



13 웹(www)

쉽게 배우는 데이터 통신과 컴퓨터 네트워크

학습목표

- ✓ 웹 서비스를 위한 클라이언트-서버 구조를 이해
- ✓ 웹 서비스를 지원하는 APM(Apache, PHP, MySQL)의 연동 방식
- ✓ HTML이 지원하는 기본 태그 명령어와 프레임 구조를 이해
- ✓ HTTP의 요청/응답 메시지의 구조와 동작 원리를 이해
- ✓ CGI의 원리를 이해하고 FORM 태그로 사용자 입력을 처리하는 방식



1절. 웹 시스템 구조

- 웹 서비스: TCP 포트 80번
- 웹 브라우저(Web Browser)란? 웹문서 작성 방법? 동적 웹 페이지?

□ 클라이언트-서버 모델

- 클라이언트와 서버의 연결 설정 및 해제는 사용자가 새로운 URL 주소를 입력할 때 마다 반복
- URL(Uniform Resource Locator)
 - 웹 서버의 자원 명칭
 - 사용하는 프로토콜, 서버의 호스트 이름, 서버 내부의 파일 경로명으로 구성
 - 예: <http://www.korea.co.kr/welcom.html>
- UNIX/Linux 시스템
 - 로그인 이름 hong
 - 사용자 홈 디렉토리 /public_html/index.html
 - <http://www.korea.co.kr/~hong>
- 윈도우 시스템
 - <http://www.korea.co.kr/hong>



1절. 웹 시스템 구조

□ 클라이언트-서버 모델

■ 연결 설정과 해제 [그림 14-1]

1. 사용자가 웹 브라우저에게 웹 서버의 URL 주소 입력
2. 웹 브라우저가 DNS 서버에게 웹 서버의 호스트 이름을 IP 주소로 변경 요청
3. 웹 브라우저가 <IP 주소 + 포트 80번>의 웹 서버와 TCP 접속 시도
4. 웹 브라우저가 웹 서버에게 최초 화면을 위한 GET 명령 전송
5. 웹 서버가 웹 브라우저에게 요청한 웹 문서를 회신
6. 웹 브라우저와 웹 서버 사이의 연결 해제
7. 웹 브라우저가 사용자 화면에 웹 문서를 출력

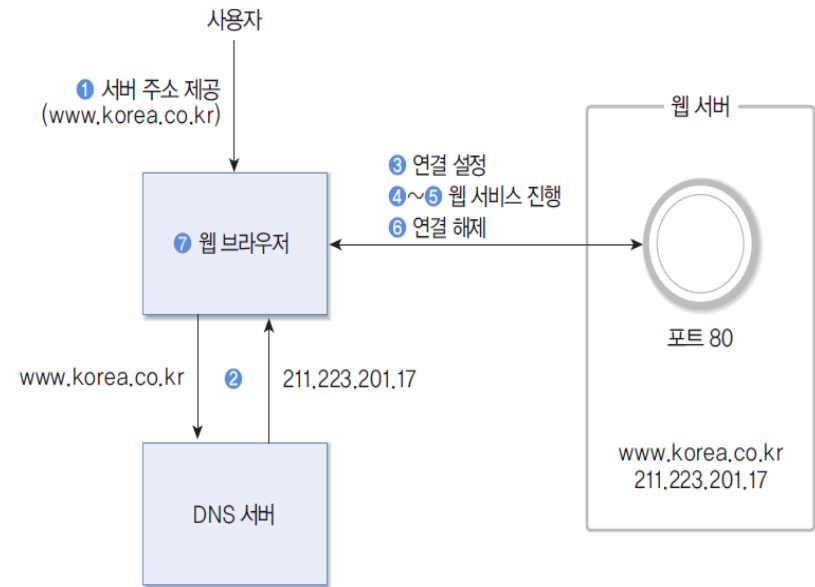


그림 13-1 웹 서비스의 동작 원리



1절. 웹 시스템 구조

□ APM에서의 서버와 클라이언트 구조

- APM (Apache, PHP, MySQL)
 - Apache: 웹 서버 프로그램
 - PHP(Personal Hypertext Preprocessor)
유닉스/리눅스 환경에서 지원되며, **HTML 언어의 기능을 보강, 동적 작업 수행**
 - MySQL: 데이터베이스 기능 지원

■ APM의 동작 원리

1. 웹 브라우저가 Apache에 웹 문서 요청
2. PHP 코드 처리 필요 시 **PHP에 요청**
3. 데이터베이스 처리 필요 시 **MySQL에 요청**
4. 데이터베이스 결과 회신
5. PHP가 실행 결과인 **HTML 코드 회신**
6. 웹 문서를 웹 브라우저에 회신

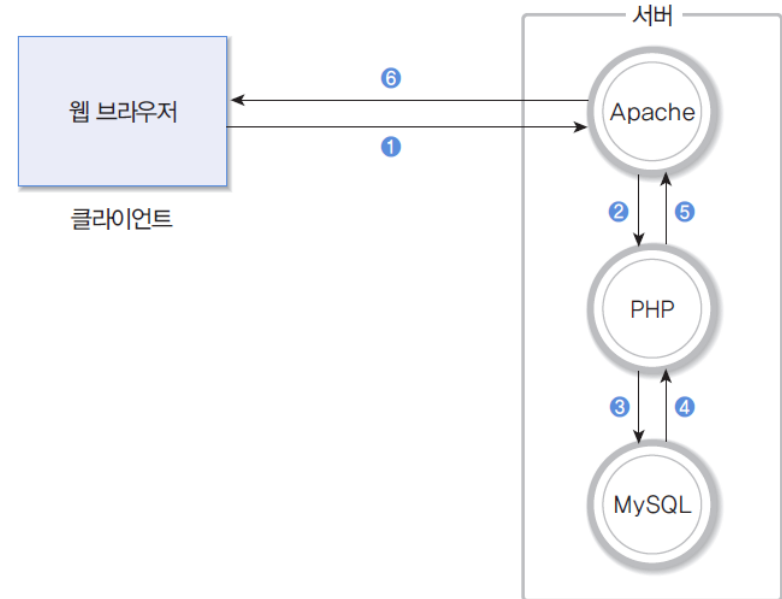


그림 13-2 APM의 동작 원리



1절. 웹 시스템 구조

□ APM에서의 서버와 클라이언트 구조

■ PHP 코드의 처리

- PHP 코드는 HTML 문서에 Embedded 형식으로 작성: **<? 와 ?> 구분자를 사용하여 HTML 코드와 구분**

– **PHP 코드가 포함된 문서의 확장자는 .php**

– 예

```
<HTML>
  <BODY>
    <? $value = 10; ?>
    출력<br>
    <? echo $value; ?>
  </BODY>
</HTML>
```

- **웹 브라우저에 회신되는 내용: PHP 코드는 서버(아파치)에서 실행되고 결과만 회신**

– 예

```
<HTML>
  <BODY>
    출력<br>
    10
  </BODY>
</HTML>
```



1절. 웹 시스템 구조

□ APM에서의 서버와 클라이언트 구조

- PHP 코드의 처리
 - 화면 출력 내용 [그림 14-3]



```
<HTML>
  <BODY>
    출력<br>
    10
  </BODY>
</HTML>
```



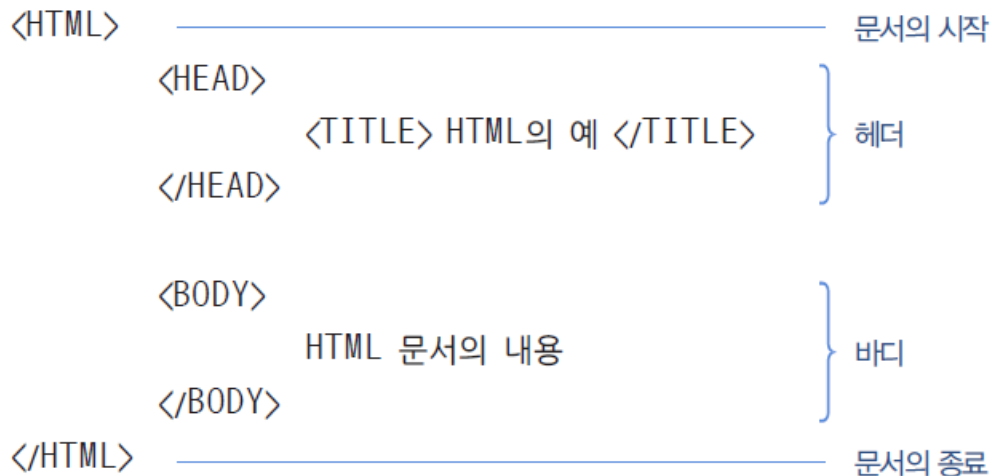
2절. HTML

□ HTML(HyperText Markup Language) 명령어

- HTML: 웹 문서를 작성하는 언어
- <HTML>로 시작하고, </HTML>로 종료
- 헤더: <HEAD>로 시작하고, </HEAD>로 종료
- 바디: <BODY>로 시작하고, </BODY>로 종료

예제 13-3

HTML 문서의 구조



2절. HTML

□ HTML 명령어

■ HTML 기본 명령어

- 태그 이름 좌우에 ‘<’ 와 ‘>’ 문자를 표기
- 시작 위치에는 태그 이름, 종료 위치에는 ‘/’ 문자와 태그 이름을 함께 표시

표 13-1 HTML 기본 태그

태그	설명
<Hn>	제목을 표시한다. n(1~6)은 제목 크기로, H1이 가장 크고 H6이 가장 작다.
<HR>	수평선을 표시한다.
<P>	문단을 구분한다.
 	다음 줄(행)로 넘어가는 줄 바꿈 기능을 수행한다.
	글자를 굵게 한다.
<I>	글자를 이탤릭으로 처리한다.
<U>	글자에 밑줄을 긋는다.
<SUB>	글자를 아래 첨자로 만든다.
	글자 크기와 색을 지정한다. n(1~7)은 글자 크기이고, fffffff(16진수)는 글자 색이다. 예를 들어, fffffff는 흰색, 000000은 검은색, ff0000은 빨간색, ffff00은 노란색이다.
	URL로 지정한 다른 웹 문서와 링크할 때 사용한다.
	지정된 그림 파일을 화면에 표시한다.



2절. HTML(태그 약자 설명- 참고)

HTML 태그 약자 설명

a href = Anchor(배에 달려있는 닻) Hypertext REFerence

img = IMaGe(이미지)

head = 머릿글

html = HyperText MarkupLanguage

b = Bold(굵은글씨)

i = Italic(이탤릭체.기울어진)

u = Underline(밑줄)

target = 표적

title = 표제

id = IDentification(신원확인)

http = HyperText Transfer Protocol

px = PiXel 픽셀

pt = PoinT 포인트

row = 행

row = 행

col = COLumn 열

bg = BackGround 배경

th = Table Header

tr = Table Row

td = Table Data

Alink = Active LINK (활성화된 링크)

Vlink = Visited LINK (방문한 링크)

br = BReak

dd = Definition Data 데이터(용어) 정의

div = DIVision (분류)

embed = EMBEDed (끼워넣다)

h1, h2 h3... = Heading 1

ul = Unordered List(순서없는 목록)

ol = ordered List(순서있는 목록)

li = Listed Item (항목)

p = Paragraph (새로운 문단)



2절. HTML

□ HTML 명령어

■ HTML 문서 작성의 예

– 그림 14-4

<HTML>

<HEAD>

<TITLE> 기본 웹 문서 </TITLE>

</HEAD>

<BODY>

<H2>기본 태그 연습</H2>

<HR>

굵은 글씨

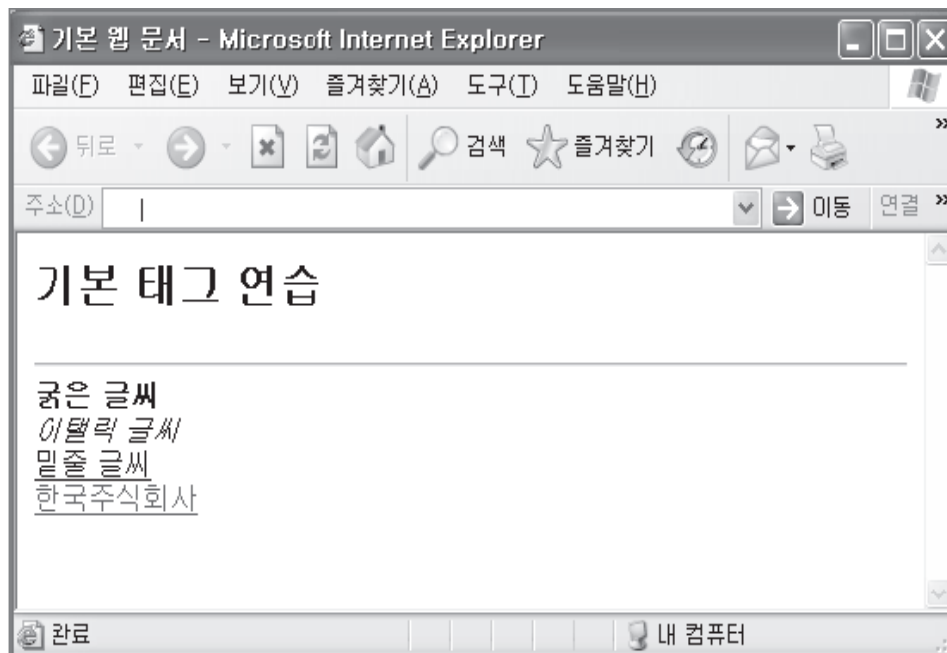
<I>이탤릭 글씨</I>

<U>밑줄 글씨</U>

한국주식회사

</BODY>

</HTML>

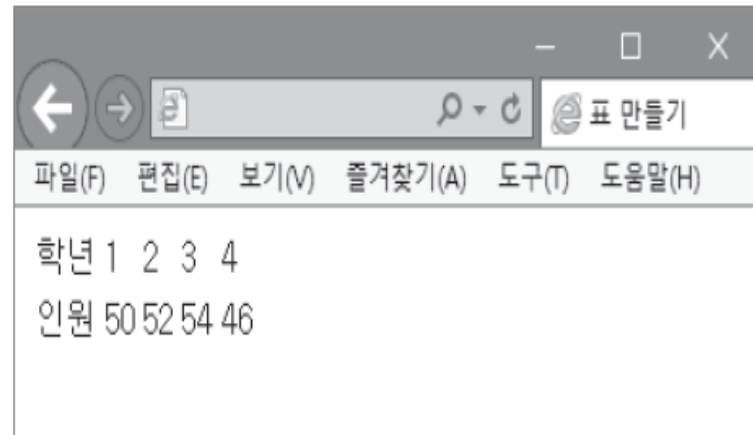


2절. HTML

□ HTML 명령어

■ HTML 표 명령어

- <TABLE>: 테이블 작성
- <TR>: 테이블의 한 줄
- <TD>: 테이블의 한 칸



<HTML>

<HEAD>

<TITLE> 표 만들기 </TITLE>

</HEAD>

<BODY>

<TABLE>

<TR> <TD>학년</TD> <TD>1</TD> <TD>2</TD> <TD>3</TD> <TD>4</TD> </TR>

<TR> <TD>인원</TD> <TD>50</TD> <TD>52</TD> <TD>54</TD> <TD>46</TD> </TR>

</TABLE>

</BODY>

</HTML>



2절. HTML

□ 프레임(frame)

- 한 화면을 여러 프레임으로 나누는 기능을 제공
- <FRAMESET> 태그

– 좌우 1:9의 예 [그림 14-6]

<HTML>

<FRAMESET cols="10%,*" border=5>

<FRAME name="left" src="left.html">

<FRAME name="right" src="right.html">

</FRAMESET>

</HTML>

– left.html

<HTML> <BODY> left </BODY> </HTML>

– right.html

<HTML> <BODY> right </BODY> </HTML>

