

## / 서블릿의 라이프 사이클

뇌를 자극하는 JSP & Servlet



#### **Contents**

### ❖ 학습목표

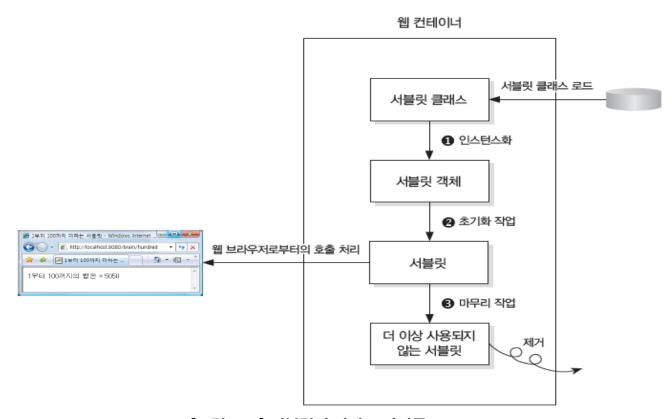
서블릿 클래스로부터 서블릿 객체가 만들어지고, 서블릿 객체가 초기화되어서 서블릿이 되고, 서블릿이 사용되고, 최종적으로 소멸되기까지의전 과정을 서블릿의 라이프 사이클이라고 한다. 이 장에서는 서브릿의 라이프 사이클에 관련된 프로그래밍 기술을 배워보자.

#### ❖ 내용

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

## 1. 서블릿의 라이프 사이클

- 서블릿 클래스는 웹 브라우저에 의해 바로 호출되는 것이 아니라 서블릿 클래스로부터 서블릿 객체가 만들어지고, 그 객체가 웹 컨테이너에 의해 초기화된 다음에 호출된다.
- 웹 브라우저의 요청을 처리할 수 있는 상태의 서블릿 객체를 서블릿이라고 한다.



[그림 6-1] 서블릿의 라이프 사이클

## 1. 서블릿의 라이프 사이클

- 웹 컨테이너는 서블릿을 언제 제거할까? 웹 컨테이너는 자신이 종료되기 전이나 웹 애 플리케이션을 리로드(unload) 하기 전에 그에 속하는 모든 서블릿을 제거한다.
- 서블릿 라이프 사이클 전체에 거쳐서 한번만 실행되어야 할 코드는 서블릿 클래스 안에 init 메서드와 destroy라는 메서드를 선언하고 그 안에 써 놓으면 된다.

## 1. 서블릿의 라이프 사이클

JSP 기술에서는 초기화 작업과 마무리 작업 단계에 해야 할 일을 jspInit와 jspDestroy 메서드 안에 써놓으면 웹 컨테이너에 의해 자동으로 호출된다.

```
JSP 페이지로부터 변환된 서블릿이
<%!
                                                          초기화될 때 해야 할 일을 기술하는 부분
 public void jspInit() {
%>
<HTML>
 <HEAD><TITLE>Hello</TITLE></HEAD>
 <BODY>
   Hello, Everyone.
 </BODY>
</HTML>
<%!
                                                          JSP 페이지로부터 변화된 서블릿이
 public void jspDestroy() {
                                                          제거되기 전에 해야 할 일을 기술하는 부분
%>
               [그림 6-3] JSP 페이지의 jspInit 메서드와 jspDestroy 메서드
```

#### ❖ init 메서드의 작성 방버

 init 메서드는 파라미터가 없는 메서드로 선언해야 하고, 리턴 타입은 void로 지정해야 하며, public 메서드로 선언해야 한다.

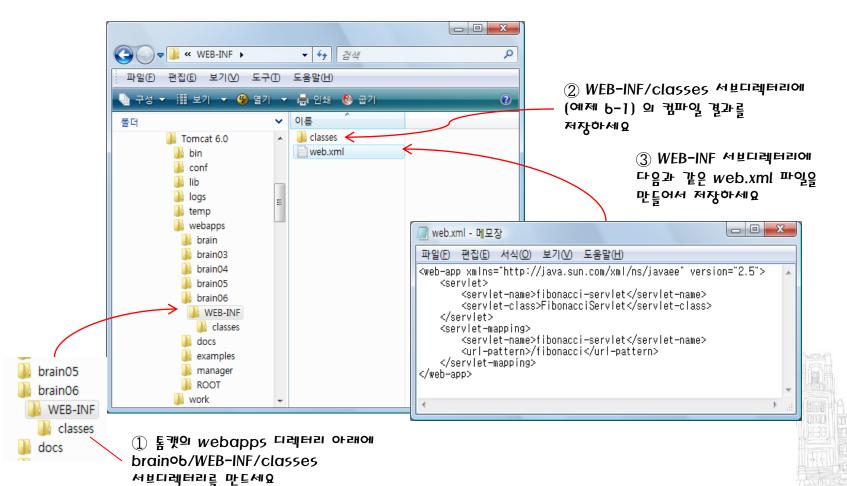
 위의 점선으로 표시된 부분에 서블릿 클래스의 초기화 작업 중에 실행해야 할 코드를 써 놓으면 init 메서드가 완성된다.

#### ❖ init 메서드의 작성 방법

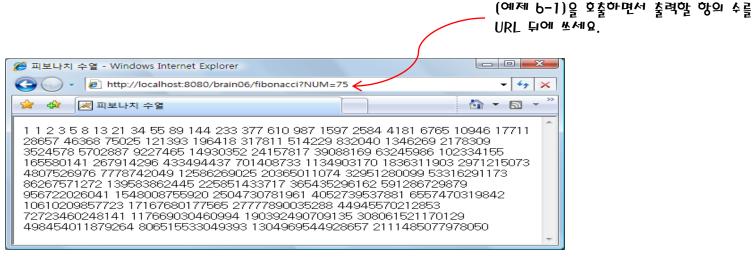
```
[예제6-1] 피보나치 수열을 출력하는 서블릿 클래스
import javax.servlet.http.*;
import javax.servlet.*;
import java.io.*;
import java.math.BigInteger;
public class FibonacciServlet extends HttpServlet {
  private BigInteger arr[];
  public void init() {
    arr = new BigInteger[100];
    arr[0] = new BigInteger( "1");
    arr[1] = new BigInteger( "1 ");
    for (int cnt = 2; cnt < arr.length; cnt++)
      arr[cnt] = arr[cnt-2].add(arr[cnt-1]);
  public void doGet(HttpServletRequest request, HttpServletResponse response)
                  throws IOException, ServletException {
    String str = request.getParameter("NUM");
    int num = Integer.parseInt(str);
    if (num > 100)
      num = 100:
    response.setContentType( "text/html;charset=euc-kr ");
    PrintWriter out = response.getWriter();
    out.println( "<HTML>");
    out.println( "<HEAD><TITLE>피보나치 수열</TITLE></HEAD> ");
    for (int cnt = 0; cnt < num; cnt++)
      out.println(arr[cnt] + "");
    out.println( "</BODY>");
    out.println("</HTML>");
```

#### ❖ init 메서드의 작성 방법

톰캣의 webapps 디렉터리 아래에 brain06이라는 이름의 웹 애플리케이션 디렉터리를 만들고, 서블릿 클래스를 컴파일한 결과를 다음과 같이 설치하고 등록한다.



#### ❖ init 메서드의 작성 방법

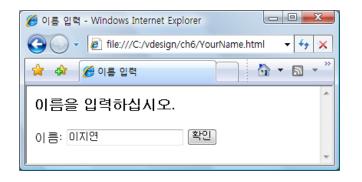


[그림 6-5] 예제 6-1의 실행 결과

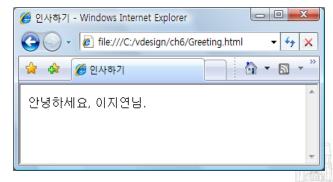


[그림 6-6] 웹 컨테이너가 시작될 때 서블릿이 초기화되도록 만드는 엘리먼트

- destroy 메서드의 작성 방법은 init 메서드와 비슷하다. 파라미터가 없어야 하고, 리턴 타입은 void여야 하고, public 메서드로 선언해야 한다.
- 하지만 init 메서드와 달리 throws 절을 쓸 수 없다.







[그림 6-7] 인사말을 출력하는 웹 애플리케이션의 화면 설계

```
http://localhost:8080/brain06/YourName.html
(그림 b-기)의 왼쪽 화면의 URL

http://localhost:8080/brain06/greeting
(그림 b-기)의 오른쪽 화면의 URL
```

```
[예제6-2] 이름을 입력받는 HTML 문서

<HEAD>

<META http-equiv= "Content-Type" content= "text/html;charset=euc-kr">

<TITLE>이름 입력</TITLE>

</HEAD>

<BODY>

<H3>이름을 입력하십시오.</H3>

<FORM ACTION=greeting>

이름: <INPUT TYPE=TEXT NAME=NAME>

<INPUT TYPE=SUBMIT VALUE= '확인'>

</FORM>

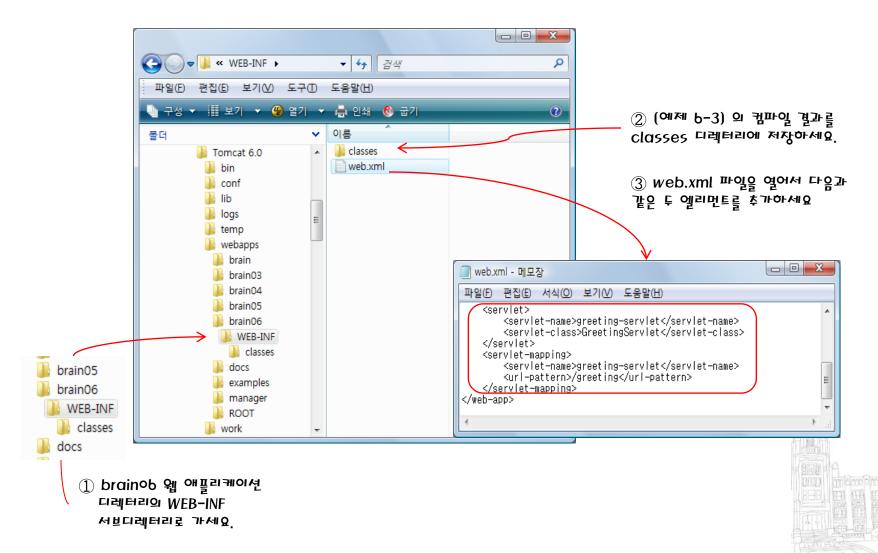
</BODY>

</HTML>
```

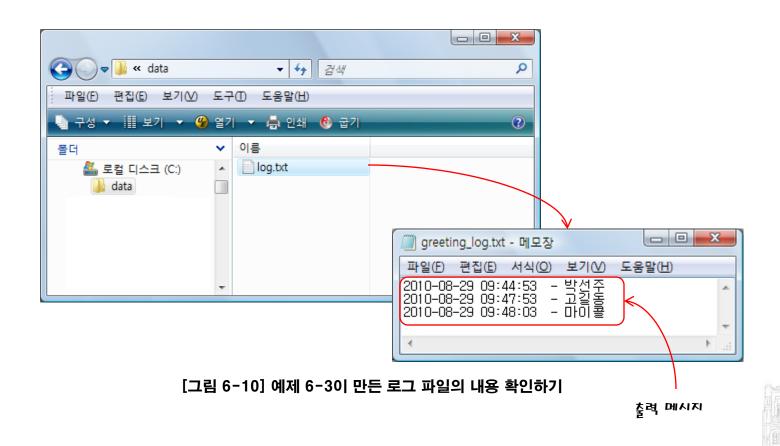
```
[예제6-3] 인사말을 출력하는 서블릿 클래스
import javax.servlet.http.*;
import javax.servlet.*;
import java.io.*;
import java.util.*;
public class GreetingServlet extends HttpServlet {
  private PrintWriter logFile;
  public void init() throws ServletException {
    try {
      logFile = new PrintWriter(new FileWriter("c:\\data\\log.txt ", true));
    catch (IOException e) {
      throw new ServletException(e);
  public void doGet(HttpServletRequest request, HttpServletResponse response)
                     throws IOException, ServletException {
    String name = request.getParameter( "NAME");
    String greeting = "안녕하세요," + name + "님.";
    if (logFile != null) {
      GregorianCalendar now = new GregorianCalendar();
      logFile.printf( "%TF %TT - %s %n ", now, now, name);
    response.setContentType( "text/html;charset=euc-kr ");
    PrintWriter out = response.getWriter();
    out.println( "<HEAD><TITLE>인사하기</TITLE></HEAD> ");
```

```
out.println( "<BODY> ");
out.println(greeting);
out.println( "</BODY> ");
out.println( "</HTML> ");
}
public void destroy() {
   if (logFile != null)
      logFile.close();
}
```

#### ❖ destroy 메서드의 작성 방법



13/38



#### ❖ 서블릿의 초기화 파라미터

서블릿의 초기화 파라미터란 서블릿의 초기화 작업에 필요한 데이터의 의미한다.

- 하나의 서블릿 클래스에 둘 이상의 초기화 파라미터를 넘겨줄 수 있다.
- 서블릿 클래스 안에서 초기화 파라미터 값을 가져오려면 getInitParameter라는 메서드를 호출하면 된다.



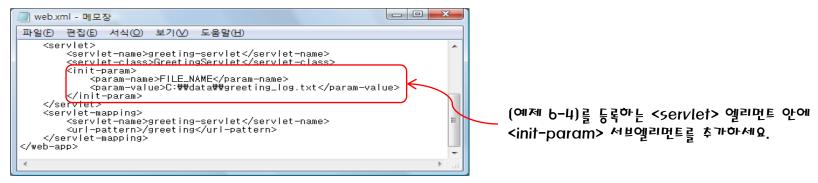
#### ❖ 서블릿의 초기화 파라미터

```
[예제6-4] 인사말을 출력하는 서블릿 클래스 – 서블릿의 초기화 파라미터 사용
import javax.servlet.http.*;
import javax.servlet.*;
import java.io.*;
import java.util.*;
public class GreetingServlet extends HttpServlet {
  private PrintWriter logFile;
  public void init() throws ServletException {
    String filename = getInitParameter( "FILE NAME ");
    try {
      logFile = new PrintWriter(new FileWriter(filename, true));
    catch (IOException e) {
      throw new ServletException(e);
  public void doGet(HttpServletRequest request, HttpServletResponse response)
                    throws IOException, ServletException {
    String name = request.getParameter( "NAME");
    String greeting = "안녕하세요," + name + "님.";
    if (logFile != null) {
      GregorianCalendar now = new GregorianCalendar();
      logFile.printf( "%TF %TT - %s %n ", now, now, name);
    response.setContentType("text/html;charset=euc-kr");
    PrintWriter out = response.getWriter();
    out.println( "<HEAD><TITLE>인사하기</TITLE></HEAD> ");
```

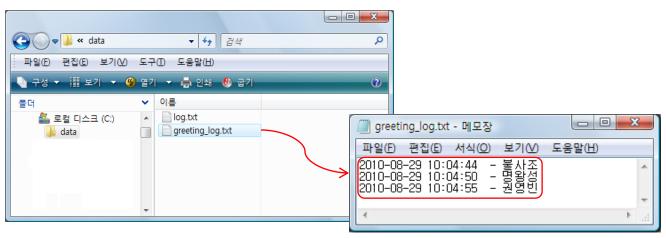
```
out.println( "<BODY> ");
out.println(greeting);
out.println( "</BODY> ");
out.println( "</HTML> ");
}
public void destroy() {
  if (logFile != null)
    logFile.close();
}
```

#### ❖ 서블릿의 초기화 파라미터

■ brain06 웹 애플리케이션 디렉터리의 WEB-INF/web.xml 파일에 서블릿을 등록할 때 다음과 같이 'FILE\_NAME'라는 이름의 초기화 파라미터를 기재한다.

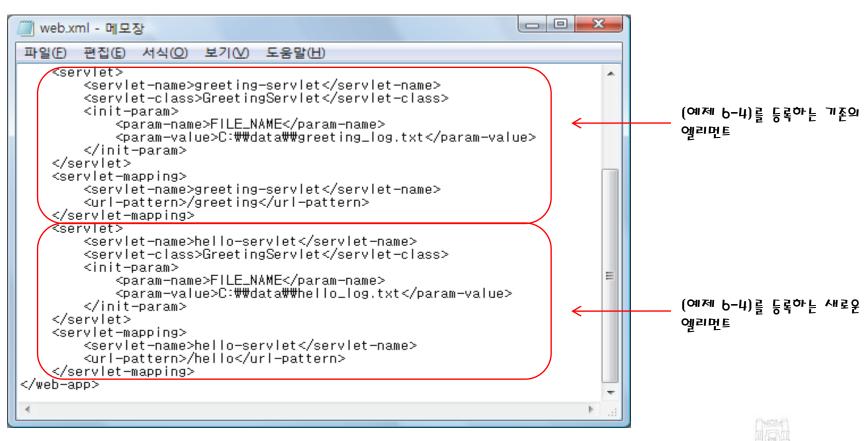


[그림 6-11] 예제 6-4의 등록 방법



[그림 6-12] 예제 6-4가 만든 로그 파일의 내용 확인하기

#### ❖ 서블릿의 초기화 파라미터



[그림 6-13] 하나의 서블릿 클래스를 가지고 두 개의 서블릿을 등록하는 방법



# Thank You!

뇌를 자극하는 JSP & Servlet

