02장. Servlet 기초

강사 강태광

Servlet 개념

1.서블릿은 JSP 표준이 나오기 전에 만들어 진 표준으로 자바에서 웹 애플리케이션을 개발할 수 있도록 하기 위해 만들어 졌으며 자바 클래스를 웹에서 호출 및 실행 할 수 있도록 한 표준
 - javax.servlet.http.HttpServlet 클래스로부터 상속받아서 작성
 - 위의 클래스는 톰캣의 servlet-api.jar 에 포함.
2.작성과정
  1) 서블릿 규칙에 따라 자바 코드 생성
  2) 작성한 코드를 컴파일해서 웹 프로젝트 WEB-INF classes 폴더에 복사
  3)경우에 따라서 web.xml 파일에서 서블릿을 주소와 매핑
  4) 웹 컨테이너 재실행
3. 서블릿 요청 처리
 -요청 방식에 따라 doGet이나 doPost 메서드를 재정의해서 처리
 - service 메서드를 재정의해서 사용할 수 있는데 이 메서드는 get 방식이나 post 방식 상관없이 호출되며 이 메서드가 호출되면 doGet이나 doPost 메서드는 호출되지 않습니다
 - 서블릿에서 요청을 처리하기 위해 오버라이딩 한 메서드는 request 객체를 이용해서 웹 브라우저의 요청 정보를 읽어 오던가 아니면 response를 이용해서 응답을 전송.
 - 응답을 전송하고자 하는 경우는 response 객체의 setContentType()메서드를 이용해서 타입과 인코딩 방식을 지정.
 - 웹 브라우저에 데이터를 전송하려면 getWriter()를 호출해서 문자열 데이터를 출력할 수 있는 PrintWriter를 가져오고 print()나 println()을 이용해서 전송하면 됨.

Servlet 선언

1.이클립스 5.0이전 버전에서는 web.xml 파일을 이용해서 매핑을 했지만 지금은 annotation의 등장으로 주소와의 매핑을 코드 안에서 가능한데 클래스 정의 상단에 @WebServlet(주소 또는 urlPatterns="패턴")의 형태로 가능

2.web.xml(웹 프로젝트의 설정 파일)에서의 매핑

```
<servlet>
    <servlet-name>서블릿이름</servlet-name>
    <servlet-class>클래스이름</servlet-class>
</servlet>
<servlet-mapping>
      <servlet-name>서블릿이름</servlet-name>
      <url-pattern>/호출url</url-pattern>
</servlet-mapping>
```

3. <servlet-mapping> 엘리먼트 안에는 전체 URL이 아니라, 웹 서버의 도메인 이름, 포트 번호, 웹 어플리케이션 디렉터리 이름을 제외한 나머지 부분만 기재해야 합니다

/* -> 무조건 수행
/aaa/* -> aaa 이면 무조건 수행
*.jsp -> jsp 확장자인 경우 무조건 수행

Servlet 선언

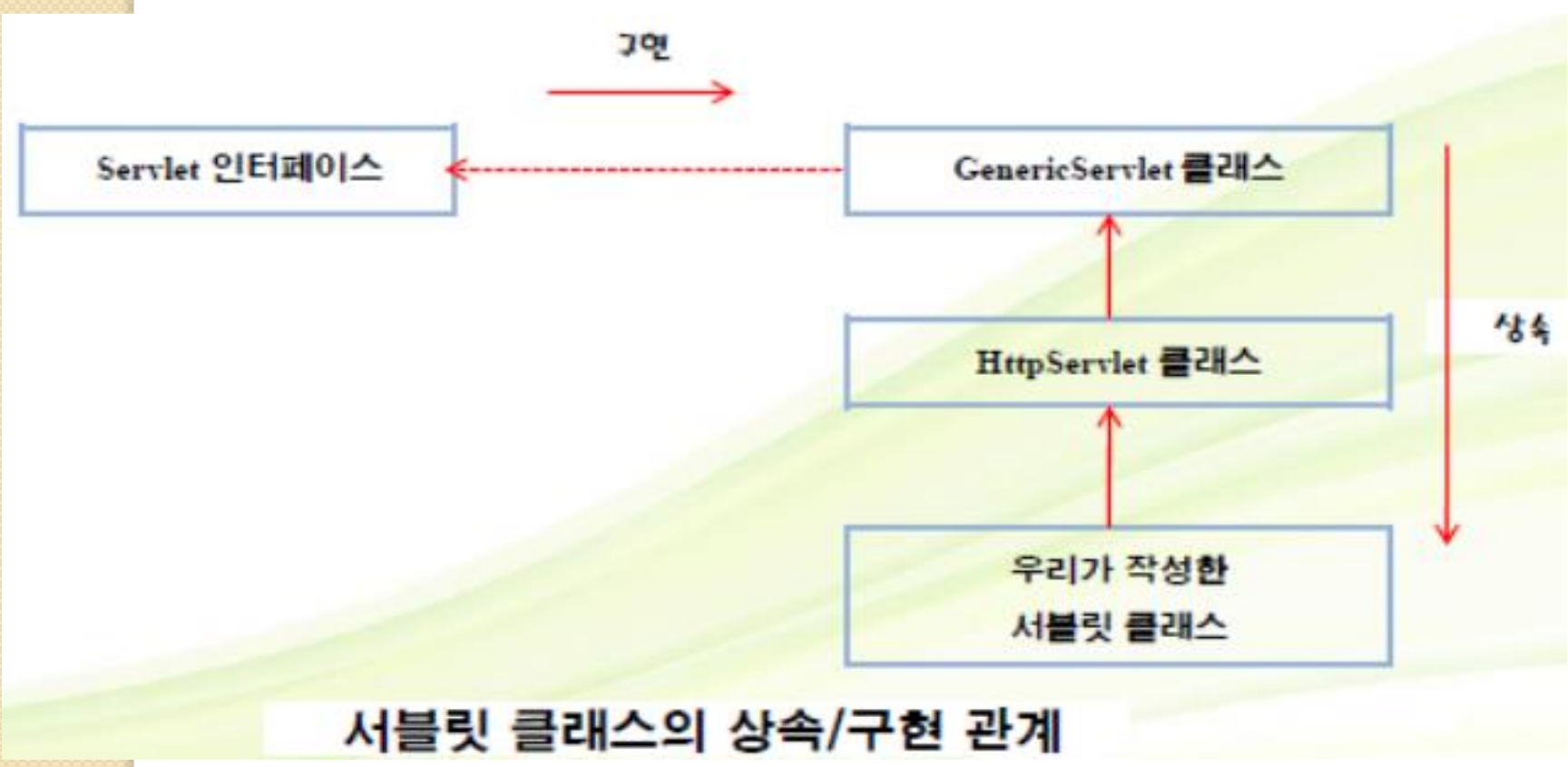
1. 매개변수(parameter- 폼 안에서 name을 가진 객체, 또는 URL의 ? 다음의 데이터) 처리 방법
   1) 단일 항목인 경우: request 객체의 getParameter("매개변수이름")으로 처리
   2) 배열인 경우: request 객체의 getParameterValues("매개변수이름")으로 처리
2. 요청 시 응답
   1)요청 시 응답은 response 객체를 이용
   2)출력을 하고자 하는 경우라면 response객체의 getWriter()를 호출해서 PriterWriter 객체를 받은 후 print 또는 println 메서드를 이용해서 html 코드를 리턴(print는 이어지는 문장을 계속 작성하고자 하는 경우 사용하고 println은 하나의 문장을 작성하기 위해서 사용)
3. 특정 페이지로 이동
   1) request 공유 RequestDispatcher 변수 =
      request.getRequestDispatcher("이동할 페이지");
      변수.forward(request, response);
   2) request 공유하지 않고 이동
      response.sendRedirect("이동할 페이지")

서블릿 클래스의 작성, 컴파일, 설치, 등록

★ 서블릿 클래스의 작성을 위한 준비

1] 서블릿 클래스를 작성할 때 지켜야 할 규칙 세 가지

- 서블릿 클래스는 javax.servlet.http.HttpServlet 클래스를 상속하도록 만들어야 한다
- doGet 또는 doPost 메서드 안에 웹 브라우저로부터 요청이 왔을 때 해야 할 일을 기술
- HTML 문서는 doGet, doPost 메서드의 두 번째 파라미터를 이용해서 출력



서블릿 클래스의 상속/구현 관계

서블릿 호출 방법

1.Get 방식
– 주소에 매개변수를 붙여서 호출하는 방식
- 주소와 매개변수를 붙여서 주소 표시줄에 입력하는 방법(?로 구분)
- a 태그를 이용해서 페이지를 요청하는 경우
- 자바 스크립트를 이용해서 요청하는 경우
- 폼에서 명시적으로 GET 방식으로 요청하는 경우
- 매개변수의 데이터는 4K 이내이며 보안성이 없음
- 폼에서 사용하면 처리가 지연되는 경우 재요청

2. Post 방식
– 매개변수를 본문에 포함시켜 전송하는 방식
- 폼에서 명시적으로 POST 방식으로 요청
- 데이터의 크기에 제한이 없으며 URL에 표시가 되지 않으므로 보안성이 우수

서블릿 클래스의 작성, 컴파일, 설치, 등록

HttpServlet 상속 서블릿 예시

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class HelloServlet extends HttpServlet {
   public void doGet(HttpServletRequest req, HttpServletResponse res)
    throws ServletException, IOException{
     res.setContentType("text/html;charset=euc-kr");
     try {
        PrintWriter out = res.getWriter();
        out.println("<HTML>");
        out.println("<HEAD><TITLE> Hello Servlet </Title></Head>");
        out.println("<BODY> Hello Servlet </BODY>");
        out.println("</HTML>");
        out.close();
     }catch(Exception e){
        getServletContext().log("Error in HelloServlet:",e);
     }
   }
}
```

서블릿 클래스의 작성, 컴파일, 설치, 등록

★ @WebServlet annotation 서블릿 예시

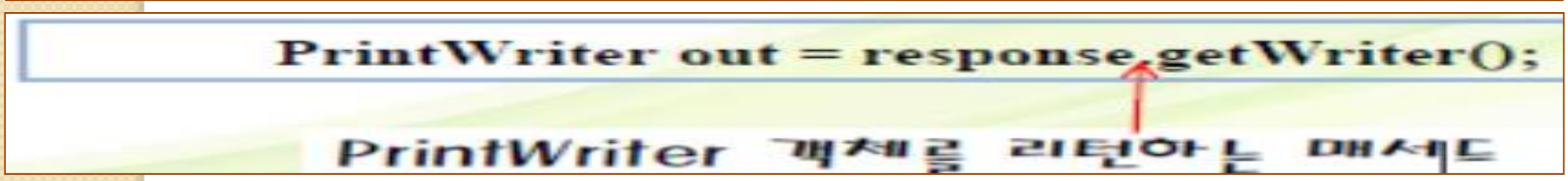
```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/AdderServlet") // servlet와 serlet-mapping 대체
public class AdderServlet extends HttpServlet {
   private static final long serialVersionUID = 1L;
   protected void doGet(HttpServletRequest request, HttpServletResponse response)
       throws ServletException, IOException {
         String str1 = request.getParameter("num1");
         String str2 = request.getParameter("num2");
         int num1 = Integer.parseInt(str1);
         int num2 = Integer.parseInt(str2);
         int total = num1 + num2;
        response.setContentType("text/html;charset=euc-kr");
        PrintWriter out = response.getWriter();
        out.println("<html><body> <h1>두수의 합</h1>");
        out.println(num1 + " + " + num2 + " = " + total);
        out.println("</body></html>"); out.close();
   }
}
```

서블릿 클래스의 작성, 컴파일, 설치, 등록

★ doGet 메서드의 골격을 만든 다음에는 안에 내용을 채워 넣는다

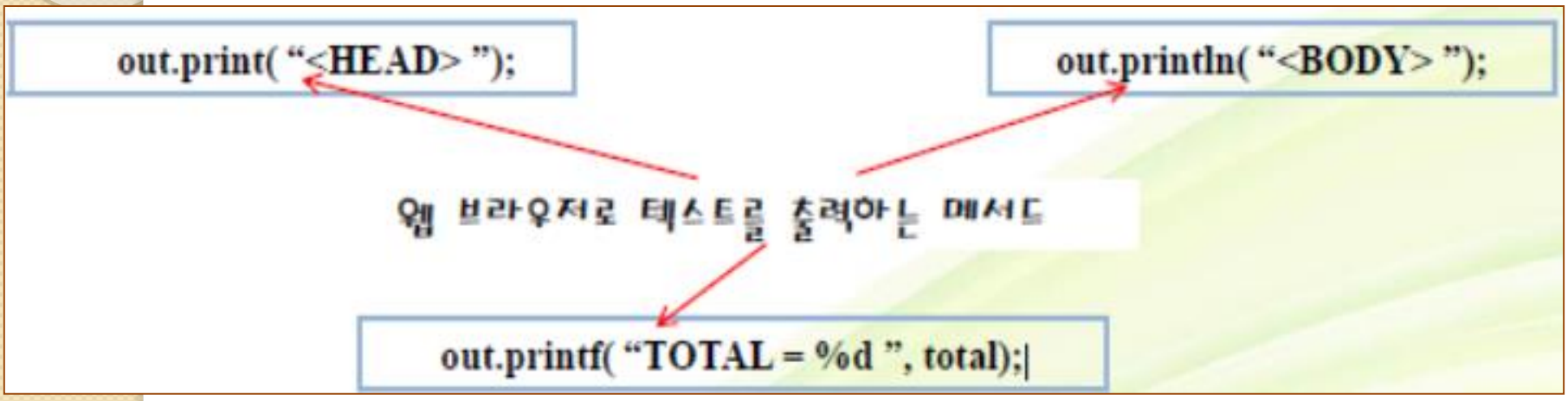
```
public class HundredServlet extends HttpServlet {
  public void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
      int total = 0;
    for (int cnt = 1; cnt < 101; cnt++)
      total += cnt;
   }
}
```

① 실행 결과를 출력하는 코드는 doGet 메서드의 두 번째 파라미터를 이용해서 작성
② 두 번째 파라미터는 javax.serlvet.http.HttpServletResponse 인터페이스 타입이며, 여기에 getWriter라는 메서드를 호출해서 PrintWriter 객체를 구한다



서블릿 클래스의 작성, 컴파일, 설치, 등록

1. PrintWriter는 본래 자바 프로그램에서 파일로 텍스트를 출력할 때 사용하는 java.io 패키지의 PrintWriter 클래스이다.
2. Response.getWriter메서드가 리턴하는 PrintWriter 객체는 파일이 아니라 웹 브라우저로 데이터를 출력한다.



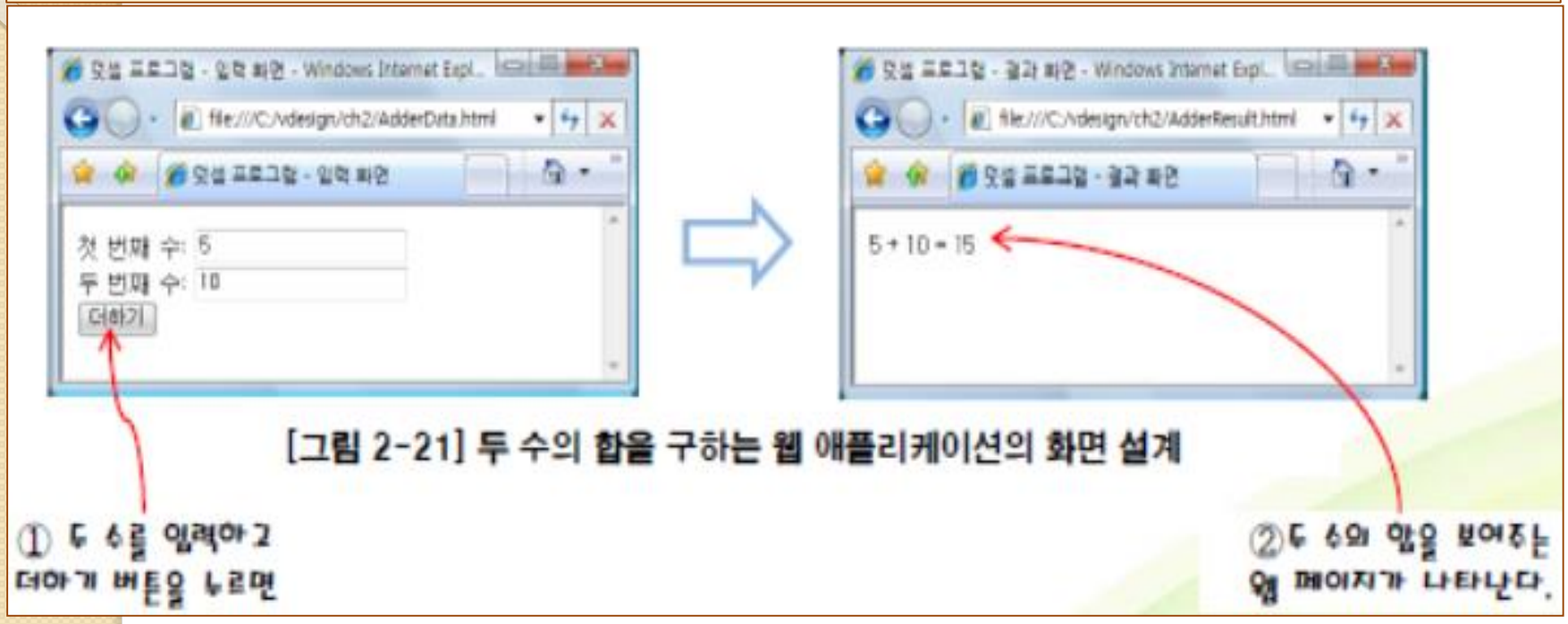
한글 HTML 문서를 출력하는 서블릿 클래스

- 한글이 포함된 HTML 문서를 출력하려면 doGet, doPost 메서드의 두번째 파라미터인 HttpServletResponse 타입의 파라미터에 대해 다음과 같은 메서드를 호출

이 명령문은 HTML을 출력하는 print, println, printf 메서드 호출문보다 앞에 와야 할 뿐만 아니라, response.getWriter 메서드 호출문보다도 먼저 와야 함

웹 브라우저로부터 데이터 입력받기

1. 웹 브라우저로부터 데이터를 입력받는 서블릿 클래스
   - 왼쪽 웹 페이지를 통해 두 수를 입력받은 후 그 둘을 합한 결과를 오른쪽 웹 페이지를 통해 보여주는 웹 애플리케이션이다



[그림 2-21] 두 수의 합을 구하는 웹 애플리케이션의 화면 설계

이상의 웹 페이지로 구성되는 웹 애플리케이션을 개발할 때는 먼저 화면 설계를 하고 다음에 각 화면의 **URL을 정하고, 코딩 작업에 들어가는 것이 좋다.**