

2. 서블릿의 기초

1. 서블릿이란?
2. 서블릿 프로그램 작성
3. 서블릿 한글 처리와 데이터 통신
4. 기타 다양한 입력 양식

1. 서블릿이란?

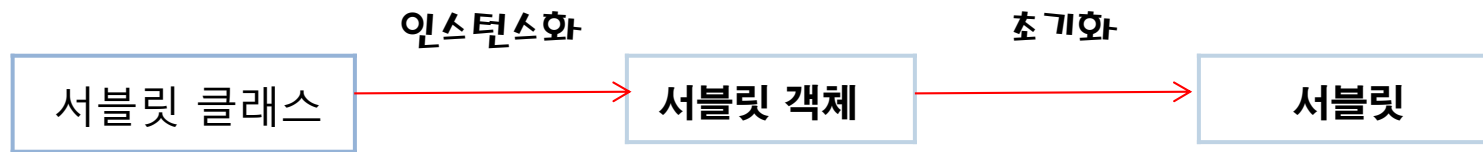
□ Servlet

- ▣ Server + Applet의 합성어
- ▣ 서버에서 실행되는 Applet, 즉 자바를 이용하여 웹에서 실행되는 프로그램을 작성하는 기술
- ▣ Java 클래스 형태의 웹 애플리케이션, 브라우저를 통해 자바 클래스가 실행
- ▣ javax.servlet.http 패키지의 HttpServlet 클래스 상속을 받아 구현

1. 서블릿이란?

□ 서블릿

- 서블릿이란 서블릿 클래스로부터 만들어진 객체이
- 웹 컨테이너는 서블릿 클래스를 가지고 서블릿 객체를 만든 다음 그 객체를 초기화해서 웹 서비스를 할 수 있는 상태로 만든 객체

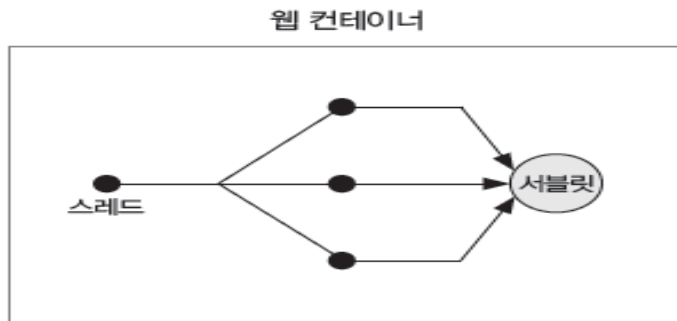


[그림 2-1] 서블릿 클래스, 서블릿 객체, 서블릿

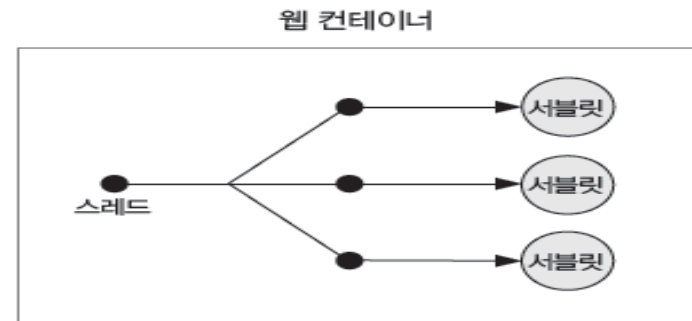
1. 서블릿이란?

□ 멀티스레드 모델

- 장점: 필요한 서블릿의 수가 적기 때문에 서블릿을 만들기 위해 필요한 시스템 자원과 서블릿이 차지하는 메모리를 절약
- 단점: 여러 스레드가 동시에 한 서블릿을 사용하기 때문에 데이터 공유 문제 발생.



(a) 멀티-스레드 모델(Multi-Thread Model)



(b) 싱글-스레드 모델(Single-Thread Model)

[그림 2-2] 멀티-스레드 모델과 싱글-스레드 모델

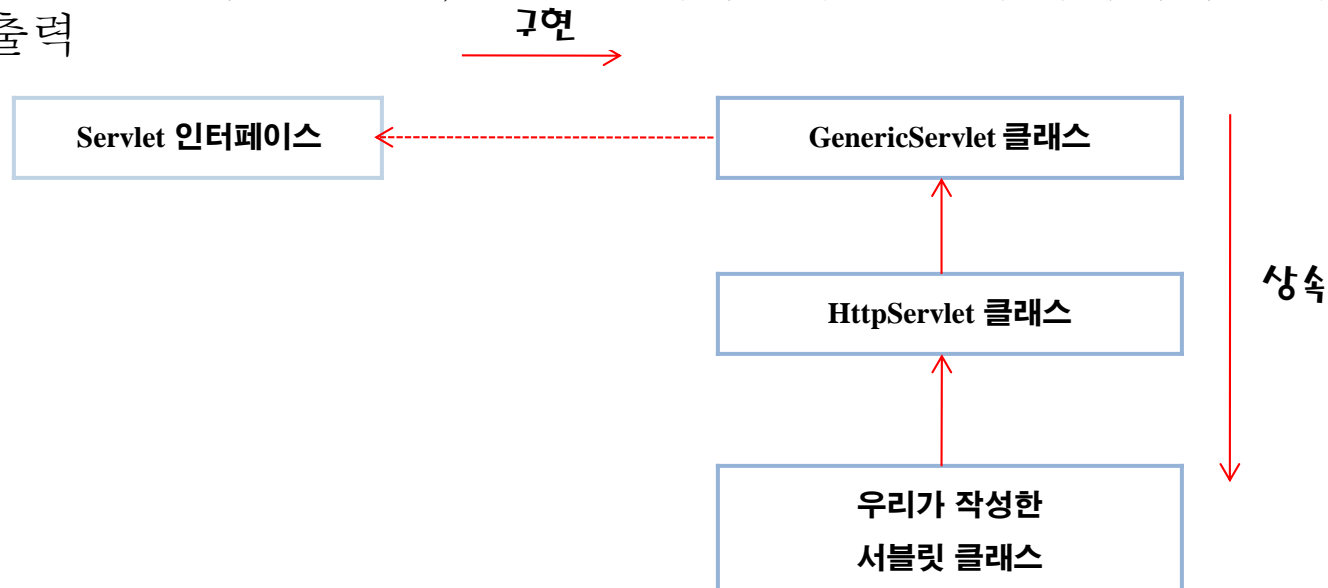
- 싱글-스레드 모델에서는 데이터 공유 문제를 걱정할 필요가 없지만 시스템 자원과 메모리가 더 많이 소모.

2. 서블릿 클래스의 작성

□ 서블릿 클래스의 작성을 위한 준비

■ 서블릿 클래스를 작성할 때 지켜야 할 규칙 세 가지

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



[그림 2-3] 서블릿 클래스의 상속/구현 관계

서블릿 클래스의 작성

□ 서블릿 클래스

```
@WebServlet("/TestServlet")
public class TestServlet extends HttpServlet {

    private static final long serialVersionUID = 1L;
    public TestServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

    protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
    }

    protected void doPost(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
    }

}
```

2. 서블릿 프로그램 작성

□ 예제 : 두 수를 더한 결과 출력 Servlet 작성

- [ch02]패키지를 선택한 후 [New -> Servlet] 선택
- [Create Servlet] 창에 (AdditionServlet)를 입력한 후 [Next] 버튼을 누른다.
- URL Mapping: 서블릿을 동작시키기 위해서 실제 자바 클래스 이름 대신 사용하는 문자열
- URL Mapping 목록에서 항목을 선택 후 [Edit] 버튼을 클릭하여 [URL Mapping] 창이 나타나면 [Pattern:] 입력창의 내용을 (/add)를 입력 한 후 OK 버튼 클릭
- 서블릿 클래스 실행 방법
 - doGet 선택
 - *doGet*: 주소를 입력해서 실행
 - *doPost*: 입력 데이터를 넘겨줌
- 서블릿 실행 : localhost:8080/studyjsp/add(Addition의 URL 패턴)

직접하기 예제

- **클래스명** : HundredServlet
- **1-100까지 더하여 결과를 출력**을 doGet() 메소드에 작성하라.

2. 서블릿 프로그램 작성

□ 서블릿 클래스 정의

- ▣ javax.servlet.http.HttpServlet 클래스를 상속받도록 작성
- ▣ 반드시 public 클래스로 작성.

```
public class HundredServlet extends HttpServlet {  
  
}
```

반드시
public 이어야함

프로그래머가 정
한 클래스 이름

서블릿 클래스의
슈퍼클래스

□ Servlet 클래스 작성

- ▣ 서블릿 클래스 안에 **doGet** 또는 **doPost** 메서드를 선언해야 함
 - 두 메서드는 **javax.servlet.http.HttpServletRequest**와 **javax.servlet.http.HttpServletResponse** 타입의 파라미터를 받고, 메서드 밖으로 **javax.servlet.ServletException**과 **java.io.IOException**을 던질 수 있도록 선언
- ▣ **javax.servlet.http.HttpServletRequest**
 - 클라이언트의 요청 처리
- ▣ **javax.servlet.http.HttpServletResponse**
 - 요청처리 결과를 클라이언트에게 응답 처리

2. 서블릿 프로그램 작성

□ 서블릿 클래스 정의

▣ doGet() 메소드

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

HttpServlet 클래스의 doGet 메서드와
리턴 타입, 파라미터 변수, 익셉션 타입이 동일해야 한다.

- **doGet** 메서드를 **public**으로 선언 : 웹 컨테이너가 웹 브라우저로부터 요청을 받아서 메서드를 호출할 때 필요.
- **doGet** 메서드의 **throws** 절의 **ServletException, IOException**는 생략 가능, 다른 익셉션을 추가할 수는 없다

2. 서블릿 프로그램 작성

- 결과 출력 HTML 문서의 종류와 인코딩 방식 지정

```
response.setContentType("text/html;charset=utf-8");
```

- 출력 스트림 얻기

```
PrintWriter out=response.getWriter();
```

- 클라이언트 결과 출력

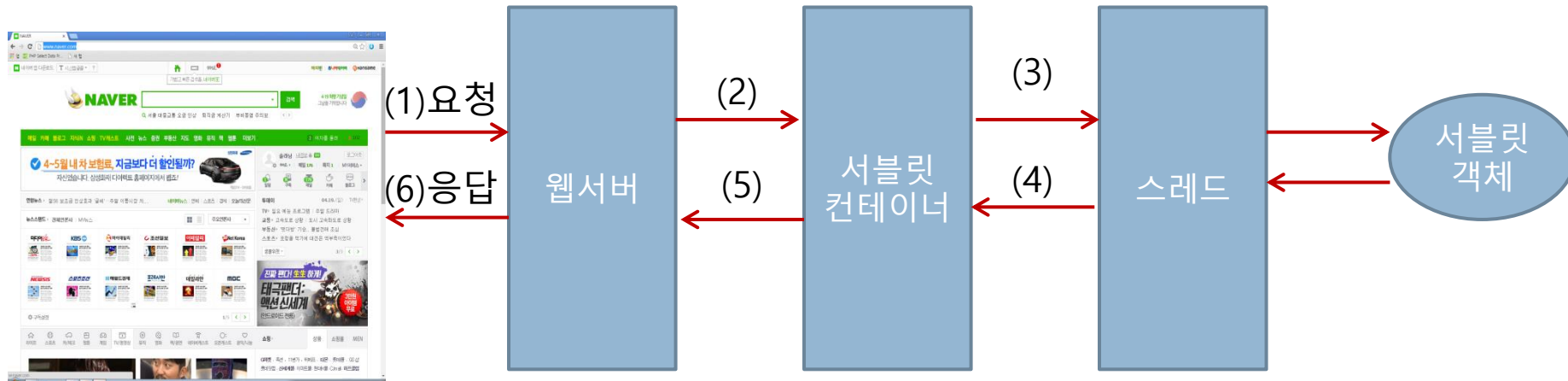
```
out.println("<html>");  
out.println("<head> <title>Servlet</title> </head>");  
out.println("<body>");  
out.println("처음으로 작성하는 Servlet 입니다");  
out.println("</body>");  
out.println("</html>");
```

- 출력 스트림 닫기

```
out.close();
```

2. 서블릿 프로그램 작성

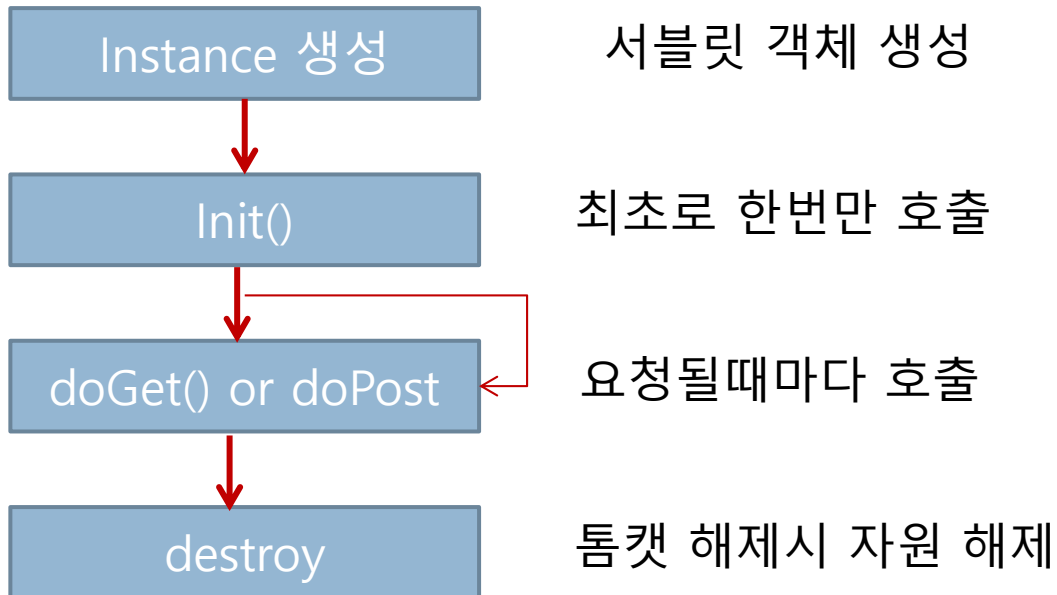
□ 서블릿 동작원리



- ① 브라우저가 서블릿 요청
- ② WAS안에 웹 서버가 서블릿 요청을 인식하여 서블릿 컨테이너에게 서블릿을 수행하도록 넘겨줌
- ③ 서블릿은 스레드를 기동하여 해당 서블릿 객체를 생성하여 이를 수행
- ④ 서블릿 객체의 작업이 종료되면 기동되었던 스레드가 종료
- ⑤ 서블릿 수행 결과가 웹 서버에 전송됨
- ⑥ 이를 클라이언트에게 전송

2. 서블릿 프로그램 작성

□ 서블릿 라이프 사이클



예제 :서블릿의 라이프 사이클 테스트

- 클래스 명: Lifecycle, URL Mapping: 기본
- Which method stubs would you like to create? Init(), destroy()를 추가 체크한다.

```
@WebServlet("/Lifecycle")
public class Lifecycle extends HttpServlet {
    private static final long serialVersionUID = 1L;
    int initCount=1;
    int doGetCount=1;
    int destroyCount=1;

    public Lifecycle() {
        super();
    }
    public void init(ServletConfig config) throws ServletException {
        System.out.println("init 메소드는 첫 요청만 호출됨:" + initCount++);
    }
    public void destroy() {
        System.out.println("destroy 메소드는 콧이 종료될때만 호출됨" + destroyCount++);
    }
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        System.out.println("doGet 메소드가 요청할때마다 호출됨:" + doGetCount++);
    }
}
```

서블릿의 한글 처리와 데이터 통신

□ 응답시 한글 처리

```
response.setContentType("text/html;charset=utf-8");
```

□ get 방식과 post 방식

```
<form method= "get"post" action="호출Servlet"> </form>
```

```
<form method= "get" action="/studyjsp/MethodServlet">  
<input type= "submit" value="get 방식으로 호출하기">  
</form>  
<br> <br>  
<form method= "post" action="/studyjsp/MethodServlet">  
<input type= "submit" value="post 방식으로 호출하기">  
</form>
```


서블릿의 한글 처리와 데이터 통신

□ Get과 post 전송 방식 서블릿

```
protected void doGet(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
// TODO Auto-generated method stub
    response.setContentType("text/html; charset=UTF-8");
    PrintWriter out=response.getWriter();
    out.println("<h1>get 방식으로 처리됨</h1>");
    out.close();
}
```

```
protected void doPost(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
// TODO Auto-generated method stub
    response.setContentType("text/html; charset=UTF-8");
    PrintWriter out=response.getWriter();
    out.println("<h1>post 방식으로 처리됨</h1>");
    out.close();
}
```

쿼리 스트링이란?

□ 쿼리 스트링

- ▣ 사용자가 입력한 데이터를 서버로 전달하는 방법
- ▣ get 방식, post 방식
- ▣ get 방식

- URL 주소뒤에 입력 데이터를 함께 보냄

리소스 ? 변수=값

ParamServlet?id=pinksung & age=15

- 요청 객체(request)와 파라미터 관련 메소드(getParameter)

```
String id=request.getParameter("id");
```

```
Int age=Integer.parseInt(request.getParameter("age");
```

예제: id와 나이 텍스트 값 넘기기

□ Param01.jsp

```
<form method="get" action="ParamServlet" name="frm">  
아이디 : <input type="text" name="id"> <br>  
나 &nbsp; 이 : <input type="text" name="age"> <br>  
<input type="submit" value="전송" onclick="return check()">  
</form>
```

예제: ParamServlet

```
protected void doGet(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
    response.setContentType("text/html;charset=EUC-KR");
    String id = request.getParameter("id");
    int age = Integer.parseInt(request.getParameter("age"));
    PrintWriter out = response.getWriter();
    out.print("<html> <body>");
    out.println("당신이 입력한 정보입니다.<br>");
    out.println("아 이 디 : ");
    out.println(id);
    out.println("<br> 나이 : ");
    out.println(age);
    //자바스크립트로 이전 페이지로 이동하는 링크를 만들어 줌
    out.println("<br> <a href='javascript:history.go(-1)''>다시 </a>");
    out.print("</body> </html>");
    out.close();
}
```

자바 스크립트 폼에 입력된 정보 유효성 체크

□ 자바 스크립트 파일 (param.js)

```
function check() {  
    if (document.frm.id.value == "") {  
        alert("아이디를 입력해주세요.");  
        document.frm.id.focus();  
        return false;  
    } else if (document.frm.age.value == "") {  
        alert("나이를 입력해주세요.");  
        document.frm.age.focus();  
        return false;  
    } else if (isNaN(document.frm.age.value)) {  
        alert("숫자로 입력해주세요.");  
        document.frm.age.focus();  
        return false;  
    } else {  
        return true;  
    }  
}
```

HTML에 외부 자바 스크립트 불러오기

```
<script type="text/javascript"  
src="param.js"> </script>
```

서블릿에서 요청시 한글 처리

□ Info.jsp

```
<form method="get" action="InfoServlet">
이름 : <input type="text" name="name"> <br>
주소 : <input type="text" name="addr"> <br>
<input type="submit" value="전송" >
</form>
<br>
<h3> post 방식에서 한글 깨짐 방지 </h3>
<form method="post" action="InfoServlet">
이름 : <input type="text" name="name"> <br>
주소 : <input type="text" name="addr"> <br>
<input type="submit" value="전송" >
</form>
```

예제 : InfoServlet

```
protected void doGet(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    String name = request.getParameter("name");
    String addr = request.getParameter("addr");
    PrintWriter out = response.getWriter();
    out.print("<html> <body>");
    out.println("당신이 입력한 정보입니다.<br>");
    out.println("이름 : ");
    out.println(name);
    out.println("<br> 주소 : ");
    out.println(addr);
    out.println("<br> <a href='javascript:history.go(-1)'>다시 </a>");
    out.print("</body> </html>");
    out.close();
}
```

InfoServlet

□ Post 방식 입력 처리

- ▣ get 방식과 동일, doPost 메소드에서 doGet()메소드 호출

```
doGet(request, response);
```

- ▣ 한글 처리

```
Request.setCharacterEncoding("UTF-8");
```

- ▣ doPost() 메소드 오버라이딩

```
protected void doPost(HttpServletRequest request,  
    HttpServletResponse response) throws ServletException, IOException {  
    doGet(request, response);  
    request.setCharacterEncoding("UTF-8");  
}
```


기타 다양한 입력 양식

□ 암호 입력 상자 예제

▣ 로그인 폼만들기(login.jsp

```
<form method="get" action="LoginServlet">  
  <label for="userid"> 아이디 : </label> <input type="text" name="id"  
    id="userid"><br> <label for="userpwd"> 암호 &nbsp;   <br>  
    호 : </label> <input type="password" name="pwd" id="userpwd"> <br>  
  <input type="submit" value="로그인">  
</form>
```

예제: loginServlet

```
protected void doGet(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    String id = request.getParameter("id");
    String pwd = request.getParameter("pwd");
    PrintWriter out = response.getWriter();
    out.print("<html> <body>");
    out.println("당신이 입력한 정보입니다.<br>");
    out.println("아이디 : ");
    out.println(id);
    out.println("<br> 비밀번호 : ");
    out.println(pwd);

    out.println("<br> <a href='javascript:history.go(-1)'>다시 </a>");
    out.print("</body> </html>");
    out.close();
}
```

Textarea와 Radio 사용

□ Radio.jsp

```
<form method="get" action="RadioServlet">
  <label for="gender"> 성별 : </label> <input type="radio" id="gender"
    name="gender" value="남자" checked> 남자 <input type="radio"
    id="gender" name="gender" value="여자"> 여자 <br>
  <br> <label for="chk_mail"> 메일 정보 수신 여부 : </label> <input
    type="radio" id="chk_mail" name="chk_mail" value="yes" checked>
수신 <input type="radio" id="chk_mail" name="chk_mail" value="no">
거 부<br>
  <br> <label for="content"> 간단한 가입 인사를 적어주세요 ^o^ </label>
  <textarea id="content" name="content" rows="3" cols="35"> </textarea> <br>
  <input type="submit" value="전송">

</form>
```

예제 : RadioServlet

```
protected void doGet(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    String gender = request.getParameter("gender");
    String chk_mail = request.getParameter("chk_mail");
    String content = request.getParameter("content");
    PrintWriter out = response.getWriter();
    out.print("<html> <body>");
    out.println("당신이 입력한 정보입니다.<hr>");
    out.println("성별 : <b>");
    out.println(gender);
    out.println("</b> <br> 메일 정보 수신 여부 : <b>");
    out.println(chk_mail);
    out.println("</b> <br> 가입 인사 : <b> <pre>");
    out.println(content);
    out.println("</b> </pre> <a href='javascript:history.go(-1)'>다시 </a>");
    out.print("</body> </html>");
    out.close();
}
```

Checkbox, getParameterValues()

□ check.jsp

```
<h2>악세사리</h2>
  관심항목을 선택하세요.
  <hr>
  <form method="get" action="CheckboxServlet">
    <input type="checkbox" name="item" value="신발"> 신발 <input
      type="checkbox" name="item" value="가방"> 가방 <input
        type="checkbox" name="item" value="벨트"> 벨트<br> <input
          type="checkbox" name="item" value="모자"> 모자 <input
            type="checkbox" name="item" value="시계"> 시계 <input
              type="checkbox" name="item" value="주얼리"> 주얼리<br> <input
                type="submit" value="전송">
  </form>
```

예제 : CheckServlet

```
protected void doGet(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    out.print("<html> <body>");
    String items[] = request.getParameterValues("item");
    if (items == null) {
        out.print("선택한 항목이 없습니다.");
    } else {
        out.println("당신이 선택한 항목입니다.<hr>");
        for (String item : items) {
            out.print(item + " ");
        }
    }
    out.println("<br> <a href='javascript:history.go(-1)'> 다시 </a>");
    out.print("</body> </html>");
    out.close();
}
```

□ Select.jsp

```
<form method="get" action="SelectServlet">
  <span style="float: left; margin-right: 20px"> <label for="job">직업</label>
    <select id="job" name="job" size="1">
      <option value="">선택하세요</option>
      <option value="학생">학생</option>
      <option value="컴퓨터/인터넷">컴퓨터/인터넷</option>
      <option value="언론">언론</option>
      <option value="공무원">공무원</option>
      <option value="군인">군인</option>
      <option value="서비스업">서비스업</option>
      <option value="교육">교육</option>
    </select>
  </span> <label for="interest" style="float: left;">관심분야</label> <select
    id="interest" name="interest" size='5' multiple="multiple">
      <option value="에스프레소">에스프레소</option>
      <option value="로스팅">로스팅</option>
      <option value="생두">생두</option>
      <option value="원두">원두</option>
      <option value="핸드드립">핸드드립</option>
    </select>
  <br><br>
  <input type="submit" value="전송" style="float: right; margin-right: 50px">
</form>
```

예제 : SelectServlet

```
protected void doGet(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
    String job = request.getParameter("job");
    String interests[] = request.getParameterValues("interest");
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    out.print("<html> <body>");
    out.println("당신이 선택한 직업 : <b>");
    out.print(job);
    out.println("</b> <hr>당신이 선택한 관심 분야 : <b>");
    if (interests == null) {
        out.print("선택한 항목이 없습니다.");
    } else {
        for (String interest : interests) {
            out.print(interest + " ");
        }
    }
    out.println("</b> <br> <a href='javascript:history.go(-1)'>다시 </a>");
    out.print("</body> </html>");
    out.close();
}
```