

JSP 프로그래밍

(재)대덕인재개발원

01 웹 프로그래밍 기초

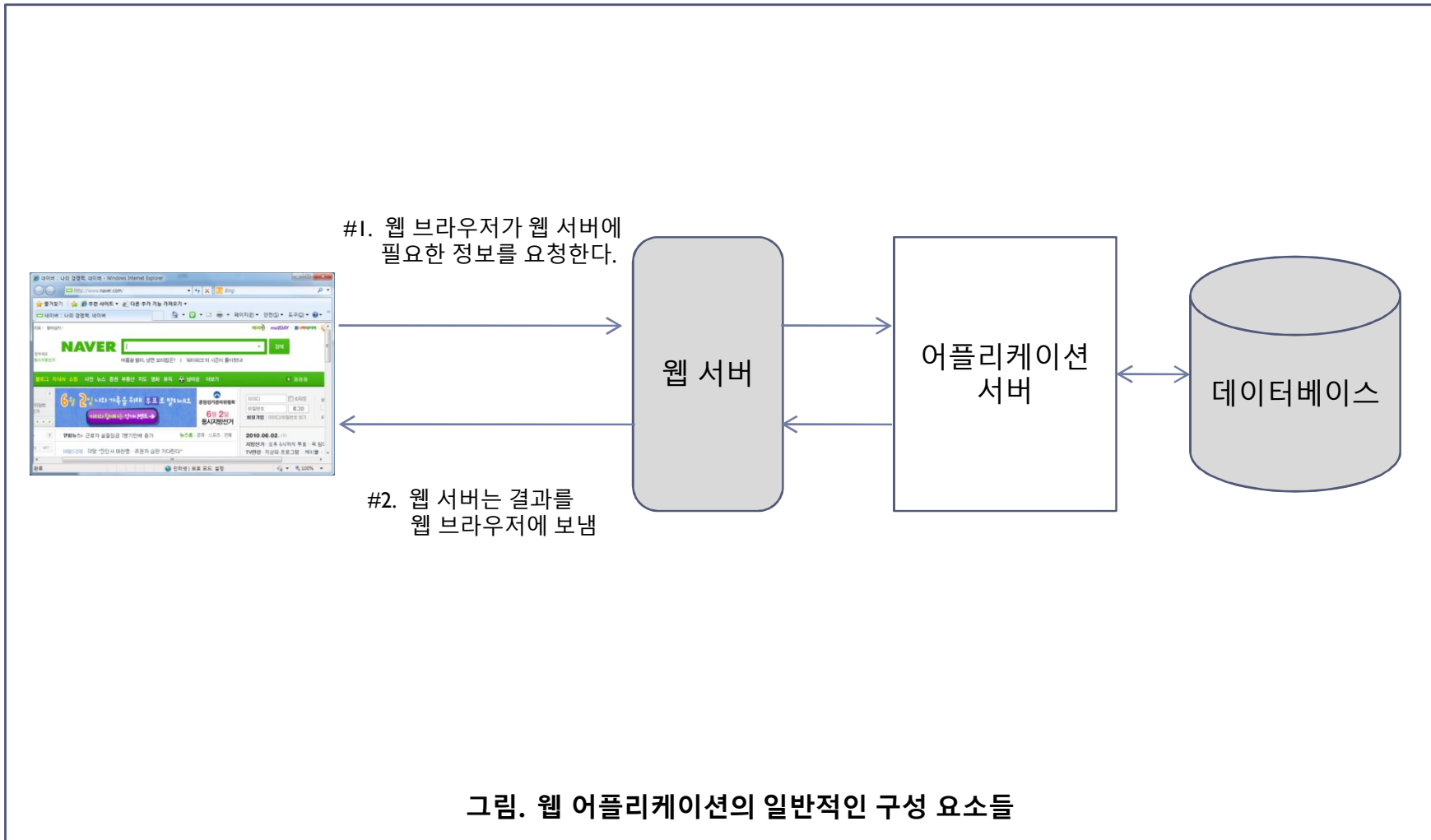


1.1 웹 어플리케이션 구성요소



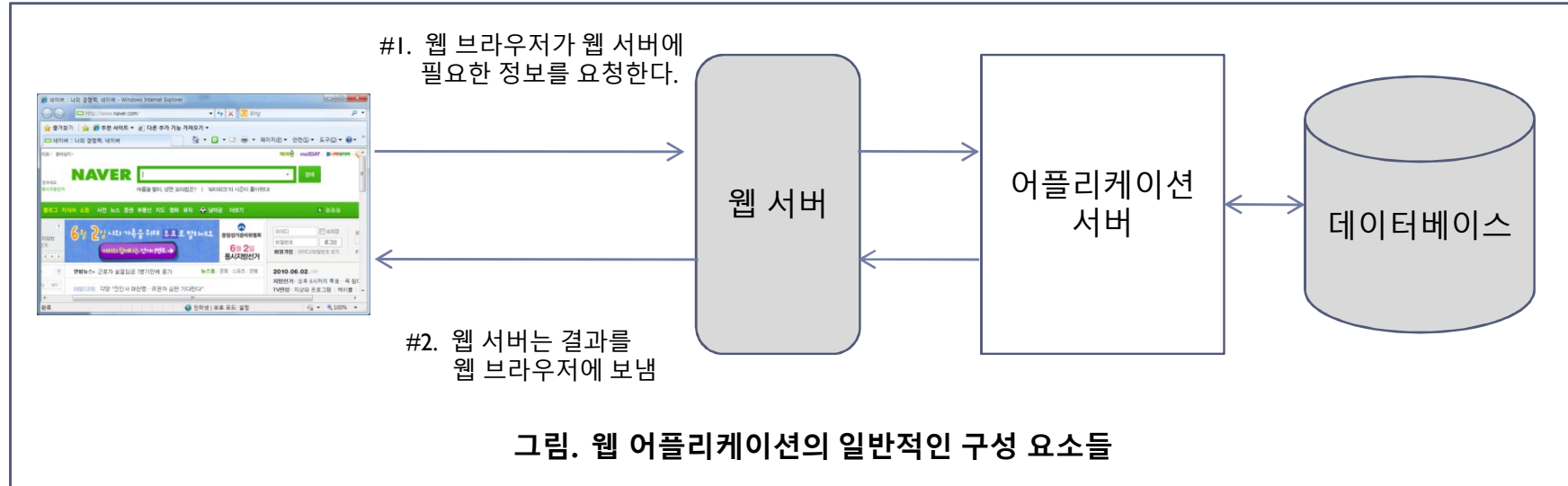
1.1 웹 어플리케이션 구성요소

p.30



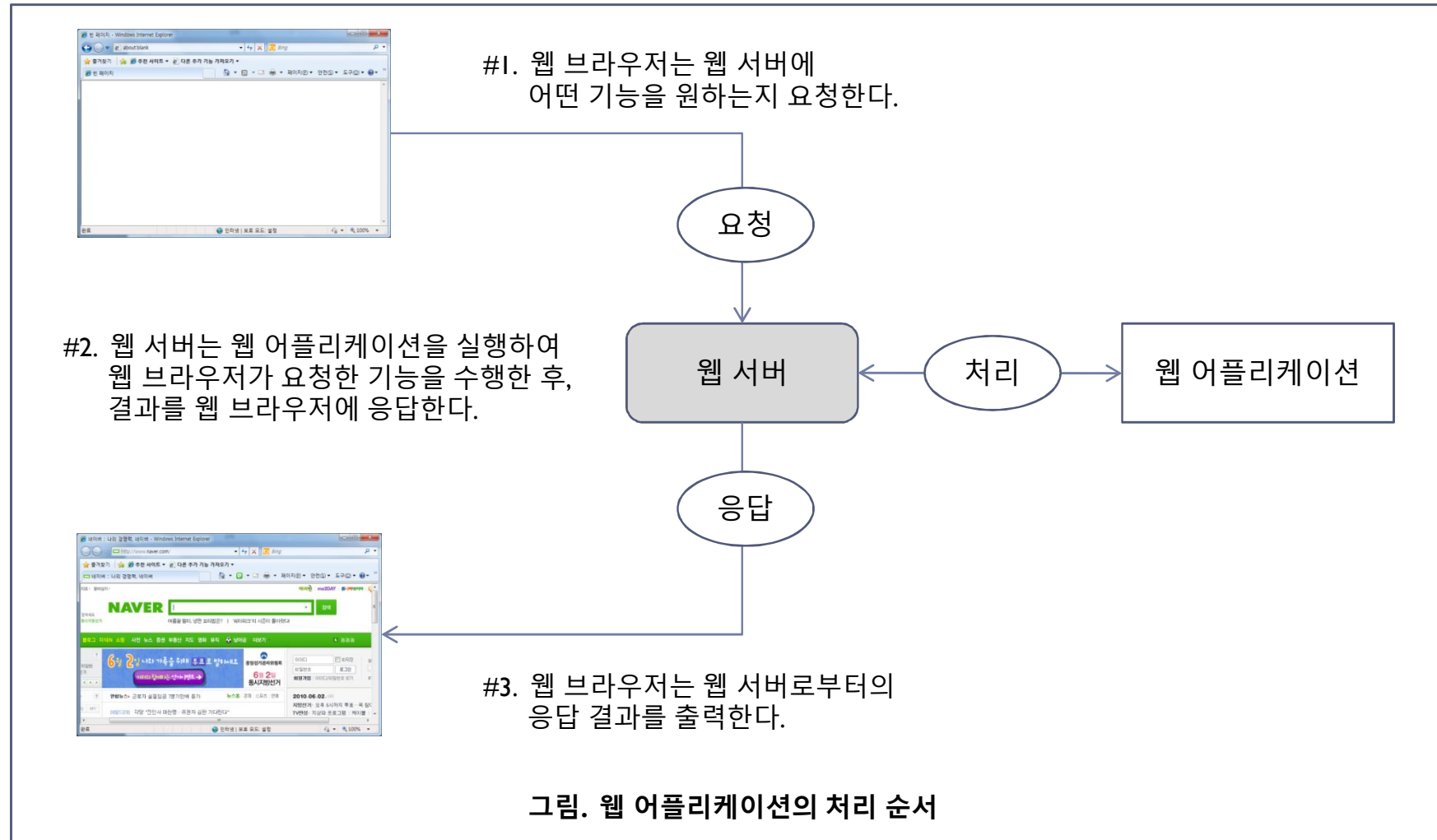
1.1 웹 어플리케이션 구성요소

p.30



구 성 요 소	역 할
웹 브라우저	사용자에게 화면(즉, 유저 인터페이스; UI)을 제공한다.
웹 서버	웹 브라우저의 요청을 받아서 알맞은 결과를 웹 브라우저에 전송한다. 만약 프로그램 처리가 필요하다면 어플리케이션 서버를 사용하거나 프로그램을 직접 호출하여 결과를 생성한다.
어플리케이션 서버	게시판의 글 목록, 로그인 처리와 같은 기능을 처리하고 그 결과를 웹 서버에 전달한다.
데이터베이스	사이트의 정보를 저장한다. 예를 들어, 회원 정보, 게시판 글 데이터 등을 저장한다.

1.2 웹 어플리케이션의 처리 순서



1.2 CGI 방식과 어플리케이션 서버 방식 p.31

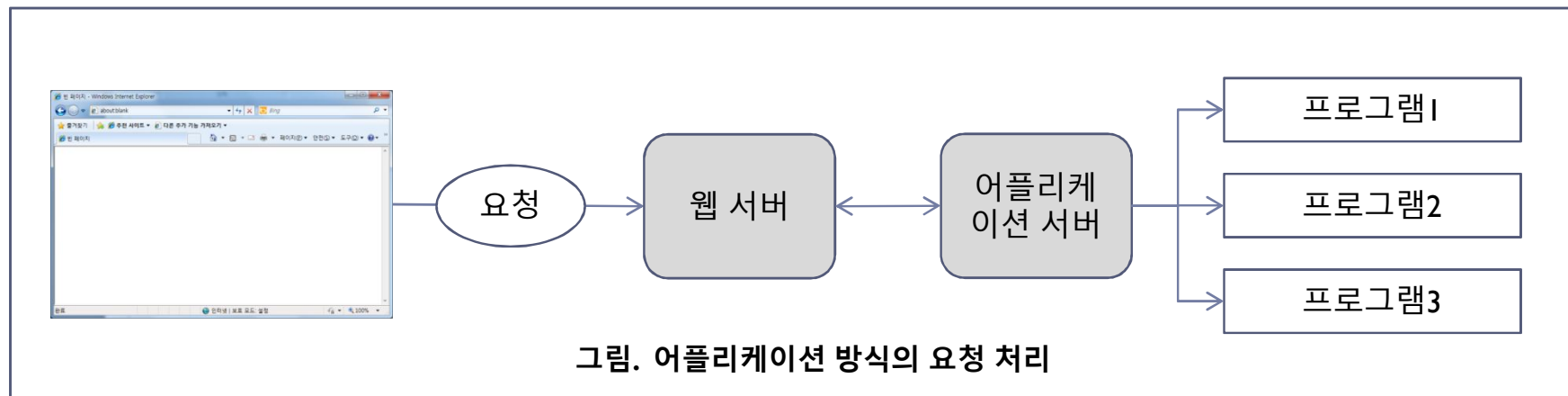
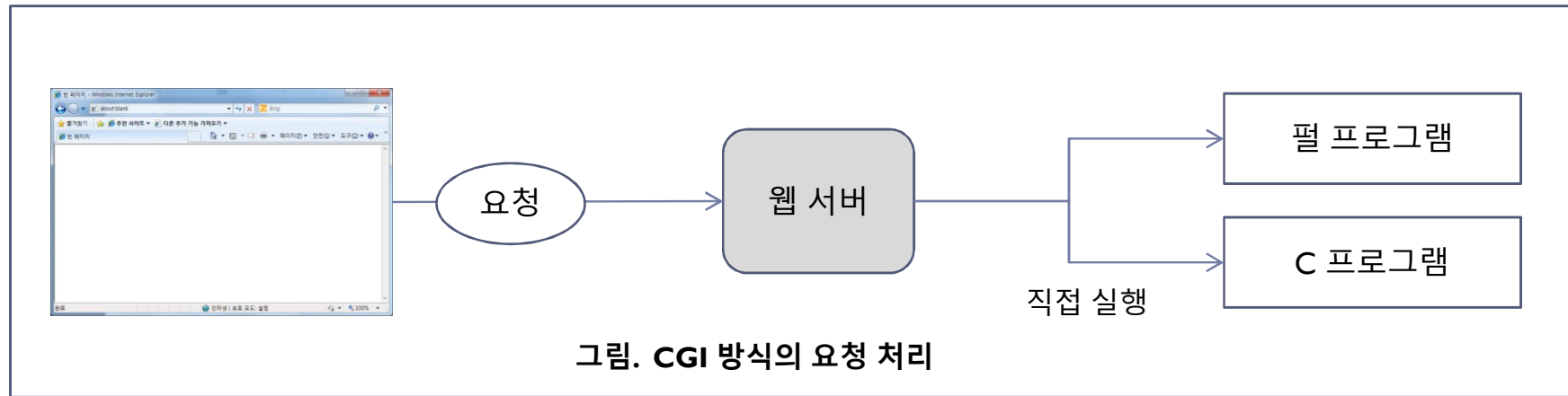
▶ CGI

- ▶ CGI는 Common Gateway Interface의 약자로서 웹 서버와 프로그램 사이에 정보를 주고받는 규칙을 의미한다. 흔히 CGI 프로그래밍이라고 하면, 펄(Perl)이나 C/C++ 언어 등을 사용하여 웹 서버를 통해서 실행할 수 있는 프로그램을 작성하는 것을 의미한다.

▶ CGI 방식과 어플리케이션 서버 방식의 차이

웹 APP 방식	요청 처리	성능
CGI	웹 브라우저가 웹 서버에 프로그램 실행을 요청하면, 웹 서버는 펄이나 C로 작성된 CGI 프로그램을 직접 실행하고, 프로그램이 생성한 결과를 웹 브라우저에 전송한다.	동일 요청에 대해 별도의 프로세스를 생성 함으로서 전체적으로 성능 저하가 발생.
어플리케이션 서버 방식	웹 서버가 어플리케이션 서버를 통해서 간접적으로 웹 어플리케이션 프로그램을 실행한다.	동일 요청에 대해 관련 프로그램을 한 번만 로딩함으로써 메모리 사용량이 적어 처리량이 높고 안정적이다.

1.2 CGI 방식과 어플리케이션 서버 방식 p.30



1.2 스크립트 방식과 실행 코드 방식

p.31

- ▶ 웹 어플리케이션 프로그래밍은 구현 방식에 따라 실행 코드 방식과 스크립트 방식으로 구분할 수 있다.
- ▶ 스크립트 방식과 실행 코드 방식의 차이

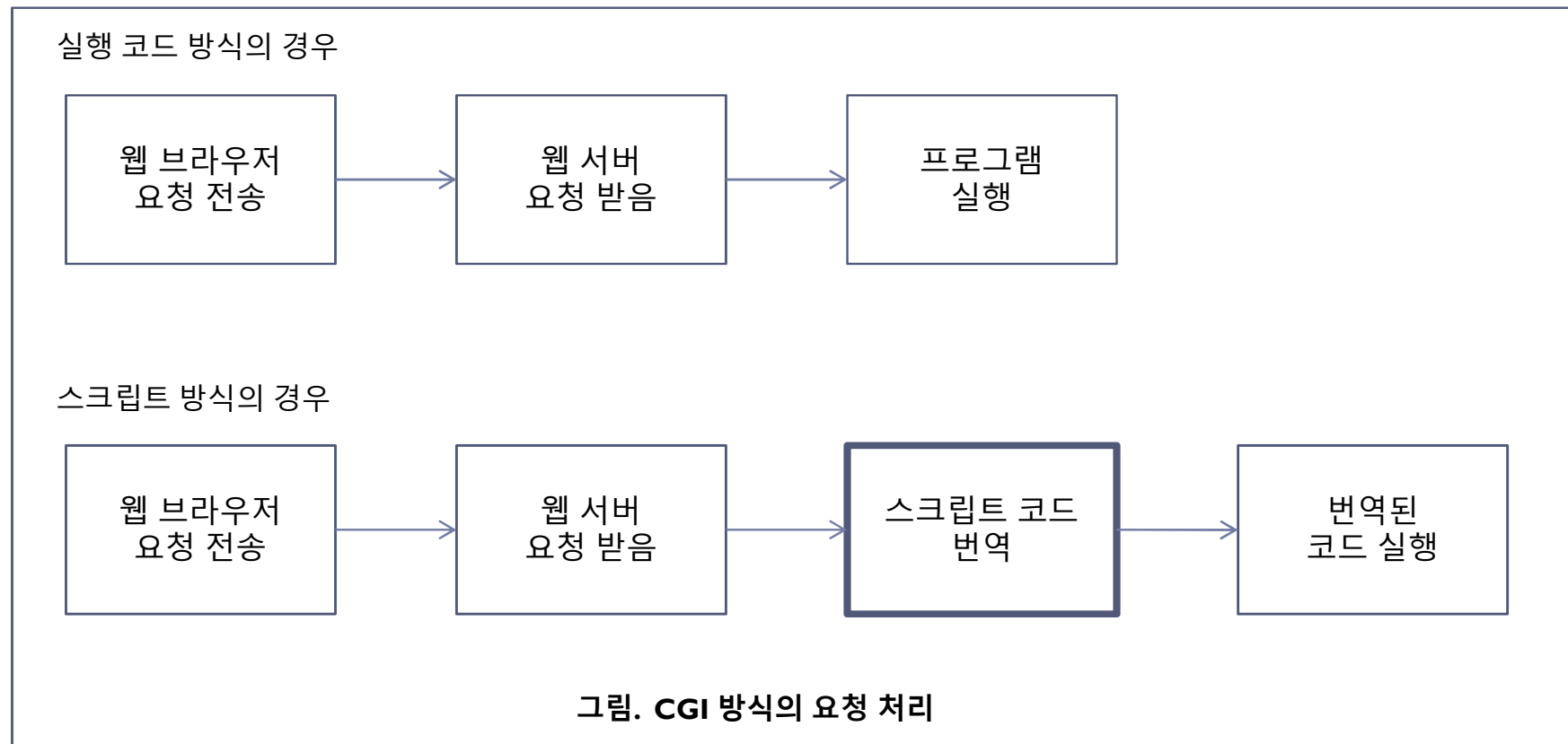
비교 항목	실행 코드 방식	스크립트 방식
코드 형태	컴파일 된 실행 프로그램	컴파일 되지 않은 스크립트 코드
실행 방식	컴파일 된 기계어 코드 직접 실행	스크립트 코드를 해석한 뒤 실행
코드 변경	소스 코드를 다시 컴파일 해야 함	스크립트 코드만 고치면 됨
종류	C 기반 CGI 프로그램	JSP, ASP.NET, PHP, Ruby 등



1.2 스크립트 방식과 실행 코드 방식

p.30

▶ 스크립트 방식과 실행 코드 방식의 차이



1.3 자바와 웹 프로그래밍

- ▶ 서블릿과 JSP
- ▶ JSP란 무엇인가?
- ▶ 웹 컨테이너
- ▶ JSP를 사용하는 이유



1.3.1 서블릿과 JSP

p.32

▶ 서블릿

- ▶ Sun Microsystems에서 웹 개발을 위해 만든 표준.
- ▶ 서블릿 규약에 따라 만든 클래스.
- ▶ 자바 코드 작성 후 컴파일 해서 클래스 파일을 만드는 실행 코드 방식.

▶ JSP

- ▶ 서블릿의 단점을 보완하기 위해 스크립트 방식의 표준.
- ▶ JSP 표준은 서블릿 표준을 기반으로 만들어졌다.
- ▶ 내부적으로 JSP 파일이 번역되면 최종 결과물로 서블릿이 만들어진다.



1.3.1 JSP란 무엇인가?

p.32

- ▶ JSP(Java Server Pages)는 스크립트 언어이다.
- ▶ 주로 웹 브라우저의 요청에 따라 알맞은 HTML 결과 화면을 생성하기 위해 사용된다.
- ▶ JSP의 특징
 - ▶ 자바 언어를 기반으로 하는 스크립트 언어로서 자바가 제공하는 기능을 그대로 사용할 수 있다.
 - ▶ HTTP와 같은 프로토콜에 따라 클라이언트의 요청을 처리하고 응답한다.
 - ▶ HTTP, XML 등 클라이언트가 요청한 문서를 생성하는 데 주로 사용된다.
 - ▶ 서블릿/EJB 등의 엔터프라이즈 기술들과 잘 융합된다.
 - ▶ 표현 언어, 표현식, 스크립트릿 등 다양한 스크립트 요소와 액션 태그 등을 제공함으로써 보다 쉽게 웹 어플리케이션을 프로그래밍 할 수 있도록 도와준다.



1.3.1 웹 컨테이너

p.32

- ▶ 웹 컨테이너(Web Container)는 웹 어플리케이션을 실행할 수 있는 컨테이너이다.
- ▶ 자바의 웹 컨테이너는 JSP와 서블릿을 지원해야 한다.
- ▶ 톰캣(tomcat), 제티(Jetty), GlassFish, WebLogic, WebSphere, JBoss 등



02 웹 프로그래밍 시작하기



2.1.1 JDK(J2SE Development Kit)

- ▶ J2SE(Java 2 Standard Edition)
 - ▶ 일반적인 데스크톱용 애플리케이션 개발을 위한 API
 - ▶ 현재 가장 최신 버전은 J2SE 14.0.1
 - ▶ <https://www.oracle.com/java/technologies/javase-downloads.html>

- ▶ J2SE 설치
 - ▶ 셋업 형태의 실행 파일이기 때문에 간단히 설치 가능
 - ▶ <https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>

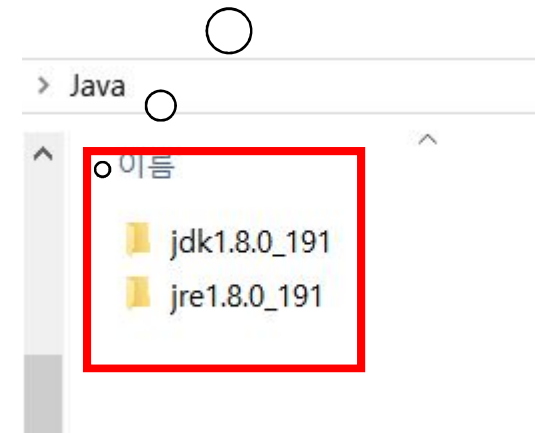
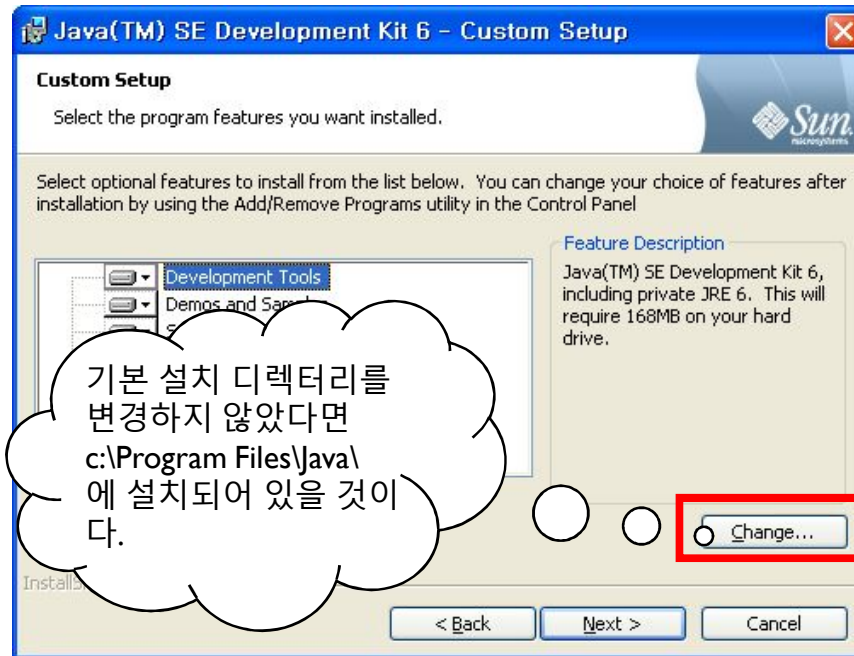
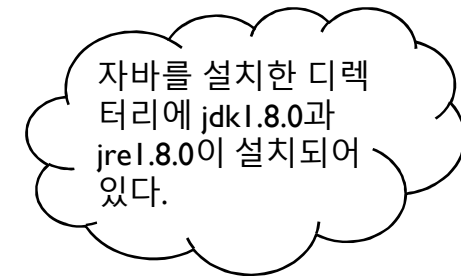
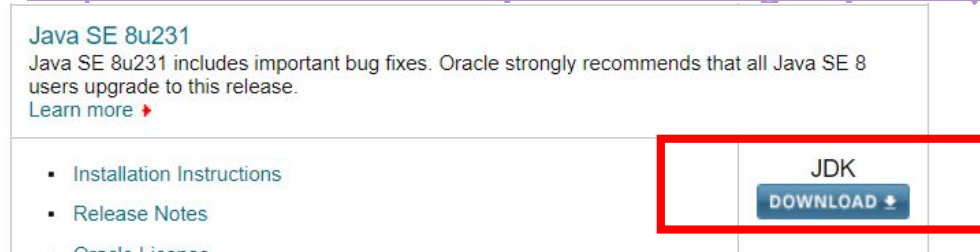
- ▶ 설치 순서
 1. J2SE 다운로드
 2. J2SE 설치
 3. 환경 설정
 4. 필요에 따라 도움말 파일 설치



2.1.2 J2SE 8.0 설치

▶ J2SE 8.0 다운로드

- ▶ <https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>



2.1.3 J2SE 8.0의 환경설정

▶ JAVA_HOME 설정

- ▶ 자바가 설치된 디렉토리를 환경변수에 추가

시스템 변수 편집

변수 이름(N):	JAVA_HOME
변수 값(V):	D:\Java\jdk1.8.0_191

▶ Path 설정

- ▶ 자바 컴파일러인 `javac.exe`와 자바 명령 해석기인 `java.exe` 및 기타 실행 파일을 어디서나 실행 가능하도록 지정
- ▶ Path 환경변수에 `%JAVA_HOME%\bin`을 추가



C:\Program Files (x86)\Intel\Intel(R) Man...
C:\Program Files\Intel\Intel(R) Managen...
%JAVA_HOME%\bin
D:\Intel_Inno\Brackets\command...
C:\Program Files\Intel\Intel(R) Man...

▶ CLASSPATH 설정

- ▶ 자바 라이브러리의 물리적인 위치를 지정하기 위해 선언

새 시스템 변수

변수 이름(N):	classpath
변수 값(V):	.;

2.1.4 TOMCAT 설치

p.20

▶ TOMCAT

- ▶ 서블릿과 JSP를 실행시키기 위한 서블릿 컨테이너(Servlet Container)
- ▶ Sun Microsystems와 Apache Software Foundation이 Jakarta 프로젝트를 통해 공동으로 개발한 서블릿 컨테이너

▶ TOMCAT의 기능

- ▶ 웹 서버(Web Server) 기능 포함
- ▶ 서블릿 컨테이너(Servlet Container) 기능
- ▶ 서블릿과 JSP를 동시에 지원

▶ 서블릿 컨테이너(Servlet Container)의 종류

- ▶ TOMCAT, Resin, Apache JServ, JRun/JRun Pro 등



2.1.5 TOMCAT 설치

▶ TOMCAT 다운로드

- ▶ <http://tomcat.apache.org/whichversion.html>

6.0.10

Please see the [README](#) file for packaging information. It explains what every distribution contains.

Binary Distributions

- Core:
 - [zip \(pgp, md5\)](#)
 - [tar.gz \(pgp, md5\)](#)
 - [Windows Service Installer \(pgp, md5\)](#)
- Deployer:
 - [zip \(pgp, md5\)](#)
 - [tar.gz \(pgp, md5\)](#)

Apache Tomcat Setup

Choose Install Location

Choose the folder in which to install Apache Tomcat.

Setup will install Apache Tomcat in the following folder. To install in a different folder, click Browse and select another folder.

Destination Folder

C:\Java\Tomcat

Space required: 9.6MB
Space available: 22.4GB

Nullsoft Install System v2.22

< Back Next > Cancel

설치할 디렉터리를 변경한다.

Apache Tomcat Configuration Options

웹 서버의 포트 번호를 입력

HTTP/1.1 Connector Port: 80

Administrator Login

User Name: admin

Password: ****

웹 서버의 관리자 아이디와 패스워드를 입력

Nullsoft Install System

< Back Next > Cancel

Apache Tomcat Setup: Java Virtual Machine path selection

Java Virtual Machine

Java Virtual Machine path selection.

Please select the path of a J2SE 5.0 JRE installed.

C:\Java\jre1.6.0

JRE가 설치된 디렉터리를 지정한다.

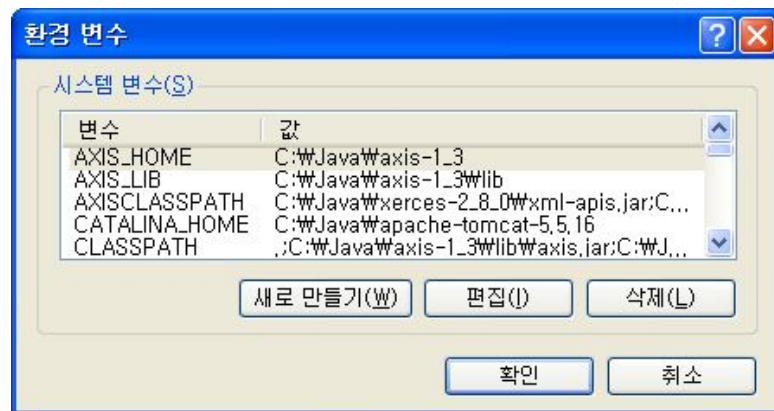
Nullsoft Install System v2.22

< Back Install Cancel

2.1.6 TOMCAT의 환경설정

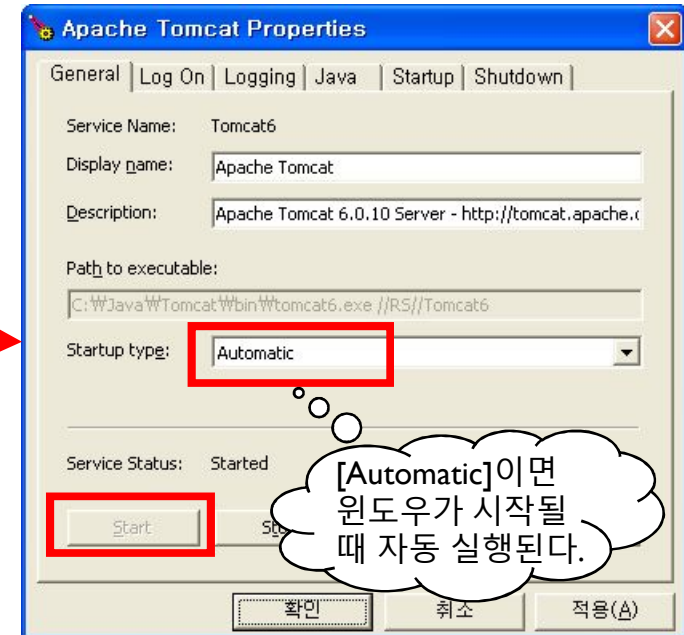
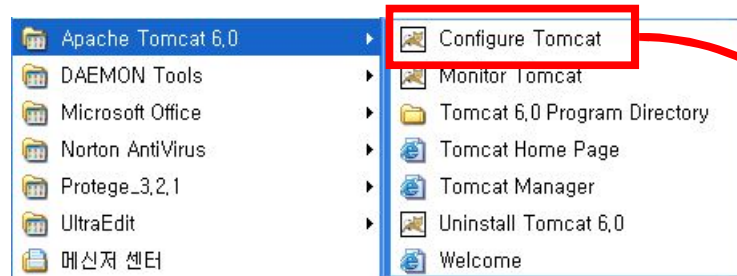
- ▶ CATALINA_HOME 설정
 - ▶ TOMCAT이 설치된 디렉터리를 환경 변수에 추가
- ▶ CLASSPATH 설정
 - ▶ TOMCAT 라이브러리의 물리적인 위치를 지정하기 위해 선언
 - ▶ 서블릿을 지원하는 클래스들을 프로그램적으로 이용하기 위해서 CLASSPATH에 servlet-api.jar 파일을 포함시켜야 한다.

servlet-api.jar 파일의
경로명을 classpath에
추가하는 것을 잊지
말자.



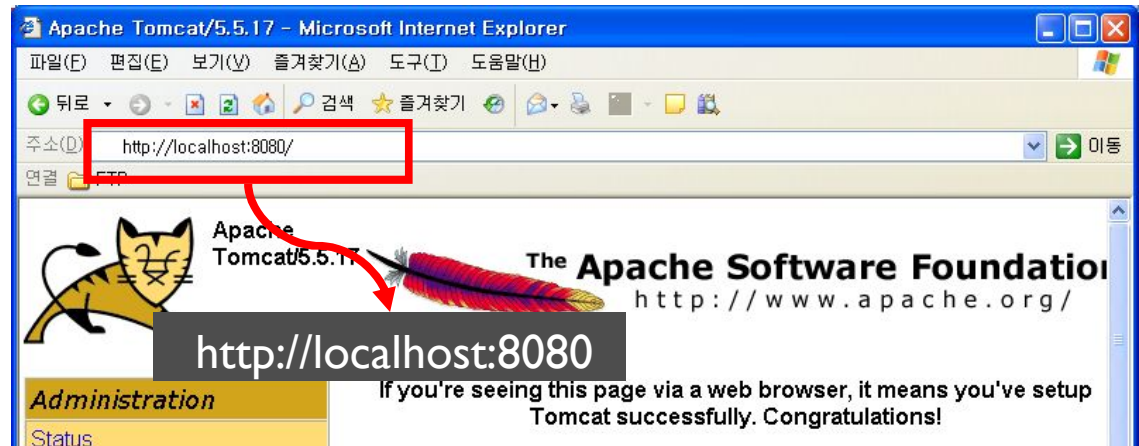
2.1.7 TOMCAT 실행

- ▶ 모든 프로그램 > Apache Tomcat 6.0 > Configure Tomcat



- ▶ TOMCAT 실행 확인
 - ▶ 웹 브라우저에서 http://localhost:8080을 입력하여 TOMCAT이 동작하는지 확인하면 된다.

- ▶ TOMCAT의 Servlet, JSP 엔진 버전 확인하기
 - ▶ TOMCAT을 설치 후 bin 디렉터리 안의 version.bat을 도스 창에서 실행



2.1.8 톰캣 설치 및 환경 구축

▶ 톰캣 디렉토리 구조

디렉토리	설명
bin	톰캣을 실행하고 종료시키는 스크립트(.bat 이나 .sh) 파일이 위치해 있다.
conf	server.xml 파일을 포함한 톰캣 설정 파일이 위치해 있다.
lib	톰캣을 실행하는데 필요한 라이브러리(.jar) 파일이 위치해 있다.
logs	톰캣 로그 파일이 위치한다.
temp	톰캣이 실행되는 동안 임시 파일이 위치한다.
webapps	웹 어플리케이션이 위치한다.
work	톰캣이 실행되는 동안 사용되는 작업 파일이 위치한다.



2.1.9 톰캣 설치 및 환경 구축

▶ 환경변수

환경변수	설명	필수	예시
JAVA_HOME	JDK 설치 디렉토리	필수	c:\jdk1.6.0
CATALINA_HOME	톰캣 설치 디렉토리. 설정하지 않은 경우 현재 디렉토리를 값으로 사용한다.	아님	c:\apache-tomcat-6.0.18
CLASSPATH	클래스 파일의 위치	필수	%CATALINA_HOME%\lib\servlet-api.jar

set_classpath.bat 파일 내용

```
@echo off
rem CLASSPATH 환경 변수 설정
set CATALINA_HOME = c:\apache-tomcat-6.0.18
set CLASSPATH = % CATALINA_HOME %\webapps\chap02\WEB-INF\classes
set CLASSPATH = %CLASSPATH%;% CATALINA_HOME %\lib\servlet-api.jar
```


2.2 웹 어플리케이션 개발 시작하기

- ▶ 웹 어플리케이션 디렉토리 구성
 - ▶ [웹컨테이너디렉토리]/webapps 아래에 어플리케이션 디렉토리 구성
 - ▶ c:\apache-tomcat-6.0.18\webapps\chap02
 - ▶ url : http://localhost:8080/chap02/...



2.2 웹 어플리케이션 개발 시작하기

p.36

▶ 간단한 JSP 작성하기

- ▶ <http://localhost:8080/chap02/now.jsp>

```
# fileName : chap02\now.jsp

<%@ page contentType = "text/html; charset=UTF-8" %>
<%@ page import = "java.util.Date" %>
<%
    Date now = new Date();
%>

<html>
<head><title>현재 시간</title></head>
<body>
현재 시각 : <%= now %>
</body>
</html>
```

2.2 웹 어플리케이션 개발 시작하기

p.33

- ▶ 간단한 서블릿 작성하기
 - ▶ 서블릿 소스 코드를 저장할 디렉토리 생성.
 - ▶ 클래스 파일을 저장할 WEB-INF\classes 디렉토리 생성.
 - ▶ CLASSPATH 환경 변수 값을 설정.
 - ▶ 서블릿 소스 코드 작성.
 - ▶ 소스 코드 컴파일 후, 생성된 클래스 파일을 classes 디렉토리에 복사.
 - ▶ `c:\...\WEB-INF\src>javac -d ../classes servletName.java`
 - ▶ WEB-INF\web.xml 파일에 서블릿 정보를 설정.
 - ▶ 웹 컨테이너 시작.
 - ▶ 웹 브라우저에서 테스트.
 - ▶ `http://localhost:8080/chap02/now`



2.2 웹 어플리케이션 개발 시작하기

p.33

▶ 간단한 서블릿 작성하기

```
# fileName : chap02\WEB-INF\src\NowServlet.java
```

```
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Date;
```

```
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
```

```
public class NowServlet extends HttpServlet{
    @Override
    protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{

        response.setContentType("text/html; charset=UTF-8");
        Date now = new Date();
        PrintWriter writer = response.getWriter();
        writer.println("<html>");
        writer.println("<head><title>현재 시간</title></head>");
        writer.println("<body>");
        writer.println("현재 시각 :");
        writer.println(now.toString());
        writer.println("</body>");
        writer.println("</html>");

        writer.close();
    }
}
```

2.2 웹 어플리케이션 개발 시작하기

p.34

▶ 간단한 서블릿 작성하기 - web.xml에 서블릿 등록하기

```
# fileName : chap02\WEB-INF\src\NowServlet.java

<?xml version="1.0" encoding="utf-8" ?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  version="2.5">

  <servlet>
    <servlet-name>now</servlet-name>
    <servlet-class>NowServlet</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>now</servlet-name>
    <url-pattern>/now</url-pattern>
  </servlet-mapping>

</web-app>
```

NowServlet 클래스를 'now' 라는 이름으로 등록한다.

URL이 /now인 경우 now 서블릿이 처리하도록 매핑한다.