
웹 프로그래밍 개요

정종욱 jongukjeong@gmail.com

웹 프로그래밍 개요

1. 인터넷(네트워크통신) 의 이해
2. HTTP
3. URL
4. 웹 프로그래밍
5. 웹 애플리케이션

1. 인터넷(네트워크 통신)의 이해

□ 인터넷의 시작

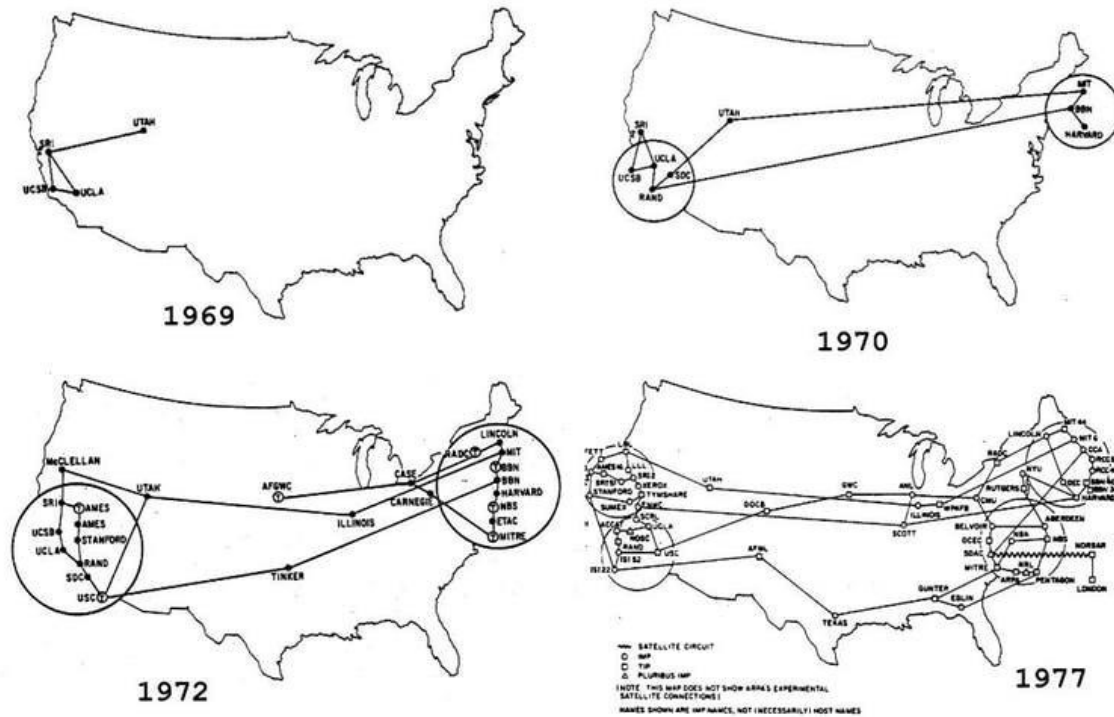
- 예전부터 과학자들은 정보가 연결되어 있는 세계를 꿈꿔 왔음
- 냉전 시대인 **1960년대**, 소련에서 최초로 인공위성 발사에 성공하자 위기를 느낀 미국은 신기술 연구를 위한 **ARPA** 부서를 창설(미 국방부 주도)

➤ 1969년 ARPANET 개발

➤ ARPANET의 주요 목적

- 정보 교환과 연구 역량 극대화 도모
- 네트워크를 분산, 하나가 단절되더라도
- 나머지에 영향이 없도록 하고자 함
- 군사 목적의 **ARPANET**은 대학, 기업과 연계 발전해 나감
- 이 중, 민간 연구용이 현재의 인터넷으로 발전

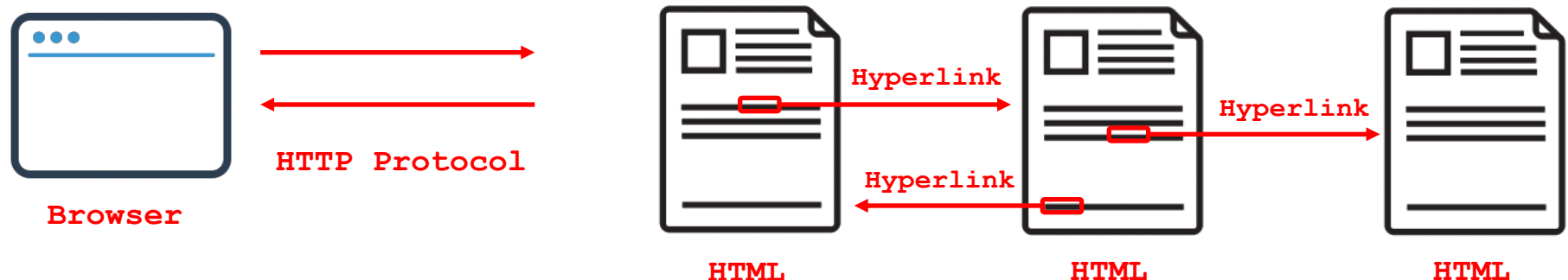
연도별 ARPANET 연결 상황



1. 인터넷(네트워크 통신)의 이해

❑ World Wide Web(WWW)의 발명

- 유럽입자물리연구소(CERN)의 팀 버너스-리(Tim Berners-Lee)가 고안1989년 HTML 개발
- **Hyperlink** 개념을 도입한 문서 형식인 HTML(HyperText Markup Language) 도입, 인터넷 상의 논문을 연결
(Hyperlink : 문서와 문서 사이를 링크로 연결, 빠르게 이동할 수 있는 개념)
- 문서에 주소를 붙이는 **URL**(Uniform Resource Locator) 개념 확립
- 인터넷에 연결된 컴퓨터 사이에서 메시지를 주고받을 수 있는 통신 규약 HTTP 고안
- WWW : 전 세계 인터넷상의 콘텐츠를 연결, 공유할 수 있는 인터넷 기반 서비스



1. 인터넷(네트워크 통신)의 이해

□ 인터넷 != WWW(World Wide Web)

인터넷 기반의 대표 서비스 중 하나

이름	프로토콜	포트	기능
WWW	HTTP	80	웹서비스
Email	SMTP/POP3/IMAP	25/110/114	이메일 서비스
FTP	FTP	21	파일 전송 서비스
DNS	DNS	23	네임서비스
NEWS	NNTP	119	인터넷 뉴스 서비스

□ 인터넷 (Internet)

TCP/IP 기반의 네트워크가 전세계적으로 확대되어 하나로 연결된 네트워크들의
네트워크 (네트워크의 결합체)

1. 인터넷(네트워크 통신)의 이해

□ TCP / IP

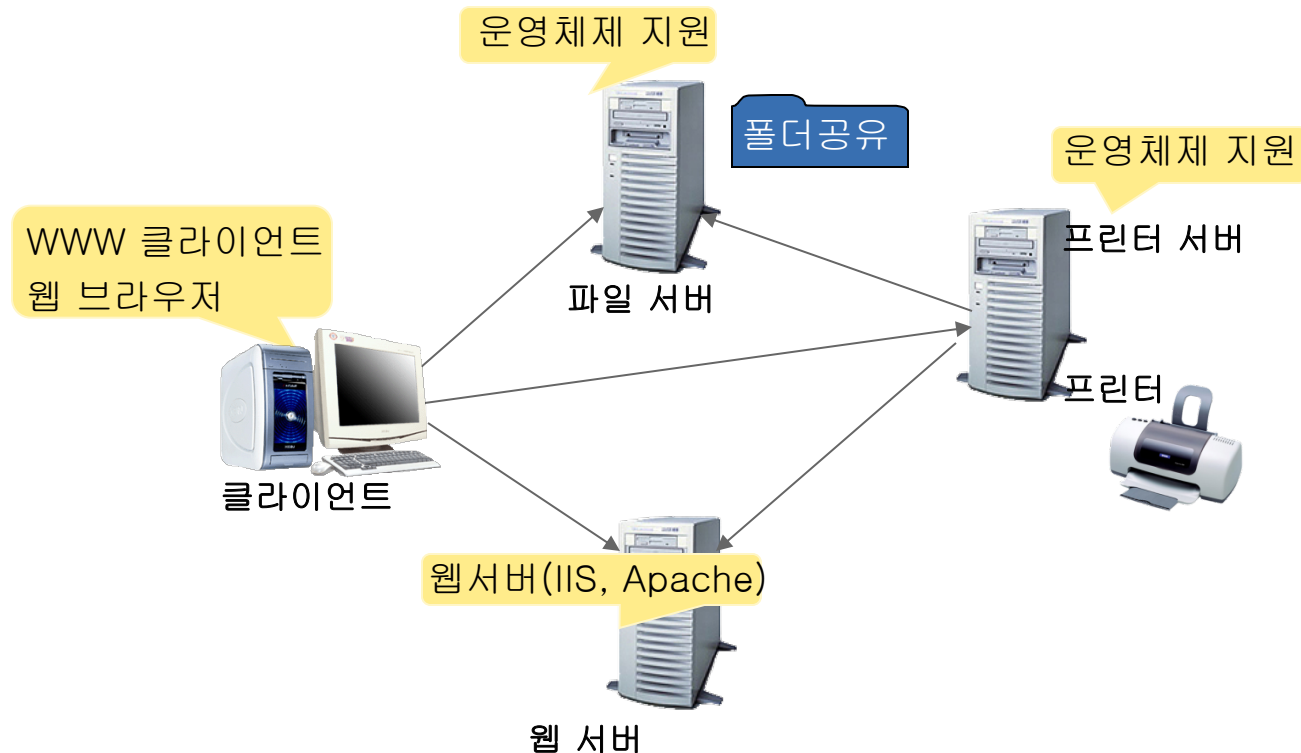
- 하드웨어, 운영체제, 접속 매체와 관계없이 동작할 수 있는 개방형 구조
- OSI 7 계층에서 4계층으로 단순화.

OSI 7계층		TCP/IP 4계층	
응용 계층	표현 계층	응용 계층	• 네트워크를 사용하는 WWW, FTP, 텔넷, SMTP 등의 응용 프로그램으로 구성.
세션 계층	전송 계층		
네트워크 계층	데이터 링크 계층	전송 계층	• 도착지까지 데이터를 전송 • 각각의 시스템을 연결 • TCP 프로토콜을 이용하여 데이터를 전송
물리 계층		인터넷 계층	• 데이터를 정의 및 경로 지정 • 정확한 라우팅을 위해 IP 프로토콜을 사용 • IP 주소가 위치하는 계층
		물리 계층	• 물리적 계층 즉 이더넷 카드와 같은 하드웨어

1. 인터넷 (네트워크 통신)의 이해

□ 서비스 (클라이언트 / 서버 통신)

- 서버 : 네트워크에서 서비스를 제공하는 컴퓨터
- 클라이언트 : 네트워크에서 서비스를 제공받는 컴퓨터
- 하드웨어적 구분은 사실상 없음



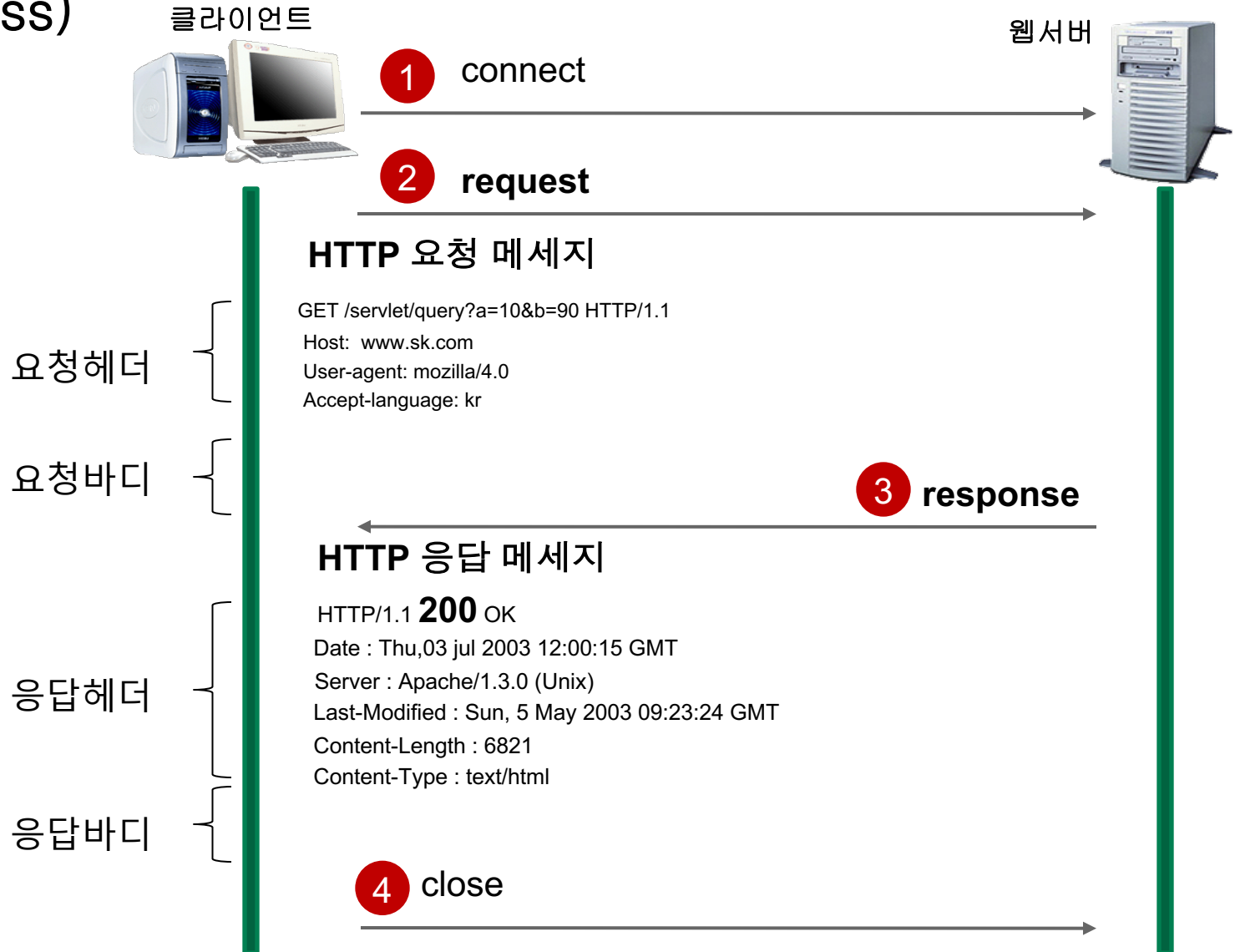
2. HTTP

□ HTTP (Hyper Text Transfer Protocol)

- WWW 서비스를 위한 TCP/IP 응용계층 프로토콜 중 하나
- 웹 서버 와 클라이언트는 HTTP를 이용한 통신
- 무 상태 연결(stateless connection)

2. HTTP

□ HTTP (stateless)



3. URL

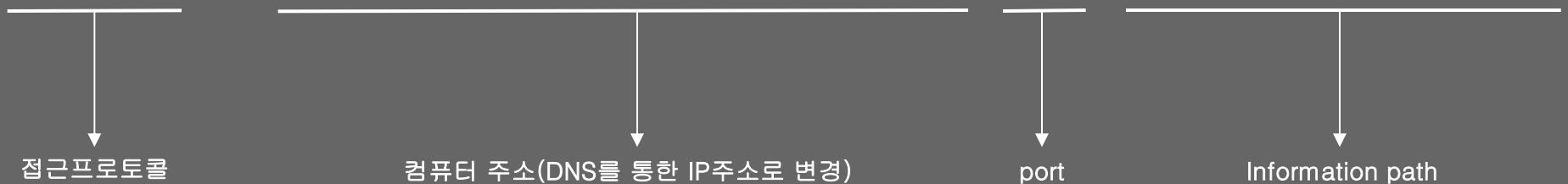
□ URL (Uniform Resource Locator)

- 인터넷 상의 자원의 위치
- 특정 웹 서버의 특정파일에 접근하기 위한 경로 혹은 주소

- 프로토콜(Protocol) : 네트워크상에서 약속한 통신규약 (Http, FTP, SMTP, POP, DHCP)
- IP : 네트워크상에서 컴퓨터를 식별할 수 있는 주소
- DNS : IP주소를 인간이 쉽게 외우도록 맵핑한 문자열
- Port : IP주소가 컴퓨터를 식별할 수 있게 해준다면, Port번호는 해당컴퓨터의 구동되고 있는 프로그램을 구분할 수 있는 번호

네이버 웹사이트

http://www.naver.com:80/index.html



3. URL

HTTP 프로토콜로 웹 서버에 액세스 하는 경우

http://user.password@www.cyber.co.kr:80/dir/file.html

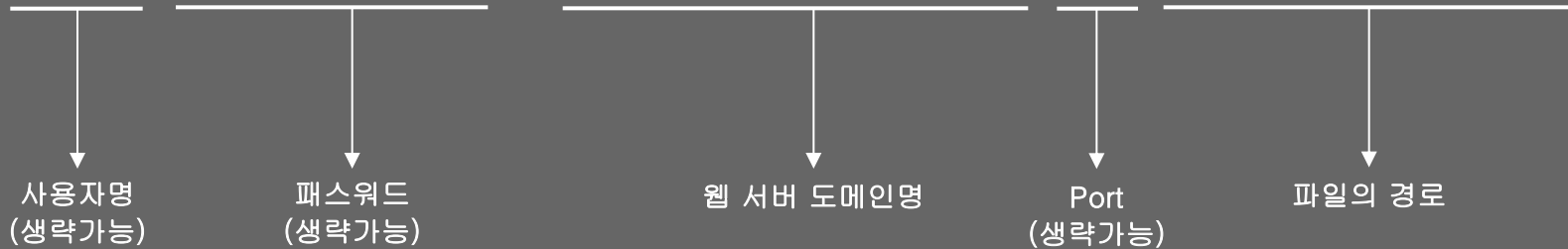


- ❑ 도메인명이란 www.cyber.co.kr과 같이 마침표(.)로 구분하여 표현하는 이름이라고 알아두자.
- ❑ 포트번호란 액세스 대상 서버 애플리케이션을 지정하여 사용하는 번호라고 생각하면된다.

3. URL

FTP 프로토콜로 파일을 다운로드 하거나 업로드 하기위해 액세스 하는 경우

`http://user.password@ftp.cyber.co.kr:21/dir/file1.html`



클라이언트 PC 자체의 파일에서 데이터를 읽어오는 경우

`file://localhost/c:/path/file1.zip`



3. URL

이메일을 송신하는 경우
mailto:tone@cyber.co.kr



A horizontal line is drawn below the email address 'mailto:tone@cyber.co.kr'. A vertical line descends from the center of this horizontal line, ending in a downward-pointing arrowhead.

이메일 주소

뉴스 그룹의 기사를 읽는 경우
news:comp.protocols.tcp-ip



A horizontal line is drawn below the newsgroup name 'news:comp.protocols.tcp-ip'. A vertical line descends from the center of this horizontal line, ending in a downward-pointing arrowhead.

뉴스그룹명

3. URL

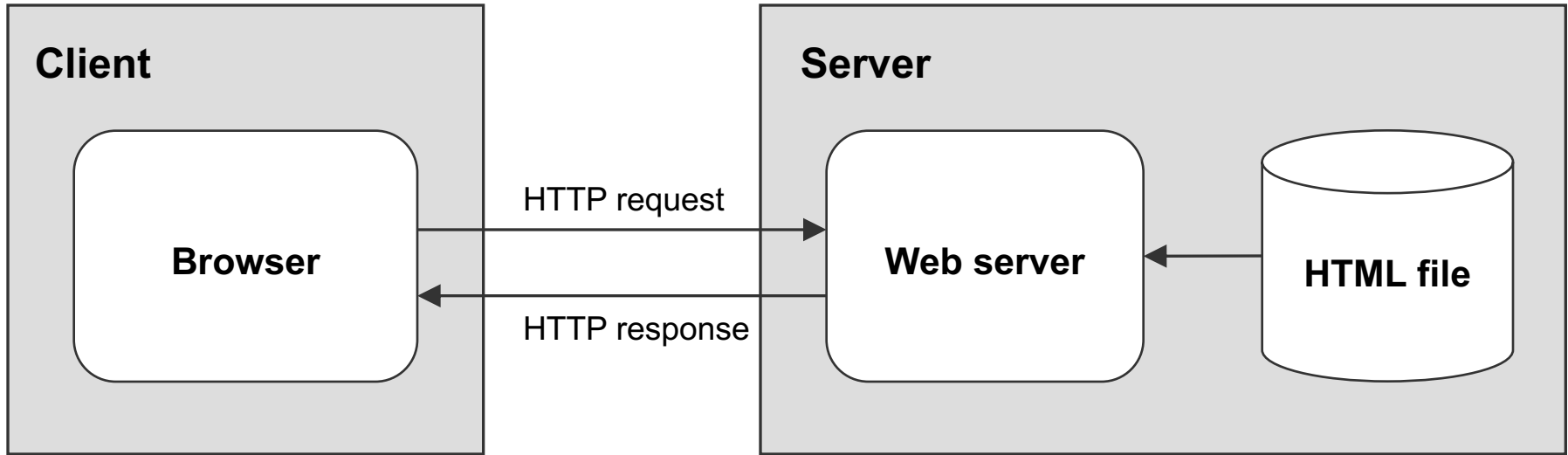
scheme:[//[user:password@]host[:port]][/]path[?query][#fragment]

프로토콜 : //호스트네임.도메인네임/경로/파일

- ❑ 프로토콜: 컴퓨터 사이의 원활한 데이터 전송을 위한 통신규약. **Ex) http, ftp, telnet**
- ❑ 호스트네임: 인터넷에 연결된 컴퓨터 이름. **Ex) www(월드와이드웹)**
- ❑ 도메인네임: 호스트네임의 일부분으로서 호스트네임을 더욱 세분화해서 식별한다
- ❑ 보통 도메인이라 줄여서 부른다 **Ex) com, kr, net**
- ❑ 디렉토리: 경로 표시. '/' 로 구분하며, 일반적으로 생각하는 카테고리 개념이다.

4. 웹 프로그래밍

□ 정적인(static) 웹 페이지 (~~웹 프로그래밍~~, 퍼블리싱)



웹 페이지는 **HTML** 이라는 표준 마크업랭귀지로 작성.

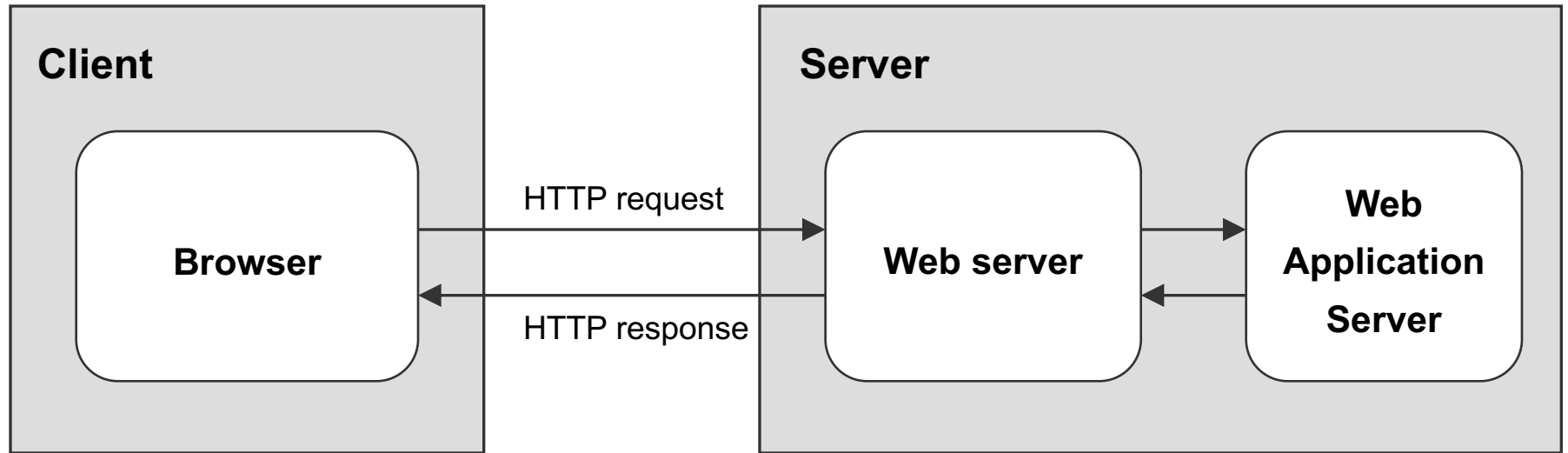
4. 웹 프로그래밍

□ 정적인(static) 웹 페이지 처리 방식

- *HTML(Hypertext Markup Language)*은 브라우저가 웹 페이지로 변환하는 언어이다.
- 정적인 웹 페이지는 파일 형태로 저장되어 있으면서 사용자의 입력에 따라 변하지 않는 HTML 문서이다.
- *HTTP(Hypertext Transfer Protocol)*는 웹 브라우저와 웹 서버가 통신하는 프로토콜이다.
- 웹 브라우저는 *HTTP 요청(HTTP request)* 메시지를 서버에 전달함으로써 웹 서버의 페이지를 요청한다.
- 웹 서버는 *HTTP 응답(HTTP response)* 메시지를 전달함으로써 HTTP 요청에 응답한다.
정적인 웹 페이지에서는 HTTP 응답이 HTML 문서를 포함한다.

4. 웹 프로그래밍

□ 동적인 웹 페이지를 만들어 내는 모든 기술



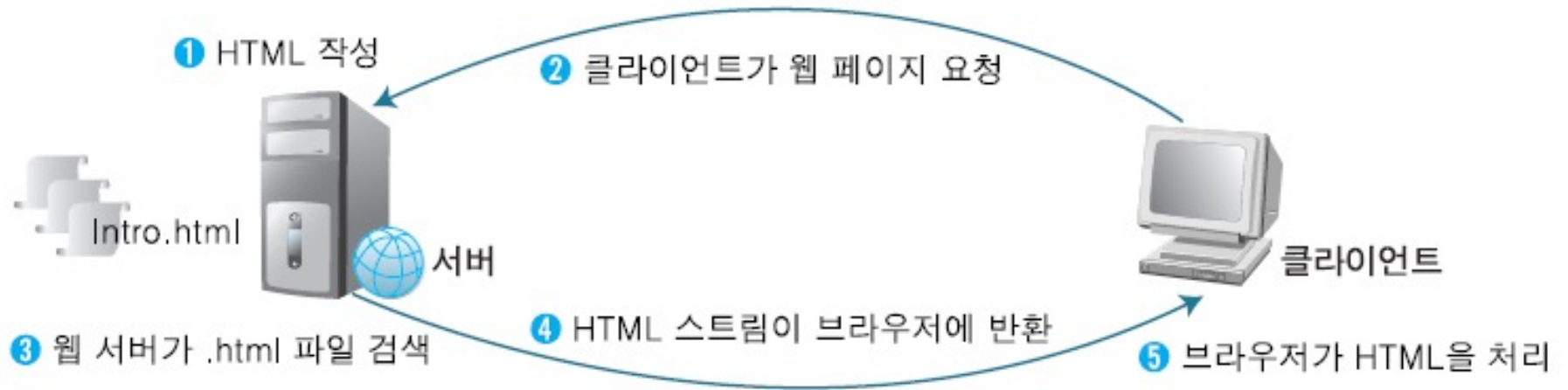
4. 웹 프로그래밍

□ 동적인(dynamic) 웹 페이지 처리 방식

- 동적인 웹 페이지는 웹 어플리케이션에 의해 생성되는 HTML 문서이다. 웹 브라우저가 웹 어플리케이션에 전달한 파라미터 값에 따라 웹 페이지가 변한다.
- 웹 서버가 동적인 웹 페이지에 대한 요청을 받으면 서버는 웹 어플리케이션으로 요청을 넘긴다. 그러면 어플리케이션이 HTML 문서를 생성하여 웹 서버로 결과를 전달한다.
- 웹 서버는 HTML 문서를 HTTP 응답(HTTP response)으로 감싼 후 브라우저로 결과를 전달한다.
- 전달 받은 HTML 문서가 정적인 HTML 파일에서 왔는지 아니면 웹 어플리케이션에 의해 동적으로 생성된 문서인지 브라우저는 알지 못한다. 어느 쪽이든 브라우저는 전달받은 HTML 문서를 화면에 표시한다.

4. 웹 프로그래밍

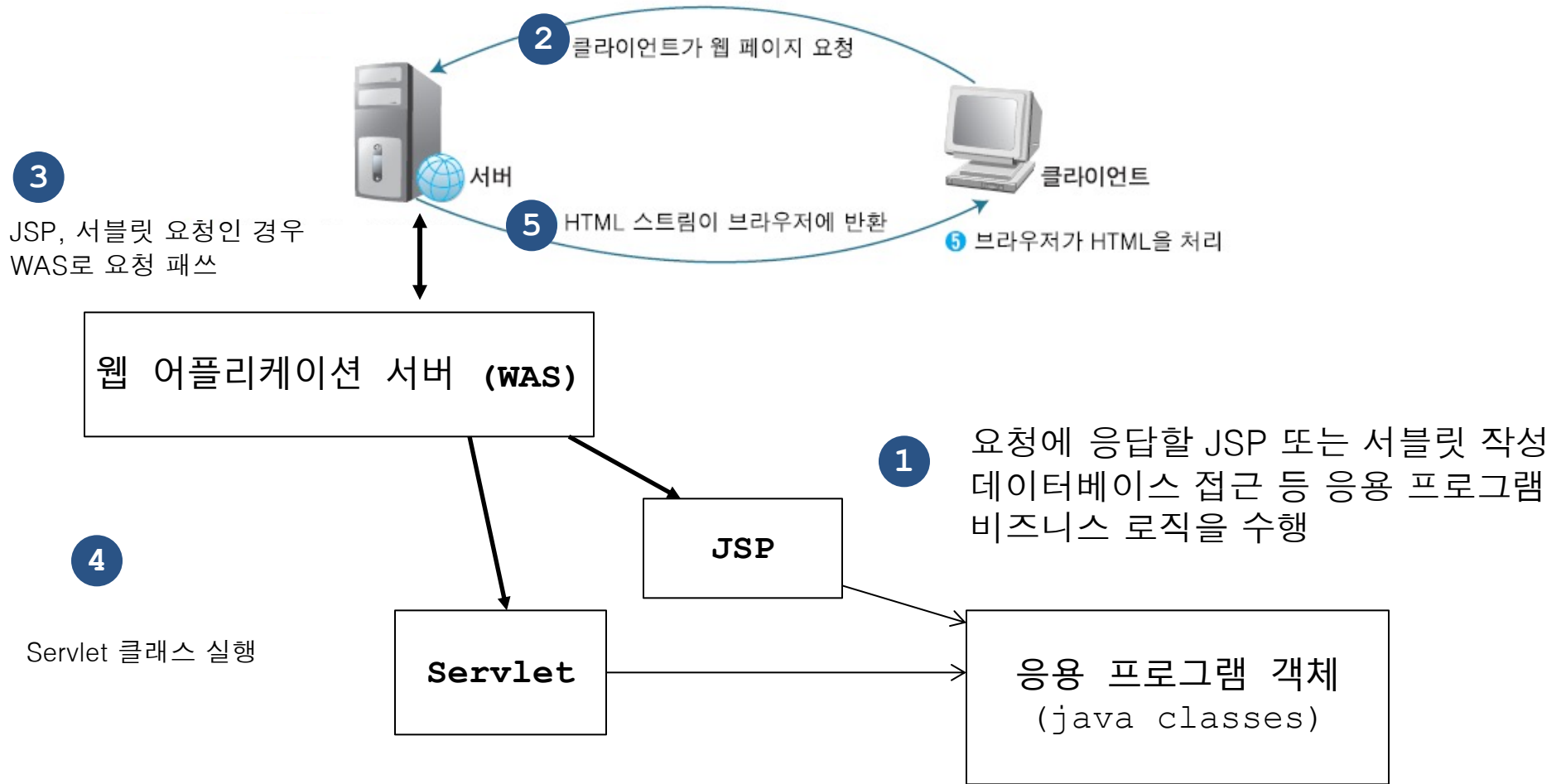
□ 정적 웹페이지 VS 동적 웹페이지



[정적 페이지 접근 시 웹 문서 전송]

4. 웹 프로그래밍

□ 정적 웹페이지 VS 동적 웹페이지



[동적 페이지 접근 시 웹 문서 전송]

5. 웹 애플리케이션

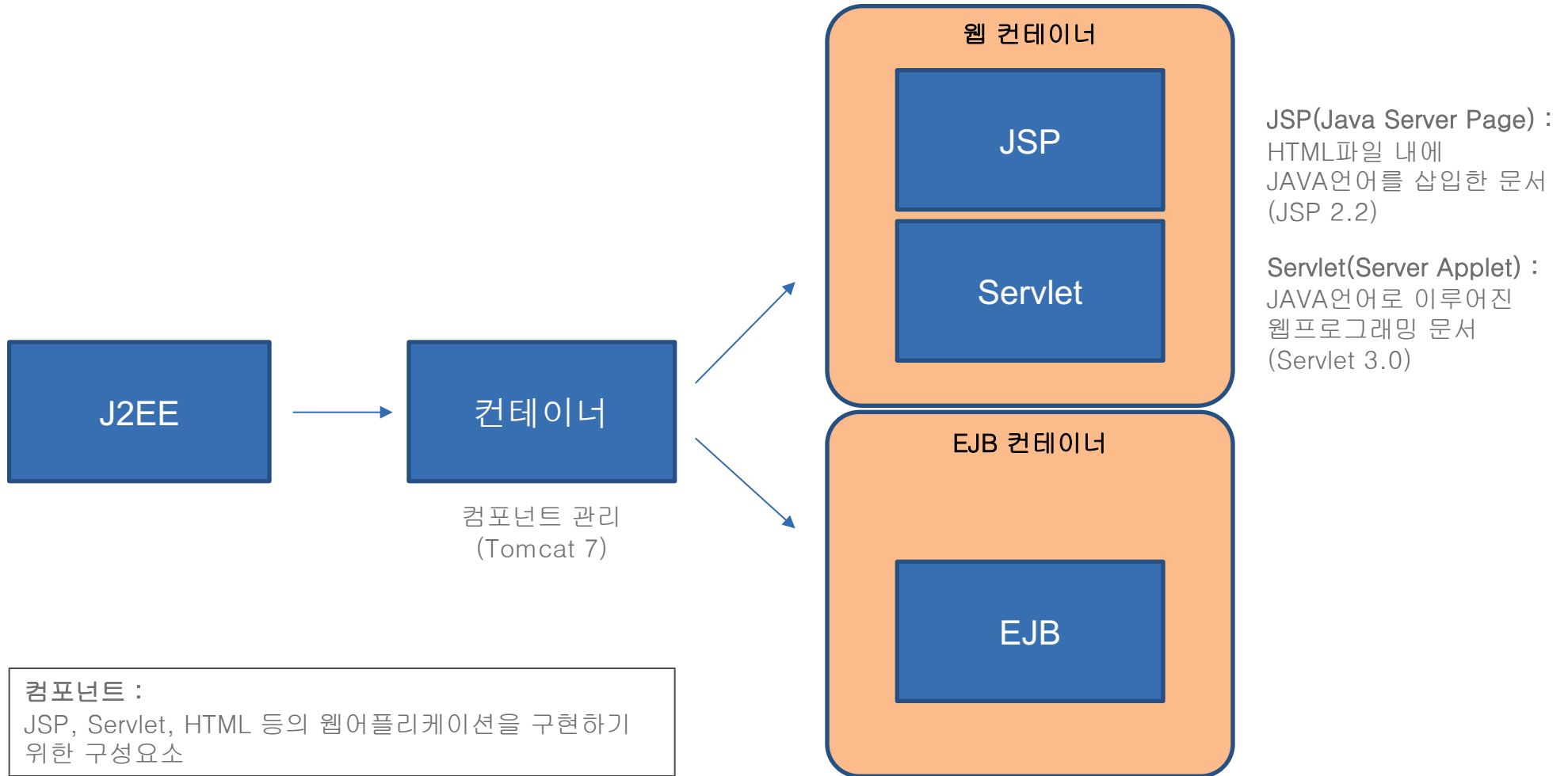
□ 웹 애플리케이션에 필요한 구성요소

- Java 웹 애플리케이션은 JSP와 서블릿으로 구성된다.
- *서블릿 & JSP 엔진* 또는 *서블릿 & JSP 컨테이너*는 서버에서 서블릿과 JSP를 구동할 수 있게 하는 소프트웨어이다.
- J2EE(Java 2 Platform, Enterprise Edition)는 웹 서버와 *서블릿 & JSP 엔진*이 어떻게 상호 작용해야 하는지를 명세하고 있다.
- *서블릿 & JSP 엔진*이 동작하기 위해서는 SDK에 접근해야 한다.
- *EJB(Enterprise Java Beans)*를 사용하는 웹 애플리케이션은 *EJB container* 로 알려진 추가적인 서버 컴포넌트가 필요하다.

5. 웹 애플리케이션

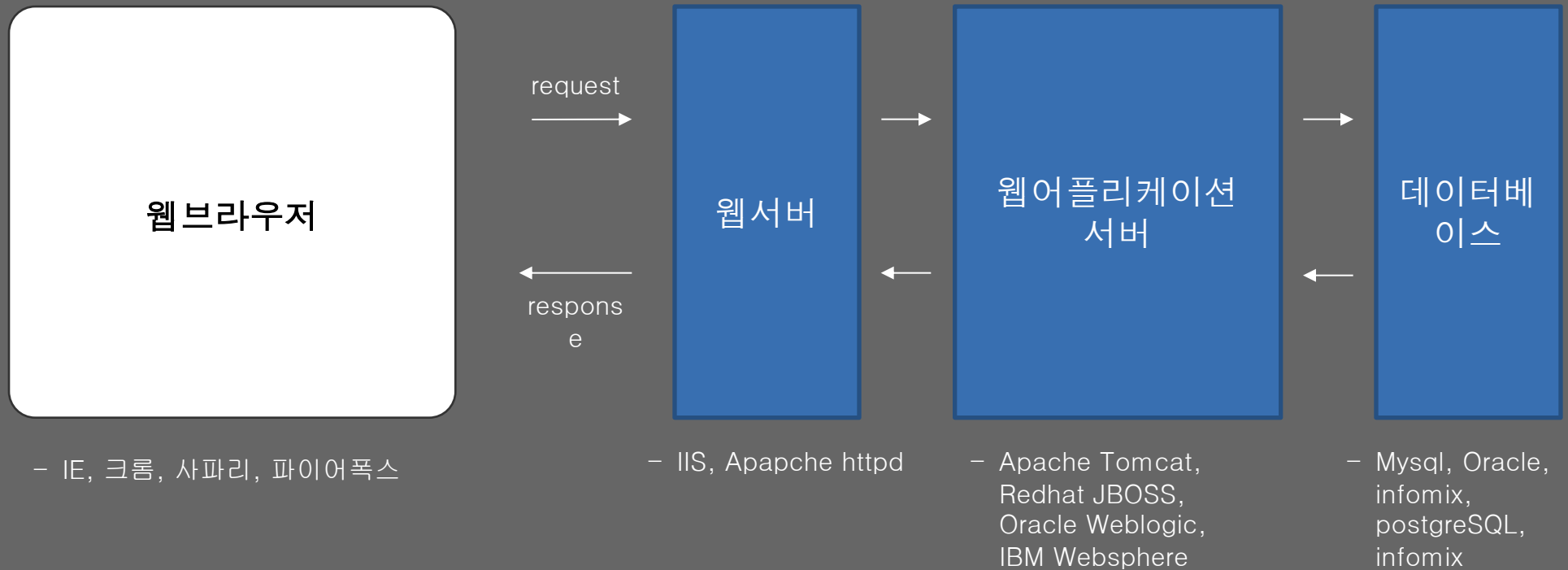
JAVA 웹이란

JAVA플랫폼(J2SE, J2EE, J2ME)중에서 J2EE를 이용한 웹프로그래밍



5. 웹 애플리케이션

- 웹서버 : 클라이언트의 요청에 의해 정보를 제공해 주는 서버 (Aphach, IIS). 별도의 구현이 필요한 로직이 있을 경우 웹어플리케이션 서버에 요청.
- 웹브라우저 : 웹서버에 정보를 요청하고, 웹서로부터 정보를 받는 매개체. 이때 HTTP 프로토콜을 사용함.



5. 웹 애플리케이션

□ JSP 코드

```
<head>
  <title>lecture 4 - Email List application</title>
</head>
<body>
  <%
    String firstName = request.getParameter("firstName");
    String lastName = request.getParameter("lastName");
  %>
  <h1>Thanks for joining our email list</h1>
  <p>Here is the information that you entered:</p>
  <table cellpadding="5" cellspacing="5" border="1">
    <tr>
      <td align="right">First name:</td>
      <td><%= firstName %></td>
    </tr>
    <tr>
      <td align="right">Last name:</td>
      <td><%= lastName %></td>
    </tr>
  </table>
```


5. 웹 애플리케이션

□ JSP (Java Server Page)

- *JSP(JavaServer Page)*는 HTML 코드 내에 Java 코드를 포함하는 형태로 구성된다.
- JSP 페이지가 처음 호출되었을 때, JSP 엔진은 JSP 코드를 서블릿으로 변환하고, 컴파일 한다. 그리고 서블릿 엔진이 서블릿을 구동한다.

5. 웹 애플리케이션

□ 서블릿 코드

```
public class EmailServlet extends HttpServlet{

    public void doGet(HttpServletRequest request,
                        HttpServletResponse response)
                        throws IOException, ServletException{

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        String firstName = request.getParameter("firstName");
        String lastName = request.getParameter("lastName");

        out.println(
            "<html>\n"
            + "<head>\n"
            + "  <title>lecture 5 - Email List application </title>\n"
            + "</head>\n"
            + "<body>\n"
```

5. 웹 애플리케이션

□ 서블릿 코드 (계속)

```
\n\n+ "<h1>Thanks for joining our email list</h1>\\n"\n+ "<p>\n    Here is the information that you entered:\n</p>\\n"\n+ "  <table cellpadding=\\\"5\\\"\n    cellspacing=\\\"5\\\"\n    border=\\\"1\\\">\\n"\n+ "    <tr><td align=\\\"right\\\">First name:</td>\\n"\n+ "        <td>" + firstName + "</td>\\n"\n+ "    </tr>\\n"\n+ "    <tr><td align=\\\"right\\\">Last name:</td>\\n"\n+ "        <td>" + lastName + "</td>\\n"\n+ "    </tr>\\n"\n+ "  </table>\\n"\n+ "</html>);
```

5. 웹 애플리케이션

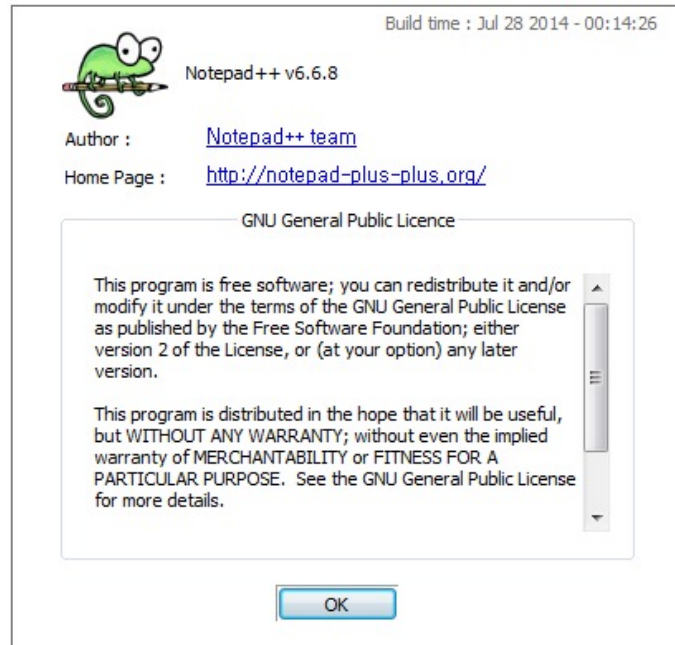
□ 서블릿(Servlet)

- 서블릿(servlet)은 서버에서 동작하는 Java 클래스이다.
- 서블릿은 HttpServlet 클래스를 상속한다.
- HTML 코드를 브라우저로 리턴하기 위해서, 서블릿은 out 객체의 println 메소드를 사용한다.
이것은 HTML 코드를 작성하기 어렵게 만드는 요소이다.
- 서블릿과 JSP로부터 최상의 결과를 얻으려면, 웹 페이지를 개발할 때 이 두가지 컴포넌트 (JSP, 서블릿)를 조화롭게 사용해야 한다. 웹 페이지를 구성하는 화면(HTML)은 JSP로 표현하고 프로세스에 관련한 부분은 서블릿이 처리하도록 해야한다.

5. 웹 애플리케이션

필요한 학습

1. JAVA : JAVA웹어플리케이션을 구현하기 위한 선행 학습 필요
2. HTML : 웹어플리케이션을 구현하기 위한 기본 언어
3. JavaScript : 클라이언트 기능을 구현하기 위한 언어
4. JQuery : JavaScript의 대표적인 라이브러리, 클라이언트 사이드 스크립트 언어를 단순화
5. CSS : 웹어플리케이션의 레이아웃 및 스타일을 지정하는 언어



<https://www.visualstudio.com/downloads/download-visual-studio-vs>

웹문서 제작 팁

웹문서를 제작하면서 반드시 알아야 하는 내용은 아니지만, 알고 있으면 손발이 편해지는 팁들에 대해서 살펴봅니다.

1. 편집 툴

- visual studio express 2013 for web :
<https://www.visualstudio.com/downloads/download-visual-studio-vs>
- notepad++ :
http://search.naver.com/search.naver?sm=tab_hyt.top&where=nexearch&ie=utf8&query=notepad%2B%2B
- editplus (유료)
- dreamweaver (유료)

2. 하위 버전 브라우저 테스트 툴

- download : <http://www.my-debugbar.com/wiki/IETester/HomePage>

3. 많은 양의 더미 텍스트가 필요한 경우

- <http://lipsum.com>

4. 단위 변환(px, em, %, point)

- <http://pxtoem.com/>

5. 전체적인 레이아웃을 제작할때 필요한 가상의 이미지

- <http://placeholder.it/> (ex. <http://placeholder.it/200x100>)

웹문서 제작 팁

웹문서를 제작하면서 반드시 알아야 하는 내용은 아니지만, 알고 있으면 손발이 편해지는 팁들에 대해서 살펴봅니다.

6. W3C(월드와이드웹 컨소시엄, 활동사항, 기술동향 등의 정보 제공)

- <http://www.w3.org/>

7. 다양한 디바이스 환경에 어울리는 미디어쿼리

- <http://mediaqueri.es/>

8. 웹 접근성 관련

- <http://www.wah.or.kr/> : 웹 접근성과 관련하여 권고되는 사항들의 자료 제공

9. w3schools

- www.w3schools.com
- HTML 정보 사이트로, HTML, XML, 브라우저, 서버 스크립트에 대한 정보를 제공.
- 다양한 튜토리얼 문서를 경험 할 수 있음.

10. 색상 정보 제공

- <http://www.colorpicker.com>