

JSP 내장객체

524730
2022년 봄학기
3/16/2022
박경신

JSP 내장 객체

□ 내장 객체(Implicit Objects)

- JSP 페이지에서 사용할 수 있도록 JSP 컨테이너에 미리 정의된 객체
- JSP 페이지가 서블릿 프로그램으로 번역될 때 JSP 컨테이너가 자동으로 내장 객체를 멤버 변수, 메소드 매개변수 등의 각종 참조 변수(객체)로 포함
- JSP 페이지에 별도의 import 문 없이 자유롭게 사용 가능
- 스크립틀릿 태그나 표현문 태그에 선언을 하거나 객체를 생성하지 않고도 직접 호출하여 사용 가능
- **request, response, out, session**, application, config, pageContext, page, exception

JSP 내장 객체

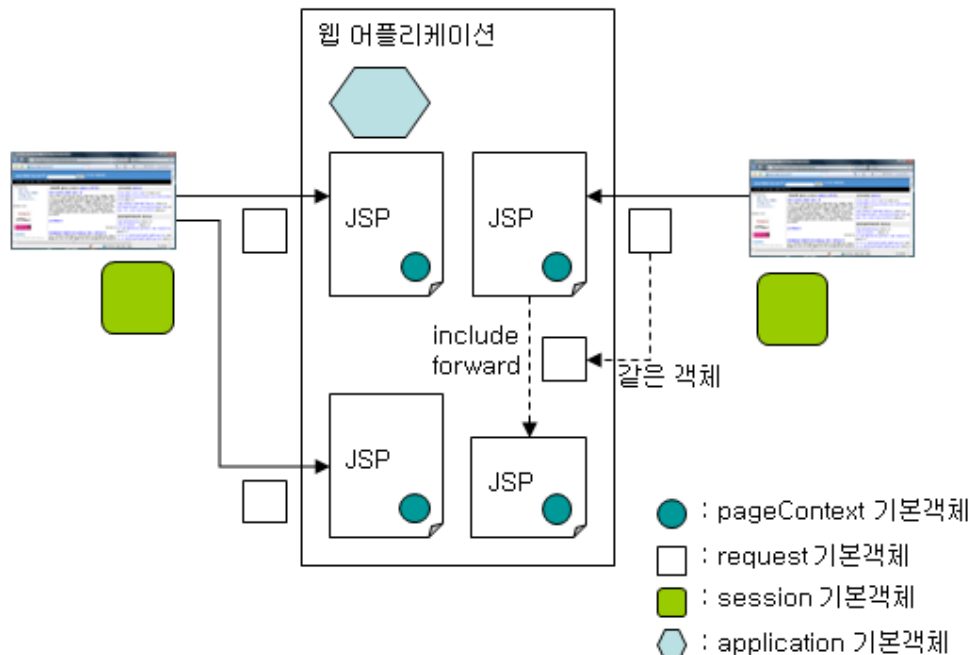
- 내장 객체(Implicit Objects) 주요활용
 - HTML 폼에서 입력한 값을 가지고 올 때
 - 세션 관련 작업을 처리할 때
 - 스크립트릿에서 브라우저 출력이 필요할 때
 - 사용자 요청을 다른 페이지로 전달할 때
 - 현재 JSP에 대한 각종 정보를 알고자 할 때

내장 객체

기본 객체	실제 타입	설명
request	javax.servlet.http.HttpServletRequest	웹 브라우저의 HTTP 요청 정보를 저장
response	javax.servlet.http.HttpServletResponse	웹 브라우저의 HTTP 요청에 대한 응답 정보를 저장
pageContext	javax.servlet.jsp.PageContext	JSP 페이지에 대한 정보를 저장
session	javax.servlet.http.HttpSession	웹 브라우저의 정보를 유지하기 위한 세션 정보를 저장
application	javax.servlet.ServletContext	웹 애플리케이션의 컨텍스트 정보를 저장
out	javax.servlet.jsp.JspWriter	JSP 페이지에 출력할 내용을 담고 있는 출력 스트림
config	javax.servlet.ServletConfig	현재 JSP 페이지의 설정 정보를 저장
page	java.lang.Object	JSP 페이지를 구현한 자바 클래스 인스턴스
exception	java.lang.Throwable	예외 객체. 에러 페이지에서 사용

내장 객체와 영역

- PAGE - 하나의 JSP 페이지를 처리할 때 사용되는 영역
- REQUEST - 하나의 HTTP 요청을 처리할 때 사용되는 영역
- SESSION - 하나의 웹 브라우저와 관련된 영역
- APPLICATION - 하나의 웹 어플리케이션과 관련된 영역



내장 객체 속성 처리 메소드

- request, session, application, pageContext는 속성 기능 제공
- 속성 처리 메소드

메소드	설명
setAttribute(name, value)	해당 내장 객체의 속성 이름이 name인 속성 값을 value로 저장
getAttribute(name)	해당 내장 객체의 속성 이름이 name인 속성 값을 가져옴
removeAttribute(name)	해당 내장 객체의 속성 이름이 name인 속성을 삭제
getAttributeNames()	해당 내장 객체의 모든 속성 이름을 java.util.Enumeration 형태로 가져옴 (단, pageContext 내장 객체는 이 메소드를 제공하지 않음)

내장 객체 속성의 활용

내장 객체	영역	사용법
pageContext	PAGE	(한번의 요청을 처리하는) 하나의 JSP 페이지 내에서 공유될 값을 저장
request	REQUEST	한번의 요청을 처리하는 데 사용되는 모든 JSP 페이지에서 공유될 값을 저장
session	SESSION	한 사용자와 관련된 정보를 JSP 들이 공유하기 위해서 사용
application	APPLICATION	모든 사용자와 관련해서 공유할 정보를 저장

request 내장 객체

- **request (javax.servlet.http.HttpServletRequest)**
- JSP 페이지에서 가장 많이 사용되는 기본 내장 객체
- 사용자 요청(request)과 관련된 정보를 처리할 때 사용
- 웹 브라우저에서 서버의 JSP 페이지로 전달하는 정보를 저장
 - 폼 페이지로부터 입력된 데이터를 전달하는 요청 파라미터 값을 JSP 페이지로 가져옴 (주로 **HTML Form 요소 선택 값과 같은 사용자 입력 정보를 읽어올 때 사용**)
 - JSP 컨테이너는 웹 브라우저에서 서버로 전달되는 정보를 처리하기 위해 **javax.servlet.http.HttpServletRequest** 객체 타입의 request 내장 객체를 사용하여 사용자의 요구 사항을 얻어냄

request 내장 객체

□ **getParameterXXX 메서드**

- 요청 파라미터는 사용자가 폼 페이지에 데이터를 입력한 후 서버에 전송할 때 전달되는 폼 페이지의 입력된 정보 형태를 말함
- 요청 파라미터는 <name=value> 형식으로 웹 브라우저에서 서버의 JSP 페이지로 전송

메소드	설명
getParameterNames()	현재 요청에 포함된 인수 이름을 Enumeration 형태로 전달
getParameter(name)	문자열 name과 같은 이름을 가진 인수 값을 가져 옴
getParameterValues(name)	문자열 name과 같은 이름을 가진 인수 값을 배열 형태로 가져 옴. checkbox, multiple list 등에 주로 사용
getParameterMap()	모든 요청된 인수와 값을 java.util.Map 객체 타입으로 전달

request 내장 객체

```
<body>
<form action="test.jsp" method="GET" >
  이름: <input type="text" name="name" > <br>
  나이: <input type="text" name="age" > <br>
  <input type="submit" value="submit" >
</form>
</body>
```

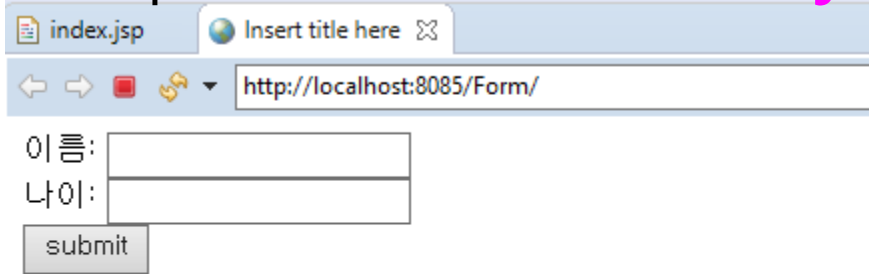
index.jsp

```
<body>
<h1>Using Get method to read form data</h1>
<ul>
  <li>이름: <%= request.getParameter("name") %>
  <li>나이: <%= request.getParameter("age") %>
</ul>
</body>
```

test.jsp

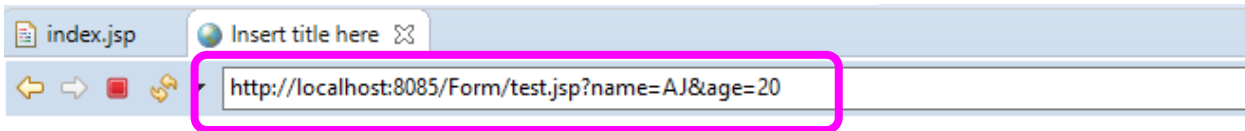
request 내장 객체

□ <http://localhost:8085/test.jsp?name=AJ&age=0>



이름:
나이:
submit

index.jsp



http://localhost:8085/Form/test.jsp?name=AJ&age=20

Using Get method to read form data

- 이름: AJ
- 나이: 20

test.jsp

request 내장 객체

```
<body>
<form action="test.jsp" method="POST" >
  이름: <input type="text" name="name" > <br>
  나이: <input type="text" name="age" > <br>
  <input type="submit" value="submit" >
</form>
</body>
```

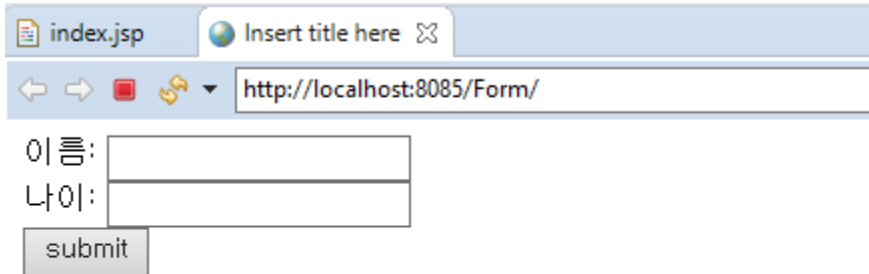
index.jsp

```
<body>
<h1>Using Get method to read form data</h1>
<ul>
  <li>이름: <%= request.getParameter("name") %>
  <li>나이: <%= request.getParameter("age") %>
</ul>
</body>
```

test.jsp

request 내장 객체

□ <http://localhost:8085/test.jsp>



index.jsp Insert title here

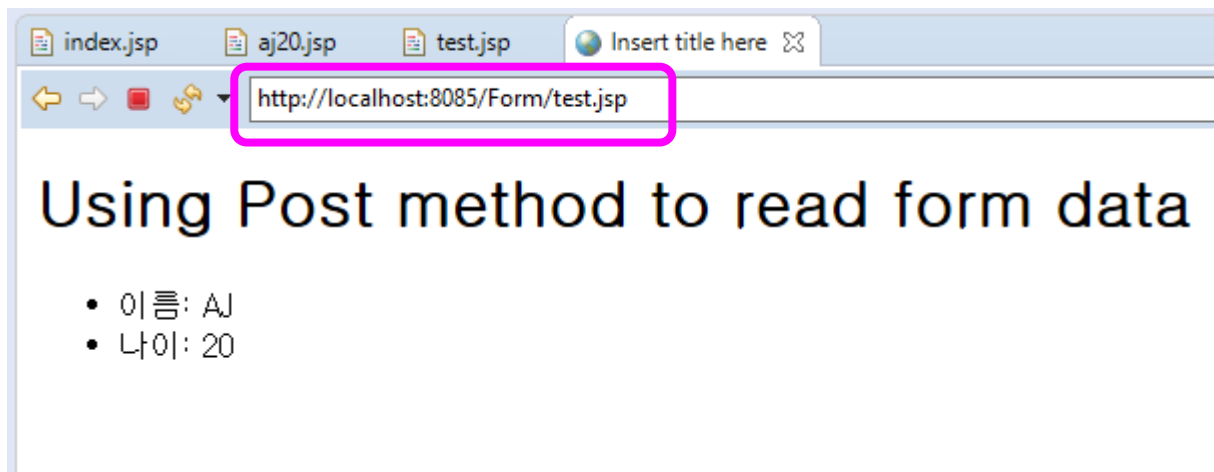
← → http://localhost:8085/Form/

이름:

나이:

submit

index.jsp



index.jsp aj20.jsp test.jsp Insert title here

← → http://localhost:8085/Form/test.jsp

Using Post method to read form data

- 이름: AJ
- 나이: 20

test.jsp

request 내장 객체

□ 요청 HTTP 헤더 관련 메서드

- 웹 브라우저는 HTTP 헤더에 추가적인 정보를 담아 서버로 전송

메소드	설명
getHeader(name)	설정된 name의 헤더 값을 가져옴
getHeaders(name)	설정된 name의 헤더 목록 값을 가져옴
getHeaderNames()	모든 헤더 이름을 가져옴
getIntHeader(name)	설정된 name의 헤더 값을 정수로 가져옴
getDateHeader(name)	설정된 name의 헤더 값을 시간 값으로 가져옴
getCookies()	모든 쿠키 값을 javax.servlet.http.Cookie의 배열 형태로 가져옴

request 내장 객체

▣ 웹브라우저/서버 관련 메소드

메소드	설명
getRemoteAddr()	웹브라우저 IP 주소를 알려줌
getMethod()	현재 요청이 GET, POST인지 가져옴
getSession()	현재 세션 객체를 가져옴
getProtocol()	현재 서버의 프로토콜을 문자열 형태로 알려줌
getRequestURI()	웹브라우저가 요청한 URI 경로를 가져옴
getContentType()	웹브라우저의 콘텐츠 유형을 가져옴
getContentLength()	웹브라우저의 요청 파라미터 길이를 가져옴
getQueryString()	웹브라우저의 전체 요청 파라미터 문자열(?다음 URL에 할당된 문자열)을 가져옴
getCharacterEncoding()	웹브라우저의 문자 인코딩을 가져옴
setCharacterEncoding()	현재 jsp 로 전달되는 내용을 지정한 캐릭터셋으로 변환. html form에서 한글 입력 시 정상적으로 처리하려면 반드시 필요

response 내장 객체

- **response (javax.servlet.http.HttpServletResponse)**
- 사용자의 요청을 처리한 결과를 서버에서 웹 브라우저로 전달하는 정보를 저장하고 서버는 응답 헤더와 요청 처리 결과 데이터를 웹 브라우저로 보냄
- JSP 컨테이너는 서버에서 웹 브라우저로 응답하는 정보를 처리하기 위해 **javax.servlet.http.HttpServletResponse** 객체 타입의 response 내장 객체를 사용하여 사용자의 요청에 응답

response 내장 객체

□ 페이지 이동 관련 메소드

- 페이지 이동 = 리다이렉션(redirectation)
 - 사용자가 새로운 페이지를 요청할 때와 같이 페이지를 강제로 이동하는 것
- 서버는 웹 브라우저에 다른 페이지로 강제 이동하도록 response 내장 객체의 리다이렉션 메소드를 제공
- 페이지 이동 시에는 문자 인코딩을 알맞게 설정해야 함

메소드	설명
<code>sendRedirect(url)</code>	설정된 URL 페이지로 강제 이동

response 내장 객체

```
<body>  
<form action= "test-redirect.jsp">  
<input type= "submit" value= "go"> <br/>  
</form>  
</body>
```

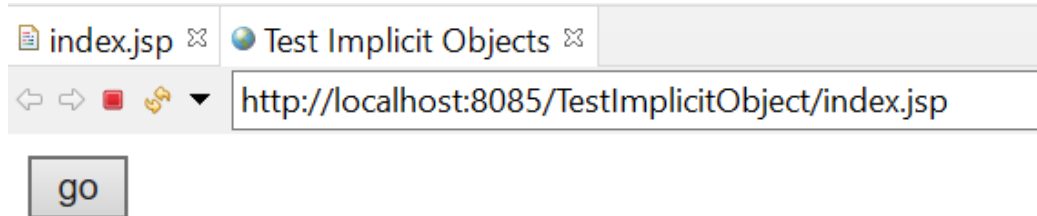
index.jsp

```
<body>  
<%  
response.sendRedirect("https://dis.dankook.ac.kr/lectures/");  
%>  
</body>
```

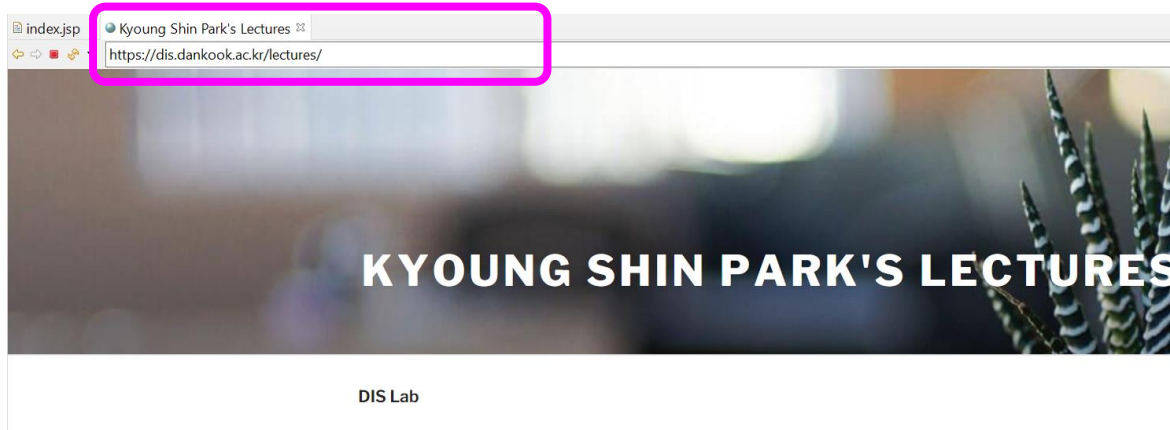
test-redirect.jsp

response 내장 객체

□ <https://dis.dankook.ac.kr/lectures/>



`index.jsp`



`test-redirect.jsp`

response 내장 객체

□ 응답 HTTP 헤더 관련 메서드

- 응답 HTTP 헤더 관련 메소드는 서버가 웹 브라우저에 응답하는 정보에 헤더를 추가하는 기능을 제공
- 헤더 정보에는 주로 서버에 대한 정보가 저장되어 있음

메소드	설명
getHeader(name)	헤더 이름 name 값을 가져옴
containsHeader(name)	헤더 이름 name이 HTTP 헤더에 포함되었는지 여부를 확인
setHeader(name, value)	헤더 이름 name에 String값 value를 설정
setIntHeader(name, value)	헤더 이름 name에 정수값 value를 설정
setDateHeader(name, date)	헤더 이름 name에 날짜/시간 date을 설정
addHeader(name, value)	헤더 이름 name에 String값 value를 추가
addIntHeader(name, value)	헤더 이름 name에 정수값 value를 추가
addDateHeader(name, date)	헤더 이름 name에 날짜/시간 date을 추가
addCookie(cookie)	쿠키를 추가

response 내장 객체

□ 응답 콘텐츠 관련 메소드

- response 내장 객체는 웹 브라우저로 응답하기 위해 MIME 유형, 문자 인코딩, 오류 메시지, 상태 코드 등을 설정하고 가져오는 응답 콘텐츠 관련 메소드를 제공

메소드	설명
setContentType(type)	웹브라우저에 응답할 MIME 유형을 설정
getContentType()	웹브라우저에 응답할 MIME 유형을 가져옴
setCharacterEncoding(charset)	웹브라우저에 응답할 문자 인코딩을 설정
getCharacterEncoding()	웹브라우저에 응답할 문자 인코딩을 가져옴
sendError(statusCode, msg)	웹브라우저에 응답할 오류 (코드 및 오류메시지)를 설정해서 보냄
setStatus(statusCode)	웹브라우저에 응답할 HTTP 코드를 설정

out 내장 객체

- **out (javax.servlet.jsp.jspWriter)**
- 웹 브라우저에 데이터를 전송하는 출력 스트림 객체
- JSP 컨테이너는 JSP 페이지에 사용되는 모든 표현문 태그와 HTML, 일반 텍스트 등을 out 내장 객체를 통해 웹 브라우저에 그대로 전달
- 스크립틀릿 태그에 사용하여 단순히 값을 출력하는 표현문 태그(<%= ...%>)와 같은 결과를 얻을 수 있음
- 복잡한 if-else 사용시 out 기본 객체 사용하면 편리

```
<%  
    if (grade > 10) {  
        out.println(gradeStringA);  
    } else if (grade > 5) {  
        out.println(gradeStringB);  
    }  
%>
```

out 내장 객체

□ 출력 및 버퍼 관련 out 메서드

메소드	설명
print(str)	설정된 str 값을 웹브라우저에 출력
println(str)	설정된 str 값을 웹브라우저에 줄바꿈과 함께 출력
newLine()	줄바꿈(\r\n 또는 \n)을 출력
getBufferSize()	현재 출력 버퍼의 크기를 가져옴
getRemaining()	현재 남아 있는 출력 버퍼의 크기를 가져옴
clear()	현재 출력버퍼에 저장되어 있는 내용을 웹브라우저에 전송하지 않고 비움. 만약 버퍼가 이미 플러시되었다면 IOException 발생
clearBuffer()	현재 출력버퍼에 저장되어 있는 내용을 웹브라우저에 전송하지 않고 비움. 만약 버퍼가 이미 플러시되었다면 IOException 발생안함
flush()	현재 출력 버퍼에 저장되어 있는 내용을 웹브라우저에 전송하고 비움
isAutoFlush()	출력 버퍼가 채워졌을 때의 처리를 결정. 자동으로 플러시하는 경우 true 반환, 그렇지 않은 경우 false 반환

session 내장 객체

- **session (javax.servlet.http.HttpSession)**
- 클라이언트와의 지속적인 연결 유지를 위한 세션 처리
- 세션은 하나의 애플리케이션 내에서만 유효함
- 두 개의 서로 다른 애플리케이션에서는 쿠키를 사용해야 함
- 세션은 단순한 쿠키 대용이 아님

session 내장 객체

□ session 메서드

메소드	설명
getID()	각 접속에 대한 세션 고유의 ID를 문자열 형태로 반환
getCreatingTime()	세션이 생성된 시간을 January 1, 1970 GMT.부터 long형 밀리세컨트 값으로 반환
getLastAccessedTime()	현재 세션으로 마지막 작업한 시간을 long형 밀리세컨트 값으로 반환
getMaxInactiveInterval()	세션 유지 시간을 초로 반환 세션 유효 시간을 줌
setMaxInactiveInterval(t)	세션 유효 시간을 t에 설정된 초 값으로 설정
invalidate()	현재 세션을 종료. 세션과 관련한 값을 모두 지움
getAttribute(attr)	문자열 attr로 설정된 세션 값을 java.lang.Object 형태로 반환
setAttribute(name,attr)	문자열 name으로 java.lang.Object attr을 설정

session 내장 객체

```
<body>
<form action= "session.jsp">
  이름: <input type= "text" name= "name"> <br>
  나이: <input type= "text" name= "age"> <br>
  <input type= "submit" value= "click">
</form>
</body>
```

index.jsp

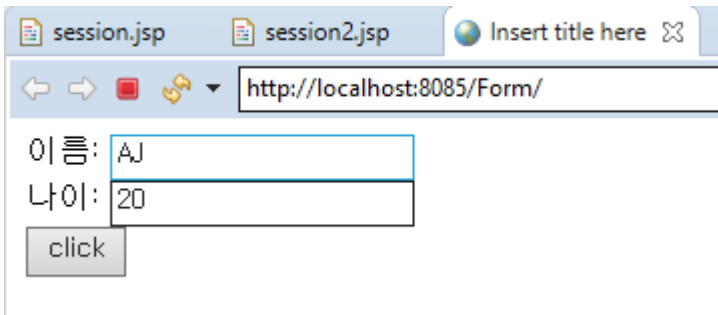
```
<body>
<%
String name = request.getParameter("name");
int age = Integer.parseInt(request.getParameter("age"));
out.println("Name: " + name + " Age: " + age);
session.setAttribute("name", name);
session.setAttribute("age", age);
%>
<a href= "session2.jsp">NextSession</a>
</body>
```

session.jsp

session 내장 객체

```
<body>
<%
String name = (String)session.getAttribute("name");
int age = (int)session.getAttribute("age");
out.println("In NextSession Name: " + name + " Age: " + age);
%>
</body>
```

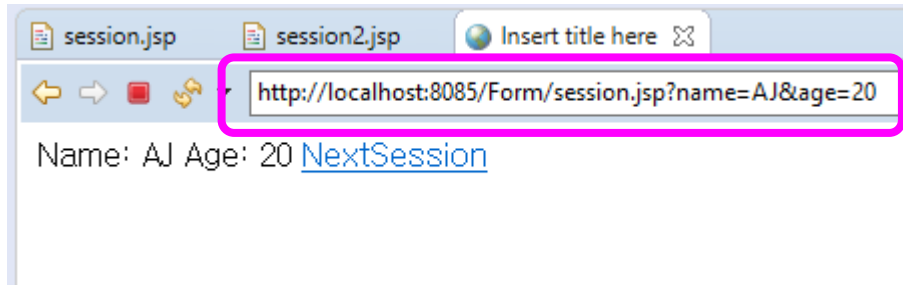
session2.jsp



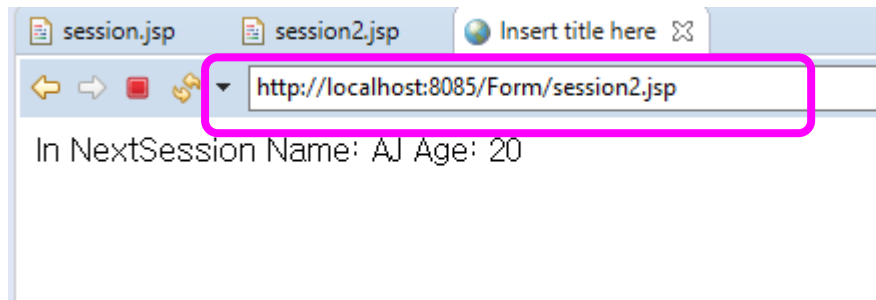
The screenshot shows a web browser window with two tabs: 'session.jsp' and 'session2.jsp'. The address bar displays 'http://localhost:8085/Form/'. The page content includes a form with two input fields: '이름:' (Name) containing 'AJ' and '나이:' (Age) containing '20'. Below the fields is a button labeled 'click'.

index.jsp

session 내장 객체



session.jsp



session2.jsp

config 내장 객체

- **config (javax.servlet.http.HttpServletConfig)**
- 애플리케이션에서 공유할 수 있는 초기화 변수 (e.g. paths, file) 접근
- web.xml 에 초기화 파라미터가 설정되어야 함.

config 내장 객체

□ config 메서드

메소드	설명
getInitParameterNames()	초기 파라미터 값의 설정 이름을 Enumeration 객체로 반환
getInitParameter(name)	문자열 name에 해당하는 초기화 파라미터 값을 반환
getServletContext()	현재 애플리케이션의 ServletContext 객체를 반환 내장 객체인 application과 동일한 객체 참조

config 내장 객체

```
<body>  
<a href="config.jsp">Click here to see config</a>
```

index.jsp

```
</body>  
<body>
```

```
<%  
    out.print("servletName: " + config.getServletName() + "<br>");  
    String website = config.getInitParameter("courseBlog");  
    out.print("Course Blog: " + website);  
%>
```

config.jsp

```
<!--  
<servlet>
```

```
    <servlet-name>welcomeServlet</servlet-name>
```

```
    <jsp-file>/config.jsp</jsp-file>
```

```
    <init-param>
```

```
        <param-name>courseBlog</param-name>
```

```
        <param-value>dis.dankook.ac.kr/lectures/</param-value>
```

```
    </init-param>
```

```
</servlet>
```

```
<servlet-mapping>
```

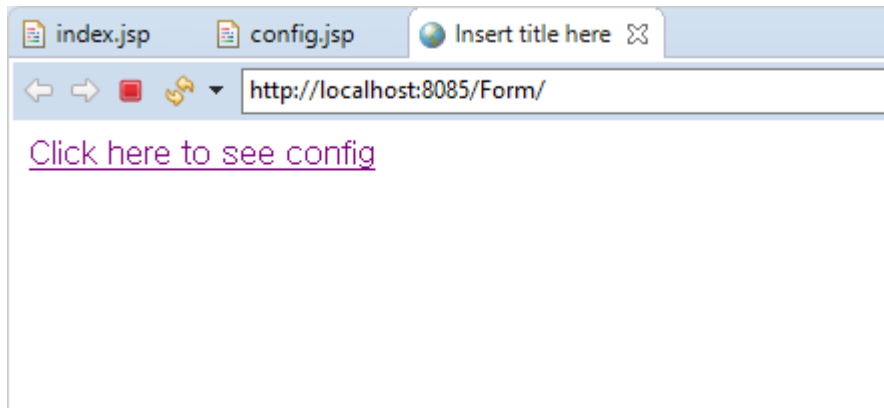
```
    <servlet-name>welcomeServlet</servlet-name>
```

```
    <url-pattern>/config.jsp</url-pattern>
```

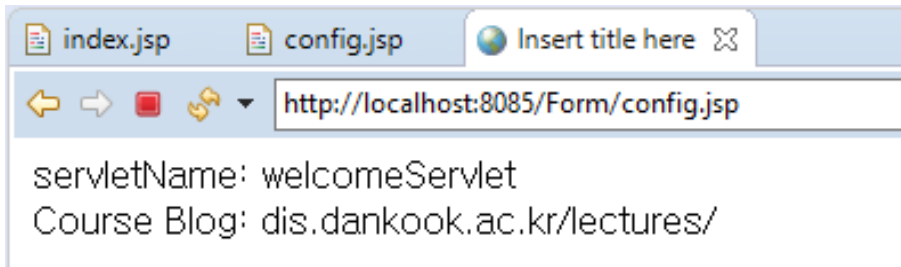
```
</servlet-mapping>
```

web.xml

config 내장 객체



index.jsp



config.jsp

application 내장 객체

- **application (javax.servlet.http.ServletContext)**
- JSP 페이지가 초기화될 때 생성되고 JSP가 소멸할 때 이 객체도 소멸
- 웹 애플리케이션(컨텍스트)에 관련된 정보를 제공
 - web.xml에서 초기화 매개 변수를 가져오는 데 사용
- 개발자를 위한 서버 정보
- 서버의 각종 자원 정보
- 로그 관련 정보
- application scope에서 속성 관련 정보(get, set, remove attribute)

application 내장 객체

- ▣ 개발자를 위한 서버 관련 정보 application 메서드

메소드	설명
getServerInfo()	JSP/서블릿 컨테이너 이름과 버전 반환
getMajorVersion()	컨테이너가 지원하는 서블릿 API의 Major 버전 정보 반환
getMinorVersion()	컨테이너가 지원하는 서블릿 API의 Minor 버전 정보 반환

application 내장 객체

▣ 서버 자원 정보 application 메서드

메소드	설명
getMimeType(filename)	문자열 filename에 지정된 파일에 대한 MIME type을 반환
getResource(path)	문자열 path에 지정된 자원을 URL 객체로 반환
getResourceAsStream(path)	문자열 path에 지정된 자원을 InputStream 객체로 반환
getRealPath(path)	문자열 path에 지정된 자원을 파일 시스템의 실제 경로로 반환
getContext(path)	문자열 path에 지정된 자원의 컨텍스트 정보를 반환
getResourceDispatcher(path)	문자열 path에 지정된 자원을 위한 request dispatcher를 생성

application 내장 객체

□ 표 로그 관련 application 메서드

메소드	설명
log(message)	문자열 message 내용을 로그 파일에 기록 로그 파일 위치는 컨테이너에 따라 다름
log(message,exception)	예외 상황과 관련한 정보를 포함하여 로그 파일에 기록

application 내장 객체

□ 속성 관련 application 메서드

메소드	설명
getAttribute(String name)	문자열 name에 해당하는 속성값이 있다면 Object 형태로 가져옴. 반환 값에 대한 적절한 형 변환이 필요
getAttributeNames()	현재 application 객체에 저장된 속성 이름을 Enumeration 형태로 가져옴
setAttribute (String name, Object value)	문자열 name 이름으로 Object 형 데이터를 저장 Object 형이므로 자바 클래스 형태로도 저장 가능
removeAttribute(String name)	문자열 name에 해당하는 속성을 삭제

application 내장 객체

```
<body>  
<a href="application.jsp">Click here to see application</a>  
</body>
```

index.jsp

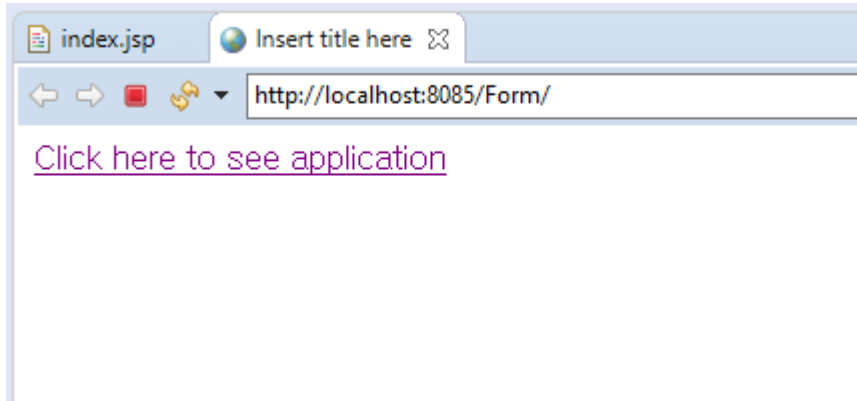
```
<body>  
<%  
    String appName = application.getInitParameter("appName");  
    out.print("App Name: " + appName);  
%>  
</body>
```

application.jsp

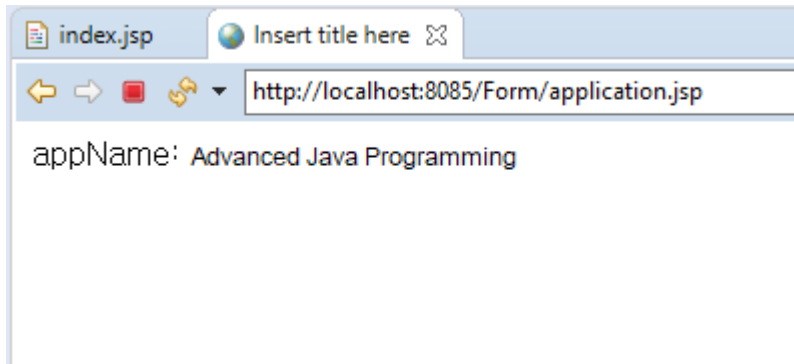
```
<context-param>  
    <param-name>appName</param-name>  
    <param-value>Advanced Java Programming</param-value>  
</context-param>
```

web.xml

application 내장 객체



index.jsp



application.jsp

page 내장 객체

- **page (javax.servlet.jsp.PageContext)**
- JSP 파일에서 자기자신(this 객체)을 참조할 때 사용

pageContext 내장 객체

- **pageContext (javax.servlet.jsp.PageContext)**
- include, forward 액션의 실제구현에 응용됨.
- 내장객체에 대한 프로그램적인 접근방법 제공
- page, request, session, application scope에서 속성 관련 정보 (get, set, remove attribute)

```
<jsp:forward page="HelloWorld.jsp" />  
→  
<%  
    out.flush();  
    pageContext.include("HelloWorld.jsp");  
%>
```

pageContext 내장 객체

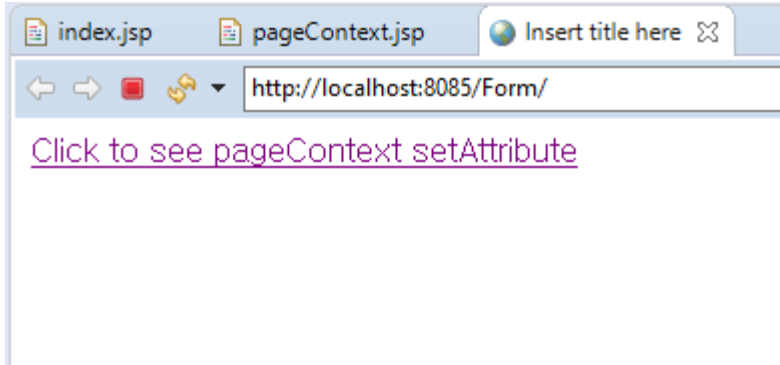
```
<body>
<%
String name = "AJ20";
int age = 20;
pageContext.setAttribute("name", name, PageContext.SESSION_SCOPE);
pageContext.setAttribute("age", age, PageContext.SESSION_SCOPE);
%>
<a href="pageContext.jsp">Click to see pageContext setAttribute</a>
</body>
```

index.jsp

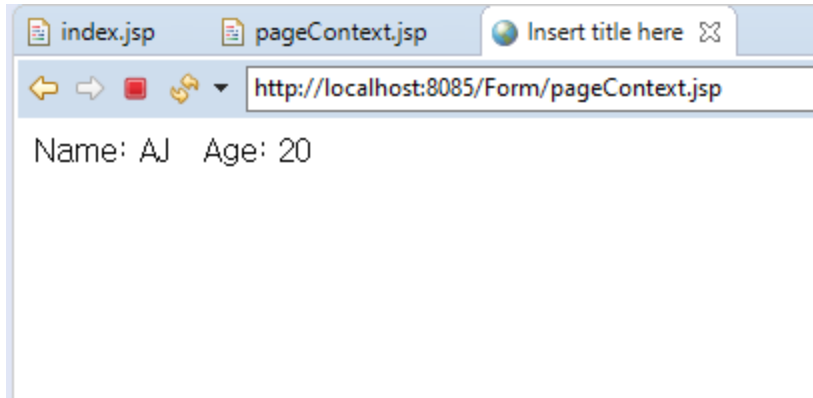
```
<body>
<%
String name = (String)pageContext.getAttribute("name",
PageContext.SESSION_SCOPE);
out.print("Name: " + name);
int age = (int)pageContext.getAttribute("age",
PageContext.SESSION_SCOPE);
out.print("Age: " + age);
%>
</body>
```

pageContext.jsp

pageContext 내장 객체



index.jsp



pageContext.jsp

exception 내장 객체

- **exception (java.lang.Throwable)**
- page 지시어에 isErrorPage="true" 로 지정된 에러 페이지에서 사용
- 현재 호출된 에러 정보 제공
- 속성 관련 exception 메소드

메소드	설명
getMessage()	문자열로 된 에러 메시지를 반환
printStackTrace()	표준 출력 스트림으로 스택 추적 정보를 출력
toString()	예외 클래스 이름과 함께 에러 메시지를 반환

exception 내장 객체

```
<body>  
<%  
out.print(10/0);  
%>  
</body>
```

index.jsp

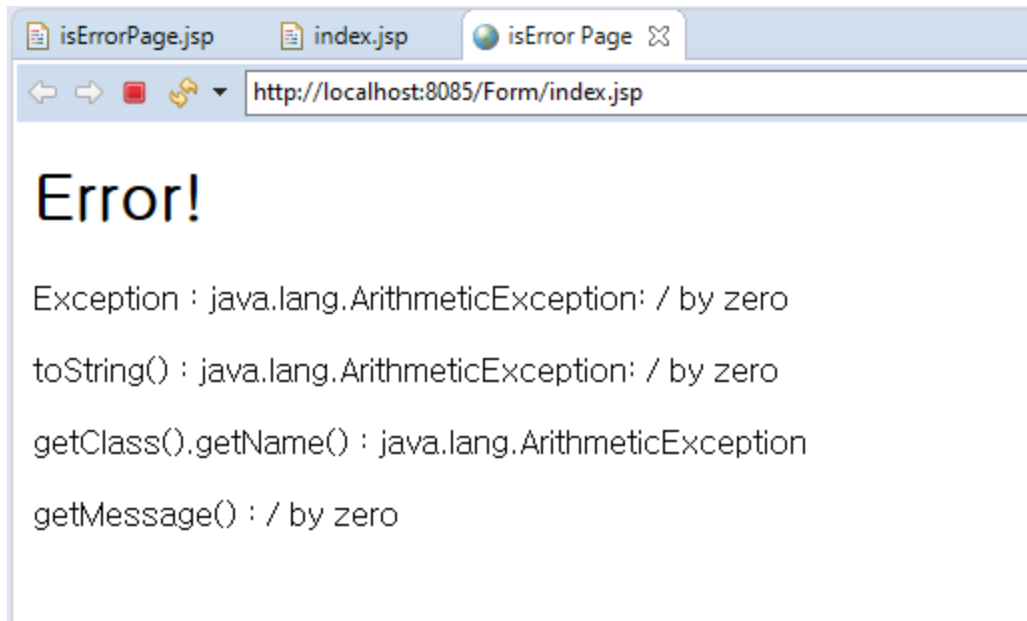
```
<%@ page isErrorPage= "true"%>  
<% response.setStatus(HttpServletResponse.SC_OK); %>  
<body>  
<h1>Error!</h1>  
<p>Exception : <%= exception %>  
<p>toString() : <%= exception.toString() %>  
<p>getClass().getName() : <%=exception.getClass().getName()%>  
<p>getMessage() : <%=exception.getMessage()%>  
</body>
```

isErrorPage.jsp

```
<error-page>  
  <exception-type>java.lang.Exception</exception-type>  
  <location>/isErrorPage.jsp</location>  
</error-page>
```

web.xml

exception 내장 객체



```
isErrorPage.jsp index.jsp isError Page
http://localhost:8085/Form/index.jsp

Error!

Exception : java.lang.ArithmeticException: / by zero
toString() : java.lang.ArithmeticException: / by zero
getClass().getName() : java.lang.ArithmeticException
getMessage() : / by zero
```

index.jsp

isErrorPage.jsp