

1강

웹과 자바

방송대 컴퓨터과학과 김희천 교수

목차

- 1 웹 관련 용어
- 2 웹 애플리케이션의 실행과 구성 요소
- 3 JSP의 특징 및 관련 개념
- 4 HTTP 프로토콜



01

웹 관련 기본 용어

➤ 웹의 의미

- 팀 버너스 리가 제안
- 인터넷 기반의 정보 공유 서비스
 - World Wide Web, WWW 또는 W3라고도 함
 - 인터넷 상에 분산되어 존재하는 다양한 정보를 통일된 방법으로 찾아볼 수 있게 하는 **정보 서비스**
- 인터넷 기반의 전 세계적 **정보 공유 공간**
 - 분산된 웹 서버에 존재하는 하이퍼텍스트(hypertext) 문서들이 서로 연결된 시스템



1) 웹(2/2)

1. 웹 관련 기본용어

➤ 하이퍼텍스트

- 웹 상에 존재하는 문서 형태로서 텍스트 외에 이미지, 동영상 등의 멀티미디어 요소를 포함
- 하이퍼링크(hyperlink)를 통해 다른 하이퍼텍스트와 연결됨
 - 문서와 문서가 연결된 구조
- 웹 브라우저를 이용해 하이퍼텍스트를 볼 수 있음
- HTML로 표현되며 HTTP 프로토콜을 사용하여 전송됨

➤ W3C

- World Wide Web Consortium의 약자
- 국제 웹 표준화 기구
 - HTML, HTTP, XML 등의 웹 관련 기술을 관리함



2) 웹 문서(1/2)

1. 웹 관련 기본용어

▶ 정적 웹 문서

- 정적인 텍스트로 문서의 내용이 바뀌지 않음
- 클라이언트가 서버에게 정적 웹 문서를 요청하면, 서버는 해당 문서를 찾아서 전달함
- 정적 웹 문서를 요청하면 항상 동일한 결과가 전달됨

```
1 <HTML>
2   <HEAD>
3     <TITLE>내 홈 페이지</TITLE>
4   </HEAD>
5   <BODY>
6     <h2>방문을 환영합니다.</h2>
7   </BODY>
8 </HTML>
```

2) 웹 문서(2/2)

1. 웹 관련 기본용어

> 동적 웹 문서

- 동적 웹 문서를 사용하여 최종 결과가 동적으로 만들어짐
 - 클라이언트 측 처리와 서버 측 처리 방식이 있음
- JSP는 동적 웹 문서를 작성하는 기술임

```
1  <%@ page language="java" contentType="text/html; charset=euc-kr" %>
2  <HTML>
3  <HEAD>
4    <TITLE> Hello JSP! </TITLE>
5  </HEAD>
6  <BODY>
7    지금 시간은 :
8    <%
9      out.print(new java.util.Date( ));
10   %>
11 </BODY>
12 </HTML>
```

02

웹 애플리케이션의 실행과 구성 요소

1) 웹 애플리케이션

2. 웹 애플리케이션의 실행과 구성 요소

➤ 웹 애플리케이션

- 웹에서 실행되는 응용 프로그램

➤ 웹 애플리케이션의 구분

- 클라이언트 측 실행/서버 측 실행
- 컴파일 방식/비컴파일 방식



2) 웹 애플리케이션의 실행(1/2)

2. 웹 애플리케이션의 실행과 구성 요소

➤ 웹 서비스의 제공과 구성 요소

■ 웹 클라이언트

- 웹 브라우저는 웹 서버에 요청을 보내고 응답 결과를 출력

■ 웹 서버

- 클라이언트의 요청을 처리하도록 프로세스를 관리
- 요청을 처리한 결과를 클라이언트에 보냄

■ 웹 애플리케이션 서버(웹 컨테이너)

- 웹 애플리케이션의 실행 환경
- JSP 프로그램을 실행시키고 결과를 웹 서버에 전달

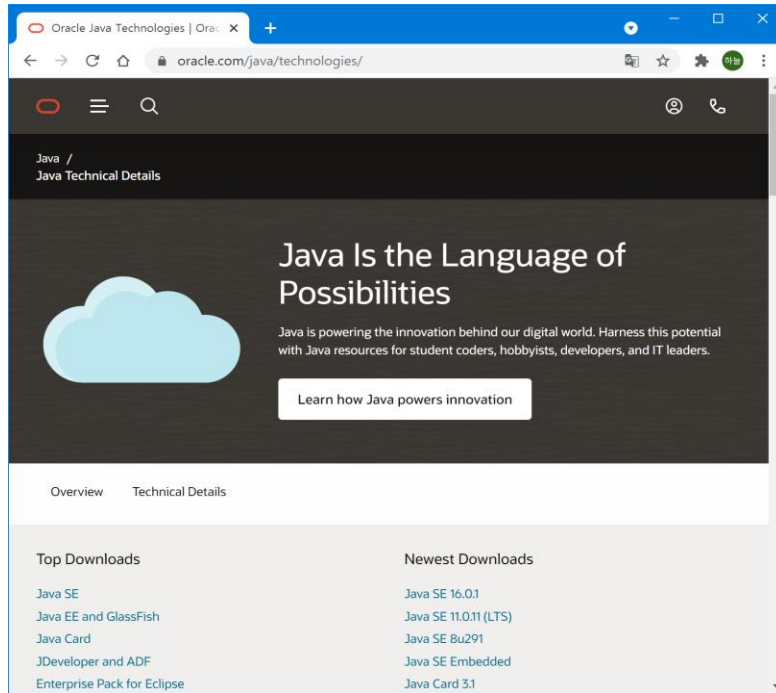
■ 데이터베이스

- 웹 서비스 수행에 필요한 데이터를 저장하고 제공

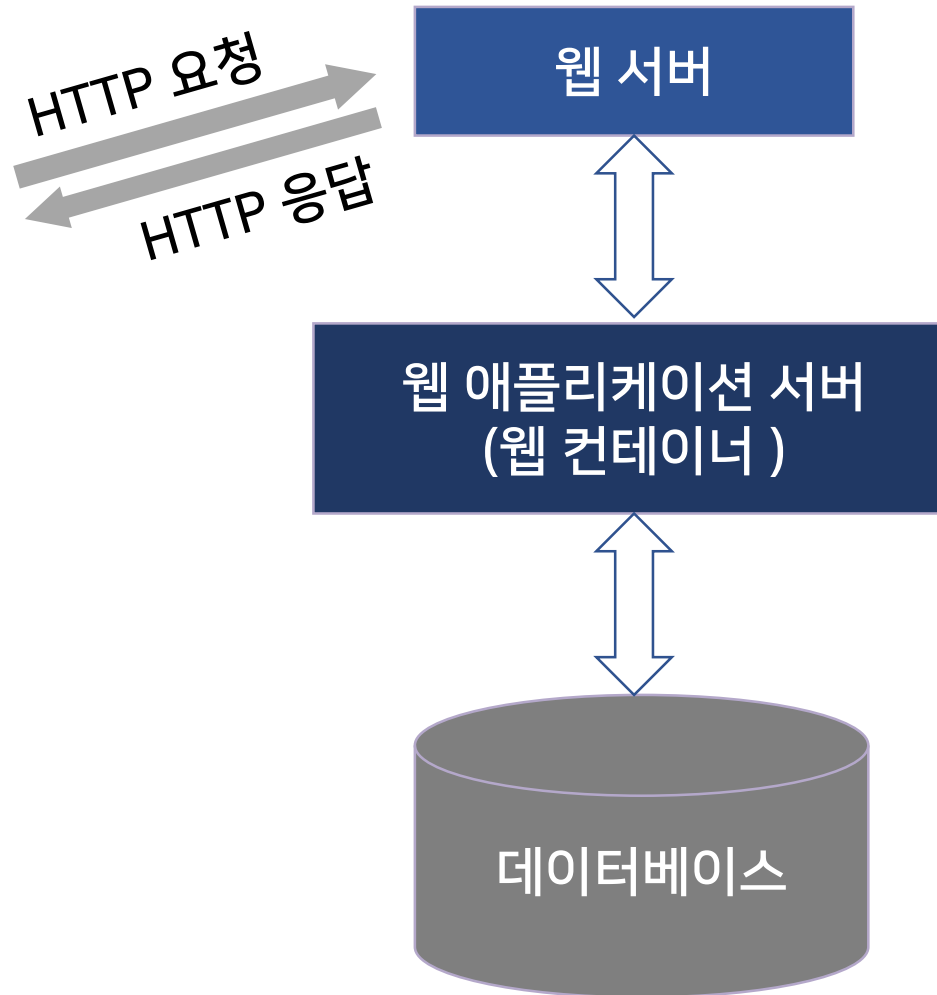


2) 웹 애플리케이션의 실행(2/2)

2. 웹 애플리케이션의 실행과 구성 요소



웹 브라우저



3) 웹 애플리케이션과 실행 위치

2. 웹 애플리케이션의 실행과 구성 요소

클라이언트 측 실행

- 웹 문서에 동적 요소를 포함시켜 클라이언트에 전송
- 웹 브라우저가 해석하여 페이지를 생성
- 애플릿, JavaScript, 플래시
- 보안의 문제

Vs

서버 측 실행

- 서버에서 실행되어 응답 문서를 동적으로 생성
- 웹 애플리케이션 서버가 수행한 결과가 브라우저에 전송됨
- 서블릿, JSP, ASP, PHP, CGI 방식
- 서버의 부담

4) 웹 애플리케이션의 구현 방식

2. 웹 애플리케이션의 실행과 구성 요소

컴파일 방식

- 컴파일 과정을 통해 실행 파일이나 바이트 코드가 만들어져 사용됨
- Java 서블릿, JSP
- Perl, C, C++을 이용한 CGI 방식

Vs

비컴파일 방식

- 매 요청마다 스크립트를 해석하여 실행하는 방식
- JavaScript는 클라이언트에서 실행되는 스크립트 언어

5) CGI와 WAS(1/4)

2. 웹 애플리케이션의 실행과 구성 요소

➤ CGI(Common Gateway Interface)

- 동적으로 웹 페이지를 생성하기 위한 방식 중 하나
 - 고급 언어 프로그램을 실행시켜 HTML 코드를 생성한 후 전달함
- 클라이언트의 요청이 있으면 해당 프로그램을 실행시키기 위해 개별 프로세스를 생성함
 - 동일한 CGI를 요청해도 요청의 개수만큼 프로세스를 생성하므로 웹 서버에 부하를 줌

```
1 printf("<html>\n\n<head><title>CGI 문서</title></head>\n\n");
2 printf("<body>\n");
3 printf("<h2>저희 사이트를 방문해 주셔서 감사합니다.</h2>\n");
4 ...
5 printf("</body>\n\n</html>");
```

➤ WAS(Web Application Server)

- 웹 애플리케이션을 실행하고 관리하는 별도의 전담 프로그램
 - 모든 요청에 대해 매번 프로세스를 생성하지 않고 하나의 자바 가상 기계 내에서 수행함
 - 요청을 처리하기 위해 스레드를 생성함
- 웹 페이지 생성 외에도 많은 기능을 수행함
 - API 제공, 부하 균형, 고장 조치 등

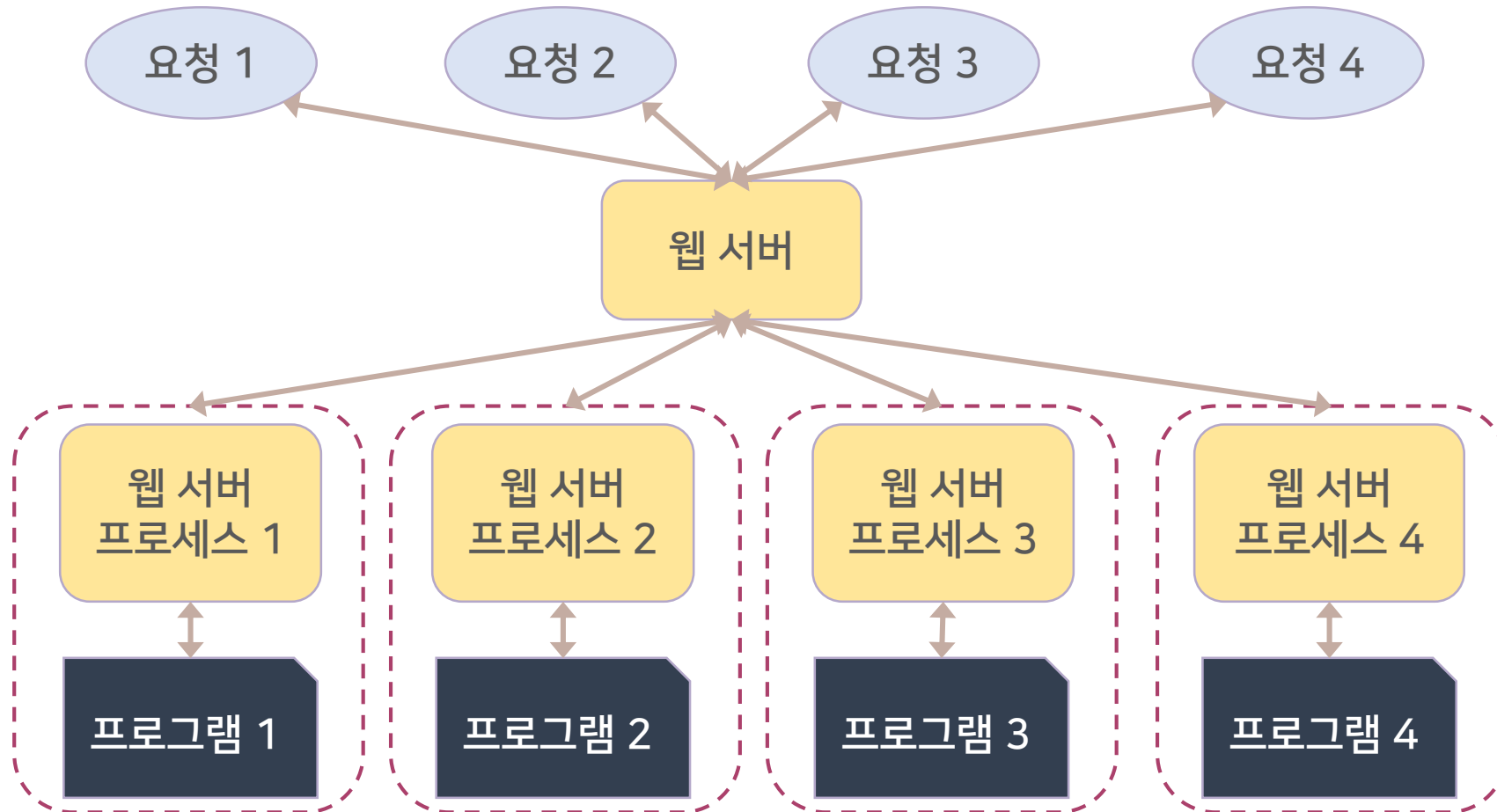


5) CGI와 WAS(3/4)

2. 웹 애플리케이션의 실행과 구성 요소

➤ CGI : 웹 서버가 응용 프로그램을 직접 호출

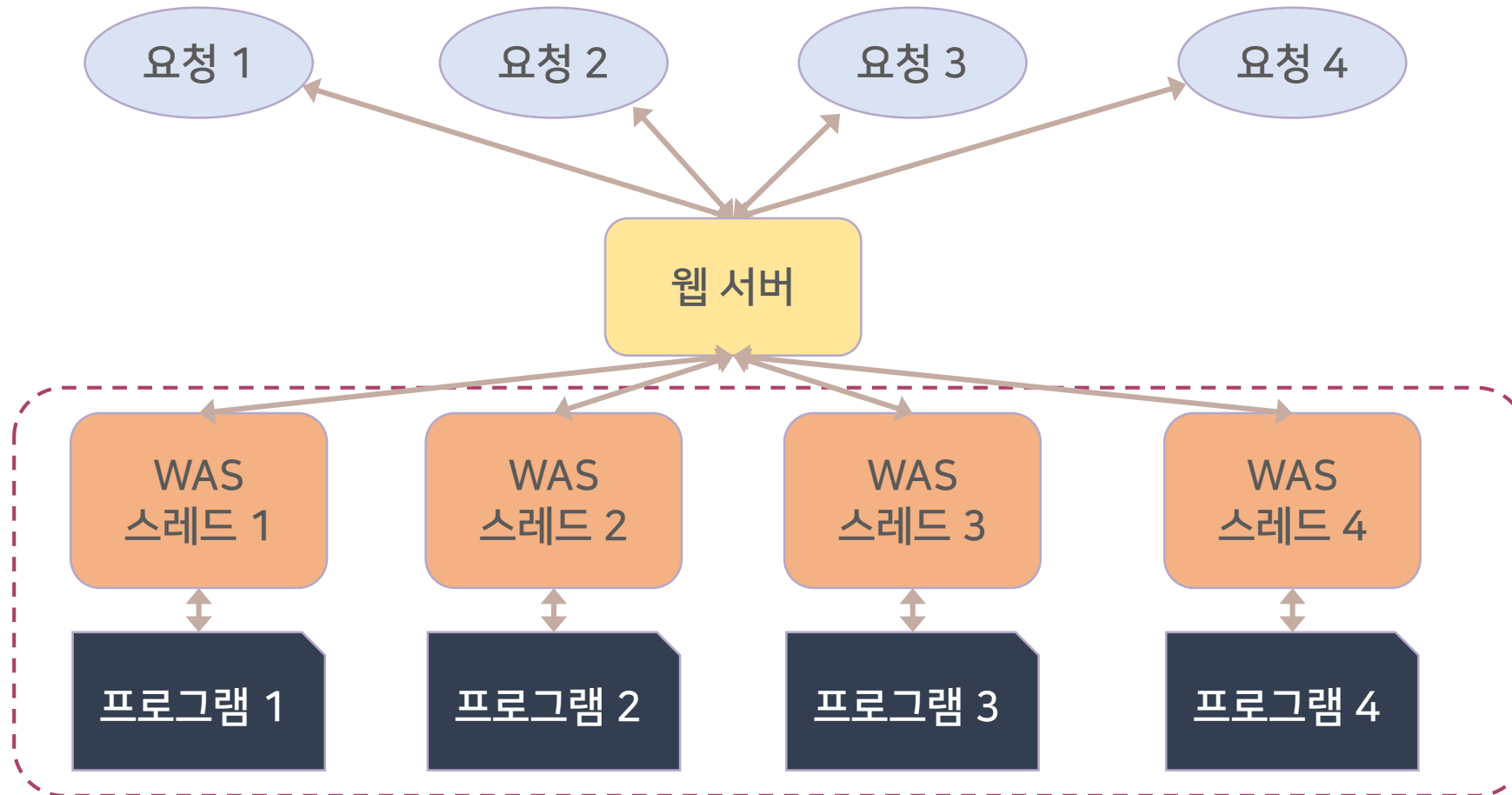
- 클라이언트의 요청이 많아지면 급격히 성능이 저하됨



5) CGI와 WAS(4/4)

2. 웹 애플리케이션의 실행과 구성 요소

- WAS : 동적 웹 서비스를 담당하는 서버
 - 스레드 방식을 사용함



➤ 웹 서버

- 클라이언트로부터의 요청을 받고 결과를 전달하는 기능
 - Apache HTTP Server, IIS, Nginx
- HTTP 프로토콜을 사용하여 클라이언트와 통신함
 - HTTP는 클라이언트와 웹 서버 간 웹 문서를 전송하기 위한 통신 규약
 - 웹 서버를 **HTTP 서버**라고 함

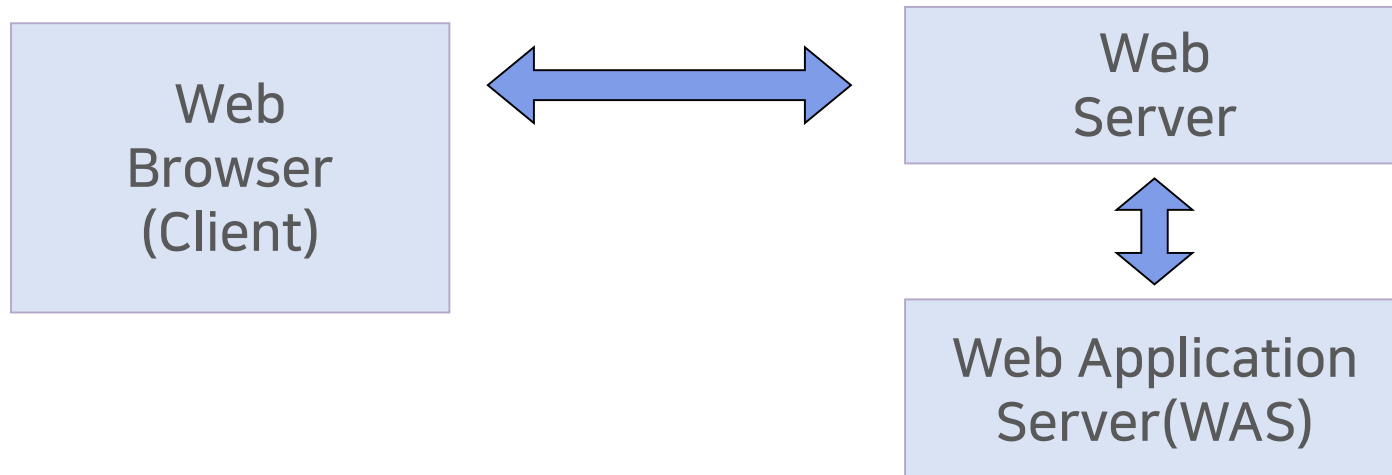
➤ 웹 서버의 구체적 기능

- 클라이언트가 요청한 웹 문서를 찾아 전달
- 클라이언트 요청에 대한 기본적 사용자 인증을 처리
- 문제가 있으면 정해진 코드 값으로 응답
- 프로그램 실행 요청이 있으면 처리 후 그 결과를 전달



➤ 웹 애플리케이션 서버

- 서버의 성능을 개선하기 위해 웹 서버의 기능을 분리
- 웹 서비스의 처리를 위해 동적 페이지를 만들거나 비즈니스 로직을 처리



03

JSP와 웹 컨테이너

➤ 서블릿

- (Server + Applet)의 합성어
 - 웹 페이지를 동적으로 생성하기 위한 서버 측 Java 클래스
 - Java 언어에 기초한 웹 프로그램의 개발 기술
- Java 언어로 서블릿 클래스를 만들고, 컴파일된 바이트 코드를 서버에 탑재하여 웹 서비스를 제공
 - 소스를 수정하면 다시 컴파일하여 서버에 탑재해야 함



1) 서블릿 프로그램(2/2)

3. JSP와 웹 컨테이너

```
1 public class HelloServlet extends HttpServlet {
2
3     public void doGet(HttpServletRequest request, HttpServletResponse response)
4         throws ServletException, IOException {
5
6         response.setContentType("text/html; charset=EUC_KR");
7         java.io.PrintWriter out = response.getWriter();
8         out.println("<HTML><HEAD><TITLE>로그인</TITLE></HEAD>");
9         out.println("<BODY><H2>헬로 서블릿</H2>");
10        out.println("오늘의 날짜와 시간은 : "+new java.util.Date());
11        out.println("</BODY></HTML>");
12    }
13 }
```

➤ JSP(JavaServer Pages)

- 서블릿 대신에 사용할 수 있는 스크립트 형식의 언어
 - HTML 페이지 내에 삽입됨
- JSP 페이지는 서블릿으로 변환됨
 - 웹 애플리케이션 서버가 **자동으로 JSP 페이지를 변환하고 컴파일**하여 웹 서비스를 제공함
- Java EE를 구성하는 기술 중의 하나



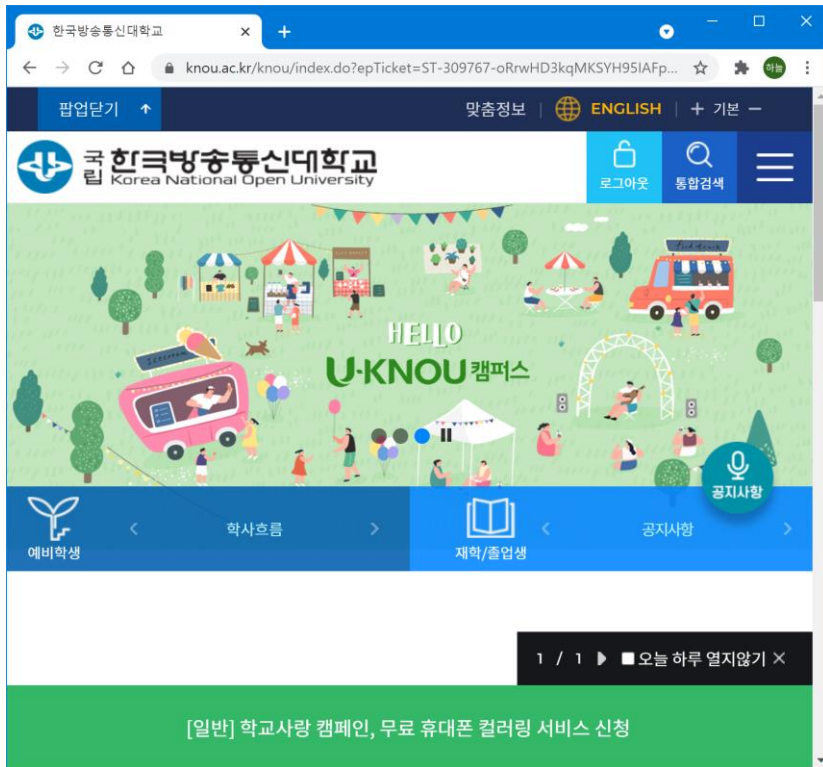
특징

- 스크립트 언어로 HTML 페이지에 삽입됨
- Java 언어의 특성을 활용
- 표현 언어, 표현식, 액션 태그 등의 스크립트적 요소를 제공
- 다양한 개발 환경이 오픈 소스로 제공됨
 - JSP는 오픈 소스 형태로 운영되므로 다양한 기술이 지원됨
- JSP 기술은 플랫폼에 독립적
 - 반면에 .NET 기술은 개발과 운용 환경에 제약이 있음



3) JSP의 처리 과정

3. JSP와 웹 컨테이너

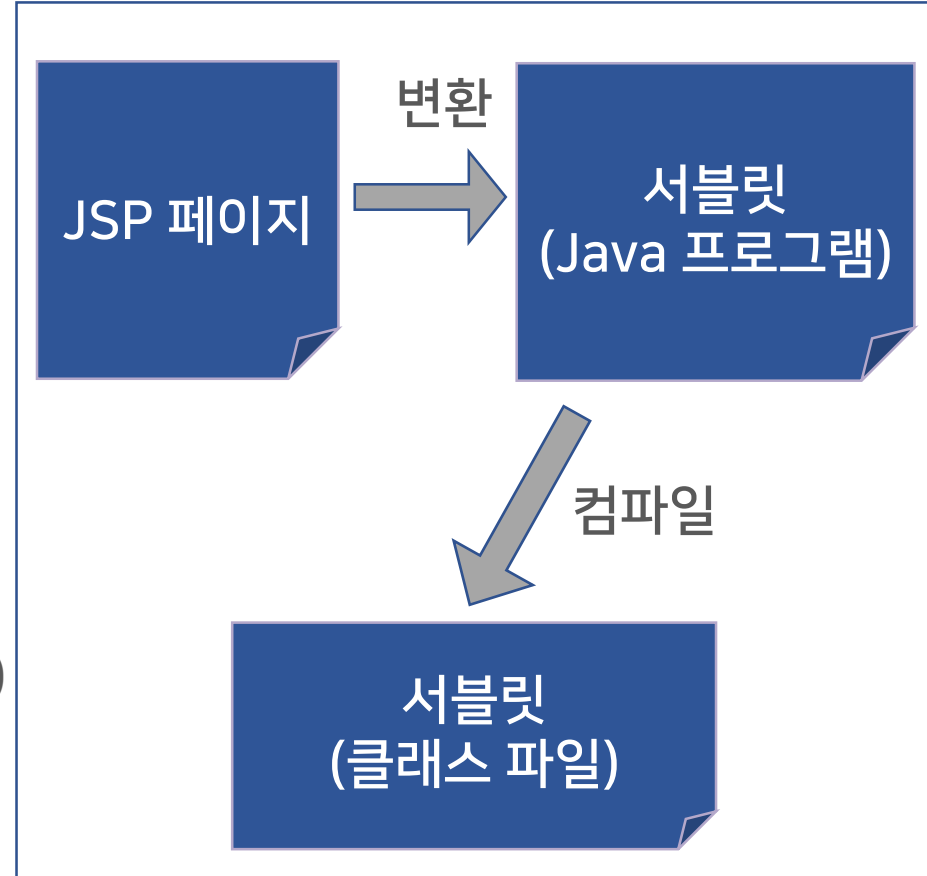


클라이언트

요청(.jsp)



응답(HTML)



WAS/웹 컨테이너

4) 웹 컨테이너(1/2)

3. JSP와 웹 컨테이너

➤ 웹 컨테이너

- 웹 컴포넌트를 저장하고 서블릿의 생명주기를 관리
 - 클라이언트의 서블릿 요청을 실행시키는 역할
- 서블릿 컨테이너라고도 함
 - Java로 구현된 서블릿 엔진(서블릿 실행 환경)
 - JSP를 서블릿으로 변환하는 기능을 포함
- Tomcat, WildFly, WebLogic, WebSphere, JBoss 등



4) 웹 컨테이너(2/2)

3. JSP와 웹 컨테이너

➤ 웹 서버에서 서블릿이 실행되기 위한 환경

- 웹 서버
- 자바 실행 환경(JDK)
- 서블릿 컨테이너(예: Tomcat)

➤ JSP 컨테이너

- JSP 페이지를 서블릿 프로그램으로 변환
- 대부분의 서블릿 컨테이너는 JSP 컨테이너 기능을 포함함
- JSP 컨테이너 자체가 서블릿으로 구현되어 있음
 - 서블릿 컨테이너에 의해 실행됨



04

HTTP 프로토콜

1) HTTP 프로토콜의 이해

➤ HTTP(HyperText Transfer Protocol) 프로토콜

- 웹 서버와 클라이언트가 통신하는 규약
 - TCP 프로토콜에 기초한 애플리케이션 계층 프로토콜
- Connection oriented & Stateless 특성
 - 요청을 위해 접속을 해야 함
 - 서버가 응답한 후에 서버는 클라이언트의 상태를 유지하지 않음
 - 웹 서버의 부담이 줄어드나 상태 관리를 위해 쿠키나 세션 등이 필요
- HTTP 요청과 응답 절차
 - 연결 설정, 요청 메시지 전송, 응답 메시지 전송, 연결 끊기

2) HTTP 요청 메시지(1/3)

➤ HTTP 요청 메시지 구조(1/2)

- 시작 라인
 - 요청 방식, URI, 버전 번호
- 예

```
GET /index.html HTTP/1.1  
Host: www.example.com  
... ..
```

- **GET**: 웹 서버에게 요청 대상에 대해 처리 방식을 지정
- **/index.html**: 서버에 존재하는 파일의 URI
- **HTTP/1.1**: HTTP의 버전

시작라인

요청 헤더

요청 몸체

2) HTTP 요청 메시지(2/3)

➤ HTTP 요청 메시지 구조(2/2)

- 요청 헤더
 - 한 라인에 하나씩 헤더 정보를 기술
 - 각 라인은 “헤더필드이름:값” 형식으로 구성됨
 - 요청 헤더의 끝에 공백 라인을 둠
- 요청 메시지의 몸체
 - POST 요청 방식에서만 의미 있음
 - HTML 폼에서 작성한 데이터를 POST 방식으로 전송할 때 사용됨

시작라인

요청 헤더

요청 몸체

2) HTTP 요청 메시지(3/3)

➤ 요청 헤더의 예

```
Host: www.w3.org
Referer: http://www.w3.org/hypertext/DataSources/Overview.htm
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,...
Accept-Language: ko
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
Content-Length: 43
Content-Type: application/x-www-form-urlencoded
```

3) HTTP 응답 메시지(1/2)

4. HTTP 프로토콜

➤ HTTP 응답 메시지 구조

- 시작 라인, 응답 헤더, 응답 몸체로 구성됨
- 시작 라인은 HTTP 버전, 서버의 응답 코드, 설명으로 구성됨
- 서버의 응답 코드와 설명
 - 200 OK - 클라이언트 요청이 성공적으로 끝남
 - 400 Bad Request - 잘못된 요청
 - 401 Unauthorized - 인증 오류
 - 403 Forbidden - 사용자 허가 모드 오류
 - 404 Not Found - 요청한 문서가 존재하지 않음
 - 405 Method Not Allowed - 요청한 방식을 지원하지 않음
 - 500 Internal Server Error - 서버에서의 실행 오류
 - 503 Server Unavailable - 일시적으로 요청을 처리할 수 없음

시작라인

응답 헤더

응답 몸체

3) HTTP 응답 메시지(2/2)

4. HTTP 프로토콜

▶ 응답 메시지 예

```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
ETag: "3f80f-1b6-3e1cb03b"
Content-Type: text/html; charset=UTF-8
Content-Length: 131
Accept-Ranges: bytes
Connection: close
```

```
<html>
<head>
  <title>An Example Page</title>
</head>
<body>
  Hello World, this is a very simple HTML document.
</body>
</html>
```

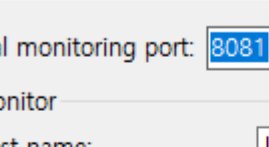


Eclipse에서 TCP/IP 모니터링 설정

▶ 설정하고 반드시 start

- Local monitoring port

- **모니터할 포트로 8081**
 - 실행 주소를 8081로 변화시켜 실행해 주면 결과가 보임



Local monitoring port: 8081

Monitor

Host name: localhost

Port: 8088

Type: HTTP

Timeout (in milliseconds): 0

☒ Start monitor automatically

OK Cancel

[illegible]

Eclipse에서 TCP/IP 모니터링 보기

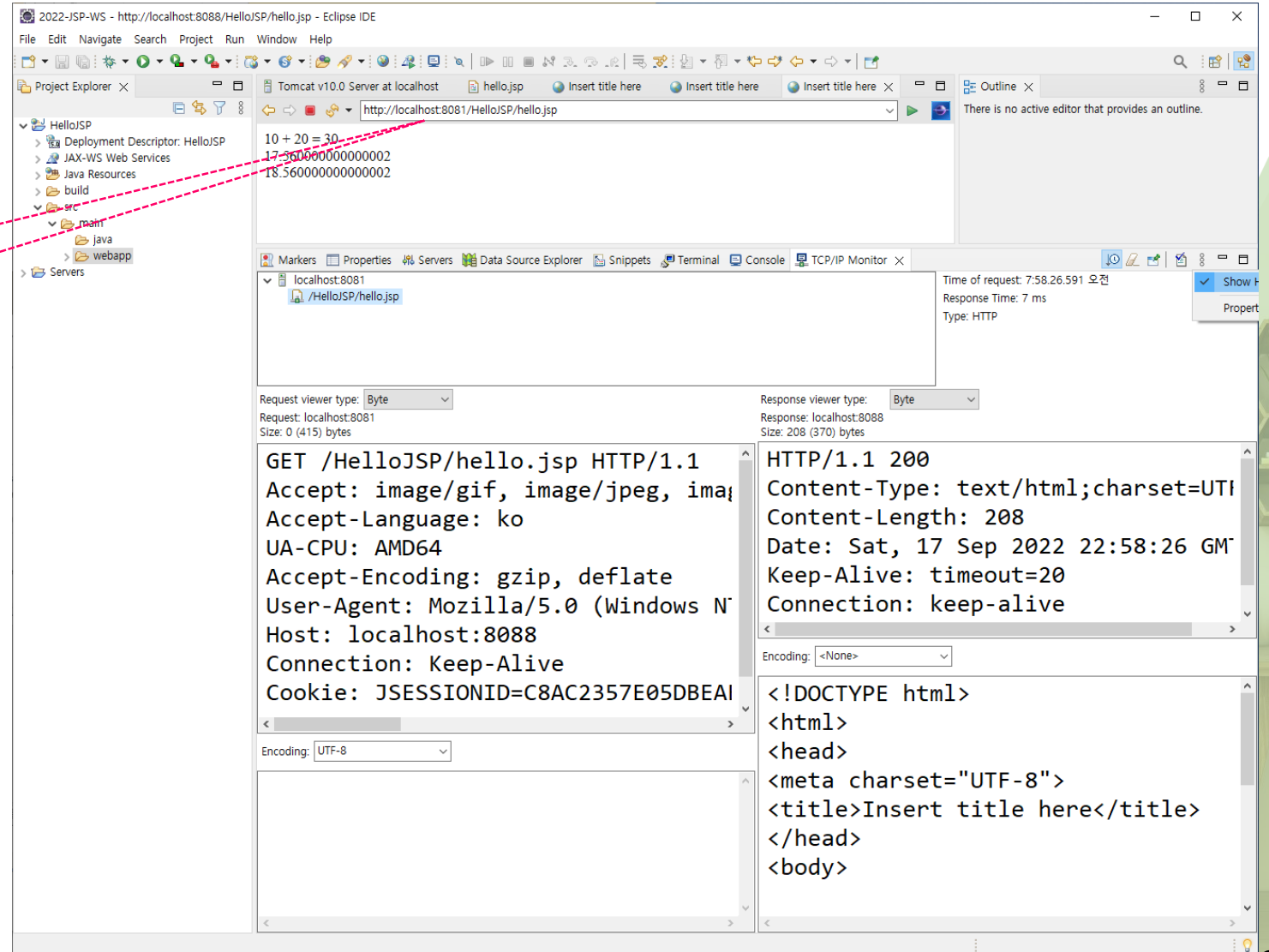
➤ 실행 URL에서

- 모니터링 지정 포트로 수정해서 실행

- <http://localhost:8081/myJSP/index.jsp>

- 다음 화면 안 보여도 실행시키면 보이게 됨

- 메뉴 Window/show view/debug/TCP/IP 모니터



2강

다음시간 안내

개발 환경 설정하기

