



6 서블릿의 라이프 사이클

뇌를 자극하는 JSP & Servlet

❖ 학습목표

- 서블릿 클래스로부터 서블릿 객체가 만들어지고, 서블릿 객체가 초기화 되어서 서블릿이 되고, 서블릿이 사용되고, 최종적으로 소멸되기까지의 전 과정을 서블릿의 라이프 사이클이라고 한다. 이 장에서는 서블릿의 라이프 사이클에 관련된 프로그래밍 기술을 배워보자.

❖ 내용

- 서블릿의 라이프 사이클
- 서블릿 클래스의 `init` 메서드와 `destroy` 메서드
- JSP 페이지의 `jspInit` 메서드와 `jspDestroy` 메서드
- 서블릿의 환경을 표현하는 `ServletContext` 객체



4. 서블릿의 환경을 표현하는 ServletContext 객체

❖ 서블릿의 환경 정보를 가져오는 방법

- 서블릿 클래스나 JSP 페이지의 환경에 관련된 정보는 javax.servlet 패키지의 ServletContext 인터페이스 타입의 객체를 이용해서 얻을 수 있다.
- 서블릿 클래스에서 이 타입의 객체를 구하기 위해서는 getServletContext라는 메서드를 호출하면 된다.

```
ServletContext context = getServletContext();
```

ServletContext 객체를 리턴하는 메서드

- ServletContext 객체에 대해 getServerInfo라는 메서드를 호출하면 서블릿이 속하는 웹 서버 종류가 리턴된다.

```
String str = context.getServerInfo();
```

웹 서버의 종류를 리턴하는 메서드



4. 서블릿의 환경을 표현하는 ServletContext 객체

❖ 서블릿의 환경 정보를 가져오는 방법

- ServletContext 객체에 대해 getMajorVersion과 getMinorVersion이라는 메서드를 호출하면 웹 컨테이너가 지원하는 서블릿 규격서의 메이저 버전과 마이너 버전이 리턴된다.

```
int num1 = context.getMajorVersion();
```

서블릿의 메이저 버전을 가져오는 메서드

```
int num2 = context.getMinorVersion();
```

서블릿의 마이너 버전을 가져오는 메서드



4. 서블릿의 환경을 표현하는 ServletContext 객체

❖ 서블릿의 환경 정보를 가져오는 방법

[예제 6-7] 웹 서버의 정보를 출력하는 JSP 페이지

```
import javax.servlet.http.*;
import javax.servlet.*;
import java.io.*;

public class ServerInfoServlet extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException {
        ServletContext context = getServletContext();
        String serverInfo = context.getServerInfo();
        int majorVersion = context.getMajorVersion();
        int minorVersion = context.getMinorVersion();
        response.setContentType( "text/html;charset=euc-kr ");
        PrintWriter out = response.getWriter();
        out.println( "<HTML> ");
        out.println( "<HEAD><TITLE>웹 서버의 정보</TITLE></HEAD> ");
        out.println( "<BODY> ");
        out.printf( "웹 서버의 종류: %s <BR> ", serverInfo);
        out.printf( "지원하는 서블릿 버전: %d.%d <BR> ", majorVersion, minorVersion);
        out.println( "</BODY> ");
        out.println( "</HTML> ");
    }
}
```



4. 서블릿의 환경을 표현하는 ServletContext 객체

❖ 서블릿의 환경 정보를 가져오는 방법

The screenshot shows an IDE window with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with a 'WEB-INF' directory. The 'WEB-INF' directory contains a 'classes' directory and a 'web.xml' file. The 'classes' directory is highlighted with a red arrow pointing to it. The 'web.xml' file is also highlighted with a red arrow pointing to it. The code editor shows the contents of the 'web.xml' file, which is an XML configuration file for a web application. The XML content is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app>
  <servlet>
    <servlet-name>server-info-servlet</servlet-name>
    <servlet-class>ServerInfoServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>server-info-servlet</servlet-name>
    <url-pattern>/server-info</url-pattern>
  </servlet-mapping>
</web-app>
```

① brain06 웹 애플리케이션 디렉터리의 WEB-INF 서브디렉터리로 가세요.

② (예제 b-3) 의 컴파일 결과를 classes 디렉터리에 저장하세요.

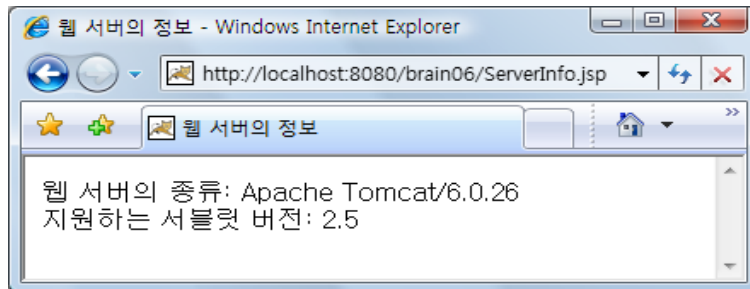
③ web.xml 파일을 열어서 다음과 같은 두 엘리먼트를 추가하세요

4. 서블릿의 환경을 표현하는 ServletContext 객체

❖ 서블릿의 환경 정보를 가져오는 방법

[예제 6-8] application 내장 변수의 사용 예를 보여주는 JSP 페이지

```
<%@page contentType= "text/html; charset=euc-kr" %>
<HTML>
  <HEAD><TITLE>웹 서버의 정보</TITLE></HEAD>
  <BODY>
    웹 서버의 종류: <%= application.getServerInfo() %> <BR>
    지원하는 서블릿 버전: <%= application.getMajorVersion() %>.<%= application.getMinorVersion() %> <BR>
  </BODY>
</HTML>
```



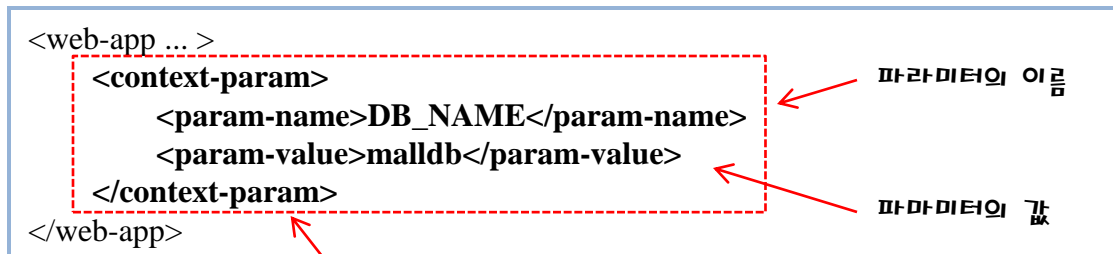
[그림 6-17] 예제 6-6의 실행 결과



4. 서블릿의 환경을 표현하는 ServletContext 객체

❖ 웹 애플리케이션의 초기화 파라미터 값을 가져오는 getInitParameter 메서드

- web.xml 파일에 웹 애플리케이션 전체에 속하는 초기화 파라미터를 기재할 수도 있다. web.xml 파일의 루트 엘리먼트인 <web-app> 엘리먼트 바로 아래 <context-param>이라는 이름의 서브엘리먼트를 쓰고, 그 안에 <param-name>과 <param-value>라는 서브엘리먼트를 쓰면 된다.



웹 애플리케이션의 초기화 파라미터

```
String str = application.getInitParameter( "DB_NAME " );
```

초기화 파라미터의 이름

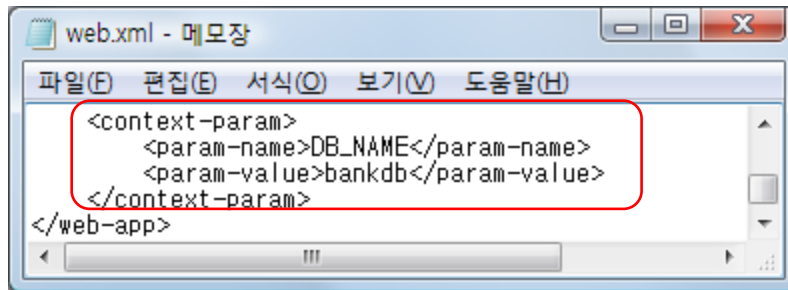


4. 서블릿의 환경을 표현하는 ServletContext 객체

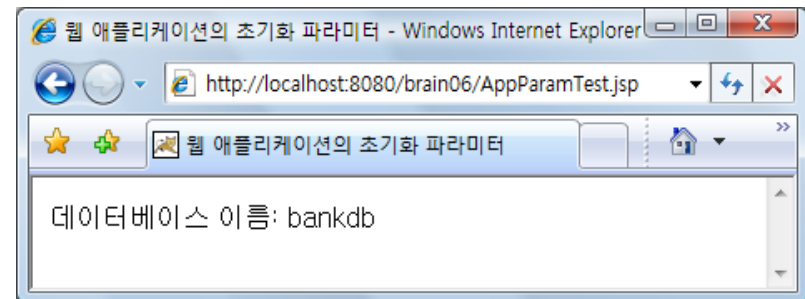
❖ 웹 애플리케이션의 초기화 파라미터 값을 가져오는 getInitParameter 메서드

[예제 6-9] 웹 애플리케이션의 초기화 파라미터를 읽는 JSP 페이지

```
<%@page contentType= "text/html; charset=euc-kr" %>
<HTML>
  <HEAD><TITLE>웹 애플리케이션의 초기화 파라미터</TITLE></HEAD>
  <BODY>
    데이터베이스 이름: <%= application.getInitParameter("DB_NAME ") %> <BR>
  </BODY>
</HTML>
```



[그림 6-22] 웹 애플리케이션의 초기화 파라미터 지정 방법



[그림 6-23] 예제 6-9의 실행 결과

4. 서블릿의 환경을 표현하는 ServletContext 객체

❖ 로그 메시지를 기록하는 log 메서드

- ServletContext 인터페이스의 log 메서드를 이용하면 로그 파일에 메시지를 기록할 수 있다.
- log 메서드를 호출할 때는 다음과 같이 파라미터로 로그 메시지를 넘겨주어야 한다.

```
application.log( “인사하기 JSP 페이지가 호출되었습니다.” );
```

로그 메시지

- log 메서드는 파라미터로 넘겨준 메시지를 톰캣의 설치 디렉터리 아래의 logs라는 이름의 서브디렉터리 안의 localhost.yyyy-mm-dd.log라는 이름의 파일에 기록한다.



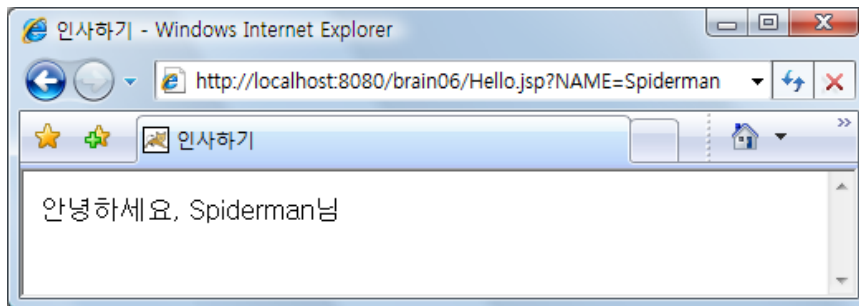
4. 서블릿의 환경을 표현하는 ServletContext 객체

❖ 로그 메시지를 기록하는 log 메서드

[예제 6-10] log 메서드의 사용 예를 보여주는 JSP 페이지

```
<%@page contentType= "text/html; charset=euc-kr " %>
<HTML>
  <HEAD><TITLE>인사하기</TITLE></HEAD>
  <BODY>
    안녕하세요, <%= request.getParameter( "NAME " ) %>님
    <% application.log( "[인사하기] JSP 페이지가 호출되었습니다." ); %>
  </BODY>
</HTML>
```

로그 파일에 기록을
합니다



[그림 6-24] 예제 6-10의 실행 결과



4. 서블릿의 환경을 표현하는 ServletContext 객체

- ❖ 같은 웹 애플리케이션에 속하는 웹 컴포넌트들끼리 데이터를 주고받는 방법
 - ServletContext 인터페이스의 `setAttribute`, `getAttribute`, `removeAttribute` 메서드는 같은 웹 애플리케이션 디렉터리에 있는 웹 컴포넌트들끼리 데이터를 공유할 수 있도록 만드는 메서드이다.
 - `setAttribute` 메서드는 웹 애플리케이션에 할당된 공유 데이터 영역에 데이터를 저장하는 기능을 한다.
 - `getAttribute` 메서드는 그 영역에 있는 데이터를 읽어오는 기능을 한다.
 - `removeAttribute` 메서드는 그 영역의 데이터를 삭제하는 기능을 한다.

```
application.setAttribute( "ID ", "lee77 ");
```

데이터 이름

데이터 값

```
String str = (String) application.getAttribute( "ID ");
```

데이터 이름

```
application.removeAttribute( "ID ");
```



4. 서블릿의 환경을 표현하는 ServletContext 객체

❖ 같은 웹 애플리케이션에 속하는 웹 컴포넌트들끼리 데이터를 주고받는 방법

[예제 6-11] 웹 애플리케이션의 데이터 영역에 데이터를 저장하는 JSP 페이지

```
<%@page contentType= "text/html; charset=euc-kr" %>
<%
    String name = request.getParameter( "NAME " );
    application.setAttribute( "NAME ", name);
%>
<HTML>
    <HEAD><TITLE>웹 애플리케이션 데이터 저장하기</TITLE></HEAD>
    <BODY>
        NAME 데이터가 저장되었습니다.
    </BODY>
</HTML>
```

[예제 6-12] 웹 애플리케이션의 데이터 영역으로부터 데이터를 읽어오는 JSP 페이지

```
<%@page contentType= "text/html; charset=euc-kr" %>
<HTML>
    <HEAD><TITLE>웹 애플리케이션 데이터 조회하기</TITLE></HEAD>
    <BODY>
        이름: <%= application.getAttribute( "NAME " ) %>
    </BODY>
</HTML>
```

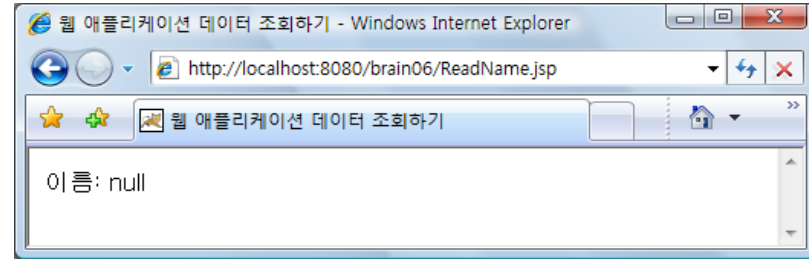
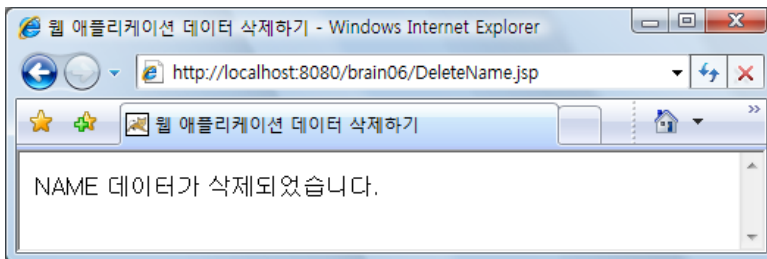
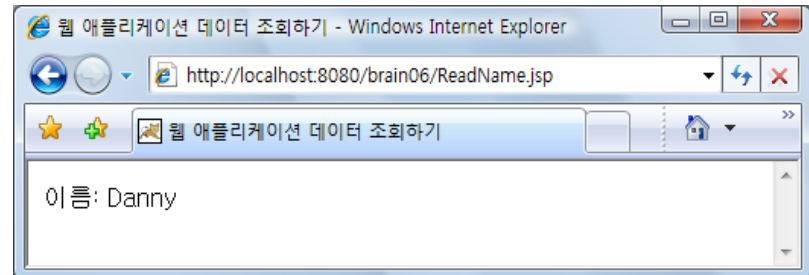
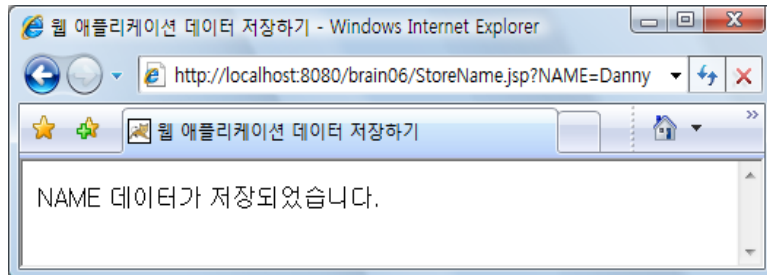


4. 서블릿의 환경을 표현하는 ServletContext 객체

❖ 같은 웹 애플리케이션에 속하는 웹 컴포넌트들끼리 데이터를 주고받는 방법

[예제 6-13] 웹 애플리케이션의 데이터 영역에 있는 데이터를 삭제하는 JSP 페이지

```
<%@page contentType= "text/html; charset=euc-kr" %>
<% application.removeAttribute( "NAME" ); %>
<HTML>
  <HEAD><TITLE>웹 애플리케이션 데이터 삭제하기</TITLE></HEAD>
  <BODY>
    NAME 데이터가 삭제되었습니다.
  </BODY>
</HTML>
```



[그림 6-26] 예제 6-11, 예제 6-12, 예제 6-13의 실행 결과

4. 서블릿의 환경을 표현하는 ServletContext 객체

❖ 웹 애플리케이션에 관련된 파일 경로명을 가져오는 메서드

- ServletContext 인터페이스에는 웹 애플리케이션의 URL 경로명을 리턴하는 `getContextPath`라는 메서드도 있다.

```
String appPath = application.getContextPath();
```

웹 애플리케이션의 URL 경로명을
리턴하는 메서드

- ServletContext 인터페이스의 `getRealPath` 메서드는 웹 애플리케이션 디렉터리 내의 파일 경로명을 파일 시스템 전체에 대한 절대 경로명으로 바꾸어서 리턴하는 메서드이다.

```
String absolutePath = application.getRealPath( "/sub1/Intro.html " );
```

웹 애플리케이션 내에서의
파일 경로명





Thank You !

뇌를 자극하는 JSP & Servlet