04강 JSP 내장 객체와 액션 태그

JSP 내장 객체

- 내장 객체란 개발자가 직접 생성한 일이 없이 바로 사용 가능한 객체를 이야기한다.
- JSP가 서블릿으로 만들어 질 때 자동으로 생성해 주기 때문에 별도의 객체 생성이 필요없다.

- 예

```
<%
    out.print("Hello World!");
%>
```

out이 가르키는 객체

- JSP를 학습한다는 것은 내장 객체을 학습한다는 것과 동일한 의미이다.

JSP 내장 객체

- JSP가 서블릿으로 변환될 때 생성되는 서블릿 파일을 살펴 보면 자동으로 생성되는 내장 객체의 종류를 확인 할 수 있다..

```
public void _jspService(final javax.servlet.http.HttpServletRequest request, final
javax.servlet.http.HttpServletResponse response)
      throws java.io.IOException, javax.servlet.ServletException {

    final javax.servlet.jsp.PageContext pageContext;
    javax.servlet.http.HttpSession session = null;
    final javax.servlet.ServletContext application;
    final javax.servlet.ServletConfig config;
    javax.servlet.jsp.JspWriter out = null;
    final java.lang.Object page = this;
    javax.servlet.jsp.JspWriter _jspx_out = null;
    javax.servlet.jsp.PageContext _jspx_page_context = null;


    try {
      response.setContentType("text/html; charset=UTF-8");
      pageContext = _jspxFactory.getPageContext(this, request, response,
                                  null, true, 8192, true);
      _jspx_page_context = pageContext;
      application = pageContext.getServletContext();
      config = pageContext.getServletConfig();
      session = pageContext.getSession();
      out = pageContext.getOut();
```

1) request
2) response
3) pageContext
4) session
5) application
6) config
7) out
8) page

JSP 내장 객체

- 내장 객체의 분류

내장 객체의 분류	내장객체	형태
입출력 관련 객체	request	HttpServletRequest request
	response	HttpServletResponse response
	out	JspWriter out
서블릿 관련 객체	page	Object page
	config	ServletConfig config
외부 환경 정보를 제공하는 객체	session	HttpSession session
	application	ServletContext application
	pageContext	PageContext pageContext
예외 관련 객체	exception	에러페이지로 지정되면 생성

JSP 내장 객체

- out 내장 객체
  - JSP의 실행결과를 클라이언트의 브라우저로 출력할 때 가장 효과적인 객체이다.
  - 서블릿에서는 다음과 같은 객체를 직접 생성해야 했지만 JSP 내장 객체로 제공된다.

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

JSP 내장 객체

- request 내장 객체
  - 클라이언트와 웹 서버 사이의 '요청'과 관련된 정보는 request객체에 저장되어 관리된다.

메소드	설명
getContextpath()	JSP페이지가 속한 웹 애플리케이션의 컨텍스트 패스를 구한다.
getMethod()	요청 방식이 GET방식인지 POST방식인지 알려준다.
getRequestURL()	요청 URL을 구한다
getRequestURI()	요청 URL에서 쿼리 스트링 부분을 제외한 부분을 구한다.
getQueryString()	요청 URL 다음에 오는 쿼리 스트링 부분을 구한다
getSession(flag)	요청에 관련된 세션 객체를 구한다.
getRequestDispatcher(path)	지정 로컬 URL에 대한 RequestDispatcher 객체를 구한다.

JSP 내장 객체

- request 내장 객체
  - 클라이언트와 웹 서버 사이의 '요청'과 관련된 정보는 request객체에 저장되어 관리된다.

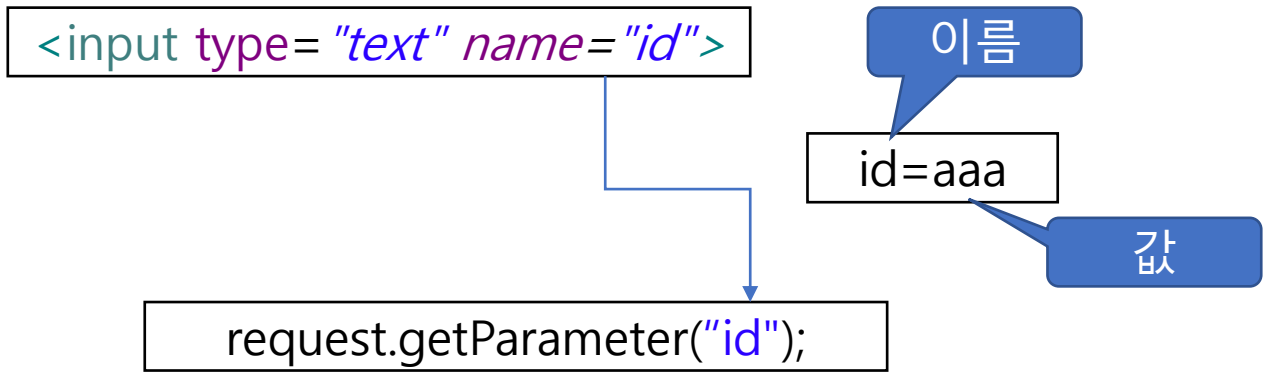
메소드	설명
getRemoteHost()	요청한 호스트의 완전한 이름을 구한다.
getRemoteAddr()	요청한 호스트의 네트워크 주소를 구한다.
getRemoteUser()	요청한 사용자의 이름을 구한다.
getSession()	세션 객체를 구한다.
getServerName()	서버의 이름을 구한다.
getProtocol()	사용중인 프로토콜을 구한다.

  - 예제1

JSP 내장 객체

- request 내장 객체
  - 요청 파라미터 관련 메소드 -예제2

메소드	설명
getParameter(String name)	지정한 이름의 파라미터를 구한다.
getParameterNames()	모든 파라미터의 이름을 구한다.
getParameterValues(String Name)	지정한 이름의 파라미터가 여러 개인 경우 사용하며 지정한 이름을 가진 파라미터의 모든 값을 배열로 구한다.



JSP 내장 객체

- response 내장 객체
  - 실행 결과를 브라우저로 되돌려 줄 때 사용하는 내장 객체
  - resoponse에서 가장 많이 사용되는 메소드는 sendRedirect이다. ->예제 3
  - 리다이렉트 는 웹 서버가 브라우저에게 지정된 페이지로 이동하도록 지시한다.
  - 예제 4- 로그인 => 로그인 성공 실패를 결정후 해당 페이지로 리다이렉트 한다.

JSP 내장 객체

- response 내장 객체
  - 웹 사이트에서 페이지를 이동시키는 방법은 두가지 이다.

1) 리다이렉트 방식
2) 포워드 방식

- 리다이렉트 방식은 response 객체의 sendRedirect 리다이렉트 방식이다.
- 리다이렉트 방식은 브라우저의 URL을 변경하여 페이지를 이동하는 방식으로 request,response 객체가 유지되지 않는다.

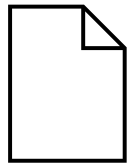
- 포워드 방식은 서버상으로 페이지가 이동되므로 브라우저에서는 페이지 이동을 알 수 없다.
- URL변경이 없고 기존 request,response 객체가 유지된다.

- 포워드 방식은 RequestDispatcher객체를 통해서 호출이 가능하다.

JSP 내장 객체

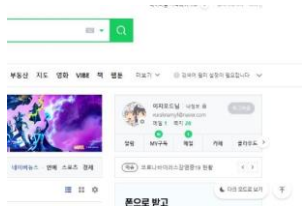
- response 내장 객체
  - 리다이렉트로 다른 페이지의 데이터를 전송하고 싶다면 쿼리스트링을 이용해야 한다.

로그인 실패



redirect.jsp

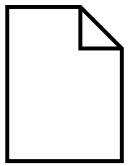
로그인 성공



response.sendRedirect("main.jsp?name="+name)

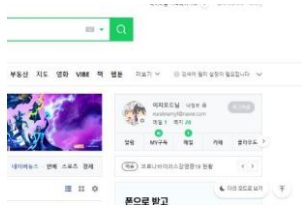
  - 그러나 포워드로 다른 페이지에 데이터를 전송하고 싶다면 request 객체가 유지되므로 해당 객체에 데이터를 담아서 보낼 수 있다. => 예제 5

로그인 실패



forward.jsp

로그인 성공



request.setAttribute("name",name);

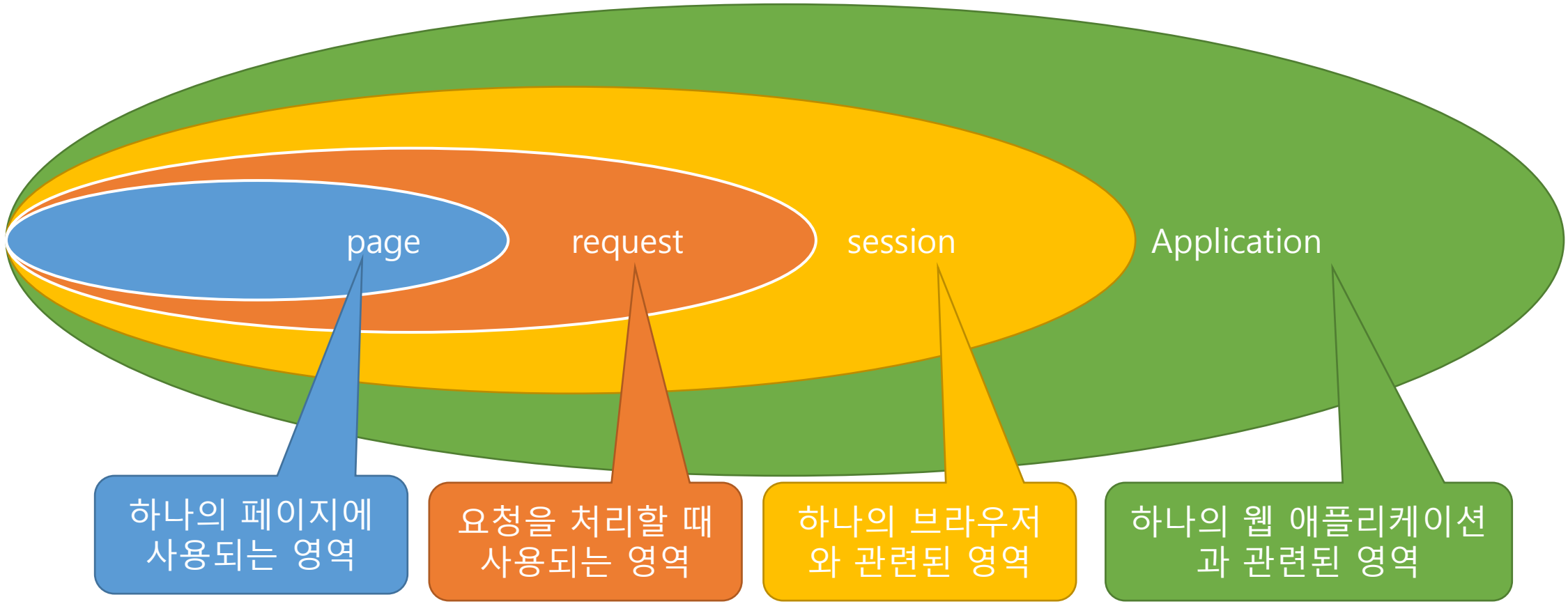
JSP 내장 객체

- 내장 객체의 영역
  - 내장 객체의 영역은 객체가 얼마 동안이나 살아 있는가를 보여주는 영역이다.
  - 영역은 총 4개로 page, request,session, application이 있다

영역	설명
page	하나의 JSP 페이지를 처리할 때 사용되는 영역
request	하나의 요청을 처리할 때 사용되는 영역
session	하나의 브라우저와 관련된 영역
application	하나의 웹 애플리케이션과 관련된 영역

JSP 내장 객체

- 내장 객체의 영역
  - 내장 객체의 영역은 객체가 얼마 동안이나 살아 있는가를 보여주는 영역이다.
  - 영역은 총 4개로 page, request,session, application이 있다



JSP 내장 객체

- 내장 객체의 영역 - page
  - page영역은 한번의 클라이언트 요청에 하나의 JSP 페이지를 범위로 갖는다.
  - 요청을 처리하는 JSP 페이지 하나당 새로운 page영역을 가지며 pageContext 객체 하나가 생성된다.

  - pageContext 객체에 정보가 저장되면 해당 페이지 내에서만 사용이 가능하다.

JSP 내장 객체

- 내장 객체의 영역 - request
  - request 영역은 한번의 클라이언트 요청과 관련이 된다.
  - 클라이언트가 페이지를 요청하면 요청한 페이지와 요청 받은 페이지 사이의 request 내장 객체에 정보를 저장 할 수 있다.
  - 브라우저가 응답 페이지를 받으면 request객체는 사라진다.

  - page와의 차이점은 page는 오직 하나의 JSP페이지만 포함하는데 반해
  request는 하나의 요청이 끝나기 까지 모든 JSP 페이지가 포함된다.

JSP 내장 객체

- 내장 객체의 영역 - session
  - session 영역은 웹 브라우저를 닫기 전까지 페이지를 이동하더라도 정보를 저장할 수 있는 객체
  - 보통 사용자 정보를 담아서 사용하는 session 객체는 로그인 인증 처리를 통해 회원 전용 페이지를 사용할 수 있게 해준다.

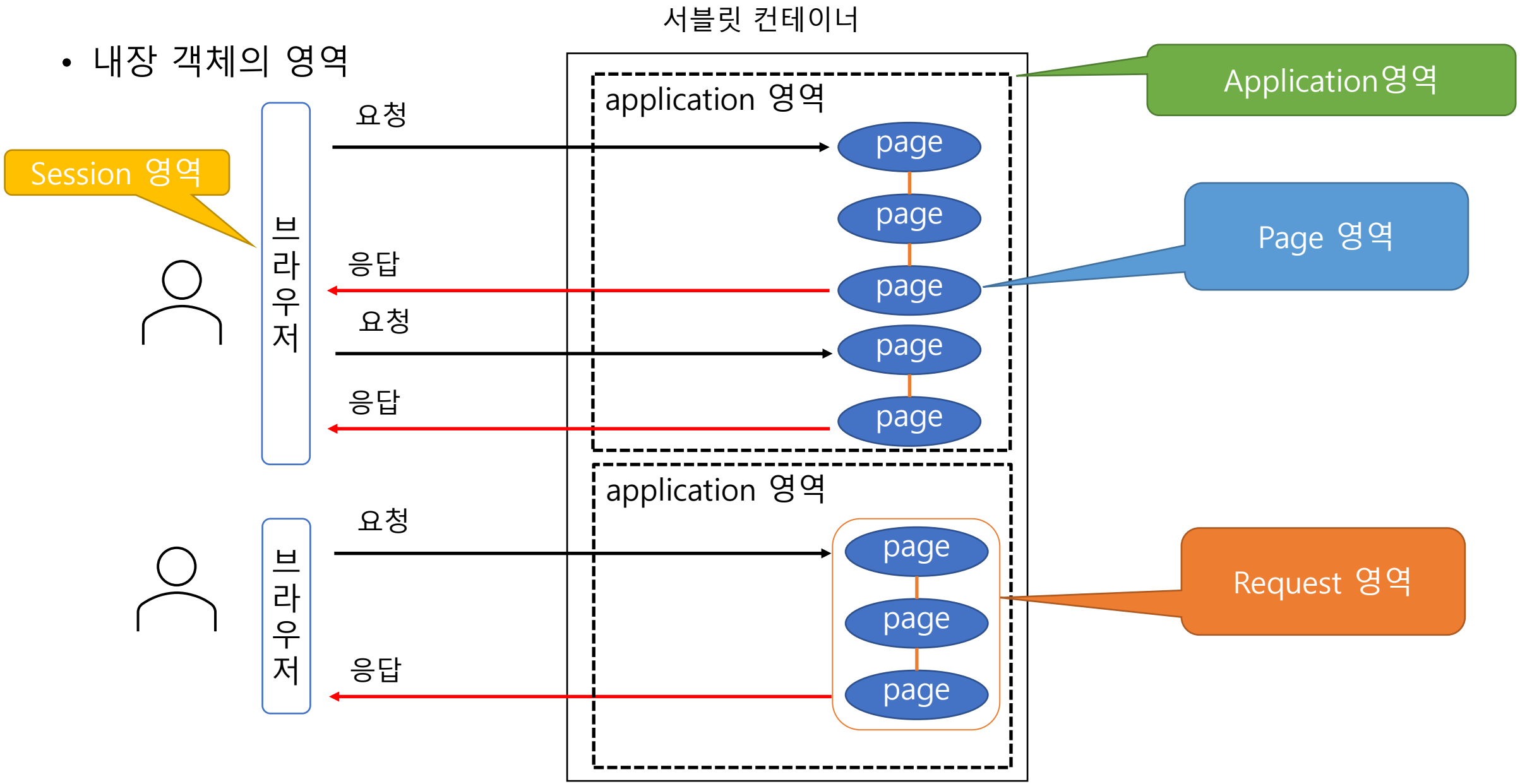
  - session은 하나의 브라우저와 관련된 영역이다.

JSP 내장 객체

- 내장 객체의 영역 - application
  - application영역은 하나의 웹 애플리케이션과 관련된 전체 영역을 포함한다.
  - 하나의 웹 애플리케이션에 속한 모든 페이지, 페이지에 대한 요청, 세션 전부 application 영역에 속한다.

JSP 내장 객체

- 내장 객체의 영역



JSP 내장 객체

- 내장 객체의 영역
  - 각각의 영역 객체에 정보를 저장하기 위한 메소드 : setAttribute(“속성 이름”,속성 값)
  - 각각의 영역 객체에 정보를 꺼내 오기 위한 메소드 : getAttribute(“속성 이름”)
  - 각각의 영역 객체에 모든 속성 이름을 가져오기 위한 메소드 : getAttributeNames()
  - 각각의 영역 객체에 지정된 속성을 삭제하는 메소드 : removeAttribute(“속성 이름”)

  - 예제 6

JSP 액션 태그

- 액션 태그는 JSP 페이지를 구성하는 요소중 하나이다.
- 보통 내장 객체를 통해서 자바 코드 형태로 작성할 수 있는 것을 태그로 표현할 수 있도록 한 것이다

- 주요 액션태그

태그의 종류	설명
<jsp:forward>	다른 사이트로 이동할 때 사용, 페이지의 흐름을 제어할 때 사용
<jsp:include>	정적 혹은 동적인 자원을 현재 페이지에 포함시킨다. 페이지 모듈화 할 때 사용
<jsp:param>	<jsp:forward>,<jsp:include>과 같이 사용되어 파라미터를 추가할 때 사용
<jsp:useBean>	빈(Bean)을 생성하고 사용하기 위한 환경을 정의하는 액션 태그
<jsp:setProperty>	액션은 빈에서 속성 값을 할당
<jsp:getProperty>	액션은 빈에서 속성 값을 얻어올 때 사용

- 이중 useBean, setProperty, getProperty는 자바 빈에서 학습하도록한다.

JSP 액션 태그

- jsp페이지에서 동일한 코드라도 자바코드로 기술하기 보다 태그를 활용하는 것이 보다 깔끔하고 가독성 높게 코딩할 수 있다.
- 액션태그는 기본 XML 문법을 따른다.(HTML과 유사, 닫기/ 반드시 표시)

JSP 액션 태그

- <jsp:forward>
- <jsp:forward>태그는 현재 페이지에서 URL로 지정된 특정 페이지로 넘어갈 때 사용하는 태그이다.

```
<jsp:forward page="이동하고자 하는 페이지주소"/>
```

위 코드는 서블릿에서 다음과 같이 변환된다.

```
RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("url");
dispatcher.forward(request,response);
```

예제 7

JSP 액션 태그

- <jsp:param>
- <jsp:forward>액션 태그로 이동하는 페이지로 정보를 추가하고 싶을 때 사용하는 태그가 <jsp:param>태그이다.

- 단독으로 쓰이지 않고 forward, include 내부에 포함되어서 사용된다.

```
<jsp:forward page="이동할 페이지">
        <jsp:param name="속성" value="속성 값" />
</jsp:forward>
```

예제 8

JSP 액션 태그

- <jsp:include>
- 웹페이지를 구축하다 보면 모든 페이지에 공통으로 사용되는 내용이 존재할 수 있다
- 이 경우 매번 같은 코드를 반복하는 불필요한 노력이 필요한데 이때 하나의 페이지 공토의 내용을 작성하고 모든 페이지에서 불러오는 방식을 사용하면 불필요한 노력을 감소시키며 동시에 유지보수에도 도움을 줄 수 있다(페이지의 모듈화)

- 상단에 보여야 할 로고나 메인 메뉴는 header.jsp에 하단에 보여야 할 저작권 표시등은 footer.jsp에 나누어 두고 이들 페이지를 현재 보여주어야 할 페이지에 동적으로 추가할 때 < jsp:include >액션 태그를 사용할 수 있다
- 사용예

```
<jsp:include page="포함하고자 하는 페이지 주소" flush="true또는 false"/>
```

  - flush는 현재 페이지의 출력 내용을 쌓인 버퍼를 어떻게 처리할 것인가 인데 true인 경우 현재 페이지 쌓인 버퍼를 페이지 모두 출력하고 이동할 페이지를 불러오는 방법인데 기본값을 false이다.

JSP 액션 태그

- <jsp: include>와 <%@ include file='url'%>는 비슷한 역할을 한다.
- 다만 <%@ include file='url'%> 의 경우 지정된 페이지가 현재 페이지에 합쳐져서 하나가 된 이후에 컴파일이 되는 형태이므로 지정된 페이지가 독립적인 형태가 아닌 현재 페이지의 일부분으로 구성된다. 그래서 전체 페이지의 변수가 공유된다.
- 그리고 <jsp: include>는 컴파일할 때 합쳐지는 것이 아니라 제어권이 지정한 페이지를 넘어갔다가 돌아오는 독립적으로 컴파일이 되는 두 페이지간의 연락을 교환하여 하나의 페이지인것 처럼 취급하는 방식이다. 그래서 전체 페이지의 변수가 공유되지 않는다.

- 예제 9