

Java Web Programming 입문

(Servlet/JSP)

23일차

오늘의 키워드

- ▶ Get / Post
- ▶ 한글처리
 - Get / Post
- ▶ HTML
 - Get / Post

Get / Post

▶ Get

- URL주소에 보낼 정보를 포함하여 정보를 전달
- HTTP Message 요청 라인
 - GET /MySample/servlet/HttpGetServlet?name=Grace HTTP/1.1
- URL에 정보가 노출되어 보안 취약
- 보낼 수 있는 정보의 크기가 1024bytes 로 제한

▶ Post

- 첨부파일 형태로 보낼 정보를 포함시켜 정보를 전달
- HTTP Message 요청 라인
 - POST /MySample/servlet/HttpPostServlet HTTP/1.1
 - ... HTTP Header ...
 - name=Grace
- URL에 정보가 노출되지 않으므로 보안성이 좋음
- 보낼 수 있는 정보의 크기 제한이 없음

HTML

▶ <form> Tag

- Client에서 Server로 정보를 전달하는 양식을 만들기 위한 Tag
- 속성
 - action
 - 전달된 정보를 처리할 CGI 프로그램의 위치를 지정
 - method
 - 정보전달 방식 지정 (Get / Post)

▶ <Input> Tag

- 입력 양식을 만들기 위한 가장 기본적인 Tag
- Text 상자 및 Button을 만들기 위한 Tag
- 속성
 - type
 - 입력 양식의 종류를 지정 (text / submit)
 - name
 - 입력 양식의 이름을 지정 (이후 Parameter 이름으로 사용)
 - value
 - <input> Tag를 이용하여 Button을 만든 경우, Button에 나타날 값을 지정

▶ formTest.html

```
<form action="formTest.jsp" method="GET">  
  <input type="text" name="name" >  
  <input type="submit" value="Submit">  
</form>
```

HTML

▶ JSP에서 Parameter 처리

◦ request Object

- Servlet
 - Client의 요청과 응답정보를 담는 HttpServletRequest, HttpServletResponse Object
- JSP 에서도 이 Object 이용

```
public void _jspService ( HttpServletRequest request,
                          HttpServletResponse response )
    throws java.io.IOException, ServletException {
    ...
}
```

- JSP Page 에서 변환된 Servlet의 _jspService() Method
 - HttpServletRequest, HttpServletResponse 가 Input Parameter
- 별도의 선언 없이 request, response Object를 이용

```
<% String str = request.getParameter("name"); %>
```

- request Object처럼 선언하지 않고 사용하는 객체 : 내장 객체

HTML

▶ formTest.jsp

```
<%@ page contentType="text/html; charset=euc-kr"%>

<html>
  <head>
    <title>JSP에서 파라미터 처리하기</title>
  </head>
  <body>
    <h2>파라미터의 값</h2>
    <%
      String name = request.getParameter("name");
      out.println(name);
    %>
  </body>
</html>
```

- <input> Tag의 name 속성에 값을 name으로 주었으므로
 - JSP Page로 전달되는 Parameter 이름도 name

한글 처리

▶ Web Browser에서 한글 처리

- 응답으로 전달된 HTTP Header 중
 - Content-Type 항목에 따라 HTML Page를 해석
- HTTP Header의 Content-Type 에
 - 잘못된 Encoding 값이 들어갈 경우 한글 처리가 제대로 되지 않음
- JSP 에서 HTTP Header의 Content-Type 지정
 - page 지시문 contentType 속성
 - 한글 Encoding 을 위해 지시문의 contentType의 속성값
 - “euc-kr”

```
<%@ page contentType="text/html; charset=euc-kr" %>
```

```
response.setContentType("text/html; charset=euc-kr");
```

한글 처리

▶ 한글 처리 (Get 방식)

- URL은 영문자만을 사용
 - Get방식에서 Parameter 값으로 한글 전달 불가
 - 한글을 내부적으로 URL Encoding 기법 이용
 - 영문자 열로 변환 후 전송
 - server.xml 파일 설정 변경

한글 처리

▶ encodingGet.html

```
<HTML>
  <HEAD>
    <TITLE>GET 방식의 인코딩</TITLE>
  </HEAD>
  <BODY>
    <form action="encodingGET.jsp" method="GET">
      <input type="text" name="name" >
      <input type="submit" value="Submit">
    </form>
  </BODY>
</HTML>
```

▶ encodingGet.jsp

```
<%@ page contentType="text/html; charset=euc-kr"%>

<html>
  <head>
    <title>GET 방식의 인코딩</title>
  </head>
  <body>
    <h2>GET 방식의 한글 인코딩</h2>
    <%
      String name = request.getParameter("name");
      out.println(name);
    %>
  </body>
</html>
```

한글 처리

▶ Get 방식에서의 한글 전송

- <http://localhost:8080/MySample/encodingGET.jsp?name=%EC%9D%B4%EB%AA%85%EC%A7%84>
- 한글을 Get 방식으로 전송할 경우
 - Encoding 과정을 거침
 - Encoding되어 전달된 한글 처리를 위해
 - Tomcat의 설정 변경
 - .../conf/server.xml

```
<Connector port                = "80"           maxHttpHeaderSize  = "8192"
             URIEncoding        = "euc-kr"        maxThreads          = "150"
             minSpareThreads     = "25"           maxSpareThreads     = "75"
             enableLookups       = "false"        redirectPort        = "8443"
             acceptCount         = "100"          connectionTimeout   = "20000"
             disableUploadTimeout = "true"
/>
```

한글 처리

▶ 한글 처리 (Post 방식)

- 정보를 첨부 형태로 전송하는 방식
- Post 방식에서 한글을 올바르게 처리하려면
 - `getParameter()` Method 호출 전 Encoding
 - request 내장 객체 내 `setCharacterEncoding()` Method 이용

한글 처리

▶ encodingPOST.html

```
<HTML>
  <HEAD>
    <TITLE>POST 방식의 인코딩</TITLE>
  </HEAD>
  <BODY>
    <form action="encodingPOST.jsp" method="POST">
      <input type="text" name="name" >
      <input type="submit" value="Submit">
    </form>
  </BODY>
</HTML>
```

▶ encodingPOST.jsp

```
<%@ page contentType="text/html; charset=euc-kr"%>
<html>
  <head>
    <title>POST 방식의 인코딩</title>
  </head>
  <body>
    <h2>POST 방식의 한글 인코딩</h2>
    <%
      String name = request.getParameter("name");
      out.println(name);
    %>
  </body>
</html>
```

한글 처리

▶ POST 방식의 메시지 처리를 위한 JSP

```
<%  
    request.setCharacterEncoding("euc-kr");  
    String name = request.getParameter("name");  
    out.println(name);  
%>
```

HTML

- ▶ **<form> tag**
 - Web Server로 전달되는 입력 양식을 만들기 위한 기본 tag
 - 속성
 - action
 - 사용자로부터 입력 받은 정보를 처리하는 Servlet 지정
 - method
 - 정보 전달 방식 지정
- ▶ **<input> tag**
 - text field(한줄), check box, radio button, button 등에 사용
- ▶ **<textarea> tag**
 - 한 줄 이상 입력이 가능한 Text Box tag
- ▶ **<select>, <option>**
 - 선택 목록을 만들기 위한 tag

HTML

▶ <input> tag

- name

- <input> tag의 이름, Parameter 이름으로 사용

- type

- <input> tag의 종류 지정
- text로 지정하면 한 줄 Text Field

```
<input type="text" name="name">
```

- Password type Text Field

- 일반 Text Field와 모양은 동일
- 입력되는 글이 "*" 기호로 표시
- 비밀번호, 주민등록번호 등 화면에 보이지 않는 Text 입력 시 사용

```
<input type="password" name="pwd">
```

- Hidden Text Field

- 사용자에게 보이지 않는 Text Field
- 사용자에게 값을 입력 받기 위한 용도가 아님
- 항상 고정되어 있는 값을 전달하기 위해 사용

```
<input type="hidden" value="3" name="hiddenText">
```

HTML

▶ <textarea> tag

- 여러 줄을 입력할 수 있는 Text Box 생성용 tag
- name
 - <textarea> 의 이름, Parameter 이름으로 사용
- cols
 - <textarea> 의 좌우 폭을 지정
- rows
 - <textarea> 의 상하 크기를 지정
- example
 - 좌우 30글자, 상하 4줄 크기의 Text Box

```
<textarea name="myTextarea" cols="30" rows="4">  
초기 표시 문자열  
</textarea>
```


HTML

▶ Button

- <input> tag를 이용하여 다양한 형태의 Button 생성
- 전송 Button
 - <form> tag의 action 속성에 지정된 자원을 요청

```
<input type="submit" value="전송">
```

- 초기화 Button
 - 모든 Text Box에 입력되어 있는 값을 초기화

```
<input type="reset" value="초기화">
```

- 파일 찾기 Button
 - Client에 존재하는 File을 선택하여 Server 전송하기 위한 용도
 - 게시판 글, 메일 송신과 같이 첨부 파일이 필요할 때 사용

```
<input type="file" name="fileName">
```

HTML

▶ Button

◦ 일반 Button

- 사용자가 직접 JavaScript와 같은 언어를 이용하여
 - 기능을 정의할 수 있는 Button 제작 시 사용

```
<input type="button" value="클릭">
```

◦ Image Button

- 일반 Button을 Image File로 지정
- type 속성을 image
- src 속성에 Image 파일에 경로 지정
- 일반 Button과 같이 JavaScript와 같은 언어로 기능 정의

```
<input type="image" src="imageButton.gif">
```

HTML

▶ 만들어볼까요 (HTML)

◦ <form> tag

```
<form    action="textExample.jsp"    method="post">
    ...
</form>
```

◦ 입력양식

```
<input    type="text"        name="name">
<input    type="password"    name="pwd">
<input    type="hidden"      name="hiddenText"    value="3">
<textarea name="myTextarea"  cols="25"    rows="4">
</textarea>
<input    type="submit"      value="Submit">
```

HTML

▶ textExample.html

```
<html>
  <head>
    <title>텍스트 필드 만들기</title>
  </head>
  <body>
    <form action="textExample.jsp" method="post">이름 :
      <input type="text"      name="name"><p>암호 :
      <input type="password"  name="pwd"><p>
      <input type="hidden"    name="hiddenText" value="3">기타 :
      <textarea name="myTextarea" cols="25" rows="4">
      </textarea><p>
      <input type="submit"    value="Submit">
    </form>
  </body>
</html>
```

HTML

▶ 만들어볼까요 (JSP)

◦ Post 방식 Message Encoding

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

◦ Parameter 값 추출

- <input> tag 내 name 속성에 지정된 값

```
String name      = request.getParameter("name"      );  
String pwd       = request.getParameter("pwd"       );  
String hidden    = request.getParameter("hiddenText");  
String etc       = request.getParameter("myTextarea");
```

HTML

▶ textExample.jsp

```
<%@ page contentType="text/html; charset=euc-kr"%>

<html>
  <head>
    <title>텍스트 필드 처리하기</title>
  </head>
  <body>
    <%
      request.setCharacterEncoding("euc-kr");
      String name      = request.getParameter("name"      );
      String pwd        = request.getParameter("pwd"        );
      String hidden     = request.getParameter("hiddenText" );
      String etc        = request.getParameter("myTextarea" );

      %>
      이름 : <%=name%><p>
      암호 : <%=pwd%><p>
      등급 : <%=hidden%><p>
      기타 : <%=etc%>
    </body>
  </html>
```

HTML

▶ Radio Button

- 여러 가지 항목 중 한가지를 선택할 수 있는 선택형 Button
- `<input>` Tag의 type 속성 radio
- 동일한 항목들에 대해 똑같은 name 속성값 부여
- value 속성의 값에 각 항목 고유 값 작성
 - 이 속성 값이 Parameter 값으로 사용

```
<input type="radio" name="gender" value="man" >남자  
<input type="radio" name="gender" value="woman">여자
```

- 남자 선택 시 전달되는 URL
 - <http://localhost:8080/MySample/selectButton.jsp?gender=man>
- 잘못 작성한 Radio Button

```
<input type="radio" name="sex" value="man" >남자  
<input type="radio" name="gender" value="woman">여자
```

HTML

▶ Check Box

- 여러 가지 항목 중 하나 이상 선택 가능한 선택 형 Button
 - <input> tag의 type 속성 값 checkbox
 - 동일 항목들은 같은 name 속성을 가져야 함
 - value 속성의 값이 Parameter 값으로 사용

```
<input type="checkbox" name="hobby" value="ski" >스키  
<input type="checkbox" name="hobby" value="inline">인라인
```


HTML

▶ 만들어 볼까요 (HTML)

◦ selectButton.html

```
<html>
  <head>
    <title>선택형 버튼 만들기</title>
  </head>
  <body>
    <form action="selectButton.jsp" method="get">
      성별 :
      <input type="radio"      name="gender"      value="Man"      >남자
      <input type="radio"      name="gender"      value="Woman"    >여자<p>
      취미 :
      <input type="checkbox"      name="hobby"        value="Ski"      >스키
      <input type="checkbox"      name="hobby"        value="Inline"   >인라인<p>
      <input type="submit" value="Submit">
    </form>
  </body>
</html>
```

HTML

▶ 선택형 Button 요청 예제 JSP

◦ Radio Button 값 추출

- <input> tag의 name 속성 값이 Parameter 이름
- request Object의 getParameter() Method 이용하여 값 추출
 - 이때 추출되는 값은 선택된 항목의 value 값

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

◦ Check Box 의 값 추출

- 한 번에 여러 개 항목 선택이 가능하므로
 - Check Box로 추출된 값을 배열(Array)형태
- getParameterValues() Method 이용하여 값 추출

```
java.lang.String[] getParameterValues(java.lang.String name)
```

```
String[] hobby = request.getParameterValues("hobby");
```

- getParameterValues() Method의 Return Type은 String[]
 - 추출되는 값은 선택된 항목의 value 속성 값 배열

```
for(int i = 0; i < hobby.length; i++) {  
    out.print(hobby[i] + " ");  
}
```

HTML

- ▶ 만들어 볼까요 (JSP)
 - selectButton.jsp

```
<%@ page contentType="text/html; charset=euc-kr"%>

<html>
  <head>
    <title>선택형 버튼 처리하기</title>
  </head>
  <body>
    <%
      String gender = request.getParameter("gender");
      String[] hobby = request.getParameterValues("hobby");
    %>
    성별 :
      <%=gender%><p>
    취미 :
    <%
      for(int i = 0; i < hobby.length; i++) {
        out.print(hobby[i] + " ");
      }
    %>
  </body>
</html>
```

HTML

▶ Select Box

- `<select>` tag
 - 여러 개의 값 중 특정 값을 선택할 수 있는 Select Box 생성
 - 속성
 - `name`
 - `<select>` tag의 이름을 지정, Parameter 이름으로 사용
 - `multiple`
 - 다중 Select Box 생성
- `<option>` tag
 - Select Box에 사용될 하나하나의 항목을 작성
 - 하나의 항목은 하나의 `<option>` tag로 작성됨
 - 속성
 - `selected`
 - 초기 선택 여부 지정
- `<select>` tag의 `name` 속성 값이 Parameter 이름
- `<option>` tag의 값이 Parameter의 값으로 전달

HTML

▶ Select Box 사용 예제 (HTML)

◦ 단일 Select Box

```
<select name="live">  
  <option selected>서울</option>  
  <option>경기도</option>  
  <option>강원도</option>  
</select>
```

◦ 다중 Select Box

```
<select name="language" multiple>  
  <option>JSP</option>  
  <option>ASP</option>  
  <option>Java</option>  
  <option>C#</option>  
</select>
```

HTML

- ▶ 만들어 볼까요 (HTML)
 - selectBox.html

```
<html>
  <head>
    <title>선택 박스 만들기</title>
  </head>
  <body>
    <form action="selectBox.jsp" method="get">
      사는 곳 :
      <select name="live">
        <option selected>서울</option>
        <option>경기도</option>
        <option>강원도</option>
      </select><p>
      개발 언어 :
      <select name="language" multiple>
        <option>JSP</option>
        <option>ASP</option>
        <option>Java</option>
        <option>C#</option>
      </select><p>
      <input type="submit" value="Submit">
    </form>
  </body>
</html>
```

HTML

▶ Select Box 사용 예제 (JSP)

- 단일 Select Box는 하나의 값만 선택 가능
 - `getParameter()` Method 이용하여 값 추출

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

- 다중 Select Box는 여러 개의 값을 선택 가능
 - `getParameterValues()` Method 이용하여 값 추출

```
String[] language = request.getParameterValues("language");  
  
for(int i = 0; i < language.length; i++) {  
    out.print(language[i] + " ");  
}
```

HTML

- ▶ 만들어 볼까요 (JSP)
 - selectBox.jsp

```
<%@ page contentType="text/html; charset=euc-kr"%>
<html>
  <head>
    <title>선택 박스 처리하기</title>
  </head>
  <body>
    <%
      String live = request.getParameter("live");
      String[] language = request.getParameterValues("language");
    %>
    사는 곳 :
    <%=live%><p>
    개발 언어 :
    <%
      for(int i = 0; i < language.length; i++) {
        out.print(language[i] + " ");
      }
    %>
  </body>
</html>
```


HTML

▶ getParameterNames() Method

- Parameter 이름 집합을 Enumeration Object로 반환
 - 전체 Parameter 를 쉽게 다룰 수 있음

```
java.util.Enumeration getParameterNames ()
```

- 사용

```
Enumeration param = request.getParameterNames ();
```

- param Object에는
 - HTML Page로부터 전달된 전체 Parameter 이름들이 있음
 - Parameter 값 추출

```
while (param.hasMoreElements ()) {  
    String name = (String) param.nextElement ();  
    out.println (name + request.getParameter (name));  
}
```

HTML

- ▶ 만들어 볼까요 (HTML)
 - parameterNames.html

```
<html>
  <head>
    <title>getParameterNames () </title>
  </head>
  <body>
    <form action="ParameterNames.jsp" method="post">
      이름 :
      <input type="text" name="name"><br>
      성별 :
      <input type="text" name="gender"><br>
      나이 :
      <input type="text" name="age"><br>
      생일 :
      <input type="text" name="birth"><br>
      <input type="submit" text="Submit">
    </form>
  </body>
</html>
```

HTML

- ▶ 만들어 볼까요 (JSP)
 - ParameterNames.jsp

```
<%@ page contentType="text/html; charset=euc-kr"%>
<%@ page import="java.util.*"%>

<html>
  <head>
    <title>getParameterNames() </title>
  </head>
  <body>
    <%
      request.setCharacterEncoding("euc-kr");
      Enumeration param = request.getParameterNames();
      while(param.hasMoreElements()) {
        String name = (String)param.nextElement();
        out.println(
          name                                +
          " : "                               +
          request.getParameter(name)         +
          "<br>"
        );
      }
    %>
  </body>
</html>
```

오늘 숙제

▶ 집에서 해보자!