

JSP Object

JSP Object Data

JAVA Class

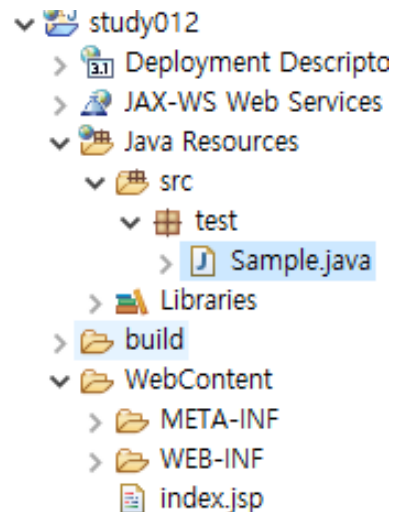
Java Class

JSP를 이용함에 있어서
java의 class를 그대로
사용할 수 있다는 점은
굉장히 큰 장점이다.

외부에 존재하는 class파일을
jsp 페이지 내부에서 이용 가능하다

import하여야 사용할 수 있다.

패키지와 클래스

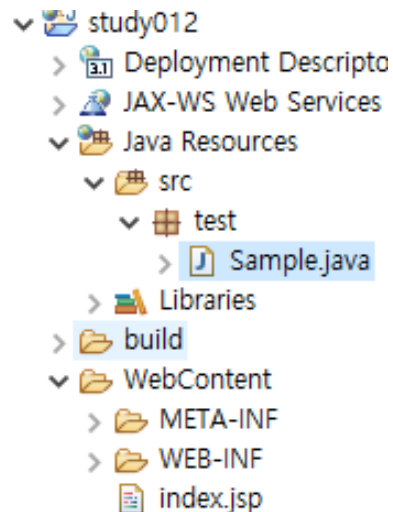


```
Sample.java
1 package test;
2
3 public class Sample {
4     public String name="홍길동";
5     public int age=19;
6     public String address="서울시 강동구";
7 }
-
```

```
index.jsp
1 <%@ page language="java" contentType="text/html; charset=utf-8"
2     pageEncoding="utf-8"%>
3 <%@ page import="test.Sample" %>
4 <!DOCTYPE html>
5 <html>
6     <head>
7         <meta charset="utf-8">
8         <title>자바 클래스의 이동</title>
9     </head>
10    <body>
11        <%
12            Sample s=new Sample();
13        %>
14        <h1>이름 : <%=s.name %></h1>
15        <h1>나이 : <%=s.age %></h1>
16        <h1>주소 : <%=s.address %></h1>
17    </body>
18 </html>
```

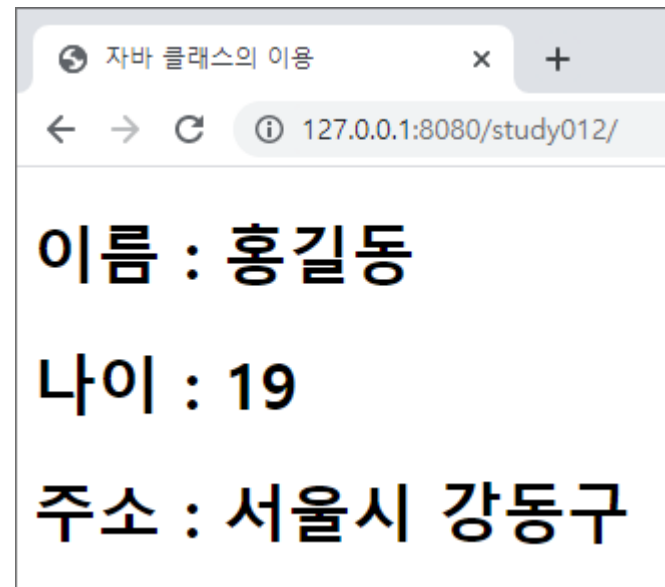
저장 경로가 달리 존재함에 주의
Import=패키지명.클래스명 을 사용한다.

패키지와 클래스



```
Sample.java
1 package test;
2
3 public class Sample {
4     public String name="홍길동";
5     public int age=19;
6     public String address="서울시 강동구";
7 }
8
```

```
index.jsp
1 <%@ page language="java" contentType="text/html; charset=utf-8"
2   pageEncoding="utf-8"%>
3 <%@ page import="test.Sample" %>
4 <!DOCTYPE html>
5 <html>
6   <head>
7     <meta charset="utf-8">
8     <title>자바 클래스의 이용</title>
9   </head>
10  <body>
11    <%
12      Sample s=new Sample();
13    %>
14    <h1>이름 : <%=s.name %></h1>
15    <h1>나이 : <%=s.age %></h1>
16    <h1>주소 : <%=s.address %></h1>
17  </body>
18 </html>
```



저장 경로가 달리 존재함에 주의
Import=패키지명.클래스명 을 사용한다.

메서드의 사용

```
Sample.java
1 package test;
2
3 public class Sample {
4     public String name="홍길동";
5     public int age=19;
6     public String address="서울시 강동구";
7
8     public String makeTable(int a, int b) {
9         String table="<table>";
10
11         for(int i=1; i<=a; i++) {
12             table+="<tr>";
13             for(int j=1; j<=b; j++) {
14                 table+="<td style='border:1px solid black;'>";
15                 table+=i;
16                 table+=", ";
17                 table+=j;
18                 table+="</td>";
19             }
20             table+="</tr>";
21         }
22         table+="</table>";
23         return table;
24     }
25 }
26 }
```

```
index.jsp
1 <%@ page language="java" contentType="text/html; charset=utf-8"
2     pageEncoding="utf-8"%>
3 <%@ page import="test.Sample" %>
4 <!DOCTYPE html>
5 <html>
6 <head>
7     <meta charset="utf-8">
8     <title>자바 클래스의 이용</title>
9 </head>
10 <body>
11 <%
12     Sample s=new Sample();
13 %>
14 <h1>이름 : <%=s.name %></h1>
15 <h1>나이 : <%=s.age %></h1>
16 <h1>주소 : <%=s.address %></h1>
17
18 <%=s.makeTable(3, 5) %>
19 </body>
20 </html>
```

자바 클래스의 이용 x +

127.0.0.1:8080/study012/

이름 : 홍길동

나이 : 19

주소 : 서울시 강동구

1, 1	1, 2	1, 3	1, 4	1, 5
2, 1	2, 2	2, 3	2, 4	2, 5
3, 1	3, 2	3, 3	3, 4	3, 5

class를 통하여 메서드의 사용이 가능하다.

생성자의 활용

```
Sample.java
1 package test;
2
3 public class Sample {
4     public String name="홍길동";
5     public int age=19;
6     public String address="서울시 강동구";
7     public String make;
8
9     public String makeTable(int a, int b) {
10         String table="<table>";
11
12         for(int i=1; i<=a; i++) {
13             table+="<tr>";
14             for(int j=1; j<=b; j++) {
15                 table+="<td style='border:1px solid black;'>";
16                 table+=i;
17                 table+=", ";
18                 table+=j;
19                 table+="</td>";
20             }
21             table+="</tr>";
22         }
23
24         table+="</table>";
25         return table;
26     }
27
28     public Sample(int a, int b) {
29         make=makeTable(a,b);
30     }
31 }
```

```
index.jsp
1 <%@ page language="java" contentType="text/html; charset=utf-8"
2     pageEncoding="utf-8"%>
3 <%@ page import="test.Sample" %>
4 <!DOCTYPE html>
5 <html>
6 <head>
7     <meta charset="utf-8">
8     <title>자바 클래스의 이용</title>
9 </head>
10 <body>
11 <%
12     Sample s=new Sample(4,4);
13 %>
14 <h1>이름 : <%=s.name %></h1>
15 <h1>나이 : <%=s.age %></h1>
16 <h1>주소 : <%=s.address %></h1>
17
18 <%=s.makeTable(3, 5) %>
19 <%=s.make %>
20 </body>
21 </html>
```

자바 클래스의 이용 x +
127.0.0.1:8080/study012/

이름 : 홍길동

나이 : 19

주소 : 서울시 강동구

1, 1	1, 2	1, 3	1, 4	1, 5
2, 1	2, 2	2, 3	2, 4	2, 5
3, 1	3, 2	3, 3	3, 4	3, 5
1, 1	1, 2	1, 3	1, 4	
2, 1	2, 2	2, 3	2, 4	
3, 1	3, 2	3, 3	3, 4	
4, 1	4, 2	4, 3	4, 4	

생성자 역시도 활용이 가능하다.

JAVA Bean

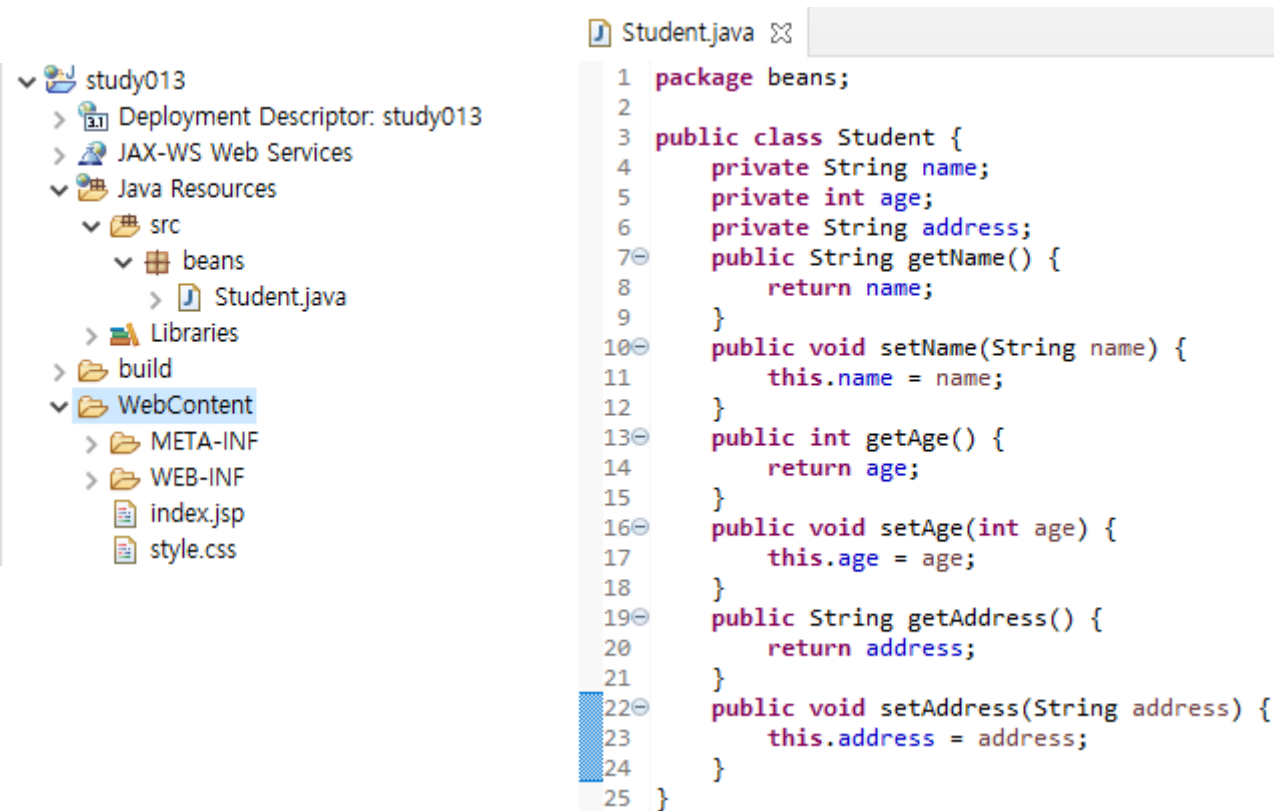
Java Bean

콩처럼 데이터를 하나씩
보관할 수 있는 객체로

페이지 마다 데이터를
자주 전송해야 하는 경우
편리하게 사용할 수 있다.

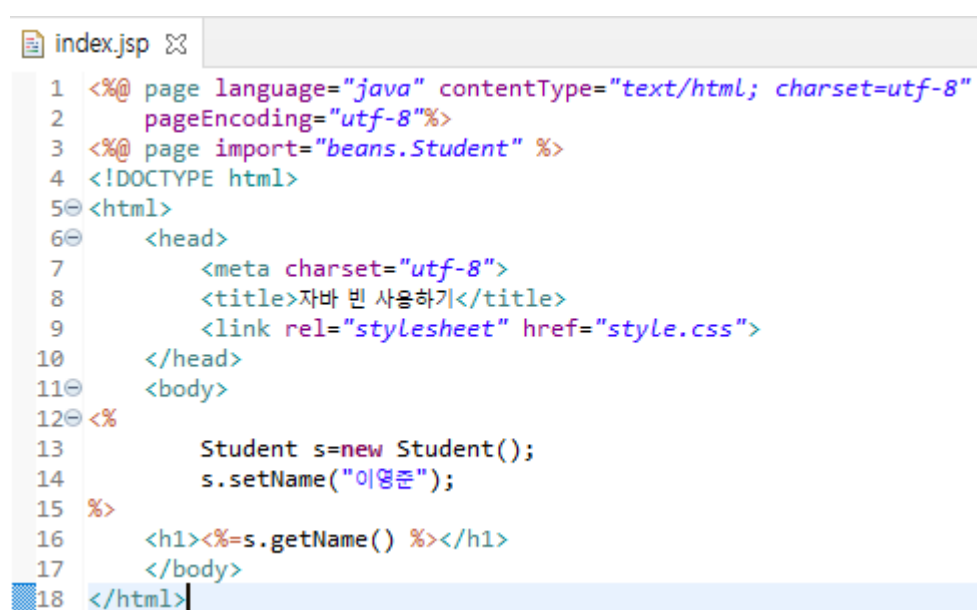
**자바빈은 데이터를 담을 변수와
저장, 추출의 2종의 메서드로 구성된다.**

자바빈 입력



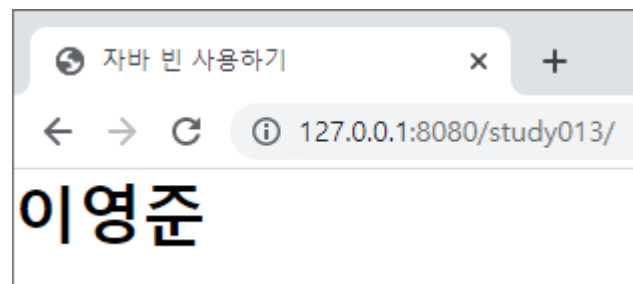
The screenshot shows an IDE with a project named 'study013'. The project structure on the left includes 'Deployment Descriptor: study013', 'JAX-WS Web Services', 'Java Resources', 'src' (containing 'beans' and 'Student.java'), 'Libraries', 'build', and 'WebContent' (containing 'META-INF', 'WEB-INF', 'index.jsp', and 'style.css'). The 'Student.java' file is open in the editor, showing the following code:

```
1 package beans;
2
3 public class Student {
4     private String name;
5     private int age;
6     private String address;
7     public String getName() {
8         return name;
9     }
10    public void setName(String name) {
11        this.name = name;
12    }
13    public int getAge() {
14        return age;
15    }
16    public void setAge(int age) {
17        this.age = age;
18    }
19    public String getAddress() {
20        return address;
21    }
22    public void setAddress(String address) {
23        this.address = address;
24    }
25 }
```



The 'index.jsp' file is open in the editor, showing the following code:

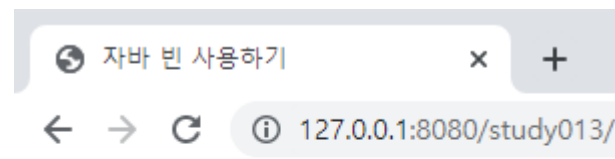
```
1 <%@ page language="java" contentType="text/html; charset=utf-8"
2     pageEncoding="utf-8"%>
3 <%@ page import="beans.Student" %>
4 <!DOCTYPE html>
5 <html>
6     <head>
7         <meta charset="utf-8">
8         <title>자바 빈 사용하기</title>
9         <link rel="stylesheet" href="style.css">
10    </head>
11    <body>
12        <%
13            Student s=new Student();
14            s.setName("이영준");
15        %>
16        <h1><%=s.getName() %></h1>
17    </body>
18 </html>
```



자바빈을 사용하면 Object를 구성할 수 있다.

액션태그를 통한 입출력

```
index.jsp
1 <%@ page language="java" contentType="text/html; charset=utf-8"
2   pageEncoding="utf-8"%>
3 <!DOCTYPE html>
4 <html>
5   <head>
6     <meta charset="utf-8">
7     <title>자바 빈 사용하기</title>
8     <link rel="stylesheet" href="style.css">
9   </head>
10  <body>
11    <jsp:useBean class="beans.Student" id="s" scope="page">
12      <!-- class부분은 import에 적었던 내용이 자리잡고 -->
13      <!-- id부분은 객체의 이름을 적게 되며 -->
14      <!-- scope에서는 해당 객체가 유지될 범위를 적게 된다. -->
15    </jsp:useBean>
16    <%s.setName("홍길동"); %>
17    <h1><%=s.getName() %></h1>
18    <jsp:setProperty name="s" property="age" value="29"></jsp:setProperty>
19    <h1><jsp:getProperty name="s" property="age"></jsp:getProperty></h1>
20  </body>
21 </html>
```

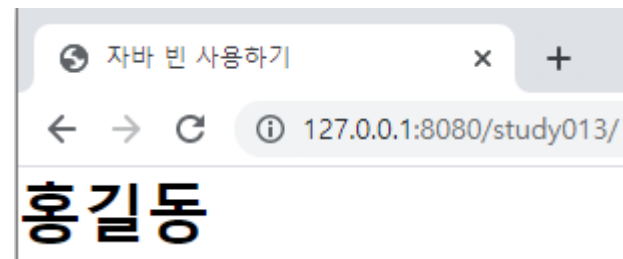


홍길동
29

액션태그를 통하여 import와 instance를 동시 실행

액션태그 사용

```
index.jsp
1 <%@ page language="java" contentType="text/html; charset=utf-8"
2   pageEncoding="utf-8"%>
3 <!DOCTYPE html>
4 <html>
5   <head>
6     <meta charset="utf-8">
7     <title>자바 빈 사용하기</title>
8     <link rel="stylesheet" href="style.css">
9   </head>
10  <body>
11    <jsp:useBean class="beans.Student" id="s" scope="page">
12      <!-- class부분은 import에 적었던 내용이 자리잡고 -->
13      <!-- id부분은 객체의 이름을 적게 되며 -->
14      <!-- scope에서는 해당 객체가 유지될 범위를 적게 된다. -->
15    </jsp:useBean>
16    <%s.setName("홍길동"); %>
17    <h1><%=s.getName() %></h1>
18  </body>
19 </html>
```



스크립트릿을 사용할 때의 번거로움을 해소 가능

Scope

Java Scope

객체가 유지되는 범위

Page : 단일 페이지 내에서만 유지

Request : request 범위에서 유지

Session : 클라이언트 범위에서 유지

Application : 프로그램 전체에서 유지
의 네 종류가 존재한다.

**서비스의 종류에 따라
맞는 형태의 scope를 사용하여야 한다.**

bean과 request

reg.java

```
1 package beans;
2
3 public class reg {
4     private String name;
5     private String userId;
6     private String pass;
7     private String email;
8     private String phone;
9     public String getName() {
10         return name;
11     }
12     public void setName(String name) {
13         this.name = name;
14     }
15     public String getUserId() {
16         return userId;
17     }
18     public void setUserId(String userId) {
19         this.userId = userId;
20     }
21     public String getPass() {
22         return pass;
23     }
24     public void setPass(String pass) {
25         this.pass = pass;
26     }
27     public String getEmail() {
28         return email;
29     }
30     public void setEmail(String email) {
31         this.email = email;
32     }
33     public String getPhone() {
34         return phone;
35     }
36     public void setPhone(String phone) {
37         this.phone = phone;
38     }
39 }
```

index.jsp

```
1 <%@ page language="java" contentType="text/html; charset=utf-8"
2     pageEncoding="utf-8"%>
3 <!DOCTYPE html>
4 <html>
5     <head>
6         <meta charset="utf-8">
7         <title>자바 빈 사용하기</title>
8         <link rel="stylesheet" href="style.css">
9     </head>
10    <body>
11        <div id="wrap">
12            <form method="post" action="register.jsp">
13                이름<input type="text" name="name"><br>
14                아이디<input type="text" name="userId"><br>
15                비밀번호<input type="password" name="pass"><br>
16                이메일<input type="text" name="email"><br>
17                연락처<input type="text" name="first" class="phone">
18                - <input type="text" name="second" class="phone">
19                - <input type="text" name="third" class="phone"><br>
20                <input type="submit" value="입력하기!">
21            </form>
22        </div>
23    </body>
24 </html>
```

style.css

```
1 @charset "utf-8";
2
3 *{
4     margin:0;
5     padding:0;
6 }
7 #wrap{
8     border:10px solid black;
9     width:200px;
10    margin:0 auto;
11    padding:25px;
12 }
13 input{
14     width:50%;
15 }
16 .phone{
17     width:15%;
18 }
```

request를 통한 데이터를 한번에 입력 가능

bean과 request

reg.java

```
1 package beans;
2
3 public class reg {
4     private String name;
5     private String userId;
6     private String pass;
7     private String email;
8     private String phone;
9     public String getName() {
10         return name;
11     }
12     public void setName(String name) {
13         this.name = name;
14     }
15     public String getUserId() {
16         return userId;
17     }
18     public void setUserId(String userId) {
19         this.userId = userId;
20     }
21     public String getPass() {
22         return pass;
23     }
24     public void setPass(String pass) {
25         this.pass = pass;
26     }
27     public String getEmail() {
28         return email;
29     }
30     public void setEmail(String email) {
31         this.email = email;
32     }
33     public String getPhone() {
34         return phone;
35     }
36     public void setPhone(String phone) {
37         this.phone = phone;
38     }
39 }
```

index.jsp

```
1 <%@ page language="java" contentType="text/html; charset=utf-8"
2     pageEncoding="utf-8"%>
3 <!DOCTYPE html>
4 <html>
5     <head>
6         <meta charset="utf-8">
7         <title>자바 빈 사용하기</title>
8         <link rel="stylesheet" href="style.css">
9     </head>
10    <body>
11        <div id="wrap">
12            <form method="post" action="register.jsp">
13                이름<input type="text" name="name"><br>
14                아이디<input type="text" name="userId"><br>
15                비밀번호<input type="password" name="pass"><br>
16                이메일<input type="text" name="email"><br>
17                연락처<input type="text" name="first" class="phone">
18                - <input type="text" name="second" class="phone">
19                - <input type="text" name="third" class="phone"><br>
20                <input type="submit" value="입력하기!">
21            </form>
22        </div>
23    </body>
24 </html>
```

style.css

```
1 @charset "utf-8";
2
3 *{
4     margin:0;
5     padding:0;
6 }
7 #wrap{
8     border:10px solid black;
9     width:200px;
10    margin:0 auto;
11    padding:25px;
12 }
13 input{
14     width:50%;
15 }
16 .phone{
17     width:15%;
18 }
```

request를 통해 데이터를 전송할 수 있다.

일반적인 방식

register.jsp

```
1  <%@ page language="java" contentType="text/html; charset=utf-8"
2      pageEncoding="utf-8"%>
3  <!DOCTYPE html>
4  <html>
5  <head>
6      <meta charset="utf-8">
7      <title>입력 확인페이지</title>
8  </head>
9  <body>
10     <%
11         request.setCharacterEncoding("utf-8");
12         String name=request.getParameter("name");
13         String userId=request.getParameter("userId");
14         String pass=request.getParameter("pass");
15         String email=request.getParameter("email");
16         String phone=request.getParameter("first")+"- "+
17             +request.getParameter("second")+"- "+
18             +request.getParameter("third");
19     %>
20     <h1>전송받은 데이터</h1>
21     <p>이름 : <%=name %></p>
22     <p>아이디 : <%=userId %></p>
23     <p>비밀번호 : <%=pass %></p>
24     <p>이메일 : <%=email %></p>
25     <p>전화번호 : <%=phone %></p>
26 </body>
27 </html>
```

전송받은 데이터

이름 : 이영준

아이디 : 아이디

비밀번호 : qlalfqjsgh

이메일 : 이@메일

전화번호 : 010-123-456

액션태그를 이용하여 내용을 축약 가능하다.

액션태그 방식

register.jsp

```
1 <%@ page language="java" contentType="text/html; charset=utf-8"
2   pageEncoding="utf-8"%>
3 <!DOCTYPE html>
4 <html>
5   <head>
6     <meta charset="utf-8">
7     <title>입력 확인페이지</title>
8   </head>
9   <body>
10    <%
11      request.setCharacterEncoding("utf-8");
12    %>
13    <jsp:useBean id="r" class="beans.reg"></jsp:useBean>
14    <jsp:setProperty name="r" property="*"></jsp:setProperty>
15    <h1>전송받은 데이터</h1>
16    <p>이름 : <%=r.getName() %></p>
17    <p>아이디 : <jsp:getProperty name="r" property="userId"></jsp:getProperty></p>
18    <p>비밀번호 : <jsp:getProperty name="r" property="pass"></jsp:getProperty></p>
19    <p>이메일 : <jsp:getProperty name="r" property="email"></jsp:getProperty></p>
20    <p>전화번호 : <jsp:getProperty name="r" property="phone"></jsp:getProperty></p>
21  </body>
22 </html>
```

입력 확인페이지

127.0.0.1:8080/study011

전송받은 데이터

이름 : 이영준

아이디 : 아이디

비밀번호 : qlalfqjsgh

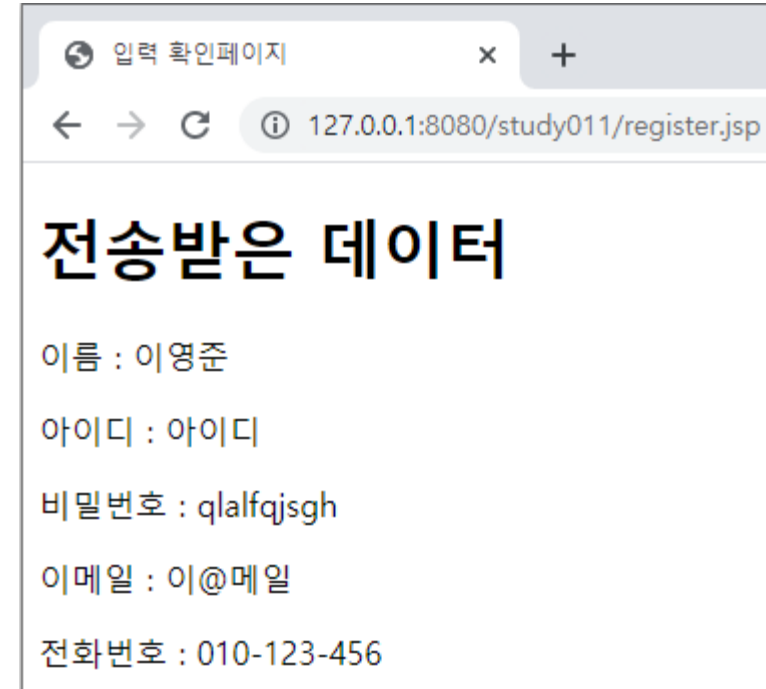
이메일 : 이@메일

전화번호 : null

액션태그도 request를 이용하므로 한글깨짐 주의

다중 입력 혼합

```
private String first;  
private String second;  
private String third;  
public String getFirst() {  
    return first;  
}  
public void setFirst(String first) {  
    this.first = first;  
}  
public String getSecond() {  
    return second;  
}  
public void setSecond(String second) {  
    this.second = second;  
}  
public String getThird() {  
    return third;  
}  
public void setThird(String third) {  
    this.third = third;  
    setPhone(first+"-"+second+"-"+third);  
}
```



A screenshot of a web browser window. The address bar shows the URL '127.0.0.1:8080/study011/register.jsp'. The page title is '입력 확인페이지'. The main content of the page is titled '전송받은 데이터' (Received Data). Below the title, the following information is displayed:

- 이름 : 이영준
- 아이디 : 아이디
- 비밀번호 : qlalfqjsgh
- 이메일 : 이@메일
- 전화번호 : 010-123-456

bean을 추가하여 정보의 가공이 가능하다.