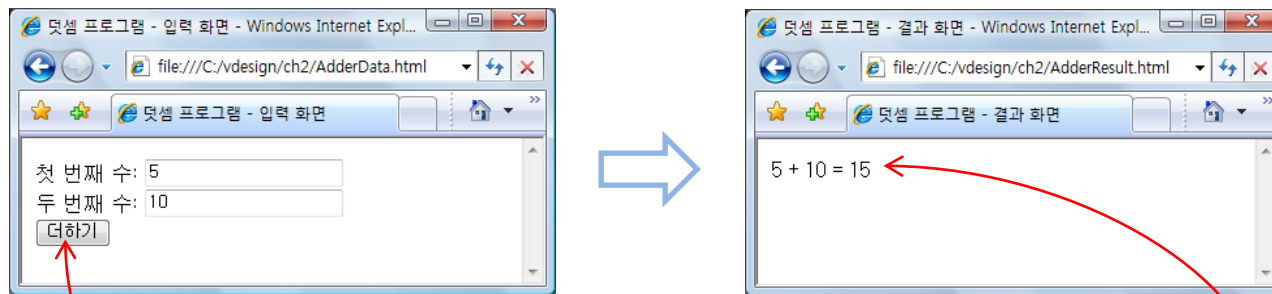


4. 웹 브라우저로부터 데이터 입력받기

❖ 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

① 두 수를 입력하고
더하기 버튼을 누르면

② 두 수의 합을 보여주는
웹 페이지가 나타난다.

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



4. 웹 브라우저로부터 데이터 입력받기

❖ 웹 브라우저로부터 데이터를 입력받는 서블릿 클래스

- 각 화면의 URL은 다음과 같이 정한다.

`http://localhost:8080/brain/AdderInput.html`

← (그림 2-21) 왼쪽 화면 URL

`http://localhost:8080/brain/adder`

← (그림 2-21) 오른쪽 화면 URL

- 왼쪽 화면은 <FORM> 엘리먼트를 사용해서 구현한다.

[예제2-3] 두 개의 수를 입력받는 HTML 문서

```
<HTML>
  <HEAD>
    <META http-equiv= "Content-Type" content= "text/html; charset=euc-kr" >
    <TITLE>덧셈 프로그램 - 입력 화면</TITLE>
  </HEAD>
  <BODY>
    <FORM ACTION=/brain/adder>
      첫 번째 수: <INPUT TYPE=TEXT NAME=NUM1><BR>
      두 번째 수: <INPUT TYPE=TEXT NAME=NUM2><BR>
      <INPUT TYPE=SUBMIT VALUE= '더하기' >
    </FORM>
  </BODY>
</HTML>
```

위 문서를 AdderInput.html이라는 이름으로 brain 웹 애플리케이션 디렉터리에 저장한다.



4. 웹 브라우저로부터 데이터 입력받기

❖ 웹 브라우저로부터 데이터를 입력받는 서블릿 클래스

- 오른쪽 화면을 구현하는 서블릿 클래스는 [예제 2-3]을 통해 입력된 두 수를 받아서 합을 계산한 후 HTML 문서로 만들어서 출력해야 한다.
- <FORM> 엘리먼트를 통해 입력된 데이터는 doGet, doPost 메서드의 첫 번째 파라미터인 HttpServletRequest 타입의 파라미터에 대해 getParameter 메서드를 호출해서 가져올 수 있다.
 - 각 <INPUT> 서브엘리먼트를 통해 입력된 데이터를 가져오기 위해서는 다음과 같은 메서드를 호출해야 한다.

```
String str = request.getParameter( "NUM1 ");
```

<INPUT> 엘리먼트의 NAME 애트리뷰트 값

- 이 메서드가 리턴하는 값은 수치 타입이 아니라 문자열 타입이다.



4. 웹 브라우저로부터 데이터 입력받기

❖ 웹 브라우저로부터 데이터를 입력받는 서블릿 클래스

- 덧셈을 하기 위해서는 문자열 데이터를 수치 타입으로 변환해야 한다.
- 문자열을 int 타입으로 변환하기 위해서는 Integer 클래스의 parseInt 메서드를, double 타입으로 변환하기 위해서는 Double 클래스의 parseDouble 메서드를 이용하면 된다.

```
int num = Integer.parseInt(str);
```

String 타입의 데이터를
int 타입으로 변환하는 메서드



4. 웹 브라우저로부터 데이터 입력받기

❖ 웹 브라우저로부터 데이터를 입력받는 서블릿 클래스

[예제2-4] 두 수의 합을 구하는 서블릿 클래스

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

public class AdderServlet extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException {
        String str1 = request.getParameter( "NUM1 " );
        String str2 = request.getParameter( "NUM2 " );
        int num1 = Integer.parseInt(str1);
        int num2 = Integer.parseInt(str2);
        int sum = num1 + num2;
        response.setContentType( "text/html;charset=euc-kr " );
        PrintWriter out = response.getWriter();
        out.println( "<HTML> " );
        out.println( "<HEAD><TITLE>덧셈 프로그램 - 결과 화면</TITLE></HEAD> " );
        out.println( "<BODY> " );
        out.printf( "%d + %d = %d ", num1, num2, sum );
        out.println( "</BODY> " );
        out.println( "</HTML> " );
    }
}
```



4. 웹 브라우저로부터 데이터 입력받기

❖ 웹 브라우저로부터 데이터를 입력받는 서블릿 클래스

- 서블릿 클래스 컴파일 후 그 결과를 brain 웹 애플리케이션 디렉터리의 WEB-INF\classes 서브디렉터리에 저장한다.
- WEB-INF 디렉터리에 있는 web.xml 파일을 열어서 다음과 같이 서블릿 클래스를 등록한다.



(예제2-4) 서블릿 클래스를
등록하는 코드

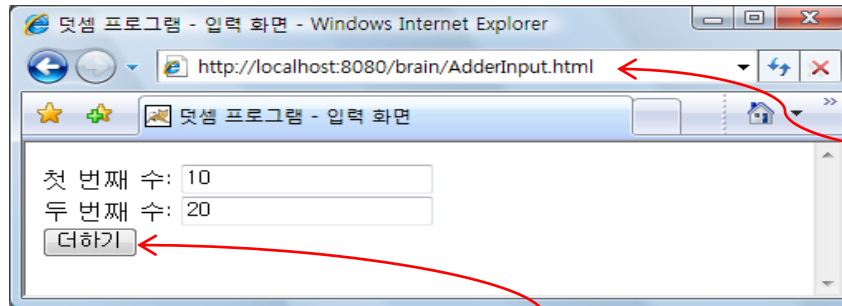
[그림 2-22] 예제 2-4의 서블릿 클래스를 등록하는 방법



4. 웹 브라우저로부터 데이터 입력받기

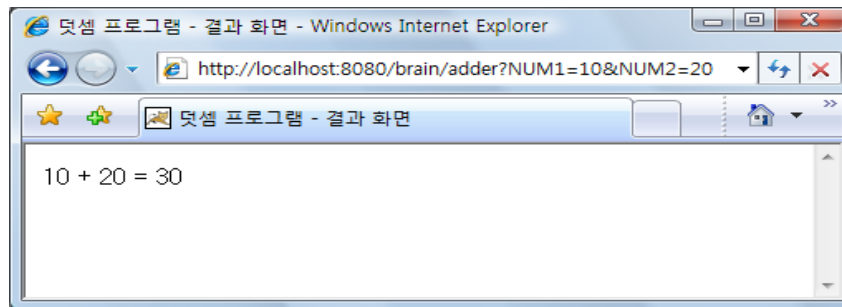
❖ 웹 브라우저로부터 데이터를 입력받는 서블릿 클래스

- 두 수의 합을 구하는 웹 애플리케이션의 실행 방법은 다음과 같다.



① (예제 2-3)의 URL을 입력한다.

② 두 수를 입력하고 더하기 버튼을 누르면 결과 화면이 나온다.



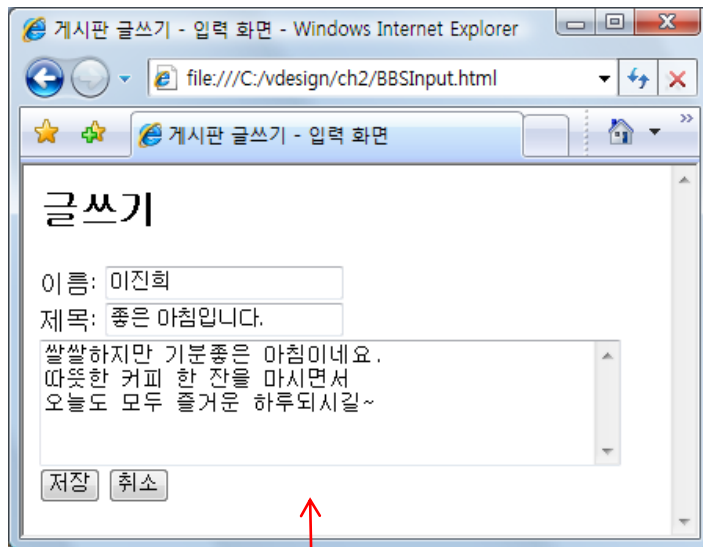
[그림 2-23] 예제 2-3, 예제 2-4의 실행 결과



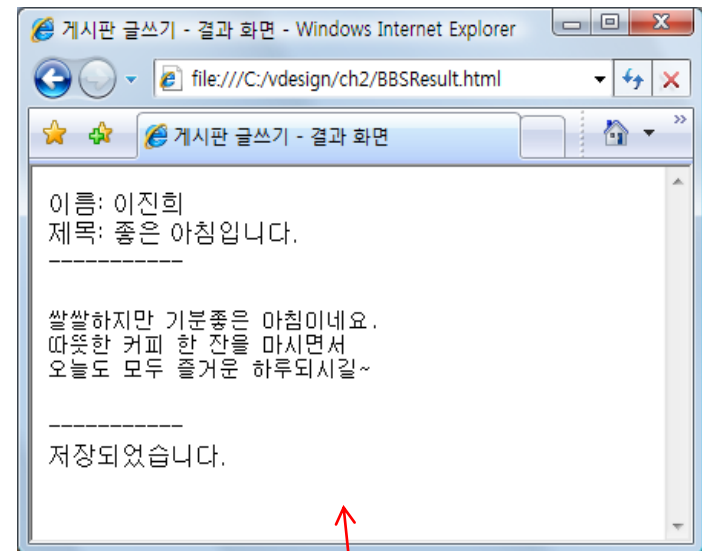
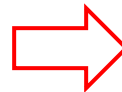
4. 웹 브라우저로부터 데이터 입력받기

❖ POST 메서드를 이용한 데이터 전송

- 웹 페이지를 통해 입력 받은 데이터를 웹 서버 쪽에 저장한 후에 또 다른 웹 페이지를 통해 저장된 결과를 보여주는 웹 애플리케이션이다.



① 이름, 제목, 내용을 입력하고 저장 버튼을 누르면



② 데이터가 웹 서버 쪽에 저장되고 결과 화면이 나타난다.

[그림 2-24] 게시판 글쓰기 애플리케이션 화면 설계



4. 웹 브라우저로부터 데이터 입력받기

❖ POST 메서드를 이용한 데이터 전송

- 입력 데이터가 클 경우에는 URL 뒷부분의 데이터가 잘려나갈 수 있으므로 URL이 아닌 별도의 영역을 통해 입력 데이터를 전송해야 한다.
- <FORM> 엘리먼트의 시작 태그에 METHOD라는 애트리뷰트를 추가하고, 애트리뷰트 값으로 POST를 지정하면 된다.

입력 데이터가 URL이 아닌 별도의 영역을 통해
전송되도록 만드는 METHOD 애트리뷰트 값

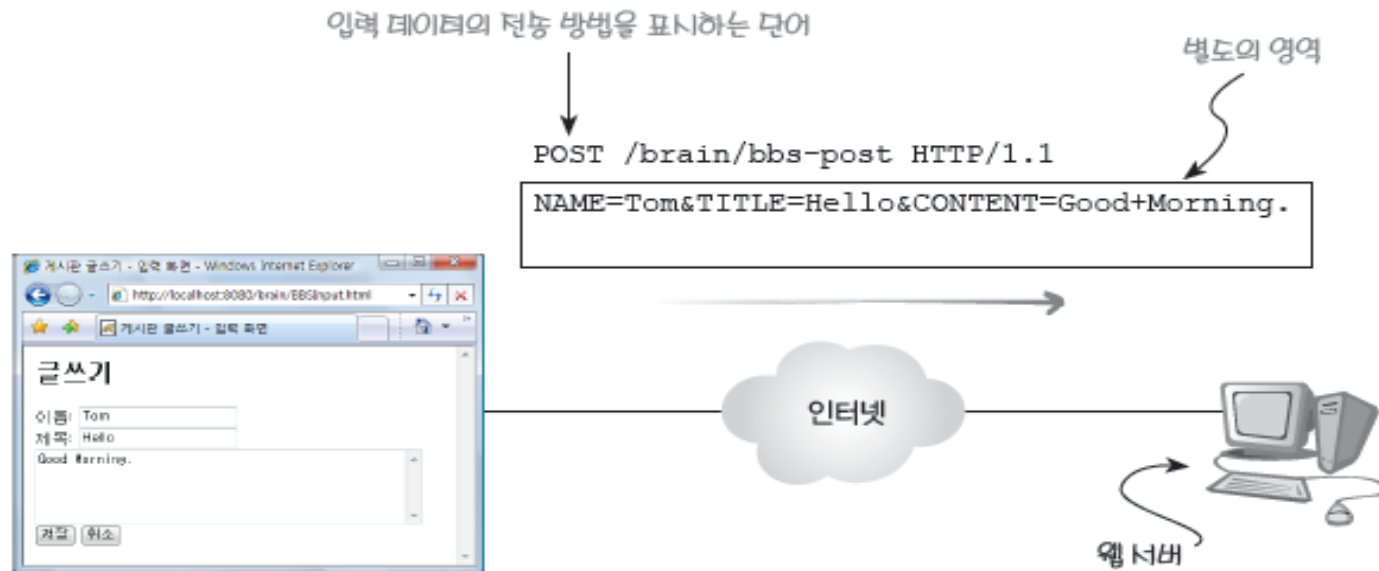
```
<FORM ACTION =/brain/bbs-post METHOD=POST>  
  이름 : <INPUT TYPE=TEXT NAME=WRITER>  
  제목 : <INPUT TYPE=TEXT NAME=TITLE>  
  <TEXTAREA NAME=CONTENT> </TEXTAREA>  
  <INPUT TYPE=SUBMIT VALUE= '저장' >  
  <INPUT TYPE=RESET VALUE= '취소' >  
</FORM>
```



4. 웹 브라우저로부터 데이터 입력받기

❖ POST 메서드를 이용한 데이터 전송

- <FORM> 엘리먼트를 통해 입력된 데이터는 URL 다음에 오는 별도의 영역을 통해 전송되며, URL 앞에는 POST라는 단어가 붙는다. 웹 서버는 이 단어를 보고 입력 데이터가 어디에 있는지 판단할 수 있다.



[그림 2-25] POST 메서드를 이용한 데이터 전송



4. 웹 브라우저로부터 데이터 입력받기

❖ POST 메서드를 이용한 데이터 전송

- URL을 정한 후 URL에 해당하는 HTML 문서를 작성한다.

`http://localhost:8080/brain/BBSInput.html`



(그림 2-24) 왼쪽 화면 URL

`http://localhost:8080/brain/bbs-post`



(그림 2-24) 오른쪽 화면 URL

- 첫 번째 화면을 구현하는 HTML 문서

[예제2-5] 게시판 글쓰기 기능의 데이터 입력을 위한 HTML 문서

```
<HTML>
  <HEAD>
    <META http-equiv= "Content-Type" content= "text/html; charset=euc-kr" >
    <TITLE>게시판 글쓰기 - 입력 화면</TITLE>
  </HEAD>
  <BODY>
    <H2>글쓰기</H2>
    <FORM ACTION=/brain/bbs-post METHOD=POST>
      이름: <INPUT TYPE=TEXT NAME=NAME><BR>
      제목: <INPUT TYPE=TEXT NAME=TITLE><BR>
      <TEXTAREA COLS=50 ROWS=5 NAME=CONTENT></TEXTAREA><BR>
      <INPUT TYPE=SUBMIT VALUE= '저장' >
      <INPUT TYPE=RESET VALUE= '취소' >
    </FORM>
  </BODY>
</HTML>
```

- 이 예제를 톰캣의 webapps\brain 디렉터리에 BBSInput.html라는 이름으로 저장한다.



4. 웹 브라우저로부터 데이터 입력받기

❖ POST 메서드를 이용한 데이터 전송

■ 입력 데이터를 처리하는 서블릿 클래스의 작성 방법

- doGet 메서드를 선언하는 대신 doPost 메서드를 선언해야 한다. 웹 컨테이너는 POST라는 단어가 붙은 URL을 받으면 doGet 메서드가 아니라 doPost 메서드를 호출하기 때문이다
- doPost 메서드는 doGet 메서드와 마찬가지로 public 키워드를 붙여서 선언해야 하고, 파라미터 변수, 리턴 타입, 익셉션 타입도 doGet 메서드와 동일하다.

```
public class BBSPostServlet extends HttpServlet {  
    public void doPost(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException {  
    }  
}
```

doGet 메서드와 리턴 타입, 파라미터 변수,
익셉션 타입이 동일합니다

- doPost 메서드 안에서 입력 데이터를 가져오는 방법과 HTML 문서를 출력하는 방법도 doGet 메서드의 경우와 동일하다.



4. 웹 브라우저로부터 데이터 입력받기

❖ POST 메서드를 이용한 데이터 전송

■ 두 번째 화면을 구현하는 서블릿 클래스

[예제2-6] 게시판 글쓰기 기능을 처리하는 서블릿 클래스 - 미완성

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

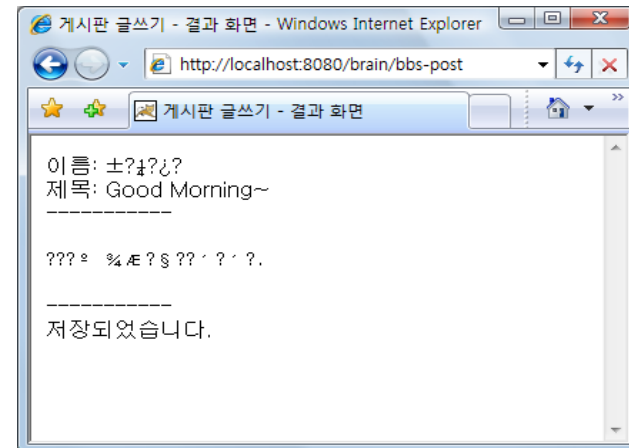
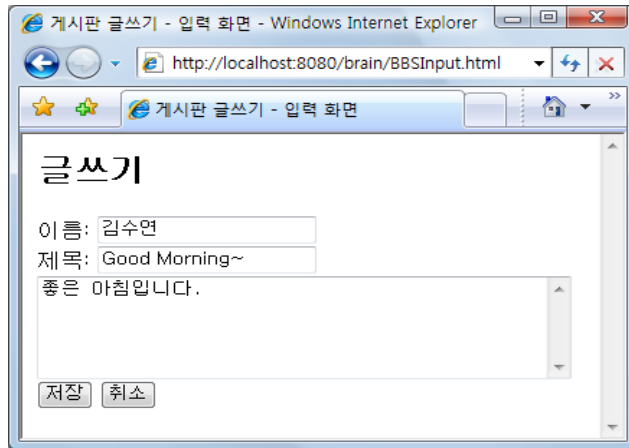
public class BBSPostServlet extends HttpServlet {
    public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException {
        String name = request.getParameter( "NAME " );
        String title = request.getParameter( "TITLE " );
        String content = request.getParameter( "CONTENT " );
        response.setContentType( "text/html;charset=euc-kr " );
        PrintWriter out = response.getWriter();
        out.println( "<HTML> " );
        out.println( "<HEAD><TITLE>게시판 글쓰기 - 결과 화면</TITLE></HEAD> " );
        out.println( "<BODY> " );
        out.printf( "이름: %s <BR> ", name );
        out.printf( "제목: %s <BR> ", title );
        out.println( "-----<BR> " );
        out.printf( "<PRE>%s</PRE> ", content );
        out.println( "-----<BR> " );
        out.println( "저장되었습니다. " );
        out.println( "</BODY> " );
        out.println( "</HTML> " );
    }
}
```



4. 웹 브라우저로부터 데이터 입력받기

❖ POST 메서드를 이용한 데이터 전송

- 위 예제는 다음과 같이 한글 데이터의 입력 처리가 제대로 되지 않는다.



[그림 2-26] 예제 2-5, 예제 2-6의 실행 결과

- 문제 해결: doPost 메서드 안에서 한글 데이터를 올바르게 가져오려면 첫 번째 파라미터 인 HttpServletRequest 파라미터에 대해 setCharacterEncoding 이라는 메서드를 호출해야 한다.

```
request.setCharacterEncoding("euc-kr");
```

↑
한글 코드 이름



4. 웹 브라우저로부터 데이터 입력받기

❖ POST 메서드를 이용한 데이터 전송

- `setCharacterEncoding` 메서드는 `getParameter` 메서드보다 반드시 먼저 호출해야 한다.

[예제2-7] 게시판 글쓰기 기능을 처리하는 서블릿 클래스 - 완성

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

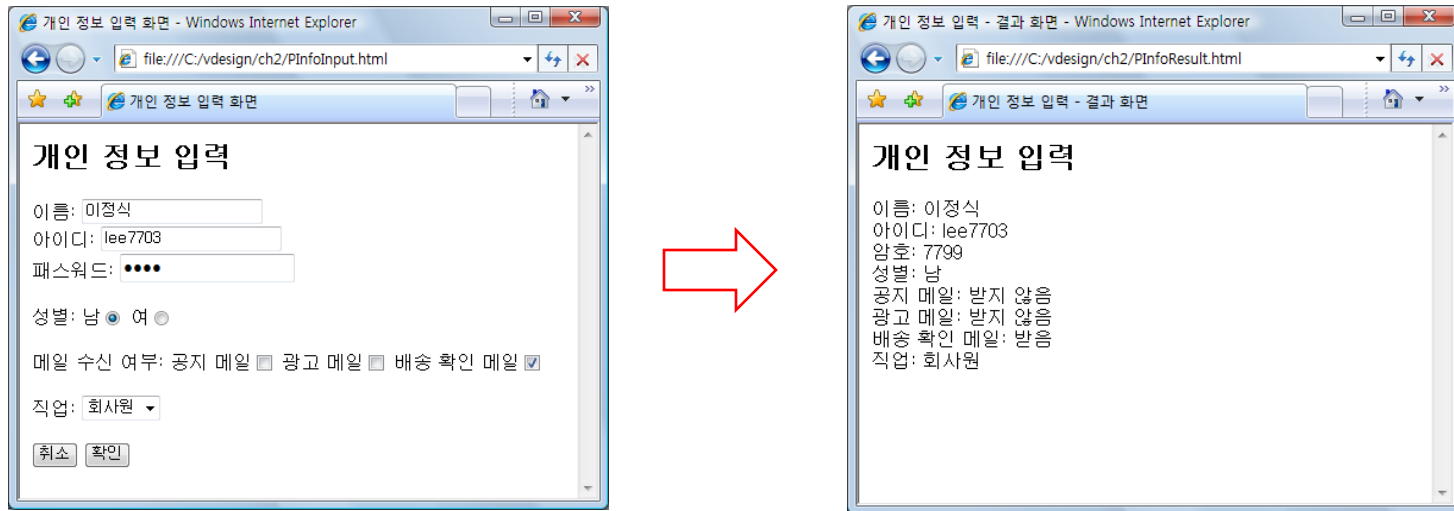
public class BBSPostServlet extends HttpServlet {
    public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException {
        request.setCharacterEncoding( "euc-kr ");
        String name = request.getParameter( "NAME " );
        String title = request.getParameter( "TITLE " );
        String content = request.getParameter( "CONTENT " );
        response.setContentType("text/html;charset=euc-kr ");
        PrintWriter out = response.getWriter();
        out.println( "<HTML> ");
        out.println( "<HEAD><TITLE>게시판 글쓰기 - 결과 화면</TITLE></HEAD> ");
        out.println( "<BODY> ");
        out.printf( "이름: %s <BR> ", name);
        out.printf( "제목: %s <BR> ", title);
        out.println( "-----<BR> ");
        out.printf( "<PRE>%s</PRE> ", content);
        out.println( "-----<BR> ");
        out.println( "저장되었습니다. " );
        out.println( "</BODY> ");
        out.println( "</HTML> ");
    }
}
```



4. 웹 브라우저로부터 데이터 입력받기

❖ 다양한 형태로 데이터 입력받기

- <FORM> 엘리먼트를 이용하면 텍스트 상자 외에도 라디오 버튼, 체크 박스, 선택 상자 등 다양한 형태로 데이터를 입력 받을 수 있다.



[그림 2-29] 개인 정보 입력 애플리케이션의 화면 설계

<http://localhost:8080/brain/PInfoInput.html>

(그림2-29) 왼쪽 화면 URL

<http://localhost:8080/brain/pinfo-result>

(그림2-29) 오른쪽 화면 URL

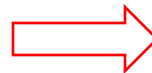


4. 웹 브라우저로부터 데이터 입력받기

❖ 다양한 형태로 데이터 입력받기

- 이름, 아이디 항목은 <INPUT> 엘리먼트의 TYPE 애트리뷰트 값을 TEXT로 지정해서 만들 수 있다. 패스워드는 <INPUT> 엘리먼트의 TYPE 애트리뷰트 값을 PASSWORD로 지정해서 만든다.

```
<INPUT TYPE=TEXT NAME=NAME>  
<INPUT TYPE=TEXT NAME=ID>  
<INPUT TYPE=PASSWORD NAME=PASSWORD>
```

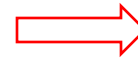


- PASSWORD 타입으로 입력한 문자는 모니터상에 나타나지 않는다.
- 성별 항목은 라디오 버튼으로 만들어야 한다. 라디오 버튼은 <INPUT> 엘리먼트의 TYPE 애트리뷰트 값을 RADIO로 지정해서 만들 수 있으며, 반드시 NAME, VALUE 애트리뷰트를 써야 한다.

```
<INPUT TYPE=RADIO NAME=GENDER VALUE=MALE>  
<INPUT TYPE=RADIO NAME=GENDER VALUE=FEMALE>
```

↑
똑같은 NAME
애트리뷰트 값

↑
각각 다른 VALUE
애트리뷰트 값



↑
한 항목을 선택하면 다른
항목의 선택이 해제된다.



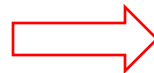
4. 웹 브라우저로부터 데이터 입력받기

❖ 다양한 형태로 데이터 입력받기

- 메일 수신 여부 항목은 체크 박스로 만들어야 한다. 체크 박스는 <INPUT> 엘리먼트의 애트리뷰트 값을 CHECKBOX로 지정해서 만들 수 있으며, NAME 애트리뷰트를 써야 한다. NAME 애트리뷰트에는 각각 다른 값을 지정해야 한다.

```
<INPUT TYPE=CHECKBOX NAME=INOTICE>  
<INPUT TYPE=CHECKBOX NAME=CNOTICE>  
<INPUT TYPE=CHECKBOX NAME=DNOTICE>
```

↑
각각 다른 VALUE
애트리뷰트 값



공지 메일 ☒ 광고 메일 ☐ 배송 확인 메일 ☒

↑
한번 클릭하면 선택되고
또 한번 클릭하면 해제된다.

- 직업 항목은 선택 상자로 만들어야 한다. 선택 상자는 <SELECT> 엘리먼트를 이용해서 만들 수 있고, 이 엘리먼트의 시작 태그와 끝 태그 사이에 선택 항목의 이름을 포함한 <OPTION> 서브엘리먼트들을 써야 한다.

```
<SELECT NAME=JOB>  
  <OPTION>회사원</OPTION>  
  <OPTION>학생</OPTION>  
  <OPTION>기타</OPTION>  
</SELECT>
```



4. 웹 브라우저로부터 데이터 입력받기

❖ 다양한 형태로 데이터 입력받기

[예제2-8] 개인 정보를 입력받는 HTML 문서

```
<HTML>
<HEAD>
  <META http-equiv= "Content-Type " content= "text/html; charset=euc-kr ">
  <TITLE>개인 정보 입력 화면</TITLE>
</HEAD>
<BODY>
  <H2>개인 정보 입력</H2>
  <FORM ACTION=/brain/pinfo-result METHOD=GET>
    이름: <INPUT TYPE=TEXT NAME=NAME><BR>
    아이디: <INPUT TYPE=TEXT NAME=ID><BR>
    비밀번호: <INPUT TYPE=PASSWORD NAME=PASSWORD><BR><BR>
    성별:
      남<INPUT TYPE=RADIO NAME=GENDER VALUE=MALE>
      여<INPUT TYPE=RADIO NAME=GENDER VALUE=FEMALE><BR><BR>
    메일 수신 여부:
      공지 메일<INPUT TYPE=CHECKBOX NAME=INOTICE>
      광고 메일<INPUT TYPE=CHECKBOX NAME=CNOTICE>
      배송 확인 메일<INPUT TYPE=CHECKBOX NAME=DNOTICE><BR><BR>
    직업:
      <SELECT NAME=JOB>
        <OPTION>회사원</OPTION>
        <OPTION>학생</OPTION>
        <OPTION>기타</OPTION>
      </SELECT><BR><BR>
      <INPUT TYPE=RESET VALUE= "취소 " >
      <INPUT TYPE=SUBMIT VALUE= "확인 " >
  </FORM>
</BODY>
</HTML>
```

- 이 예제를 톰캣의 webapps\brain 디렉터리에 PInfoInput.html 라는 이름으로 저장한다.



4. 웹 브라우저로부터 데이터 입력받기

❖ 다양한 형태로 데이터 입력받기

■ 서블릿 클래스의 작성 방법

- <INPUT> 엘리먼트의 TYPE 애트리뷰트 값이 TEXT 또는 PASSWORD일 경우 다음과 같은 방법으로 입력 데이터를 가져올 수 있다.

```
<INPUT TYPE=TEXT NAME=NAME>  
<INPUT TYPE=TEXT NAME=ID>  
<INPUT TYPE=PASSWORD NAME=PASSWORD>
```



```
String name = request.getParameter( "NAME " );  
String id = request.getParameter( "ID " );  
String password = request.getParameter( "PASSWORD " );
```

텍스트 상자에 입력된 값

NAME 애트리뷰트 값



4. 웹 브라우저로부터 데이터 입력받기

❖ 다양한 형태로 데이터 입력받기

- 라디오 버튼의 경우 동일한 NAME 애트리뷰트 값을 갖는 모든 라디오 버튼에 대해 `getParameter` 메서드를 한 번만 호출해야 한다.

```
<INPUT TYPE=RADIO NAME=GENDER VALUE=MALE>  
<INPUT TYPE=RADIO NAME=GENDER VALUE=FEMAIL>
```

```
String gender = request.getParameter( "GENDER " );
```

선택된 항목의
VALUE 애트리뷰트 값

NAME 애트리뷰트 값

- 체크 박스의 경우 각각의 체크 박스에 대해 `getParameter` 메서드를 한 번씩 호출해야 한다.

```
<INPUT TYPE=CHECKBOX NAME=INOTICE>  
<INPUT TYPE=CHECKBOX NAME=CNOTICE>  
<INPUT TYPE=CHECKBOX NAME=DNOTICE>
```

```
String iNotice = request.getParameter( "INOTICE " );  
String cNotice = request.getParameter( "CNOTICE " );  
String dNotice = request.getParameter( "DNOTICE " );
```

'on' 또는 null

NAME 애트리뷰트 값



4. 웹 브라우저로부터 데이터 입력받기

❖ 다양한 형태로 데이터 입력받기

- 선택 상자의 경우 <SELECT> 엘리먼트의 NAME 애트리뷰트 값을 getParameter 메서드에 넘겨줘야 한다.

```
<SELECT NAME=JOB>  
  <OPTION>회사원</OPTION>  
  <OPTION>학생</OPTION>  
  <OPTION>기타</OPTION>  
</SELECT>
```



```
String job = request.getParameter( "JOB" );
```

선택된 항목의
<OPTION> 엘리먼트 내용

NAME 애트리뷰트 값



4. 웹 브라우저로부터 데이터 입력받기

❖ 다양한 형태로 데이터 입력받기

- [그림 2-29]의 오른쪽 화면을 구현하는 서블릿 클래스

[예제2-8] 개인 정보를 입력받는 HTML 문서

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

public class PersonalInfoServlet extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException {
        String name = request.getParameter( "NAME " );
        String id = request.getParameter( "ID " );
        String password = request.getParameter( "PASSWORD " );
        String gender = request.getParameter( "GENDER " );
        if (gender.equals( "MALE " ))
            gender = "남 " ;
        else
            gender = "여 " ;
        String iNotice = request.getParameter( "INOTICE " );
        String cINotice = request.getParameter( "CNOTICE " );
        String dNotice = request.getParameter( "DNOTICE " );
        String job = request.getParameter( "JOB " );
        response.setContentType( "text/html;charset=euc-kr " );
        PrintWriter out = response.getWriter();
        out.println( "<HTML> " );
        out.println("<HEAD><TITLE>개인 정보 입력 - 결과 화면</TITLE></HEAD> ");
```

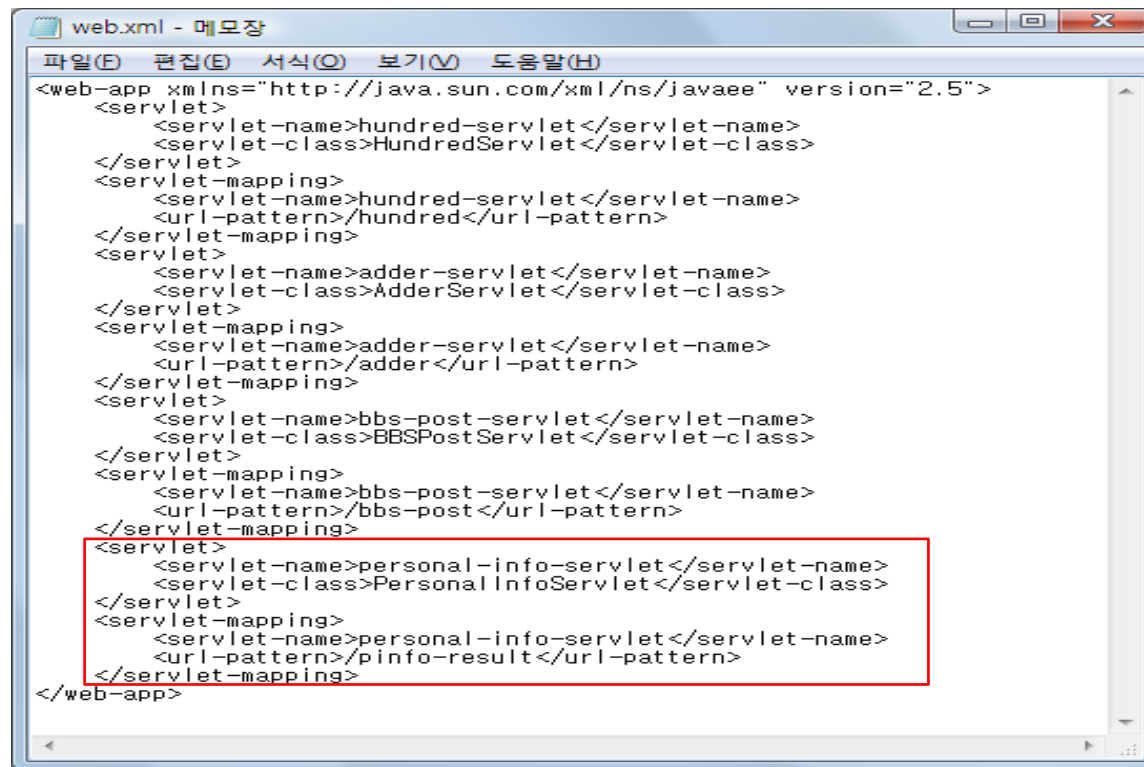
```
        out.println( "<BODY> " );
        out.println( "<H2>개인 정보 입력</H2> " );
        out.printf( "이름: %s <BR> ", name );
        out.printf( "아이디: %s <BR> ", id );
        out.printf( "암호: %s <BR> ", password );
        out.printf( "성별: %s <BR> ", gender );
        out.printf( "공지 메일: %s <BR> ", noticeToHangul(iNotice));
        out.printf( "광고 메일: %s <BR> ", noticeToHangul(cINotice));
        out.printf( "배송 확인 메일: %s <BR> ", noticeToHangul(dNotice));
        out.printf( "직업: %s <BR> ", job );
        out.println( "</BODY> " );
        out.println( "</HTML> " );
    }
    private String noticeToHangul(String notice) {
        if (notice == null)
            return "발지 않음 " ;
        else if (notice.equals( "on " ))
            return "발음 " ;
        else
            return notice;
    }
}
```



4. 웹 브라우저로부터 데이터 입력받기

❖ 다양한 형태로 데이터 입력받기

- 서블릿 클래스를 컴파일해서 결과를 톰캣의 webapps\brain\WEB-INF\classes 디렉터리에 저장한 후, WEB-INF 디렉터리의 web.xml 파일을 열어서 서블릿 클래스를 등록한다.



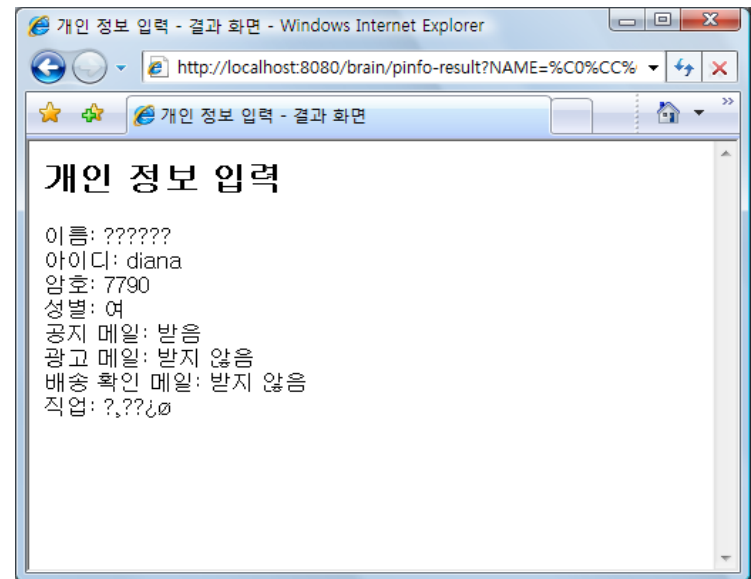
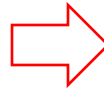
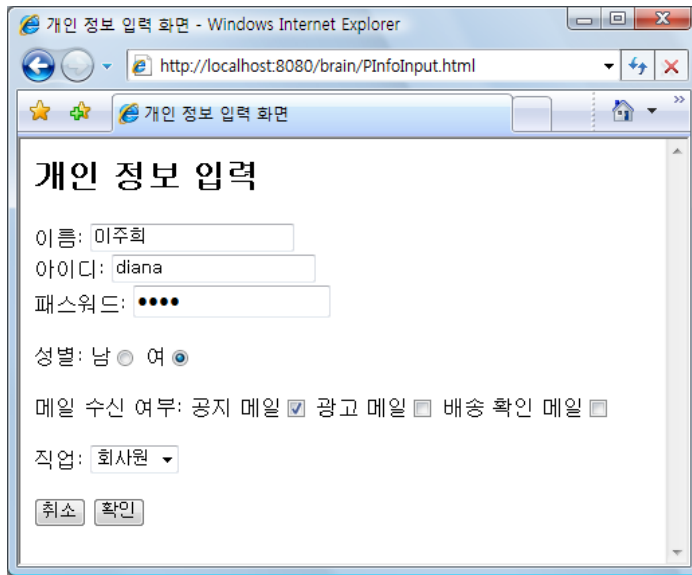
```
<web-app xmlns="http://java.sun.com/xml/ns/javaee" version="2.5">
  <servlet>
    <servlet-name>hundred-servlet</servlet-name>
    <servlet-class>HundredServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>hundred-servlet</servlet-name>
    <url-pattern>/hundred</url-pattern>
  </servlet-mapping>
  <servlet>
    <servlet-name>adder-servlet</servlet-name>
    <servlet-class>AdderServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>adder-servlet</servlet-name>
    <url-pattern>/adder</url-pattern>
  </servlet-mapping>
  <servlet>
    <servlet-name>bbs-post-servlet</servlet-name>
    <servlet-class>BBSPostServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>bbs-post-servlet</servlet-name>
    <url-pattern>/bbs-post</url-pattern>
  </servlet-mapping>
  <servlet>
    <servlet-name>personal-info-servlet</servlet-name>
    <servlet-class>PersonalInfoServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>personal-info-servlet</servlet-name>
    <url-pattern>/pinfo-result</url-pattern>
  </servlet-mapping>
</web-app>
```



4. 웹 브라우저로부터 데이터 입력받기

❖ 다양한 형태로 데이터 입력받기

- 다음 그림과 같이 한글 문제가 생길 경우



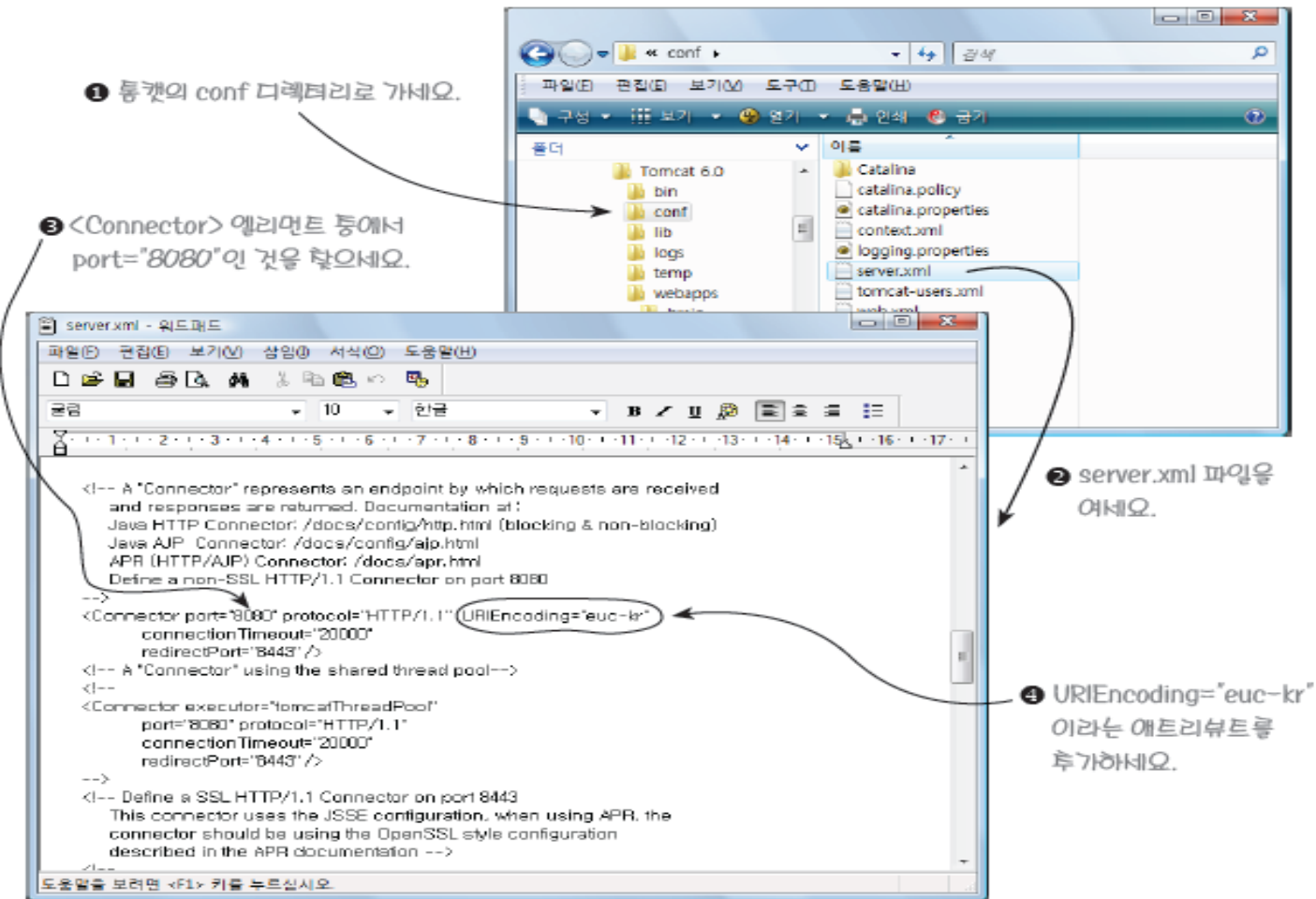
[그림 2-31] 예제 2-8, 2-9의 실행 결과 - 잘못된 결과

- 해결 방법: 톰캣의 conf 디렉터리에 있는 server.xml 파일은 열면 <Connector> 엘리먼트가 있는데, 그 중 port 애트리뷰트 값이 8080인 것을 찾아서 그 엘리먼트에 URLEncoding이라는 이름의 애트리뷰트를 추가하고 애트리뷰트 값으로 “euc-kr”를 지정한다.



4. 웹 브라우저로부터 데이터 입력받기

❖ 다양한 형태로 데이터 입력받기



[그림 2-32] <FORM> 엘리먼트의 GET 메서드로 한글을 입력받기 위해 해야 할 일



4. 웹 브라우저로부터 데이터 입력받기

❖ 다양한 형태로 데이터 입력받기

- server.xml 파일을 수정해서 저장하고, 톱캣을 재기동한 다음 예제를 다시 실행했을 때의 결과

개인 정보 입력 화면 - Windows Internet Explorer

http://localhost:8080/brain/PInfoInput.html

개인 정보 입력 화면

개인 정보 입력

이름: 김현수

아이디: kimhs77

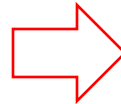
패스워드:

성별: 남 ☐ 여 ☒

메일 수신 여부: 공지 메일 ☒ 광고 메일 ☐ 배송 확인 메일 ☒

직업: 학생

취소 확인



개인 정보 입력 - 결과 화면 - Windows Internet Explorer

http://localhost:8080/brain/pinfo-result?NAME=%B1%E8%

개인 정보 입력 - 결과 화면

개인 정보 입력

이름: 김현수

아이디: kimhs77

암호: 9876

성별: 여

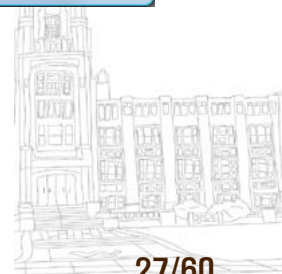
공지 메일: 받음

광고 메일: 받지 않음

배송 확인 메일: 받음

직업: 학생

[그림 2-33] 예제 2-8, 2-9의 실행 결과 - 올바른 결과





Thank You !

뇌를 자극하는 JSP & Servlet