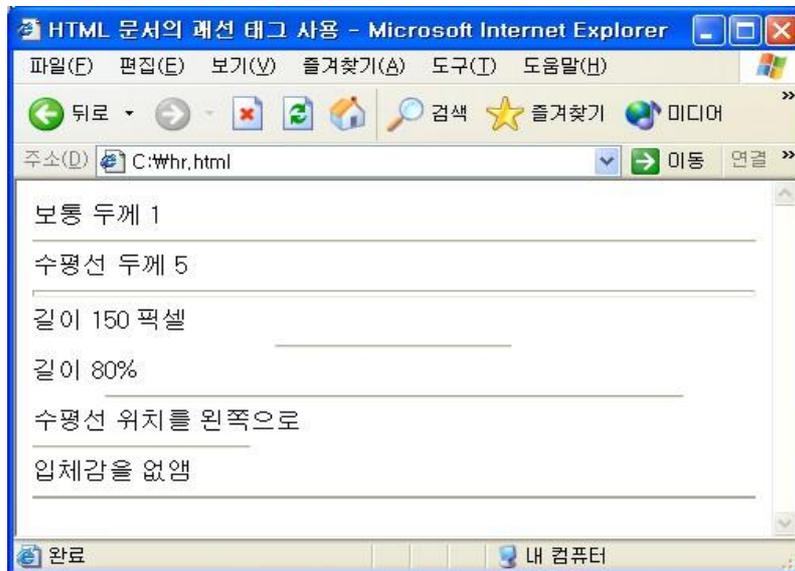
형 식 : <HR [속 성]>

**SIZE** = "픽셀 수" : 수평선의 두께를 지정한다.  
**WIDTH** = "픽셀 수" 또는 "가로 폭에 대한 비율(%)" : 수평선의 길이를 지정한다.  
**ALIGN** = "LEFT" 또는 "CENTER" 또는 "RIGHT" : 수평선의 위치를 지정한다.  
**NOSHADOW** = 입체감과 음영의 효과를 제거 (보통 선으로 표현) 한다.

```

<HTML>
<HEAD> <TITLE> HTML 문서의 과선 태그 사용 </TITLE>
</HEAD>
<BODY>
  보통 두께 1 <HR>
  수평선 두께 5 <HR SIZE= "5">
  길이 150 픽셀 <HR WIDTH= "150">
  길이 80% <HR WIDTH= "80%">
  수평선 위치를 왼쪽으로 <HR ALIGN= "LEFT" WIDTH=
    "30%">
  입체감을 없앴 <HR NOSHADE>
</BODY>
</HTML>

```



라. 정렬 태그 <HR>

가운데 정렬의 경우 <CENTER> 태그를 사용할 수도 있으나, 다음과 같이 <DIV> 태그의 다양한 속성으로 정의할 수도 있다.

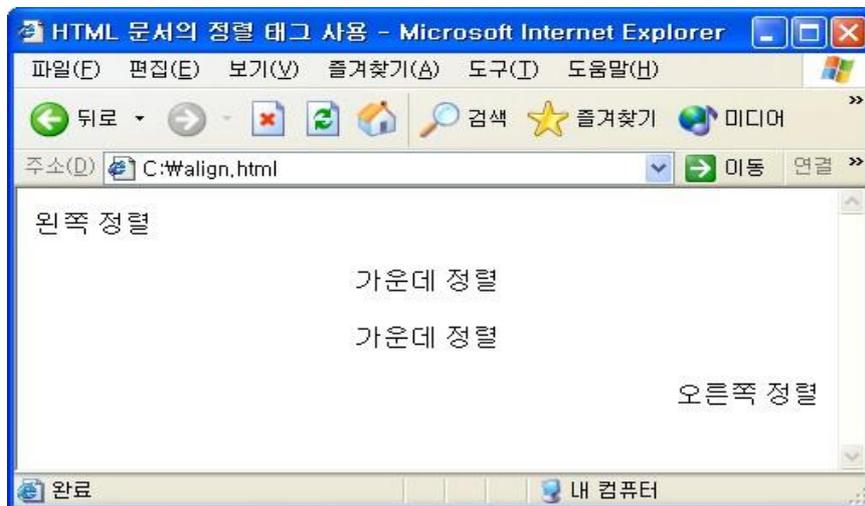
**형식 : <CENTER> ~ </CENTER>**

**형식 :**  
**<DIV ALIGN = "LEFT" 또는 "RIGHT" 또는 "CENTER"> ~ </DIV>**

```

<HTML>
  <HEAD> <TITLE> HTML 문서의 정렬 태그 사용</TITLE>
</HEAD>
  <BODY>
    <DIV ALIGN = "LEFT"> 왼쪽 정렬 </DIV> <P>
    <DIV ALIGN = "CENTER"> 가운데 정렬 </DIV> <P>
    <CENTER> 가운데 정렬 </CENTER> <P>
    <DIV ALIGN = "RIGHT"> 오른쪽 정렬 </DIV> <P>
  </BODY>
</HTML>

```



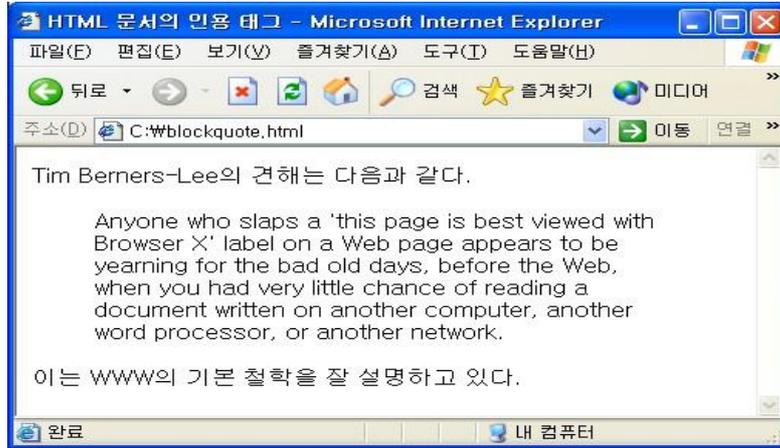
마. 인용 태그

형 식 : <BLOCKQUOTE> ~ </BLOCKQUOTE>

```

<HTML>
  <HEAD><TITLE>HTML 문서의 BLOCKQUOTE 태그</TITLE></HEAD>
  <BODY>
    Tim Berners-Lee의 견해는 다음과 같다.
    <BLOCKQUOTE>
    <P>Anyone who slaps a 'this page is best viewed with Browser X'
    label on a Web page appears to be yearning for the bad old days,
    before the Web, when you had very little chance of reading a
    document written on another computer, another word processor,
    or another network.
    </BLOCKQUOTE>
    이는 WWW의 기본 철학을 잘 설명하고 있다.
  </BODY>
</HTML>

```



## 바. <PRE> 태그

<PRE> 태그는 입력한 대로 보여주기 위한 태그로서, HTML 문서에서 입력한 “한 번 이상의 공백과 스페이스를 그대로 표현”할 수 있다.

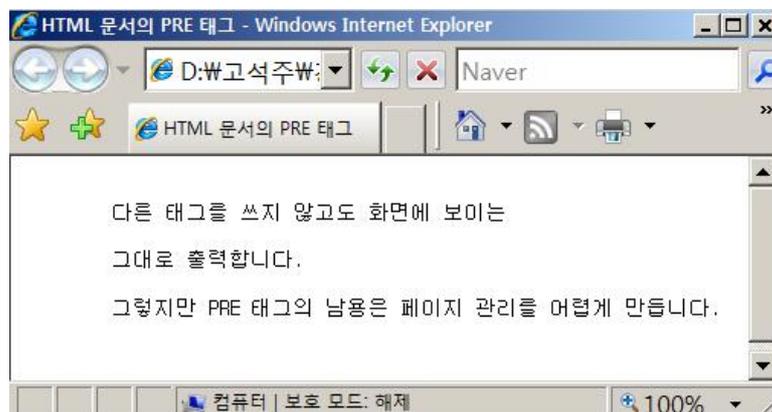
**형식 : <PRE> ~ </PRE>**

```
<HTML>
<HEAD> <TITLE> HTML 문서의 PRE 태그 </TITLE> </HEAD>
<BODY>
  <PRE>

    다른 태그를 쓰지 않고도 화면에 보이는
    그대로 출력합니다.

    그렇지만 PRE 태그의 남용은 페이지 관리를 어렵게 만듭니다.

  </PRE>
</BODY>
</HTML>
```



### 3.2.4 목록 태그

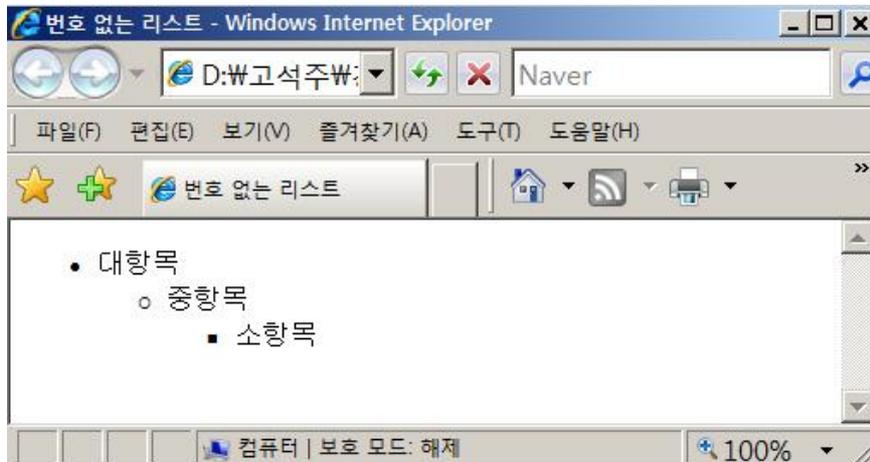
목록 태그는 시각적으로 짜임새 있는 문서 작성을 위해 자주 사용된다.

```
<UL> : 번호가 없는 리스트 (Unordered List)
  - <LI> : 리스트 항목
<OL> : 번호가 있는 리스트 (Ordered List)
  - <LI> : 리스트 항목
<MENU> : 메뉴 리스트
  - <LI> : 리스트 항목
<DIR> : 디렉토리 리스트
  - <LI> : 리스트 항목
<DL> : 용어 리스트 (Definition List)
  - <LH> : 리스트 제목
  - <DT> : 용어의 이름
  - <DD> : 용어의 설명
```

#### 가. Unordered List <UL>

```
형 식 :
<UL>
  <LI TYPE= "DISK" 또는 "CIRCLE" 또는 "SQUARE">
    리스트 항목
</UL>
```

```
<HTML>
<HEAD> <TITLE> 번호 없는 리스트 </TITLE> </HEAD>
<BODY>
  <UL>
    <LI>대항목
      <UL>
        <LI>중항목
          <UL>
            <LI>소항목
          </UL>
        </UL>
      </UL>
    </UL>
  </BODY>
</HTML>
```



#### 나. Ordered List <OL>

```
형 식 : <OL>
        <LI> 리스트 항목
      </OL>
```

#### 다. 기타 리스트

```
형 식 (메뉴 리스트) :
  <MENU>
    <LI> 리스트 항목
  </MENU>
형 식 (디렉토리 리스트) :
  <DIR>
    <LI> 리스트 항목
  </DIR>
형 식 (용어 리스트) :
  <DL>
    <LH>리스트의 제목</LH>
    <DT>용어의 제목
    <DD>용어의 의미
  </DL>
```

### 3.2.5 연결 태그

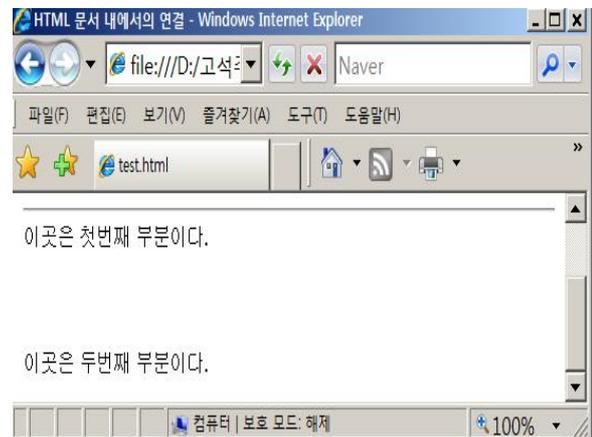
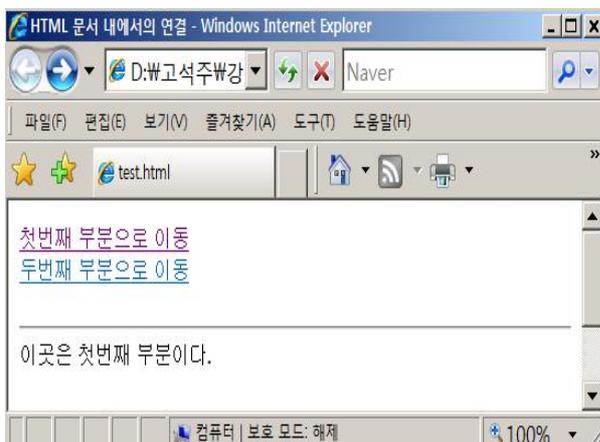
#### 가. HTML 문서 내에서의 연결

동일 문서내의 특정 위치로 이동하는 경우로서, 주로 HTML 문서가 매우 길 경우에 사용한다.

```
형 식 : <A HREF ="#이름"> ~ </A>
        문서 내에서의 연결 링크를 설정한다.

        <A NAME ="이름"> ~ </A>
        문서 내의 이동할 곳의 위치를 지정한다.
```

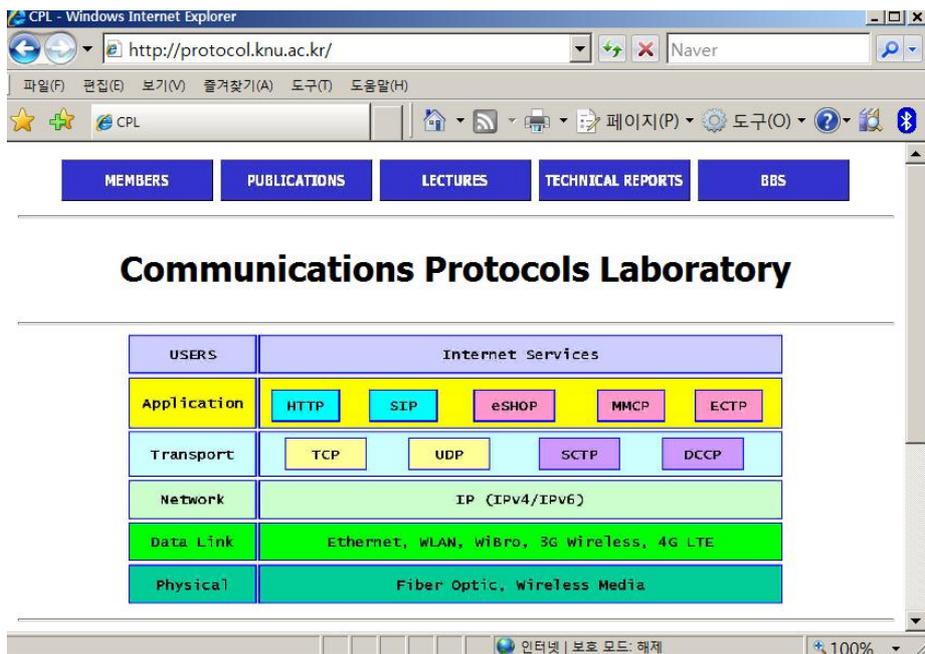
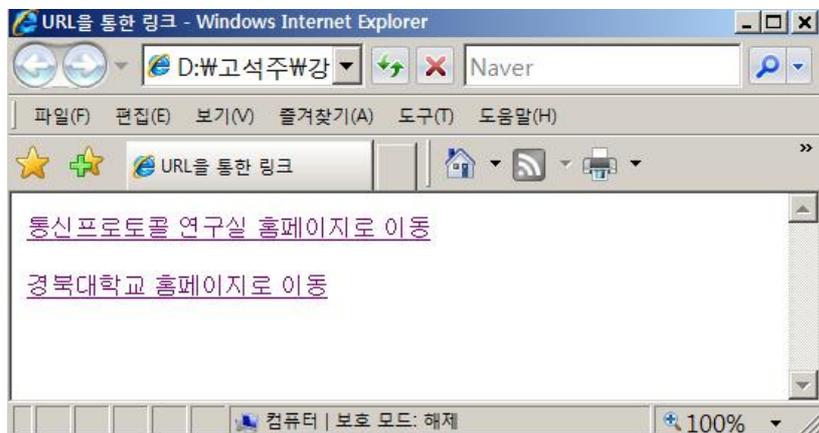
```
<HTML>
  <HEAD><TITLE> HTML 문서 내에서의 연결 </TITLE></HEAD>
  <BODY>
    <A HREF = "#이름1"> 첫번째 부분으로 이동</A> <BR>
    <A HREF = "#이름2"> 두번째 부분으로 이동</A> <BR>
    <P> <HR>
    <A NAME = "이름1"> 이곳은 첫번째 부분이다.</A>
      <BR><BR><BR><BR>
    <A NAME = "이름2"> 이곳은 두번째 부분이다.</A> <BR>
  </BODY>
</HTML>
```



나. URL을 통한 문서간의 연결

형식 :  
 <A HREF = "링크할 문서의 URL"> 브라우저에 나타나는 텍스트 </A>

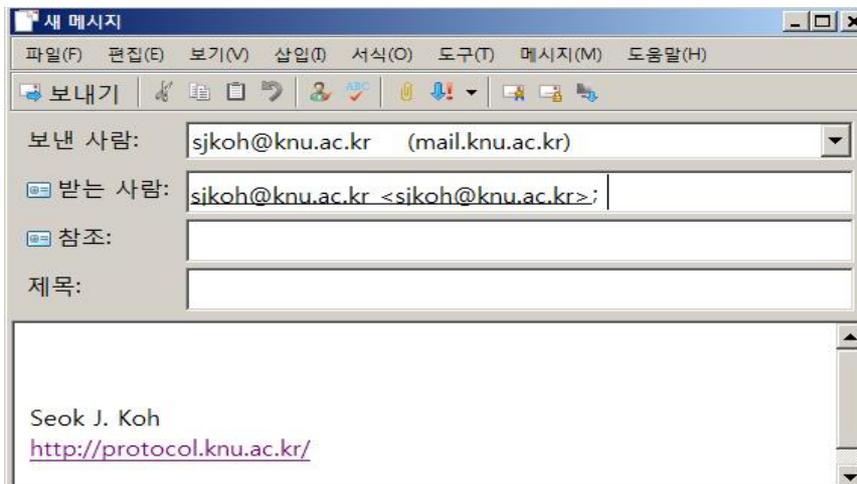
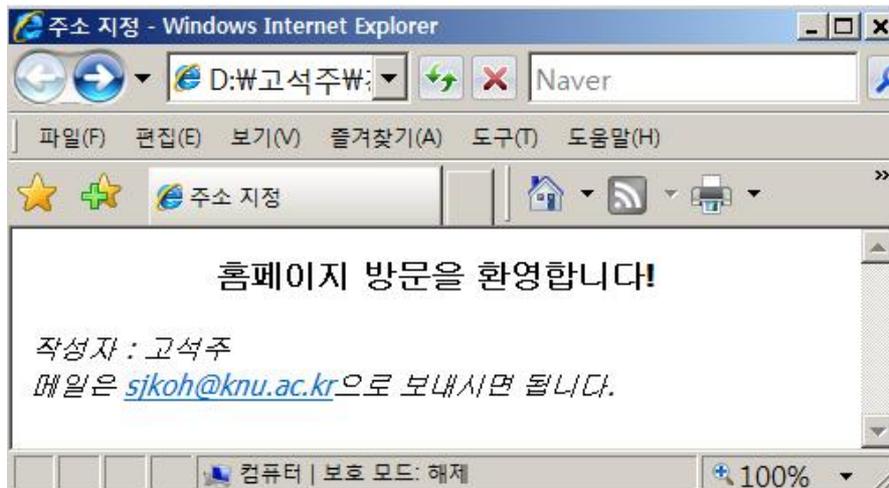
```
<HTML>
<HEAD><TITLE> URL을 통한 링크 </TITLE></HEAD>
<BODY><P>
  <A HREF = "HTTP://protocol.knu.ac.kr/"> 연구실 홈페이지 </A><P>
  <A HREF = "HTTP://www.knu.ac.kr/"> 경북대학교 홈페이지 </A>
</BODY>
</HTML>
```



다. 주소 지정 태그

형 식 : <ADDRESS> ~ </ADDRESS>

```
<HTML>
<HEAD> <TITLE> 주소 지정 </TITLE> </HEAD>
<BODY >
  <CENTER> <H3> 홈페이지 방문을 환영합니다! </H3>
  </CENTER>
  <ADDRESS> 작성자 : 고석주 <BR>
    메일은 <A HREF= "mailto:sjkoh@knu.ac.kr">
    sjkoh@knu.ac.kr</A> 으로 보내시면 됩니다.
  </ADDRESS>
</BODY>
</HTML>
```



### 3.3 이미지 삽입하기

이 절에서는 웹 브라우저를 이용한 이미지(image) 삽입 및 처리 방법에 대하여 알아본다.

#### 3.3.1 <IMG> 태그

먼저 <IMG> 태그 사용법에 대해 살펴보자.

형 식 : <IMG SRC = "이미지 파일 이름" ALT = "이미지 파일 설명" >  
SRC : 이 속성은 필수이고, GIF 또는 JPG형식의 이미지를 사용한다.  
ALT : 이미지에 마우스를 가져가면 파일에 대한 설명이 나타난다.

```
<HTML>  
<HEAD> <TITLE> 이미지 삽입하기 </TITLE> </HEAD>  
<BODY>  
  <CENTER>  
    <H2> 이미지 삽입 </H2>  
    <IMG SRC = "logo_knu.gif" ALT = "경북대학교" >  
  </CENTER>  
</BODY>  
</HTML>
```



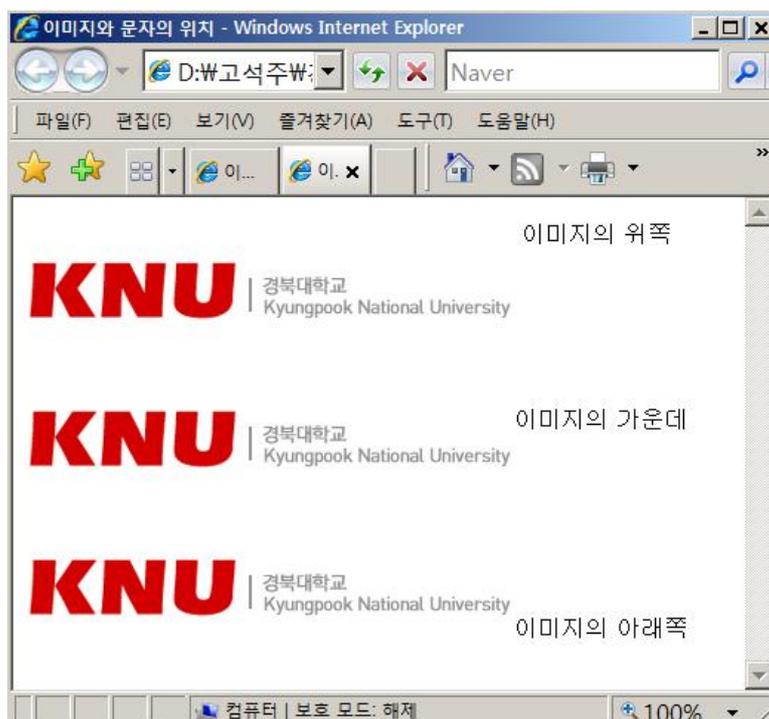
### 3.3.2 <IMG> 태그의 속성

이미지와 문자의 정렬을 위해 "ALIGN 속성"을 사용한다. 이를 통해 이미지의 특정 위치에 문자를 어떻게 위치시킬 것인가를 결정할 수 있다.

ALIGN = "TOP" : 이미지의 위 부분에 문자를 출력한다.  
ALIGN = "MIDDLE" : 이미지의 가운데 부분에 문자를 출력한다.  
ALIGN = "BOTTOM" : 이미지의 아래 부분에 문자를 출력한다.  
ALIGN = "LEFT" : 이미지를 왼쪽에 위치 시킨다.  
ALIGN = "RIGHT" : 이미지를 오른쪽에 위치 시킨다.

먼저 상하 정렬에 대한 예제이다.

```
<HTML>  
<HEAD> <TITLE> 이미지와 문자의 위치 </TITLE> </HEAD>  
<BODY>  
<IMG SRC = "logo_knu.gif" ALIGN = "TOP"> 이미지의 위쪽 <P>  
<IMG SRC = "logo_knu.gif" ALIGN = "MIDDLE">이미지의 가운데 <P>  
<IMG SRC = "logo_knu.gif" ALIGN = "BOTTOM">이미지의 아래쪽  
</BODY>  
</HTML>
```



다음은 좌우 정렬에 대한 예제이다.

```
<HTML>
<HEAD> <TITLE> 이미지와 문자의 위치 </TITLE> </HEAD>
<BODY>
  <IMG SRC = "logo_knu.gif" ALIGN = "LEFT">
    이미지는 왼쪽 <BR>
    문자는 오른쪽 <P>
  <BR> <BR>
  <IMG SRC = "logo_knu.gif" ALIGN = "RIGHT">
    이미지는 오른쪽 <BR>
    문자는 왼쪽 <P>
</BODY>
</HTML>
```



이미지의 크기 조절을 위해 WIDTH와 HEIGHT 속성을 이용한다.

형 식 :

```
<IMG SRC = "이미지 파일명" WIDTH = "픽셀수 또는 비율(%)"
                                     HEIGHT = "픽셀수 또는 비율(%)" >
```

WIDTH : 이미지의 가로 크기를 정한다.  
HEIGHT : 이미지의 세로 크기를 정한다.

한편, 이미지의 여백 조절을 위해 VSPACE와 HSPACE 속성을 이용한다.

형 식 :

```
<IMG SRC="이미지 파일명" VSPACE="픽셀수 또는 비율(%)"  
HSPACE="픽셀수 또는 비율(%)">
```

VSPACE : 이미지 상하의 여백을 준다.

HSPACE : 이미지 좌우의 여백을 준다.

```
<HTML>  
<HEAD><TITLE> 이미지의 여백 주기 </TITLE></HEAD>  
<BODY>  
  이미지의 상하에 <BR>  
  <IMG SRC = "logo.gif" VSPACE = "20" ALIGN = "LEFT">  
  여백을 <BR>  
  20픽셀 주었다. <P>  
  <BR> <BR> <BR>  
  이미지의 좌우에 <BR>  
  <IMG SRC = "logo.gif" HSPACE = "20" ALIGN = "LEFT">  
  여백을 <BR>  
  20픽셀 주었다.  
</BODY>  
</HTML>
```

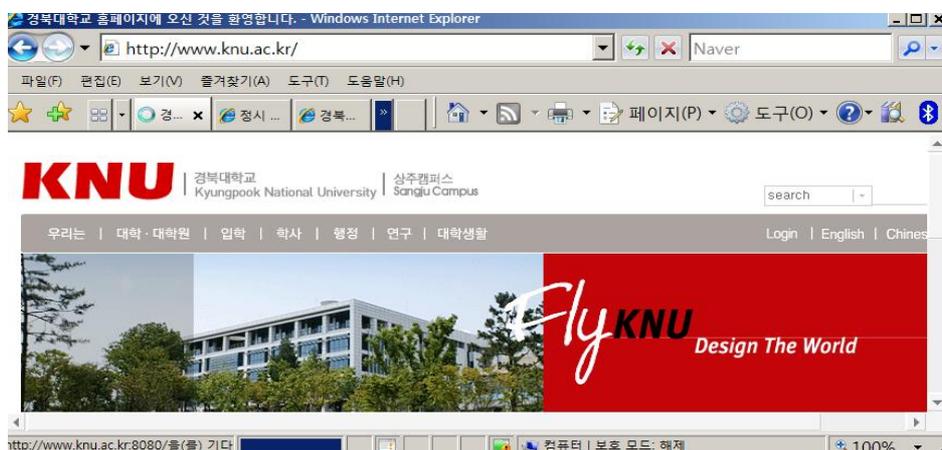


### 3.3.3 이미지로 문서 연결

하이퍼링크를 이용한 문서연결은 텍스트는 물론 이미지를 통해서도 가능하다.

```
형 식 : <A HREF = "링크할 곳의 파일명" >  
        <IMG SRC = "이미지 파일명" [BORDER="n"]> </A>
```

```
<HTML>  
<HEAD> <TITLE> 이미지로 문서 연결하기 </TITLE> </HEAD>  
<BODY>  
<CENTER>  
<H2> 이미지로 문서 연결하기 </H2>  
    그림을 누르면 해당 페이지로 이동합니다.<p>  
    <A HREF="http://www.knu.ac.kr/">  
        <IMG SRC = "logo.gif"> </A>  
</CENTER>  
</BODY>  
</HTML>
```



### 3.4 테이블 만들기

정보의 조직적 표현을 테이블(table)이 자주 사용된다. 이 절에서는 기본적인 테이블 태그 및 테이블의 세밀한 제어를 위해 사용되는 태그들을 알아본다.

#### 3.4.1 테이블 기본 태그

테이블 작성을 위해 사용되는 태그들은 다음과 같다.

```
<TABLE> : 테이블을 선언하는 태그이다.  
<TR> : Table Row, 테이블 내의 한 행을 정의한다.  
<TD> : Table Data, 각 행의 셀을 만들 때 사용한다.  
<TH> : Table Header, 행에 있는 셀의 제목을  
지정하여, 제목을 강조되게 표시한다.  
<CAPTION> : 테이블의 제목을 표시한다.
```

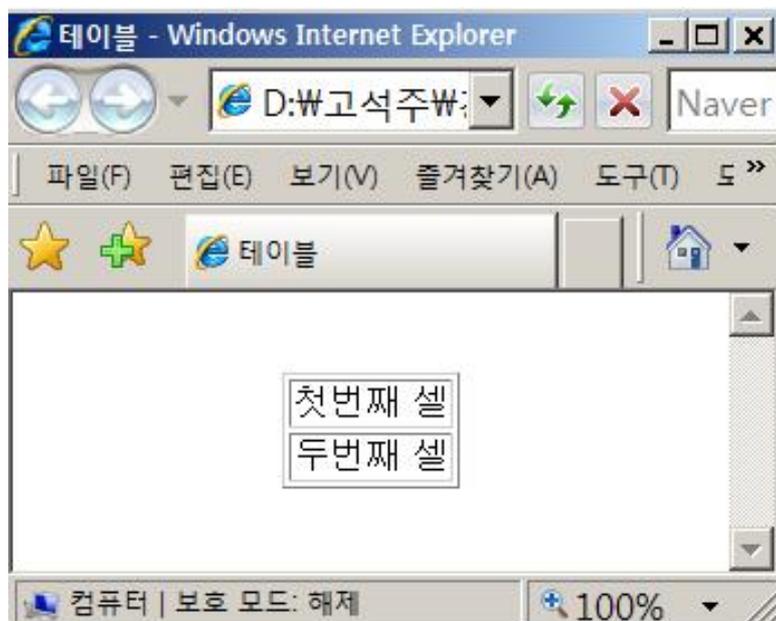
먼저 테이블 선언을 위해 다음 태그를 사용한다.

```
형 식 : <TABLE> ~ </TABLE>
```

테이블의 세밀한 제어를 위해서 <TD> 태그의 다양한 속성을 활용한다.

```
<TD ALIGN = "LEFT" 또는 "RIGHT" 또는 "CENTER">  
: 셀의 데이터에 대한 좌우 정렬 방법이다.  
<TD VALIGN = "TOP" 또는 "MIDDLE" 또는 "BOTTOM" 또는 "BASELINE">  
: 셀의 데이터에 대한 상하 정렬 방법이다.  
<TD ROWSPAN> : 셀의 행을 합치기 위한 태그이다.  
<TD COLSPAN> : 셀의 열을 합치기 위한 태그이다.  
<TD WIDTH> : 셀의 수평 길이(픽셀 단위)를 나타낸다.  
<TD HEIGHT> : 셀의 수직 길이(픽셀 단위)를 나타낸다.
```

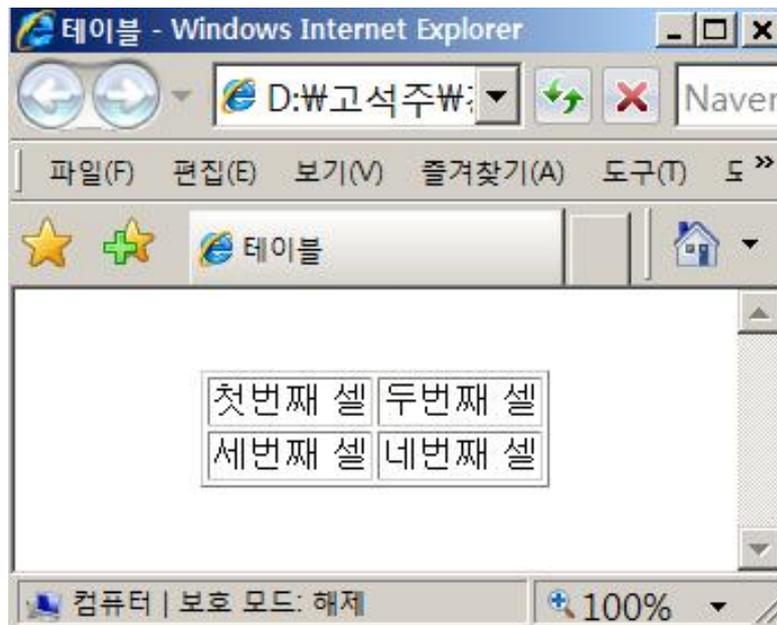
```
<HTML>
<HEAD> <TITLE> 테이블 </TITLE> </HEAD>
<BODY>
  <CENTER> <BR>
  <TABLE BORDER="1">
    <TR>
      <TD>첫번째 셀</TD>
    </TR>
    <TR>
      <TD>두번째 셀</TD>
    </TR>
  </TABLE>
</CENTER>
</BODY>
</HTML>
```



```

<HTML>
  <HEAD> <TITLE> 테이블 </TITLE> </HEAD>
  <BODY>
    <CENTER> <BR>
    <TABLE BORDER="1">
      <TR>
        <TD>첫번째 셀</TD>
        <TD>두번째 셀</TD>
      </TR>
      <TR>
        <TD>세번째 셀</TD>
        <TD>네번째 셀</TD>
      </TR>
    </TABLE>
  </CENTER>
</BODY>
</HTML>

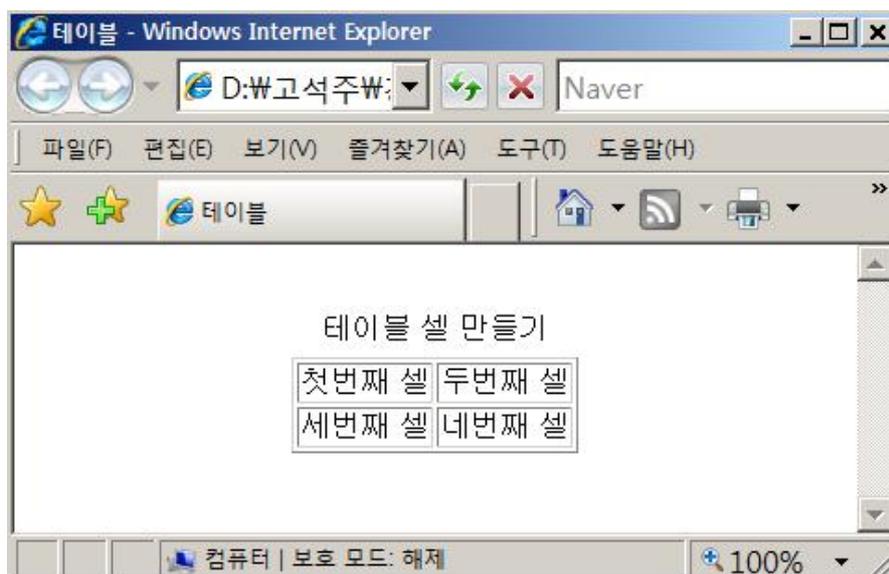
```



테이블의 제목은 다음과 같이 <CAPTION> 태그를 사용하여 지정한다.

형 식 : <CAPTION ALIGN = "TOP" 또는 "BOTTOM">  
테이블의 제목 </CAPTION>

```
test.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<HTML>
<HEAD> <TITLE> 테이블 </TITLE>
</HEAD>
<BODY>
<CENTER> <BR>
<TABLE BORDER="1">
<CAPTION ALIGN="TOP">
테이블 셀 만들기
</CAPTION>
<TR>
<TD>첫번째 셀</TD>
<TD>두번째 셀</TD>
</TR>
<TR>
<TD>세번째 셀</TD>
<TD>네번째 셀</TD>
</TR>
</TABLE>
</CENTER>
</BODY>
</HTML>
```



여러 개의 셀을 합치기 위한 태그로는 <TD> 혹은 <TH>를 사용한다.

➤ 행 합치기 태그

```
형식 : <TD ROWSPAN = n> ~ </TD>  
<TH ROWSPAN = n> ~ </TH>
```

➤ 열 합치기 태그

```
형식 : <TD COLSPAN = n > ~ </TD>  
<TH COLSPAN = n > ~ </TH>
```

```
<HTML>  
<HEAD> <TITLE> 테이블 </TITLE>  
</HEAD>  
<BODY>  
<CENTER> <BR>  
<TABLE BORDER="1">  
<TR>  
<TH ROWSPAN=2> 수도 </TH>  
<TH> 한국 </TH>  
<TH> 영국 </TH>  
</TR>  
<TR>  
<TD> 서울 </TD>  
<TD> 런던 </TD>  
</TR>  
</TABLE>  
<BR> <BR>
```

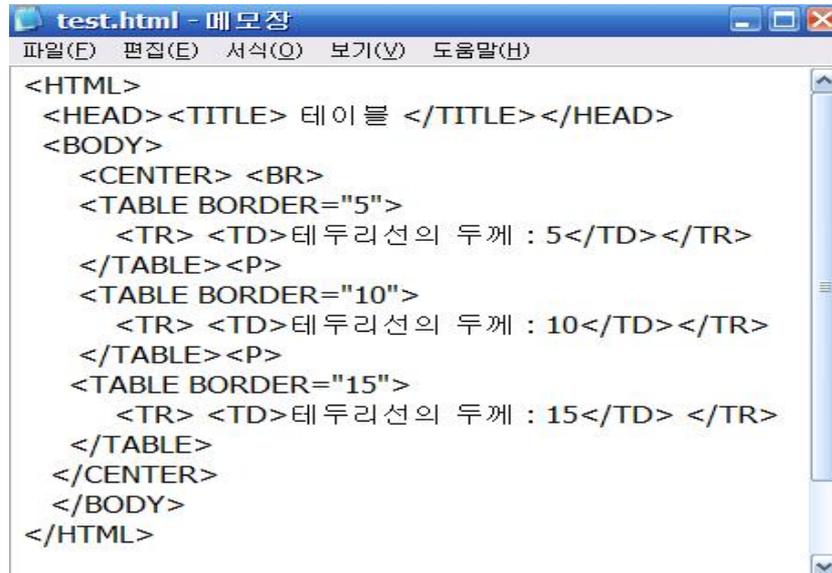
```
<TABLE BORDER="1">  
<TR> <TH COLSPAN=2> 수도  
</TH> </TR>  
<TR>  
<TH> 한국 </TH>  
<TH> 영국 </TH>  
</TR>  
<TR>  
<TD> 서울 </TD>  
<TD> 런던 </TD>  
</TR>  
</TABLE>  
</CENTER>  
</BODY>  
</HTML>
```



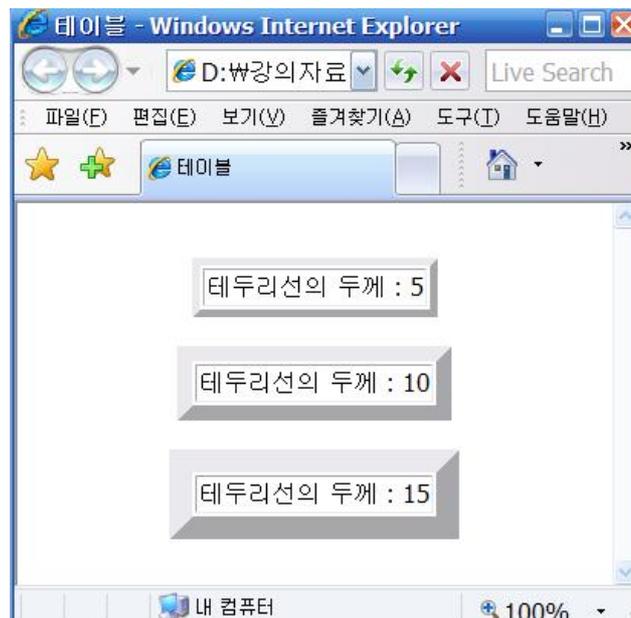
### 3.4.2 테이블 서식 지정

#### 가. 테이블 외곽선 지정

형 식 : <TABLE BORDER = "픽셀수"> ~ </TABLE>



```
<HTML>
<HEAD><TITLE> 테이블 </TITLE></HEAD>
<BODY>
  <CENTER> <BR>
  <TABLE BORDER="5">
    <TR> <TD>테두리선의 두께 : 5</TD></TR>
  </TABLE><P>
  <TABLE BORDER="10">
    <TR> <TD>테두리선의 두께 : 10</TD></TR>
  </TABLE><P>
  <TABLE BORDER="15">
    <TR> <TD>테두리선의 두께 : 15</TD> </TR>
  </TABLE>
</CENTER>
</BODY>
</HTML>
```



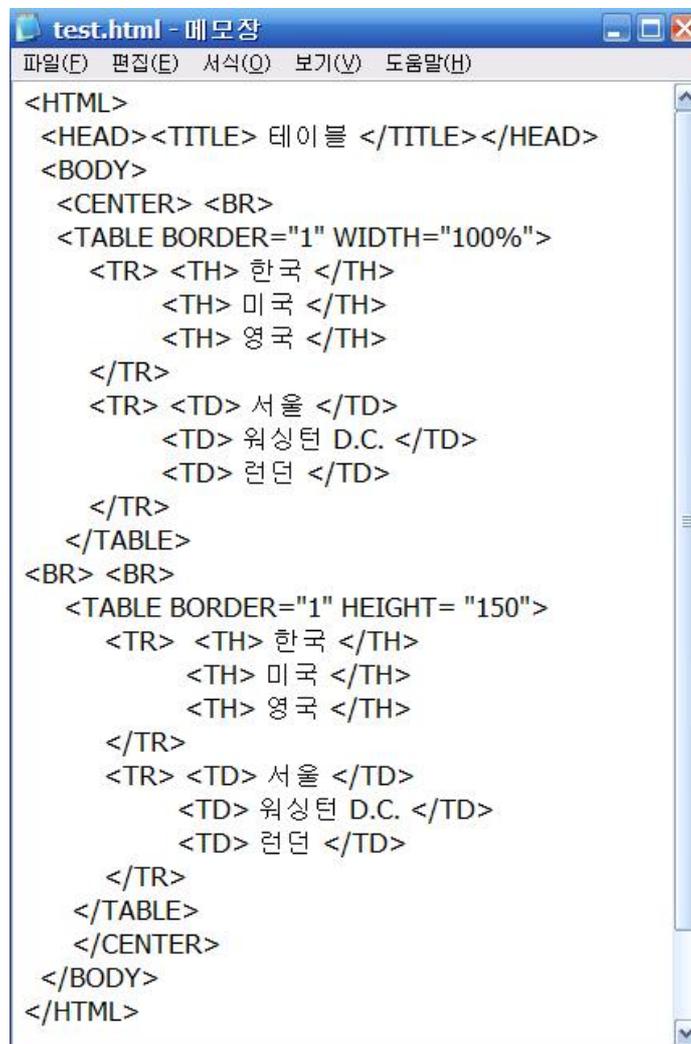
## 나. 테이블 크기 조정

### ➤ 폭 조절

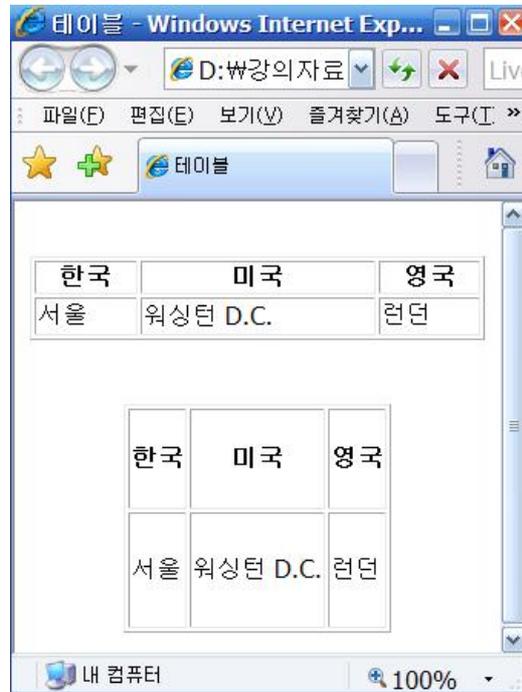
형 식 : <TABLE WIDTH= "픽셀수" 또는 "가로폭에 대한 비율(%)">  
~ </TABLE>

### ➤ 높이 조절

형 식 : <TABLE HEIGHT= "픽셀수" 또는 "세로폭에 대한 비율(%)" >  
~ </TABLE>



```
test.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<HTML>
<HEAD><TITLE> 테이블 </TITLE></HEAD>
<BODY>
<CENTER> <BR>
<TABLE BORDER="1" WIDTH="100%">
<TR> <TH> 한국 </TH>
<TH> 미국 </TH>
<TH> 영국 </TH>
</TR>
<TR> <TD> 서울 </TD>
<TD> 워싱턴 D.C. </TD>
<TD> 런던 </TD>
</TR>
</TABLE>
<BR> <BR>
<TABLE BORDER="1" HEIGHT="150">
<TR> <TH> 한국 </TH>
<TH> 미국 </TH>
<TH> 영국 </TH>
</TR>
<TR> <TD> 서울 </TD>
<TD> 워싱턴 D.C. </TD>
<TD> 런던 </TD>
</TR>
</TABLE>
</CENTER>
</BODY>
</HTML>
```



다. 테이블 여백 조절

### 형식

- <TABLE CELSPACING = "픽셀수">  
: 셀과 셀 사이의 여백을 조절
- <TABLE CELLPADDING = "픽셀수">  
: 셀과 문자 사이의 여백을 조절

```

test.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<HTML>
<HEAD><TITLE> 테이블의 여백
</TITLE></HEAD>
<BODY>
<CENTER> <BR>
<TABLE BORDER="1" CELSPACING="10">
<TR>
<TD>첫번째 셀</TD>
<TD>두번째 셀</TD>
</TR>
</TABLE>
<P>
<TABLE BORDER="1" CELLPADDING="10">
<TR>
<TD>첫번째 셀</TD>
<TD>두번째 셀</TD>
</TR>
</TABLE>
</CENTER>
</BODY>
</HTML>

```



라. 테이블 색 지정

```
형 식 :  
<TABLE BGCOLOR = "색상명" 또는 "RGB 값"> ~ </TABLE>  
<TR BGCOLOR = "색상명" 또는 "RGB 값"> ~ </TR>  
<TD BGCOLOR = "색상명" 또는 "RGB 값"> ~ </TD>
```

### 3.4.2 테이블 정렬

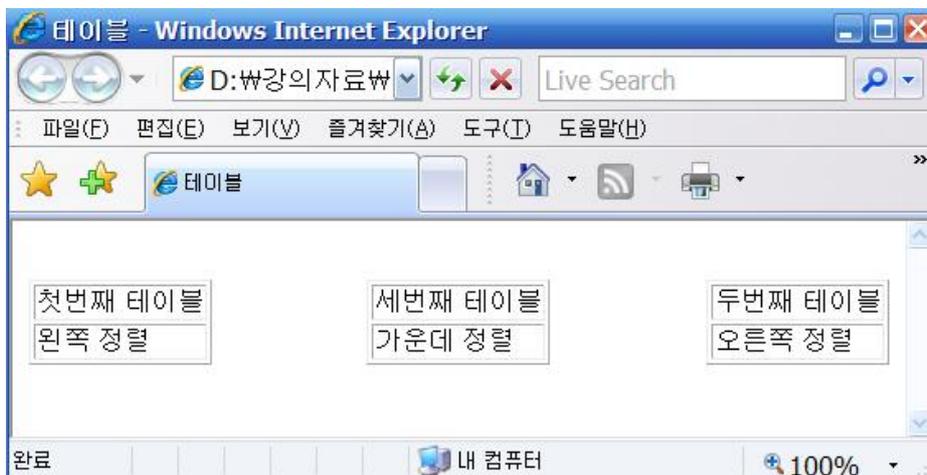
<Table> 태그의 align 속성을 활용하여 테이블을 정렬할 수 있다.

```
형 식 : <TABLE ALIGN = "LEFT" 또는 "RIGHT"> ~ </TABLE>
        <CENTER>
        <TABLE> ~ </TABLE>
        </CENTER>
```



```
test.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<HTML>
<HEAD> <TITLE> 테이블 </TITLE> </HEAD>
<BODY>
<BR>
<TABLE ALIGN = "LEFT" BORDER="1">
<TR> <TD>첫번째 테이블</TD> </TR>
<TR> <TD>왼쪽 정렬</TD> </TR>
</TABLE>

<TABLE ALIGN = "RIGHT" BORDER="1">
<TR> <TD>두번째 테이블</TD> </TR>
<TR> <TD>오른쪽 정렬</TD> </TR>
</TABLE>
<CENTER>
<TABLE BORDER="1">
<TR> <TD>세번째 테이블</TD> </TR>
<TR> <TD>가운데 정렬</TD> </TR>
</TABLE>
</CENTER>
</BODY>
</HTML>
```



## 3.5 프레임 만들기

프레임(frame) 태그를 사용하여 웹 문서의 한 화면을 여러 개의 창으로 분할할 수 있으며, 창마다 독립적인 기능을 포함할 수 있다. 프레임은 효과적인 웹 문서제작에 자주 사용된다.

### 3.5.1 프레임의 기본 구조

프레임의 기본 구조는 다음과 같다.

- <FRAMESET>: 프레임의 선언
- <FRAME>: 프레임의 분할

```
<HTML>
  <HEAD> <TITLE>프레임의 기본구조</TITLE> </HEAD>
  <FRAMESET>
    <FRAME>
    <FRAME>
  </FRAMESET>
</HTML>
```

형식 :

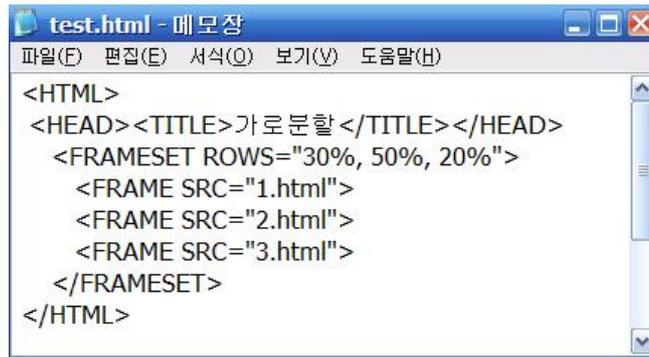
```
<FRAMESET ROWS 또는 COLS = "프레임 수와 크기 지정">
  <FRAME SRC = "프레임에 표시할 파일명">
</FRAMESET>
```

프레임의 크기는 다음의 방식으로 지정할 수 있다.

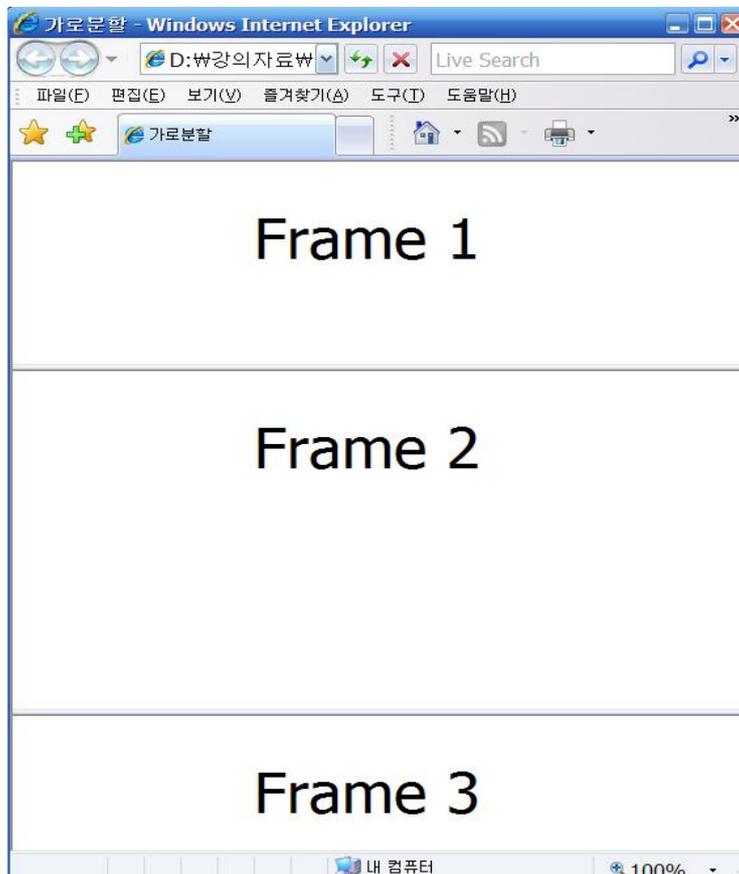
- 픽셀 단위로 지정
- 웹 브라우저 크기에 대한 비율(%)로 지정
- 나머지 전부를 \*로 지정

가. 프레임의 가로 분할

다음 예제는 프레임을 3개의 가로 창으로 분할하는 (30%, 50%, 20%) 예제이다.



```
<HTML>
<HEAD><TITLE>가로분할</TITLE></HEAD>
  <FRAMESET ROWS="30%, 50%, 20%">
    <FRAME SRC="1.html">
    <FRAME SRC="2.html">
    <FRAME SRC="3.html">
  </FRAMESET>
</HTML>
```



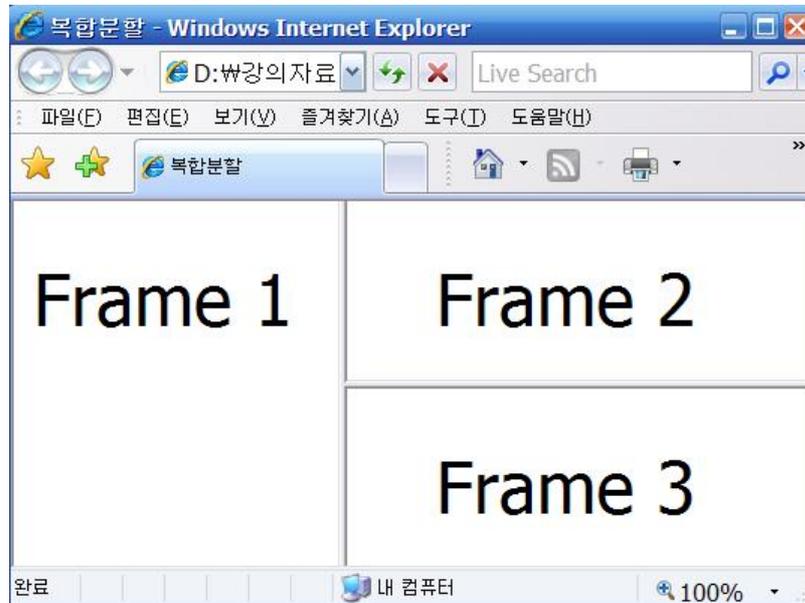
나. 프레임의 세로 분할

```
test.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<HTML>
<HEAD><TITLE>세로분할</TITLE></HEAD>
  <FRAMESET COLS="300, 500">
    <FRAME SRC="1.html">
    <FRAME SRC="2.html">
  </FRAMESET>
</HTML>
```



다. 가로 및 세로 분할

```
test.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<HTML>
<HEAD> <TITLE>복합분할</TITLE> </HEAD>
<FRAMESET COLS="150, *">
  <FRAME SRC="1.html">
  <FRAMESET ROWS="50%, 50%">
    <FRAME SRC="2.html">
    <FRAME SRC="3.html">
  </FRAMESET>
</FRAMESET>
</HTML>
```



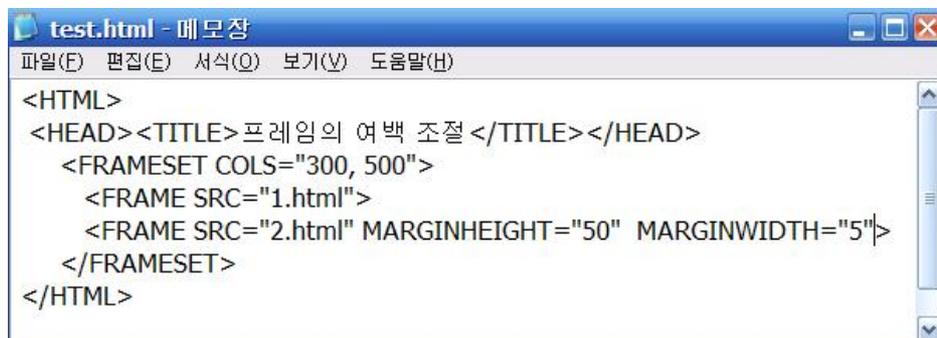
### 3.5.2 프레임의 서식 지정

프레임의 테두리선 변경은 다음 형식을 따른다.

**형 식 :** <FRAMESET BORDER="픽셀수">

프레임의 여백 조절은 다음 형식을 따른다.

**형 식 :**  
 <FRAME MARGINWIDTH="픽셀수" MARGINHEIGHT="픽셀수">





프레임의 크기 변경을 금지하고자 하는 경우에는 다음 형식으로 지정할 수 있다.

형 식 : <FRAME NORESIZE>

프레임의 스크롤바 지정을 통해, 스크롤바의 표시 여부를 결정할 수 있다.

형 식 :  
<FRAME SCROLLING = "YES" 또는 "NO" 또는 "AUTO">

### 3.5.3 타겟 프레임의 지정

현재 프레임에서 특정 프레임으로 이동하고자 하는 경우 타겟 프레임을 지정할 수 있다.

형 식 : <FRAME NAME = "프레임 이름" >  
<A HREF = "타겟 파일 이름" TARGET = "문서를 출력할  
프레임의 이름">  
브라우저에 나타날 텍스트  
</A>

```
test.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

<HTML>
<HEAD><TITLE>타겟 프레임 지정 </TITLE></HEAD>
  <FRAMESET COLS="200, *">
    <FRAME SRC = "left.html" NAME = "left">
    <FRAME SRC = "right.html" NAME = "right">
  </FRAMESET>
</HTML>
```

```
left.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

<HTML>
<HEAD><TITLE>타겟 지정하기</TITLE></HEAD>
<BODY>
<CENTER>
  <H2>메뉴</H2>
  <A HREF="1.html" TARGET="right">프레임 1</A> <P>
  <A HREF="2.html" TARGET="right">프레임 2</A> <P>
</CENTER>
</BODY>
</HTML>
```

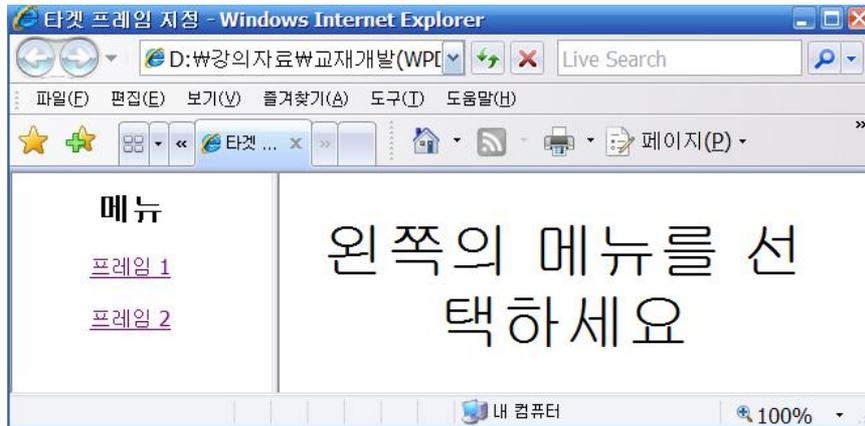
```
right.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

<HTML>
<CENTER><BR>
<FONT SIZE="10">
왼쪽의 메뉴를 선택하세요
</HTML>
```

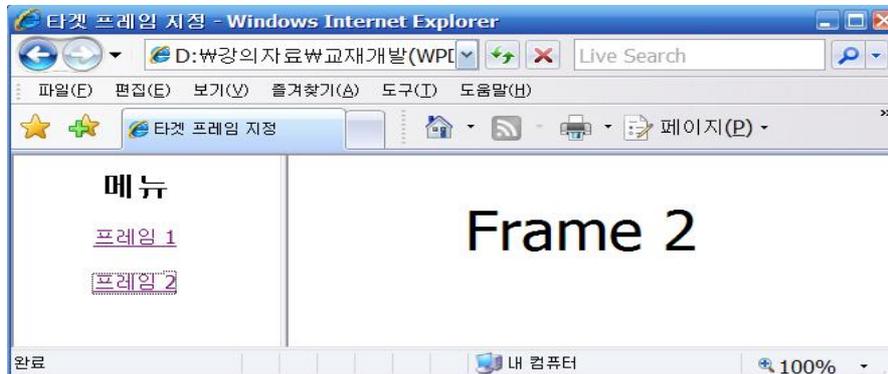
```
2.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

<HTML>
<CENTER><BR>
<FONT SIZE="10">
Frame 2
</HTML>
```

“test.html”을 실행하는 경우 결과 화면은 다음과 같다.



위 화면에서 “프레임 2”를 선택한 경우의 결과 화면은 다음과 같다.



그 외 다양한 타겟 기능을 정리하면 아래와 같다.

- TARGET = “\_blank” : 기존 프레임 외에 새로운 윈도우를 생성한 후 링크된 문서를 표시한다.
- TARGET = “\_self” : 링크된 문서를 링크가 있는 현재 프레임에 표시한다. (기본)
- TARGET = “\_top” : 프레임이 없어지고 윈도우 전체에 링크된 문서를 표시한다.
- TARGET = “\_parent” : 현재 프레임을 없애고 현재 프레임의 parent 프레임에 링크된 문서를 표시한다.

## 3.6 입력 양식 만들기

정보의 조직적 표현을 테이블(table)이 자주 사용된다. 이 절에서는 기본적인 테이블 태그 및 테이블의 세밀한 제어를 위해 사용되는 태그들을 알아본다.

양방향의 웹 서비스 구현을 위해서는 사용자와 웹 서버간의 상호적인 정보 교류가 필수적이다. HTML에서는 사용자로부터 정보 입력을 위해 다양한 입력 양식을 제공한다.

### 3.6.1 입력 양식 태그

사용자의 입력 데이터를 웹 서버로 전송하기 위해 <FORM> 태그를 사용한다.

형 식 :

```
<FORM ACTION = "URL" METHOD = "GET" 또는 "POST"> </FORM>
```

- ACTION : 입력 데이터를 처리할 CGI 프로그램의 URL이다.  
생략되면 현재 문서가 사용된다.
- METHOD : 사용자가 입력한 데이터를 서버에 전달하는 방법이다.
  - \* GET : - 디폴트로 데이터가 환경변수를 통하여 서버에 전달된다.  
- 데이터의 라인 수가 제한된다.
  - \* POST : - 데이터가 표준 입력 방식으로 서버에 전달된다.  
- 데이터의 라인 수에 제한이 없으므로 주로 사용된다.

한편, <INPUT> 입력 태그를 사용하여 다양한 속성을 정의할 수 있다.

```
형 식 : <INPUT TYPE = [속 성]  
        NAME = "입력 양식 이름"  
        VALUE = "입력 양식에서 표시될 문자열"  
        SIZE = "문자 수"  
        CHECKED = [속 성]  
        MAXLENGTH = "문자 수" >
```

각 속성은 아래와 같이 다양한 값을 가질 수 있다.

▪ TYPE

- TEXT : 문자 입력
- PASSWORD : 문자 입력 시 데이터를 \*로 표시
- CHECKBOX : 체크상자를 생성
- RADIO : 여러 개 중 하나를 선택하는 입력 양식 생성
- HIDDEN : 숨겨진 입력 양식을 생성
- IMAGE : 이미지 입력 양식을 생성
- FILE : 파일 업로드 입력 양식을 생성
- SUBMIT : CGI 프로그램에 데이터를 전달
- RESET : 입력 내용을 취소

▪ NAME

- 입력한 내용을 지정할 변수의 이름을 지정
- SUBMIT과 RESET을 제외한 모든 TYPE의 필수요소

▪ VALUE

- TEXT, PASSWORD : 입력 값의 기본값을 지정
- CHECKBOX, RADIO : 해당 버튼이 선택되었을 때 CGI에 보내는 값을 정의
- SUBMIT, RESET : 버튼에 표시될 내용을 지정

▪ CHECKED

- CHECKBOX, RADIO의 기본값으로 선택

▪ SIZE = "문자수" (기본값=20)

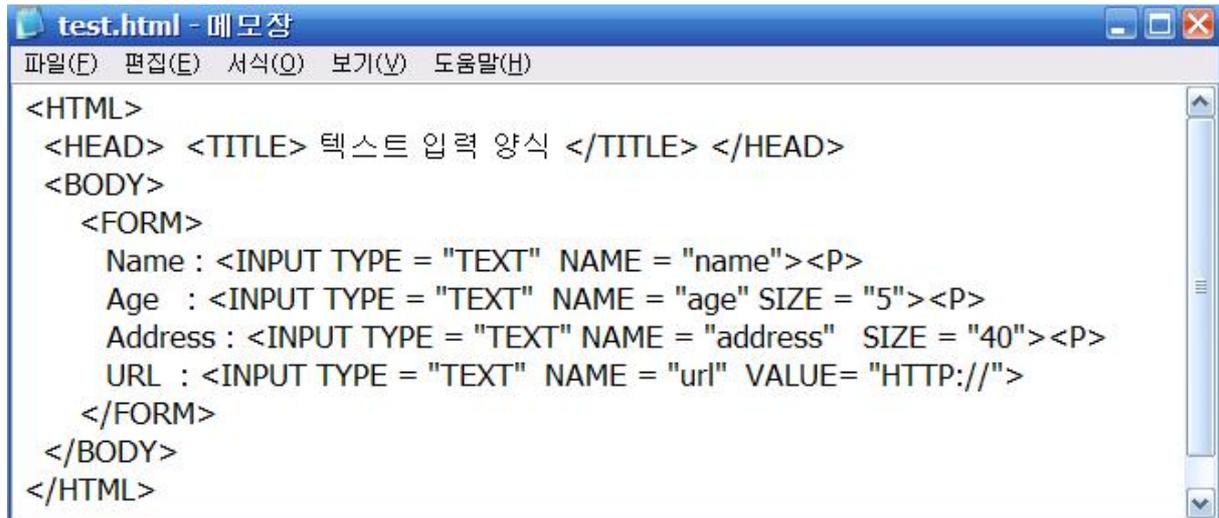
- TEXT, PASSWORD 입력 부분에 보이는 문자수를 정의

▪ MAXLENGTH = "문자수"

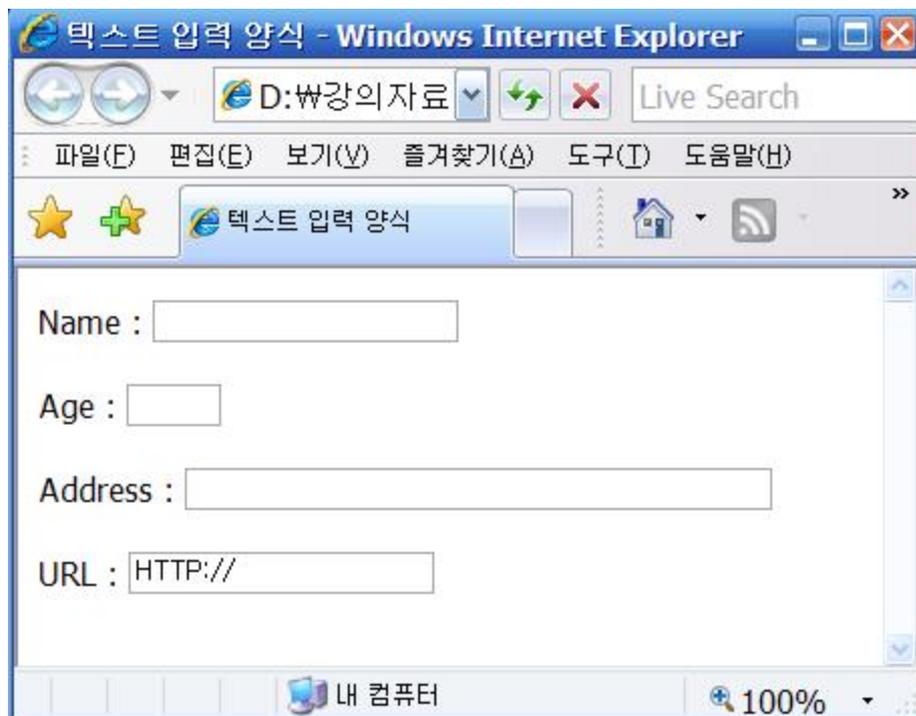
- TEXT, PASSWORD에서 입력 받을 수 있는 최대 문자수를 정의

### 3.6.2 입력 양식 만들기 예제

<INPUT TYPE = "TEXT"> 예제



```
<HTML>
<HEAD> <TITLE> 텍스트 입력 양식 </TITLE> </HEAD>
<BODY>
  <FORM>
    Name : <INPUT TYPE = "TEXT" NAME = "name"><P>
    Age  : <INPUT TYPE = "TEXT" NAME = "age" SIZE = "5"><P>
    Address : <INPUT TYPE = "TEXT" NAME = "address" SIZE = "40"><P>
    URL  : <INPUT TYPE = "TEXT" NAME = "url" VALUE = "HTTP://">
  </FORM>
</BODY>
</HTML>
```



<INPUT TYPE = "PASSWORD"> 예제

```
test.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<HTML>
<HEAD> <TITLE> PASSWORD 입력 양식 </TITLE> </HEAD>
<BODY>
<FORM>
  Your Name : <INPUT TYPE = "PASSWORD" NAME= "PS" >
</FORM>
</BODY>
</HTML>
```



<INPUT TYPE = "CHECKBOX"> 예제

```
test.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<HTML>
<HEAD><TITLE> CHECKBOX 입력 양식 </TITLE></HEAD>
<BODY>
<FORM>
  <INPUT TYPE = "CHECKBOX" NAME = "F1" VALUE = "A" > 사과 <BR>
  <INPUT TYPE = "CHECKBOX" NAME = "F2" VALUE = "O" CHECKED> 귤 <BR>
  <INPUT TYPE = "CHECKBOX" NAME = "F3" VALUE = "B" > 바나나
</FORM>
</BODY>
</HTML>
```



## <INPUT TYPE = "RADIO"> 예제

```
test.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<HTML>
<HEAD> <TITLE> RADIO 입력 양식 </TITLE> </HEAD>
<BODY>
<FORM>
<INPUT TYPE = "RADIO" NAME = "F1" VALUE = "A"> 사과
<INPUT TYPE = "RADIO" NAME = "F1" VALUE = "O" CHECKED> 귤
<INPUT TYPE = "RADIO" NAME = "F1" VALUE = "B"> 바나나
</FORM>
</BODY>
</HTML>
```



## <SELECT> 선택 목록 양식 예제

```
test.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<HTML>
<HEAD> <TITLE> 선택 목록 입력 양식 </TITLE> </HEAD>
<BODY>
<FORM>
<SELECT NAME = "fruit" SIZE = "3">
<OPTION> 바나나
<OPTION> 딸기
<OPTION> 복숭아
<OPTION> 포도
<OPTION> 사과
</SELECT>
</FORM>
</BODY>
</HTML>
```



### 3.6.3 문서 입력창 만들기

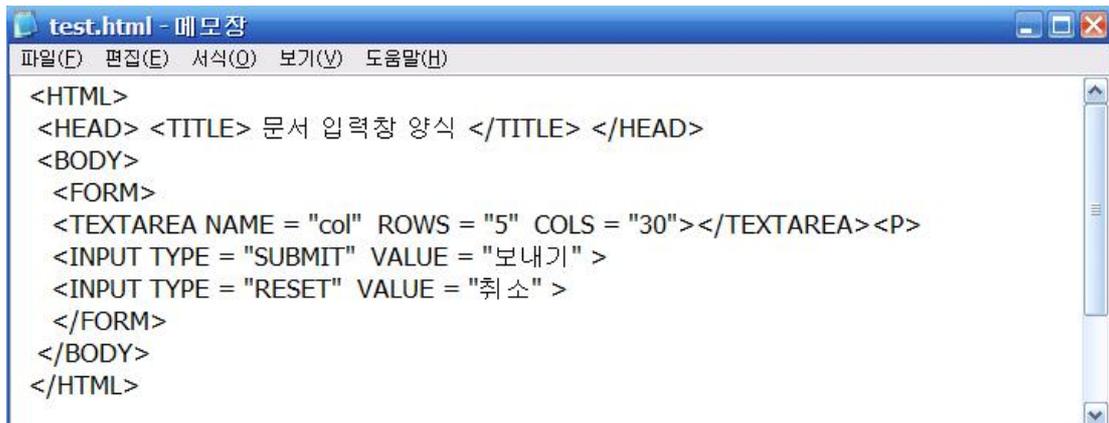
<TEXTAREA> 태그를 사용하여 문서입력 창을 만들 수 있다.

형 식 : <TEXTAREA [속성]> ~ </TEXTAREA>

- NAME = "변수명" : 입력한 내용을 저장할 변수의 이름을 지정한다.
- ROWS = "행의 수" : 브라우저에 표시되는 줄 수를 지정한다. (기본값은 1)
- COLS = "열의 수" : 브라우저에 표시되는 열 수를 지정한다. (기본값은 20)

다음 예제에서는 가로 5줄, 세로 30줄의 문서 입력 창을 제공한다.

- "INPUT TYPE=SUBMIT": 전송 기능
- RESET 속성: 취소 기능



```
<HTML>
<HEAD> <TITLE> 문서 입력창 양식 </TITLE> </HEAD>
<BODY>
  <FORM>
    <TEXTAREA NAME = "col" ROWS = "5" COLS = "30"></TEXTAREA><P>
    <INPUT TYPE = "SUBMIT" VALUE = "보내기" >
    <INPUT TYPE = "RESET" VALUE = "취소" >
  </FORM>
</BODY>
</HTML>
```



### 3.7 멀티미디어 활용하기

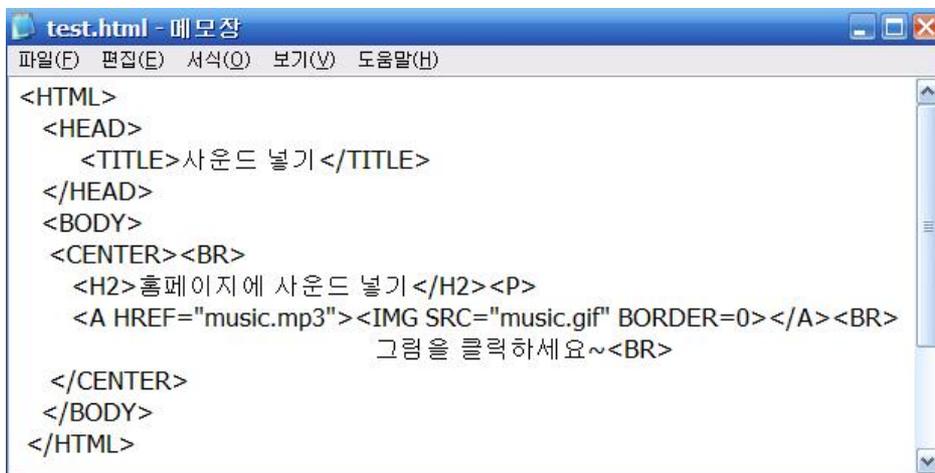
HTML 문서안에서 사운드 혹은 동영상 들의 멀티미디어 파일을 재생시킬 수 있다.

#### 3.7.1 사운드 포함하기

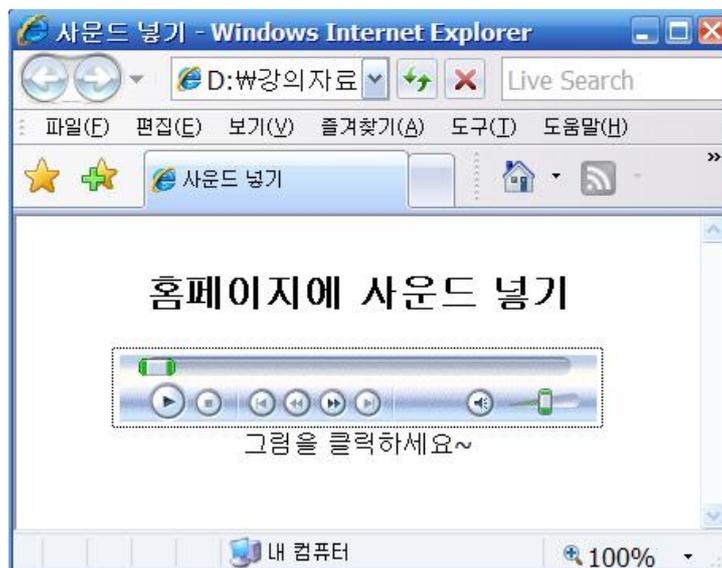
HTML 문서에 사운드 넣기는 일반적인 HTML을 작성하는 것과 동일하다. 링크의 대상을 사운드 파일의 URL이나 파일명을 지정하면 된다.

형 식 :

```
<A HREF="사운드 파일명"> 브라우저에 표시할 문자 </A>  
<A HREF="사운드 파일명"><IMG SRC="이미지 파일의 URL"> </A>
```



```
test.html - 메모장  
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)  
<HTML>  
  <HEAD>  
    <TITLE>사운드 넣기</TITLE>  
  </HEAD>  
  <BODY>  
    <CENTER><BR>  
      <H2>홈페이지에 사운드 넣기</H2><P>  
      <A HREF="music.mp3"><IMG SRC="music.gif" BORDER=0></A><BR>  
        그림을 클릭하세요~<BR>  
    </CENTER>  
  </BODY>  
</HTML>
```



HTML 문서에 배경 음악 넣기 기능을 활용하여, 홈페이지의 내용이 화면에 나타남과 동시에 음악 재생을 시작할 수 있다. 방문자들에게 홈페이지의 내용을 효과적으로 전달하고자 하는 경우에 사용되며, 다음 2가지 방식이 주로 사용된다.

<EMBED> 태그를 이용하여 사운드 넣기

형 식 :

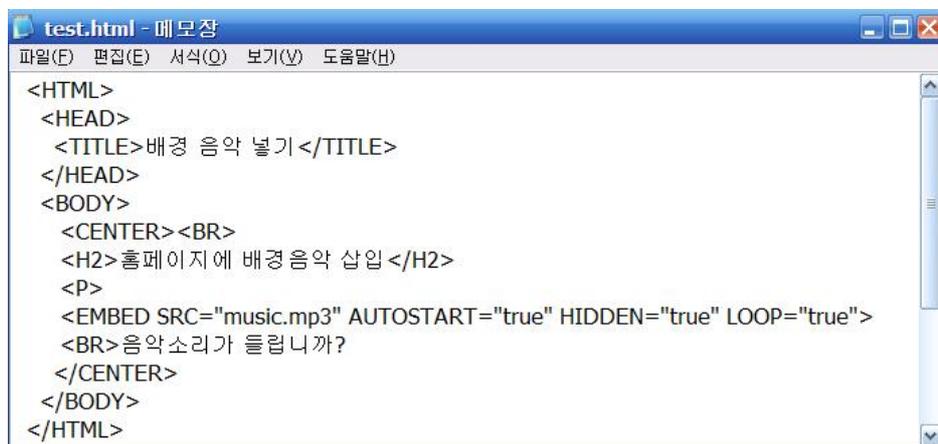
```
<EMBED SRC="사운드 파일명" AUTOSTART="true" 또는 "false"
: 홈페이지 접속과 동시에 재생
HIDDEN="true" 또는 "false"
: 플레이어 숨김 여부 지정
LOOP="true" 또는 "false"
: 무한 반복 여부 지정
VOLUME="숫자">
: 볼륨 설정 (범위: -10000~0)
```

<BGSOUND> 태그를 이용하여 사운드 넣기

형 식 :

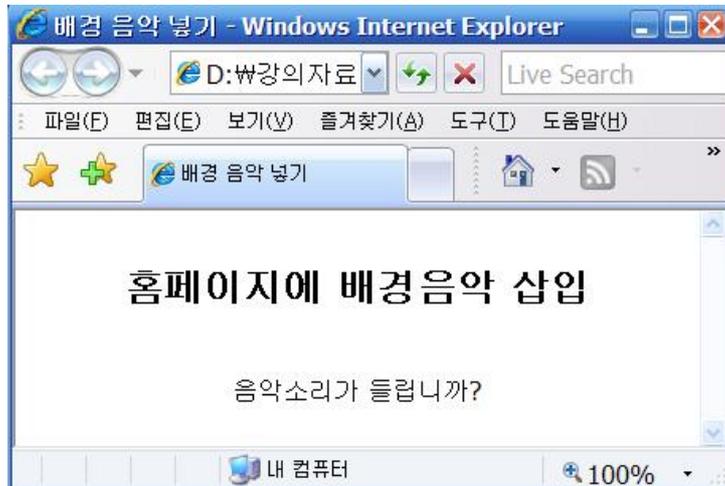
```
<BGSOUND SRC="사운드 파일명" LOOP="숫자" 또는 "infinite"
: 반복 회수 및 무한 반복 지정
VOLUME="숫자" >
: 볼륨 설정 (범위: -10000~0)
```

다음은 <EMBED> 태그를 사용한 예제이다.

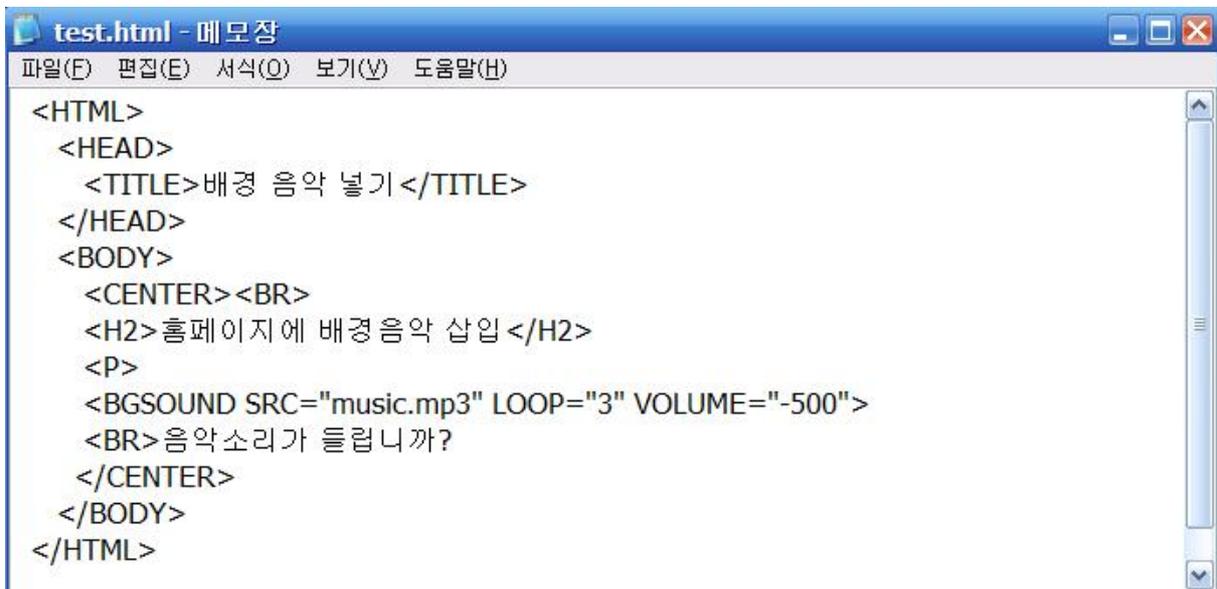


```
test.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

<HTML>
<HEAD>
<TITLE>배경 음악 넣기</TITLE>
</HEAD>
<BODY>
<CENTER><BR>
<H2>홈페이지에 배경음악 삽입</H2>
<P>
<EMBED SRC="music.mp3" AUTOSTART="true" HIDDEN="true" LOOP="true">
<BR>음악소리가 들립니까?
</CENTER>
</BODY>
</HTML>
```



다음은 <BGSOUND> 태그를 사용한 예제이다.



결과 화면은 상기 예제와 같다.

### 3.7.2 동영상 포함하기

사운드와 마찬가지로 HTML 문서에 동영상 넣는 방법도 일반적인 HTML을 작성법과 동일하다. 링크의 대상을 동영상 파일의 URL이나 동영상 파일명을 지정한다.

<EMBED> 태그를 사용하여 동영상 넣기

형 식 :

```
<EMBED SRC = "동영상 파일명"           : 재생할 동영상 파일명 지정
      AUTOSTART = "true" 또는 "false"  : 자동실행 속성 지정
      WIDTH = "픽셀 단위의 숫자"       : 동영상의 가로 폭
      HEIGHT = "픽셀 단위의 숫자">    : 동영상의 세로 폭
```

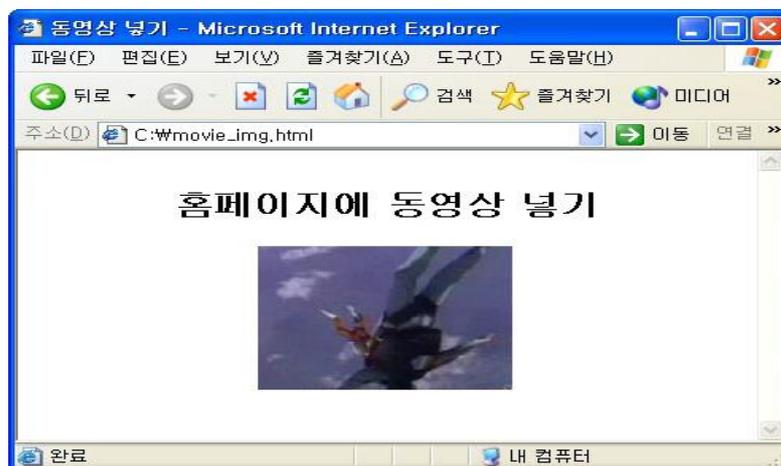
```
<HTML>
<HEAD>
  <TITLE>동영상 넣기</TITLE>
</HEAD>
<BODY>
  <CENTER><BR>
  <H2>홈페이지에 동영상 넣기</H2>
  <P>
  <EMBED SRC="test.mpg" AUTOSTART="true" WIDTH="160"
          HEIGHT = "120">
  </CENTER>
</BODY>
</HTML>
```

<IMG> 태그를 사용하여 동영상 넣기

형 식 :

<IMG SRC = "이미지 파일명" : DYN SRC 속성을 지원하지 않는 브라우저에서 보여줄 이미지  
DYN SRC = "동영상 파일명" : 재생할 동영상 파일명 지정  
WIDTH = "픽셀 단위의 숫자" : 동영상의 가로 폭  
HEIGHT = "픽셀 단위의 숫자 "> : 동영상의 세로 폭

```
<HTML>  
<HEAD>  
  <TITLE>동영상 넣기</TITLE>  
</HEAD>  
<BODY >  
  <CENTER><BR>  
  <H2>홈페이지에 동영상 넣기</H2> <P>  
  <IMG SRC="movie.gif" DYN SRC="test.mpg" WIDTH="160"  
    HEIGHT = "120">  
  </CENTER>  
</BODY>  
</HTML>
```



## 제 4 장. CSS

### 4.1 CSS 개요

2 장에서 언급하였듯이 클라이언트 측의 동적인(dynamic) HTML 문서를 제작하기 위해서는 자바스크립트 및 CSS(Cascading Style Sheet) 등의 기법이 요구된다. CSS 를 포함한 스타일시트 기법은 문서의 외형을 다양하게 표현하기 위해 생긴 개념이다. 우리가 흔히 사용하는 HWP 및 Word의 스타일시트와 유사한 개념이며, 구체적으로 다음의 기능들을 제공한다.

- 페이지 레이아웃과 폰트, 색상, 배경 및 웹 페이지 상의 구성 요소들의 스타일을 세부적으로 제어
- 웹 페이지의 스타일을 문서의 내용과 분리시켜 스타일 정의 부분의 수정으로 전체 문서의 스타일을 변경
- 스타일을 별도의 문서 파일로 유지하여 하나의 문서만 수정하고도 한꺼번에 여러 페이지의 외형 수정 가능

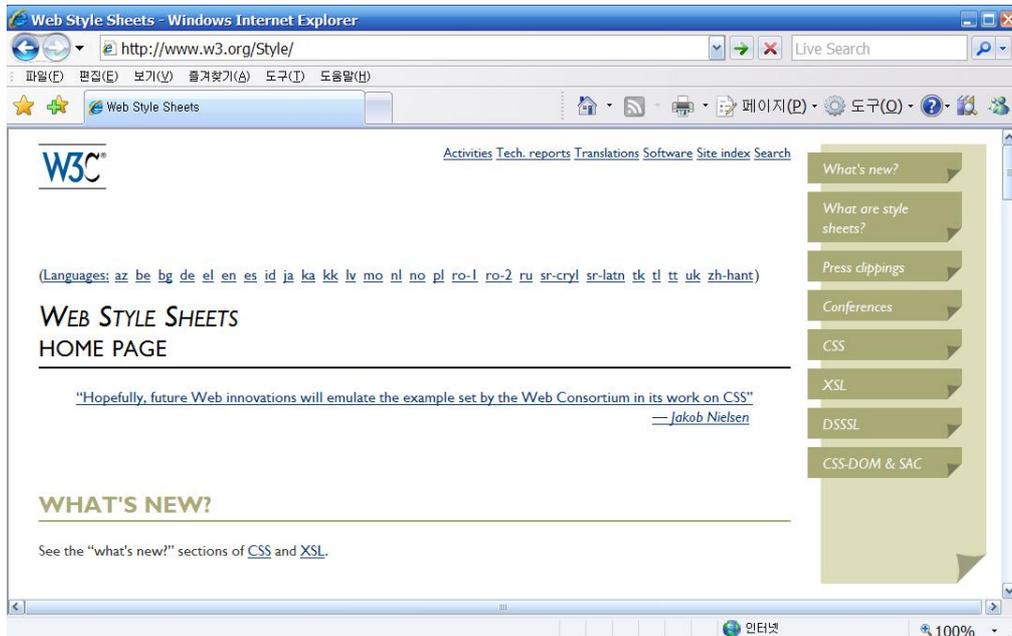
HTML 에서는 문서가 브라우저에서 표현되는 방식인 스타일을 지정하기 위하여 태그와 속성을 이용한다. 따라서, 내용마다 각각 스타일을 정의해 주어야 하므로 동일한 내용을 반복하는 불편함이 따른다. 특히, 문서의 내용과 스타일이 혼재되어 HTML 소스를 이해하는데도 어려움이 있다. 이에, 다음의 이유로 스타일시트의 사용이 권장된다.

- 문서의 물리적 스타일 정보를 논리적인 내용으로부터 분리
- 문서의 내용을 알아보기 용이하게 함
- 반복되는 스타일 정의로 인한 불편 해소
- 스타일의 변경 용이
- 스타일 시트는 여러 문서에도 반복 적용이 가능
- 이에 따라, 일관성 있는 웹 문서 제작이 용이

스타일 시트에는 다음 종류가 있다.

- CSS(Cascading Style Sheet): 속성과 속성값을 이용하여 스타일을 정의
- JSSS(JavaScript Style Sheet): 자바스크립트로 스타일 정보를 제어하는 방법

JSSS 는 이전의 넷스케이프에서 지원되던 방식이고, 최근에는 W3C 의 표준인 (아래 그림 참조) CSS 방법을 사용하는 것이 보편적이다.



구체적으로 CSS 를 사용하면, 스타일의 변경과 웹 페이지의 일관성을 유지가 용이하다는 장점이 있다.

- <H3> 태그에 이탤릭 효과와 글자 색으로 녹색을 지정하기 위해서 HTML에서는 모든 <H3>태그에 대해 일일이 다음과 같이 설정

```
<H3> <I> <FONT color="green"> 녹색 이탤릭 </FONT> </I> </H3>
```

- 스타일 시트를 사용하면 다음과 같이 <H3>의 스타일을 선언함

```
H3 { font-style : italic ; color : green }
```

- 한 번만 정의해 주면 웹 페이지에 포함된 모든 <H3> 태그에 같은 스타일을 적용함

또한, 스타일 시트를 사용하게 되면 문서의 내용과 스타일 정보가 분리되어 다음과 같은 장점을 얻을 수 있다.

- 문서의 내용을 알아보기 쉽게 함
- 똑같은 내용의 스타일 정보를 반복해서 작성해야 하는 번거로운 작업을 피할 수 있음
- 스타일 시트 부분만을 수정하여 문서 전체의 스타일을 변경할 수 있음
- 스타일의 변경이 보다 쉽고 일관성 있게 이루어짐

## 4.2 스타일 시트의 사용법

스타일 시트를 HTML 문서에 삽입하는 방법은 크게 다음 3가지로 구분된다.

- 내부 스타일 시트 (Embedded Style Sheet): 같은 웹 페이지 안에서 정의
- 외부 스타일 시트 (Linked Style Sheet): 스타일 정보를 별도의 문서로 저장하고 이를 참조하여 사용하는 방법
- 라인 스타일 시트 (Inline Style Sheet): 일반 태그에 스타일을 지정하여 특정 태그 범위 내에서만 스타일을 적용하는 방법

### 4.2.1 내부 스타일 시트

일반적인 방법으로 HTML 문서의 <HEAD> 태그 내에 스타일 정보를 기술하여 정의한다.

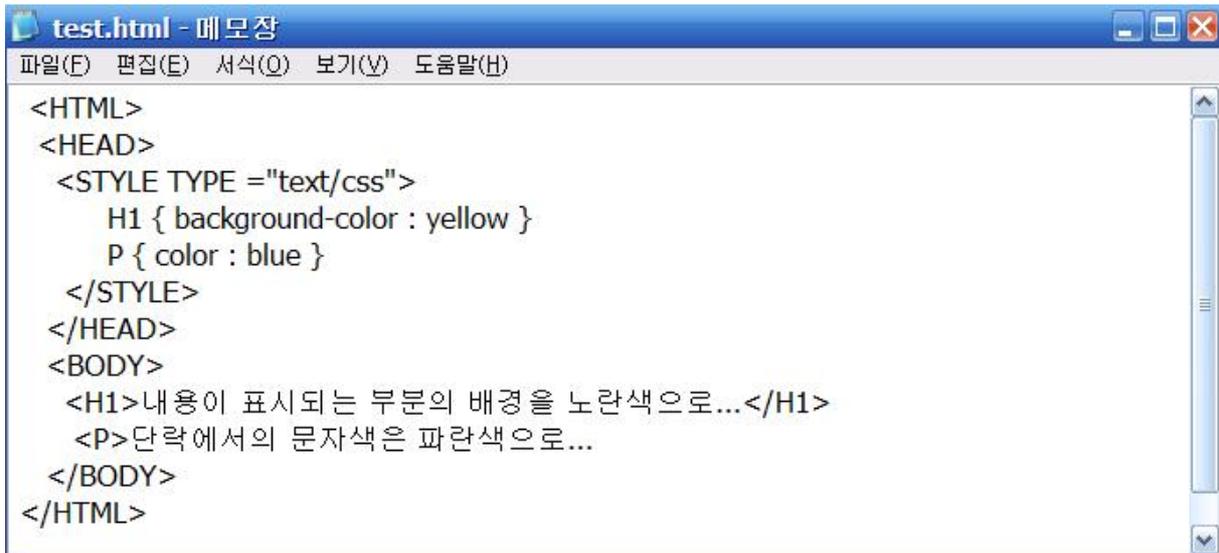
- <STYLE> 태그는 <HEAD> 태그와 </HEAD> 태그 사이에 위치
- <STYLE> 태그에서 지정한 스타일을 웹 페이지 전체에 반영

```
형식 : <HEAD>
      <STYLE TYPE="text/css" 또는 "text/JavaScript">
      <!--
          선택자 { 속성 : 속성값 }
      -->
      </STYLE>
</HEAD>
```

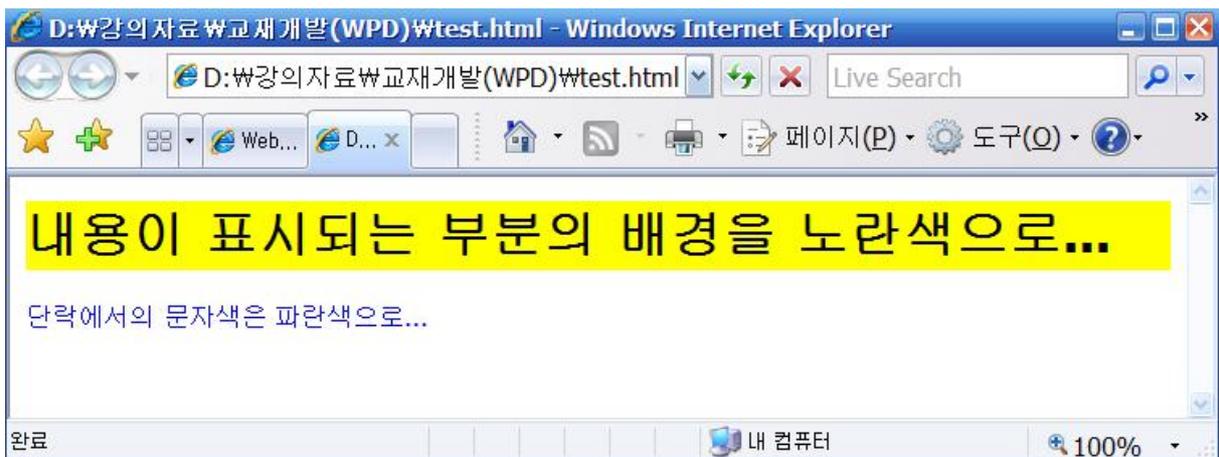
TYPE 속성에는 스타일 시트를 정의하는 문법에 따라 속성값을 지정해 주어야 한다

- CSS는 "text/css", JSS는 "text/JavaScript"로 속성값을 지정
- 넷스케이프와 익스플로러에서 모두 스타일을 사용할 수 있도록 일반적으로 CSS 방식을 주로 사용
- 주석 표시인 '<!--'와 '-->'는 스타일 시트를 지원하지 않는 브라우저가 스타일을 무시하도록 하기 위해 사용

다음 예제는 제목 태그인 <H1> 태그의 배경을 노란색으로 하고, 단락 태그인 <P> 태그에 포함되는 문자들을 파란색으로 나타내기 위한 스타일을 정의하고 있다.



```
<HTML>
<HEAD>
  <STYLE TYPE = "text/css">
    H1 { background-color : yellow }
    P { color : blue }
  </STYLE>
</HEAD>
<BODY>
  <H1>내용이 표시되는 부분의 배경을 노란색으로...</H1>
  <P>단락에서의 문자색은 파란색으로...
</BODY>
</HTML>
```



## 4.2.2 외부 스타일 시트

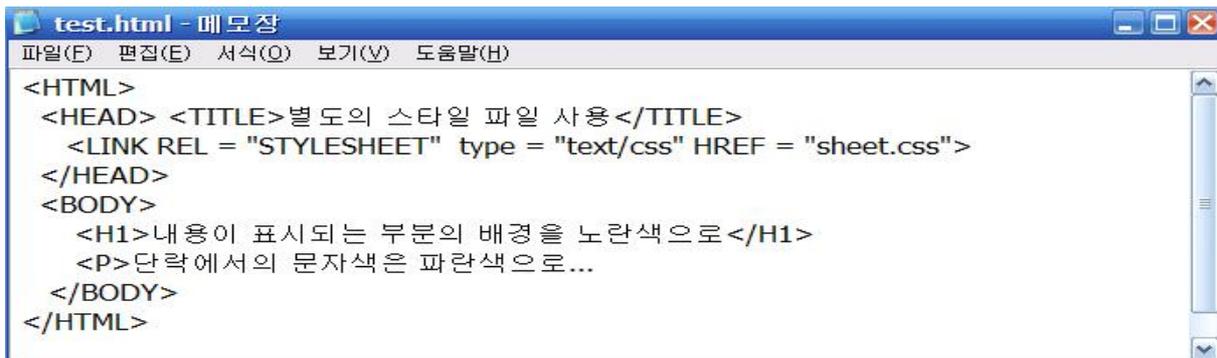
스타일을 정의하는 스타일 시트 문서를 만들어 별도의 파일(\*.css)로 저장하는 방법이다.

- ▶ HTML의 <LINK> 태그에서 URL을 지정하여 연결
- ▶ TYPE 속성을 지정하고 HREF 속성에서 URL을 지정
- ▶ 스타일 시트 문서가 단독 파일로 존재하기 때문에 여러 개의 HTML 문서들이 스타일을 공유

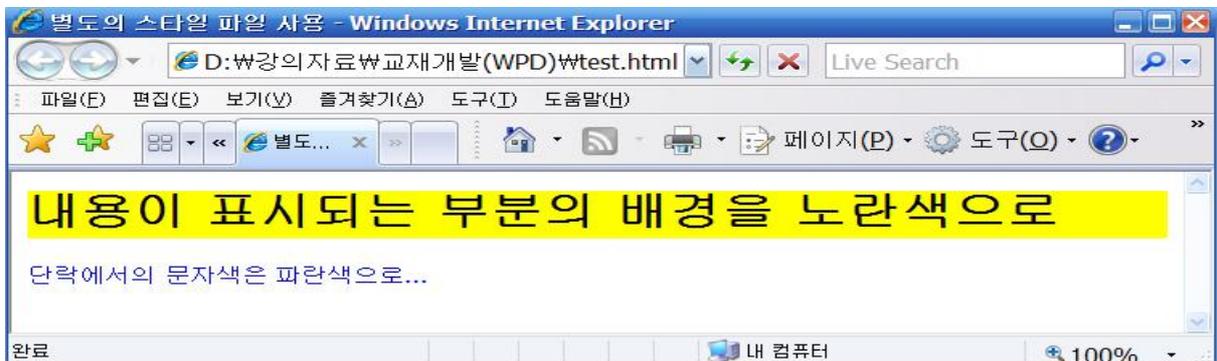
```
형 식 : <HEAD>  
        <LINK REL = "STYLESHEET" TYPE = "text/css" 또는  
        "text/JavaScript" HREF="스타일 시트의 URL">  
</HEAD>
```



```
sheet.css - 메모장  
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)  
H1 { background : yellow }  
P { color : blue }
```



```
test.html - 메모장  
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)  
<HTML>  
<HEAD> <TITLE>별 도의 스타일 파일 사용</TITLE>  
<LINK REL = "STYLESHEET" type = "text/css" HREF = "sheet.css">  
</HEAD>  
<BODY>  
<H1>내용이 표시되는 부분의 배경을 노란색으로</H1>  
<P>단락에서의 문자색은 파란색으로...  
</BODY>  
</HTML>
```



### 4.2.3 라인 스타일 시트

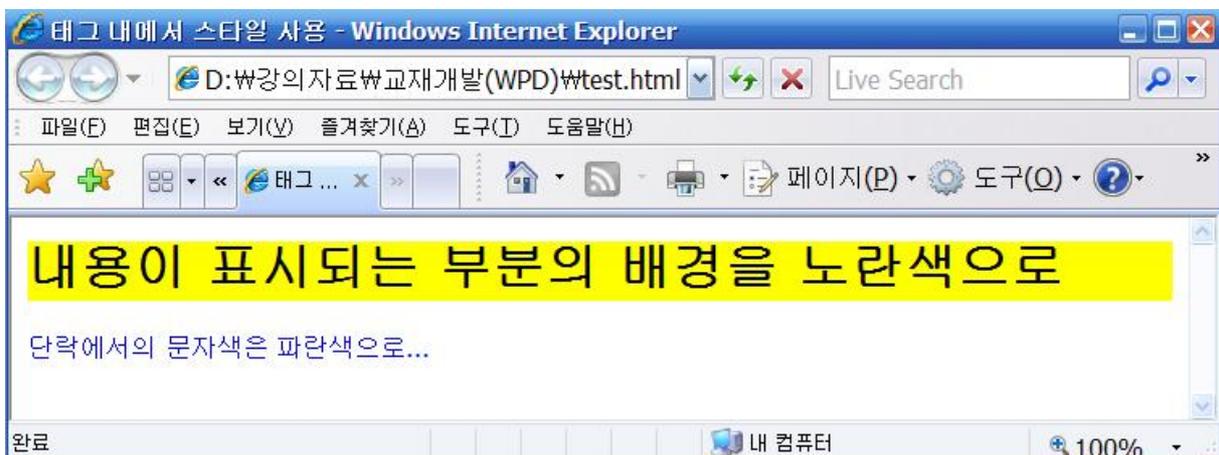
일반 태그에 스타일을 지정하는 방식이다.

- 해당 태그가 적용되는 범위에 한정하여 스타일을 적용
- 태그 안에서 스타일을 정의하고 스타일 옵션과 값을 설정
- 해당 태그의 종료 태그가 사용되기 전까지 지정된 스타일이 적용됨

형 식 : <태그 STYLE = "속성:속성값"> ~ </태그>



```
test.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<HTML>
<HEAD>
<TITLE>태그 내에서 스타일 사용</TITLE>
</HEAD>
<BODY>
<H1 STYLE = "background-color : yellow">
  내용이 표시되는 부분의 배경을 노란색으로</H1>
<P STYLE="color : blue">
  단락에서의 문자색은 파란색으로...</P>
</BODY>
</HTML>
```



## 4.3 스타일 정의 방법

### 4.3.1 기본 방법

스타일의 정의는 스타일을 적용할 대상과 대상에 적용할 속성과 속성값으로 구성된다.

- 스타일을 적용할 대상을 선택자(Selector)라 하며 일반적으로 HTML 태그들이 선택자로 사용됨
- 선택자를 지정하고 '{와 }' 안에 속성과 속성값으로 이루어진 스타일 정보를 기술
- 속성과 속성값은 콜론(:)으로 연결하여 정의

형 식 : 선택자 { 속성 : 속성값 }

- 예 : <P> 태그에 포함되는 내용을 빨간색으로 보이도록 <P> 태그의 스타일을 정의

P { color : red }

하나의 태그에 여러 개의 속성을 지정할 수도 있다.

- 하나의 태그에 대해 배경색과 글자 색 그리고 글자의 크기 등 다양한 속성을 동시에 적용
- 선언부에서 속성과 속성값을 원하는 만큼 나열하여 여러 가지 스타일을 지정
- 각 속성과 속성값은 세미콜론(;)으로 구분

형 식 : 선택자 { 속성1:속성값1; 속성2:속성값2; ... 속성n:속성값n }

- 예 : <P> 태그에 대해 배경은 파란색으로 지정하고, 글자는 노란색으로 지정

P { background : blue; color : yellow }

여러 가지 태그에 동일한 속성들을 지정해야 할 필요가 있을 때에는 다음과 같이 공통 특성을 정의하여 사용한다.

- 같은 스타일을 적용할 선택자들을 콤마(,)로 묶어 스타일을 정의
- 공통 속성이 여러 개이면 이들을 앞에서처럼 한꺼번에 정의
  - 형 식 : 선택자1, 선택자2, ... , 선택자n { 속성 : 속성값 }
- 다음 두 종류의 스타일은 동일한 결과를 나타냄
  - H1 { background : yellow }      H1, P { background : yellow }
  - P { background : yellow }

#### 4.3.2 CLASS 속성을 이용한 스타일 정의

클래스를 사용하면 원래의 태그 스타일이 있더라도 그 스타일을 적용하지 않고 클래스 스타일로 정의한 속성이 적용된다.

- 특정 태그에만 적용되는 클래스의 스타일을 정의하려면 선택자로 스타일을 적용하고자 하는 태그의 이름을 쓰고, 점(.)과 클래스 이름을 사용
  - 형 식 : 선택자 . 클래스명 { 속성 : 속성값 }
- 예 : <P> 태그 중 클래스 속성이 blue인 태그에만 스타일을 지정
  - P.blue { background : blue }

- 클래스로 지정된 스타일을 적용하기 위해 HTML 문서의 <BODY> 태그 내에서 다음과 같은 형식으로 사용
  - 형 식 : <태그명 CLASS = "클래스 이름"> ~ </태그명>
  - 예 : 태그가 적용되는 범위의 배경색으로 파란색을 지정

<P CLASS = "blue">이 단락은 CLASS 속성을 이용한 태그입니다.</P>

- 모든 태그에 적용되는 클래스 스타일을 적용하려면 다음과 같이 선택자에 태그 이름 없이 마침표(.)와 클래스 이름만을 사용
  - <BODY> 태그 안에 위치하는 어떤 태그에서도 태그의 CLASS 속성값으로 클래스 이름만 지정해주면 해당 스타일 사용 가능

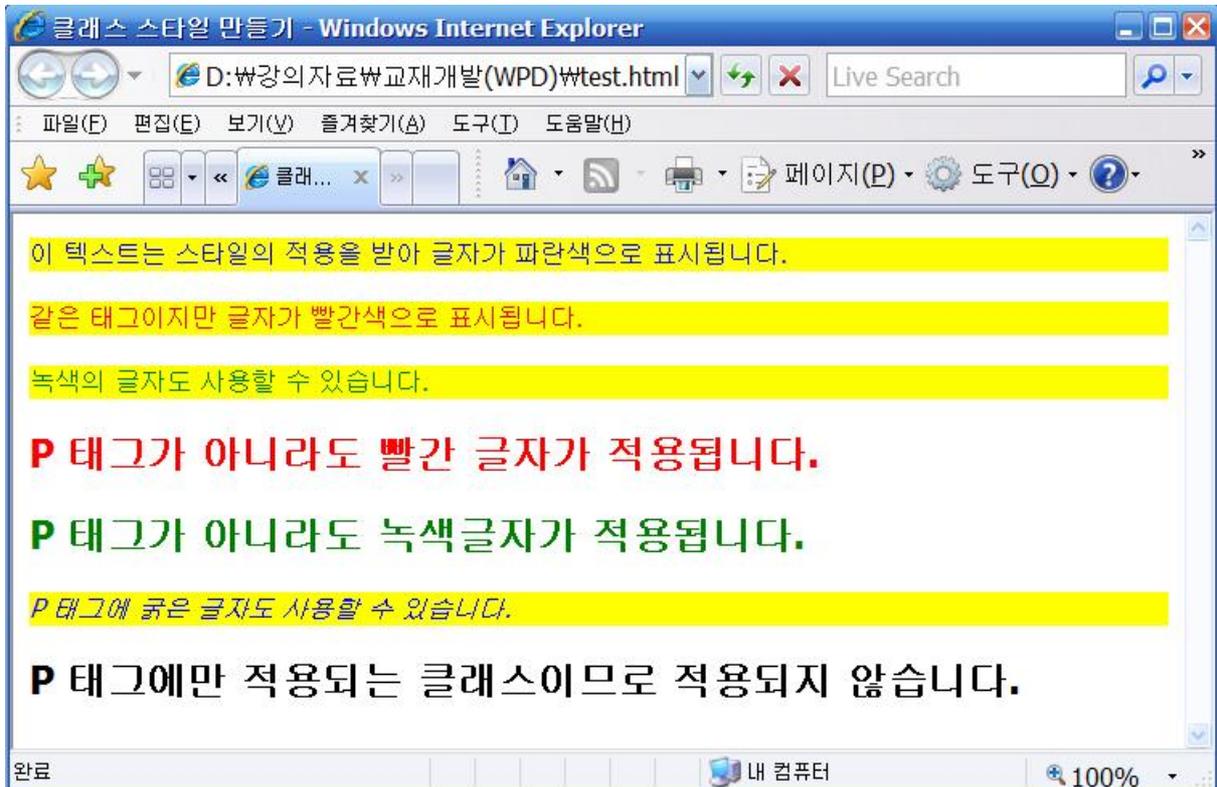
형 식 : . 클래스명 { 속성 : 속성값 }

다음은 태그에 적용되는 스타일과 모든 태그에 적용되는 스타일을 비교하는 예제이다.

```

test.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<HTML>
  <HEAD>
    <TITLE> 클래스 스타일 만들기 </TITLE>
    <STYLE TYPE ="text/css">
      P { background : yellow; color : blue }
      P.italic { font-style : italic }
      .redtxt { color : red }
      .greentxt { color : green }
    </STYLE>
  </HEAD>
  <BODY>
    <P> 이 텍스트는 스타일의 적용을 받아 글자가 파란색으로 표시됩니다.
    <P CLASS="redtxt"> 같은 태그이지만 글자가 빨간색으로 표시됩니다.
    <P CLASS="greentxt"> 녹색의 글자도 사용할 수 있습니다.
    <H2 CLASS="redtxt"> P 태그가 아니라도 빨간 글자가 적용됩니다.</H2>
    <H2 CLASS="greentxt"> P 태그가 아니라도 녹색글자가 적용됩니다.</H2>
    <P CLASS="italic"> P 태그에 굵은 글자도 사용할 수 있습니다.
    <H2 CLASS="italic"> P 태그에만 적용되는 클래스이므로 적용되지 않습니다.
  </H2>
  </BODY>
</HTML>
  
```

상기 예제의 결과 화면은 다음과 같다.



#### 4.3.3 ID 속성을 이용한 스타일 정의

스타일 시트를 정의할 때, 개별적으로 고유한 이름을 붙여서 스타일을 정의하는 방식이다.

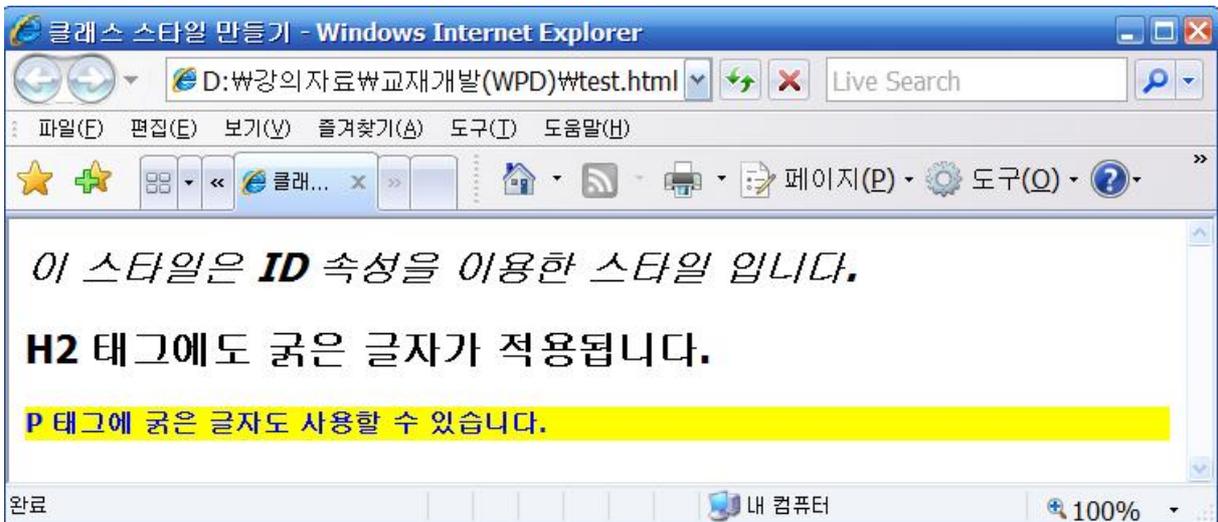
- 이미 지정된 스타일 속성 이외에 또 다른 속성을 지정하여 사용 가능
- CSS 구문에서는 다음의 형식과 같이 '#' 기호를 사용하여 ID 이름과 스타일을 정의

형 식 : #ID이름 { 속성 : 속성값 }

- HTML 태그에서는 ID 이름을 사용하여 해당 태그에 대한 스타일을 지정

형 식 : <태그명 ID = "ID이름"> ~ </태그명>

```
test.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<HTML>
<HEAD><TITLE>클래스 스타일 만들기</TITLE>
<STYLE TYPE = "text/css">
P { background : yellow; color : blue; }
.boldtxt { font-weight : bold; }
#once { font-style : italic;}
</STYLE>
</HEAD>
<BODY>
<H2 ID = "once"> 이 스타일은 ID 속성을 이용한 스타일 입니다.</H2>
<H2 CLASS = "boldtxt"> H2 태그에도 굵은 글자가 적용됩니다.</H2>
<P CLASS = "boldtxt"> P 태그에 굵은 글자도 사용할 수 있습니다.
</BODY>
</HTML>
```



## 4.4 스타일 정의를 위한 속성

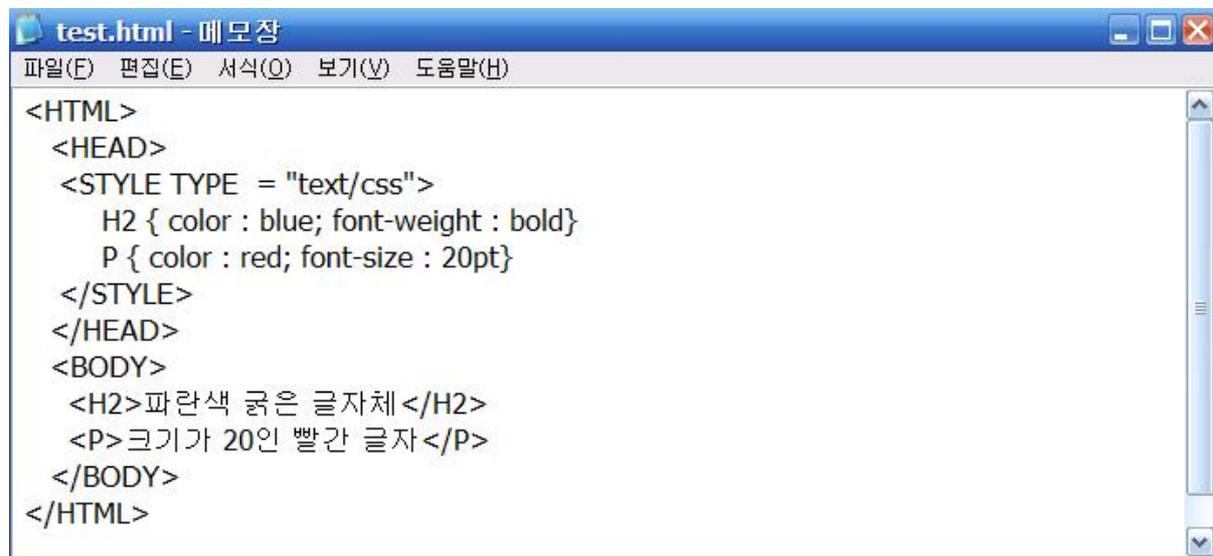
이 절에서는 스타일 시트를 활용하기 위해서 어떤 속성들이 있는지, 그리고 속성값으로 무엇이 올 수 있는지에 대해 살펴본다.

### 4.4.1 폰트 속성

폰트와 관련된 속성을 이용하여 글자체, 모양, 크기, 굵기 등의 스타일을 지정한다.

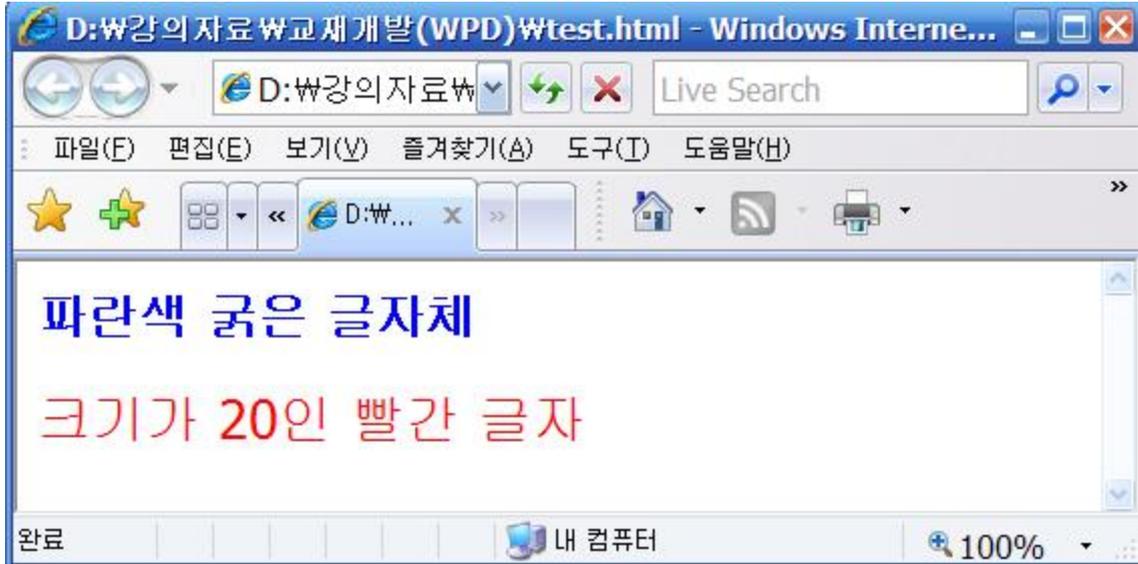
속성	속성값	설명과 사용 예
font-family	글자체(굴림, 바탕, arial, ...)	글자체 이름 font-family: Times New Roman, Arial
font-style	normal, italic, oblique	글자 모양 지정 font-style:italic
font-size	px(픽셀), pt(포인트), in(인치), cm(센티미터), %(퍼센트), xx-small, x-small, small, medium, large, x-large, xx-large, larger, smaller	글자 크기 font-size:16pt, font-size:20px
font-weight	normal, bold, light, bolder, lighter	글자 굵기 font-weight:bold
font-variant	normal, small-caps	작은 대문자 font-variant:small-caps

다음 예제에서 <H2> 태그는 font-weight 속성을 이용하여 파란색의 글자로 굵게 표시하였고, <P> 태그에 대해서는 빨간색 글자로 20포인트의 크기로 문자열을 출력하고 있다.



```

test.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<HTML>
  <HEAD>
    <STYLE TYPE = "text/css">
      H2 { color : blue; font-weight : bold}
      P { color : red; font-size : 20pt}
    </STYLE>
  </HEAD>
  <BODY>
    <H2>파란색 굵은 글자체 </H2>
    <P>크기가 20인 빨간 글자</P>
  </BODY>
</HTML>
  
```



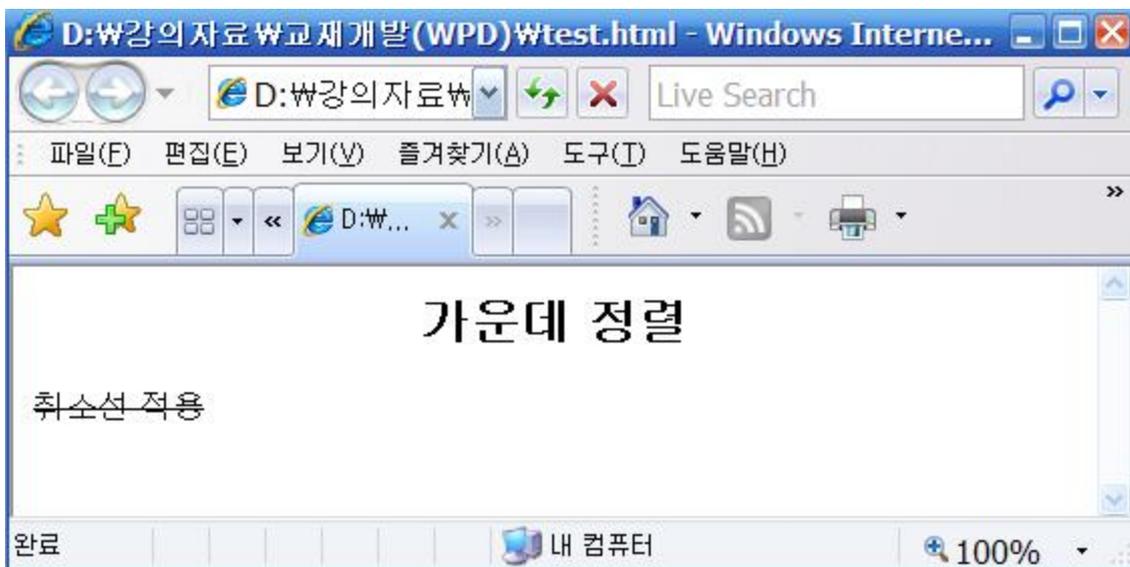
#### 4.4.2 텍스트 속성

텍스트의 배치, 모양, 간격 등을 자유롭게 조절한다.

속성	속성값	설명과 사용 예
text-align	left, center, right, justify	가로 정렬, text-align:center
vertical-align	baseline, bottom, middle, text-bottom, text-top, top, sub, super	세로 정렬 vertical-align:middle
text-indent	pt(포인트), px(픽셀), in(인치), cm(센티미터) 또는 %(퍼센트)	들여쓰기 text-indent:30px
text-decoration	none, underline, overline, line-through, blink	글자 꾸밈 text-decoration:underline
text-transform	none, uppercase, lowercase, capitalize	대소문자 변환 text-transform:uppercase
word-spacing	pt, px, in, cm	단어 간격, word-spacing:30px
letter-spacing	pt, px, in, cm	글자 간격, letter-spacing:0.5cm
line-height	pt, px, in, cm	줄 간격, line-height:10pt

다음 예제에서는 텍스트 관련 속성 중 text-align 속성을 이용하여 속성값을 center로 지정하여 가운데 정렬을 수행하고, text-decoration 속성의 속성값을 line-through로 지정하여 문자열에 취소선을 적용한다.

```
test.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<HTML>
<HEAD>
  <STYLE TYPE = "text/css">
    H2 { text-align : center}
    P { text-decoration : line-through}
  </STYLE>
</HEAD>
<BODY>
  <H2>가운데 정렬</H2>
  <P>취소선 적용</P>
</BODY>
</HTML>
```



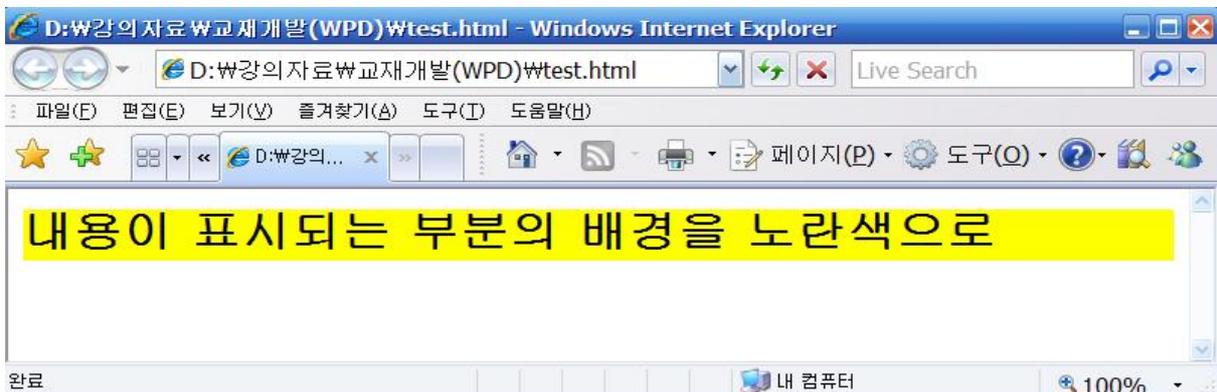
#### 4.4.3 색과 배경 속성

스타일 시트를 이용하면 배경색 뿐만 아니라 이미지를 연결하여 배경 이미지로 지정 가능하다.

속성	속성값	설명과 사용 예
color	색상명 또는 RGB 값	색상 color : red
background-color	색상명 또는 RGB 값	배경색 background-color : blue
background-image	배경 이미지의 URL	배경 이미지 background-image : url("sky.jpg")
background-repeat	no-repeat, repeat-x, repeat-y, repeat	배경 이미지 반복 background-repeat : repeat-x
background-attachment	fixed, scroll	배경 이미지 고정 background-attachment : fixed
background-position	top, bottom, left, right, center (조합해서 사용 가능)	배경 이미지 위치 background-position : top left

```

test.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<HTML>
<HEAD>
  <STYLE TYPE = "text/css">
    H1 { background-color : yellow }
  </STYLE>
</HEAD>
<BODY>
  <H1>내용이 표시되는 부분의 배경을 노란색으로</H1>
</BODY>
</HTML>
  
```



#### 4.4.4 위치 속성

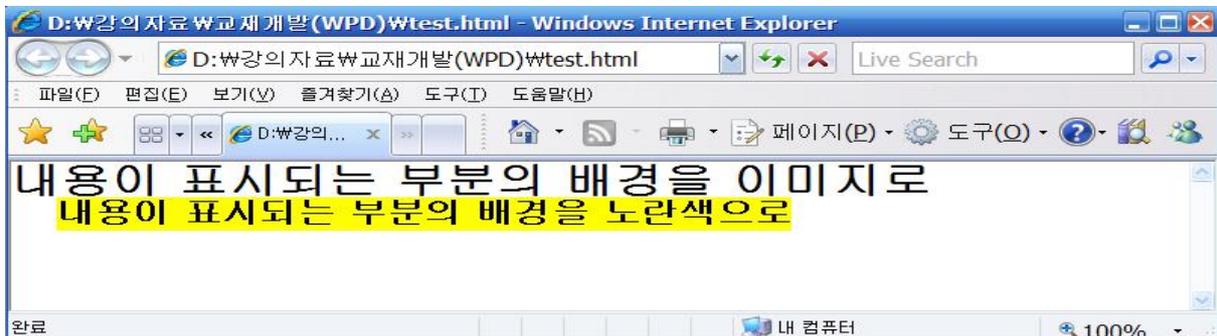
웹 문서의 텍스트, 이미지 등의 위치를 세부적으로 지정 가능하다.

속성	속성값	설명과 사용 예
position	absolute, relative	위치 속성, position : absolute
left	pt, px, in, cm, %, auto	좌측 여백 지정, left : 20pt
top	pt, px, in, cm, %, auto	상단 여백 지정, top : 10px
width	pt, px, in, cm	요소의 넓이 지정, width : 150pt
height	pt, px, in, cm	요소의 높이 지정, height : 120px
visibility	visible, hidden	보이기, 감추기 속성 지정, visibility : visible
z-index	정수 값, auto (값이 작을수록 아래에 있는 층)	겹치는 순서 지정, z-index : 1, z-index : 3

다음 예제는 브라우저에 나타나는 요소의 위치를 절대 위치로 지정하고 z-index 속성과 top, left 속성을 이용하여 내용이 표시되는 부분이 약간 겹쳐 보이도록 스타일을 정의하고 있다.

```

test.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<HTML>
<HEAD>
  <STYLE TYPE = "text/css">
    H1 { position : absolute; background-image :
      url (./balloon .jpg); top : 0; left :0; z-index :1 }
    H2 { position : absolute; background-color:yellow;
      top :30; left : 30; z-index:2; }
  </STYLE>
</HEAD>
<BODY>
  <H1>내용이 표시되는 부분의 배경을 이미지로</H1>
  <H2>내용이 표시되는 부분의 배경을 노란색으로</H2>
</BODY>
</HTML>
  
```



#### 4.4.5 목록 속성

스타일 시트의 목록 속성에서는 HTML의 <UL> 또는 <LI> 태그에서 불릿(Bullet)의 표현을 확장하여 불릿의 종류를 임의로 지정할 수 있다. 또한, HTML에서 사용되는 불릿의 형태 뿐만 아니라 이미지를 사용 가능하다.

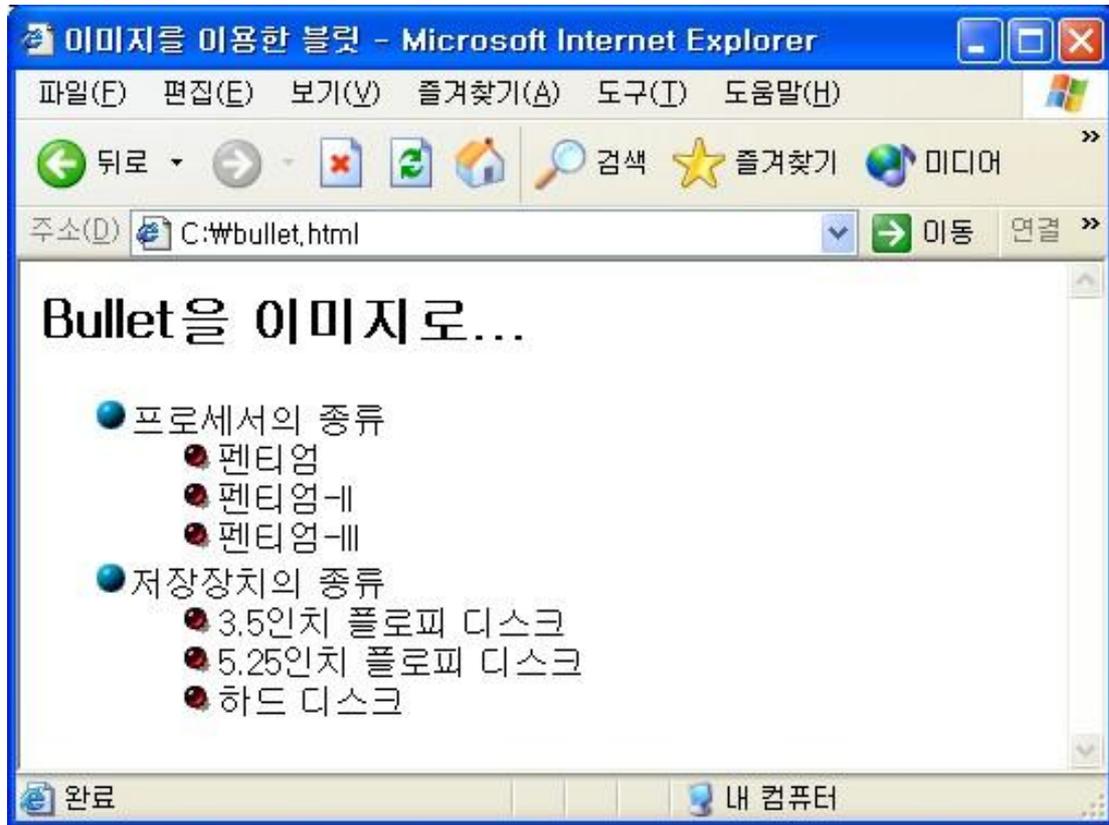
속성	속성값	설명과 사용 예
list-style-type	none, disk, circle, square, decimal, lower-roman, upper-roman, lower-alpha, upper-alpha	불릿의 모양 지정 list-style-type:circle
list-style-image	불릿 이미지의 URL	불릿의 이미지 지정 list-style-image:url("arrow.gif")
list-style-position	inside, outside	목록의 위치 지정 list-style-position:outside

다음은 불릿의 형태로 이미지를 사용한 예이다. 두개의 이미지 파일이 필요하다.

```

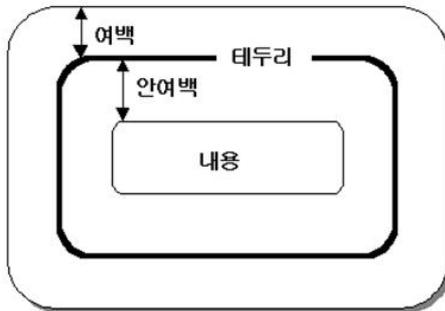
test.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<HTML>
  <HEAD><TITLE>이미지를 이용한 불릿</TITLE>
  <STYLE TYPE = "text/css">
    UL LI { list-style-image : url("blet1.gif") }
    UL LI UL LI { list-style-image : url("blet2.gif") }
  </STYLE>
</HEAD>
<BODY>
  <H2>Bullet을 이미지로...</H2>
  <UL>
    <LI>프로세서의 종류
      <UL>
        <LI>펜티엄</LI>
        <LI>펜티엄-II</LI>
        <LI>펜티엄-III</LI>
      </UL>
    </LI>
    <LI>저장장치의 종류
      <UL>
        <LI>3.5인치 플로피 디스크</LI>
        <LI>5.25인치 플로피 디스크</LI>
        <LI>하드 디스크</LI>
      </UL>
    </LI>
  </UL>
</BODY>
</HTML>

```



#### 4.4.6 박스 스타일 속성

스타일 시트를 사용하여, 블록(Block) 요소가 상자로 둘러 싸여 있는 것처럼 처리할 수 있다. 각 상자는 여백(margin)과 테두리(border), 안여백(padding) 등의 구성을 포함하고, 배경 이미지나 색상 지정도 가능하다.



여백 (margin) : 블록 테두리의 외부 여백  
 테두리 (border) : 블록 테두리의 외형 특성  
 안여백 (padding) : 테두리와 블록의 내용 사이의 간격  
 내용 (content) : 문서의 내용

박스 스타일 속성에는 다음이 있다.

속성	속성값	설명과 사용 예
width	pt, px, in, cm, %	내용 영역의 너비 width : 140pt
height	pt, px, in, cm, %	내용 영역의 높이 height : 50px
float	none, left, right	블록 영역의 위치 float : right
border-style	double, inset, groove, solid, ridge, outset	테두리의 형태 border-style : solid
border-width	pt, px, in, cm, %	테두리의 폭 border-width : 20pt
border-color	색상명 또는 RGB 값	테두리의 색상 border-color : navy
margin, margin-top, margin-bottom, margin-left, margin-right	pt, px, in, cm, %	여백 margin-top : 50pt
padding, padding-top, padding-bottom, padding-left, padding-right	pt, px, in, cm, %	안여백 padding-left : 10px

다음 예제는 테두리(border)와 관련된 속성과 속성값을 이용하여 스타일을 정의한다.

```
test.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<HTML>
<HEAD> <TITLE>border 속성 예제 문서 </TITLE>
<STYLE>
.style1 {border-style:double; border-width:10pt; border-color:skyblue}
.style2 {border-style:inset; border-width:10pt; border-color:navy}
.style3 {border-style:groove; border-width:10pt; border-color:skyblue}
.style4 {border-style:solid; border-width:10pt; border-color:navy}
.style5 {border-style:ridge; border-width:10pt; border-color:skyblue}
.style6 {border-style:outset; border-width:10pt; border-color:navy}
</STYLE>
</HEAD>
<BODY>
<TABLE> <TR>
<TD> <P CLASS = "style1">DOUBLE BORDER.</P> </TD>
<TD> <P CLASS = "style2">INSET BORDER.</P> </TD>
<TD> <P CLASS = "style3">GROOVE BORDER.</P> </TD>
</TR>
<TR>
<TD> <P CLASS = "style4">SOLID BORDER.</P> </TD>
<TD> <P CLASS = "style5">RIDGE BORDER.</P> </TD>
<TD> <P CLASS = "style6">OUTSET BORDER.</P> </TD>
</TR>
</TABLE>
</BODY>
</HTML>
```



## 제 5 장. 자바 스크립트

### 5.1 자바 스크립트 소개

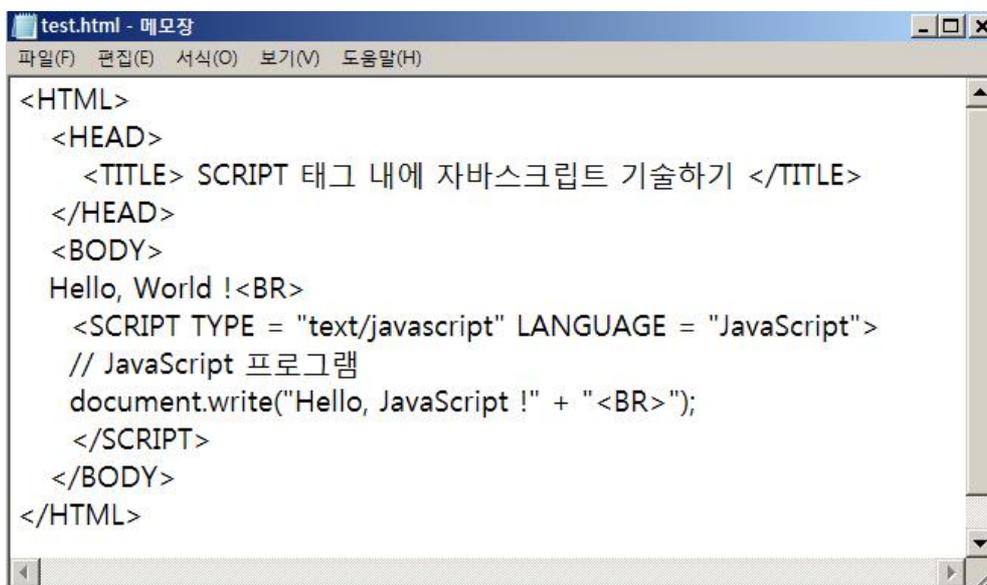
자바 스크립트(Java Script)란?

- 대화형 인터넷 환경 구축을 위해 넷스케이프 2.0 에서 소개된 간단한 객체지향언어
- 현재, 넷스케이프 4.0 이상에서 버전 1.2 가 사용됨
- ECMA(European Computer Manufacturers Association) 표준문서에 정의
- 플랫폼에 독립적으로 해석되며, 자바스크립트를 지원하는 브라우저에서 실행
- 마이크로소프트의 인터넷 익스플로러는 자바스크립트와 호환 가능한 Jscript 를 자체 구현하여 내장하고 있음

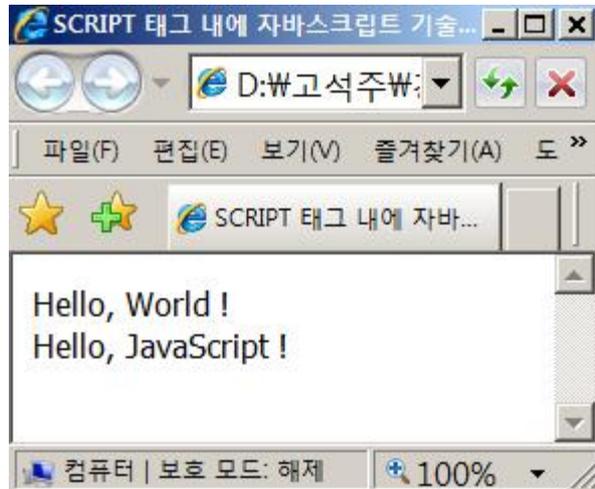
자바 스크립트의 특징을 정리하면 다음과 같다.

- 주로 HTML 문서에 삽입되어 브라우저의 인터프리터에 의해 실행되는 객체지향 언어
- 사용자에게 의해 발생하는 사건(Event)을 처리할 수 있는 동적인 홈페이지 제작을 가능하게 함
- 인터넷 환경에 적합하게 설계
- HTML 과 브라우저간 상호작용 할 수 있는 기능을 제공

먼저 간단한 예제를 살펴보자.



```
test.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<HTML>
  <HEAD>
    <TITLE> SCRIPT 태그 내에 자바스크립트 기술하기 </TITLE>
  </HEAD>
  <BODY>
    Hello, World !<BR>
    <SCRIPT TYPE = "text/javascript" LANGUAGE = "JavaScript">
      // JavaScript 프로그램
      document.write("Hello, JavaScript !" + "<BR>");
    </SCRIPT>
  </BODY>
</HTML>
```



## 5.2 자바스크립트 기초

### 5.2.1 HTML과 자바스크립트

다음과 같이 HTML 문서안에 자바 스크립트를 적용할 수 있다.

```
<script src="" type="" language="" defer="" >
  // JavaScript 프로그램
</script >
```

- src
  - 소스코드 파일 위치에 대한 URL을 명시
  - \*.js를 확장자로 가짐.
- type
  - "text/javascript"
- language
  - LiveScript
  - JavaScript
- defer
  - 스크립트에 의해 문서가 추가 생성되지 않음을 명시
  - boolean(true/false) 값을 가짐.

## 5.2.2 자바스크립트의 어휘

어휘란 문법적으로 의미있는 최소 단위를 말하며, 보통 토큰이라고도 한다. 토큰의 종류는 다음과 같다.

- 명칭(Identifier)
- 예약어(Reserved Words)
- 리터럴(Literal)
- 주석(Comments)

### 가. 명칭(identifier)

변수와 함수를 식별하기 위하여 붙이는 이름으로써, 다음 사항에 유의해야 한다.

- 대소문자를 구분
- 문자, 밑줄(\_), \$로 시작
- 길이에선 제한이 없다.
- 지정어는 명칭으로 사용 불가

**바른 명칭 :** sum, sum1, \_sum, money\_sum, moneySum, money\$sum  
**틀린 명칭 :** 1sum, sum!, money#sum, money sum, function

### 나. 예약어(Reserved Words)

- 프로그래밍 언어 설계 시에 그 기능과 용도가 이미 지정되어 있는 단어
- 변수명이나 함수명, 메소드명, 객체명으로 사용 불가

현재 사용 중인 자바스크립트의 예약어는 다음과 같다.

<b>break</b>	<b>for</b>	<b>new</b>	<b>var</b>
<b>continue</b>	<b>function</b>	<b>return</b>	<b>void</b>
<b>delete</b>	<b>if</b>	<b>this</b>	<b>while</b>
<b>else</b>	<b>in</b>	<b>typeof</b>	<b>with</b>

향후 사용 예정인 자바스크립트의 예약어는 다음과 같다.

<b>abstract</b>	<b>do</b>	<b>import</b>	<b>short</b>
<b>boolean</b>	<b>double</b>	<b>instanceof</b>	<b>static</b>
<b>byte</b>	<b>enum</b>	<b>int</b>	<b>super</b>
<b>case</b>	<b>export</b>	<b>interface</b>	<b>switch</b>
<b>catch</b>	<b>extends</b>	<b>long</b>	<b>synchronized</b>
<b>char</b>	<b>final</b>	<b>native</b>	<b>throw</b>
<b>class</b>	<b>finally</b>	<b>package</b>	<b>throws</b>
<b>const</b>	<b>float</b>	<b>private</b>	<b>transient</b>
<b>debugger</b>	<b>goto</b>	<b>protected</b>	<b>try</b>
<b>default</b>	<b>implements</b>	<b>public</b>	<b>volatile</b>

#### 다. 리터럴(Literal)

자신의 표기법이 곧 자신의 값이 되는 상수를 의미하며, 리터럴의 종류는 다음과 같다.

- 숫자(numeric)
- 문자열(string)
- 논리형(boolean) 상수
- 널(null): 아무 것도 없음을 표현하는 용도로 사용

#### 라. 주석(Comments)

- 프로그램의 이해를 돕기 위해 코드를 설명하는 문장
- 프로그램 실행에는 아무런 영향을 주지 않으나, 프로그램 유지보수에 중요한 역할 수행
- 한 줄 주석(line comment): `// Comment`
- 텍스트 주석(text comment): `/* Comment */`

### 5.2.3 자바스트립트의 자료형

자바 스크립트 언어의 자료 저장을 위해 변수(Variable)를 사용한다.

- 변수 이름은 명칭 형태
- 예약어를 사용할 수 없다
- 선언부와 정의부로 구성
- 선언 예:

```
var a, b, c; // 변수 a, b, c 선언.
```

```
var t = true; // 변수의 선언과 함께 값을 정의.
```

한편, 배열은 동일한 자료 객체 여러 개가 모인 구조체이며, 자바스크립트에서는 배열을 객체로 취급한다.

### 5.2.4 연산자(operator)

자바 스크립트에서 사용하는 연산자는 다음과 같다.

- 산술 연산자 : +, -, \*, /, %,
- 비교 연산자 : ==, !=, >, <, >=, <=
- 논리 연산자 : &&, ||, !
- 비트 연산자 : &, |, ^, ~, <<, >>, >>>
- 배정 연산자 : =, +=, -=, \*=, /=, %=, <<=, >>=, >>>=, ^=
- 조건 연산자 : (조건) ? 결과 1 : 결과 2
- 문자열 연산자 : +, +=

```
test.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

<HTML>
  <HEAD>
    <TITLE> 자바스크립트의 연산자 사용 </TITLE>
  </HEAD>
  <BODY>
    <SCRIPT TYPE = "text/javascript" LANGUAGE = "JavaScript">

      var a, b, c, d;    // 변수 a, b, c, d 선언.
      var f = false;    // 변수 선언과 함께 값을 지정함.
      a = 2;            // 변수 a에 정수 2를 할당.
      b = 3.14;         // 변수 b에 부동소수점 수 3.14 할당.
      c = "Result is : "; // 변수 c에 문자열 할당.

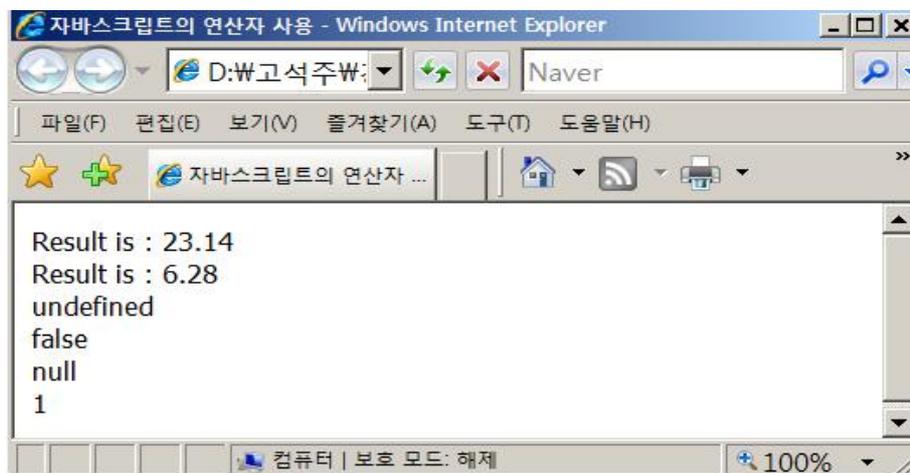
      // a와 b는 문자열로 형 변환됨. 문자열 연산자 '+' 가 사용됨.
      document.write(c + a + b + "<BR>");

      // 연산자의 우선순위에 따른 변화.
      document.write(c + a * b + "<BR>");

      // 선언 후 값이 지정되지 않은 변수 d의 사용.
      document.write(d + "<BR>");

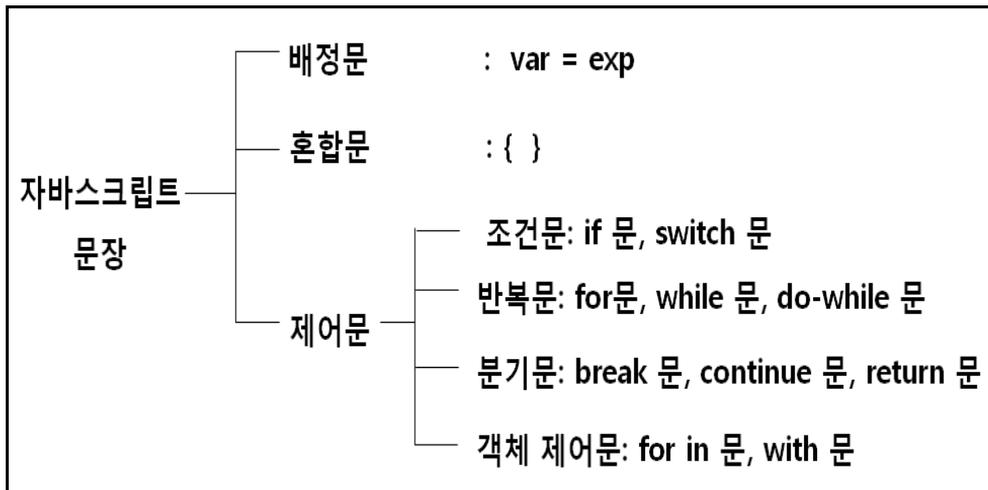
      // 불린 값의 출력. 문자열로 변환되어 출력됨.
      document.write(f + "<BR>");

      // 널 값을 가진 변수 e의 사용.
      var e = null;
      document.write(e + "<BR>");
      document.write(e + 1 + "<BR>");
    </SCRIPT>
  </BODY>
</HTML>
```



## 5.2.5 문장(statement)

문장은 문제해결을 위해 알고리즘을 표현하는 방법이다. 자바 스크립트에서 제공하는 문장의 종류는 다음과 같다.



몇 가지 주요 문장 형식을 살펴보면 다음과 같다.

- The *if* Statement
  - if ( Expression ) Statement*
  - if ( Expression ) Statement else Statement*
- Iteration Statement
  - The *while* Statement
    - while ( Expression ) Statement*
  - The *for* Statement
    - for ( Expression<sub>opt</sub>; Expression<sub>opt</sub>; Expression<sub>opt</sub> ) Statement*
    - for ( var VariableDeclarationList; Expression<sub>opt</sub>; Expression<sub>opt</sub> ) Statement*
  - The *for..in* Statement
    - for ( LeftHandSideExpression in Expression ) Statement*
    - for ( var Identifier Initializer<sub>opt</sub> in Expression ) Statement*

- Jump Statement
  - **continue** ;
  - **break** ;
  - **Return** [no *LineTerminator* here] *Expression*<sub>opt</sub> ;
- The *with* Statement
  - **with** ( *Expression* ) *Statement*

```

with (Math) {
    a = PI * r*r;    // Math.PI
    x = r*cos(theta); // Math.cos(theta)
    y = r*sin(theta); // Math.sin(theta)
}

```

### 5.2.6 함수(function)

- 메소드나 사용자 정의 객체를 정의할 때 이용
- 함수의 형태 :

```

function name(argument list) {
    // body
}

```

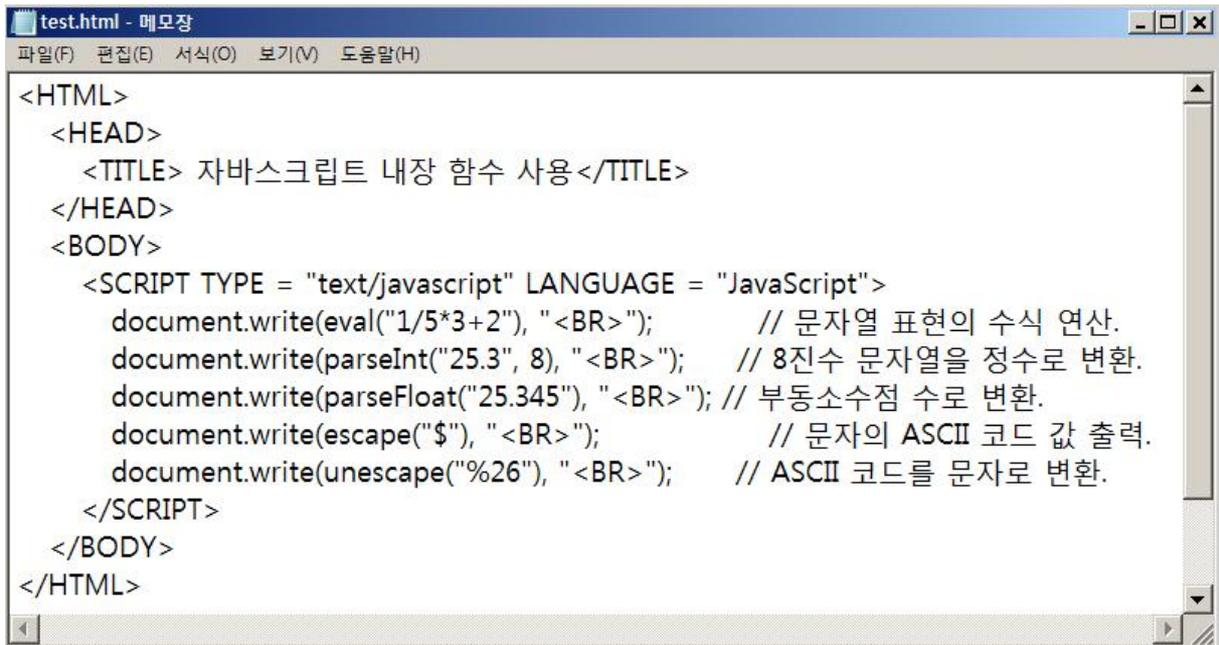
- 예약어 **function**으로 시작
- **return type**을 명시하지 않음
- 매개변수의 형을 기술하지 않음
- 예 :
 

```

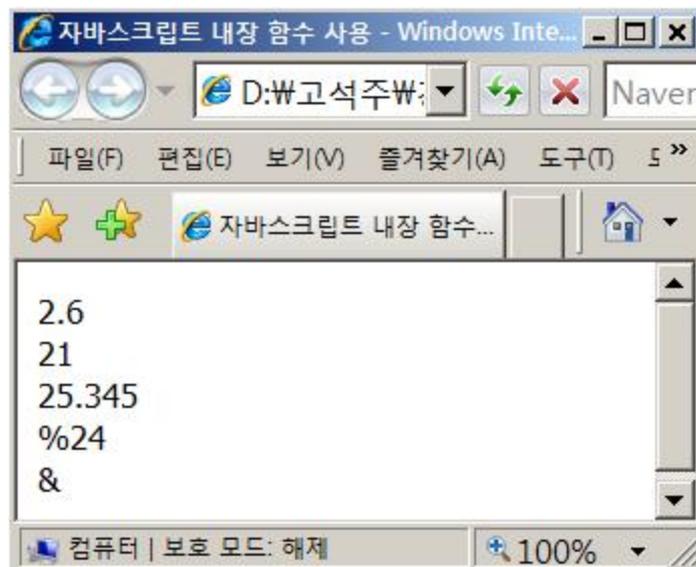
function squareSize(row, col) {
    document.write("Square size : " + row * col);
    return row * col;
}

```

“내장 함수”란 미리 정의되어 있는 global function 를 의미한다. 적절히 활용하면 스크립트 프로그래밍의 효율성이 높아진다. 다음 예제를 통해 내장함수의 종류를 알아보자.



```
<HTML>
  <HEAD>
    <TITLE> 자바스크립트 내장 함수 사용 </TITLE>
  </HEAD>
  <BODY>
    <SCRIPT TYPE = "text/javascript" LANGUAGE = "JavaScript">
      document.write(eval("1/5*3+2"), "<BR>"); // 문자열 표현의 수식 연산.
      document.write(parseInt("25.3", 8), "<BR>"); // 8진수 문자열을 정수로 변환.
      document.write(parseFloat("25.345"), "<BR>"); // 부동소수점 수로 변환.
      document.write(escape("$"), "<BR>"); // 문자의 ASCII 코드 값 출력.
      document.write(unescape("%26"), "<BR>"); // ASCII 코드를 문자로 변환.
    </SCRIPT>
  </BODY>
</HTML>
```



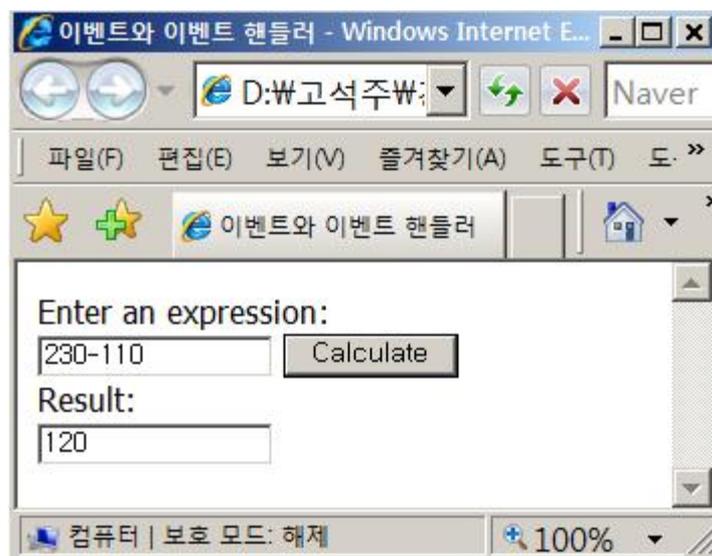
## 5.2.7 이벤트 처리

있는 global function 를 의미한다. 적절히 활용하면 스크립트 프로그래밍의 효율성이 높아진다. 다음 예제를 통해 내장함수의 종류를 알아보자.

- **이벤트(Event)**
  - 사용자에 의한 특정 행위의 결과로 발생.
  - 일반적으로 사용자와 프로그램 사이에 상호작용을 처리
  - 예
    - 하이퍼링크 위에 마우스 커서가 위치하는 경우
    - 사용자의 버튼 클릭
  - 이벤트 드리븐(Event driven) 방식
    - 이벤트 발생에 따라 이벤트 핸들러가 실행되는 방식.
- **이벤트 핸들러(Event Handler)**
  - 이벤트 처리를 위해 정의된 함수나 메소드
  - 기술 형태 :
    - `<INPUT TYPE="button" VALUE="확인" onClick="JavaScriptCode()">`
- **이벤트의 종류(11개)**
  - abort, blur, click, change, error, focus, load, mouseout, mouseover, select, submit
- **이벤트 핸들러**
  - 이벤트 이름 앞에 on을 붙임.
  - 예: onClick
- **의미**
  - 마우스 관련 --- onClick, onMouseout, onMouseover
  - 포커스와 관련 --- onBlur, onFocus
  - 로딩 중 --- onAbort, onError, onLoad
  - 선택과 관련 --- onChange, onSelect
  - 전송과 관련 --- onSubmit

```
test.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

<HTML>
  <HEAD>
    <TITLE>이벤트와 이벤트 핸들러</TITLE>
  </HEAD>
  <BODY>
    <SCRIPT TYPE = "text/javascript" LANGUAGE = "JavaScript">
      function compute(myForm) { // 이벤트 핸들러
        // 내장 함수 eval을 이용하여 계산.
        myForm.result.value = eval(myForm.expr.value);
      }
    </SCRIPT>
    <FORM>
      Enter an expression: <BR>
      <INPUT TYPE="text" NAME="expr" SIZE=15 >
      <INPUT TYPE="button" VALUE="Calculate" onClick="compute(this.form)"> <BR>
      Result: <BR>
      <INPUT TYPE="text" NAME="result" SIZE=15 >
    </FORM>
  </BODY>
</HTML>
```



## 5.3 자바 스크립트 객체

### 5.3.1 사용자 정의 객체

#### ■ 객체

::= 속성 + 메소드

#### ■ 속성(property)

- 객체의 상태를 나타내는 변수들의 선언 집합

#### ■ 메소드(method)

- 객체의 행위를 정의하는 함수의 형태
- 객체와 연관된 함수를 의미

```
function Book(title, author, pages) {  
    this.title = title;  
    this.author = author;  
    this.pages = pages;  
}
```

```
myBook = new Book("JavaScript", "John" 255);  
myBook.translator = "Jane";
```

- `this` --- 객체 자신을 가리킴
- `new` --- 인스턴스 생성 연산자
- `dot notation` --- 객체와 속성 사이에

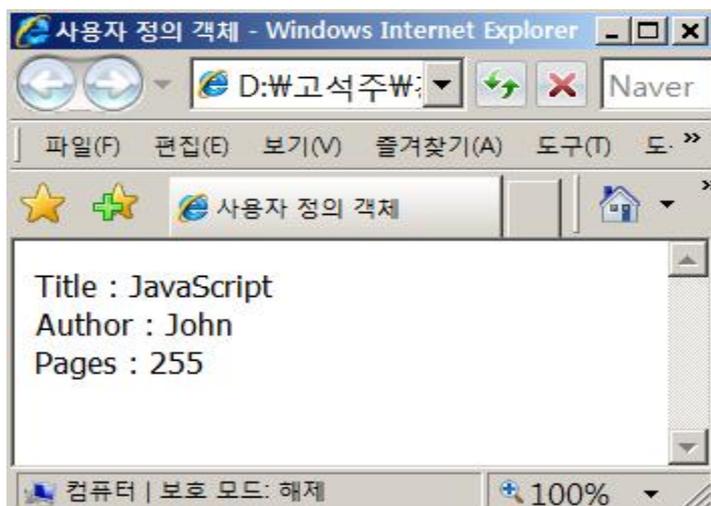
다음 예제를 살펴보자.

```
test.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

<HTML>
<HEAD>
  <TITLE> 사용자 정의 객체 </TITLE>
  <SCRIPT TYPE = "text/javascript" LANGUAGE = "JavaScript">

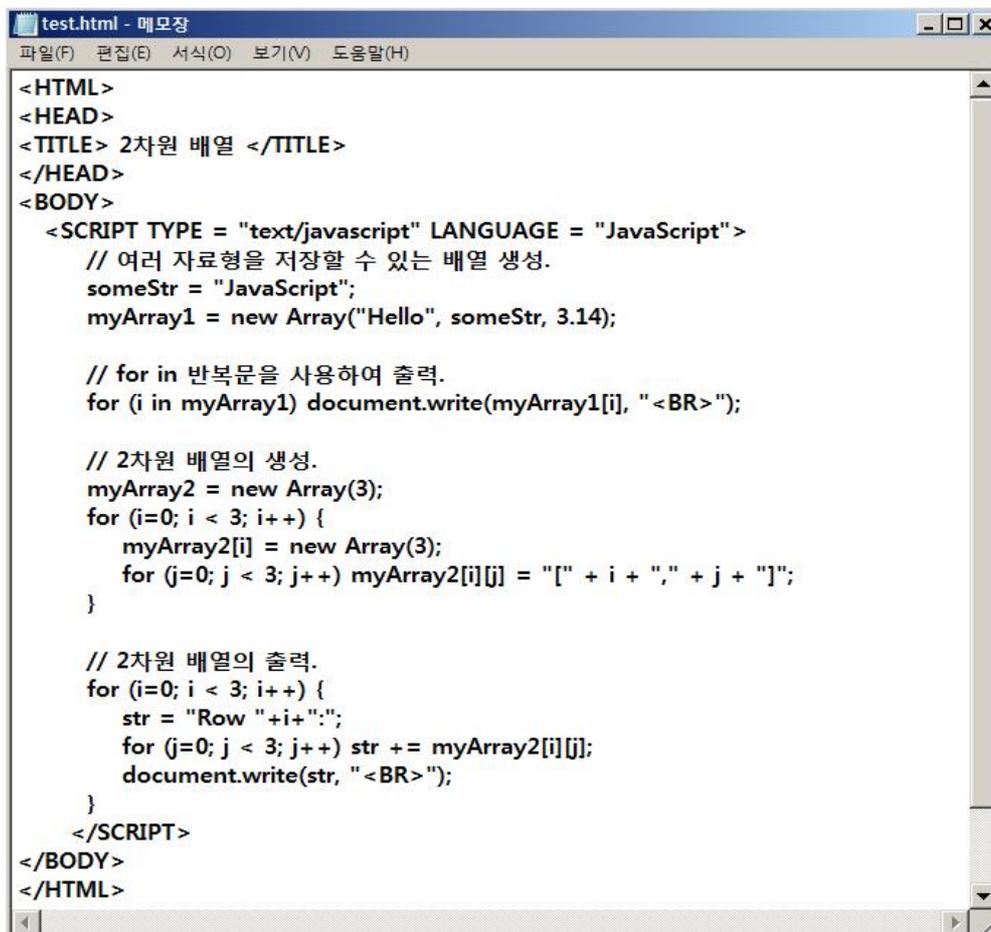
  // 메소드 정의
  function displayInfo() {
    document.write("Title : " + this.title + "<BR>");
    document.write("Author : " + this.author + "<BR>");
    document.write("Pages : " + this.pages + "<BR>");
  }

  //객체 정의
  function Book(title, author, pages) {
    this.title = title;           // 속성 정의
    this.author = author;
    this.pages = pages;
    this.displayInfo = displayInfo; // 메소드 추가
  }
</SCRIPT>
</HEAD>
<BODY>
  <SCRIPT TYPE = "text/javascript" LANGUAGE = "JavaScript">
    myBook = new Book("JavaScript", "John", 255); // 객체 인스턴스 생성
    myBook.displayInfo();                       // 객체의 메소드 호출
  </SCRIPT>
</BODY>
</HTML>
```



## 5.3.2 내장 객체

- 내장 객체
  - 객체를 정의하지 않고 사용할 수 있는 자바스크립트의 객체
    - 자바스크립트에 미리 정의되어 있는 객체
    - Java에서 미리 정의되어진 클래스와 유사
  - 사용 빈도가 높은 많은 메소드들을 포함
  - 내장 객체의 종류
    - Array, Boolean, Date, Function, Math, Number, Object, RegExp, String
- 내장 객체의 사용
  - new 연산자에 의해 내장 객체를 사용하여 인스턴스를 생성
    - someString = "JavaScript";
    - myArray = **new** Array("hello", someString, 3.14);

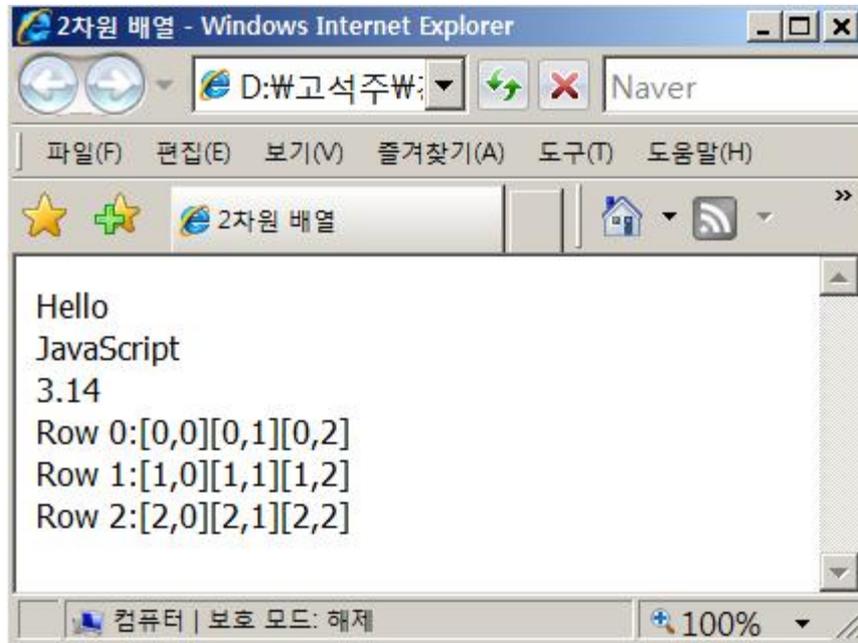


```
<HTML>
<HEAD>
<TITLE> 2차원 배열 </TITLE>
</HEAD>
<BODY>
  <SCRIPT TYPE = "text/javascript" LANGUAGE = "JavaScript">
    // 여러 자료형을 저장할 수 있는 배열 생성.
    someStr = "JavaScript";
    myArray1 = new Array("Hello", someStr, 3.14);

    // for in 반복문을 사용하여 출력.
    for (i in myArray1) document.write(myArray1[i], "<BR>");

    // 2차원 배열의 생성.
    myArray2 = new Array(3);
    for (i=0; i < 3; i++) {
      myArray2[i] = new Array(3);
      for (j=0; j < 3; j++) myArray2[i][j] = "[" + i + "," + j + "]";
    }

    // 2차원 배열의 출력.
    for (i=0; i < 3; i++) {
      str = "Row " + i + " : ";
      for (j=0; j < 3; j++) str += myArray2[i][j];
      document.write(str, "<BR>");
    }
  </SCRIPT>
</BODY>
</HTML>
```



다른 내장객체의 사용 예를 살펴보자.

■ Boolean 객체

- // Boolean 객체를 사용해 참/거짓의 할당.

```
isTrue = new Boolean(false);  
document.write(isTrue + "<BR>");
```

■ Date 객체

- // Date 객체의 getYear와 getMonth 메소드의 사용.

```
today = new Date("Oct 28, 1999 12:00:00");  
document.write(today.getYear(), ",", today.getMonth(), "<BR>");
```

■ Function 객체

- // Function 객체를 사용해 새로운 함수의 생성.

```
var println = new Function("str", "document.write(str + '<BR>')");  
println("This is from println function.");
```

## ■ Math 객체

- // Math 객체의 수학 함수.

```
with (Math) {
    document.write("abs(-3.14) is " + abs(-3.14) + "<BR>");
    document.write("sin(0.5) is " + sin(0.5) + "<BR>");
    document.write("Random number is " + random() + "<BR>");
}
```

## ■ Number 객체

- // Number 객체를 사용해 숫자에 설명을 추가.

```
myNum = new Number(3342);
myNum.description="Phone number : ";
document.write(myNum.description + myNum + "<BR>");
```

## ■ Object 객체

```
document.write("(Decimal : Binary) ==> ");
for (x = 0; x < 5; x++) {
    document.write("(", x.toString(10), " : ", x.toString(2), ")");
    if (x!=4) document.write(", ");
}
document.write("<BR>");
```

## ■ RegExp 객체

- // RegExp 객체를 이용해 특정 문자 패턴을 검색.

```
document.write(/ab+c/.exec("cabbcbzbz") + " is same as ");
myRe= new RegExp (" ab+c");
token = myRe.exec("cabbcbzbz");
document.write(token + "<BR>");
```

### 5.3.3 표준 객체

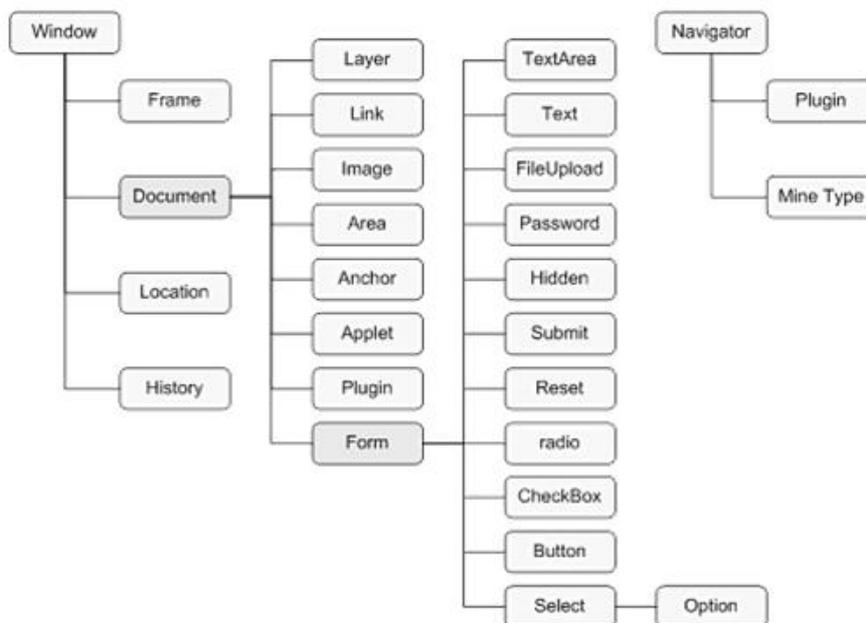
#### ■ 표준 객체

- HTML 문서가 브라우저에 로드되면 브라우저는 HTML 문서의 <FORM> 태그나 <A>(anchor) 태그 등 관련된 정보에 기초하여 많은 자바스크립트 객체를 생성한다. 이를 표준 객체 또는 네비게이터 객체라 부른다.
- 네비게이터를 동적으로 제어할 수 있다.
- 계층적 구조를 갖고 있다.

#### ■ 계층적 구조

- 하위 객체는 상위 객체의 속성으로 정의되어 있다.
- 예를 들어, form1이라는 이름의 객체는 document 객체의 속성인 form 객체의 인스턴스이며, 자바스크립트에서 document.form1 형태로 참조할 수 있다.
- 네비게이터 객체의 모든 하위 객체는 많은 속성과 메소드 그리고 이벤트 핸들러를 갖고 있다.

## 계층적 구조



```

test.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

<HTML>
<HEAD>
<TITLE> 표준 객체의 사용 </TITLE>
<SCRIPT TYPE = "text/javascript" Language = "JavaScript">
var index=0; // 배열의 인덱스를 저장하기 위한 전역변수
function addOption(myForm) { // Add 버튼을 클릭했을 때
var val = myForm.file_name.value;
if(val.length!=0) {
myForm.mySelect.options[index] =
new Option(val, "add option", true);
index++;
}
}
function delOption(myForm) { // Delete 버튼을 클릭했을 때
if(index!=0) index--;
myForm.mySelect.options[index] = null;
}
</SCRIPT>
</HEAD>
<BODY>
<FORM NAME="myForm">
<CENTER>
<SELECT NAME="mySelect" SIZE=3>
<OPTION>List of Files Selected from File Browsing...
</SELECT><BR><BR>
<INPUT TYPE="button" VALUE="Add" onClick="addOption(this.form)">
<INPUT TYPE="button" VALUE="Delete"
onClick="delOption(this.form)"> <BR><BR>
File : <INPUT TYPE="file" NAME="file_name">
</CENTER>
</FORM>
</BODY>
</HTML>

```



## 5.4 동적인 웹 문서 만들기

이제 기존의 정적인 문서에서 벗어나, 자바스크립트를 기반으로 사용자의 요구에 호응하는 동적인(dynamic) 문서를 제작하는 방법을 알아보자.

### ◇ 프레임(frame)

- 웹 문서를 효과적으로 배치하여 많은 내용을 일관성 있게 보여 준다.

### ◇ 레이어(layer)

- 문서의 특정 영역을 지정하여 별도의 내용(contents)을 표현할 수 있도록 하며, 겹치거나 투명하게 나타낼 수 있다.

### ◇ 쿠키(cookie)

- 웹 페이지의 현재 상태를 클라이언트의 컴퓨터에 저장하여 관리할 수 있다. 즉 쿠키를 이용해 사용자의 정보를 저장하고 다음 방문 시 참조하여 활용할 수 있다.

### ◇ 자바 애플릿 코드와 연동

- 자바애플릿과 자바스크립트 코드는 서로 독립적으로 수행되지만 서로 접근이 가능하다.

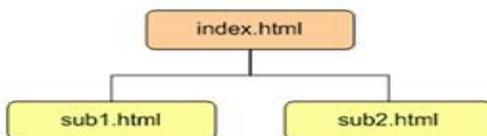
	애플릿	자바스크립트
언어부류	객체지향언어	객체지향언어를 추구
실행 방법	컴파일 방법	인터프리티브 방법
자료형	강하다 (strongly typed language)	약하다 (weakly typed language)
브라우저	독립적으로 실행	브라우저객체활용
HTML연계	독립적	종속적

### 5.4.1 프레임(frame) 객체

- HTML의 <FRAME> 태그에 의해 생성된 각 프레임은 **Frame** 객체를 참조해 통신 및 구조를 변경할 수 있다.
- **Frame** 객체
  - window 객체의 하위 객체. 따라서, window 객체에서 제공된 모든 기능을 사용할 수 있다.
  - **속성** : (추가 속성)
    - frames : 프레임 안에 있는 모든 프레임에 관하여 배열로 제공
    - name : <FRAME> 태그내의 NAME 속성에 지정된 프레임의 이름
    - parent : 부모 프레임을 나타냄.
    - self : 프레임 자신을 나타낼 때 사용
    - length : 프레임 안에 포함된 프레임의 수를 지정
- Frame 객체에서 제공하는 메소드 :
  - setTimeout() : 지정된 시간 후에 실행할 때 사용
  - clearTimeout() : setTimeout() 메소드로 설정된 시간을 해제
- Frame 객체에서 제공하는 이벤트 핸들러 :
  - onFocus: 프레임이 초점을 얻음으로써 활성화될 때 발생
  - onBlur : 프레임이 초점을 잃음으로써 비활성화될 때 발생

## 프레임 예제

- 프레임 구조



- 기능:
  - 왼쪽 프레임인 sub1.html에서 입력한 데이터를 오른쪽 프레임인 sub2.html에서 참조하는 방법.
  - 이때, parent.frames[0]로 참조가 가능
    - parent.frames[0] ==> sub1.html ==> top.frames[0]
    - parent.frames[1] ==> sub2.html ==> top.frames[1]

```
index.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

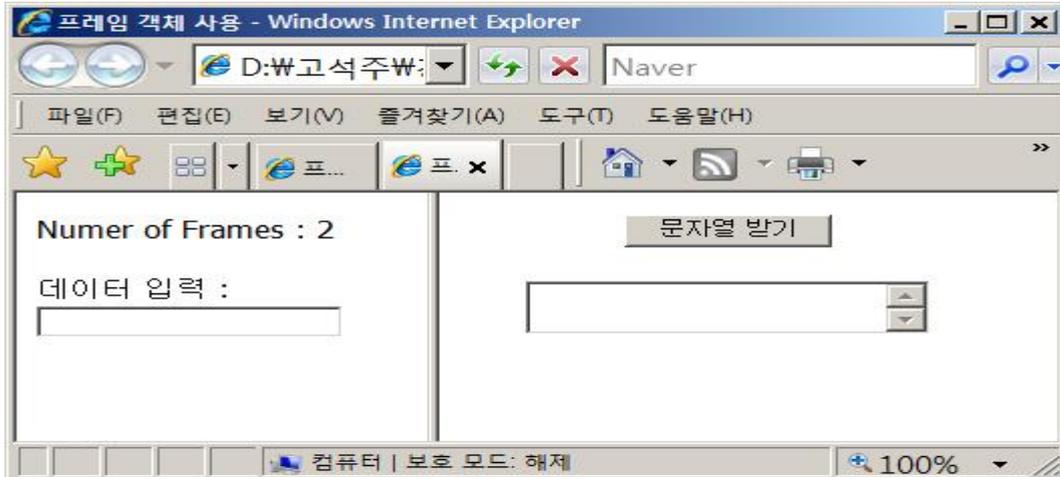
<HTML>
  <HEAD>
    <TITLE>프레임 객체 사용</TITLE>
  </HEAD>
  <FRAMESET COLS="40%, 60%">
    <FRAME SRC="sub1.html">
    <FRAME SRC="sub2.html">
  </FRAMESET>
</HTML>
```

```
sub1.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

<HTML>
  <HEAD>
    <SCRIPT LANGUAGE = "JavaScript">
      document.write("Numer of Frames : " + top.length + "<BR>"); // 프레임 수
    </SCRIPT>
  </HEAD>
  <BODY>
    <FORM NAME="left">
      데이터 입력 :
      <INPUT TYPE="text" NAME="inputLn">
    </FORM>
  </BODY>
</HTML>
```

```
sub2.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

<HTML>
  <HEAD>
    <TITLE>프레임 객체를 통하여 동적으로 변경</TITLE>
    <SCRIPT LANGUAGE = "JavaScript">
      function disp() {
        var str = parent.frames[0].document.left.inputLn.value
        document.right.output.value = str
      }
    </SCRIPT>
  </HEAD>
  <BODY>
    <CENTER>
    <FORM NAME="right">
      <INPUT TYPE="button" VALUE="문자열 받기" onClick="disp()">
      <BR><BR>
      <TEXTAREA NAME="output" ROWS=2 COLS=25 onFocus="disp()">
    </TEXTAREA>
    </FORM>
  </BODY>
</HTML>
```



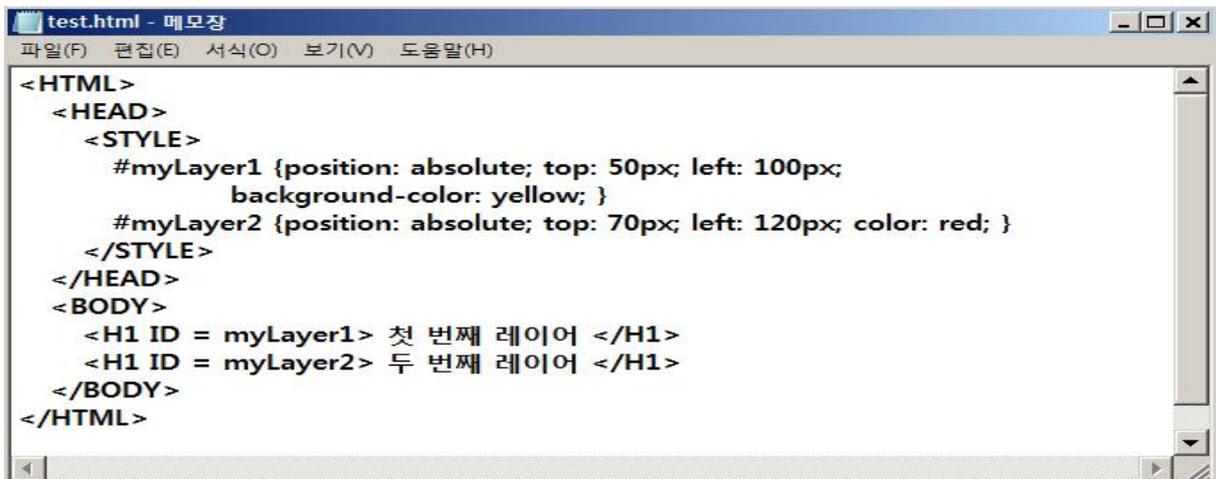
#### 5.4.2 레이어(layer) 객체

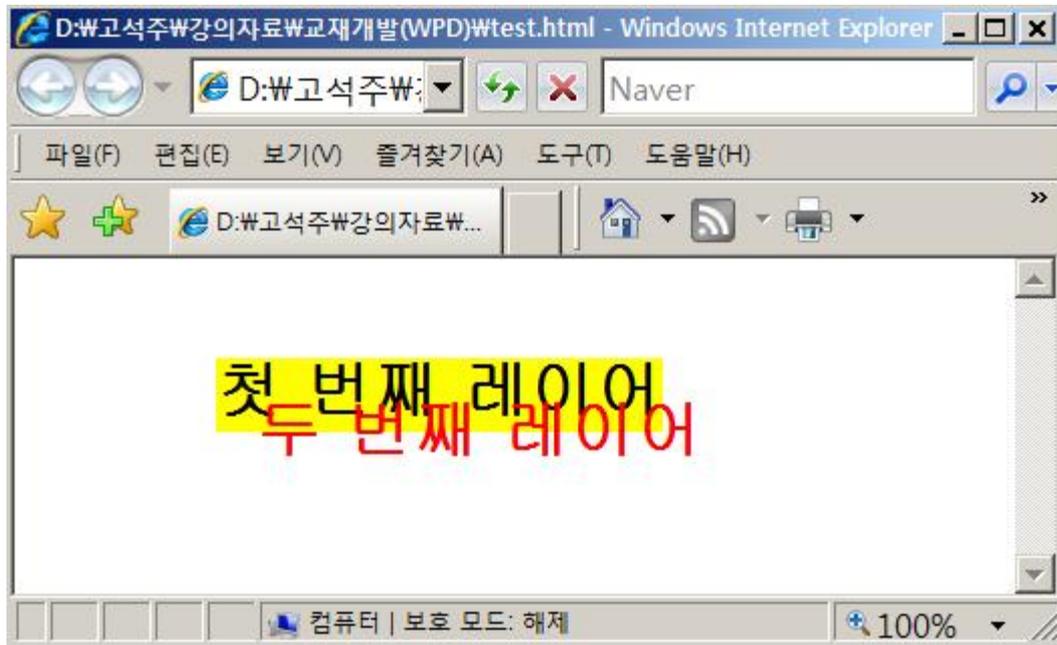
- HTML 문서에서 특정 영역을 지정하여 별도의 내용을 보여줄 수 있는 기능.
- 그래픽 프로그램처럼 겹치거나, 투명하게 처리할 수 있으며, 자바스크립트를 이용함으로써 동적으로 제어할 수 있다.
- 레이어는 스타일시트를 이용하거나 또는 <LAYER> 태그를 이용해 작성할 수 있다.
- 스타일시트를 이용한 레이어

```

<STYLE>
    #LayerNameId { 속성1: 값1; 속성2: 값2; ... }
</STYLE>

```





## 레이어 - Layer 객체

- Layer 객체를 이용하는 방법
  - `document.layers[index].propertyName`
  - `document.layers[index].methodName`
- 속성
  - `name`, `left`, `top`, `height`, `clip.top`, `clip.left`, `visibility`
- 메소드
  - `moveAbove()`, `moveBy()`, `resizeBy()`
  - 레이어의 위치 및 크기를 변경

### 5.4.3 쿠키(cookie)

#### ■ 쿠키

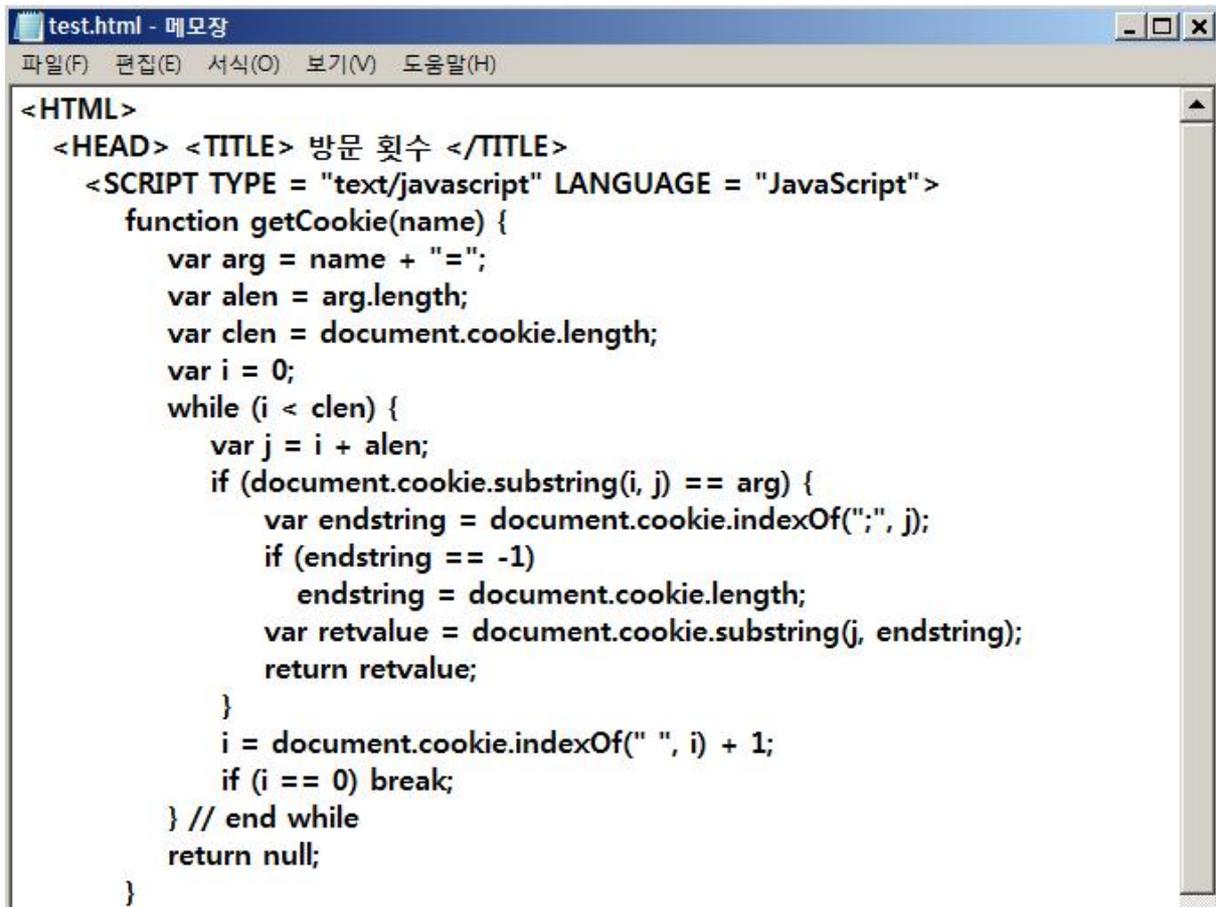
- 클라이언트에 대한 정보를 클라이언트에 저장.
- 나중에 서버에서 활용.
- HTTP에 기반 - 헤더 정보를 이용

#### ■ 저장

- `document.cookie = "name = Kildong Hong";`
- `document.cookie = "password = 1234";`

#### ■ 사용

- 스트링으로 다루어 정보를 추출함.

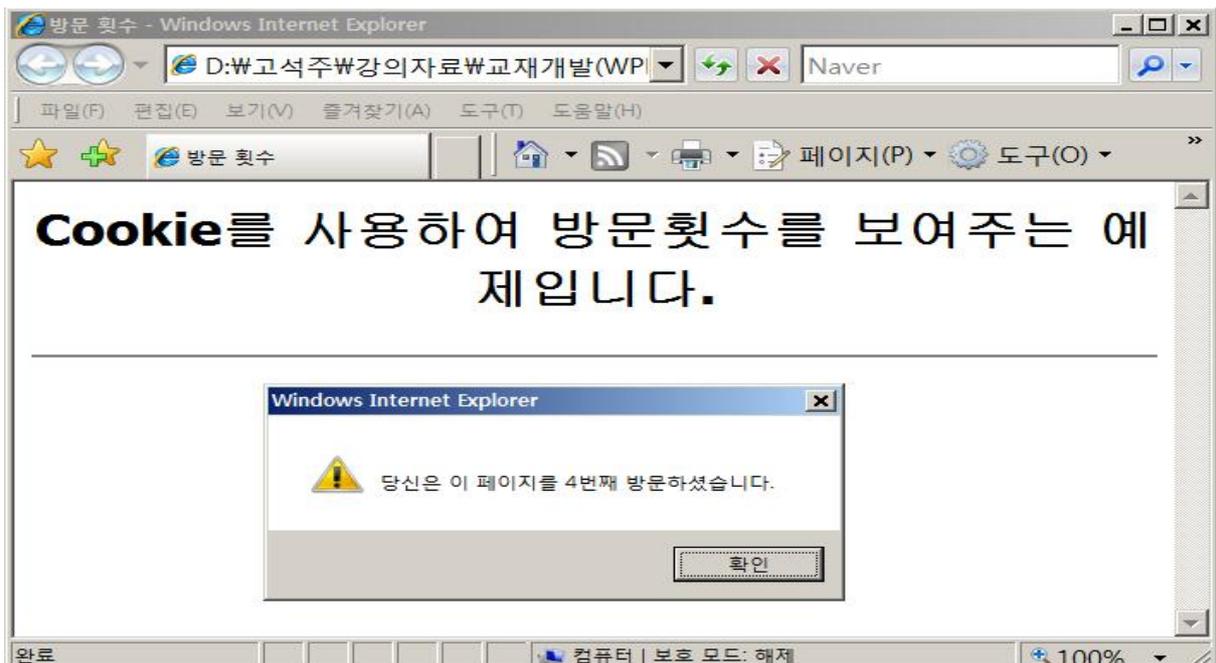


```
test.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<HTML>
<HEAD> <TITLE> 방문 횟수 </TITLE>
<SCRIPT TYPE = "text/javascript" LANGUAGE = "JavaScript">
function getCookie(name) {
    var arg = name + "=";
    var alen = arg.length;
    var clen = document.cookie.length;
    var i = 0;
    while (i < clen) {
        var j = i + alen;
        if (document.cookie.substring(i, j) == arg) {
            var endstring = document.cookie.indexOf(";", j);
            if (endstring == -1)
                endstring = document.cookie.length;
            var retvalue = document.cookie.substring(j, endstring);
            return retvalue;
        }
        i = document.cookie.indexOf(" ", i) + 1;
        if (i == 0) break;
    } // end while
    return null;
}
```

```

function setCookie (name, value, expires) {
    document.cookie = name + "=" + escape (value) +
        ((expires == null) ? "" : (";expires=" + expires.toGMTString()));
}
function display() {
    var visit;
    var expdate = new Date();
    expdate.setTime(expdate.getTime() + (365*24*60*60*1000));
    if (!(visit = getCookie("xxxx"))) visit = 0;
    visit++;
    setCookie("xxxx", visit, expdate);
    alert("\n"+"당신은 이 페이지를 "+visit+"번째 방문하셨습니다.\n");
}
</SCRIPT>
</HEAD>
<BODY onLoad="display()">
    <CENTER><H1> Cookie를 사용하여 방문횟수를 보여주는 예제입니다. </H1>
    <HR NOSHADE>
</CENTER>
</BODY>
</HTML>

```

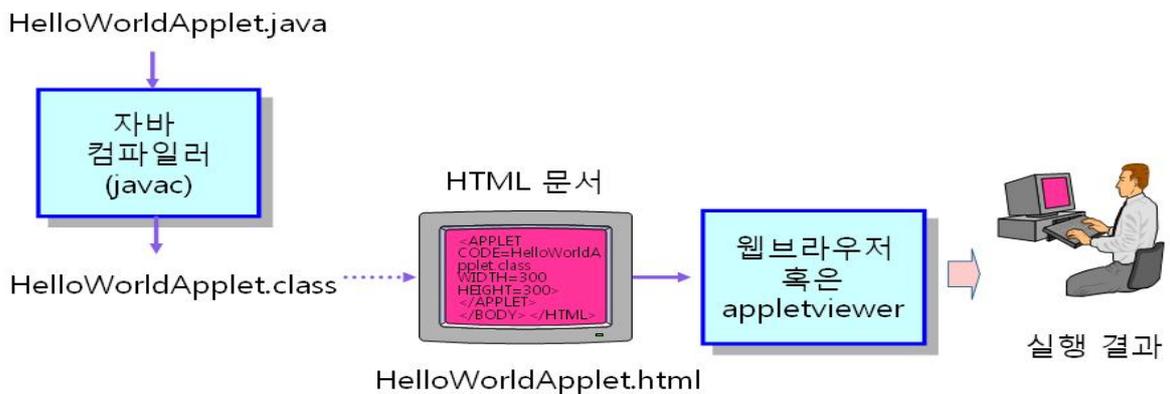


#### 5.4.4 자바 애플릿(applet) 코드와의 연동

다음 예제는 애플릿이 수행되면서 처음 화면에 "Hello World!"를 출력하고, 자바스크립트를 사용하여 "감추기" 버튼과 "보여주기" 버튼으로써 화면에 효과를 나타내 준다.

여기서, \*.java 파일은 "자바 컴파일러(java)"를 통해 컴파일되어야 한다. 이를 위해 "자바 컴파일러"가 컴퓨터에 설치되어 있어야 한다 (예: JDK 툴킷 등). 자바스크립트와 자바 애플릿과의 연동 방식을 개념적으로 정리하면 다음과 같다. 보다 자세한 자바애플릿 관련 내용은 별도의 교재 혹은 강의자료를 참조하길 바란다.

## 자바 애플릿 수행과정



```
>HelloWorldApplet.java - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

import java.awt.Graphics;

public class HelloWorldApplet extends java.applet.Applet {
    public void paint(Graphics g) {
        g.drawString("Hello World!", 5, 25);
    }
}
```

```
>HelloWorldApplet.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

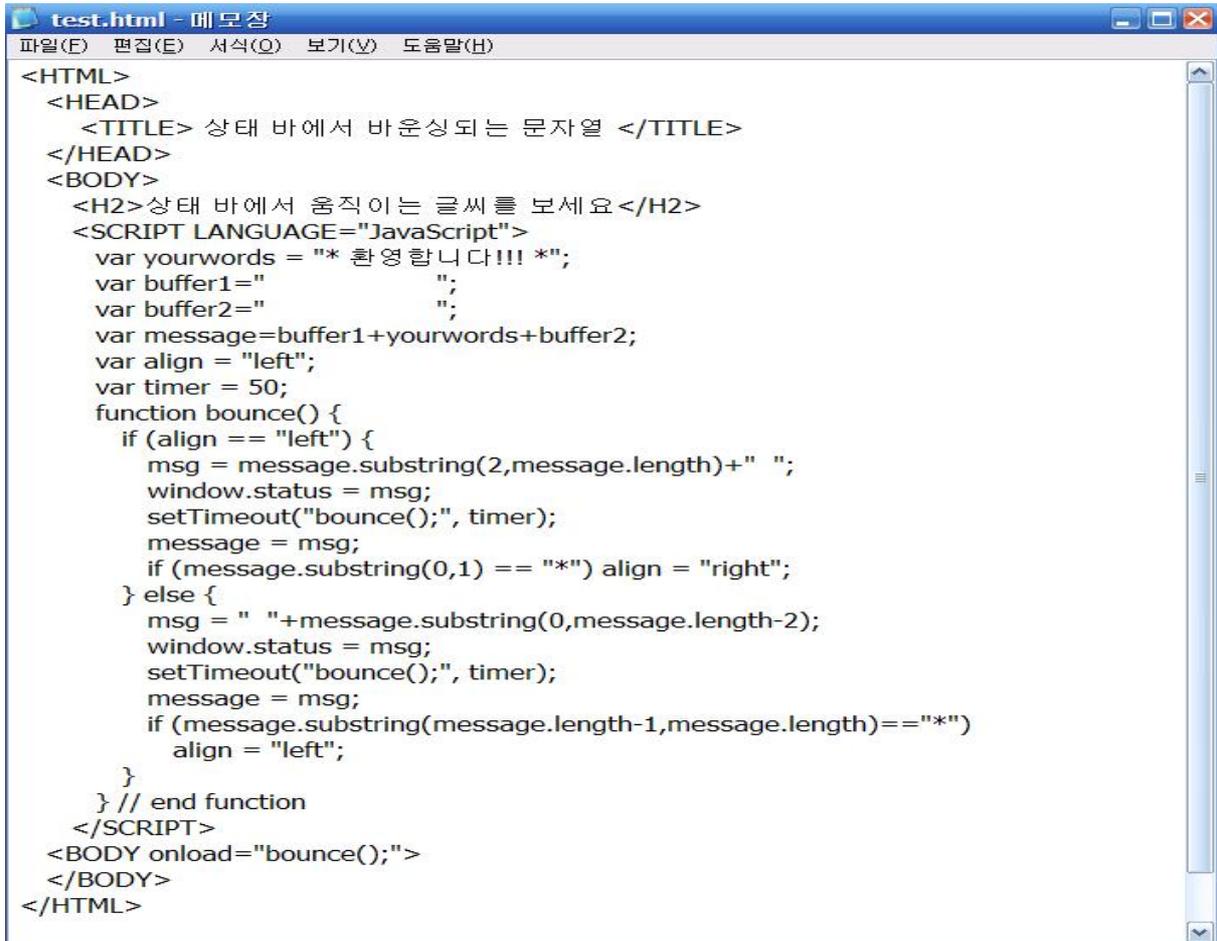
<HTML>
  <HEAD>
    <TITLE> Hello </TITLE>
  </HEAD>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      function hideHello() { // hide() 메소드를 이용한 감추기 기능
        document.Hello.hide();
      }
      function showHello() { // show() 메소드를 이용한 보여주기 기능
        document.Hello.show();
      }
    </SCRIPT>
  </HEAD>
  <BODY>
    <APPLET CODE="HelloWorldApplet.class" NAME="Hello" WIDTH=300 HEIGHT=
  </APPLET>
    <FORM>
      <INPUT TYPE="button" VALUE="감추기" onClick="hideHello()">
      <INPUT TYPE="button" VALUE="보여주기" onClick="showHello()">
    </FORM>
  </BODY>
</HTML>
```



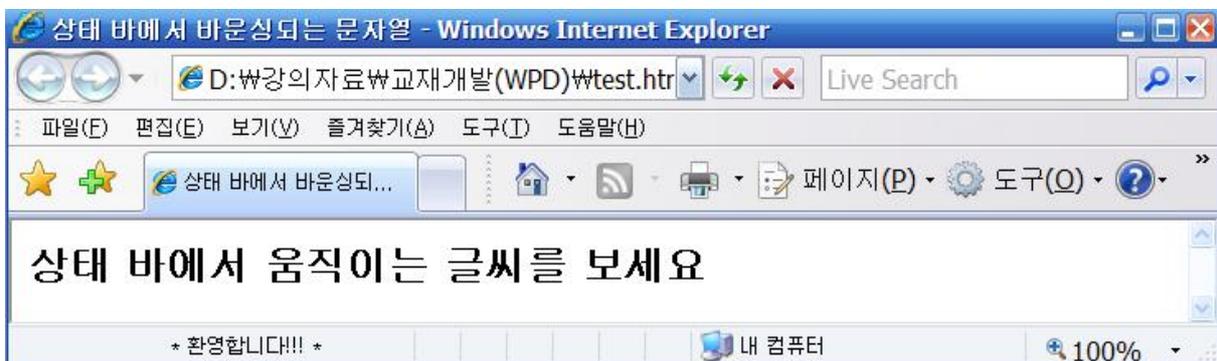
## 5.5 자바스크립트 활용

### 5.5.1 상태 바에서 움직이는 글자

- 상태 바에 글자가 왼쪽과 오른쪽을 왕복 하도록 하는 예제
- 시간을 주기로 글자가 바운싱 되는 것처럼 문자열을 만들어 상태 바에 출력한다.

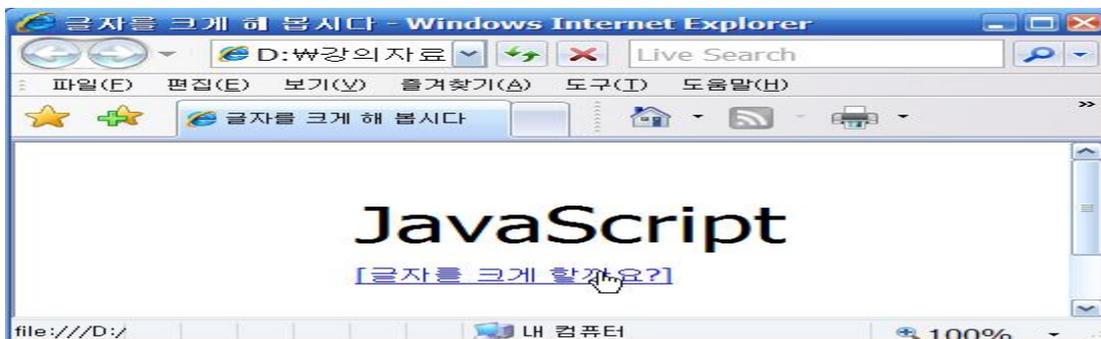
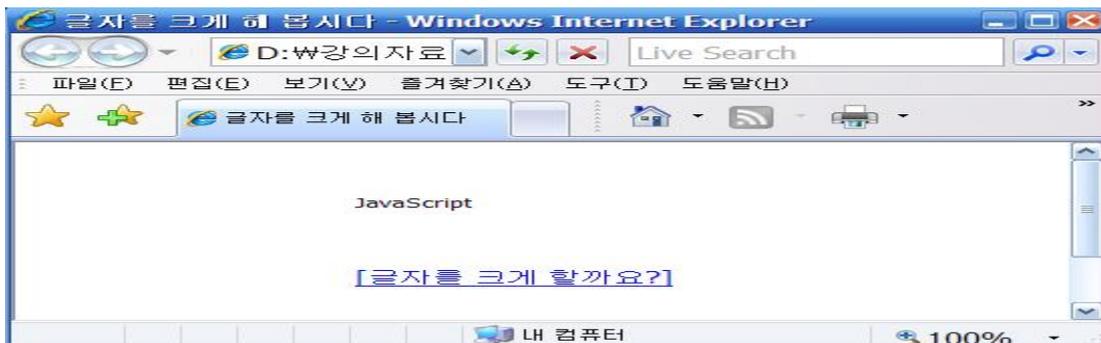


```
<HTML>
<HEAD>
  <TITLE> 상태 바에서 바운싱되는 문자열 </TITLE>
</HEAD>
<BODY>
  <H2>상태 바에서 움직이는 글씨를 보세요</H2>
  <SCRIPT LANGUAGE="JavaScript">
    var yourwords = "* 환영합니다!!! *";
    var buffer1="";
    var buffer2="";
    var message=buffer1+yourwords+buffer2;
    var align = "left";
    var timer = 50;
    function bounce() {
      if (align == "left") {
        msg = message.substring(2,message.length)+" ";
        window.status = msg;
        setTimeout("bounce();", timer);
        message = msg;
        if (message.substring(0,1) == "*") align = "right";
      } else {
        msg = " "+message.substring(0,message.length-2);
        window.status = msg;
        setTimeout("bounce();", timer);
        message = msg;
        if (message.substring(message.length-1,message.length)=="*")
          align = "left";
      }
    }
  } // end function
</SCRIPT>
<BODY onload="bounce();">
</BODY>
</HTML>
```

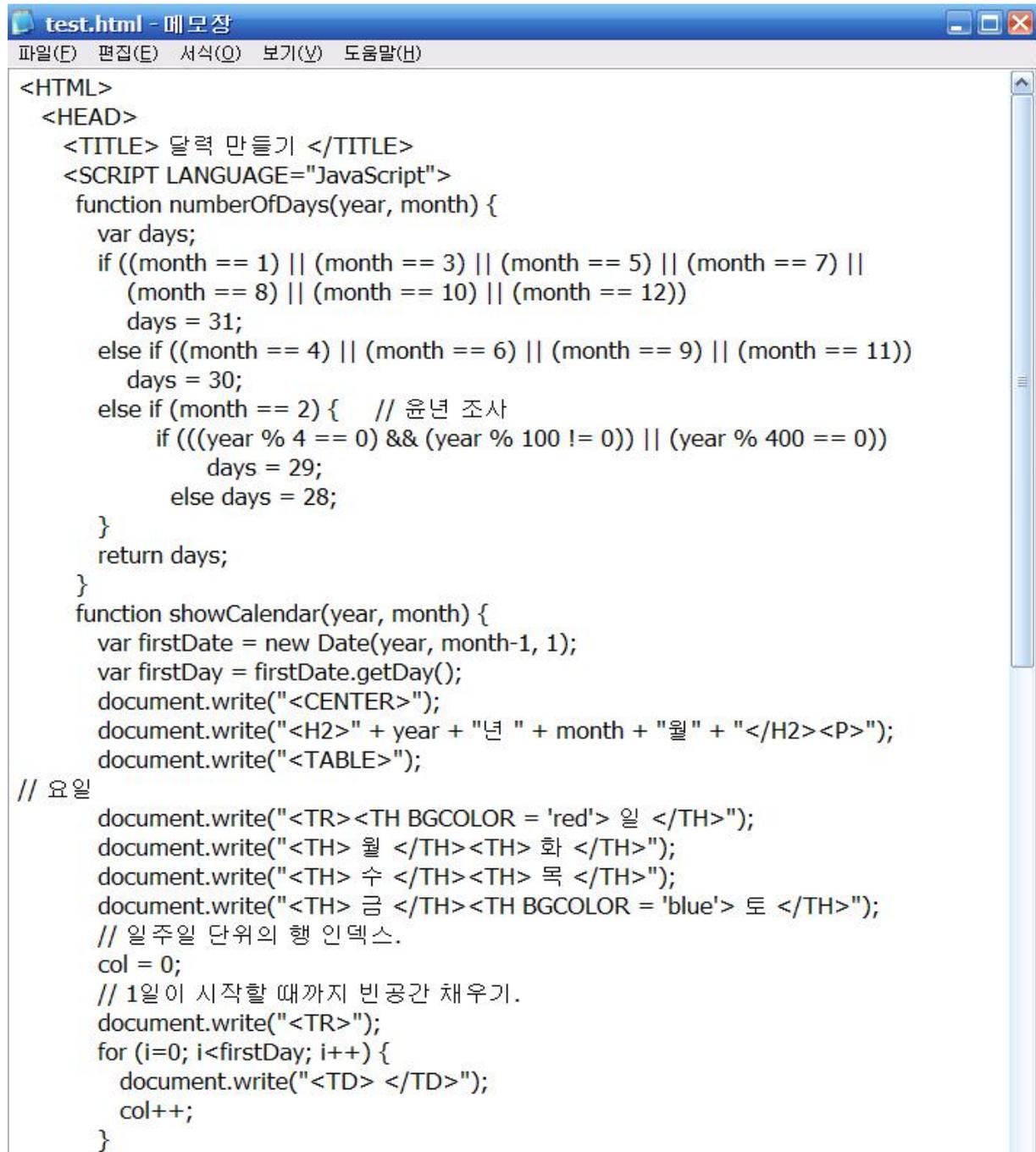


## 5.5.2 마우스를 올리면 커지는 글자

```
test.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<HTML>
  <HEAD>
    <TITLE> 글자를 크게 해 봅시다 </TITLE>
    <SCRIPT LANGUAGE="JavaScript">
      function changeBig(text){
        scrn.innerHTML = text.fontSize(+7);
      }
      function changeSmall(text){
        scrn.innerHTML = text.fontSize(2);
      }
    </SCRIPT>
  </HEAD>
  <BODY>
    <DIV ID="scrn" STYLE="position:absolute; left:169px; top:50px;
      width:337px; height:127px; z-index:1">
      <FONT SIZE=2>JavaScript</FONT>
    </DIV>
    <BR><BR><BR>
    <DIV ID="btn" STYLE="position:absolute; left:169px; top:120px;
      width:337px; height:127px; z-index:1">
      <A HREF="#1" onMouseOver="changeBig(scrn.innerHTML)"
        onMouseOut="changeSmall(scrn.innerHTML)">
        [글자를 크게 할까요?]
      </A>
    </DIV>
  </BODY>
</HTML>
```



### 5.5.3 달력 만들기



The image shows a web browser window titled "test.html - 메모장". The browser's address bar and menu bar are visible. The main content area displays the following JavaScript code:

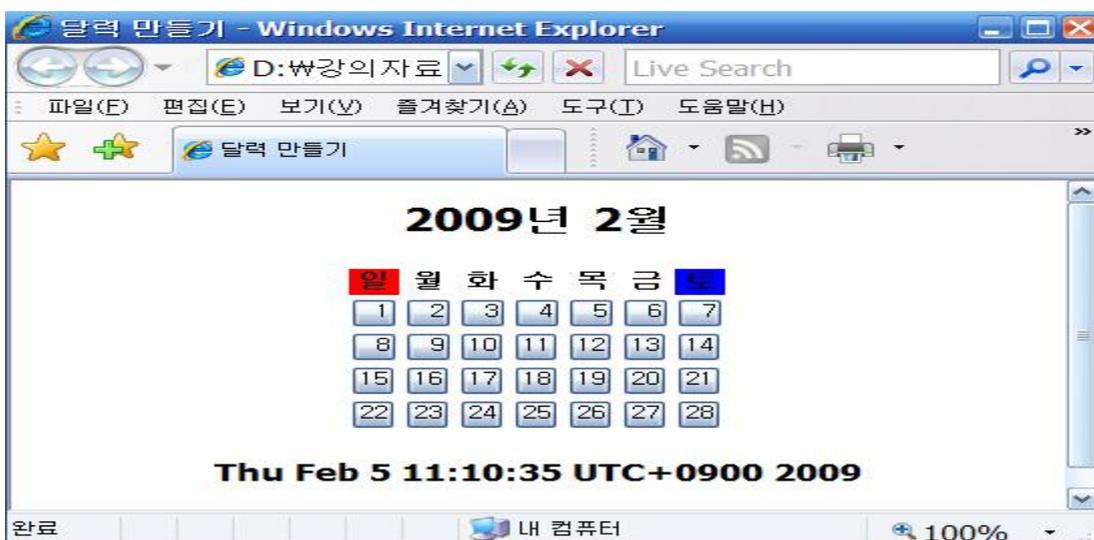
```
<HTML>
<HEAD>
<TITLE> 달력 만들기 </TITLE>
<SCRIPT LANGUAGE="JavaScript">
function numberOfDays(year, month) {
    var days;
    if ((month == 1) || (month == 3) || (month == 5) || (month == 7) ||
        (month == 8) || (month == 10) || (month == 12))
        days = 31;
    else if ((month == 4) || (month == 6) || (month == 9) || (month == 11))
        days = 30;
    else if (month == 2) { // 윤년 조사
        if (((year % 4 == 0) && (year % 100 != 0)) || (year % 400 == 0))
            days = 29;
        else days = 28;
    }
    return days;
}
function showCalendar(year, month) {
    var firstDate = new Date(year, month-1, 1);
    var firstDay = firstDate.getDay();
    document.write("<CENTER>");
    document.write("<H2>" + year + "년 " + month + "월 " + "</H2><P>");
    document.write("<TABLE>");
// 요일
    document.write("<TR><TH BGCOLOR = 'red'> 일 </TH>");
    document.write("<TH> 월 </TH><TH> 화 </TH>");
    document.write("<TH> 수 </TH><TH> 목 </TH>");
    document.write("<TH> 금 </TH><TH BGCOLOR = 'blue'> 토 </TH>");
// 일주일 단위의 행 인덱스.
    col = 0;
// 1일이 시작할 때까지 빈공간 채우기.
    document.write("<TR>");
    for (i=0; i<firstDay; i++) {
        document.write("<TD> </TD>");
        col++;
    }
}
```

```

test.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

document.write("<FORM>");
for (i=1; i<=numberOfDays(year, month); i++) { // 날짜 표시
  // 버튼을 이용.
  document.write("<TD>");
  if (i < 10)
    document.write("<INPUT TYPE='button' VALUE=" + " " + i + ">");
  else
    document.write("<INPUT TYPE='button' VALUE=" + i + ">");
  document.write("</TD>");
  col++;
  if (col == 7) { // 새로운 행 지정.
    document.write("</TR>");
    col = 0;
  }
} // end for
document.write("</FORM>");
document.write("</TABLE>");
document.write("</CENTER>");
} // end showCalendar
</SCRIPT>
</HEAD>
<BODY>
  <CENTER>
    <SCRIPT LANGUAGE="JavaScript">
      var today = new Date();
      showCalendar(today.getYear(), today.getMonth() + 1);
      document.write("<H3>" + today.toString() + "</H3><P>");
    </SCRIPT>
  </CENTER>
</BODY>
</HTML>

```



## 제 6 장. ASP

먼저, ASP의 개요를 정리하면 다음과 같다.

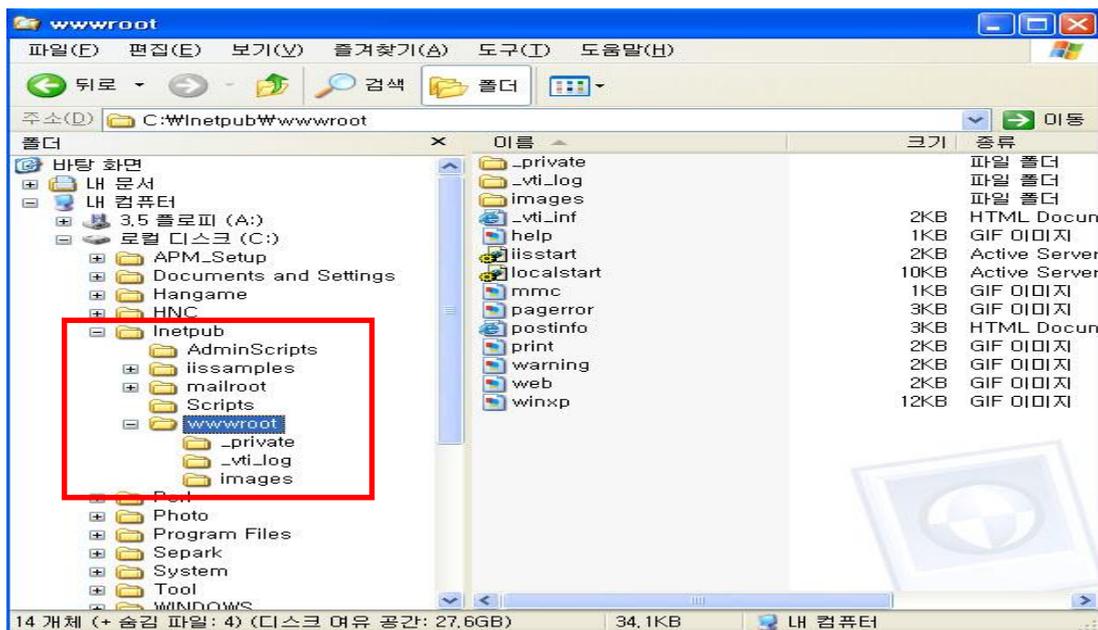
- Active Server Page의 약자
- Microsoft사에서 개발한 IIS(Internet Information Server)의 세 번째 버전부터 지원
- 기존 CGI의 단점들을 보완하고 다양한 기능을 추가한 윈도우 서버 기반의 시스템
- Asp.dll을 이용하여 스레드로 클라이언트 요청을 처리함으로써 서버의 부하를 줄임

### 6.1 윈도우즈 웹 서버

#### 6.1.1 IIS를 이용한 웹 서버 구축

먼저, 윈도우 환경에서 웹서버 구축 절차를 알아보자. (여러분의 시스템/환경마다 다를 수 있음)

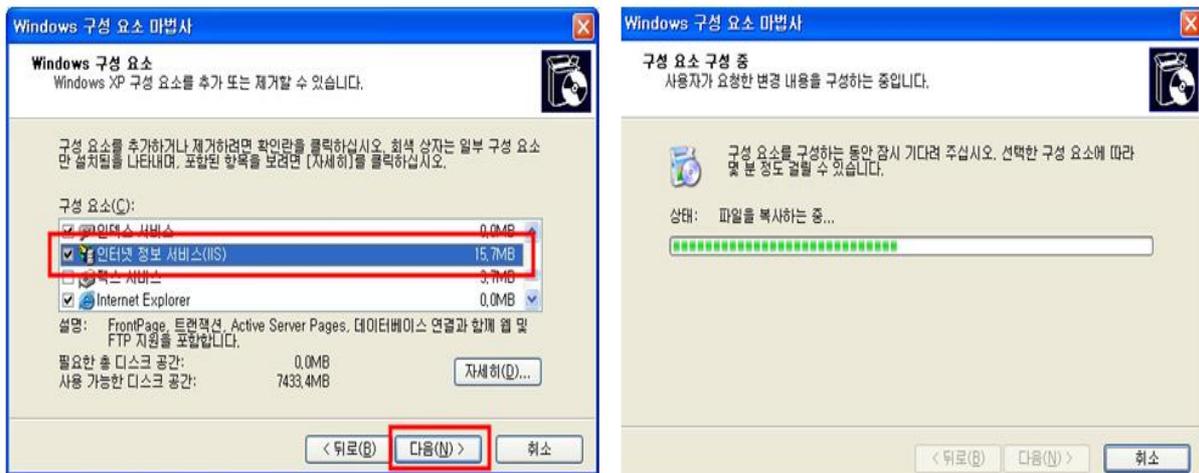
- 1) 먼저, 운영 체제가 윈도우 XP Professional 이면 다음과 같이 "C:/inetpub" 폴더가 있는지 확인한다. 이와 같은 폴더가 없으면 윈도우 CD를 준비하고 다음의 설치 과정을 따른다.



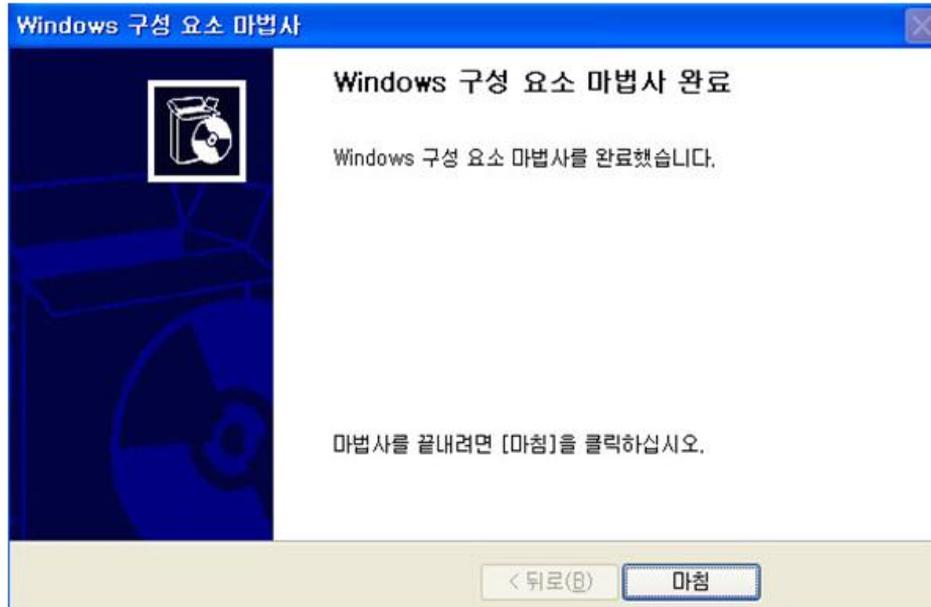
- 2) '[시작]->[제어판]->[프로그램 추가/제거]'를 실행하면 다음과 같은 화면이 나타난다. 왼쪽의 리스트 중에서 'Windows 구성 요소 추가/제거'를 선택한다.



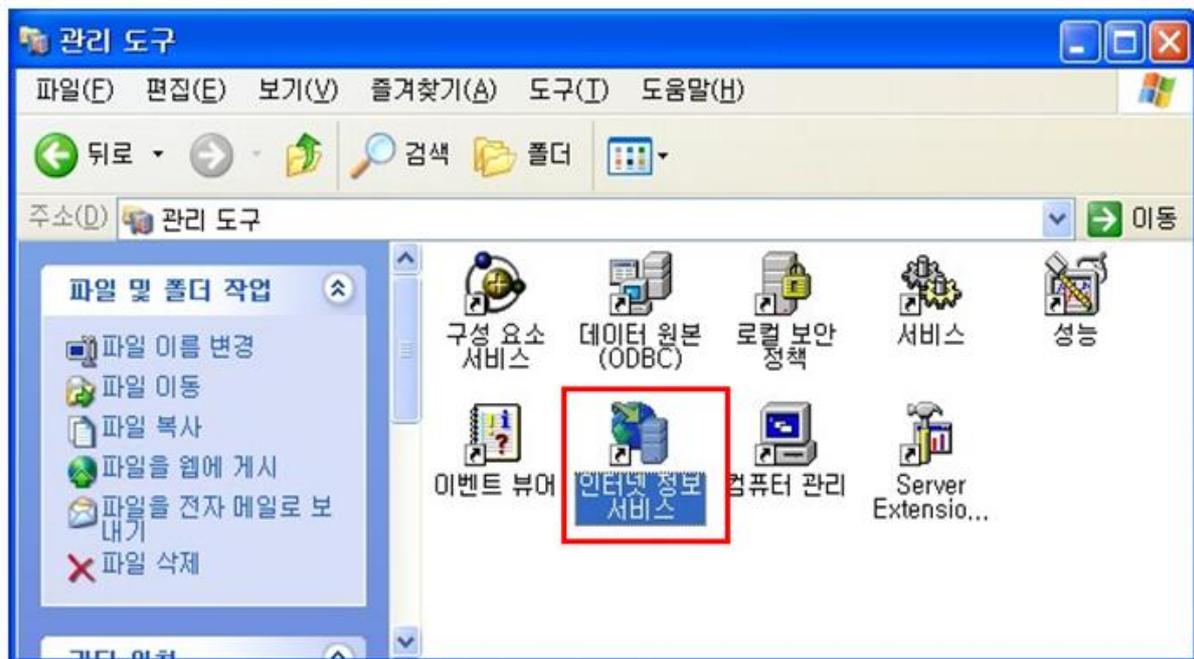
- 3) 'Windows 구성 요소 마법사' 화면이 나타나면 '인터넷 정보 서비스(IIS)'를 선택한 후 '다음' 버튼을 클릭한다. 아래의 화면과 같이 구성 요소를 설치한다.



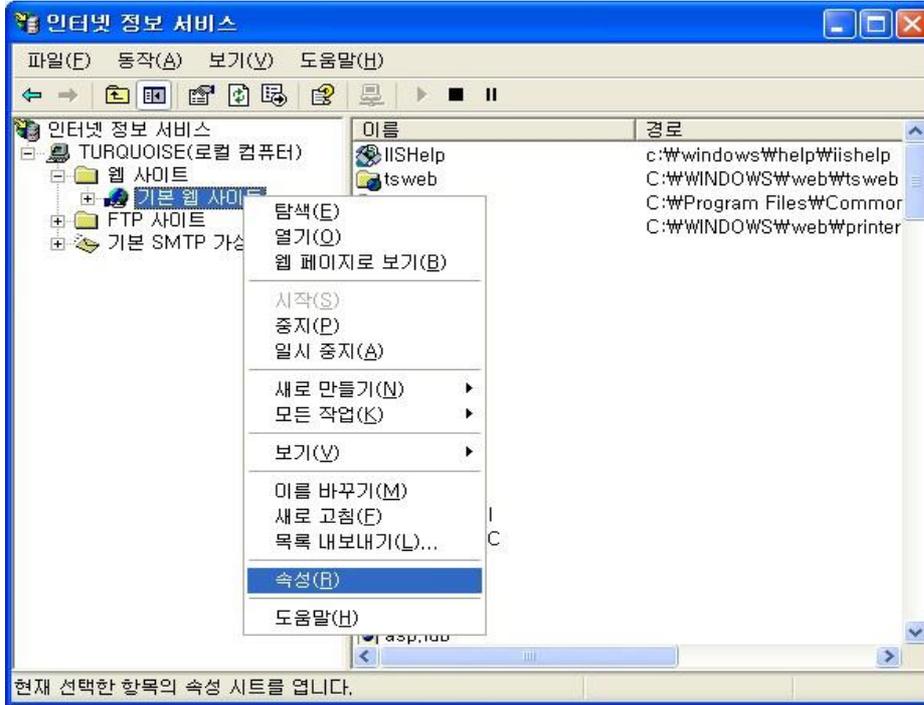
4) 구성 요소 설치가 완료되면 '마침' 버튼을 클릭하고, (1)에서 설명한 폴더가 생겼는지 확인한다.



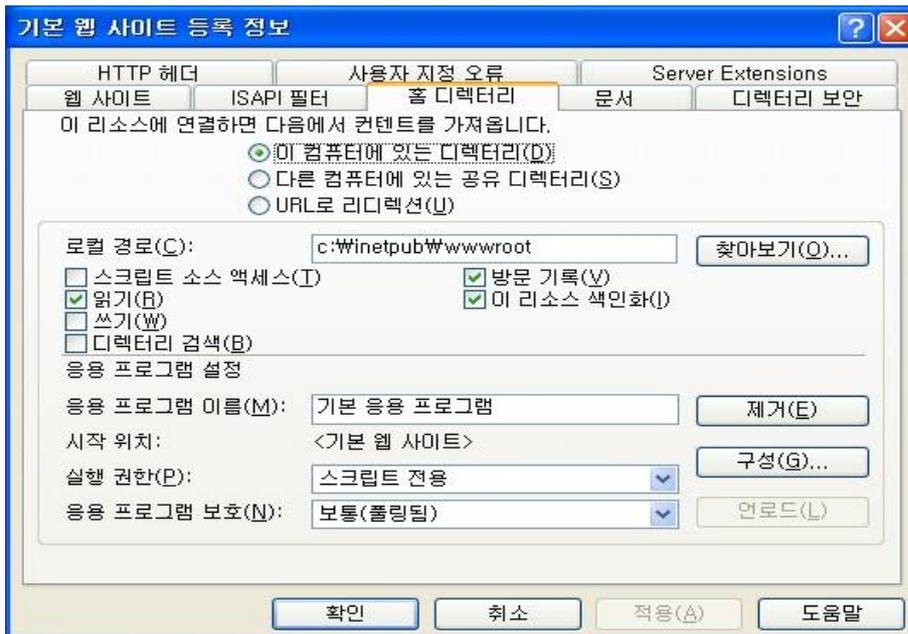
5) '[시작] -> [제어판] -> [관리 도구] -> [인터넷 정보 서비스]'를 클릭한다.



6) '기본 웹 사이트'를 선택한 후, 오른쪽 버튼을 클릭하여 '속성'을 선택한다.



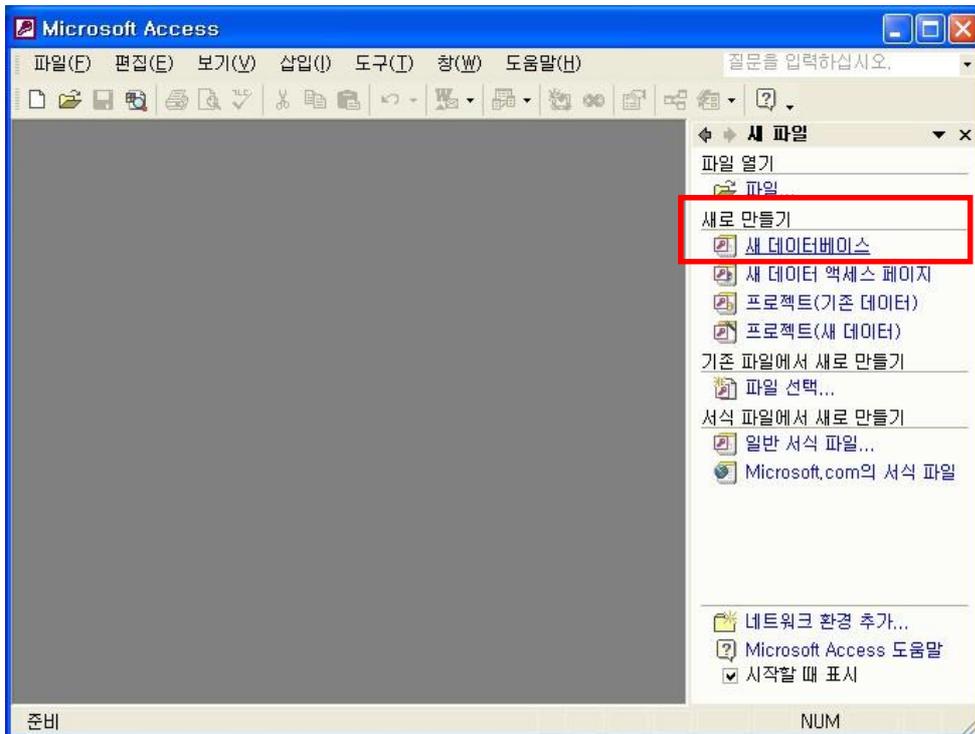
7) '홈 디렉토리'를 선택하여 기본으로 다음과 같은 설정이 되어 있는지 확인한다.



## 6.1.2 Access를 이용한 데이터베이스 구축

ASP 프로그래밍을 위해선 데이터베이스의 구축이 요구된다. 다음 절차에 따라 데이터베이스를 구축해 보자.

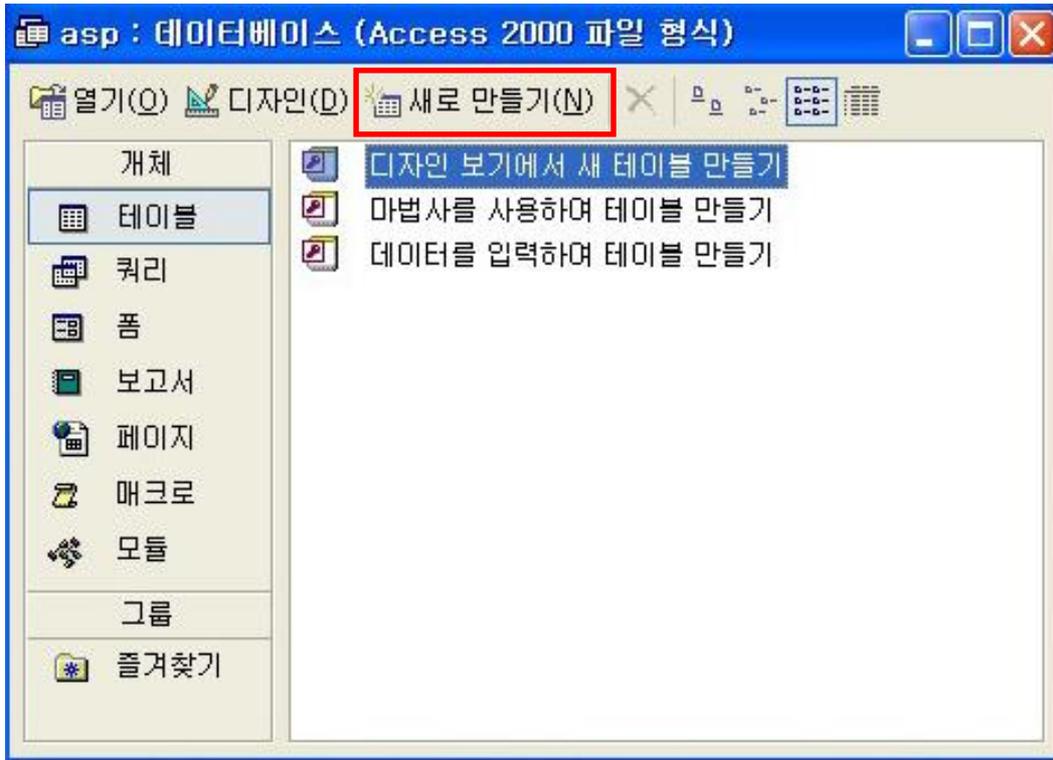
- 1) Access 를 실행하여 새로운 데이터베이스를 생성하기 위하여 “새로 만들기”의 “새 데이터베이스”를 클릭한다.



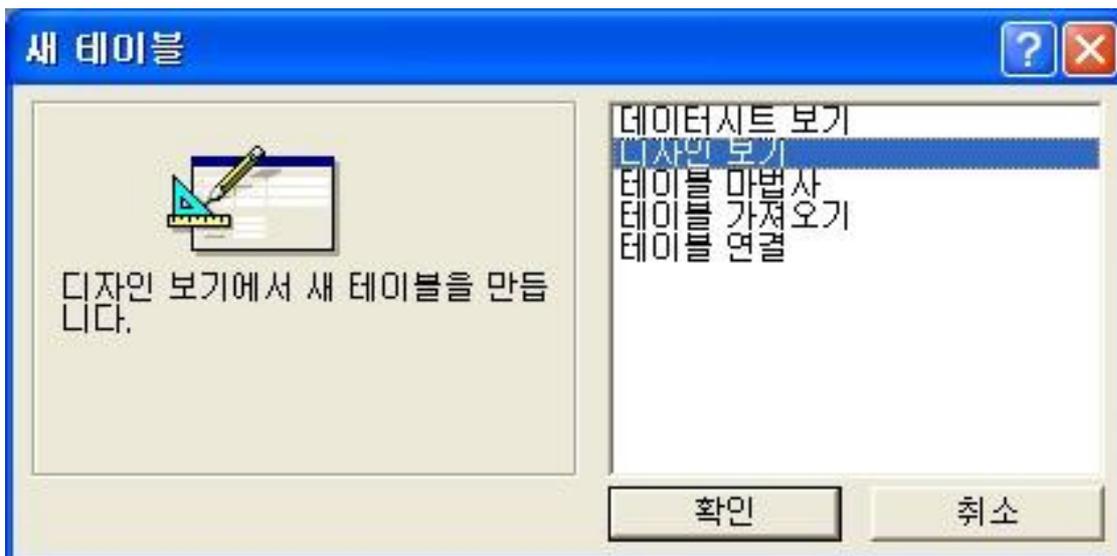
- 2) wwwroot 디렉토리에 데이터베이스를 생성하기 위하여 파일 이름에 'board\_db'라 입력하고 '만들기' 단추를 누른다.



3) 데이터베이스에 테이블 생성하기 위하여 '새로 만들기' 단추를 누른다.



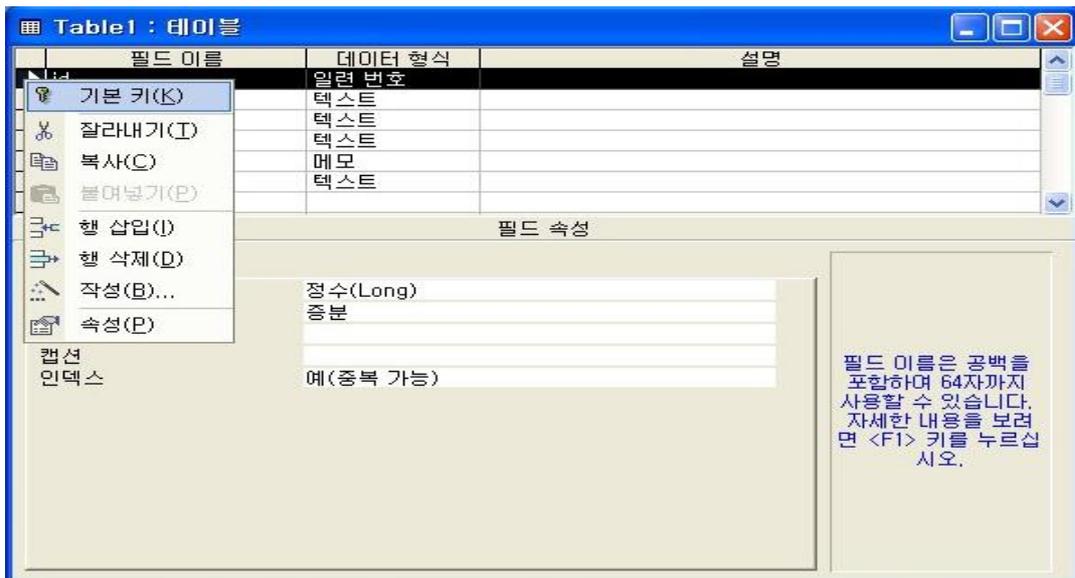
4) (3)의 결과 다음과 같은 화면이 출력되며, 이 부분에서 '디자인 보기'를 선택하고 '확인' 단추를 누른다.



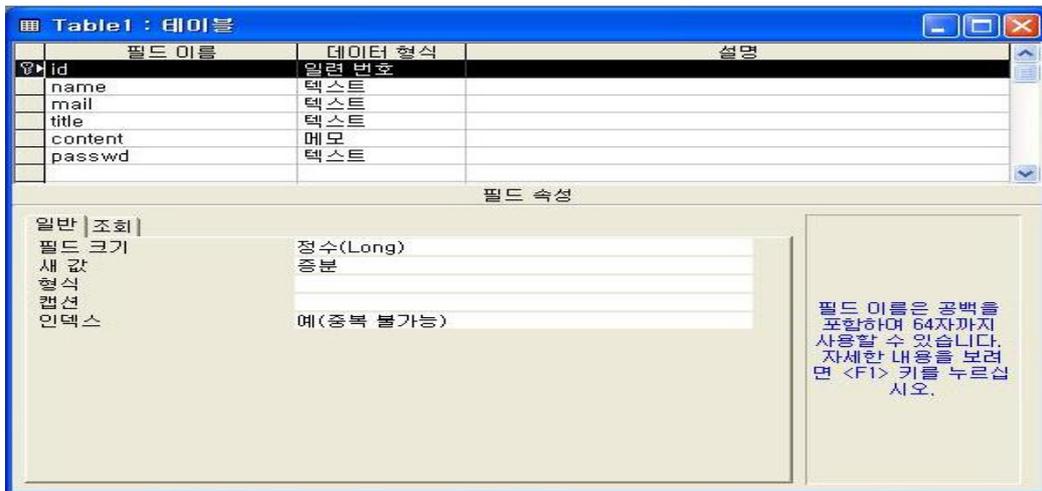
5) 테이블 작성 화면에서 다음과 같이 입력한다.



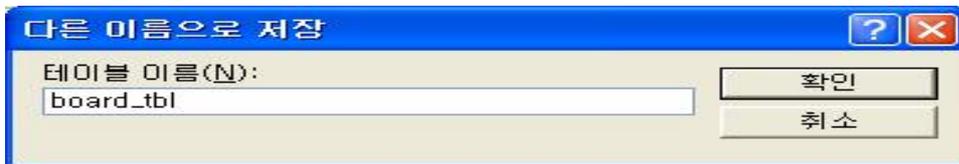
6) 다음으로 id 필드를 선택하여 마우스의 오른쪽 단추를 누른다.



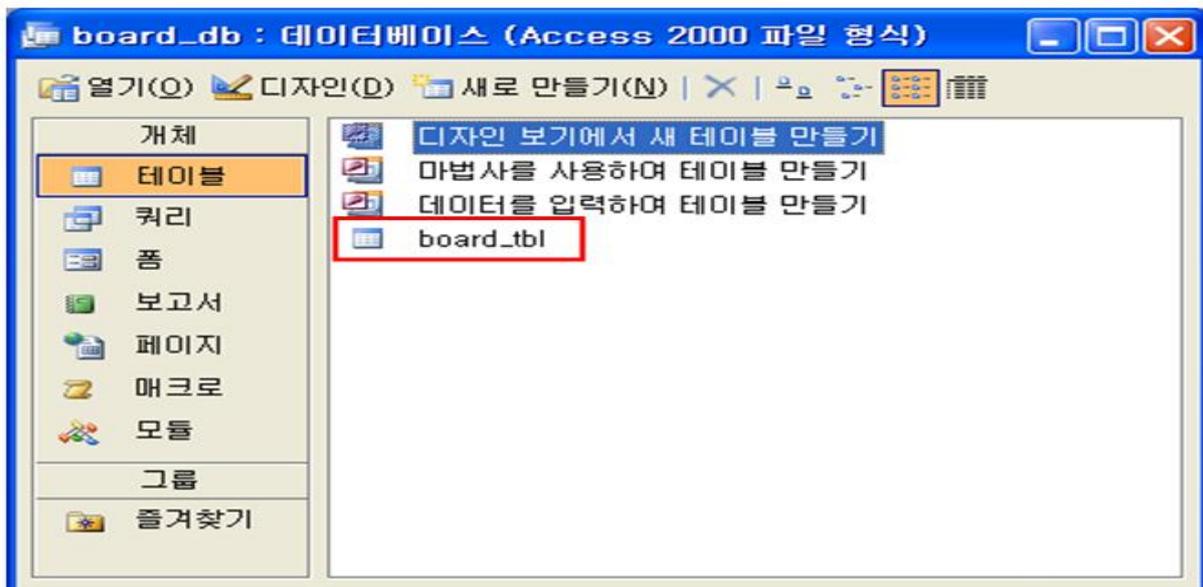
7) 기본키를 선택하여 id 필드를 기본키로 설정한다.



8) 다음 저장 버튼을 누른 후 테이블 이름을 'board\_tbl'이라 입력한다.



9) 지금까지 수행한 결과는 다음 화면과 같다.



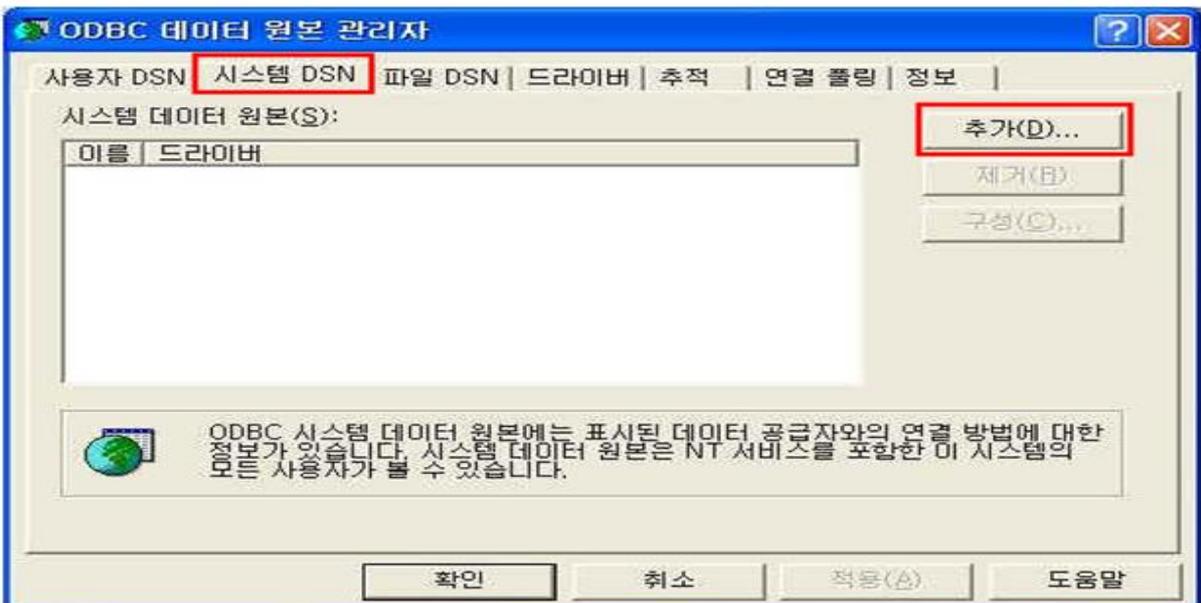
### 6.1.3 ODBC 설정

다른 프로그램 또는 외부에서 DB 를 접근하기 위해서 ODBC(Open DataBase Connectivity) 설정이 필요하다.

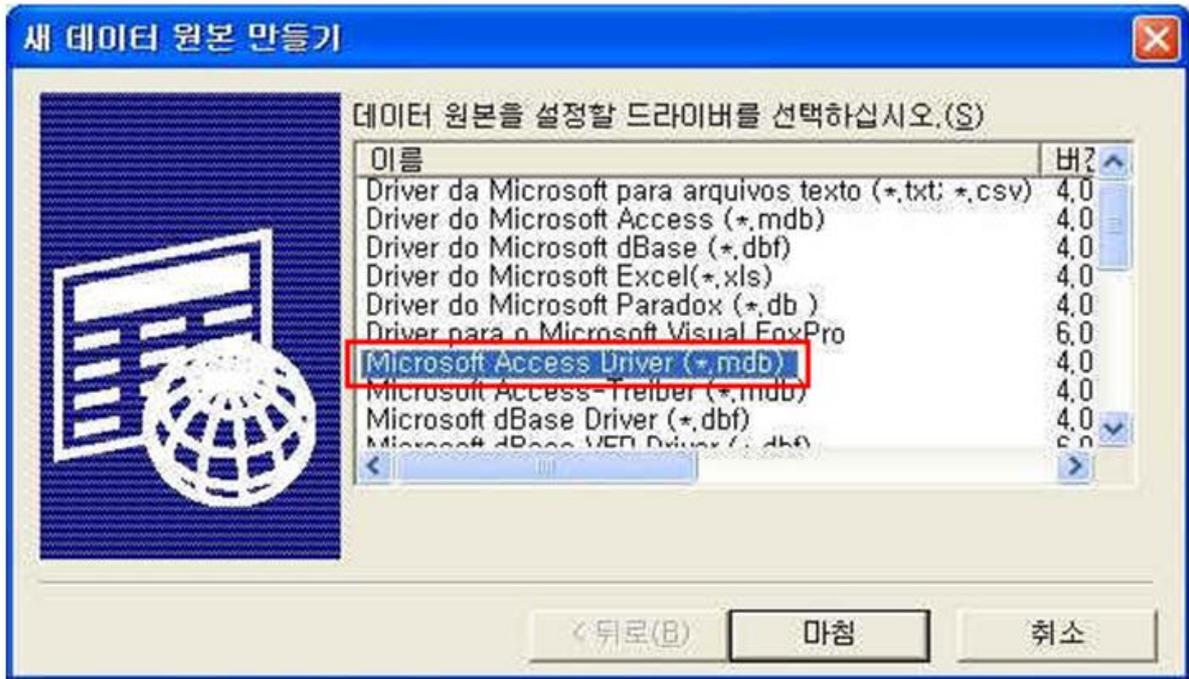
1) '[시작] -> [제어판]-> [관리도구]'에서 ODBC 를 선택한다.



2) '시스템 DSN'을 선택하여 '추가' 단추를 누른다



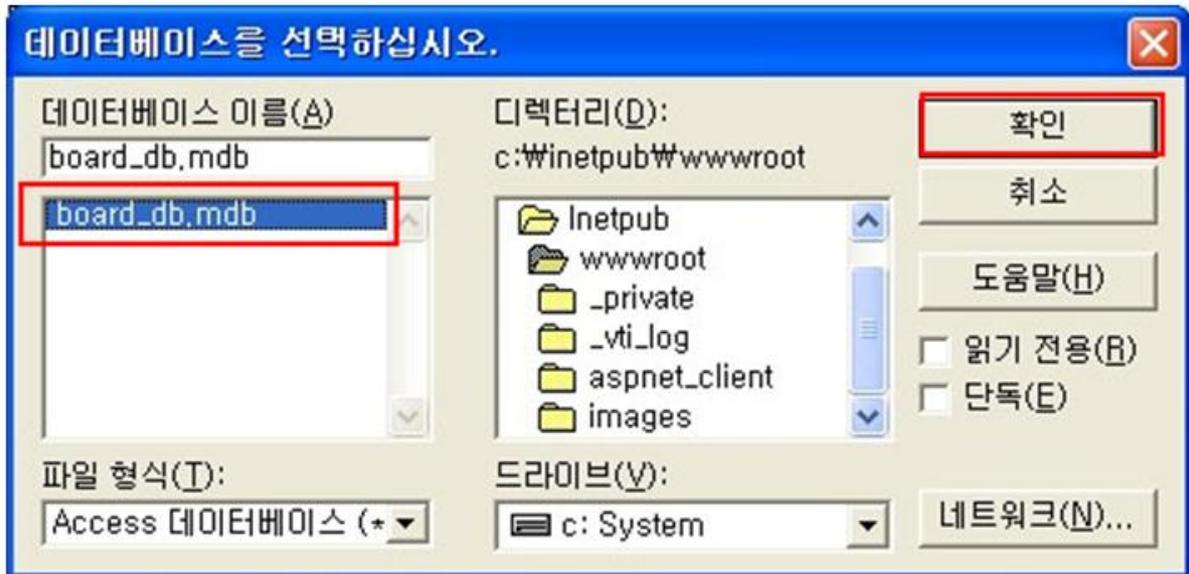
3) 데이터 원본 드라이버 중 'Microsoft Access Driver(\*.mdb)'를 선택한 후 '마침' 단추를 누른다.



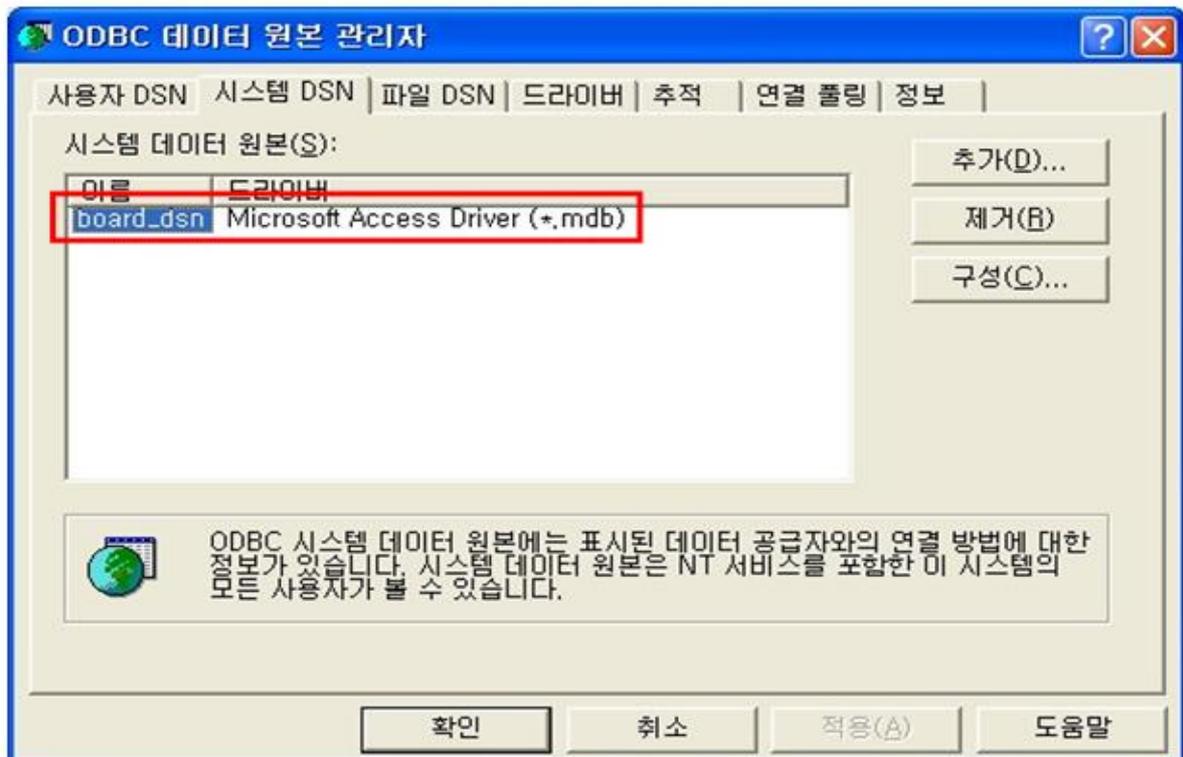
4) 데이터 원본 이름(Data Source Name)을 'board\_dsn'이라 입력한다. 또한 이 화면에서 데이터베이스 '선택' 단추를 누른다.



- 5) 데이터베이스 선택 화면에서 wwwroot 에 있는 'board\_db.mdb'를 선택하고 '확인' 단추를 누른다.



- 6) ODBC 를 성공적으로 작성한 결과는 다음과 같으며 화면에서 'board\_dsn'을 확인한다.



## 6.2 ASP 프로그래밍

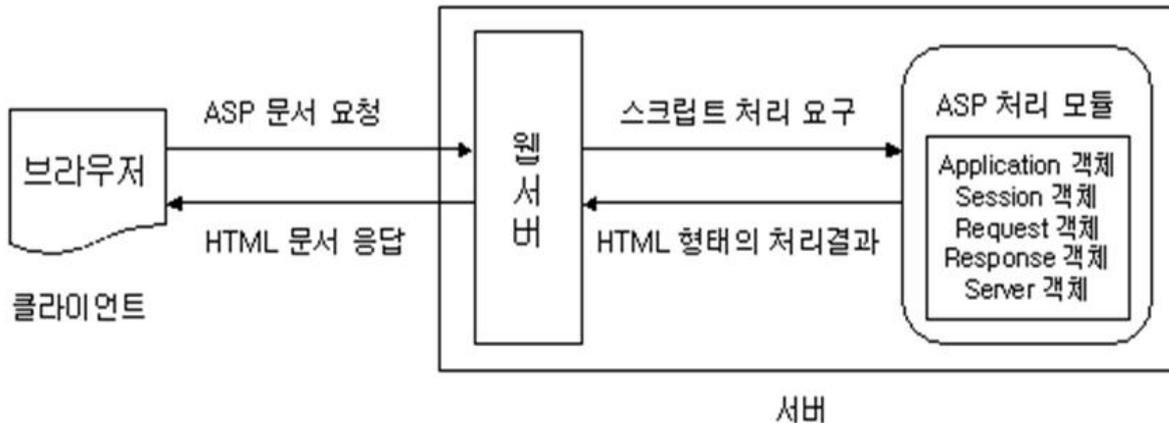
### 6.2.1 ASP 기초

ASP의 특징은 다음과 같다.

- 윈도우즈 서버 기반에서 구현
- 기존 유닉스 환경에 비하여 사용자에게 친숙한 인터페이스를 제공
- Server Side Script 를 이용하여 복잡한 서버 프로그램을 사용하지 않고서도 CGI (Common Gateway Interface)와 유사한 프로그래밍이 가능.

기존 CGI 환경에서, 서버는 클라이언트에게 정보를 제공하고, 이를 넘겨받은 클라이언트가 데이터를 처리하여 사용자에게 전달하는 과정을 수행하게되는데, 이로 인해 클라이언트의 부하가 커지는 동시에, 처리해야 할 스크립트들이 제한되는 문제점이 있었다. 반면에, ASP 는 이러한 CGI 프로그래밍의 단점을 극복하도록 개발되었다.

ASP 프로그래밍의 구조를 개략적으로 살펴보면 다음과 같다.



- (1) 클라이언트가 ASP 파일을 요청하면, ASP 처리 모듈이 이 파일 안의 스크립트를 처리
- (2) 그 결과를 웹 서버가 클라이언트에게 HTML 형태의 문서로 응답

스크립트를 처리할 때 ASP 는 내장 객체를 사용하며, 스크립트의 기능 확장을 위해 웹 서버 컴포넌트를 이용한다. ASP 는 내장 객체들을 이용하여 데이터베이스 연결, 입출력 처리, 파일 처리 등을 쉽게 할 수 있다.

## 6.2.2 Visual Basic 스크립트

ASP 문서는 대개 비주얼 베이직 스크립트를 사용한다.

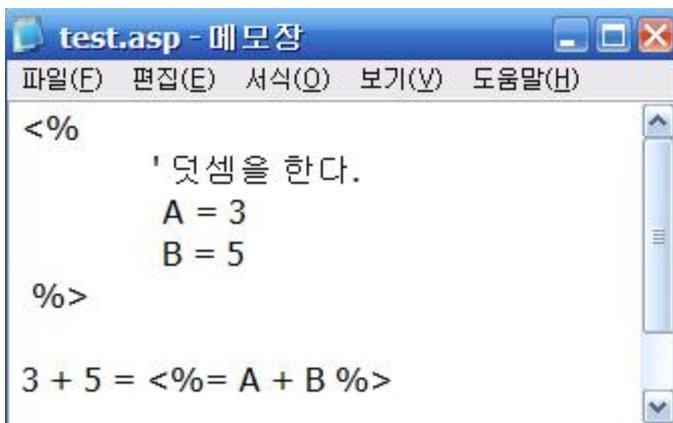
가. 기본 태그

- 보통 HTML의 태그를 주로 사용
- <Script> 태그
  - 스크립트 영역을 나타내는 데에 이용
  - 문서에 사용할 스크립트의 종류를 정의하는 속성
  - 클라이언트에서 실행할 지 아니면 서버에서 실행할 지를 나타내는 스크립트 실행 장소를 표시하는 속성
  - 속성 RUNAT의 기본값은 "Client"이고, 그 값이 "Server"의 경우 이 스크립트는 서버에서 실행됨

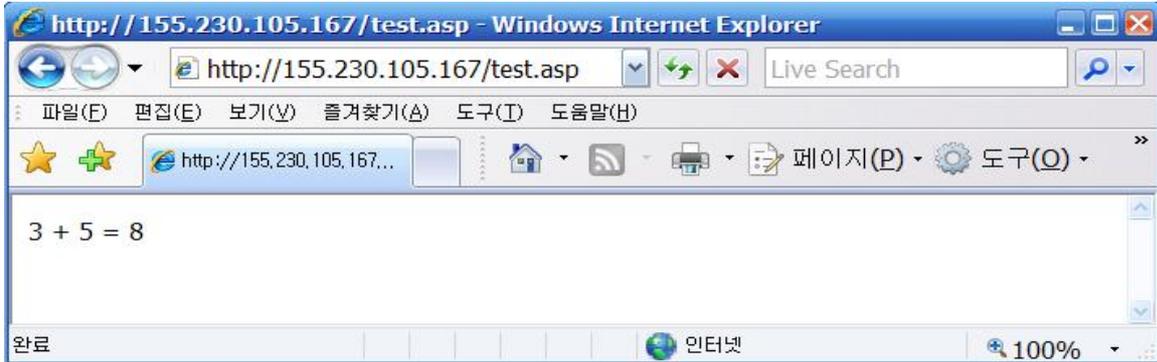
형식 :

```
<SCRIPT LANGUAGE="VBScript" RUNAT="Server">  
~ </SCRIPT>
```

- <%...%> 태그
  - 스크립트 영역을 나타내는데 <Script> 태그 이외에 <%...%> 태그를 이용
  - 서버 스크립트인 경우에만 해당
  - <%...%> 태그 부분에 수행하려는 코드를 넣으면 됨
  - ' = ' 를 사용하여 변수의 내용을 브라우저 화면에 출력할 수 있음



```
test.asp - 메모장  
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)  
<%  
    ' 덧셈을 한다.  
    A = 3  
    B = 5  
%>  
  
3 + 5 = <%= A + B %>
```



## ■ 주석

- 스크립트나 코드의 부수적인 설명
- 사용자는 주석을 이용하여 기능이나 목적을 나타낼 수 있음
- 스크립트나 코드로 처리되지 않기 때문에 프로그램에 아무런 작용도 하지 않음

형식 :  
 ' 주석 내용                    또는                    REM 주석 내용

## 나. 변수

- 변수의 이름
  - 알파벳이나 '\_' 문자나 숫자를 포함한 문자 형태로 255개를 넘지 않는 범위에서 문자들로 구성
  - 변수의 이름의 첫 글자는 항상 알파벳 문자
- 변수는 사용하기 전에 변수 선언을 하여야 함
- 변수 선언

형식 :  
 Dim A  
 Dim B, C

- <% Option Explicit %>
  - 변수를 명시적으로 선언함을 의미
  - 꼭 변수를 선언해야 함

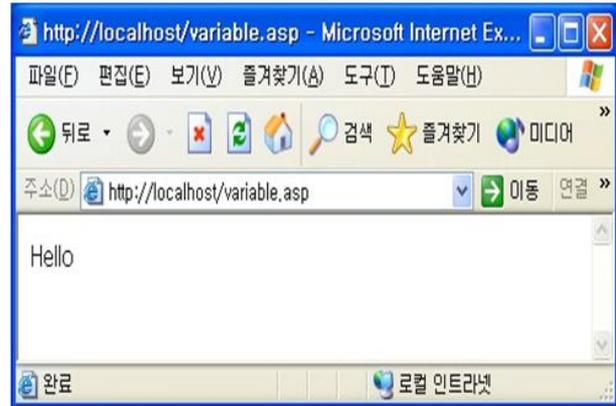
▪ 예제 : variable.asp

```

1: <% Option Explicit %>
2: <%
3:   Dim text
4:   text = "Hello"
5: %>
6: <%= text %>

```

▪ 예제 결과



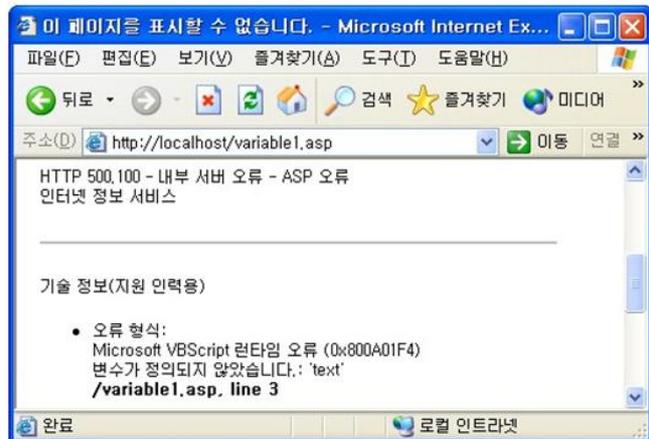
▪ 예제 : variable1.asp

```

1: <% Option Explicit %>
2: <%
3:   text = "Hello"
4: %>
5: <%= text %>

```

▪ 예제 결과



■ Variant

- VBScript에서는 하나의 데이터형만을 제공
- 이 하나의 데이터형으로 정수, 문자열 등의 다양한 형태의 데이터를 보관할 수 있음
- Variant는 서브형이 존재

다. 연산자

- 변수와 값을 가지고 연산하는데 필요한 기호
  - 덧셈, 뺄셈, 곱셈, 나눗셈 등의 연산을 함
  - 숫자들의 크기나 참, 거짓을 판단하는 데도 이용
- 
- 수학 연산자

수학 연산자	의미
+	덧셈
-	뺄셈
*	곱셈
/	나눗셈 (부동소수로 결과 출력)
\	나눗셈 (정수형으로 결과 출력)
mod	나머지

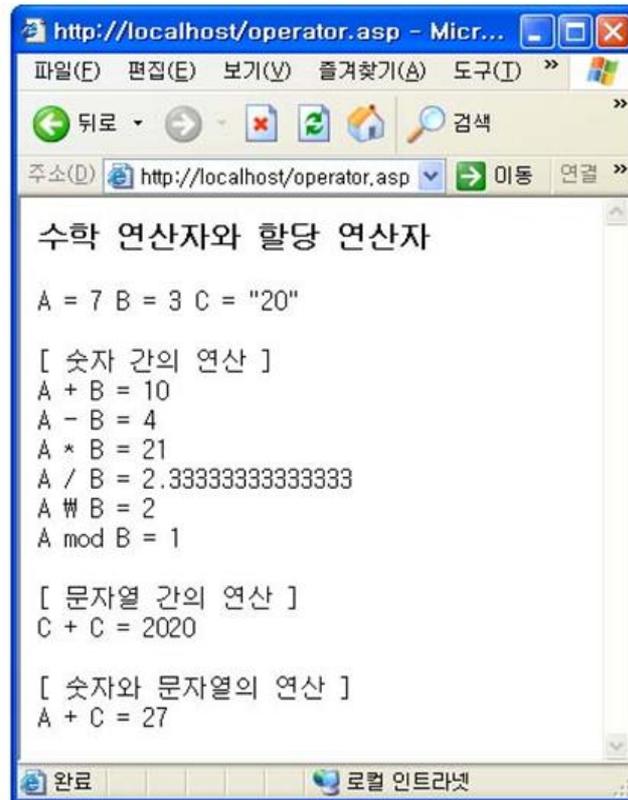
- 할당 연산자

할당 연산자	의미
=	변수에 값을 할당

- 예제 : operator.asp

```
1: <H3> 수학 연산자와 할당 연산자 </H3>
2: <%
3:   A = 7
4:   B = 3
5:   C = "20"
6: %>
7: A = 7  B = 3  C = "20"
8: <P>[ 숫자 간의 연산 ]<BR>
9: A + B = <%= A + B %> <BR>
10: A - B = <%= A - B %> <BR>
11: A * B = <%= A * B %> <BR>
12: A / B = <%= A / B %> <BR>
13: A \ B = <%= A \ B %> <BR>
14: A mod B = <%= A mod B %>
15: <P>[ 문자열 간의 연산 ]<BR>
16: C + C = <%= C + C %>
17: <P>[ 숫자와 문자열의 연산 ]<BR>
18: A + C = <%= A + C %>
```

▪ 예제 결과



▪ 비교 연산자

비교 연산자	의미
=	같다.
<>	같지않다.
<	작다.
>	크다.
<=	작거나 같다.
>=	크거나 같다.

- 논리 연산자

논리 연산자	의미
Not	부정
And	논리곱
Or	논리합
Xor	배타적 논리합
Eqv	동등
Imp	만약 (A implies B) 가 True 라면 True 값을 할당한다.

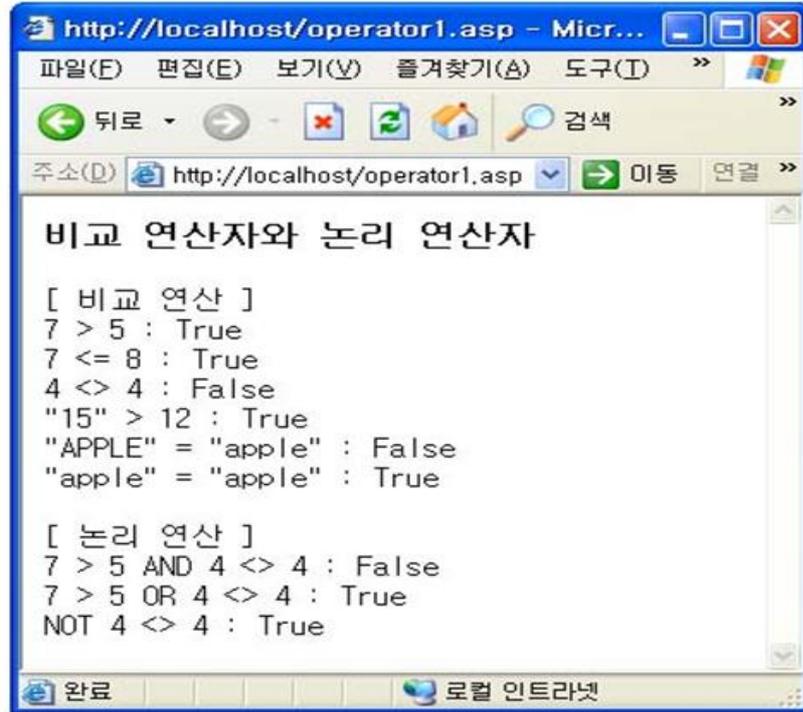
- 예제 : operator1.asp

```

1: <H3> 비교 연산자와 논리 연산자 </H3>
2: [ 비교 연산 ] <BR>
3: 7 > 5      : <%= 7 > 5 %> <BR>
4: 7 <= 8     : <%= 7 <= 8 %> <BR>
5: 4 <> 4     : <%= 4 <> 4 %> <BR>
6: "15" > 12 : <%= "15" > 12 %> <BR>
7: "APPLE" = "apple" : <%= "APPLE" = "apple" %> <BR>
8: "apple" = "apple" : <%= "apple" = "apple" %> <BR>
9: <BR> [ 논리 연산 ] <BR>
10: 7 > 5 AND 4 <> 4 : <%= 7 > 5 AND 4 <> 4 %> <BR>
11: 7 > 5 OR 4 <> 4 : <%= 7 > 5 OR 4 <> 4 %> <BR>
12: NOT 4 <> 4 : <%= NOT 4 <> 4 %>

```

- 예제 결과



- & 연산자

- 문자열을 연결
- 위에서 '+' 연산자를 이용하여 문자열을 연결하는 것을 보았지만, 숫자간의 덧셈 연산과 혼동될 수 있으므로 보통 '&' 연산자를 이용

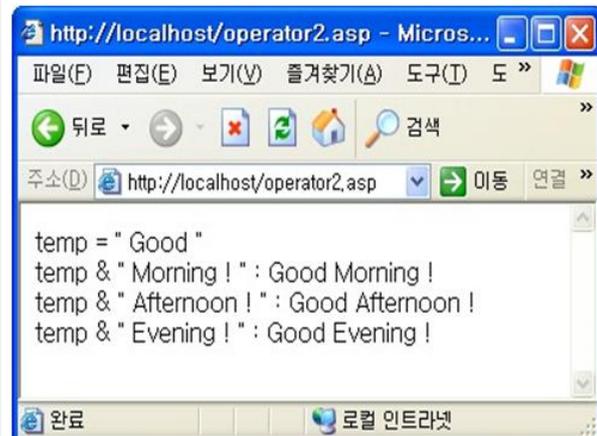
- 예제 : operator2.asp

```

1 : <%
2 :   temp = " Good "
3 : %>
4 :   temp = " Good " <BR>
5 :   temp & " Morning ! " :
   <%= temp & " Morning ! " %> <BR>
6 :   temp & " Afternoon ! " :
   <%= temp & " Afternoon ! " %>
   <BR>
7 :   temp & " Evening ! " :
   <%= temp & " Evening ! " %>

```

- 예제 결과



## 라. 배열

- 배열은 같은 변수에 하나 이상의 값을 저장하고자 할 때 사용하는 것
- 1차원, 2차, 3차 배열도 이용
- 'Dim' - 괄호를 이용하여 크기를 선언
- 스크립트의 실행 도중에 배열의 크기와 내용을 변경할 경우에는 ReDim문을 함께 이용하여 선언
- 배열의 크기를 변경하지 않을 경우에는 Dim만을 사용하여도 됨
- 이전의 배열 내용을 보존하여 추가로 공간을 늘리고 싶을 때 Preserve 문을 이용

### ■ 예제 : array.asp

```
1: <%
2:   Dim jewel()
3:   Redim jewel(3)
4:
5:   jewel(0) = "pearl"
6:   jewel(1) = "crystal"
7:   jewel(2) = "jade"
8:   jewel(3) = "sapphire"
9: %>
10: [ 배열 jewel(3)의 내용 ]<BR>
11: <%= jewel(0)%>. <%= jewel(1)%>. <%= jewel(2)%>. <%= jewel(3)%>.
12:
13: <%
14:   Redim Preserve jewel(5)
15:   jewel(4) = "emerald"
16:   jewel(5) = "diamond"
17: %>
18: <br><br>[ 기존의 배열에 내용을 추가한 jewel(5)의 내용 ]<BR>
19: <%= jewel(0)%>. <%= jewel(1)%>. <%= jewel(2)%>. <%= jewel(3)%>.
20: <%= jewel(4)%>. <%= jewel(5)%>.
```

▪ 예제 결과



마. 조건문

- 조건에 따라 실행문을 선택하는 것
- **If 문**
  - If문은 주어진 조건의 결과에 따라 실행하는 부분이 다름

```
형식
If 조건 Then
    ' 참 ' 일 경우 실행
Else
    ' 거짓' 일 경우 실행
End If
```

▪ 예제 : if.asp

```
1: <%  
2:  score = 95  
3:  If (score >= 90) Then  
4:    temp = "Pass"  
5:  Else  
6:    temp = "Fail"  
7:  End If  
8: %>  
9: <%= temp %>
```

▪ 예제 결과



▪ Select Case 문

- 조건을 여러 개 두어 한번에 검사
- 비교할 값을 각 Case 문과 비교하고 일치하는 경우에는 그 해당 스크립트를 실행

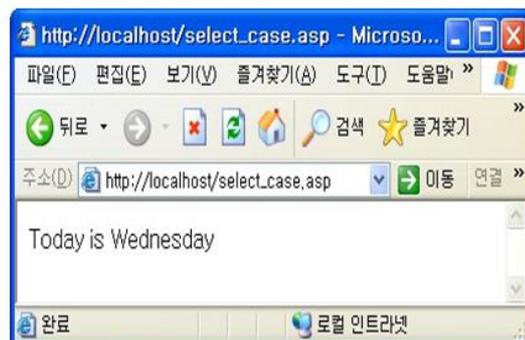
형식 :

```
Select Case 변수이름  
  Case " 조건 1 "  실행 1  
  Case " 조건 2 "  실행 2  
  Case " 조건 3 "  실행 3  
  Case Else       실행 4  
End Select
```

▪ 예제 : select\_case.asp

```
1 : <%  
2 :  today = " Wed. "  
3 :  Select Case today  
4 :  Case " Mon. "  show = "Monday"  
5 :  Case " Tue. "  show = "Tuesday"  
6 :  Case " Wed. "  show = "Wednesday"  
7 :  Case " Thur. " show = "Thursday"  
8 :  Case " Fri. "  show = "Friday"  
9 :  Case " Sat. "  show = "Saturday"  
10 : Case Else     show = "Sunday"  
11 : End Select  
12 : %>  
13 : Today is <%= show %>
```

▪ 예제 결과



## 바. 반복문

- 어떤 문장을 여러 번 반복하고자 할 때 이용
- For 문
  - 시작값과 끝값을 정하여 그것을 이용하여 문장을 반복
  - Step을 이용하여 변환하는 값을 정해줌
  - 특별히 지정해 주지 않는다면 값을 하나씩 증가

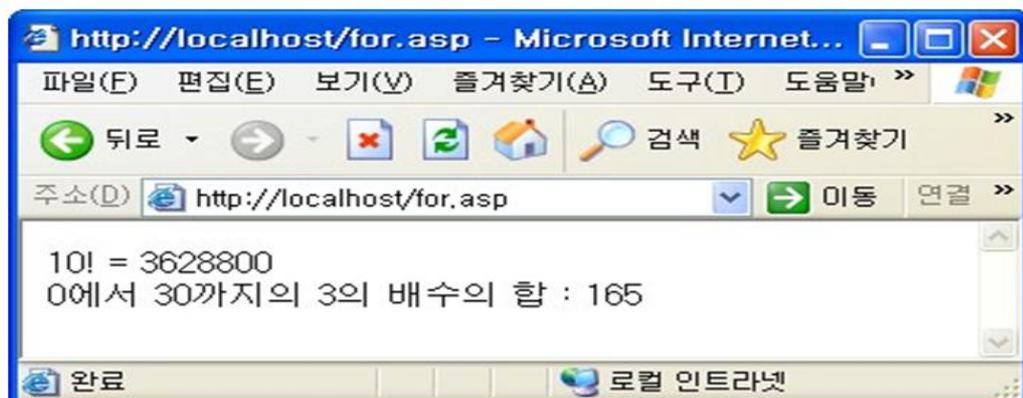
형식 :

```
For i = 시작값 to 끝값 Step 변환값  
    문장  
Next
```

- 예제 : for.asp

```
1: <%  
2:     Fac = 1  
3:     For i = 1 to 10  
4:         Fac = Fac * i  
5:     Next  
6: %>  
7: 10! = <%= Fac %> <BR>  
8: <%  
9:     temp = 0  
10:    For i = 0 to 30 Step 3  
11:        temp = temp + i  
12:    Next  
13: %>  
14: 0에서 30까지의 3의 배수의 합 : <%= temp %>
```

- 예제 결과



- Do 문
  - Do 문은 Loop문을 만날 때까지 이들 사이에 존재하는 문장을 반복
  - While과 Until를 이용하여 조건을 두는데, 조건을 처음 시작할 때 또는 마지막에 둘 수도 있음

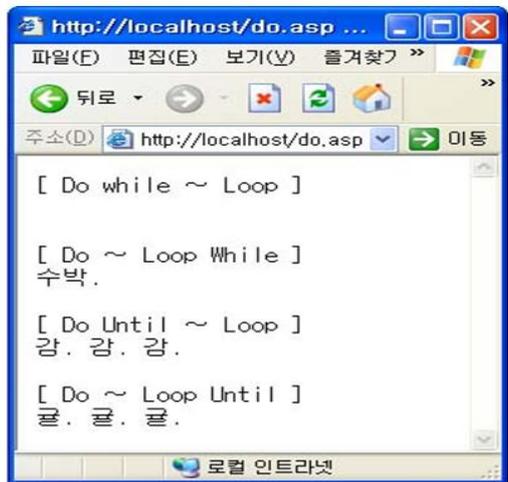
```

형식 :
      Do While (또는 Until) 조건
      문장
      Loop
      또는
      Do
      문장
      Loop While (또는 Until) 조건
  
```

■ 예제 : do.asp

<pre> 1: [ Do while ~ Loop ] &lt;P&gt; 2: &lt;% 3:     i = 0 4:     Do while i &lt; 0 5:         i = i + 1 6:     %&gt; 7:     딸기. 8: &lt;% Loop %&gt; 9: 10: &lt;BR&gt; [ Do ~ Loop While ] &lt;BR&gt; 11: &lt;% 12:     i = 0 13:     Do 14:         i = i + 1 15:     %&gt; 16:     수박. 17: &lt;% Loop While i &lt; 0 %&gt; 18:   </pre>	<pre> 19: &lt;P&gt;[ Do Until ~ Loop ] &lt;BR&gt; 20: &lt;% 21:     i = 0 22:     Do Until i = 3 23:         i = i + 1 24:     %&gt; 25:     감. 26: &lt;% Loop %&gt; 27: 28: &lt;P&gt;[ Do ~ Loop Until ] &lt;BR&gt; 29: &lt;% 30:     i = 0 31:     Do 32:         i = i + 1 33:     %&gt; 34:     귤. 35: &lt;% Loop Until i = 3 %&gt;   </pre>
---	--

■ 예제 결과



- While 문
  - 조건이 참일 동안만 주어진 문장을 반복 수행
  - 조건이 거짓이 되면 즉시 수행을 그만 둠

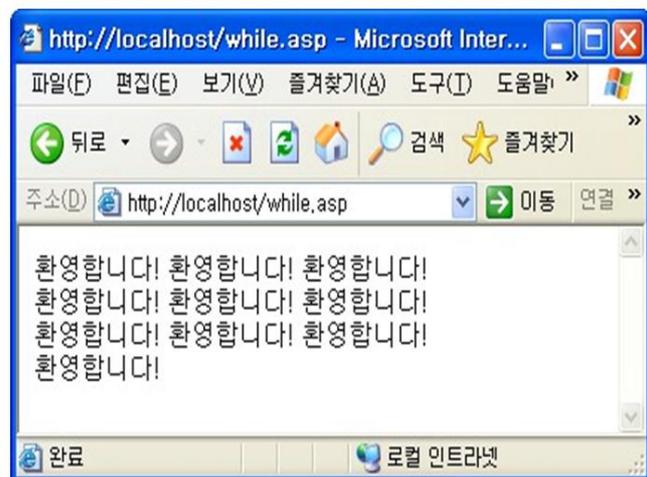
형식 :

```
While 조건
    문장
WEnd
```

- 예제 : while.asp

```
1 : <%
2 :     i = 0
3 :     While( i <10)
4 :         i = i + 1
5 :     %>
6 :     환영합니다!
7 :
8 : <% If (i mod 3) = 0 Then %>
9 :
10 :     <BR>
11 :
12 : <% End If %>
13 :
14 : <% WEnd %>
```

- 예제 결과



사. 프로시저 (procedure)

- 여러 번 자주 실행해야 할 부분을 하나의 모듈로 모아 놓은 것
- 서브루틴(Subroutine)
- 함수(Function)
  
- 서브루틴
  - 실행중인 스크립트 부분에서 Sub를 부르면 매개변수를 이용하여 Sub는 해당 부분을 수행

형식 :

```
Sub 서브이름 (매개변수)
...
End Sub
```

- 함수
  - Function은 매개변수를 이용하여 값을 받아서 수행한 후에 그 결과를 Function의 이름으로 전달

형식 :

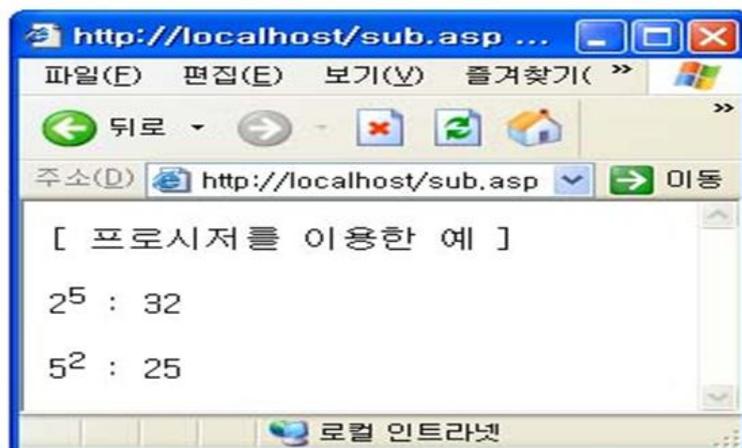
```
Function 함수이름 (매개변수)
...
함수이름 = 리턴값
End Function
```

■ 예제 : sub.asp

```
1: <%  
2:   Function square (i, J)  
3:     value = 1  
4:     k = 0  
5:     While k < J  
6:       value = value * i  
7:       k = k + 1  
8:     WEnd  
9:     square = value  
10: End Function  
11:  
12: Sub Show(a)  
13:   Response.Write(a)
```

```
14: End Sub  
15: %>  
16: <%  
17:   Show (" [ 프로시저를 이용한  
18:     예 ]")  
19:   p = square(2, 5)  
20: %>  
21: <P>   2<SUP>5</SUP> : <%=  
22:   p %>  
23:   q = square(5, 2)  
24: %>  
25: <P>   5<SUP>2</SUP> : <%=  
26:   q %>
```

■ 예제 결과



아. 내장 함수

[표] VBScript에서의 내장 함수

함수	설명	사용 예
Date()	현재의 날짜를 알려주는 함수	<%= Date() %>
TypeName()	데이터의 타입을 알려주는 함수	<%= TypeName(56.7)%>
CStr()	데이터를 문자열로 바꾸는 함수	<%= CStr(56.7) %>
CInt()	데이터를 정수형으로 바꾸는 함수	<%= CInt("56.7") %>
Replace()	문장 중에서 특정한 부분을 찾아서 다른 데이터로 교체하는 함수	<%= Replace("This is a book", "book", "desk") %>
UCase()	알파벳 글자를 대문자로 바꾸는 함수	<%= UCase("milk") %>
LCase()	알파벳 글자를 소문자로 바꾸는 함수	<%= LCase("MILK") %>
StrComp()	두 문자열이 같은지를 비교하는 함수	<%= StrComp("HOUSE","House",1) %>
FormatDateTime()	날짜에 대한 출력 형식을 결정하는 함수	<%= FormatDateTime(Date(), vbLongDate) %>
FormatPercent()	퍼센트를 표시하는 함수	<%= FormatPercent(0.4507, 1) %>
FormatNumber()	숫자를 표시하는 함수	<%= FormatNumber(35.987, 2) %>

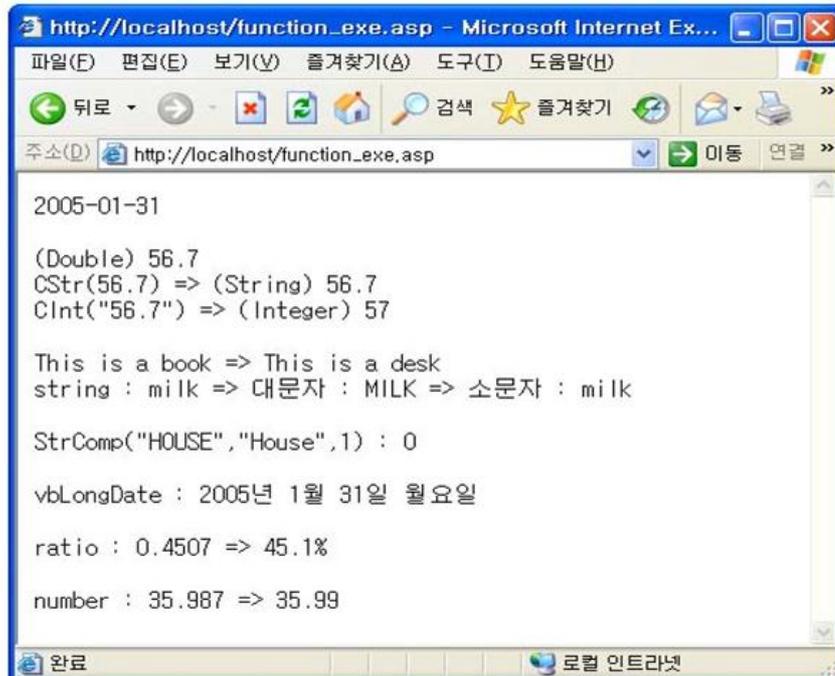
▪ 예제 : function\_exe.asp

```

1: <%=Date() %>
2: <P><%=TypeName(56.7)%> <%= 56.7 %>
3: <BR>CStr(56.7) => (<%=TypeName(CStr(56.7))%>) <%= CStr(56.7) %>
4: <BR>CInt("56.7") => (<%=TypeName(CInt("56.7"))%>) <%= CInt("56.7") %>
5: <P>This is a book => <%= Replace("This is a book", "book", "desk") %>
6: <BR>string : milk
7: => 대문자 : <%= UCase("milk") %> => 소문자 : <%= LCase("MILK") %>
8: <P>StrComp("HOUSE","House",1) : <%= StrComp("HOUSE","House",1)%>
9: <P>vbLongDate : <%= FormatDateTime(Date(), vbLongDate) %>
10: <P>ratio : 0.4507 => <%= FormatPercent(0.4507, 1) %>
11: <P>number : 35.987 => <%= FormatNumber(35.987, 2) %>

```

■ 예제 결과



### 6.2.3 ASP 객체

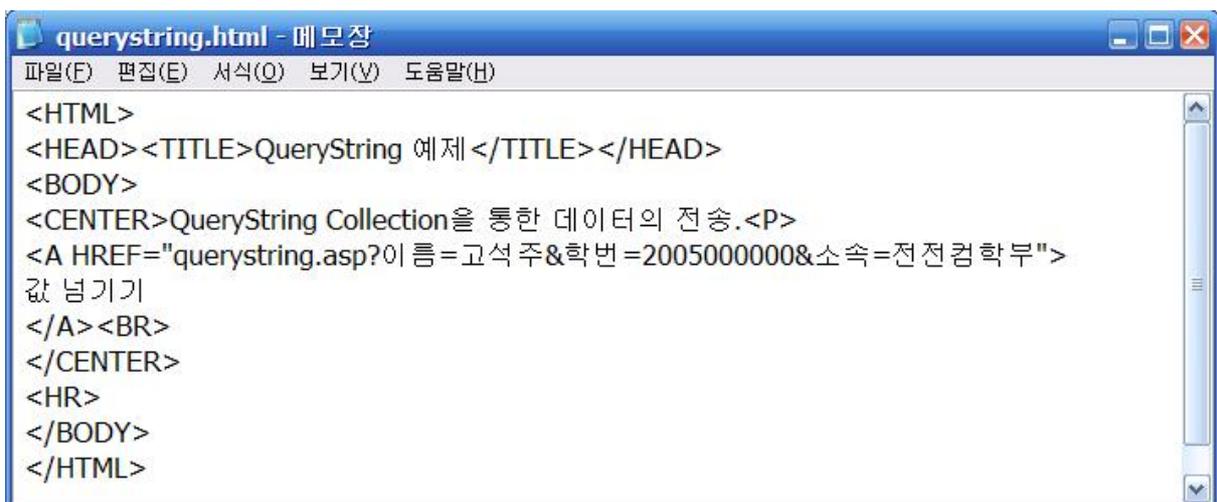
ASP 에서 지원하는 주요 객체는 다음과 같다.

- Request 객체
- Response 객체
- ADO 객체

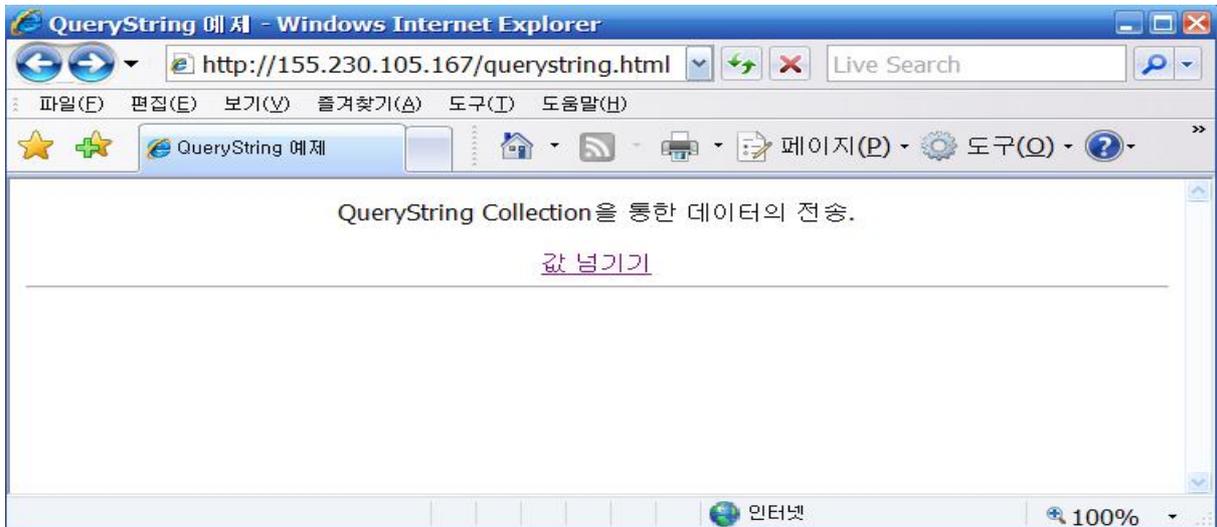
#### 가. Request 객체

- Request 객체 (사용자의 정보를 받아오는 객체)
  - QueryString Collection
  - Form Collection
- QueryString Collection
  - 사용자의 입력 없이 다른 페이지에 값을 전달
- `<a href="A.asp?변수1=값1&변수2=값2">`

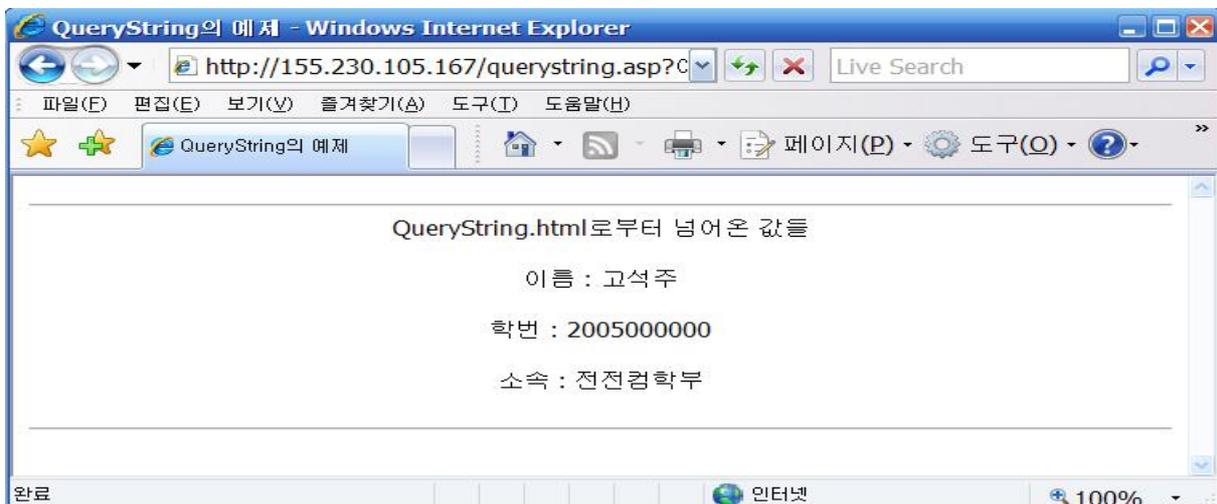
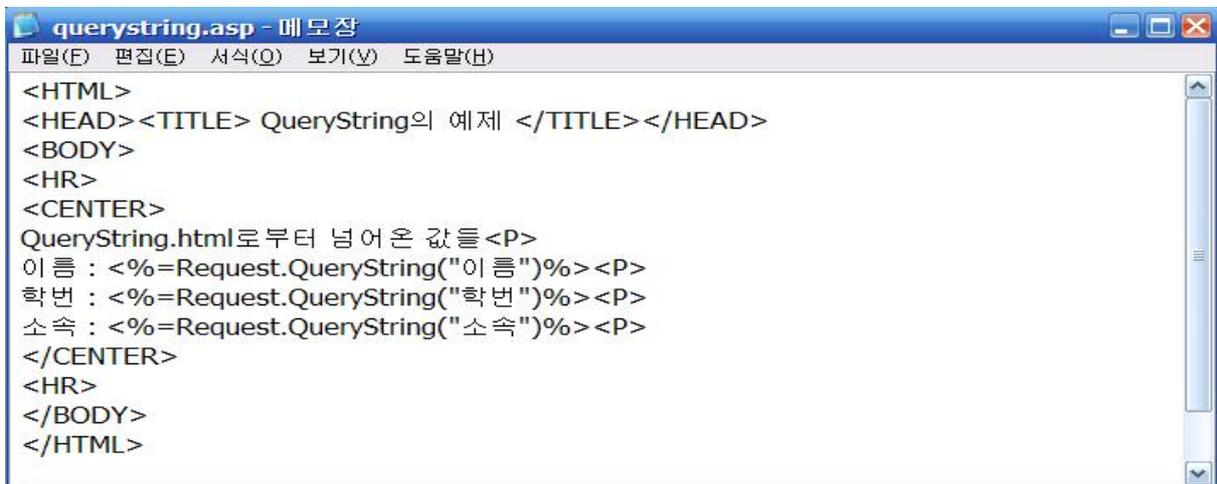
예제 QueryString.html - 값 넘기기



```
<HTML>
<HEAD><TITLE>QueryString 예제 </TITLE></HEAD>
<BODY>
<CENTER>QueryString Collection을 통한 데이터의 전송.<P>
<A HREF="querystring.asp?이름=고석주&학번=2005000000&소속=전전컴학부">
값 넘기기
</A><BR>
</CENTER>
<HR>
</BODY>
</HTML>
```



예제 QueryString.asp- 넘어온 값 받기



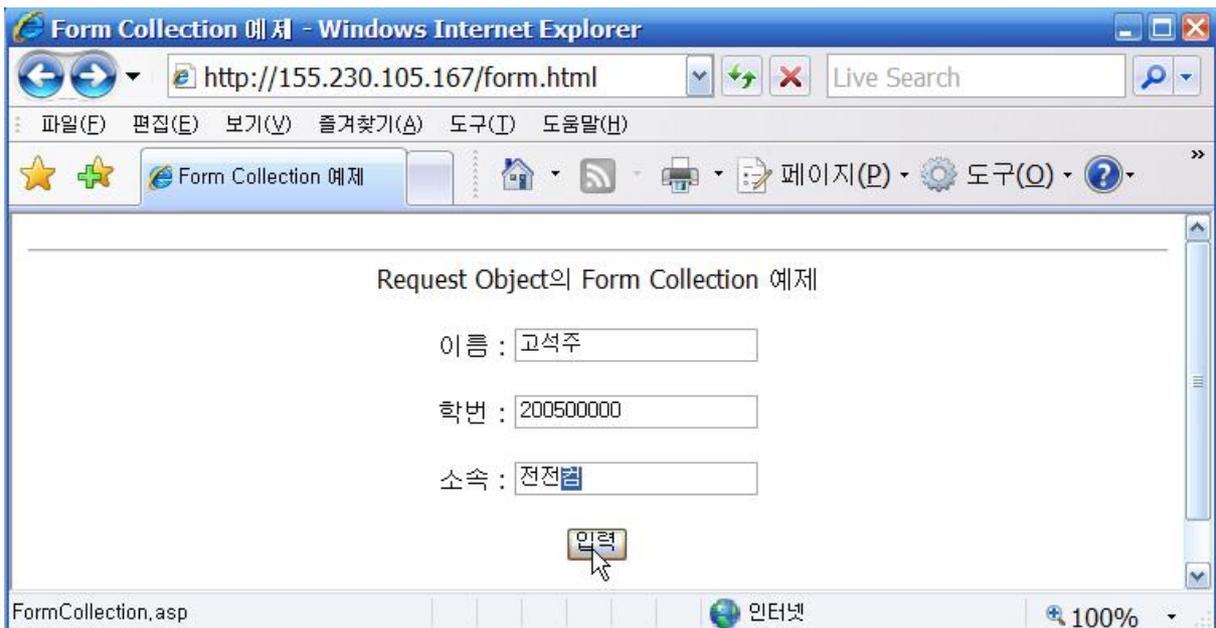
- Form Collection
  - 사용자로부터 직접 입력 받은 값을 처리

예제: Form.html - 값 넘기기

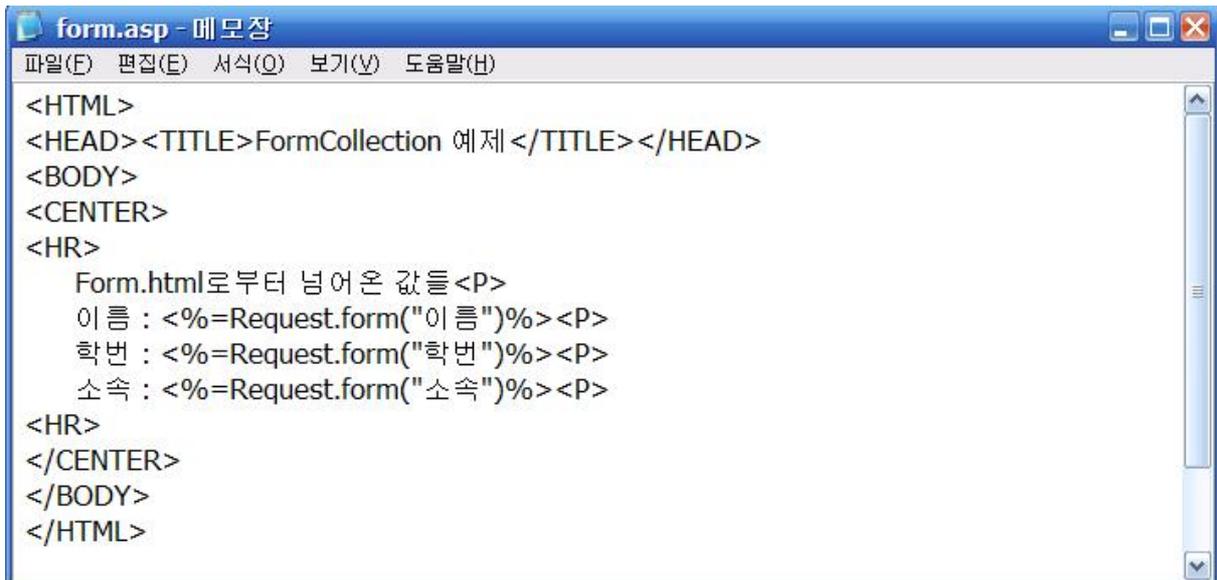
```

form.html - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<HTML>
<HEAD><TITLE> Form Collection 예제 </TITLE></HEAD>
<BODY>
<CENTER>
<HR>
Request Object의 Form Collection 예제 <P>
<FORM ACTION="form.asp" METHOD="post">
이름 : <INPUT TYPE="text" NAME="이름"><P>
학번 : <INPUT TYPE="text" NAME="학번"><P>
소속 : <INPUT TYPE="text" NAME="소속"><P>
<INPUT TYPE="submit" VALUE="입력">
</FORM>
<HR>
</BODY>
</HTML>

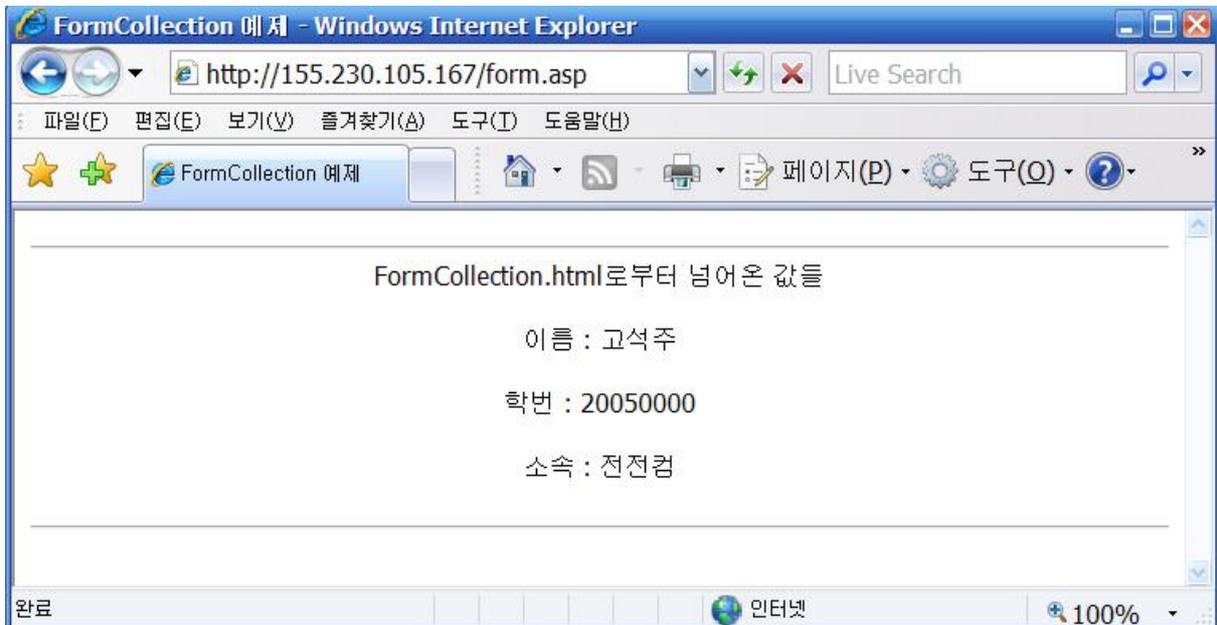
```



예제: Form.asp - 넘어온 값 받기



```
<HTML>
<HEAD><TITLE>FormCollection 예제 </TITLE></HEAD>
<BODY>
<CENTER>
<HR>
  Form.html로부터 넘어온 값들 <P>
  이름 : <%=Request.form("이름")%> <P>
  학번 : <%=Request.form("학번")%> <P>
  소속 : <%=Request.form("소속")%> <P>
<HR>
</CENTER>
</BODY>
</HTML>
```



## 나. Response 객체

### ■ Response 객체

- 서버에서 클라이언트로 데이터를 전송할 때 사용
- Response 객체의 메소드

메소드	내용
Write	브라우저에 문자열을 출력
Redirect	브라우저에서 다른 URL의 페이지로 이동

### ■ Write 메소드를 이용한 문자열의 출력

스크립트를 이용한 형식 :

문자열 출력의 경우 : <%= 출력할 문자열에 대한 변수 %>  
정수값 출력의 경우 : <%= 출력할 정수값에 대한 변수 %>

Response.Write의 형식 :

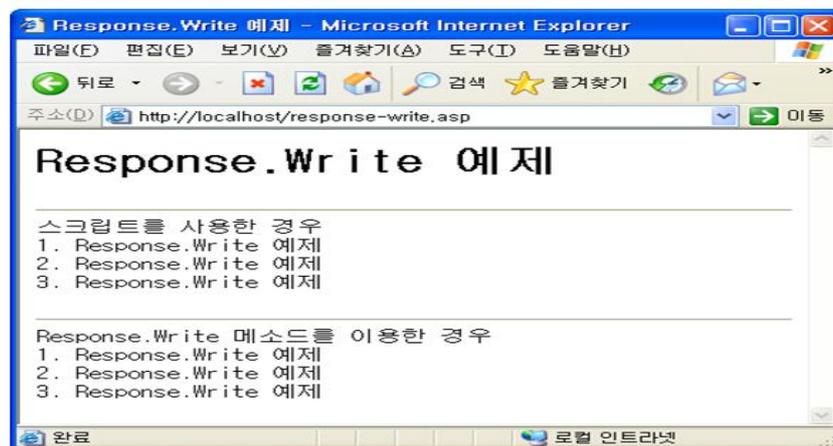
```
<%  
    Reponse.Write( " 문자열 또는 변수 값 " )  
    Reponse.Write( " 정수 값 또는 변수 값 " )  
%>
```

■ 예제 : Response-Write.asp

```
1: <HTML>
2: <HEAD><TITLE>Response.Write 예제</TITLE></HEAD>
3: <BODY>
4: <H1> Response.Write 예제 </H1>
5: <HR>
6: <% str= "서버 스크립트를 사용한 경우 <BR>" %>
7: <%= str %>
8:
9: <% For i = 1 to 3 %>
10: <% = i %>. Response.Write 예제 <BR>
11: <% Next %>
12: <P><HR>
```

```
13: <% Response.Write("Response.Write 메소드를 이용한 경우") %><BR>
14: <%
15:     For i = 1 to 3
16:         strTemp = i & ". Response.Write 예제 <BR>"
17:         Response.Write( strTemp )
18:     Next
19: %>
20: </BODY>
21: </HTML>
```

■ 예제 결과



- Redirect 메소드를 이용한 페이지 이동

Redirect 메소드의 형식 :

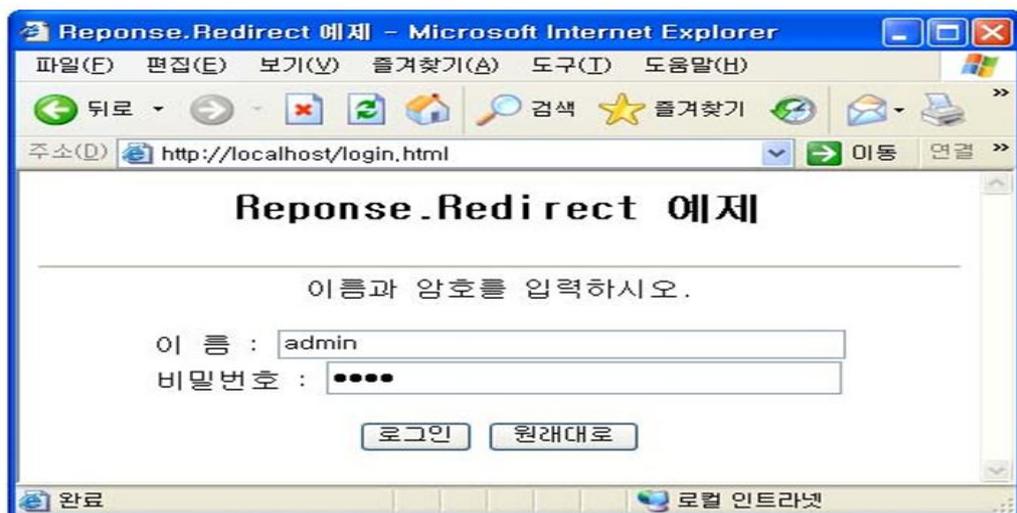
```
<%  
  Reponse.Redirect ("이동할 문서 또는 URL 주소")  
%>
```

- 예제 : login.html

```
1: <HTML>  
2: <HEAD><TITLE> Reponse.Redirect 예제 </TITLE></HEAD>  
3: <BODY>  
4:   <CENTER>  
5:     <H2> Reponse.Redirect 예제 </H2>  
6:     <HR>
```

```
7:     이름과 암호를 입력하십시오.  
8:  
9:     <FORM ACTION="loginresult.asp" METHOD="post">  
10:       이 름 : <INPUT TYPE="Text" NAME="name" SIZE="40"> <BR>  
11:       비밀번호 : <INPUT TYPE="Password" NAME="pass" SIZE="40"> <P>  
12:       <INPUT TYPE="Submit" VALUE="로그인">  
13:       <INPUT TYPE="RESET" VALUE="원래대로">  
14:     </FORM>  
15:   </CENTER>  
16: </BODY>  
17: </HTML>
```

- 예제 결과



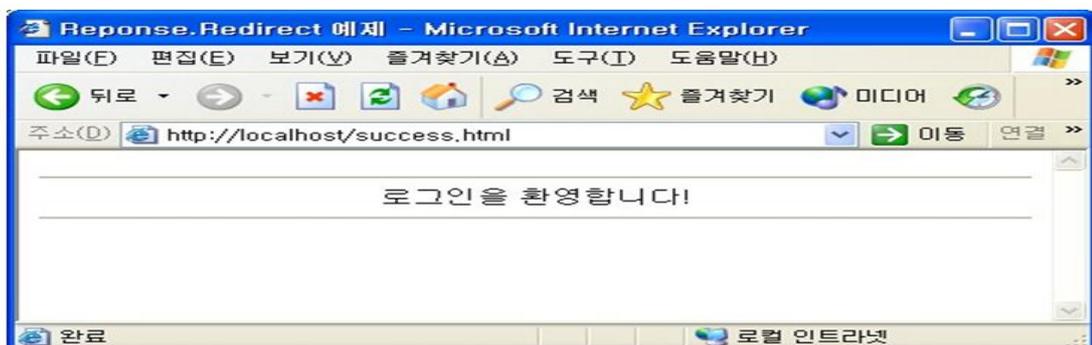
- 예제 : loginresult.asp

```
1 : <%
2 :
3 :   If ( (request.form("name")="admin") and
4 :         (request.form("pass")="1234") ) Then
5 :     Response.Redirect("success.html")
6 :
7 :   Else
8 :
9 :     Response.Redirect("fail.html")
10 :
11 :   End if
12 : %>
```

- 예제 : success.html - 로그인을 성공적으로 수행한 경우

```
1 : <HTML>
2 :   <HEAD><TITLE> Reponse.Redirect 예제 </TITLE></HEAD>
3 :   <BODY>
4 :     <HR><CENTER>
5 :
6 :       로그인을 환영합니다!<BR>
7 :
8 :     </CENTER> <HR>
9 :   </BODY>
10: </HTML>
```

- 예제 결과



#### 다. ADO (ActiveX Data Object) 객체

ADO 객체는 데이터베이스 프로그래밍에 자주 사용되는 객체이다. 데이터베이스에 접속하여 각종 기능을 처리하기 위한 메소드와 속성을 제공한다. 주로 사용되는 ADO 객체로는 connection, command, recordset 등이 있다.

- Connection 객체
  - 데이터 소스에 대한 연결 방법을 제공
- 예제 : ADO에서 connection 객체의 생성

```
1: Set conn_obj = Server.CreateObject("ADODB.Connection")
2:     conn_obj.Open "DSN=board_dsn;UID=;Password="
```

- Connection 객체의 메소드

메소드	내용
Open	데이터베이스의 연결을 시작한다.
Close	데이터베이스의 연결을 종료한다.

- Command 객체
  - 데이터베이스 내용에 대하여 실행되는 명령어 (SQL 명령어)
- 예제 : ADO에서 command 객체의 생성

```
1: Set cmd_obj = Server.CreateObject("ADODB.Command")
2:     cmd_obj.CommandText = "Select * from board_tbl"
3:     cmd_obj.CommandType = adCmdText
4:     cmd_obj.ActiveConnection = conn_obj
```

▪ Command 객체의 속성

속성	내용
ActiveConnection	Command 객체가 속해 있는 Connection 객체를 명시
CommandText	실행할 명령을 나타내는 텍스트 스트링
CommandType	CommandText의 형식을 지정 (본 책에서는 SQL문을 지정)

▪ Command 객체의 메소드

메소드	내용
Execute	CommandText 속성에 저장되어 있는 명령을 실행

▪ Recordset 객체

- 데이터베이스의 데이터를 처리하기 위해 사용되는 객체
- 데이터베이스의 데이터에 대한 삽입, 삭제, 수정 등의 연산 수행

▪ Recordset 객체의 속성

속성	내용
ActiveConnection	레코드셋이 사용하고 있는 Connection 객체
BOF, EOF	파일의 시작, 파일의 끝
CursorType	레코드셋에서 사용하는 커서의 종류
LockType	편집 중 레코드에 지정된 락의 종류
RecordCount	현재 레코드셋에 있는 레코드의 수

▪ Recordset 객체의 메소드

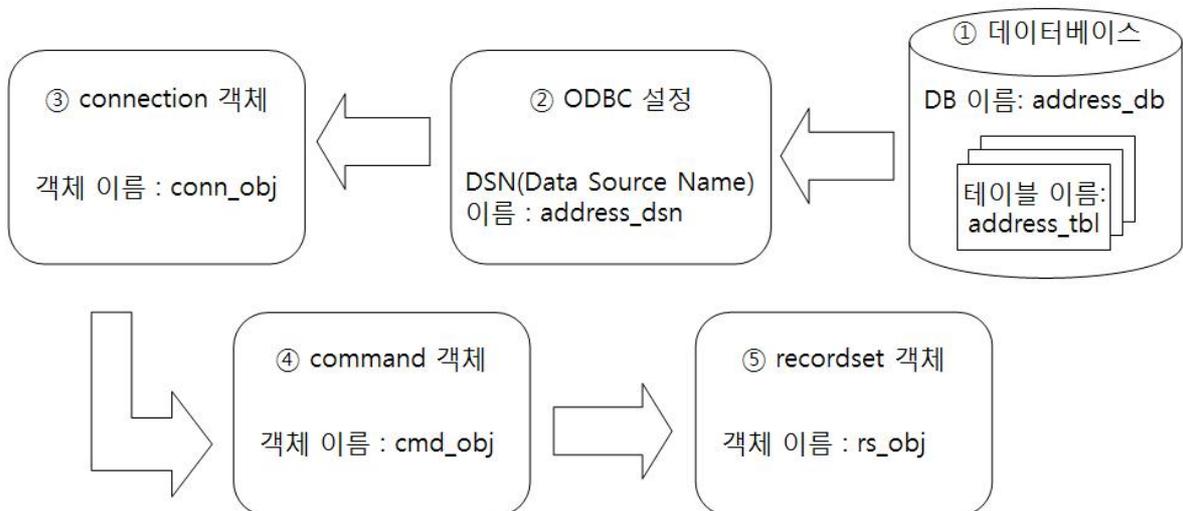
메소드	내용
AddNew	레코드셋 내에 새로운 레코드 생성
Close	레코드셋 닫기
Delete	현재 레코드 삭제
MoveFirst	현재 레코드의 위치를 처음으로 이동
MoveLast	현재 레코드의 위치를 마지막으로 이동
MoveNext	현재 레코드의 위치를 다음으로 이동
MovePrevious	현재 레코드의 위치를 이전으로 이동
Open	레코드셋 열기
Update	현재 레코드의 수정 사항을 저장

▪ 예제 : ADO에서 recordset 객체의 생성

```

1: Set rs_obj = Server.CreateObject("ADODB.Recordset")
2:     rs_obj.Open cmd_obj, , adOpenStatic, adLockPessimistic
    
```

라. 데이터베이스 활용 방법



- (1) 그림의 ①은 데이터베이스를 나타낸 것으로 DB와 테이블 이름으로 각각 address\_db, address\_tbl을 사용한다.
- (2) 그림의 ②는 데이터베이스와의 연결을 위해 ODBC를 설정하는 과정으로 DSN의 이름은 address\_dsn을 사용한다.
- (3) 그림의 ③은 connection 객체를 생성하여 데이터베이스에 연결하는 과정으로 객체 이름은 conn\_obj를 사용한다.
- (4) 그림의 ④는 command 객체를 생성하여 SQL 문을 저장하며, 이를 connection 객체와 연결하는 과정으로 객체 이름은 cmd\_obj를 사용한다.
- (5) 그림의 ⑤는 recordset 객체를 생성하여 저장된 SQL 문을 실행하고, 데이터베이스로부터 원하는 질의 결과를 얻는 과정을 나타내며 객체 이름은 rs\_obj를 사용한다.

#### 데이터베이스 연결 및 실행 절차를 나타낸 ASP 코드

```

1: ' Adovbs.inc 파일 포함
2: <!-- #include file="adovbs.inc" -->
3:
4: ' 데이터베이스 연결
5: Set conn_obj = Server.CreateObject("ADODB.Connection")
6:     conn_obj.Open "DSN=address_dsn;UID=;Password="
7:
8: ' SQL 명령어 지정
9: Set cmd_obj = Server.CreateObject("ADODB.Command")
10:     cmd_obj.CommandText = "Select * from address_tbl"
11:     cmd_obj.CommandType = adCmdText
12:     cmd_obj.ActiveConnection = conn_obj
13:
14: ' SQL문 실행
15: Set rs_obj = Server.CreateObject("ADODB.Recordset")
16:     rs_obj.Open cmd_obj, , adOpenStatic, adLockPessimistic

```

- 2행은 외부 파일을 불러 사용하는 부분으로 ADO용 상수를 정의해놓은 Adovbs.inc라는 파일을 포함하는데, 이는 11행에서 사용되는 adCmdText를 '&H0001'으로 정의함으로써 데이터베이스 명령어가 SQL 문임을 지정하는 것
- 이는 또한 16행의 adOpenStatic과 adLockPessimistic은 cursor type과 lock type을 지정한 것임
- 5행에서는 connection 객체를 생성하며, 6행에서는 데이터베이스에 DSN을 이용하여 접속
- 9행에서는 command 객체를 생성하게 되고, 10행에서는 CommandText 속성에 수행할 SQL 명령어를 지정
- 10행의 SQL 명령어를 이용하여 address\_tbl 테이블에 있는 데이터를 가져옴
- 11행은 앞서 설명하였듯이 명령어를 SQL 문으로 지정하는 부분임
- 12행에서는 command 객체의 ActiveConnection 속성을 이용하여 connection 객체를 지정
- 15행에서는 recordset 객체를 생성하며, 16행에서는 15행에서 지정한 rs\_obj 레코드셋에 대하여 cmd\_obj에 지정한 SQL 명령을 실행
- adovbs.inc

```

<%
Const adCmdText = &H0001

'---- CursorTypeEnum Values ----
Const adOpenForwardOnly = 0
Const adOpenKeyset = 1
Const adOpenDynamic = 2
Const adOpenStatic = 3

'---- LockTypeEnum Values ----
Const adLockReadOnly = 1
Const adLockPessimistic = 2
Const adLockOptimistic = 3
Const adLockBatchOptimistic = 4
%>

```

마. 예제

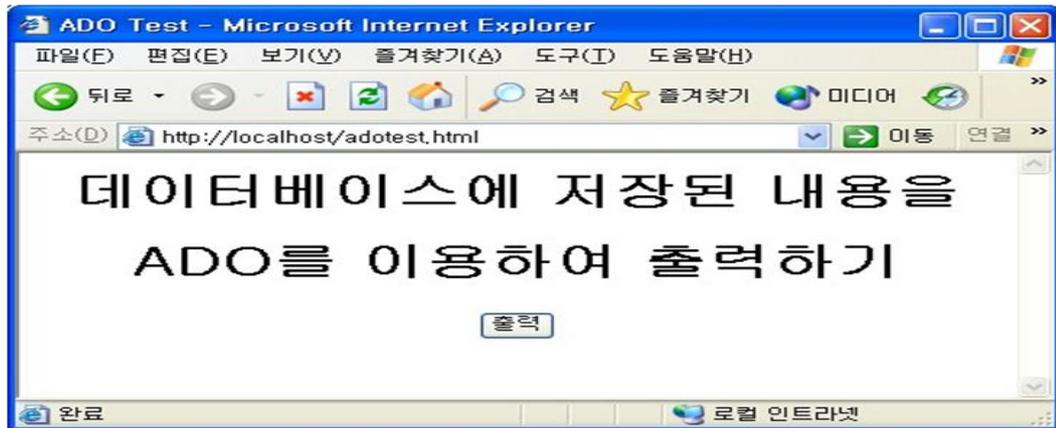
- 본 예제에서 사용될 데이터베이스와 테이블명
  - 데이터베이스명 : address\_db
  - 테이블명 : address\_tbl(필드명 : Name, E-mail, Address)
  - ODBC 설정시 DSN명 : address\_dsn
  
- address\_tbl 테이블의 구성

필드 명	데이터 타입	설명
Name	Varchar	이름
E-mail	Varchar	전자 우편
Address	Varchar	주소

- 필요한 화면 및 해당 파일명
  - 초기 화면: adotest.html
  - 데이터베이스에 저장된 내용을 출력하는 화면: adotest.asp
  
- 예제: adotest.html

```
1: <HTML>
2: <HEAD> <TITLE> ADO Test </TITLE> </HEAD>
3: <BODY>
4:   <CENTER>
5:     <H1>
6:       데이터베이스에 저장된 내용을 <P>ADO를 이용하여 출력하기
7:     </H1>
8:     <P>
9:       <FORM ACTION="adotest.asp" method="post">
10:        <INPUT TYPE="submit" Value="출력">
11:      </FORM>
12:    </CENTER>
13:  </BODY>
14: </HTML>
```

- 예제 결과



- 예제 : adotest.asp

```

1: <HTML>
2: <!-- #include file="adovbs.inc" -->
3: <HEAD><TITLE> ADOTest </TITLE></HEAD>
4: <BODY><CENTER><P><BR>
5: <H1> 출력 결과 </H1>
6: <%
7:     Set conn_obj = Server.CreateObject("ADODB.Connection")
8:     conn_obj.Open "DSN=address_dsn;UID=;Password="
9:     Set cmd_obj = Server.CreateObject("ADODB.Command")
10:    cmd_obj.CommandText = "SELECT * FROM address"
11:    cmd_obj.CommandType = adCmdText
12:    cmd_obj.ActiveConnection = conn_obj
13:    Set rs_obj = Server.CreateObject("ADODB.Recordset")
14:    rs_obj.Open cmd_obj, , adOpenStatic, adLockPessimistic
15: %>

```

```

16: <TABLE border=1>
17: <TR>
18:   <TD align="center">Name</TD>
19:   <TD align="center">E-mail</TD>
20:   <TD align="center">Address</TD>
21: </TR>
22:
23: <% Do While Not rs_obj.EOF %>
24: <TR>
25:   <TD align="center"> <%=rs_obj("Name")%> </TD>

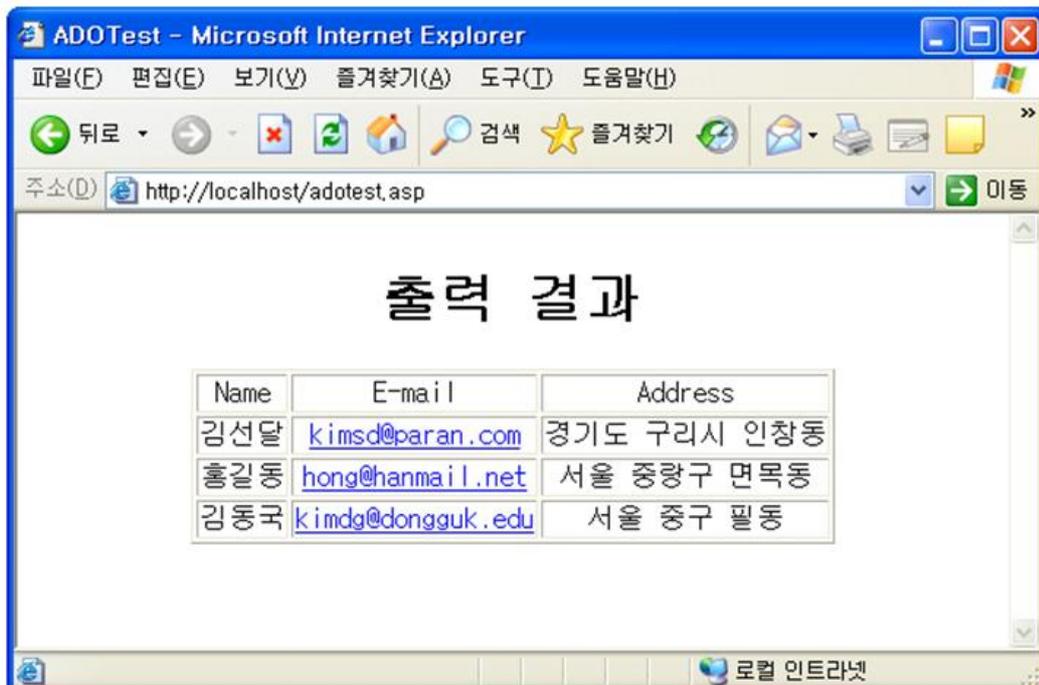
```

```

26: <TD align="center"> <a href="<%= rs_obj("E-mail")%>"> <%=
rs_obj("E-mail")%> </a> </TD>
27: <TD align="center"> <%= rs_obj("Address")%> </TD>
28: </TR>
29: <%
30:     rs_obj.MoveNext
31:     Loop
32: %>
33: </TABLE>
34: </BODY>
35: </HTML>

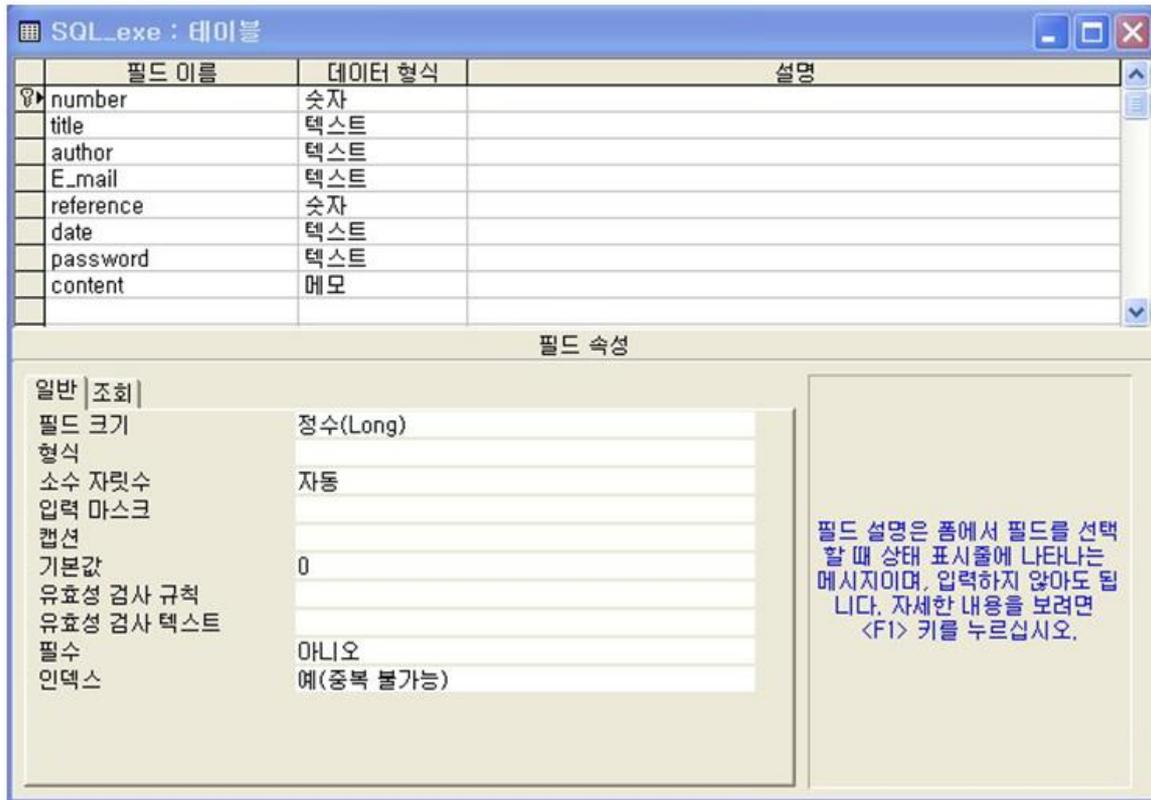
```

▪ 예제 결과



## 6.3 SQL 프로그래밍

다음 기본 테이블을 바탕으로 SQL 명령어를 살펴보자.



### 6.3.1 Insert문

- 기존의 테이블에 새로운 행을 삽입하는 것

형식 1 :

```
Insert Into 테이블(열 이름 리스트)
Values (열 값 리스트)
```

형식 2 :

```
Insert Into 테이블(열 이름 리스트)
Select문
```

- Insert 문의 예

```

INSERT INTO SQL_exe
VALUES (1, "반가워요", "홍길동", "hong@hanmail.net", 4, "05/01/14", "1111", "반갑습니다!!");
  
```

- 데이터를 입력한 결과

	number	title	author	E_mail	reference	date	
▶	1	반가워요	홍길동	hong@hanmail.	4	05/01/14	11
	2	안녕?	김동국	kimdg@donggu	20	05/02/03	00
	3	여러분	관리자	admin@donggu	10	05/02/10	22
	4	날씨가 좋네요	홍길동	hong@hanmail.	16	05/02/25	33
	5	재미있는 이야기	홍길동	hong@hanmail.	38	05/03/07	77
	6	공부합시다	김선달	kimsd@paran.c	24	05/04/02	56

레코드: 1 전체: 6

### 6.3.2 Select문

- 테이블의 내용을 검색하는 데에 이용

형 식 :

Select [All | Distinct] 열 리스트

From 테이블

[Where 조건]

[Group by 열 리스트]

[Order by 열 리스트 [Asc | Desc]]

- 열 전부를 검색
  - 테이블의 내용 전부를 볼 수 있음

<질 의>



<결 과>

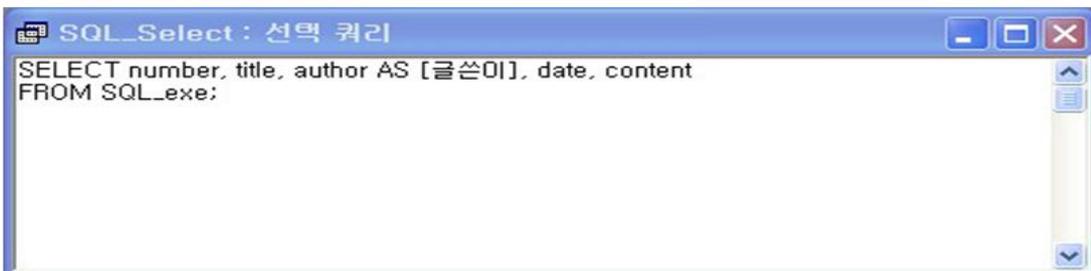
SQL\_Select : 선택 쿼리

	number	title	author	E_mail
▶	1	반가워요	홍길동	hong@hanmail.
	2	안녕?	김동국	kimdg@donggu
	3	여러분	관리자	admin@donggu
	4	날씨가 좋네요	홍길동	hong@hanmail.
	5	재미있는 이야기	홍길동	hong@hanmail.
	6	공부합시다	김선달	kimsd@paran.c

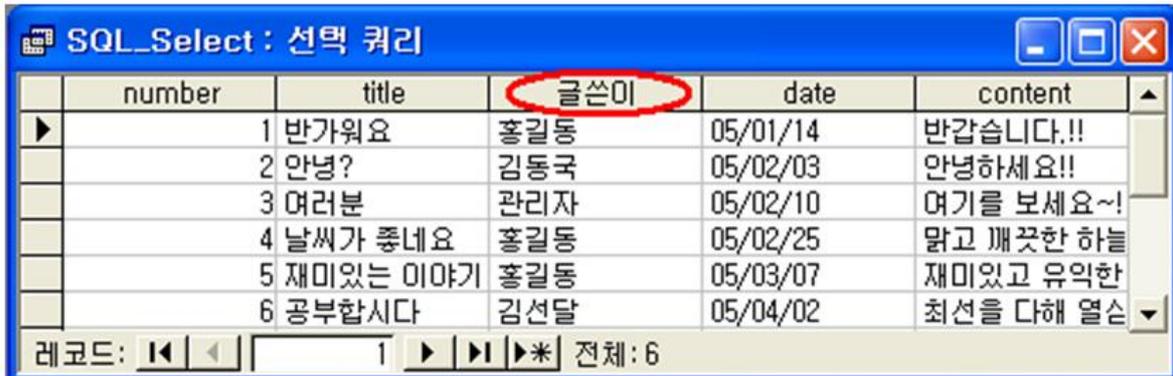
레코드: 1 전체: 6

- 열 이름 바꾸어서 출력
  - 열 이름을 바꾸어서 출력

<질 의>



<결과>



number	title	글쓴이	date	content
1	반가워요	홍길동	05/01/14	반갑습니다!!
2	안녕?	김동국	05/02/03	안녕하세요!!
3	여러분	관리자	05/02/10	여기를 보세요~!
4	날씨가 좋네요	홍길동	05/02/25	맑고 깨끗한 하늘
5	재미있는 이야기	홍길동	05/03/07	재미있고 유익한
6	공부합시다	김선달	05/04/02	최선을 다해 열심

- Distinct를 이용한 검색
  - 사용 안 했을 때

<질의>



```
SELECT author
FROM SQL_exe;
```

<결과>



author
홍길동
김동국
관리자
홍길동
홍길동
김선달

- 사용 했을 때 : 중복된 결과를 한번만 보여줌

<질 의>



<결 과>

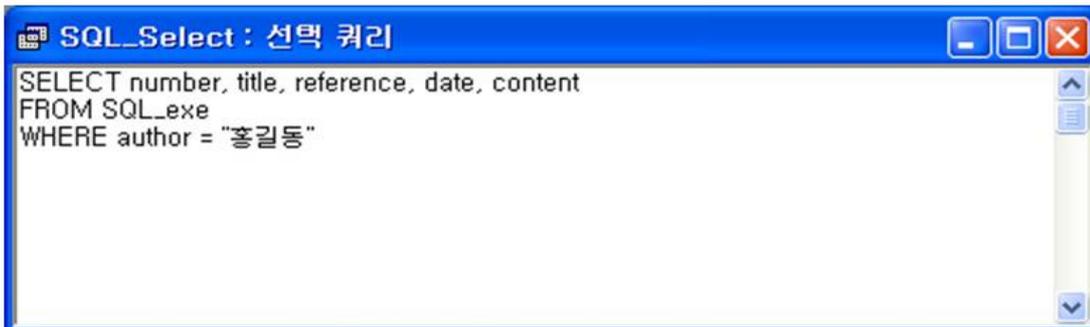


- Where절을 이용한 조건 검색
  - Where절
    - 검색을 할 때 보다 더 자세히 검색하고 싶을 때 조건을 제시하는 절

비교 연산자	의미
=	같다.
≠	같지않다.
<	작다.
>	크다.
≤	작거나 같다.
≥	크거나 같다.

논리 연산자	의미
Not	부정
And	논리곱
Or	논리합

<질 의>



```

SQL_Select : 선택 쿼리
SELECT number, title, reference, date, content
FROM SQL_exe
WHERE author = "홍길동"
  
```

<결 과>



number	title	reference	date	content
1	반가워요	4	05/01/14	반갑습니다!!
4	날씨가 좋네요	16	05/02/25	맑고 깨끗한 하늘
5	재미있는 이야기	38	05/03/07	재미있고 유익한

레코드: 4 전체: 4

### 6.3.3 Update문

- 기존 테이블 안의 내용을 수정하는 것

형 식 :

Update 테이블

Set 열 이름 = 새로운 값

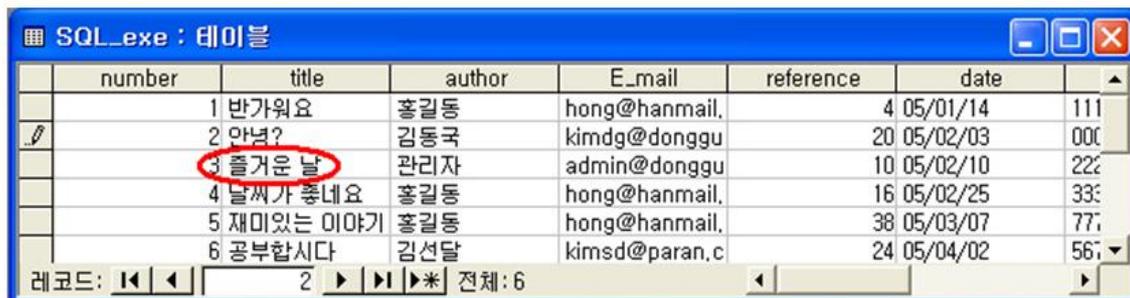
[Where 조건]

<질 의>



```
UPDATE SQL_exe
SET title = "즐거운 날"
WHERE number = 3
```

<결 과>



number	title	author	E_mail	reference	date	
1	반가워요	홍길동	hong@hanmail,	4	05/01/14	111
2	안녕?	김동국	kimdg@donggu	20	05/02/03	000
3	즐거운 날	관리자	admin@donggu	10	05/02/10	222
4	날씨가 좋네요	홍길동	hong@hanmail,	16	05/02/25	333
5	재미있는 이야기	홍길동	hong@hanmail,	38	05/03/07	777
6	공부합시다	김선달	kimsd@paran.c	24	05/04/02	555

레코드: 2 전체: 6

#### 6.4.4 Delete문

- 기존의 테이블에서 행을 삭제하는 것

형 식 :  
Delete  
From 테이블  
[Where 조건]

#### <질 의>



```
DELETE
FROM SQL_exe
WHERE number = 1
```

#### <결 과>



	number	title	author	E_mail	reference	date
	2	안녕?	김동국	kimdg@donggu	20	05/02/03
	3	즐거운 날	관리자	admin@donggu	10	05/02/10
	4	날씨가 좋네요	홍길동	hong@hanmail,	16	05/02/25
	5	재미있는 이야기	홍길동	hong@hanmail,	38	05/03/07
	6	공부합시다	김선달	kimsd@paran.c	24	05/04/02

레코드: 2 전체: 5

## 제 7 장. 모바일 웹 프로그래밍 개요

### 7.1 무선 인터넷의 이해

#### ▪ 무선 인터넷이란

무선 인터넷이란 무선 인터넷이 되는 무선 장비를 통해 이동통신망을 이용하여 인터넷 접속하는 것을 말합니다.

무선 인터넷은 이동(Mobile)+무선(Wireless)의 뜻을 가지고 있습니다.

흔히 말하는 모바일(Mobile) 장비로는 휴대폰 만을 떠올리는 경우가 많은데 모바일은 휴대가 가능하고 무선인터넷이 가능한 PDA, 노트북 등의 장비들이 포함됩니다. 즉 무선 단말기로 이동 중에 무선망을 통하여 인터넷 서비스에 액세스하고 음성/데이터/영상정보 등 정보를 제공받을 수 있도록 하는 환경과 기술을 말합니다.

#### ▪ 무선 인터넷의 특징

- 편재성 (Ubiquity) : 어디서나 사용 가능
- 접근성 (Accessibility) : 언제 어디서나 인터넷 접속 가능
- 보안성 (Security) : 개인 휴대로 인한 보안 가능
- 편리성 (Convenience) : 손쉽게 사용 가능
- 위치성(Localization) : 사용자의 위치를 검색
- 즉시 접속성(Instant Connectivity) : 빠른 시간 내에 정보 찾을 수 있음
- 개인성(Personality) : 이동통신 사용자의 개인화와 차별화된 고객 서비스



- 호환성 : 서로 다른 단말기와 플랫폼들 사이에서도 네트워크 상에서 모든 서비스를 이용
- 효율성 : 가능한 많은 사용자에게 가능한 많은 양의 서비스를 제공
- 신뢰성 : 안정적인 접속과 신뢰성 있는 서비스를 제공
- 안전성 : 보안상의 문제로부터 보호

무선 인터넷을 통해 제공되는 서비스로는 다음이 있다.

- 정보 서비스 : 모바일 banking, 모바일 쇼핑, 광고 등 정보 제공 서비스



- 엔터테인먼트 서비스 : 모바일 게임, 모바일 멜로디 다운로드, 모바일 채팅 등 커뮤니티에 관련된 서비스



- 모바일 포털 : 유무선 통합 콘텐츠, 메신저 서비스, 전자우편 서비스 등



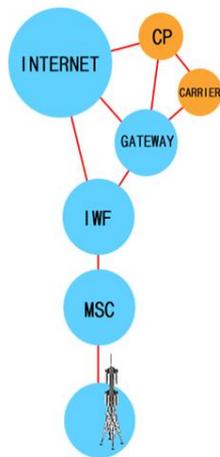
무선 인터넷의 산업구조를 살펴보면, 크게 다음 4 가지의 사업자 유형이 있다.

▪ 단말기 관련 제조업체

▪휴대폰, PDA 등 모바일 장비들을 제조하는 업체입니다. 현재 무선 인터넷 환경에 맞추기 위해 이동통신 사업자와 솔루션 업체 및 CP과의 협조가 필요합니다.

▪ 이동통신 사업자(Carrier)

▪무선 인터넷 서비스를 제공하는 SKT, KTF, LGT 통신사를 말합니다. 휴대폰으로 무선 인터넷 접속하면 나오는 메뉴를 통해 서비스를 제공합니다.



IWF(Inter-Working Function) : 망 연동 장치  
 MSC(Mobile Switching Center) : 전체 시스템을 통제하는 이동통신 시스템

▪ 콘텐츠 제공 사업자(CP)

▪컨텐츠를 제작 및 제공하는 사업자로 흔히 CP(Contents Provider)라고 부릅니다. 제작한 컨텐츠를 이동통신 사업자와 제휴를 통해 서비스를 제공하는 것이 일반적입니다. 이때 이동통신 사업자와 수익배분을 통해 이익을 나누는 방식을 취하고 있습니다. 무선망 개방에 따라 이동통신 사업자를 거치지 않고 직접 운영하기도 합니다. 이때는 숫자만 입력하면 손쉽게 접속할 수 있는 모바일 인터넷 주소 WINC 서비스를 이용하여 서비스를 제공합니다.

▪ 솔루션 업체

▪무선 인터넷 서비스에 필요한 각종 시스템 애플리케이션 기술을 개발합니다. 유 무선 통신, 인터넷 및 컨텐츠들이 융합되는 분야로서 업체 간의 기술협력과 아이디어 교류가 필요합니다. 이동통신 사업자나 CP가 주 고객입니다.

## 7.2 무선 인터넷 모델

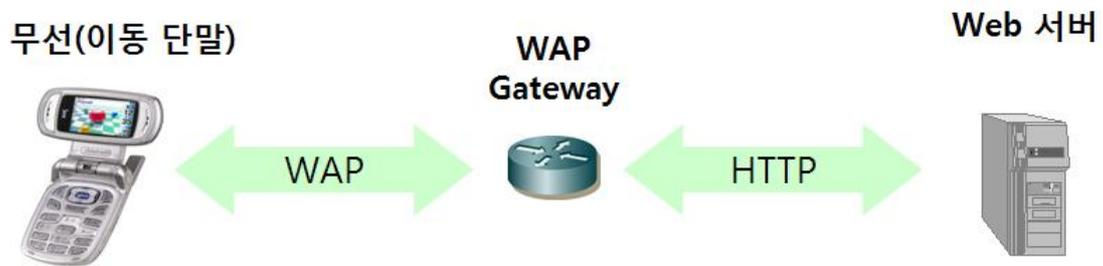
현재 제공되고 있는 무선 인터넷 구조로는 다음 2 가지 모델이 있다.

### •WAP 모델

WAP의 구조는 홈페이지 프로토콜인 HTTP와 유사하지만 휴대 단말기(Client)와 인터넷 서버 사이에 WAP Gateway가 존재합니다.

WAP Gateway은 WAP 프로토콜과 인터넷 TCP/IP (HTTP) 프로토콜을 중간에서 변환해줍니다. 즉, 모든 휴대 단말기의 인터넷 서비스 요구는 WAP Gateway를 거치도록 되어 있고, Gateway는 WAP 프로토콜에 따라 요청 받은 서비스를 기존 인터넷 유선 망을 통해 다시 서비스를 요청합니다.

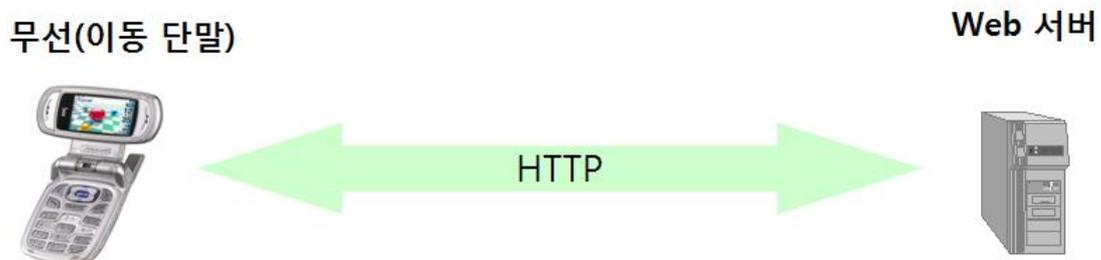
이어서 Gateway가 인터넷 서버로부터 응답을 받고 다시 서비스를 최초 요청했던 휴대 단말기에게 WAP 프로토콜로 전송 함으로서 모든 과정이 이루어집니다.



### ▪ MME 모델

MME (Microsoft Mobile Explorer)는 마이크로소프트사에서 WAP 구조를 따르지 않고 HTTP 프로토콜의 기존 인터넷 서비스 환경을 그대로 이용해서 사용하는 서비스입니다. 서버와 클라이언트 사이에 Gateway가 필요하지 않으므로 WAP 모델보다 구축 비용이 저렴하고 HTML의 subset인 m-HTML을 기본 언어로 사용하고 있어서 확장성이 좋은 편입니다.

브라우저는 mHTML을 지원하는 ME(Mobile Explorer)을 사용하고 있습니다. 우리나라에서는 016, 018에서 ME (mHTML)를 사용해 서비스하고 있습니다.



### 7.3 휴대폰 종류

초창기 휴대폰은 아날로그 셀룰러 방식으로 음성 통신 위주의 기능을 제공하였습니다.

이후에 디지털 셀룰러(cellular) 방식의 휴대폰으로 바뀌면서 음성과 데이터를 함께 전송하는 2 세대 휴대폰이 등장하게 되었습니다. 그리고 PCS 휴대폰이 등장하면서 휴대폰의 보급이 급진적으로 이루어 졌으며 CDMA 기술에 힘입어 음악과 컬러 그래픽 등 멀티미디어를 부분적으로 지원하는 휴대폰이 탄생되었습니다.

CDMA2000을 지원하는 휴대폰을 2.5 세대 휴대폰이라고 부르기도 합니다. 요즘에는 다양한 모양을 갖추고 있거나 전화기 기능 이외에 PDA 기능을 가지고 있는 휴대폰, 디지털 카메라가 내장되어 있는 휴대폰, MP3 음악을 연주하는 휴대폰 등이 등장하고 있습니다.



#### CDMA2000

CDMA (Code Division Multiple Access)는 각 채널의 신호에 부호화 처리를 하여 구별할 수 있도록 다원 접속을 실현하는 코드 분할 다중 접속방식을 말합니다.

CDMA2000은 기존의 휴대폰보다 데이터 전송 속도가 10배 정도 빠른 144 Kbps 속도를 제공하므로 애니메이션이나 칼라 등의 멀티미디어를 실행하기에 용이합니다.

인터넷 요금도 기존의 휴대폰은 인터넷 접속한 시간으로 계산했던 것에 비해 다운로드 받은 데이터 크기로 요금이 책정 방식을 사용하기 때문에 보다 유용하게 인터넷을 사용할 수 있습니다.



동영상 재생 및 3D 게임이 가능한 멀티미디어 휴대폰

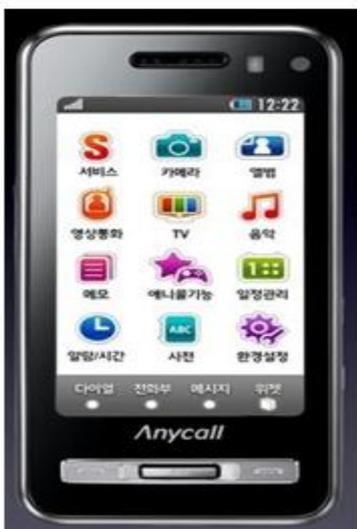
## IMT-2000

IMT (International Mobile Telecommunication) 2000은 ITU-T 국제표준기구에서 제안한 표준 기술이 채택된 차세대 이동통신 서비스를 말합니다. 기존의 휴대폰이 1.25MHz 채널대역폭을 사용하여 64Kbps 속도로 데이터 전송이 실행되지만 IMT-2000은 5MHz의 채널대역폭을 가지고 있고 전세계적 표준화 및 동일 주파수를 활용하여 세계 어디서든지 사용이 가능하며 최대 2Mbps 속도의 데이터전송이 가능하여 음성, 데이터 등 멀티미디어 정보의 전송할 수 있습니다. 앞으로 휴대폰을 통해서 영화를 보고 즐길 수 있는 시대가 열리게 됩니다. 그래서 IMT-2000을 3세대 서비스라고 부릅니다. 상용화가 지연되고 있는 방식입니다. IMT2000은 동기식 방식인 CDMA2000과 비동기 방식인 W-CDMA로 구분할 수 있습니다. CDMA2000 방식은 미국, 캐나다를 중심으로 많이 사용하고 있는 방식이고 WCDMA는 유럽에서 많이 사용하는 방식입니다.



### <터치폰(Touch Phone) & 풀브라우징(full browsing)>

지난해 휴대폰 시장에는 삼성전자의 햅틱폰 시리즈와 LG전자의 뷰티폰, 프라다폰 후속모델 등이 선보이면서 고기능 휴대폰 시장을 이끌었다. 이 제품들은 넓은 LCD 화면과 터치 기능, 향상된 사용편의성 등이 특징으로, 음악, 동영상 감상 등의 다양한 멀티미디어 이용이 용이해졌다. 무엇보다 이통사들이 이전의 단순한 화면과 느린 속도의 인터넷 환경을 개선시킴으로써 이용률을 더욱 높이고자 했다.



<차세대 휴대폰 UI>

인간과 교감...살아숨쉬는 폰이 온다

'단순한 터치폰은 가라. 더욱 쉽고 역동적으로 인간과 교감하는 휴대폰이 온다!'

플 터치스크린 폰이 확대되면서 사람의 오감(五感)을 자극할 수 있는 사용자환경(UI)을 만드는 것이 화두다. 보고 듣고 만지는 것은 이제 어느 수준까지는 올라왔다. 향후 맛과 냄새를 어떻게 상용화 할 수 있을 지가 관건이다.



▶삼성, 입체이고 역동적인 화면 제공...진화된 3D UI 공개 = 지난해 사용자 감성을 자극하는 혁신적인 UI인 햅틱 UI(글로벌하게는 터치위즈 UI)를 장착한 햅틱폰으로 국내외 플터치스크린 시장 트렌드를 주도했던 삼성전자는 터치스크린 UI 주도권을 이어가기 위해 한단계 진화한 UI를 선보인다.

삼성은 메뉴나 화면 이동 등에 다양한 3D 그래픽을 채용하는 '3D UI'를 공개한다. 3D UI는 1차원 UI에서 한발 더 나아가, 실제와 유사한 입체이고 역동적인 화면을 제공한다. 삼성은 플터치폰 뿐아니라 제품마다 특화된 UI도 공개할 것으로 알려졌다. 지난해 좋은 반응을 보인 워젯기능도 날씨, 주식, 스포츠 등 다양한 정보를 온라인을 통해 추가 가능하도록 할 방침이다.

▶LG, 애플 UI와 삼성 햅틱UI 장점 벤치마킹, 야심찬 신규 UI준비= LG전자는 애플 아이폰과 삼성의 햅틱 UI의 장점을 뽑아낸 신규 UI를 개발,공개한다. 애플 아이폰처럼 멀티터치가 가능하고 다양한 대기화면, 모션 UI 등이 가능한 것으로 알려졌다.

특히 LG가 선보일 UI는 애플 아이폰과 같은 정전용량 방식을 채택, 두 손가락을 이용한 멀티터치가 가능하다. 이를 통해 사진을 확대하거나 축소하는 등의 두 손가락을 통한 다양한 기능을 실행할수 있다.

## 7.4 모바일 플랫폼

▪ 플랫폼(Platform)이란 어떤 환경에서 사용할 수 있도록 해주는 프로그램을 말합니다. 예를 들어 윈도우 환경에서 워드 작업을 하기 위해서 한글 프로그램을 설치했다면 한글 프로그램의 플랫폼은 윈도우가 됩니다. 무선 인터넷에서는 ME나 WAP으로는 다양한 멀티미디어를 실행할 수 없으므로 멀티미디어 서비스를 이용할 수 있도록 해주는 플랫폼이 필요합니다. 그래서 VM이라는 플랫폼이 필요하게 되었습니다.

▪ VM(Virture Machine)란 사용자가 서버에서 필요한 컨텐츠나 응용프로그램을 다운로드 받아 휴대폰으로 구동할 수 있는 플랫폼 기술을 말합니다. VM은 필요한 데이터를 다운로드받아서 사용하는 플러그인 형태와 프로그램 전체를 실행 파일로 제작하는 방식으로 나누어 볼 수 있습니다. 플랫폼은 다양한 장비와 프로그램과 호환성을 유지해야 하며 안정적이고 실행 속도가 빨라야 합니다.

### ▪ VM의 종류

#### •XVM

▪XVM (eXtended Virtual Machine)이란 모바일용 자바(J2ME) 기반으로 개발한 VM입니다. 자바의 단점인 속도 문제를 개선하고 자바의 장점을 부각하여 개발하였으며 게임전용으로 개발된 KVM을 이용해서 KKVM 전용 게임들을 개발하여 SKT와 공동 개발 계약을 체결하였습니다. SKT와 공동 개발한 SKVM을 이용한 nTOP 서비스를 시행하고 있습니다. 사운드, 이미지, 백라이트, 진동 등 멀티미디어 기능이 강력합니다.

#### •BREW

▪BREW (Binary Runtime Environment for Wireless)이란 미국 퀄컴에서 제작한 CDMA 용으로 개발된 플랫폼으로 J2ME보다 빠르면서 C,C++도 지원합니다. 단말기에 새로운 기능을 추가할 경우 작업 시간을 단축시킬 수 있으며 음성 제어도 가능합니다. KTF는 BREW를 이용하여 매직앤 멀티팩 서비스를 제공하고 있습니다.

### ▪ VM의 종류

#### •WIPI

▪ WIPI(Wireless Internet Platform for Interoperability)는 이동통신 업체들이 같은 플랫폼을 사용하도록 하기 위해서 국가 정책 사업의 일환으로 SK텔레콤(주)·KTF(주)·LG텔레콤(주) 등 이동통신 3개사와 한국무선인터넷표준화포럼(KWISF)이 공동으로 처음 개발되었습니다. 한국 표준 플랫폼으로 사용자가 계속으로 늘 것으로 전망되는 VM입니다.

## 7.5 모바일 무선인터넷 시뮬레이터

이 절에서는 본 교재에서 다루는 모바일 무선인터넷 시뮬레이터의 설치 안내를 제공한다. 본 교재에서 예제로 다루는 무선인터넷 시뮬레이터는 다음과 같다.

- ① KTF ME1.3b: m-HTML 문서 실습용
- ② Openwave 6/7: WML 1 혹은 2 문서 실습용
- ③ KTF KUN
- ④ UP

상기 시뮬레이터 코드는 다음 경로에서 받을 수 있다.

- <http://protocol.knu.ac.kr/lecture/wpd/simulators>

Openwave 시뮬레이터를 제외하면 모두 압축을 풀어 바로 실행할 수 있게 되어 있다.

아래 예제는 상기한 시뮬레이터가 설치되어 있는 상황에서, mHTML 및 WML 문서에 대한 예제이다. 여기서는 편의상 155.230.105.167 서버에서 실행한 문서 결과를 보여준다.

보다 상세한 내용은 다음 사이트의 문서를 참조하기 바란다.

- <http://protocol.knu.ac.kr/report.html>
- 문서번호: CPL-TR-09-02

### 7.5.1 KTF ME1.3b 시뮬레이터

ME1.3b 는 mHTML 을 지원하는 브라우저이다.

ME1.3b 브라우저로 아래 사이트에 접속하면 다음과 같은 화면을 볼 수 있다.

- ▷ <http://155.230.105.167/mobile/asp/bloodtype/mhtml/index.asp>

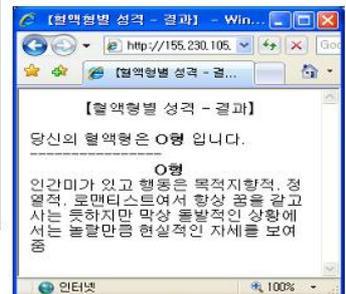
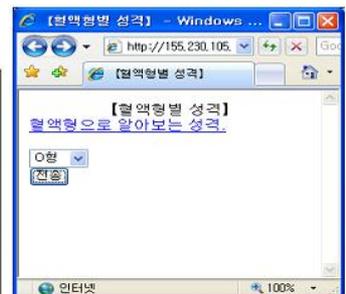
또한 mHTML 은 HTML 의 부분집합이므로 Internet Explorer 로도 볼 수 있다.



```

Mobile Explorer 121c-a
File Test Help
Open URL
URL: /Lecture/mobile/bloodtype/mhtml/bloodtype_input.asp
OK Cancel

S: <meta name="mobile_browser" co
S: </head>Wr
S: <body>Wr
S: <div><center><a href> [혈액형별
S: <div><a href="#">혈액형 입력
S: </div>Wr
S: --> Loading document (type='text/html;cha
S: Title = [혈액형별 성격]
S: or name="form_bloodtype" method="post"
S: ture/mobile/bloodtype/mhtml/bloodtype_
S: <div><select name="bloodty
S: <option value="A" >A형
S: <option value="B" >B형
S: <option value="O" >O형
S: <option value="AB" >AB
S: </select></div>Wr
S: <div><input type="submit"
S: </form>Wr
S: <btn name="상위" href="http://
S: 07">Wr
S: </body>Wr
S: </html>Wr
S: Wr
S: *EOF*
  
```



▲ KTF ME 브라우저로 확인한 결과.  
(주소 입력시 [http://site\\_name/mhtml/xxx.asp](http://site_name/mhtml/xxx.asp) 로 직접 입력)

▲ Internet Explorer로 확인한 결과

## 7.5.2 Openwave 6.2.2 시뮬레이터

Openwave 시뮬레이터는 WML, WML2, X-HTML 등을 모두 지원하는 브라우저이다.

아래 사이트로 접속을 하면 다음과 같은 화면을 볼 수 있다.

- ▷ <http://155.230.105.167/mobile/asp/bloodtype/wml/index.asp>
- ▷ <http://155.230.105.167/mobile/asp/bloodtype/wml2/index.asp>
- ▷ <http://155.230.105.167/mobile/asp/bloodtype/xhtml/index.asp>



▲ Openwave 6.2.2 시뮬레이터로 확인한 결과.  
(가운데 오류화면은 post 방식을 사용한 경우)

※ Openwave 6.2.2 시뮬레이터의 한글 설정: Tools / Options / Device / Language에서 설정

### 7.5.3 UP 및 KUN 시뮬레이터

그 밖에 UP 시뮬레이터와 KTF KUN 브라우저의 실행 결과 화면은 다음과 같다.



▲ UP 시뮬레이터로 확인한 결과.

※ UP 시뮬레이터 한글 설정: Setting / Device Settings 메뉴의 Language, Charset 수정



▲ KTF KUN 2.1 브라우저로 확인한 결과.

## 제 8 장. mobile HTML

### 8.1 mHTML과 KUN (KTF Unified Navigator)

#### 8.1.1 개요

##### mHTML(mobile-HTML)

마이크로소프트에서 ME(Mobile Explorer) 브라우저를 출시하면서 내놓은 모바일용 HTML입니다. HTML을 기반으로 하고 있으며 기본적인 HTML 태그를 그대로 사용하므로 작업하기 편리하지만 스크립트를 지원하지 않는 등 제약 또한 많이 가지고 있습니다.

우리나라에서는 016, 018의 KTF에서 지원합니다. mHTML은 확장자가 html로 유선 홈페이지의 문서와 같으므로 서로 호환이 가능합니다.

mHTML 문서를 열기 위해서는 ME(Mobile Explorer) 브라우저를 사용합니다.

휴대폰의 기능이 좋아짐에 따라 보다 다양한 표현이 요구가 됨에 따라 개발된 브라우저가 KUN(KTF Unified Navigator)입니다.

KUN은 KTF에서 내놓은 웹브라우저 이름으로 기존의 ME 브라우저보다 다양한 HTML 언어를 인식하여 폭넓게 문서를 꾸밀 수 있습니다.



ME 브라우저용 문서



KUN 브라우저용 문서

## 8.1.2 mHTML 언어 구조

mHTML은 HTML과 동일한 구조를 가지고 있습니다.

<html> 태그를 사용해서 HTML임을 선언하고 <head> 태그에는 문서에 대한 정보에 관련된 내용을 입력하고 <body> 태그에는 화면에 출력할 내용을 입력합니다.

KUN 역시 mHTML 형식을 고스란히 따르고 있습니다. 단지 mHTML보다 더 많은 HTML 태그들을 지원합니다. 예를 들어 mHTML은 표에 다양한 속성들을 지원하지 않으므로 표를 이용해서 문서를 꾸밀 수 없지만 KUN은 표에 다양한 속성을 지원하므로 표를 이용해서 문서를 꾸밀 수 있습니다.

mHTML과 KUN의 가장 큰 특징은 KUN에서는 문서를 구성하는 레이아웃에 표를 이용한다는 점이라고 말할 수 있습니다.

```
<html>
  <head>

  </head>
  <body>
    <!-- 주식 -->
      내용
  </body>
</html>
```

### mHTML 기본 구조

```

<html>
<head>
</head>
<body>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr><td height="1" bgcolor="#FF00FF"></td></tr>
<tr><td height="19" align="center" bgcolor="#FF9900"><font color="black"><b>타이
틀</b></font></td></tr>
<tr><td height="1" bgcolor="#FF00FF"></td></tr>
<tr><td height="1" bgcolor="#FF00FF"></td></tr>
</table>
    본문내용<br>
</body>
</html>

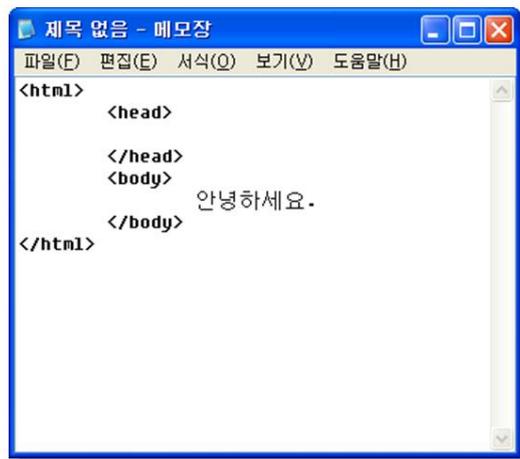
```

### 표를 이용하는 KUN 기본 구조

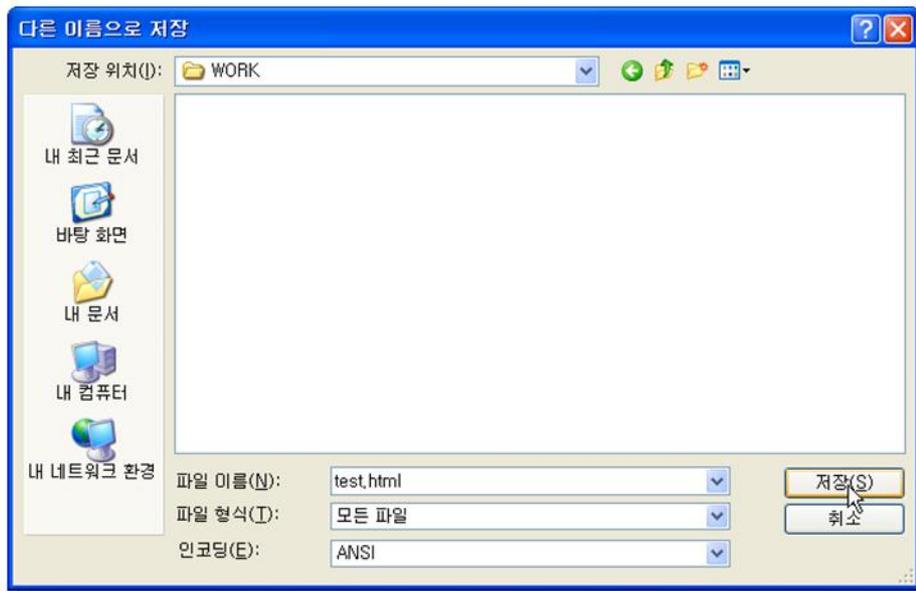
#### ● ME 브라우저로 문서 열기

메모장과 같은 텍스트 편집 프로그램을 사용해서 소스를 편집하고 HTML 형식으로 문서를 저장한 후 [KTF\_ME] 시뮬레이터로 문서를 열어 볼 수 있습니다.

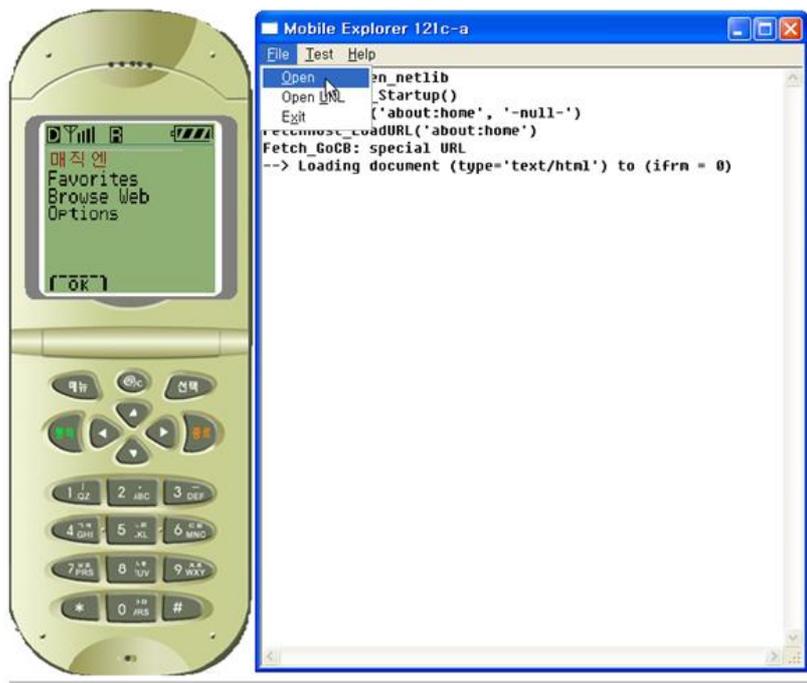
1. 메모장을 실행한 다음 mHTML 소스를 입력합니다.



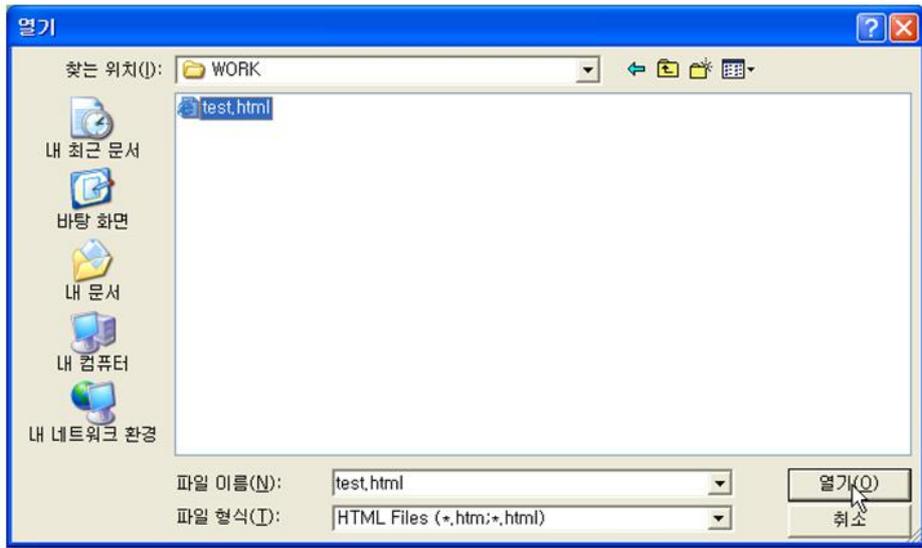
2. [파일>저장] 메뉴를 클릭해서 문서를 저장합니다. 이때 파일 형식은 [모든 파일]을 선택하고 [파일 이름] 항목에는 '파일이름.html' 식으로 입력합니다.



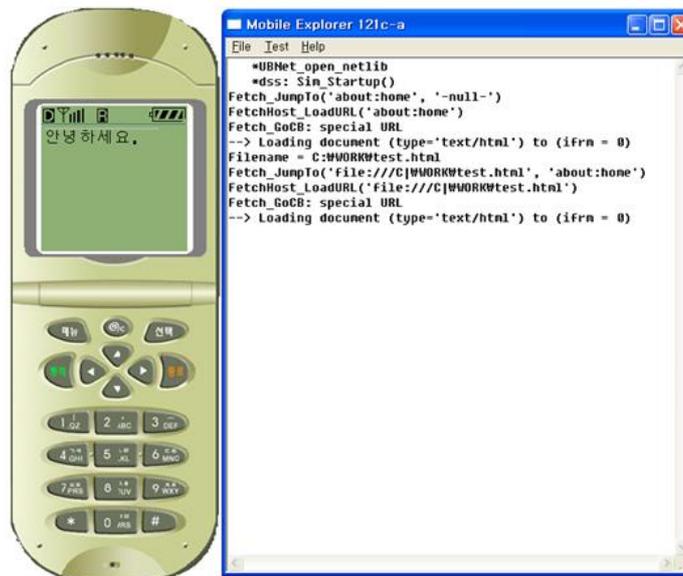
3. [KTF\_ME] 시뮬레이터 프로그램을 실행한 다음 [File>Open] 메뉴를 누릅니다.



4. [열기] 대화상자가 나타나면 문서를 저장한 폴더와 파일을 선택한 다음 [열기] 단추를 클릭합니다.



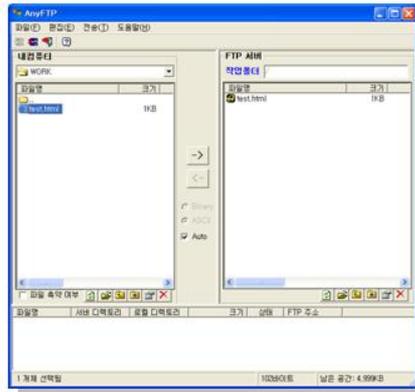
5. 결과 화면이 출력됩니다.



● KUN 브라우저로 문서 열기

[KUN] 시뮬레이터는 내 컴퓨터에 저장된 문서를 불러 올 수 없으므로 “별도의 서버” 계정에 올린 후 문서를 확인해야 합니다.

1.서버에 만든 자신의 계정에 작업한 문서를 전송합니다.



2.[KUN] 시뮬레이터를 실행한 다음 [URL] 목록을 선택합니다.



3. [URL 입력] 페이지가 나오면 문서를 올린 주소를 입력하고 [이동] 단추를 선택합니다.



4. 작업한 내용이 열립니다.



HTML은 인터넷 익스플로러에서도 지원하므로 주소 입력줄에 주소를 입력해서 내용을 확인할 수 도 있습니다.



## 8.2 mHTML 기본 태그

### 1> 문단 태그 <p> / <div>

문단을 정의해주는 태그들 입니다. align 속성을 사용해서 문단을 정렬합니다.  
<div> 태그는 KUN 브라우저에서 동작합니다.

```
<p align=" " >내용</p><div align=" " >내용</ div>
```

align : 정렬을 설정합니다. left, center, right

#### [예제] 글자 입력

글을 입력하고 <p>와 <div> 태그를 사용해서 문단을 정의하고 align 속성을 이  
용해서 정렬을 해보도록 하겠습니다.

```
<html>  
<head></head>  
<body><p align="center">애니빌더</p>  
<div align="right"> 애니빌더는 각종 모바일 언어로 무선 인터넷 폰페이지를  
제작해주는 도구입니다.</div>  
</body>  
</html>
```



ME브라우저



KUN브라우저

## 2> 글자 관련 태그

글자를 꾸며주는 속성 태그입니다. ME 브라우저에서는 동작하지 않으며 KUN 브라우저에서는 지원하지만 브라우저에 탑재되어 있는 글꼴에 따라 적용되지 않을 수 있습니다.

### [예제]글자 속성

글자 속성을 설정해주는 태그를 이용해서 문서에 입력한 글자를 꾸미는 예제입니다.

```
<html>
<head> </head>
<body>
<center> 태 권 도 </center>
<p><b>태권도</b>는 무술로서 만이 아닌 무예로서의<br>
<cite>철학적 정신세계</cite>가 내재된 <strong>우리 겨레의 고유한 전통적 산물
</strong><br>
이라 할 수 있다. 지구상에는 각기 다른 많은 민족들이<br>
생존경쟁의 역사 속에서 발전을 거듭해 왔으며 그 중 <br>
우리 민족은 5,000년이라는 유구한 역사와 전통성을<br>
유지해 오면서 <blockquote>민족의 정신을 계승 발전</blockquote>시켜 왔다. </p>
</body>
</html>
```



ME브라우저



KUN브라우저

## ❖글자 크기 태그 <h1>~<h6>

글자의 크기를 h1에서 h6까지 6단계로 조절합니다. 숫자가 클수록 글자는 작아 집니다. KUN 브라우저에서 동작하나 브라우저에 탑재된 폰트의 크기에 따라 지원 여부가 결정됩니다.

```
<hn>문장</hn>  
n: 1~6
```

### [예제]글자 크기

<hn> 속성을 이용해서 글자 크기를 설정합니다.

이 기능은 휴대폰에 따라 지원되기도 하고 지원하지 않기도 합니다.

```
<html>  
  <head>  </head>  
  <body>  
    <h1>애니빌더</h1> <br>  
    <h3>애니빌더 폰피에 오신 것을 환영합니다.</h3>  
  </body>  
</html>
```



### 3> 문서 속성 태그

#### ❖ 문서 속성 태그 <body>

문단을 지정하고 문서 색 및 글자, 링크가 설정되어 있는 글자의 색을 설정합니다. 색상은 16 가지의 색이름을 입력해서 설정이 가능하며 컬러 액정일 경우와 4가지 색 액정일 경우와 색이 다릅니다. 색상 설정은 KUN 브라우저에서 지원합니다.

```
<body bgcolor=" " text=" " link=" " alink=" " blink=" " btn=" " btn2=" " href=" " href2=" " >  
    본문  
</body>
```

bgcolor : 문서 배경에 색을 지정합니다.

text : 문서에 입력한 글자의 색을 지정합니다.

link : 하이퍼링크가 설정되어 있는 글자에 색을 지정합니다.

alink : 하이퍼링크를 실행한 글자의 색을 설정합니다.

blink : 하이퍼링크가 설정되어 있는 곳에 색을 설정합니다.

btn : 하단 메뉴의 왼쪽 이름을 설정합니다.

btn2 : 하단 메뉴의 오른쪽 이름을 설정합니다.

href : 왼쪽 하단 메뉴를 실행할 경우 경로를 설정합니다.

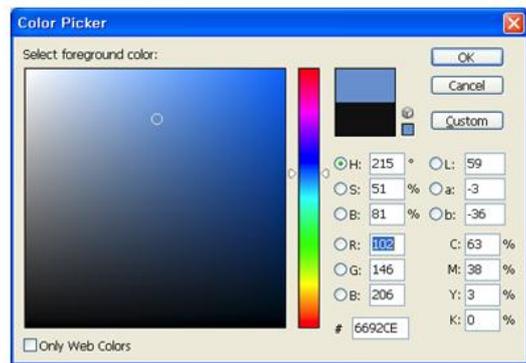
href2 : 오른쪽 하단 메뉴를 실행할 경우 경로를 설정합니다.

#### ▼ 색 입력값

16 컬러나 4 가지 색 컬러는 다음의 색을 지원합니다.

색에 해당되는 이름을 적어주면 색을 표현해줍니다.

256 컬러 이상은 HTML 코드값을 사용합니다. 컬러액정 휴대폰은 256 컬러 이상의 색을 지원하며 KUN에서도 트루컬러의 색을 모두 지원합니다. HTML 코드값은 포토샵 등의 그래픽 프로그램에서 HTML코드값을 알 수 있습니다. HTML코드값은 # 기호와 함께 숫자와 알파벳으로 구성된 6자리 기호로 구성되어 있습니다.



## [예제] 문서에 색 설정하기

색 기호와 HTML 코드값을 이용해서 문서 배경과 글자 등에 색을 넣은 예제입니다.

```
<body bgcolor="#FF0000" text="white" link="blue" alink="navy" btn="메인" btn2="상위" href="a.htm" href2="b.htm">
<center><b>아마존 정글</b></center>
아마존 정글은 지구 전체 삼림의 30%를 차지하며, 이곳의 총 넓이는 남북한을 합친 우리나라 전체면적의 30배를 능가한다. <br><br>
<a href="c.htm">관련자료</a>
</body>
```



#### 4> 배경 이미지 태그

##### ❖ 배경 이미지 넣어주는 background 속성

<body> 태그에 background 속성을 사용해서 문서에 배경 그림을 넣어 줍니다. 이외에 여러 가지 속성을 사용해서 배경을 화면 가운데에 배치하거나 문서를 스크롤해도 고정시키는 등의 작업을 할 수 있습니다. 배경 그림은 다운로드시 패킷이 부과되는 요인으로 작용하므로 특별한 이유가 없을 경우 사용을 권장하지 않습니다. 이 기능은 KUN 브라우저에서만 동작합니다.

```
<body background=" " halign=" " valign=" " repeat=" " fixed=" " repeat_x=" " repeat_y=" ">
```

background : 백그라운드 이미지의 URL 지정

halign : 배경 그림의 가로 정렬을 설정합니다. left, center, right

valign : 배경 그림의 세로 정렬을 설정합니다. top, middle, bottom

repeat : 배경 그림을 반복해서 붙여 넣을지를 설정합니다.

fixed : 화면을 스크롤할 때 배경 그림도 함께 스크롤하게 할지를 설정합니다. yes, no

repeat\_x : 배경 그림을 가로 방향의 반복 회수 지정

repeat\_y : 배경 그림을 세로 방향의 반복 회수 지정

#### [예제] 문서 배경 가운데에 그림 넣기

<body> 태그에 background 속성을 사용해서 배경 그림을 문서 가운데에 배치하게 만든 예제입니다. 문서를 스크롤해도 항상 그림이 문서 가운데에 위치합니다.

```
<body background="bg.jpg" halign="center" valign="middle" repeat="no">
```

내용

```
</body>
```



## 5> 글자 속성 태그

### ❖ 글자 속성 태그 <font>

글자의 속성을 설정합니다. 태그 안에 size와 color 속성을 사용해서 색과 크기 값을 설정합니다.

<font> 태그에 설정한 내용이 <body> 태그에서 설정한 속성보다 우선적으로 설정됩니다. <font> 태그는 KUN 브라우저에서 지원하며 글자 크기는 브라우저에 탑재되어 있는 글꼴에 따라 지원 여부가 결정됩니다.

```
<font size=" " color=" " >문장</font>
```

#### [예제] 글자 색과 크기 설정

<font> 태그를 사용해서 글자 색과 크기를 설정하는 예제입니다. <font>로 설정한 내용은 <body> 태그에 설정한 속성보다 우선합니다. 역시 글자 크기는 휴대폰의 탑재되어 있는 글꼴에 따라 지원하기도 하고 지원 안하기도 합니다.

```
<body bgcolor="red" text="white">  
<center><b>메소포타미아</b></center>  
<font size="10" color="yellow">메소포타미아 문명이란 비옥한 반월지대의 대부분을  
차지하는 티그리스강, 유프라테스강 유역을 중심으로 번영한 고대문명을 말한  
다.</font> <br><br>  
</body>
```



## 6> 스크롤 태그

### ❖ 스크롤 태그 <marquee>

태그에 입력한 내용을 왼쪽 방향으로 스크롤하게 만듭니다.  
다양한 속성을 이용해서 꾸밀 수 있지만 보통 <marquee type="auto">을 설정해서 사용합니다. 이 기능은 ME에서는 동작하지 않습니다.

```
<marquee> 내용 </marquee>
```

- behavior : 글자가 흐르는 방식을 설정합니다.
  - slide : 영역을 처음부터 끝까지 한번만 흐릅니다.
  - scroll : 오른쪽에서 왼쪽으로 흐릅니다.
  - alternate : 글자의 시작과 끝을 연결하여 흐릅니다.
  - continuous : 오른쪽에서 왼쪽으로 반복해서 흐릅니다.
- direction : 스크롤 방향을 설정합니다. left, right
- loop : 스크롤 횟수를 설정합니다. 0~255까지 숫자를 입력할 수 있습니다. '0'을 입력하면 무한 반복 실행합니다.
- vspace : 위와 아래에 여백을 만듭니다.
- width : 좌우에 여백을 만듭니다.
- bgcolor : 배경에 색을 설정합니다.
- src : 태그에 그림을 포함해서 흐르게 합니다.
- type : 'auto'로 설정하면 영역이 글자 길이보다 작을 경우에만 스크롤하게 합니다.

### [예제]스�크롤 효과

<marquee> 태그를 사용해서 글자를 움직이게 만든 예제입니다.  
하나는 behavior 속성과 direction, loop 속성을 이용하여 글자가 오른쪽으로 반복해서 흐르게 만들고 또 하나는 type 속성을 사용해서 글자의 길이가 영역보다 넓을 경우에만 스크롤되도록 만들어 보겠습니다.

```
<body>  
<center><b>KUN 이란</b></center><br>  
<marquee behavior="scroll" direction="right" loop="0">KUN이란 KTF Unified  
Navigator의 약자입니다.</marquee><br>  
<marquee type="auto">KUN은 mHTML보다 다양한 태그를 인식해주는 브라우저입니  
다.</marquee>  
</body>
```



7> 수평선 태그

❖ 수평선 태그 <hr>

[예제] 수평선 넣기

<hr> 태그를 사용해서 수평선을 넣어서 꾸민 예제입니다. 하나는 속성을 설정해서 화면의 50% 길이만큼만 수평선이 표시되도록 만들었고 하나는 화면 전체에 꽉 차도록 수평선을 만들었습니다.

```

<body>
<center><b>이벤트</b></center><br>
원음 소리나라<br>
미디 소리나라<br>
노래방<br>
<hr width="50%" size="2"> //두께가 2 픽셀이고 너비가 화면의 반을 차지하는
수평선을 만듭니다.
캐릭터 그림나라<br>
애니메이션 그림나라<br>
<hr> //화면 너비에 꽉차는 수평선을 만듭니다.
메인<br>
통합검색<br>
</body>

```



## 8> 경로설정 태그

### ❖ 경로 설정 태그 <base> 태그

<head> 태그에 위치해서 문서의 기본 경로를 설정합니다. 경로를 설정하면 <body> 태그 안의 경로에 모두 반영됩니다. KUN 브라우저에서 지원합니다.

```
<base href=" ">
```

### [예제]기본 경로 설정하기

<base> 태그를 사용해서 기본 경로를 설정한 예제입니다.  
그림 경로에는 기본 경로를 중심으로 경로가 설정됩니다.

```
<html>
<head>
  <base href="http://www.anybil.com/list/main.html" >
</head>
<body>
우리 회사의 로고는 입니다.
</body>
</html>
```

## 9> 배경음악 태그

### ❖ 배경 음악 태그 <bgsound>

문서를 열면 경로에 설정한 음악 파일을 실행하여 멜로디를 들려 줍니다. src 속성에 파일 이름을 상관없이 '\*.vib'라고 입력하면 휴대폰을 진동시킵니다.

```
<bgsound src="음악파일이름">
```

#### [예제] 배경 음악 연주

입력한 음악 파일을 연주합니다. 이때 음악 파일의 크기는 1Kb 이하로 설정하도록 합니다.

```
<html>
<head>
</head>
<body>
<bgsound src="음악파일이름">
눈을 지그시 감으시고 음악을 들어 보세요.
</body>
</html>
```

#### [예제] 휴대폰 진동

파일이름에 상관없이 vib라는 확장자를 사용하면 문서 로딩시 휴대폰을 진동하게 만들어 줍니다.

```
<html>
<head>
</head>
<body>
<bgsound src="a.vib">
휴대폰이 진동하나요?
</body>
</html>
```

10> 타이머를 설정하는 <meta> 태그

<head> 태그에 위치하여 메타 정보를 설정합니다. 입력한 시간이 지나면 지정된 페이지가 열리게 해주는 Refresh 속성만 지원합니다. 이 기능은 오류를 발생할 확률이 높기 때문에 권장하지 않는 기능입니다.

```
<meta http-equiv="refresh" content="3;url= 경로 설정 ">
```

[예제] 3초 후 홈페이지 열리게 만들기

<meta> 태그를 사용해서 문서가 열린지 3초가 지나면 지정된 홈페이지로 이동하게 만든 예제입니다.

```
<html>  
<head><meta http-equiv="refresh" content="3;url=  
http://www.anybil.com"></head>  
<body>잠시 후 애니빌 홈페이지로 이동합니다.  
</body>  
</html>
```



11> 광고 태그 <adver>

<adver>는 화면 하단에 표시되는 광고 메시지를 출력해주는 태그입니다.  
문서에 접근할 때 로딩할 때만 표시되면 간단한 글이나 그림 등으로 꾸밀 수 있습니다.

```
<adver type=" " inline=" ">
```

- type : 입력할 내용의 형식을 입력합니다. 그림 형식을 사용할 경우 nbmp, 글자를 입력할 경우 str를 입력
- inline : type 속성에 'str'로 설정했을 경우 광고로 사용할 글을 입력합니다. 'nbmp'를 설정했을 경우에는 BASE64 형식의 인라인 코드를 입력합니다.

[예제] 광고 메시지 표시하기

<adver> 태그를 사용해서 페이지 이동시 광고 메시지가 나타나게 만든 페이지입니다.  
문서 로딩하는 동안에만 표시되므로 문서로딩이 빠를 경우에는 나타나지 않을 수 있습니다.

```
<body>  
<adver type="str" inline="CODMEDIA 홈페이지입니다. "> //광고 메시지 표시  
<br>  
이곳은 CODMEDIA 홈페이지입니다. <br>  
앞으로 많은 관심 부탁드립니다.<br>  
<a href="애니빌홈가기http://www.anybil.com">애니빌홈가기</a>  
</body>
```



## 8.3 그림과 링크, 목록 관련태그

### 8.3.1 <img> 태그

#### ❖ 그림을 삽입해주는 <img> 태그

그림을 삽입해주는 태그입니다.

경로를 암호화하기 위해 BASE64 문자열로 디코딩한 그림 경로를 이용할 경우에는 inline 속성을 이용합니다.

```
  
<img inline" " type=" " >
```

-src : 그림 경로를 설정합니다.

-inline : BASE64 문자열로 디코딩 되어 있는 DATA 값을 갖습니다. 이때 type 속성을 이용해서 그림의 속성값을 입력합니다.

예) <img type="sis" inline="XDfEaT^&+AaaER....." >

#### [예제] 그림 삽입

<img> 태그를 사용해서 gif,와 jpg 그림을 삽입한 예제입니다. ME 브라우저에서는 JPG 가 출력이 안되고 KUN 브라우저에서는 GIF 그림이 출력 안 되는 것을 볼 수 있습니다.

```
<html>  
<head></head>  
<body>  
<center>  
  
<p>날아 보자<br>  
  
</center>  
</body>  
</html>
```



## ❖픽토그램 그림 넣기

픽토그램이란 브라우저에 탑재되어 있는 그림들로 지정된 경로만 입력해서 그림을 불러 올 수 있습니다.

보통 문서에 삽입한 그림은 그림이 있는 서버에서 불러 오기 때문에 문서 로딩을 저하시키는 문제를 가지고 있지만 픽토그램은 브라우저에서 불러 오기 때문에 문서 로딩에 큰 영향을 끼치지 않습니다. 통신사에서조차 픽토그램의 활용을 적극 추천하고 있습니다. 픽토그램은 수시로 업그레이드가 되기 때문에 버전에 따라 차이가 있을 수 있습니다.

픽토그램을 사용하는 방법은 <img> 태그에 픽토그램 리스트를 참조하여 사용하고 싶은 그림의 해당 경로를 입력해줍니다.

픽토그램의 그림 형식은 SIS입니다.

```

```

### [예제] 픽토그램 넣기

픽토그램을 이용해서 꾸민 예제입니다. 픽토그램은 브라우저와 버전에 따라 차이가 있습니다. 여기서는 KUN 브라우저 v3에 지원하는 픽토그램을 이용해서 만들어 보겠습니다.

```
<body>  
<center>새로나온 그림 </center>  
뗏목 이벤트 <br>  
파스텔 동화 <br>  
최고가 되는 4가지 방법 <br>  
어린왕자 <br>  
그림나라 메인 <br>  
매직앤홈 <br>  
통합검색 <br>  
</body>
```



### 8.3.2 <A> 태그

#### ❖ 링크를 만들어 주는 <A> 태그

<a>는 문단을 지정해주는 태그로서 href 속성과 함께 사용하여 하이퍼링크를 설정할 때 사용합니다.

accesskey 속성을 사용하면 휴대폰 키 버튼으로 동작시킬 수 있고 btn와 btn2 속성을 사용하여 하단 메뉴에 이름을 표시할 수 있습니다.

```
<a href=" " accesskey=" " btn=" " btn2=" " title=" " focus>
```

-accesskey : 단말기 키패드 번호를 설정합니다. 1-9번까지 설정이 가능합니다.

-btn, btn2 : 하단 메뉴에 왼쪽과 오른쪽에 표시할 이름을 설정합니다.

-focus : 페이지를 열 때 포인터가 위치할 곳을 지정합니다.

-title : 설명글을 입력합니다.

## [예제] 메뉴 만들기

<a href> 태그를 사용해서 메뉴를 만든 예제입니다. accesskey 속성을 사용했을 경우 '①' 과 같이 원형 특수기호를 입력해서 사용해서 사용자가 실행할 휴대폰 키를 알려 주어야 합니다. accesskey 속성이 없는 경우 비워 두기 보다는 주목성 있는 특수기호를 사용하는 것이 좋습니다. title 속성을 이용하면 메뉴에 커서가 위치할 때 풍선 도움말로 표시할 수 있습니다.

```
<body>
<center>새로나온 그림</center>
<a href="a.html" btn="GO" title="이벤트" focus>♣뺏다 이벤트</a><br>
<a href="b.html" title="파스텔동화" accesskey="1">① 파스텔 동화<br></a>
<a href="c.html" title="최고 4가지 방법" accesskey="2">② 최고가 되는 4가지 방법
</a><br>
<a href="d.html" title="어린왕자" accesskey="3">③ 어린왕자</a><br>
<a href="e.html" title="메인">☞그림나라 메인</a><br>
<a href="f.html" btn="홈" title="매직앤홈">☞매직앤홈</a><br>
<a href="g.html" btn="검색" title="검색">☞통합검색</a><br>
</body>
```



### 8.3.3 이미지 맵 태그

#### ❖ 이미지맵을 만들어 주는 <map>, <area> 태그

이미지 맵이란 하나의 이미지에 영역을 설정해서 링크를 지정해주는 기능을 말합니다. HTML에서 지원했던 기능을 KUN 브라우저에서 제공합니다.

이미지맵을 설정할 이미지에 usemap 속성을 설정해서 이름을 지정한 후 <map> 태그로 이름을 지정한 곳에 링크를 설정합니다. 링크는 사각형, 원형, 다각형으로 영역을 설정하고 좌표로 영역을 지정합니다.

```
  
<map name="이름">  
<area shape=" " coords="x1, y1, x2, y2" href=" ">  
</map>
```

**usemap** : <map>을 설정할 이름을 지정합니다.

**shape** : 링크 영역의 모양을 설정합니다.

rec - 사각형 모양을 설정합니다.

circle - 원형 모양을 설정합니다.

poly - 다각형 모양을 설정합니다.

**coords** : 링크 영역 좌표를 설정합니다.

sharp 속성이 rec일 경우 왼쪽 점의 x와 y값, 오른쪽 점의 x와 y값

sharp 속성이 circle일 경우 원의 중심의 x와 y값, 반지름 값

sharp 속성이 polygon일 경우 꼭지점의 x와 y값을 순서대로 나열

## 이미지맵 영역 잘 만드는 방법

### ▼ 좌표를 설정하는 방법

좌표를 잡는 방법은 포토샵을 이용하는 것이 좋습니다.

이미지를 불러 온 후 영역을 설정할 곳의 시작점에 마우스 포인터를 위치하여 [Info] 패널에 있는 좌표값을 가지고 옵니다.

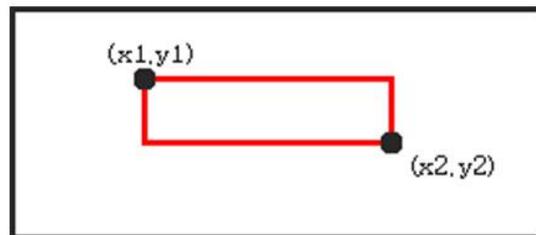
같은 방법을 끝점에도 마우스 포인터를 위치하여 좌표값을 가지고 옵니다.



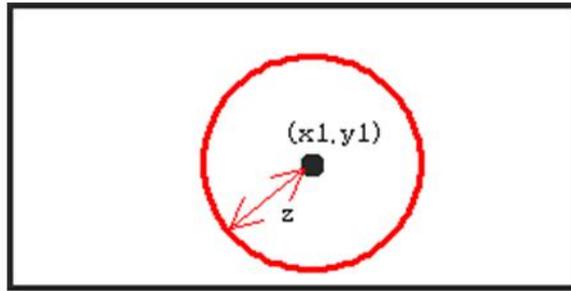
### 영역 설정하는 방법

영역은 좌표로 설정하는데 좌표는 그림의 왼쪽 상단을 (0,0)으로 지정하고 오른쪽으로 떨어진 거리는 x, 아래로 떨어진 거리를 y 값으로 설정합니다.

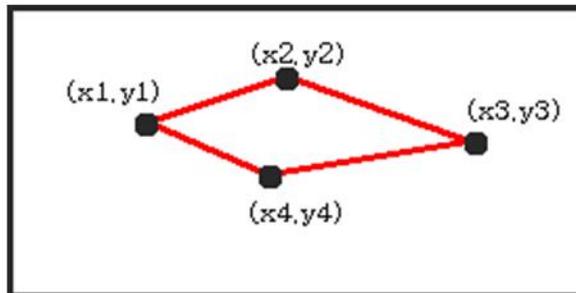
```
<area shape="rec" coords="x1,y1,x2,y2" href=" " >
```



`<area shape="circle" coords="x1,y1,z" href=" " >`



`<area shape="poligon" coords="x1,y1,x2,y2,x3,y3,x4,y4" href=" " >`



### [예제] 이미지맵으로 메뉴 만들기

그림에서 이미지맵을 설정해서 사각형 영역의 두곳에 링크를 설정한 예제입니다. 휴대폰의 방향키를 움직이면 링크를 설정한 곳 영역 선택이 바뀝니다.

```
<body>

<map name="m1">
<area sharp="rec" coords="2,6,52,21"
  href="http://www.anybil.com"> <area sharp="rec"
  coords="2,25,50,45" href="http://www.mnuri.co.kr">
</map>
</body>
```



### 8.3.4 목록 관련 태그

#### ❖ 목록을 만들어 주는 <dir>, <li>

하위 목록을 만들 때 사용하는 태그입니다.

<dir> 태그 안에 목록에 사용할 글자 앞에 <li> 태그를 입력해서 꾸미면 목록 글자가 한 단계 밑으로 들여쓰기가 됩니다.

같은 방법으로 중복해서 사용할 수 있습니다.

```
<dir>
  <li>목록
  <li>목록
</dir>
```

[예제] 간단한 목록 만들기

<dir>과 <li> 태그를 사용해서 자기 소개 목록을 만든 예제입니다.  
<dir> 태그 안에 <dir> 태그를 추가하여 만들었습니다.

```
<body>
<center> <b>자기소개<b> </center>
<dir> <li> <b>이름</b>
<dir> <li> 안창현</dir>
<li> <b>취미</b>
<dir> <li> 영화감상
<li> 사진촬영
</dir>
<li> <b>자주가는곳</b>
<dir> <li> <a href="http://www.anybil.com">애니빌</a></li>
      <li> <a href="http://www.mnuri.co.kr">엠누리</a></li>
</dir></dir>
</body>
```



❖ 목록을 만들어 주는 <menu>

단일 컬럼 메뉴 목록을 보여주는 메뉴입니다.

<li> 태그를 사용해서 목록을 정의하고 <menu> 태그로 묶어 줍니다.  
이 기능은 ME 브라우저에서는 동작하지 않습니다.

```
<menu>
  <li>목록1
  <li>목록2.....
</menu >
```

[예제] 간단 목록 요약 페이지 만들기

<menu>와 <li> 태그를 사용해서 목록을 만든 예제입니다.

```
<body>
<center> <b>그림 형식 소개</b> </center>
<b>JPG 특징<b>
<menu>
  <li>고해상도 지원
  <li>압축을 사용 가능
  <li>인터넷 표준 그림 형식
</menu>
<hr>
<b>GIF 특징<b>
<menu>
  <li>애니메이션 제작 가능
  <li>인터레이스 기능
  <li>인터넷 표준 그림 형식
</menu>
<hr>
<b>SIS 특징<b>
<menu>
  <li>애니메이션 제작 가능
  <li>압축을 우수
  <li>모바일 표준 기본 그림
  형식
</menu>
</body>
```



### ❖리스트 목록을 만들어 주는 <ol>

목록에는 <li> 태그를 사용한 후 리스트 목록의 형식을 입력한 <ol> 태그로 닫아주면 숫자, 로마숫자, 영문자로 번호가 삽입됩니다.

간단한 목록에 번호를 붙이고 싶을 때 사용합니다.

이 기능은 ME 브라우저에서는 동작하지 않습니다.

```
<ol start=" " type=" " >
  <li>목록1
  <li>목록2
.....
</ol>
```

start : 번호의 초기 값으로 설정합니다.

type : 번호의 유형을 지정합니다.

1 : 숫자, a : 알파벳 소문자, A : 알파벳 대문자, i : 로마자 소문자, I : 로마자 대문자

### [예제] 로마자 리스트 목록 만들기

<ol>와 <li> 태그를 사용해서 로마자의 리스트 목록을 만든 예제입니다.

```
<body>
<center><b>폰페이지 제작 순서<b></center>
<br><br>
<ol start="0" type="I" >
<li>환경설정
<li>폰페이지 문서 제작
<li>제너레이터 실행
<li>모바일 이미지 제작
<li>DB 연동
<li>FTP 전송
</ol>
</body>
```



## 8.4 표로 꾸며보는 폰페이지

### 8.4.1 <table> 태그

#### ❖ <table><tr><td> 태그

테이블을 만들려면 <table> 태그를 사용해서 테이블임을 정의하고 <tr> 태그를 사용해서 줄을 구분합니다.

구분된 줄에서 셀을 추가하려면 <td> 태그를 사용해서 셀을 추가한 후 내용을 입력해서 표를 꾸밉니다.

KUN 브라우저에서는 표에 관련된 다양한 속성들을 지원하는 반면 ME 브라우저에서는 columns 속성을 이용하여 표에 만들어 넣을 열의 개수만 입력이 가능하고 표의 경계선도 투명하게만 표시가 가능합니다.

```
<table title="소개" border=" " width=" " cellpadding=" " cellspacing=" " align=" ">
<caption>문자열</caption>
  <tr align=" " valign=" ">
    <td align=" " valign=" " rowspan=" " colspan=" ">셀</td>
  </tr>
</table>
```

- title : 표의 제목을 입력합니다.
- border : 표 테두리의 두께를 입력합니다.
- width : 표 너비를 입력합니다.
- cellpadding : 셀과 내용의 간격을 입력합니다.
- cellspacing : 셀과 셀 사이의 간격을 입력합니다.
- align : 좌우로 정렬합니다.
- valign : 상하로 정렬합니다.
- rowspan : 행과 행을 합칩니다.
- colspan : 열과 열을 합칩니다.

[예제] 표를 이용한 자기소개

ME 브라우저에서 지원하는 태그와 속성을 이용해서 제작한 표입니다.  
ME 브라우저를 사용할 경우 보통 액정 크기가 작기 때문에 2열 이상은 사용하기가 어렵습니다.

```
<table title="소개" columns=" " >
  <tr>
    <td>이 름</td>
    <td>안창현</td>
  </tr>
  <tr>
    <td>취 미</td>
    <td>영화감상</td>
  </tr>
  <tr>
    <td>이메일</td>
    <td>cod@cod.com</td>
  </tr>
</table>
```



[예제] 표를 이용하여 타이틀 만들기

KUN 브라우저에서 지원하는 다양한 표 속성을 이용해서 꾸민 표 예제입니다.  
ME 브라우저에서는 동작하지 않습니다.

```
<body>
<table width="100%" cellpadding="0" cellspacing="0">
<tr><td height="1" bgcolor="#99CCFF"></td></tr>
<tr><td height="20" bgcolor="#3366CC" align="center">
  <font color="#FFFFFF"><b>그림보기</b></font></td>
</tr>
<tr><td height="1" bgcolor="#003399"></td></tr>
</table>
<br>
①내 폰으로 저장<br>
②친구에게 보내기<br>
③커플링<br>
④바구니에 넣기<br>
</body>
```



## 8.4.2 <frame> 태그

### ❖ 화면을 나누어 주는 <frame>

화면을 나누어서 각각 다른 문서가 열리도록 해주는 태그입니다.

<frame> 태그로 각각 분할될 페이지에 내용을 설정하고

<frameset> 태그로 묶어 줍니다.

활용빈도가 적은 기능으로 KUN 브라우저에서는 정상적으로 동작하지 않습니다.

```
<frameset rows=" , ">  
<frame src="top.html" name="aa" focus>  
<frame src="bottom.html" name="bb">  
</frameset>
```

**rows** : 화면을 나눌 비율을 콤마(,)로 구분해서 입력합니다.

**name** : 선택한 프레임에 이름을 설정합니다.

**focus** : 문서를 열 때 커서를 위치하게 합니다.

### [예제] 프레임 나누기

프레임 태그를 사용해서 화면을 나누어서 꾸며보고 프레임의 메뉴를 실행하면 다른 프레임에 문서가 나타나도록 꾸며보자.

프레임셋 문서에서 name 속성을 사용해서 각 프레임에 이름을 설정한다.

```
<html>  
<head> </head>  
<frameset rows="2,4">  
<frame src="top.html" name="aa" focus>  
<frame src="bottom.html" name="bb">  
</frameset> </html>
```



## 8.5 mHTML용 입력양식

### 8.5.1 <form> & <input>

#### ❖ 입력양식을 정의해주는 <form> 태그

mHTML은 HTML과 동일하게 <form> 태그를 사용해서 입력양식을 정의합니다.

<form> 태그 안에 입력양식을 입력하고 실행키는 <input> 태그를 이용해서 만듭니다. 실행키를 누르면 입력양식에 입력한 값을 action 속성에 입력한 경로로 전송합니다.

```
<form action=" " method=" " >  
  입력양식  
</form>
```

-action : 입력양식의 값을 전달할 주소

-method : 전달 방식을 선택합니다.

#### ❖ 글상자를 만들어 주는 <input> 태그

글을 입력할 수 있는 입력양식을 만들어 주는 입력양식 태그입니다. 한 줄 글상자와 비밀번호 글상자로 구분할 수 있습니다.

```
<input type=" " size=" " name=" " value=" " mode=" " >
```

-type : 입력양식 종류를 선택합니다.

text : 글을 입력할 수 있는 한 줄 글상자 입력양식을 만듭니다.

password : 입력한 글이 '\*'로 표시해주는 비밀번호 입력양식을 만듭니다.

submit : 입력한 값을 <form> 태그의 action 속성에 입력한 주소로 전달.

reset : 입력양식에 입력한 값을 초기화합니다.

button : submit와 같은 실행 기능을 가지고 있지 않은 버튼 단추입니다.

-name : 입력양식의 이름을 입력합니다.

-size : 입력양식에 입력할 수 있는 글자수를 입력합니다.

-value : 입력양식에 초기값을 입력합니다.

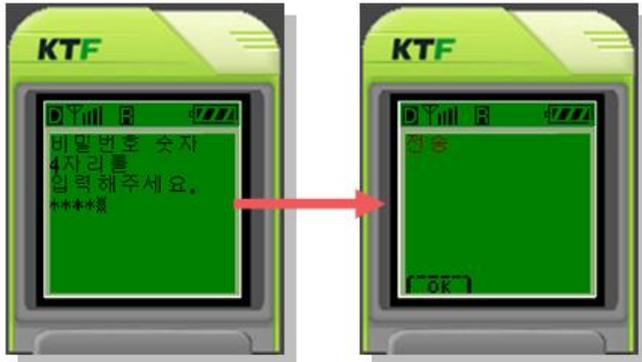
-mode : 입력 양식을 설정합니다.

(a : 영문 소문자, A : 영문 대문자, 1 : 숫자, K : 한글

**[예제]**

비밀번호 텍스트 입력양식을 이용해서 비밀번호를 입력하게 만든 예제입니다.  
비밀번호 텍스트 입력양식은 입력한 내용이 '\*' 기호로 표시됩니다.  
브라우저별로 어떤 차이가 있는지 살펴 보도록 합니다.

```
<html>  
<head> </head>  
<body>  
<form action="a.asp" method="post">  
비밀번호 숫자 4자리를 입력해주세요.<br>  
<input type="password" name="pass" size="4">  
<input type="submit" value="전송">  
</form>  
</body>  
</html>
```



ME브라우저



KUN브라우저

### [예제] 회원가입 페이지

텍스트 입력양식을 이용해서 이름, 주민등록번호, 아이디, 비밀번호를 입력하는 회원가입 페이지. 입력양식에 입력한 값을 b.asp로 전달하여 입력양식에 입력한 값을 화면에 출력.

```
<html>
<head> </head>
<body>
<table width="100%" cellpadding="0" cellspacing="0">
<tr> <td height="1" bgcolor="#99CCFF"> </td> </tr>
<tr>
<td height="20" bgcolor="#3366CC" align="center">
<font color="#FFFFFF"> <b> 회원가입 </b> </font>
</td>
</tr>
<tr> <td height="1" bgcolor="#003399"> </td> </tr>
</table>
<br>
<form action="b.asp" method="post">
이름 <input type="text" name="nm" size="10" mode="K"> <br> 주민등록번호 <br>
<input type="text" name="nb" size="14" mode="1"> <br> 아이디 <br>
<input type="text" name="id" size="14" mode="a"> <br> 비밀번호 <br>
<input type="password" name="pass" size="14"> <br>
<input type="submit" value="가입">
</form>
</body>
</html>
```

### b.asp

```
<html>
<body>
회원 가입이 완료되었습니다.
<p>이름 : <% =request("nm") %></p>
<p>주민등록번호 : <% =request("nb") %></p>
<p>아이디 : <% =request("id") %></p>
<p>비밀번호 : <% =request("pass") %></p>
</body>
</html>
```



❖ 장문을 입력하게 해주는 <textarea> 태그

여러 줄의 글을 입력할 수 있게 해주는 입력양식입니다.  
 row 속성으로 글상자의 줄 수를 지정하고 cols 속성으로 한 줄에  
 입력할 수 있는 글자의 수를 입력합니다.  
 ME 브라우저에서는 동작하지 않습니다.

```
<textarea name=" " rows=" " cols=" " ></textarea>
```

- name : 입력양식의 이름을 입력합니다.
- rows : 글상자의 줄 수를 입력합니다.
- cols : 한 줄에 입력 가능한 글자 개수를 입력합니다.

[예제] 자기소개 페이지

이름과 설명 글상자를 이용해서 입력값이 출력되도록 구성한 예제입니다.  
 이름과 자기소개 글을 받아서 c.asp 로 넘긴 후 ASP로 각 입력양식의 해당 값을 화면에  
 출력하게 만들었습니다.

```
<form action="c.asp" method="post">
이름 : <input type="text" name="id"> <br>
소개 : <textarea name="rt" rows="5" cols="60"> </textarea> <br>
<input type="submit" value="전송">
<input type="reset" value="다시입력">
</form>
```

c.asp

```
<html><body><p><% =request("id") %>님의 자기소개 글입니다.
</p><p><% =request("rt") %></p></body></html>
```



## 8.5.2 목록선택 양식

### ❖ 목록 선택 양식 <select>, <option> 태그

목록을 선택할 때 사용하는 입력양식입니다.

<option> 태그로 목록을 정의하고 <select> 태그로 묶어 줍니다.

사용자가 선택한 목록의 value 값을 <select> 태그의 name 속성에 할당되어 <form> 태그의 action 속성에 입력한 주소로 전송됩니다.

```
<form>
<select name=" " type=" ">
    <option value=" ">목록</option>
.....
</select>
</form>
```

-name : 입력양식의 이름을 설정합니다.

-type : 셀렉트 입력양식의 종류를 설정합니다. 'dropdown'라고 입력하면 드롭 다운단추가 만들어 집니다.

-value : 목록을 선택했을 때 전달할 값을 입력합니다.

#### [예제] 문제풀이 페이지

Select 입력양식을 이용해서 문제 풀이 페이지를 만든 예제입니다.

문제에 대한 답을 목록으로 구성한 후 각 목록에 대한 설명을 value 속성에 입력합니다. 사용자가 답을 선택하면 value 값이 화면에 출력되도록 만들었습니다.

```
<form action="d.asp" method="post" >
[질문]다음 중 KTF에서 사용하는 브라우저를 골라 보세요.<br>
<select name="qna" >
    <option value="WML 문서를 보여주는 브라우저입니
다.">AUR</option> //목록을 선택하면 value 속성에 입력한 값을 전달합
니다.
    <option value="정답입니다.">KUN</option>
    <option value="WML 문서를 보여주는 브라우저입니다.">OPENWAVE
SDK</option>
</select><input type="submit" value="전송" >
</form>
```

d.asp

```
<html>
<body>
<p><% =request("qna") %></p>
</body>
</html>
```



- 'AUR' 또는 'OPENWAVE SDK'를 선택했을 경우



- 'KUN'을 선택했을 경우

### [예제] 선택 메뉴 페이지

Select 입력양식에 type 속성을 'dropdown'로 설정하여 목록을 한 눈에 볼 수 있도록 만들어 본 예제

선택할 메뉴가 많을 경우 한 번에 메뉴 중에 하나를 고르게 하거나 다른 메뉴로 이동하게 하고 싶을 경우에 자주 사용

ASP에서 IF 문을 사용하여 구성하면 선택한 메뉴의 값을 검색하여 원하는 결과를 실행하게 만들 수 있음

```

<a href="a1.html" accesskey="1">①내폰으로 저장</a><br>
<a href="a2.html" accesskey="2">②친구에게 보내기</a><br>
<a href="a3.html" accesskey="3">③커플링</a><br>
<a href="a4.html" accesskey="4">④바구니에 넣기</a><br>
<form action="e.asp" method="post">
<table width="100%" cellpadding="0" cellspacing="1">
<tr><td width="16"></td>
<td width="40">
<select type="dropdown" name="mn" >
<option value="1">1/4</option>
<option value="2">2/4</option>
<option value="3">3/4</option>
<option value="4">4/4</option>
</select>
</td>
<td><input type="submit" value="이동"></td>
</tr></table></form>

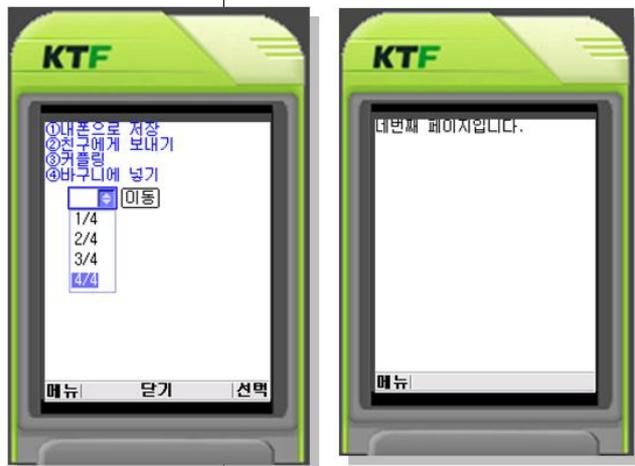
```

e.asp

```

<%
Response.Buffer=true
mn=request("mn")
If (mn="1") Then
Response.Redirect "a1.html"
ElseIf (mn="2") Then
Response.Redirect "a2.html"
ElseIf (mn="3") Then
Response.Redirect "a3.html"
Else
Response.Redirect "a4.html"
End If
%>

```



## 제 9 장. WML

### 9.1 WML 개요

WML(Wireless Markup Language)은 WAP 프로토콜에 따른 “무선 인터넷의 제약된 환경에 최적화된 언어”로서 기존의 HTML 과 같이 태그를 사용하여 누구나 쉽게 제작할 수 있도록 만든 마크업 언어이다.

WML의 특징은 다음과 같다.

- 카드를 이용한다!

WML은 하나의 <wml> deck 안에 여러 개의 <card>를 이용해서 문서를 꾸밀 수 있습니다.

WML deck 안에는 여러 개의 Card를 사용할 수 있지만 일반적으로 하나의 카드만 화면에 출력합니다.

- 히스토리 기능이 있다!

WML은 메뉴 구조를 구성할 수 있고 메뉴를 이용하여 열었던 페이지 이전 또는 다음 페이지로 이동할 수 있습니다.

열었던 페이지를 볼 수 있는 이유는 휴대폰에는 히스토리 스택 기능을 가지고 있기 때문입니다.

- wbmp 그림 형식을 이용한다!

WML에서 사용하는 그림은 좁은 대역폭에도 빠른 전송 속도를 제공하는 wbmp 형식을 이용합니다. 현재 wbmp 형식은 흑백만 지원하고 있지만 고해상도의 그림 형식도 개발이 이루어지고 있습니다.

컬러 그림은 모바일에 최적화되어 제작된 SIS 형식을 이용합니다.

- WML은 대소문자를 구분한다!

WML은 대소문자를 구분하므로 작업할 때 소문자로 통일하는 것이 좋습니다.

- WML은 XML 문서의 문자셋을 따른다!

WML은 XML이 서브셋이므로 XML과 같은 문자셋을 이용합니다.

US-ASCII / UNICODE 2.0 / ISO-8859-1/UTF-16

WML 문서의 기본 구조에 대해서 살펴보자.

```
<?xml version="1.0" encoding="ks_c_5601-1987"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML
1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
  <head>

  </head>
  <card>
    <!-- 주석 -->
    내용
  </card>
  <card>
    내용
  </card>
</wml>
```

- DTD 선언: DTD(Data Type Definition)란 문서에서 사용할 태그들을 정의하기 위한 일련의 구문규칙들이 정의되어 있는 파일을 불러 옵니다.
- Head : <meta> 태그 등 문서 전체에 관련된 정보를 입력하는 곳입니다.
- 설명글 : 문서에 관련된 설명글을 입력합니다. 설명글을 문서에 영향을 끼치지 않습니다.
- WML Deck : <wml>부터 </wml>까지 영역을 WML Deck라고 부릅니다.
- Card : <card> 태그로 구성된 단위로 하나의 문서에 여러 개의 카드를 위치할 수 있습니다.

#### ❖ WML 문서 제작하는 방법

##### ● 메모장을 이용하는 방법

메모장을 이용해서 손쉽게 WML 문서를 편집하고 WML 문서를 보여주는 SDK 시뮬레이터 프로그램을 사용해서 작업한 내용을 확인합니다.



```
제목 없음 - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<?xml version="1.0" encoding="ks_c_5601-1987"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
  <head>
  </head>
  <card>
  </card>
  <card> 안녕하세요.
  </card>
</wml>
```

## 9.2 WML 기본 태그

### ❖ <card> 태그

<card>는 프레임에 나타나게 할 내용을 정의하는 태그로서 하나의 문서에 여러 개의 <card> 태그가 올 수 있습니다.

여러 개의 <card>가 입력되어 있는 경우 보통 맨 위에 입력한 <card> 내용이 화면에 출력되고 다른 <card>는 화면에 나타나지 않습니다.

화면 전환을 이용하여 카드와 카드로 이동 할 수 있습니다.

```
<card id="" title="" onenterforward="" onenterbackward="" ontimer=""
  newcontext="" ordered="">
  contents
</card>
```

#### [속성]

- . id : card를 지정하는 이름입니다. 이름에 한글이나 공백은 사용할 수 없고 영문 대소문자를 구분합니다.
- . title : card의 설명글을 입력하는 이름입니다.
- . newcontext : <go> 태그로 해당 card 이동시 초기화 여부를 설정합니다. 속성값 : true / false
- . orderd : card의 내용을 순서대로 보여줄 것인지 사용자가 선택해서 보여 줄 것인지를 지정합니다. 속성값 : true / false
- . oneterforward : <go> 태그를 사용하여 해당 카드로 이동했을 때 이동할 URL을 설정합니다.
- . oneterbackward : <prev> 태그를 사용하여 해당 카드로 이동했을 때 이동할 URL을 설정합니다.
- . ontimer : 카드 안에 <timer> 태그를 사용하여 타이머 시간이 종료되었을 때 이동할 URL을 지정합니다.

## ❖ <do> 태그

휴대폰에서 Accept 키나 Options 키와 같은 키를 눌렀을 경우에 동작을 지정합니다.

```
<do type=" " label=" " name=" " option=" " ">  
  실행  
</do>
```

### [속성]

**type** : 동작 방법을 설정합니다.

- Accept : 휴대폰의 [확인(Accept)] 키를 눌렀을 경우에 동작을 지정합니다.
- options : 휴대폰의 [메뉴(Options)] 키를 눌렀을 경우에 동작을 지정합니다.
- delete : 휴대폰의 [취소>Delete)] 키를 눌렀을 경우에 동작을 지정합니다.
- prev : 휴대폰의 [이전] 키를 눌렀을 경우에 동작을 지정합니다.
- vnd.call : '☎ 라벨' 모양으로 기능키를 표시합니다.
- vnd.up : '⊙ 라벨' 모양으로 기능키를 표시합니다.
- vnd.next : 'Ⓧ 라벨' 모양으로 기능키를 표시합니다.
- vnd.prev : '▲ 라벨' 모양으로 기능키를 표시합니다.
- delete : '▶ 라벨' 형식으로 기능키를 표시합니다.
- reset : '▶ 라벨' 형식으로 기능키를 표시합니다.
- prev : '◀ 라벨' 형식으로 기능키를 표시하고 [취소] 단추를 누르면 <do> 태그 사이에 입력한 <go> 태그를 실행합니다.
- vnd.skmn1~9 : 1~9의 숫자로 메뉴를 표시합니다.

## ❖ <go> 태그

동작이 발생했을 경우 이동 경로를 설정합니다.

<do> 태그와 함께 사용하여 휴대폰 특정키의 이동 경로를 설정할 때 많이 사용합니다. 보통 닫기 태그를 사용하지 않은 경우가 많습니다.

```
<go accept-charset="" method="" href=""></go>
```

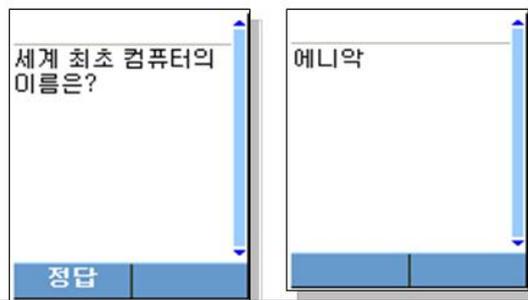
### [속성]

- . **accept-charset** : 사용하는 문자셋을 선택합니다.
- . **method** : 전송 방식을 선택합니다. 기본값은 **get**입니다. 속성값 : **get / post**
- . **href** : 이동 경로를 설정합니다. 속성값에는 홈페이지 주소를 입력하거나 같은 문서의 카드로 이동할 경우에는 '#카드 이름' 순으로 입력해서 이동 경로를 설정합니다.

### [예제] 카드 이동

문서를 열면 첫 번째 카드 내용이 출력되고 accept 키를 누르면 두 번째 카드의 내용이 출력되는 예제

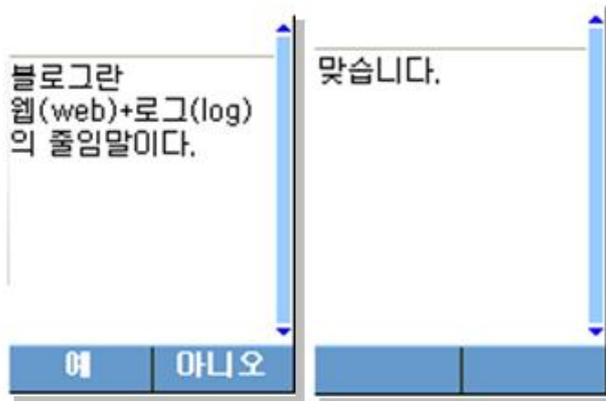
```
<?xml version="1.0" encoding="ks_c_5601-1987"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML
1.1/EN" "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
  <card id="a1">
    <p>세계 최초 컴퓨터의 이름은?</p>
    <do type="accept" label="정답"> // [정답]이라고 표기되는 라벨을 만듭니다.
      <go href="#a2"> // 라벨을 실행하면 a2 카드로 이동합니다.
    </do>
  </card>
  <card id="a2">
    <p>에니악</p>
  </card>
</wml>
```



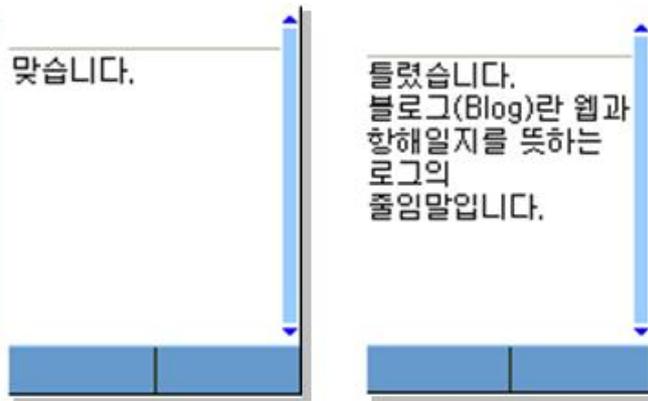
### [예제]OX 문제 출제

Accept 키와 Options 키를 이용하여 질문의 답변에 따라 각각 다른 카드로 이동

```
<?xml version="1.0" encoding="ks_c_5601-1987"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML
1.1/EN" "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
  <card id="a1">
    <p>블로그란 웹(web)+로그(log)의 줄임말이다.</p>
    <do type="accept" label="예">
      <go href="#a2">
    </do>
    <do type="options" label="아니오">
      <go href="#a3">
    </do>
  </card>
  <card id="a2">
    <p>맞습니다.</p>
  </card>
  <card id="a3">
    <p>틀렸습니다. 블로그(Blog)란 웹과 항해일지를 뜻하는 로그의 줄임말입니다.</p>
  </card>
</wml>
```



▲ [예]를 눌렀을 경우



▲ [아니오]를 눌렀을 경우

## ❖ <template> 태그

문서에 포함되어 있는 모든 카드에 동일하게 적용할 내용을 정의 합니다.  
<template> 태그에 있는 내용은 카드 위치에 상관없이 동일하게 적용됩니다.  
단 카드 안에 있는 내용이 <template>의 내용보다 우선으로 실행됩니다.

```
<template>  
  contents  
</template>
```

### [속성]

- . onenterforward : <go> 태그를 사용하여 해당 카드로 이동했을 때 이동할 URL을 설정합니다.
- . onenterbackward : <prev> 태그를 사용하여 해당 카드로 이동했을 때 이동할 URL을 설정합니다.
- . ontimer : 카드 안에 <timer> 태그를 사용하여 타이머 시간이 종료되었을 때 이동할 URL을 지정합니다.

### [예제]연속 퀴즈 문제

Accept 키와 Options 키를 template 태그에 담아두어 카드 위치에 상관없이 해당키가 화면에 나타나도록 구성한 예제.

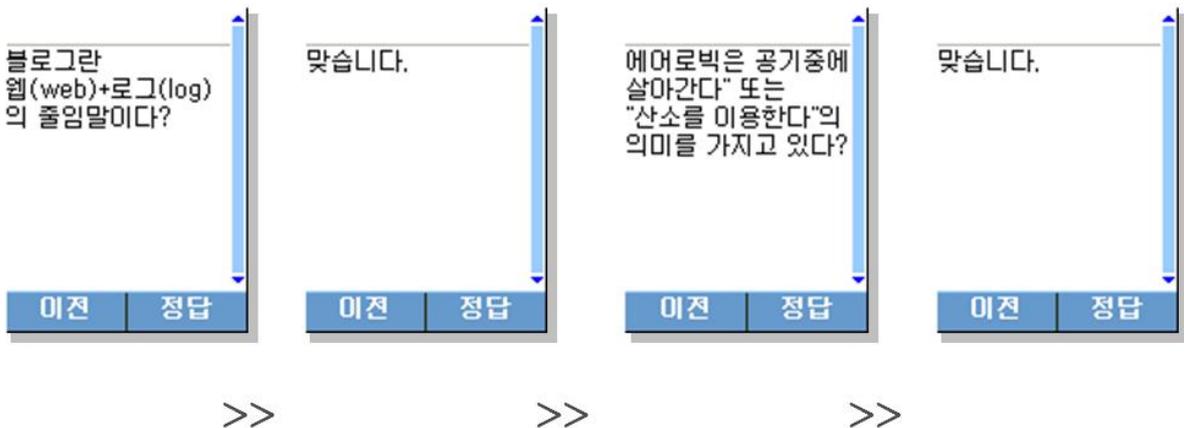
특히 a1 카드에는 onenterbackward 속성을 사용하여

a3 카드에서 <prev/> 태그로 a1 카드로 이동하면 a2 카드로 이동되도록 설정.

```

<?xml version="1.0" encoding="ks_c_5601-1987"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML
1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<template>
  <do type="accept" label="이전">
    <prev/>
  </do>
  <do type="options" label="정답">
    <go href="#a3">
  </do>
</template>
<card id="a1" onenterbackward="#a2">
  <p>블로그란 웹(web)+로그(log)의 줄임말이다?</p>
</card>
<card id="a2">
  <p>에어로빅은 공기중에 살아간다" 또는 "산소를 이용한다"의 의미를 가지고 있다?</p>
</card>
<card id="a3">
  <p>맞습니다.</p>
</card>
</wml>

```



## ❖ <noop> 태그

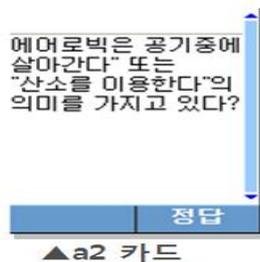
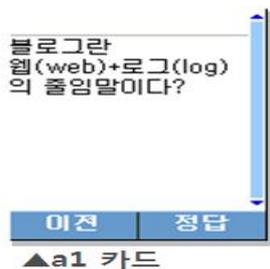
아무 동작도 실행하지 않도록 해주는 태그입니다. 실행하지 않게 할 태그 안에 넣어서 사용합니다. 보통 닫기 태그를 사용하지 않은 경우가 많습니다.

```
<noop>
  contents
</noop>
```

### [예제] 라벨 지우기

앞선 예제에서 a2 카드에 <noop> 태그를 사용해서 Accept 키가 실행되지 않도록 구성

```
<?xml version="1.0" encoding="ks_c_5601-1987"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML
  1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<template>
  <do type="accept" label="이전">
    <prev/>
  </do>
  <do type="options" label="정답">
    <go href="#a3">
  </do>
</template>
<card id="a1" onenterbackward="#a2">
  <p>블로그란 웹(web)+로그(log)의 줄임말이다?</p>
</card>
<card id="a2">
  <p>에어로빅은 공기중에 살아간다" 또는 "산소를 이용한다"의 의미를 가지고 있다?</p>
  <do type="accept" label="이전">
    <noop/>
  </do>
</card>
<card id="a3">
  <p>맞습니다.</p>
</card>
</wml>
```



## ❖ <meta> 태그

<meta> 태그는 <head> 태그에 위치하여 접근 권한 정보를 제공합니다.

```
<head>
  <meta http-equiv=" " content=" " forua=" " />
</head>
```

### [속성]

- . http-equiv : http-equiv="Cache-Control" 다음과 같이 사용하여 속성들이 http 헤더로 해석될 수 있도록 합니다.
- . name : 속성의 이름을 설정합니다.
- . forua : 메타 정보 값이 브라우저에서 사용될지 설정합니다. true/false 중 선택
- . content : 속성과 관련된 메타 데이터의 값을 표시합니다. 기본값은 30일이며 'content="max-age=no-cache"'라고 선언하면 캐시를 사용하지 않습니다.
- . scheme : 속성값을 해석하는데 사용되는 품이나 구조를 결정합니다.

### ▼ 사용 예

```
<meta http-equiv="Cache-Control" content="max-age=no-cache"
forua="true"/>
```

문서 내용을 휴대폰의 캐시에 저장하지 않고 항상 서버에서 데이터를 가져옵니다.

```
<meta name="vnd.up.markable" forua="true" content="false"/>
```

즐거찾기를 하지 못하도록 설정합니다.

```
<meta name="vnd.up.markable" forua="true" content="URL"/>
```

즐거찾기할 때 URL 설정합니다. content 속성에 URL을 입력합니다.

## ❖ <access> 태그

<head> 태그 안에 위치하여 문서의 접근 권한을 설정합니다.

즉 <access> 태그로 설정된 주소로 경유해야만 문서를 열어 주도록 해줍니다.

```
<head>  
  <access domain="211.53.39.203" path="/" />  
</head>
```

### [속성]

**domain** : 해당 덱에 접근이 가능한 다른 WML Deck의 URL을 표시하며, 기본값은 해당 WML Deck이 포함된 URL입니다.

**path** : WML Deck에 있는 카드에 접근할 수 있는 다른 WML Deck의 URL루트를 표시(기본값="/")

### 9.3 글과 그림 관련 태그

#### ❖ <p> 태그

문단을 정의해주는 태그로서 문단 정렬 및 문단 스크롤 하는 역할을 수행합니다. HTML에서는 <p> 태그를 사용하지 않아도 되지만 WML에서는 테이블이나 그림 삽입 등 요소를 넣을 때 꼭 입력해야 합니다.

```
<p align=" " mode=" ">  
내용  
</p>
```

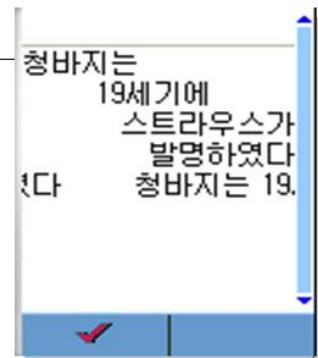
#### [속성]

- . align : 문단을 정렬합니다.
  - left : 문단을 왼쪽으로 정렬합니다.
  - center : 문단을 가운데로 정렬합니다.
  - right : 문단을 오른쪽으로 정렬합니다.
- . mode : 문단의 내용을 한 줄로 스크롤 할지 여부를 설정합니다.
  - wrap : 문단 스크롤을 실행하지 않습니다.
  - nowrap : 문단 스크롤을 합니다.

#### [예제] 문단 정렬

<p>태그에 align 속성을 사용해서 글자 정렬하고 mode 속성 이용해서 글자를 스크롤하게 만들어주는 예제입니다.

```
<?xml version="1.0" encoding="ks_c_5601-1987"?>  
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML  
1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml">  
<wml>  
  <card>  
    <p>청바지는</p>  
    <p align="center">19세기에 </p>  
    <p align="right">스트라우스가 발명하였다</p>  
    <p mode="nowrap">청바지는 19세기에 스트라우스가 발명하였다</p>  
  </card>  
</wml>
```



## ❖ 특수 문자 입력 관련 태그

WML 소스에 사용되는 <, ', 등의 문자들은 WML 소스에 사용되는 문자이므로 다른 방법으로 표기를 해주어야 합니다.

다음과 같은 특수기호는 해당 입력기호를 사용해서 표현 합니다.

### 표시입력기호

< &lt;

> &gt;

' &apos;

" &quot;&

&amp;\$\$

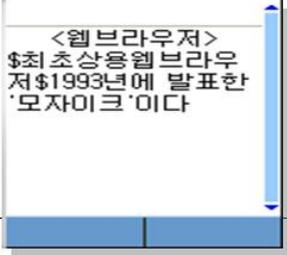
\$공백&nbsp;sp;

소프트 하이픈&shy;

### [예제] 특수 문자 출력

특수문자를 이용해서 문서를 꾸민 예제입니다.

```
<?xml version="1.0" encoding="ks_c_5601-1987"?><!DOCTYPE wml
PUBLIC "-//WAPFORUM//DTD WML
1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card>
<p align="center">&lt;웹브라우저&gt;</p>
<p>$$최초상용웹브라우저$$ 1993년에 발표한 &apos;모자이크&apos;이다
</p>
</card>
</wml>
```



❖ 글자 속성 태그 <i> / <b> / <u> / <em> / <small> / <strong> / <big>

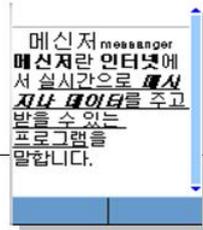
글자를 예쁘게 꾸밀 수 있는 글자 속성에 관련된 태그입니다.  
HTML에서 사용하는 태그와 같습니다.

태그	설명
<i>내용</i>	글자를 이탤릭체로 표시합니다.
<b>내용</b>	글자를 두껍게 표시합니다.
<u>내용</u>	글자에 밑줄을 표시합니다.
<em>내용</em>	글자를 강조합니다.
<small>내용</small>	작은 글자로 표시합니다.
<strong>내용</strong>	힘찬 글꼴로 표시합니다.
<big>내용</big>	큰 글자로 표시합니다.

[예제]글자 속성 출력

글자 속성에 관련된 태그를 이용해서 글자를 예쁘게 꾸며 봅니다.

```
<?xml version="1.0" encoding="ks_c_5601-1987"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML
1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card>
<p align="center"><big>메신저</big><small>messenger</small></p>
<p><b>메신저</b>란 <strong>인터넷</strong>에서 <u>실시간으로
<em><i>메시지나 데이터</i></em>를 주고 받을 수 있는 프로그램</u>을
말합니다. </p>
</card>
</wml>
```



## ❖ 표 관련 태그 <table> / <tr> / <td>

<table> 태그를 이용해서 표를 정의한 다음 줄을 구분해주는 <tr> 태그, 열을 구분해주는 <td> 태그로 표를 만들 수 있습니다.

기본 구조는 HTML과 유사하지만 열의 개수를 `columns` 속성을 사용해서 정의해야 하고 <table> 태그는 <p> 태그 안에 위치해야 한다는 특징을 가지고 있습니다.

WML에서는 표의 테두리가 나타나지 않습니다.

```
<p>
<table title=" " align=" " columns=" " >
  <tr>
    <td> 내용</td>
    ....
  </tr>
</table>
</p>
```

### [속성]

`title` : 테이블에 레이블을 지정합니다.

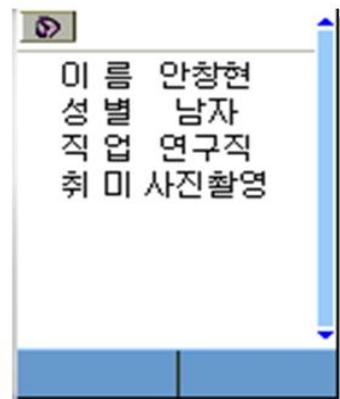
`align` : 표를 정렬합니다.

`columns` : 열의 개수를 입력합니다.

### [예제] 표 만들기

4행 2열의 표를 이용해서 자기소개 문서를 꾸민 예제입니다.

```
<?xml version="1.0" encoding="ks_c_5601-1987"?><!DOCTYPE wml
PUBLIC "-//WAPFORUM//DTD WML
1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card>
<p><table align="center" columns="2" >
<tr> <td>이 름</td><td> 안창현</td></tr>
<tr><td>취 미</td><td>사진촬영</td></tr>
<tr><td>취 미</td><td>사진촬영</td></tr>
</table></p>
</card>
</wml>
```



## ❖ <img> 태그

그림을 삽입해주는 태그로서 사용 방법은 HTML과 동일합니다.

```

```

### [속성]

**alt** : 그림이 안보일 때 대신 보이게 할 글을 입력합니다.

**src** : 그림 경로를 설정합니다.

**align** : 그림 정렬을 설정합니다.

**height** : 그림의 높이를 지정합니다.

**width** : 그림의 너비를 지정합니다.

**vspace** : 그림의 좌우 여백을 설정합니다.

**hspace** : 그림의 위아래 여백을 설정합니다.

### [예제] 그림 출력

문서에 그림을 삽입해서 꾸며 봅니다.

```
<?xml version="1.0" encoding="ks_c_5601-1987"?> <!DOCTYPE wml
PUBLIC "-//WAPFORUM//DTD WML
1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card><p></p>
</card>
</wml>
```



## ❖픽토그램 삽입 태그

<img> 태그에 `localsrc` 속성을 사용하면 브라우저에 저장되어 있는 그림을 다운로드 없이 빨리 불러 올 수 있습니다.

`localsrc` 속성값에 지정된 이름을 넣어주면 해당 그림 아이콘을 출력해줍니다. 이와 같은 그림을 픽토그램(Pictogram)이라고 부릅니다.

```

```

### [예제] 픽토그램 출력

문서에 그림을 삽입해서 꾸며 봅니다.

```
<?xml version="1.0" encoding="ks_c_5601-1987"?><!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card>
<p>010-000-0000</p>
<p>codmedia@codmedia.com</p>
<p>02-000-0000</p>
<p>02-000-0000</p>
</card>
</wml>
```



## 9.4 링크 관련 태그

### ❖ <a> 태그

하이퍼링크를 설정할 때 사용하는 태그로서 경로를 설정해주는 href 속성과 함께 사용합니다.

```
<a href=" " title=" " >  
글자, 그림 등  
</a>
```

#### [속성]

href : 링크를 실행할 경로를 설정합니다.

title : 링크의 레이블을 표시합니다.

#### [예제] 하이퍼링크

<a> 태그를 사용해서 사용자가 목록을 선택할 수 있는 리스트 메뉴를 생성

```
<?xml version="1.0" encoding="ks_c_5601-1987"?> <!DOCTYPE wml PUBLIC "-  
//WAPFORUM//DTD WML  
1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml">  
<wml>  
<card>  
<p>음 악  
</p>  
<p><a href="a.wml" title="GO">음악소개</a></p>  
<p><a href="b.wml" title="GO">음악게시판</a></p>  
<p><a href="c.wml" title="GO">티켓예매</a></p>  
</card>  
</wml>
```



## ❖ <anchor> 태그

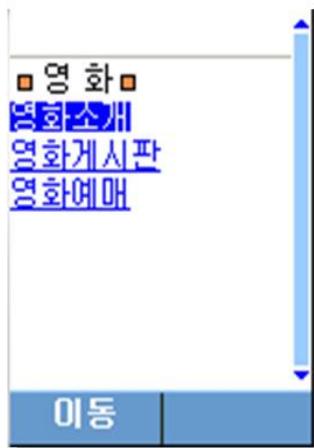
<anchor> 태그는 링크를 설정할 때 사용하는 태그 중 하나로서 <a> 태그와 기능이 유사하지만 경로를 설정해주는 속성을 사용할 수 없으므로 <go> 태그와 함께 사용한다는 특징을 가지고 있습니다.

```
<anchor title=" " ><go href=" " /> 내용 </anchor>
```

### [예제] 하이퍼링크

<anchor> 태그를 사용해서 링크를 실행할 수 있는 리스트 메뉴

```
<?xml version="1.0" encoding="ks_c_5601-1987"?><!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card>
<p>영 화
</p>
<anchor title="이동" ><go href="a.wml"/>영화소개</anchor>
<anchor title="이동" ><go href="b.wml"/>영화게시판</anchor>
<anchor title="이동" ><go href="c.wml"/>영화예매</anchor>
</card>
</wml>
```



## ❖ <select> / <option> 태그

<select>는 목록에서 선택하게 해주는 태그로서 <option> 태그와 <do> 태그를 함께 사용합니다. <select> 태그 안에 <option> 태그를 사용하여 리스트를 구성하고 <do> 태그를 사용해서 지정한 목록을 선택합니다.

<do> 태그는 <select> 태그가 있는 <card> 안에 위치해야 동작합니다.

단, <select> 태그 안에 <a> 또는 <anchor> 태그는 위치할 수 없습니다.

```
<select title=" " multiple=" " name=" " value=" " iname=" " ivalue=" "
  tabindex=" ">
<option value=" " onpick=" ">내용</option>
...
</select>
```

### [속성]

**title** : 태그의 레이블을 입력합니다.

**multiple** : 목록을 다중 선택을 가능 유무를 설정합니다. true / false

**name** : 변수의 이름을 입력합니다.

**value** : 기본값의 변수 이름을 입력합니다.

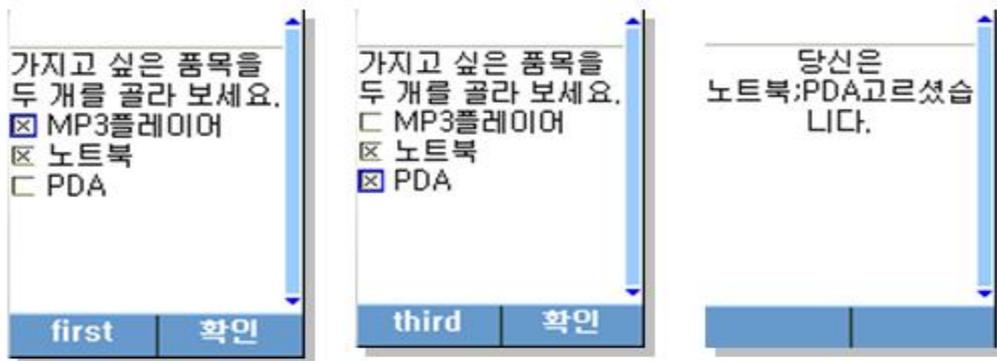
**iname** : 인덱스를 저장하는 변수의 이름을 입력합니다.

**ivalue** : iname 변수의 기본값을 입력합니다.

**onpick** : 하이퍼링크를 실행할 URL을 입력합니다.

### [예제6-4-4]여러 개 목록 선택

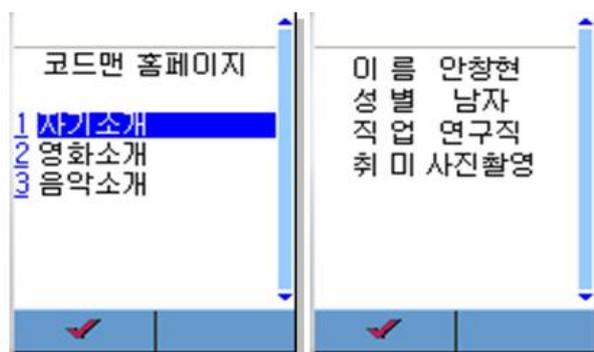
```
<?xml version="1.0" encoding="KS_C_5601-1987"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML
  1.1/EN" "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="select">
<p>가지고 싶은 품목을 두 개를 골라 보세요.<br/>
<select name="order" iname="index" ivalue="1;2" multiple="true" title="인기품목">
<option value="MP3플레이어" title="first">MP3플레이어</option>
<option value="노트북" title="second">노트북</option>
<option value="PDA" title="third">PDA</option>
</select>
<do type="vnd.next" label="확인">
<go href="#a2"/>
</do></p>
</card>
<card id="a2"> <p align="center">당신은 $(order) 고르셨습니다.</p></card>
</wml>
```



[예제]목록 선택해서 링크 실행

<select> 태그에 입력한 <option> 태그에 'onpick' 속성을 사용해서 목록을 선택하면 지정한 URL로 링크가 실행되도록 구성한 예제입니다.  
 이때 목록에는 번호가 순서대로 붙습니다.

```
<?xml version="1.0" encoding="KS_C_5601-1987"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="select">
<p align="center">코드맨 홈페이지</p><br/>
<select name="order" iname="index" title="인기품목">
<option value="자기소개" onpick="a.wml">자기소개</option> //목록을 선택하면
a.wml 문서를 엽니다.
<option value="영화소개" onpick="a.wml">영화소개</option>
<option value="음악소개" onpick="a.wml">음악소개</option>
</select>
</p>
</card>
</wml>
```



## ❖ <optgroup> 태그

<select> 태그 안에 위치하여 <select> 태그 안에 입력한 <option> 태그들을 그룹으로 만들어 주는 태그입니다.

```
<select>
<optgroup title=" ">
<option> 태그 목록
</optgroup>
</select>
```

### [속성]

title : 그룹으로 묶을 이름

### [예제] 목록을 그룹으로 구분

<optgroup> 태그를 사용해서 <select> 태그 안에 있는 <option> 리스트를 내용별로 구분했습니다. <optgroup> 태그를 사용했을 경우 휴대폰 서비스 및 기종에 따라 결과 화면이 다르게 나타납니다.

우리커플?	
궁금하신 항목을 선택하세요.	
<input type="checkbox"/>	이름
<input type="checkbox"/>	생년월일
<input type="checkbox"/>	취미
-----	
<input checked="" type="checkbox"/>	이름
<input checked="" type="checkbox"/>	생년월일
<input checked="" type="checkbox"/>	취미
취미	보기

결과입니다	
우리 커플은?	
김철수는;생일은 1980,10,14;취미는 영화감상입니다.;박영희는;생일은 1980,04,02이고;취미는 재즈댄스입니다.	

## ❖ <prev> 태그

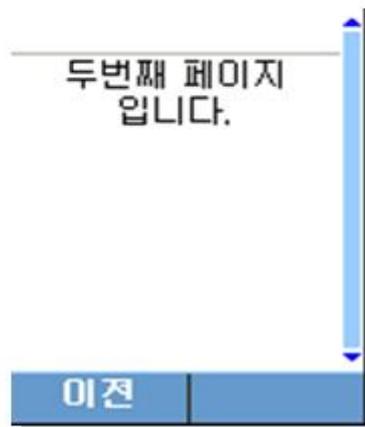
이전 경로로 이동하게 해주는 태그입니다. 페이지에서 페이지 또는 카드에서 카드로 바로 전에 열었던 내용을 보여 줍니다.

```
<prev>내용</prev>
```

### [예제] 이전 페이지로 이동

<do> 태그 안에 <prev/> 태그를 단독으로 사용하여 옵션 키를 누르면 이전 페이지로 이동하게 만든 예제입니다.

```
<?xml version="1.0" encoding="ks_c_5601-1987"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML
1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="a1">
<p><a href="#a2">다음 페이지로 이동</a> <br/> </p>
</card>
<card id="a2">
<p align="center">두번째 페이지 입니다.</p>
<do type="accept" label="이전"><prev/></do>
</card>
</wml>
```



## ❖ <timer> 태그

시간을 지정해주는 태그입니다.

<timer> 태그와 함께 <onevent> 태그를 사용해서 지정된 시간이 흐르면 <onevent> 태그의 내용을 실행하게 만들 수 있습니다.

```
<onevent type="ontimer">  
<go href=" ">  
</onevent>  
<timer name=" " value=" "/>
```

### [속성]

**name** : 변수명을 설정합니다.

**value** : 1/10초 단위로 시간을 설정합니다.

### [예제] 순차적으로 글자 출력

<timer>와 <onevent> 태그를 이용해서  
글자가 2초마다 순차적으로 바뀌는 효과



### [예제] 순차적으로 그림 출력

그림이 지정된 시간마다 순차적으로 바뀌게 하여  
마치 연결된 애니메이션처럼 보이도록 함



## 9.5 입력양식 관련 태그

### ❖ <input> 태그

WML에서 입력양식은 <input> 태그를 사용해서 만들 수 있으며 종류는 글상자와 비밀번호 글상자 두 가지를 만들 수 있습니다.

입력양식에 입력된 값은 <do> 태그를 이용하여 <go> 태그에 입력한 주소에 입력값을 보낼 수 있습니다.

```
<input name=" " title=" " type=" " value=" " format=" "
emptyok=" " maxlength=" " tabindex=" "/>
```

#### [속성]

Name	: 변수의 이름을 입력합니다.
title	: 태그의 이름을 설정합니다.
type	: 입력양식의 종류를 설정합니다.
text	: 글을 입력할 수 있습니다.
password	: 비밀번호를 입력할 때 사용하면 '*'로 표시됩니다.
value	: 입력받을 변수의 기본값
format	: 입력 값의 포맷을 지정합니다.
A	: 문자, 알파벳 대문자
a	: 문자, 알파벳 소문자
N	: 모든 숫자
X	: 문자, 숫자, 알파벳 대문자(대소문자 변경 불가)
x	: 문자, 숫자, 알파벳 소문자(대소문자 변경 불가)
M	: 문자, 숫자, 알파벳 대문자(대소문자 변경 가능)
m	: 문자, 숫자, 알파벳 소문자(대소문자 변경 가능)
emptyok	: 입력 값이 없을 경우 허용 여부를 설정합니다. true, false
maxlength	: 입력 값의 최대 길이를 설정합니다.

The image shows four examples of WML input forms, each with a blue border and a vertical scrollbar on the right. Each form has two input fields: one for ID and one for Password. Below each form is a blue button with white text.

- Form 1:** ID field contains 'ok', Password field contains '\*\*\*\*'. Button: ID 입력
- Form 2:** ID field contains 'hyunian', Password field contains '\*\*\*\*'. Button: (empty)
- Form 3:** ID field contains 'hyunian', Password field contains '\*\*\*\*'. Button: PWD
- Form 4:** ID field contains 'hyunian', Password field contains '\*\*\*\*'. Button: send

### ❖ <setvar> 태그

<go>, <prev>, <spawn>, <refresh> 태그에 사용하여 변수와 값을 지정하는 태그입니다.

```
<setvar name=" " value=" ">
```

### ❖ <refresh> 태그

변수의 값이 설정되어 있을 경우 변수 값을 다시 설정하도록 해주는 태그입니다.

```
<refresh>  
  변경 내용  
</refresh>
```

#### [예제] 재입력 입력양식

<do> 태그에 <refresh> 태그를 사용해서 옵션 키를 누르면 입력양식에 입력한 내용이 지워지도록 만든 예제입니다.

The image shows two side-by-side screenshots of a web form. The left screenshot shows the form with the following content: 'id 입력: HYUNIN' and 'password 입력: \*\*\*\*'. Below the form is a blue button with a refresh icon and the text '재입력'. The right screenshot shows the same form after the refresh button is clicked, with the input fields empty. The 'id 입력:' label is still present, but the text 'HYUNIN' is gone. The 'password 입력:' label is also present, but the text '\*\*\*\*' is gone. The '재입력' button is still visible at the bottom.

## 9.6 WML 2.0

### 9.6.1 WML 2.0 소개

#### ❖ WML2.0이란

WML 언어를 사용하는 WAP1.0은 표현에 대한 제약이 많고 유선 홈페이지와 호환이 되지 않아 개발이 힘들고 게이트웨이가 있어야 하므로 개발 비용이 많이 들어가는 등 여러 가지 문제점을 가지고 있습니다.

이러한 문제를 해결하기 위해서 2001년 말에 차세대 무선 인터넷 규격인 WAP 2.0을 발표하였습니다.

WAP2.0은 WML2를 언어를 사용하는데 WML2는 XHTML Basic 언어에서 모바일 환경에 맞도록 내용을 확장한 XHTML MP(Mobile Profile)와 WCSS(Wireless CSS) MP를 기본으로 사용하여 WML1.0보다 다양한 표현이 가능할 뿐만 아니라 유선 홈페이지와 호환이 가능하다는 특징을 가지고 있으면서 기존의 WML1.0과 완벽하게 호환됩니다.

우리나라에서는 WML2로 제작한 문서를 불러오기 위해서는 SKT에서 Nate<sup>®</sup> Browser, LGT에서는 Lion 브라우저를 이용해서 사용하고 있습니다.



## 9.6.2 X-HTML 문서 작성 방법

### ● XHTML 구조

XHTML 구조는 HTML과 유사한 구조를 가지고 있습니다.  
단 XHTML을 정의해주는 DTD를 꼭 정의해주어야 합니다.

```
<?xml version="1.0" encoding="ks_c_5601-1987"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic
1.0//EN" "http://www.w3.org/TR/XHTML-basic/XHTML-
basic10.dtd" >
<html>
  <head> </head>
  <body title="제목">
    내용
  </body>
</html>
```

### ● XHTML 규칙

XHTML은 HTML과 유사하지만 규칙이 엄격하여 다음 사항을 지키지 않으며 오류가 발생하게 됩니다. XHTML 문서를 작성할 때 필요한 규칙에 대해서 살펴 보겠습니다.

▼ 여러 개의 태그를 함께 사용할 경우 먼저 열린 태그가 먼저 닫혀야 합니다.

[예]

틀린 예 - <i>italic <b>italic bold</i> bold </b>

바른 예 - <i>italic <b>italic bold</b></i> <b> bold </b>

▼ 닫기 태그가 없을 경우 '/>' 식으로 표시한다!

닫기 태그가 없이 열기 태그만 사용될 경우 '>' 앞에 '/' 표시를 하여 '<태그이름 />' 식으로 표시해야 합니다.

[예]

<br/> <hr/>

▼ 태그와 속성은 모두 영문 소문자로 표시한다!

XHTML은 대문자와 소문자를 구분하므로 모두 소문자로 표시합니다.

▼ 태그의 속성값은 항상 따옴표로 표시한다!

HTML에서는 태그의 속성값에 따옴표를 표시하지 않아도 되었지만 WML2에서는 속성값에 따옴표를 표시해야 합니다.

[예]



## ❖XHTML 문서 작성 예제

XHTML은 HTML 기반을 가지고 있다는 면에서 KUN과 유사하지만 스타일 (style)을 사용하지 않는 KUN에 비해 XHTML에서는 스타일을 사용할 수 있음.

여기서는 XHTML 문서 작성 예를 살펴보고, Openwave V7 시뮬레이터를 이용해서 작업 문서를 확인함.

### [예제] 표 예제

XHTML은 HTML4.0 형식으로 따르고 있으므로 홈페이지의 HTML과 제작 방법이 같습니다. 단 속성들은 스타일을 사용하기를 권장하고 있으므로 가급적 스타일을 이용하도록 합니다. 위 예제는 테이블을 이용해서 제목을 꾸민 예제입니다. 표의 경계선을 표시하고 싶을 경우에는 'style="border-style:solid"' 속성을 이용해서 꾸미며 휴대폰 액정의 크기가 제각기 다른 점을 감안하여 표의 너비는 % 단위로 설정하도록 합니다.



### [예제] 표 스타일 예제

표에 스타일을 적용하여 표의 왼쪽과 오른쪽 테두리를 안 보이도록 만든 예제. 그리고 표 시작과 끝에 <div> 태그에 align 속성을 사용해서 표를 정렬.



[예제] 메뉴 구성하기

하이퍼링크를 설정해서 메뉴를 구성한 예제. `accesskey` 속성을 이용해서 자판을 눌러 실행할 수 있는 등 설정 방법은 KUN과 동일. `<marquee>` 태그를 사용해서 글자를 흐르게 만들고 `<blink>` 태그를 사용해서 글자를 깜박이게 만들 수 있으며 `style="color:색상"` 속성을 이용하면 글자 색을 지정할 수 있음. 스타일을 설정한 글자에 커서를 위치하면 색상이 반전됨.



[예제] 문서에 그림 넣기

문서에 그림과 픽토그램을 삽입해서 꾸민 예제. 그림 삽입하는 방법은 WML과 동일. 픽토그램 경로는 브라우저마다 다르므로 사용할 브라우저에서 제시하는 픽토그램 경로에 맞추어 작업해야 함.



[예제] 배경 그림 넣기

문서 배경에 그림을 넣거나 테이블에 그림을 넣어서 꾸민 예제.

해당 태그에 스타일을 사용해서 'style="background-image:URL"' 식으로 입력해서 배경에 그림을 넣거나, 'style="background-color:색상;'" 식으로 입력해서 배경에 색을 지정 (NATE에서 이용하고 있는 구성임)



[예제] 입력양식 예제

입력양식을 이용해서 신상명세를 입력하는 예제

A screenshot of a form titled '신상명세 입력' (Personal Information Input). The form has a yellow background and a blue header. The fields and options are: '이름' (Name) with a text input field containing 'writting...'; '성별:' (Gender) with radio buttons for '남성' (Male) and '여성' (Female); '취미' (Hobbies) with checkboxes for '사진촬영' (Photography), '영화감상' (Movie watching), and '인라인' (Inline); and '나이' (Age) with a dropdown menu showing '30대' (30s). At the bottom are two buttons: '전송' (Send) and '재입력' (Re-input).

### 9.6.3 WML1.0을 WML2.0으로 변환하기

#### WML1.0을 WML2.0으로 변환하기

WML2.0 DTD에서는 기존의 WML1.0 문서도 불러 올 수 있는 기능을 제공함.

WML2.0을 지원하는 브라우저에서는 WML1.0의 소스를 WML2.0 환경에 맞추어 변환하여 문서를 불러옴.

이와 같이 문서를 변환하는 기능을 마이그레이션(Migration)이라고 함.

마이그레이션은, WML1.0의 기능을 고스란히 유지하면서 변환하는 방법과 XHTML로 변경하는 방법이 있으며,  
휴대폰에 설치되어 있는 브라우저마다 처리 방식이 다름.

XHTML 사용하기를 권장하는 네이트의 Nate<sup>®</sup>Browser 브라우저는 XHTML으로 마이그레이션을 수행함

#### ▶ WML2.0으로 DTD 선언

WML2.0 환경에 알맞도록 처리하기 위해 WML2.0 DTD 형식으로 변경해주어야 합니다.

```
<!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.0//EN"  
"http://www.wapforum.org/DTD/xhtml-mobile10.dtd">
```



```
<?xml version="1.0" encoding="ks_c_5601-1987"?>  
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML  
2.0//EN" "http://www.wapforum.org/DTD/wml20.dtd">
```

▶ 태그 변환

<wml> 태그는 <html>로 변경하고 <card> 태그는 <body> 태그로 변경.

만일 문서 안에 여러 개의 <card>를 사용했을 경우에는 페이지를 나누거나 <card> 태그 앞에 'wml:'을 입력해 줌.

이외에 기타 WML 태그들 앞에 'wml:'라고 표시해주면 WML2에서 인식됨.

<wml>        -> <html>  
<card>       -> <body>  
<WML태그>   -> <wml:WML태그>

[하나더] WML 확장 모듈

WML2.0은 XHTML MP와  
'wml:'를 붙여서 사용하는 WML 확장 모듈 모두를 지원함.

WML 확장 모듈을 사용할 경우 손쉽게 문서를 변환 가능.  
그러나 최근에는 XHTML을 사용하는 추세이므로,  
가급적 XHTML을 이용해서 작업하기를 권장함.

[결론]

WML2.0은 WML1.0보다 오히려 KUN과 유사한 형식을 가지고 있습니다.  
이는 모바일 언어가 유선 홈페이지의 HTML과 닮아가고 있다는 것을 의미하며  
결국 SKT, KTF, LGT 업체 간의 폰페이지 교환 뿐만 아니라 휴대폰으로 유선  
홈페이지를 볼 수 있음을 시사하고 있음을 알 수 있습니다.

실제로 Nate<sup>®</sup>Browser 개발업체인 인프라웨어(<http://www.infraware.co.kr>)  
에서 개발한 [Embider™ Browser 5] 브라우저에서는 유선 홈페이지를 고스란  
히 휴대폰으로 볼 수 있는 등 고도화된 화려한 기술을 제공하고 있습니다.



“이 교재는 2008년도 삼성전자 정보통신트랙과  
경북대학교 초일류 모바일-디스플레이산업  
인력양성사업단에 의하여 지원되었음.”