

Chapter 02

Servlet과 JSP의 이해

강의 목적

■ 강의 목적

- ▶ 웹 서버에서 동작하는 백엔드(backend) 프로그램 개발 방법을 학습
- ▶ Java를 기반으로 Servlet/JSP 프로그래밍 학습

■ 강의 내용

• Servlet과 JSP

- ▶ 서블릿과 JSP의 개요와 작성방법

• JSP 기본 문법

- ▶ 내장 객체, 액션 태그, 쿠키와 세션, 자바 빈즈 등

• 데이터베이스와 JDBC

- ▶ MySQL과 JDBC를 이용한 데이터베이스 연산 방법

웹 클라이언트와 웹 서버의 동작

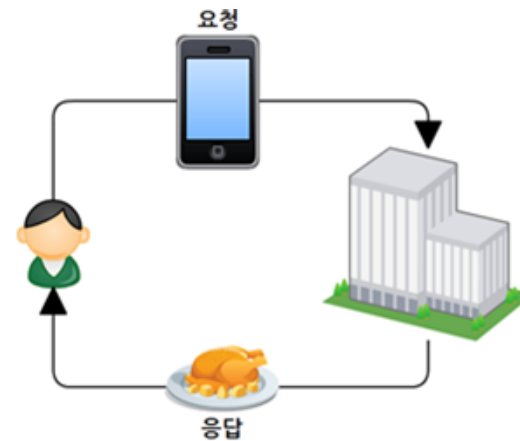
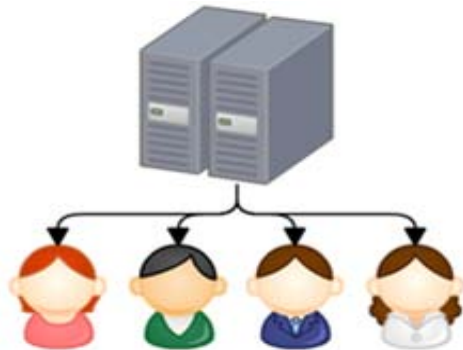
■ 웹은 클라이언트/서버 방식으로 동작

• 웹 클라이언트

- ▶ 서버에 자료를 요청하는 주체
- ▶ 브라우저, 모바일 기기 등

• 웹 서버

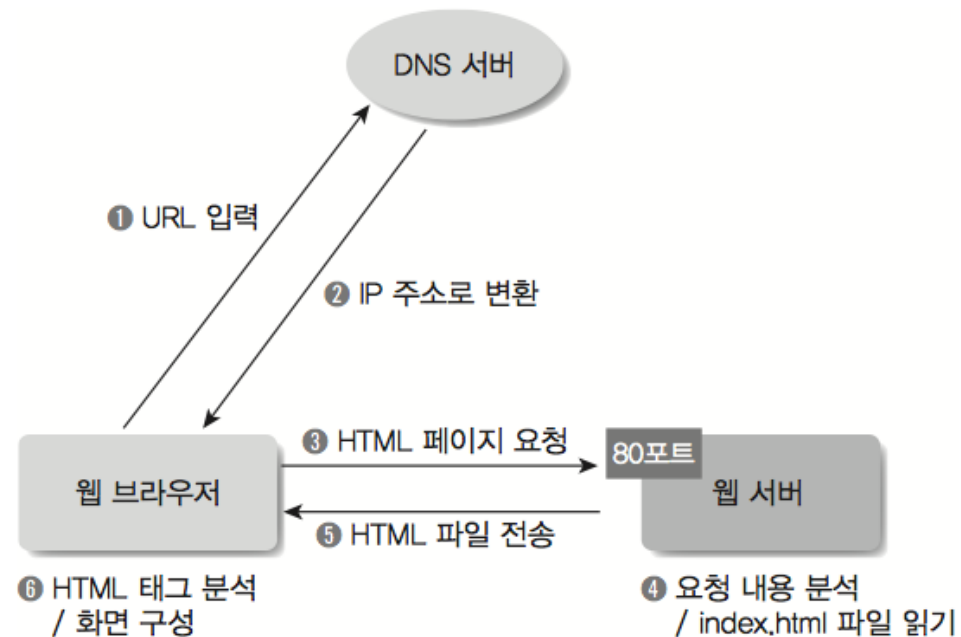
- ▶ 클라이언트의 요구를 처리한 다음 전달해 주는 주체
- ▶ Apache, Tomcat, IIS 등



웹 클라이언트와 웹 서버의 동작

• 웹 서비스의 동작 과정

- ① 웹 브라우저에서 `http://www.xxx.com/index.html`을 입력
- ② `www.xxx.com` 도메인의 IP 주소를 DNS 서버로부터 받음
- ③ IP 주소의 해당 서버 80번 포트로 접속을 시도
- ④ 웹 서버는 요청 내용을 분석하고 요청된 `index.html` 파일을 디스크에서 읽음
- ⑤ 웹 서버는 파일 내용을 텍스트 그대로 요청한 클라이언트에 전송
- ⑥ 웹 브라우저는 웹 서버에서 보내는 텍스트 내용 중 HTML 태그를 분석해 적절히 변환하여 화면을 구성



정적 웹 프로그래밍 & 동적 웹 프로그래밍

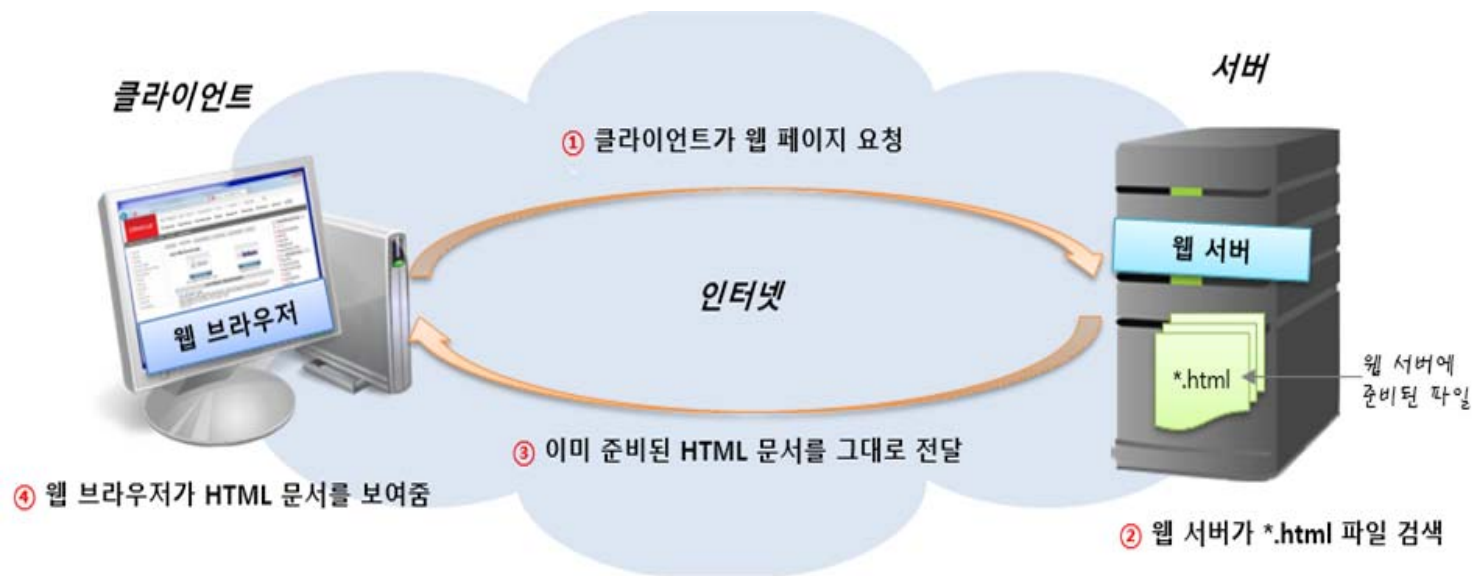
■ 정적 웹 프로그래밍 (static web programming)

- 정적 웹 페이지 (static web page)를 작성하기 위한 프로그래밍

- 정적 웹 페이지

- ▶ 항상 동일한 내용을 출력하는 웹 페이지
- ▶ 이미지나 텍스트로 구성된 HTML 문서

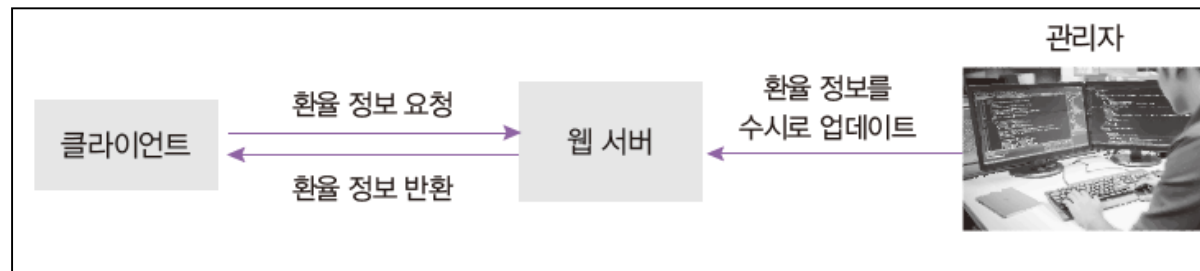
```
<% out.println("<h1>안녕하세요</h1>"); %>
```



정적 웹 프로그래밍 & 동적 웹 프로그래밍

• 정적 웹 프로그래밍의 특징

- ▶ 웹 서버에 미리 보여줄 HTML 페이지, CSS, 이미지, 자바스크립트 파일을 저장
 - 브라우저에서 요청 할 경우 그대로 전달
- ▶ 사용자는 페이지가 변경되지 않는 한 고정된 웹 페이지를 보게 됨
 - 주로 화면의 디자인을 구성하거나 클라이언트의 이벤트를 처리함
 - 환율 정보나 주가 정보 등 실시간 정보를 표시하기에는 적합하지 않는 방식



실시간으로 변하는 환율 정보나 주가 정보를 제공하기 위해서는 관리자가 수작업으로 HTML 코드를 주기적으로 업데이트해야함

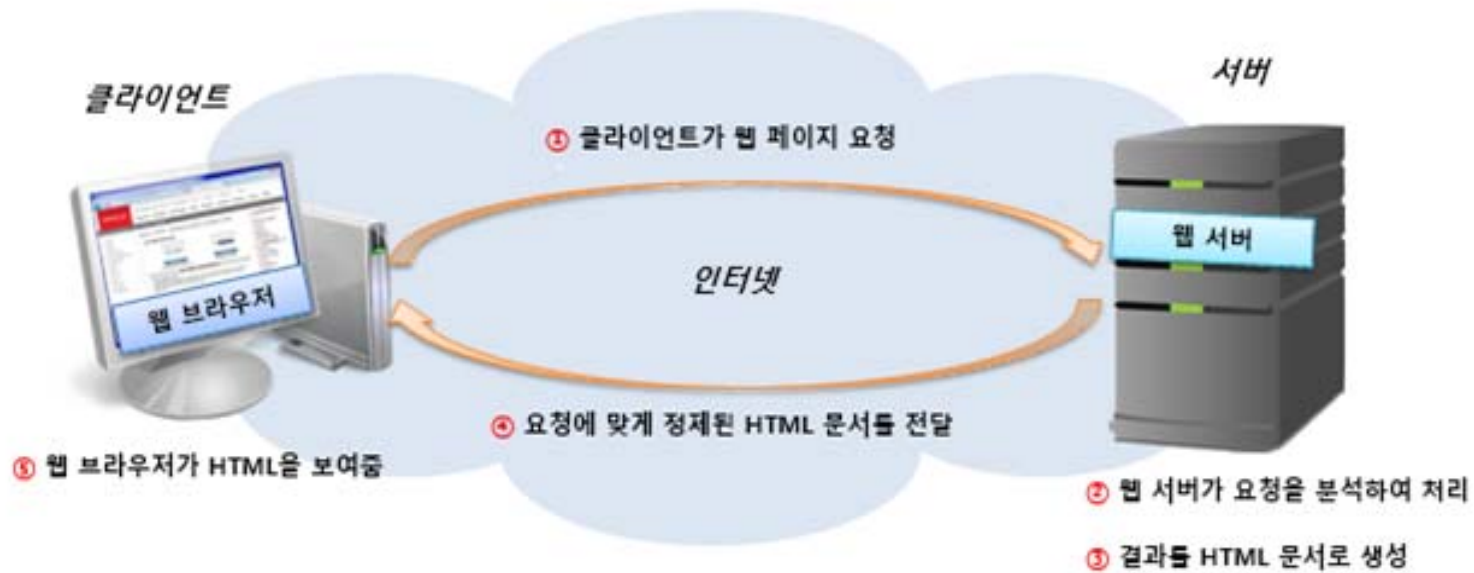
- 사용자에게 디자인같은 고정된 정보만 제공
- 정보 수정 시 관리자가 직접 HTML 소스를 수정하여 사용자에게 정보를 제공
- 실시간 정보를 제공받기를 바라는 사용자들에게는 적합하지 않는 방식

정적 웹 프로그래밍 & 동적 웹 프로그래밍

■ 동적 웹 프로그래밍(dynamic web programming)

- 동적 웹 페이지(dynamic web page)를 작성하기 위한 프로그래밍
- 동적 웹 페이지
 - ▶ 동일한 소스 코드이지만 조건이나 상태에 따라 서로 다른 내용을 출력하는 웹 페이지

```
<%= new java.util.Date() %>
```



정적 웹 프로그래밍 & 동적 웹 프로그래밍

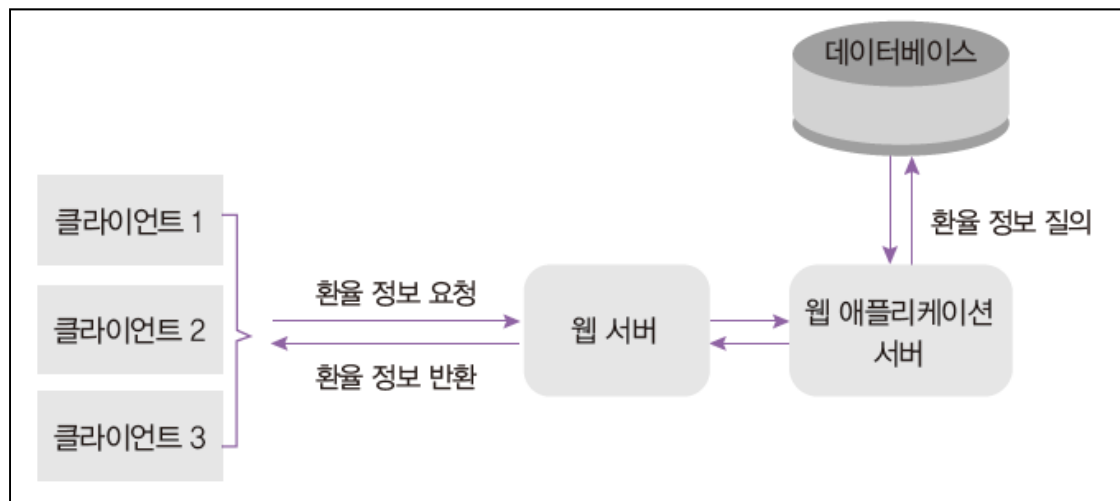
■ 동적 웹 프로그래밍의 특징

- 정적 웹프로그래밍에서 관리자 역할을 웹 애플리케이션 서버(WAS : Web Application Server)가 수행
 - ▶ 클라이언트의 요청이 있을 때마다 데이터베이스에 접근해서 실시간 정보를 얻어와서 클라이언트에게 전송
- CGI 동작 방식과 어플리케이션 동작 방식이 있음
 - ▶ 초기에는 동적 방식으로 프로그램을 제공하는 기능은 CGI(Common Gate Interface)를 사용함
 - ▶ CGI 방식은 프로세스(process)를 기반으로 동작
 - ▶ 프로그래밍 언어인 펄(Perl)이 대표적인 CGI언어임
- 차후에는 어플리케이션 동작 방식을 사용함
 - ▶ 어플리케이션 방식은 스레드(thread)를 기반으로 동작
 - ▶ JSP는 어플리케이션 동작 방식으로 운영됨

정적 웹 프로그래밍 & 동적 웹 프로그래밍

- 동적 웹 프로그래밍의 예

- ▶ 클라이언트의 요청이 있을 때마다 데이터베이스에 접근해서 실시간 환율 정보를 얻어와서 클라이언트에게 전송
- ▶ 그 역할을 웹 어플리케이션 서버가 담당함



일반 프로그램과 웹 프로그램

■ 일반 프로그램과 웹 프로그램

• 일반적인 프로그램

- ▶ 컴퓨터에 설치 후 사용(스마트폰, 태블릿도 동일함)
- ▶ 기본적으로 해당 디바이스에서 프로그램이 실행되고 경우에 따라 필요한 데이터는 서버로부터 수신함

• 웹 프로그램

- ▶ 별도의 설치 없이 서버에 접속하는 것만으로 필요한 기능/서비스를 이용할 수 있음
 - 프로그램은 서버에서 실행되고 실행 결과만 컴퓨터의 브라우저를 통해 보여짐.
 - 데이터 입력, 메뉴 선택 등 사용자와의 상호작용을 처리해야 되는 프로그램적인 요소도 있음

• 웹 프로그램은 서버와 클라이언트의 협력에 의해 구현됨.

- ▶ 웹 프로그램 개발을 위해서는 서버 프로그래밍 기술과 클라이언트 프로그래밍 기술을 모두 알아야 함

구분	종류
클라이언트 기술	HTML, 자바스크립트, CSS
서버 기술	서블릿, JSP, ASP.Net, PHP
클라이언트/서버 공통 기술	자바, C, C#.Net

일반 프로그램과 웹 프로그램

■ 클라이언트 기술

- 기본적으로 브라우저에 의해 처리되는 요소

- 화면 구성, 스타일, 동적 이벤트 처리 등 사용자 상호작용 담당

- HTML(Hyper Text Markup Language)

- 웹 서비스를 표현하기 위해 사용하는 언어로 <HTML></HTML> 과 같은 마크업 구조.
- HTML5는 차세대 웹 개발 언어로 사용되고 있음.

- 자바스크립트(JavaScript)

- 자바와 유사한 문법구조를 제공하는 웹 클라이언트 개발 언어(실제 자바와는 무관함)
- 웹 브라우저에서 해석되며 웹 브라우저 성능 평가에서 매우 중요한 요소가 됨
 - Ajax(Asynchronous Javascript and XML) 기법을 사용한 비동기 서버 연결 기술이 많이 사용되고 있음
 - jQuery 와 같은 공개 라이브러리가 최근 많이 사용됨
 - JSON(JavaScript Object Notation)은 클라이언트 서버간 정보 교환에 널리 사용.

- CSS(Cascading Style Sheet)

- HTML에서 레이아웃이나 디자인 요소를 분리
- 최근 웹 클라이언트 개발은 HTML5+CSS3+JavaScript(jQuery) 임

일반 프로그램과 웹 프로그램

■ 서버 기술

- HTML 파일은 파일 내용을 수정하기 전까지는 내용이 변하지 않는 정적인 구조
 - ▶ 사용자 요청에 따라 다른 정보를 제공하거나 데이터베이스를 통한 서비스를 위해서는 별도의 프로그램 기술이 필요
- ASP.Net
 - ▶ MS 윈도우 기반의 서버 프로그램 기술 언어로 .Net 기반의 컴포넌트 사용 가능.
 - ▶ MS 윈도우 종속과 상용 라이선스 정책으로 대규모 공개 웹 서비스 개발에는 자주 사용되지 않음
- PHP(Professional Hypertext Preprocessor)
 - ▶ 처리속도가 빠르고 다양한 운영체제와 웹 서버 환경에서 실행 가능.
 - APM = Apache + PHP + MySQL
 - ▶ 오픈소스 게시판, 위키 프로그램, 연구용으로 널리 사용됨.
- JSP(Java Server Page)
 - ▶ 자바 서블릿 기반의 웹 프로그래밍 기술
 - ▶ 자바의 모든 기능을 사용할 수 있으며 안정성과 확장성이 뛰어나 가장 많이 사용되고 있음

Servlet과 JSP의 관계

Servlet과 JSP의 관계

■ 서블릿(Servlet)과 JSP(Java Server Page)

- 서블릿은 자바를 이용한 서버 프로그래밍 기술임
 - 초기 웹 프로그래밍 기술인 CGI(Common Gateway Interface)를 대체하기 위해 개발
- 느린 처리 속도, 많은 메모리 요구, 불편한 화면 제어 등의 한계 발생
 - PHP, ASP 등 서버 스크립트 언어 등장
- JAVA에 대한 지식이 많아야 개발이 가능
 - 다른 언어에 비해 초보자의 접근이 어려움
- 서블릿을 대체하기 위해 JSP가 개발됨
 - JSP는 PHP와 유사한 형태이지만 내부적으로는 Servlet으로 변환되어 실행됨
 - HTML을 중심으로 자바 프로그램과의 유기적인 연결을 지원.

Servlet과 JSP의 관계

■ Servlet과 JSP의 비교

```
public class HelloWorldServlet extends HttpServlet {
```

Servlet

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

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

```
        out.println("<HTML><HEAD><TITLE>로그인 </TITLE></HEAD>");
        out.println("<BODY><H2>Hello World : 헬로월드</H2>");
        out.println("오늘의 날짜와 시간은 : "+new java.util.Date());
        out.println("</BODY></HTML>");
```

```
    }
```

```
}
```

```
<%@ page contentType="text/html; charset=utf-8" %>
```

```
<HTML>
```

```
    <HEAD><TITLE>Hello World</TITLE></HEAD>
```

```
    <BODY>
```

```
        <H2>Hello World : 헬로월드</H2>
```

```
        오늘의 날짜와 시간은 : <%= new java.util.Date() %>
```

```
    </BODY>
```

```
</HTML>
```

JSP

Servlet과 JSP의 관계

- Servlet

- ▶ 자바 클래스 형태를 취하고 있음
- ▶ 화면에 출력하고자 하는 모든 내용은 `out`객체의 `println()` 메서드를 사용해야 함
 - 화면에 출력하고자 하는 내용이 많을 수록 코드가 복잡해짐

```
public class HelloWorldServlet extends HttpServlet {  
  
    public void doGet(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException {  
  
        response.setContentType("text/html; charset=EUC_KR");  
        PrintWriter out = response.getWriter();  
  
        out.println("<HTML><HEAD><TITLE>로그인 </TITLE></HEAD>");  
        out.println("<BODY><H2>Hello World : 헬로월드</H2>");  
        out.println("오늘의 날짜와 시간은 : "+new java.util.Date());  
        out.println("</BODY></HTML>");  
  
    }  
  
}
```


Servlet과 JSP의 관계

- JSP

- ▶ 스크립트 언어 구조

- HTML문서 내에 스크립트 형태로 사용됨
 - JAVA의 모든 기능을 사용할 수 있지만 직관적인 구조로 쉽게 작성할 수 있음

```
<%@ page contentType="text/html; charset=utf-8" %>

<HTML>

  <HEAD><TITLE>Hello World</TITLE></HEAD>

  <BODY>
    <H2>Hello World : 헬로월드</H2>
    오늘의 날짜와 시간은 : <%= new java.util.Date() %>
  </BODY>

</HTML>
```

- Servlet을 대체하는 기술로 등장했지만 상호 보완하는 기술로 사용되고 있음

Servlet과 JSP의 관계

■ JSP의 특징

- 자바의 모든 기능을 사용할 수 있음
 - 자바에서 사용되는 모든 패키지와 클래스를 사용할 수 있음
 - 다른 언어는 API 기반이므로 기능에 한계가 있음
- 서블릿으로 컴파일된 후 메모리에서 처리됨
 - 많은 사용자의 접속도 원활하게 처리할 수 있음
 - JSP 또는 다른 서블릿 간의 데이터를 쉽게 공유 할 수 있음
 - JSP는 Servlet으로 변환되어 실행되기 때문
- 빈즈(Beans)라고 하는 자바 컴포넌트를 사용할 수 있음
- 커스텀 태그를 만들어 사용할 수 있음
 - 또한 JSTL(JSP Standard Tag Library)과 같은 태그 라이브러리를 이용할 수 있음
- 스트러츠, 스프링 @MVC 등 다양한 프레임워크와 결합하여 개발할 수 있음

Servlet과 JSP의 관계

- [참고] 자바 관련 기본 기술 경험의 요구 수준

기술	설명	요구 수준
자바	소스코드를 작성하기 위한 프로그래밍 기본 언어로서, Java SE를 기준으로 한다.	<ul style="list-style-type: none">• 자바 기본• 상속, 오버로딩, 오버라이딩• java.util, java.io 패키지• 예외 핸들링• 객체지향 개념• 인터페이스 구현• 스레드
JDBC	Java DataBase Connectivity의 약자로서, 자바에서 데이터베이스 프로그래밍을 하기 위한 기술이다.	<ul style="list-style-type: none">• JDBC 드라이버 세팅• PreparedStatement• 기초 SQL문• ResultSet• 데이터 핸들링
서블릿	JSP의 기본이 되는 자바 기반의 웹 프로그래밍 핵심 기술이다.	<ul style="list-style-type: none">• 서블릿 구조 이해• request, response 처리• 간단한 서블릿 프로그래밍• GET/POST 처리

• [참고] 고급 웹 프로그래밍을 위한 주변 기술 경험의 요구 수준

기술	필요성	요구 수준
데이터베이스	프로그램의 데이터를 처리하려고 할 때 반드시 필요하다.	<ul style="list-style-type: none"> • 다양한 SQL문의 사용 • 데이터베이스 연계 프로그래밍 경험 • 데이터베이스 함수 및 내장 프로시저
XML	eXtensible Markup Language의 약자로서, 확장 가능한 구조적 문서 표현을 제공한다. 많은 프로그램에서 데이터 구조를 XML 기반으로 처리한다.	<ul style="list-style-type: none"> • XML 스키마 및 DTD 이해 • XML DOM 개요
모바일 프로그래밍	최근에는 스마트폰을 중심으로 하는 모바일 기반의 개발이 증가하고 있는 추세다.	<ul style="list-style-type: none"> • 안드로이드 혹은 아이폰 앱 개발 경험 • 하이브리드 앱 개발 경험
프레임워크	개발자로 하여금 더욱 좋은 프로그램을 만들 수 있도록 미리 제공되는 틀을 말한다.	<ul style="list-style-type: none"> • 소프트웨어 아키텍처 이해 • 스프링 프레임워크 • 스프링3 @MVC

JSP의 처리 과정

■ 서블릿 컨테이너 내부 과정

- ▶ ① 웹 서버로부터 JSP에 대한 사용자 요청이 컨테이너로 전달
- ▶ ② 요청 JSP에 대한 서블릿이 존재하면 다음 단계로 진행
 - 존재하지 않을 경우 JSP를 java 파일로 변환한 다음 .class 파일로 컴파일
- ▶ ③ 컴파일된 서블릿 클래스를 컨테이너의 메모리에 적재
- ▶ ④ ~ ⑤ 데이터베이스 처리 혹은 별도의 능을 위한 클래스 호출 등이 있다면 실행
 - 결과를 취합해 HTML 형태로 구성
- ▶ ⑥ HTML 형태의 결과를 웹 서버를 경유해 사용자 브라우저에 전달

