

## 5. 내장 객체와 액션 태그

# 목차

- ▶ 내장 객체
- ▶ 영역 객체와 속성
- ▶ 액션 태그

# 내장 객체

- ▶ JSP 페이지에서 선언하지 않고 사용할 수 있는 객체, 웹 컨테이너가 제공하는 고정된 이름의 객체
- ▶ JSP가 서블릿 형태로 자동 변환된 코드 내에 포함되어 있는 멤버변수, 메서드 매개변수 등의 각종 참조 변수(객체)를 말함
- ▶ `java.servlet` 패키지 아래 8개의 객체, 예외 처리를 위한 `java.lang` 패키지 아래 1개의 객체

# 내장 객체

참조 변수 이름(내장객체)	자바 클래스	주요 역할
request	javax.servlet.http.HttpServletRequest	HTML 폼 요소의 선택 값 등 사용자 입력 정보를 읽으려고 사용한다.
response	javax.servlet.http.HttpServletResponse	사용자 요청에 대한 응답을 처리하려고 사용한다.
pageContext	javax.servlet.jsp.PageContext	현재 JSP 실행에 대한 context 정보를 참조하려고 사용한다.
session	javax.servlet.http.HttpSession	클라이언트의 세션 정보를 처리하려고 사용한다.
application	javax.servlet.ServletContext	웹 서버의 애플리케이션 처리와 관련된 정보를 참조하려고 사용한다.
out	javax.servlet.jsp.JspWriter	사용자에게 전달하기 위한 output 스트림을 처리하려고 사용한다.
config	javax.servlet.ServletConfig	현재 JSP의 초기화 환경을 처리하려고 사용한다.
page	java.lang.Object	현재 JSP의 클래스 정보를 보려고 사용한다.
exception	java.lang.Throwable	예외 처리를 하려고 사용한다.

# 내장 객체

## ▶ 내장 객체의 4가지 분류

부류	java.lang	javax.servlet	javax.servlet.http	javax.servlet.jsp
JSP 페이지에 관련된 객체	page	config		
페이지 입출력에 관련된 객체			request, response	out
컨텍스트에 관련된 객체		application	session	pageContext
에러에 관련된 객체	exception			

# 내장 객체

## ▶ request 객체

- ▶ request는 사용자 요청과 관련된 기능을 제공하는 내장 객체로 `javax.servlet.http.HttpServletRequest` 클래스에 대한 참조 변수임
- ▶ 주로 클라이언트에서 서버로 전달되는 정보를 처리하기 위해 사용함
- ▶ 대표적으로 HTML 폼을 통해 입력된 값을 JSP에서 가져올 때 사용함

# 내장 객체

## ▶ request 주요 메소드

메서드	설명
getParameterNames()	현재 요청에 포함된 매개변수의 이름을 열거(Enumeration) 형태로 넘겨준다.
getParameter(name)	문자열 name과 이름이 같은 매개변수의 값을 가져온다.
getParameterValues(name)	문자열 name과 이름이 같은 매개변수의 값을 배열 형태로 가져온다. checkbox, multiple list 등에 주로 사용한다.
getCookies()	모든 쿠키 값을 javax.servlet.http.Cookie의 배열 형태로 가져온다.
getMethod()	현재 요청이 GET이나 POST 형태로 가져온다.
getSession()	현재 세션 객체를 가져온다.
getRemoteAddr()	클라이언트의 IP 주소를 알려준다.
getProtocol()	현재 서버의 프로토콜을 문자열 형태로 알려준다.
setCharacterEncoding()	현재 JSP로 전달되는 내용을 지정한 캐릭터셋으로 변환해준다. HTML 폼에서 한글 입력을 정상적으로 처리해주려면 반드시 필요하다.

# 내장 객체

## ▶ 요청 파라미터와 관련된 메소드

리턴 타입	메소드명	설명
String	getParameter(String name)	name이란 이름으로 지정된 파라미터에 할당된 값을 리턴한다. 지정된 이름의 파라미터가 없으면 null을 리턴한다.
String[]	getParameterValues(String name)	name이란 이름으로 지정된 파라미터의 모든 값을 String 배열로 리턴한다. 하나의 이름으로 여러 개의 값을 가질 수 있는 checkbox와 같은 태그를 사용했을 때에 주로 사용되며 하나의 이름에 하나의 값만 가지는 파라미터는 getParameter(String name) 메소드를 사용하는 것이 좋다.
Enumeration	getParameterNames()	요청에 포함된 모든 파라미터 이름을 java.util.Enumeration 객체로 리턴한다.



# 내장 객체

## ▶ 실습

### Request 예제입니다.

이름   
성별 남 ☐ 여 ☐  
취미 독서 ☐ 게임 ☐ TV시청 ☐ 축구 ☐ 기타 ☐

### Request 예제입니다.

이름   
성별 남 ☒ 여 ☐  
취미 독서 ☐ 게임 ☐ TV시청 ☒ 축구 ☒ 기타 ☐



### Request 예제입니다.

이름 : 홍길동  
성별 : 남자  
취미 : TV시청 축구

## ▶ 실습 – “requestTest1\_Form.jsp”

교재 p.180

```
9 <body>
0 <h1>Request 예제입니다.</h1>
1 <form action="requestTest1.jsp" method="post">
2 <table>
3 <tr>
4 <td>이름</td>
5 <td><input type="text" name="name"></td>
6 </tr>
7 <tr>
8 <td>성별</td>
9 <td>
0 남<input type="radio" name="gender" value="male">
1 여<input type="radio" name="gender" value="female">
2 </td>
3 </tr>
4 <tr>
5 <td>취미</td>
6 <td>독서<input type="checkbox" name="hobby" value="독서">
7 게임<input type="checkbox" name="hobby" value="게임">
8 TV시청<input type="checkbox" name="hobby" value="TV시청">
9 축구<input type="checkbox" name="hobby" value="축구">
0 기타<input type="checkbox" name="hobby" value="기타">
1 </td>
2 </tr>
3 <tr>
4 <td colspan="2"><input type="submit" value="전송"></td>
5 </tr>
6 </table>
7 </form>
8 </body>
```

▶ 실습 - "requestTest1.jsp"

▶ `<% request.setCharacterEncoding("UTF-8"); %>` 추가

[illegible]

# 내장 객체

## ▶ HTTP 헤더 정보와 관련된 메소드들

리턴 타입	메소드명	설명
String	getHeader(String headerName)	HTTP 요청 헤더에 headerName으로 지정된 이름으로 할당된 값을 리턴한다. headerName으로 지정된 이름이 없을 경우 null을 리턴한다.
Enumeration	getHeaders(String headerName)	headerName으로 지정된 이름으로 할당된 모든 값을 java.util.Enumeration 객체로 리턴한다.
Enumeration	getHeaderNames()	HTTP 요청 헤더에 포함된 모든 헤더 이름을 java.util.Enumeration 객체로 리턴한다.
int	getIntHeader(String headerName)	headerName 헤더의 값을 int 타입으로 리턴한다. 지정된 헤더값을 int로 변환할 수 없을 경우에는 NumberFormatException이 발생하고 headerName 헤더가 없을 경우에는 -1을 리턴한다.

## ▶ 실습

헤더 이름	헤더 값
accept	image/jpeg, application/x-ms-application, image/gif, application/xaml+xml, im excel, application/vnd.ms-powerpoint, application/msword, */*
accept- language	ko-KR
cache-control	no-cache
ua-cpu	AMD64
accept- encoding	gzip, deflate
user-agent	Mozilla/5.0 (Windows NT 6.2; Win64; x64; Trident/7.0; rv:11.0) like Gecko
host	localhost:8080
connection	Keep-Alive

▶ 실습 : "requestTest2.jsp"

교재 p.184

```
1 <table>
2     <tr>
3         <td>헤더이름</td>
4         <td>헤더값</td>
5     </tr>
6 <%
7     Enumeration e=request.getHeaderNames();
8     while(e.hasMoreElements()) {
9         String headerName=(String)e.nextElement();
10    %>
11    <tr>
12        <td><%=headerName %></td>
13        <td><%=request.getHeader(headerName) %></td>
14    <%}%>
15 </table>
```

# 내장 객체

## ▶ 세션 정보와 관련된 메소드들

리턴 타입	메소드명	설명
HttpSession	getSession()	요청한 클라이언트에 할당된 HttpSession 객체를 반환한다. 이전에 생성된 HttpSession 객체가 없으면 새로운 객체를 생성해 할당한다.
HttpSession	getSession(Boolean create)	create가 true일 경우 getSession() 메소드와 동일한 결과를 리턴하지만 create를 false로 지정하면 이전에 생성된 HttpSession 객체가 없을 경우 null을 리턴한다.
String	getRequestedSessionId()	요청한 클라이언트에 지정된 세션의 ID를 문자열로 리턴한다.
boolean	isRequestedSessionIdValid()	요청에 포함된 클라이언트의 세션 ID가 유효하면 true를, 아니면 false를 리턴한다.



# 내장 객체

## ▶ 쿠키, URL/URI, 요청 방식과 관련된 메소드들

리턴 타입	메소드명	설명
Cookie[]	getCookies( )	HTTP 요청 메시지의 헤더에 포함된 쿠키를 javax.servlet.http.Cookie 배열로 리턴한다.
String	getServerName( )	서버의 도메인명을 문자열로 리턴한다.
int	getServerPort( )	서버의 포트 번호를 int형으로 리턴한다.
StringBuffer	getRequestURL( )	요청 URL을 StringBuffer로 리턴한다.
String	getRequestURI( )	요청 URI를 문자열로 리턴한다.
String	getQueryString( )	요청에 사용된 쿼리 문장을 문자열로 리턴한다.
String	getRemoteHost( )	클라이언트의 호스트 이름을 문자열로 리턴한다.
String	getRemoteAddr( )	클라이언트의 IP 주소를 문자열로 리턴한다.
String	getProtocol( )	요청에 사용된 프로토콜 이름을 문자열로 리턴한다.
String	getMethod( )	요청에 사용된 요청 방식(GET, POST 등)을 문자열로 리턴한다.
String	getContextPath( )	해당 JSP 페이지의 컨텍스트 경로를 문자열로 리턴한다.



▶ 실습

쿠키, URL/URI, 요청방식에 관련된 정보 예제

쿠키정보	JSESSIONID(E412150FBF0966A73EEBA877FCF8DC18)
서버 도메인명	localhost
서버 포트번호	8080
요청 URL	http://localhost:8080/ch5/requestTest3.jsp
요청 URI	/ch5/requestTest3.jsp
요청 쿼리	null
클라이언트 호스트명	0:0:0:0:0:0:0:1
클라이언트 IP 주소	0:0:0:0:0:0:0:1
프로토콜	HTTP/1.1
요청방식	GET
컨텍스트 경로	/ch5

▶ 실습 : "requestTest3.jsp"

교재 p.187

```
<body>  
    <h3>쿠키, URL/URI, 요청방식에 관련된 정보 예제</h3>  
    <table border="1">  
        <tr>  
            <td>쿠키정보</td>  
  
            <%  
Cookie[] cookie=request.getCookies();  
if(cookie==null){  
%>  
                <td>쿠키가 존재하지 않습니다</td>  
  
            <%  
} else {  
    for(int i=0; i<cookie.length;i++){  
        %>  
        <td><%=cookie[i].getName()%>  
           (<%=cookie[i].getValue()%>) <br/>  
          &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~</td>  
        <%  
    }  
}  
%>  
        </tr>  
        <tr>  
            <td>서버 도메인명</td>  
            <td><%=request.getServerName() %>  
        </td>
```

```

<tr>
    <td>서버 포트번호</td>
    <td><%=request.getServerPort() %>
</td>
<tr>
    <td>요청 URL</td>
    <td><%=request.getRequestURL() %>
</td>
<tr>
    <td>요청 URI</td>
    <td><%=request.getRequestURI() %>
</td>
<tr>
    <td>요청 쿼리</td>
    <td><%=request.getQueryString() %>
</td>
<tr>
    <td>클라이언트 호스트명</td>
    <td><%=request.getRemoteHost() %>
</td>
<tr>
    <td>클라이언트 IP 주소</td>
    <td><%=request.getRemoteAddr() %>
</td>
<tr>
    <td>프로토콜</td>
    <td><%=request.getProtocol() %>
</td>

```

```

3      <tr>
4          <td>요청방식</td>|
5          <td><%=request.getMethod() %>
6      </td>
7      <tr>
8          <td>컨텍스트 경로</td>
9          <td><%=request.getContextPath() %>
10     </td>
11 </table>
12 </body>

```

# 내장 객체

- ▶ response 객체
  - ▶ 클라이언트의 요청에 대한 HTTP 응답을 나타내는 객체
  - ▶ javax.servlet.http.HttpServletResponse 인터페이스를 사용해 객체 생성

리턴 타입	메소드명	설명
없음	setHeader(String headerName, String headerValue)	응답에 포함될 헤더 정보에 headerName의 이름으로 headerValue 값을 설정해 추가한다.
없음	addCookie(Cookie cookie)	javax.servlet.http.Cookie 타입의 쿠키 객체를 응답 헤더에 추가한다. 쿠키에 대해서는 Chapter 8에서 자세히 다룬다.
없음	sendRedirect(String url)	지정된 URL로 요청을 재전송한다.
없음	setContentType(String type)	응답 페이지의 contentType을 설정한다.

# 내장 객체

- ▶ response 객체

- ▶ 실습

교재 p.191

- ▶ responseTest1.jsp 파일 실행 -> responseTest2.jsp 파일로 요청이 전송됨

현재 페이지는 **responseTest1.jsp**에 의해  
리다이렉트 된 페이지입니다.

## responseTest1.jsp

```
1 <%  
5 response.sendRedirect("responseTest2.jsp");  
5 %>  
7 <html>  
8 <head>  
9 <meta http-equiv="Content-Type" content="text/html">  
10 <title>Insert title here</title>  
11 </head>  
12 <body>  
13 <h3>responseTest1 파일 본문</h3>  
14 </body>  
15 </html>
```

## responseTest2.jsp

```
9 <body>  
10 <h3>  
11 현재 페이지는 responseTest1.jsp에 의해<br>  
12 리다이렉트 된 페이지입니다.  
13 </h3>  
14 </body>
```

# 내장 객체

- ▶ pageContext 객체
  - ▶ javax.servlet.jsp.PageContext 클래스를 상속해 웹 컨테이너가 구현한 객체로 JSP 페이지와 관련된 프로그램에서 다른 내장 객체를 얻어내거나 현재 페이지의 요청과 응답의 제어권을 다른 페이지로 넘겨주는데 사용



# 내장 객체

## ▶ pageContext 객체

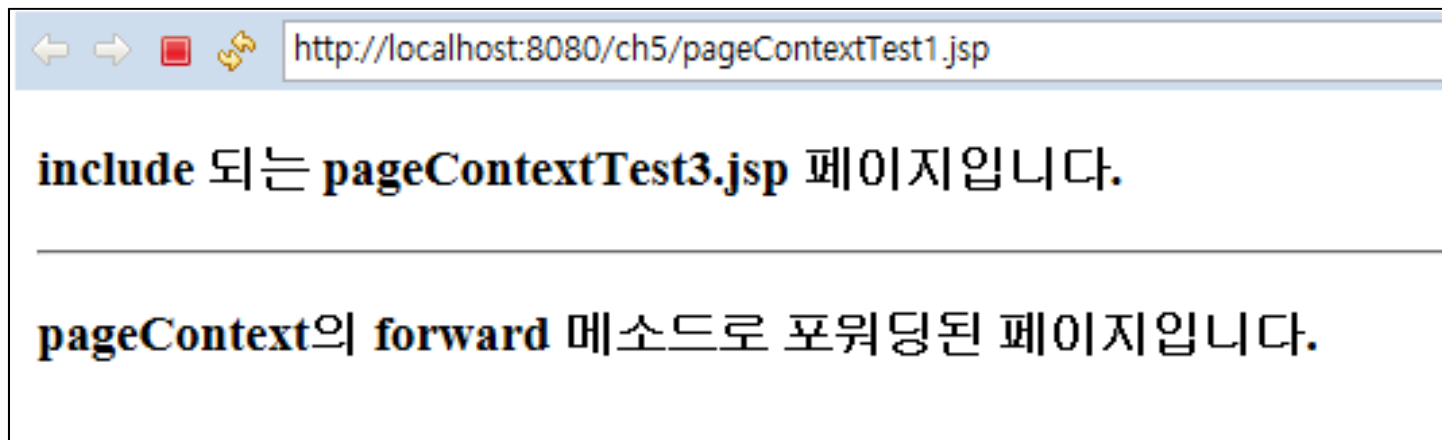
리턴 타입	메소드명	설명
ServletRequest	getRequest( )	클라이언트의 요청 정보를 담고 있는 객체를 리턴한다(request 내장 객체를 리턴한다).
ServletResponse	getResponse( )	요청에 대한 응답 객체를 리턴한다(response 내장 객체를 리턴한다).
JspWriter	getOut( )	응답 출력 스트림을 리턴한다(out 내장 객체를 리턴한다).
Object	getPage( )	서블릿 인스턴스 객체를 리턴한다(page 내장 객체를 리턴한다).
ServletConfig	getServletConfig( )	서블릿의 초기 설정 정보를 담고 있는 객체를 리턴한다(config 내장 객체를 리턴한다).
ServletContext	getServletContext( )	서블릿의 실행 환경 정보를 담고 있는 객체를 리턴한다(application 내장 객체를 리턴한다).
HttpSession	getSession( )	클라이언트의 세션 정보를 담고 있는 객체를 리턴한다(session 내장 객체를 리턴한다).
없음	forward(String url)	현재 페이지의 요청과 응답에 관한 제어권을 URL로 지정된 주소로 영구적으로 넘긴다. forward된 페이지의 요청 처리가 종료되면 응답도 종료된다.
없음	include(String url)	현재 페이지의 요청과 응답에 관한 제어권을 URL로 지정된 주소로 임시로 넘긴다. include된 페이지의 처리가 끝나면 제어권은 다시 원래의 페이지로 돌아온다. 따라서 include로 지정된 페이지의 내용을 원래 페이지에 삽입하는 효과를 가진다.

# 내장 객체

## ▶ 실습

교재 p.193

- ▶ pageContextTest1.jsp 파일 실행
  - ▶ forward 메소드를 사용하여 요청과 응답에 관한 제어권을 pageContextTest2.jsp로 영구적으로 넘김
  - ▶ sendRedirect 메소드와 달리 주소는 유지함
- ▶ pageContextTest2.jsp 파일은 본문의 내용과 include 메소드로 포함된 pageContextTest3.jsp 파일의 내용을 화면에 나타냄



## pageContextTest1.jsp

```
8 </head>
9 <% pageContext.forward("pageContextTest2.jsp"); %>
10 <body>
11 pageContextTest1 본문
12 </body>
```

## pageContextTest2.jsp

```
8 </head>
9 <% pageContext.include("pageContextTest3.jsp"); %>
10 <body>
11 <h3>pageContext의 forward 메소드로 포워딩된 페이지입니다.</h3>
12 </body>
```

## pageContextTest3.jsp

```
9 <body>
10 <h3>include 되는 pageContextTest3.jsp 페이지입니다.</h3>
11 <hr>
12 </body>
```

# 내장 객체

## ▶ session 객체

리턴 타입	메소드명	설명
String	getId( )	해당 세션의 세션 ID를 문자열로 리턴한다. 세션 ID는 session 객체 생성 시에 웹 컨테이너에 의해 자동으로 할당된다.
long	getCreationTime( )	1970년 1월 1일 00시 00분 00초(epoch)부터 해당 세션이 생성된 순간까지의 경과 시간을 밀리초로 계산하여 long형으로 리턴한다.
long	getLastAccessedTime( )	epoch로부터 해당 세션에 마지막으로 접근된 시간까지의 경과 시간을 밀리초로 계산하여 long형으로 리턴한다.
int	getMaxInactiveInterval( )	클라이언트의 요청이 없을 시 서버가 해당 세션을 유지하도록 지정된 시간을 초 단위의 정수로 리턴한다.
없음	invalidate( )	세션의 속성 값으로 저장된 모든 객체를 반납하여 해당 세션을 종료시킨다.
boolean	isNew( )	새로운 세션일 경우 true를 리턴하고 기존에 세션이 유지되고 있으면 false를 리턴한다.
없음	setMaxInactiveInterval(int seconds)	클라이언트의 요청이 없더라도 세션을 유지할 시간을 초 단위의 정수값으로 설정한다. 음수로 설정할 경우 세션은 무효화(invalidate)되지 않는다.
없음	forward(String url)	현재 페이지의 요청과 응답에 관한 제어권을 URL로 지정된 주소로 영구적으로 넘긴다. forward된 페이지의 요청처리가 종료되면 응답도 종료된다.
없음	include(String url)	현재 페이지의 요청과 응답에 관한 제어권을 URL로 지정된 주소로 임시로 넘긴다. include된 페이지의 처리가 끝나면 제어권은 다시 원래의 페이지로 돌아온다. 따라서 include로 지정된 페이지의 내용을 원래 페이지에 삽입하는 효과를 가진다.

## ▶ 실습 : sessionTest1.jsp

## 세션 테스트

isNew():true  
생성시간:1506508636899  
최종 접속 시간:1506508636899  
세션ID:D5AA59739624D02FEBA6C5AB7D6822C8

```
<% session.setMaxInactiveInterval(10); %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html
<title>Insert title here</title>
</head>
<body>
<h2>세션 테스트</h2>
isNew():<%=session.isNew()%><br>
생성시간:<%=session.getCreationTime()%><br>
최종 접속 시간:<%=session.getLastAccessedTime()%><br>
세션ID:<%=session.getId()%><br>
</body>
```

# 내장 객체

- ▶ application 객체
  - ▶ javax.servlet.ServletContext 클래스를 상속받아 웹 컨테이너가 구현하여 제공하는 객체
  - ▶ 해당 웹 애플리케이션의 실행 환경을 제공하는 서버의 정보와 서버측 자원에 대한 정보를 얻어내거나 해당 애플리케이션의 이벤트 로그를 다루는 메소드들 제공
  - ▶ 웹 애플리케이션(컨텍스트) 전체를 관리하는 객체
  - ▶ application 객체를 통해 각 서블릿이나 JSP에서 공유하려고 하는 각종 정보를 설정하고 참조할 수 있음

# 내장 객체

## ▶ application 객체

리턴 타입	메소드명	설명
int	getMajorVersion( )	Servlet API 스펙의 Major 버전을 int로 리턴한다.
int	getMinorVersion( )	Servlet API 스펙의 Minor 버전을 int로 리턴한다.
String	getServerInfo( )	서블릿/JSP 컨테이너의 이름과 버전을 문자열로 리턴한다.
String	getMimeType(String file)	서버에 존재하는 file이란 이름을 가진 파일의 MIME 타입을 문자열로 리턴한다.
java.net.URL	getResource(String path)	path로 지정된 경로의 자원을 URL 객체로 리턴한다. 자원이 존재하지 않으면 null을 리턴한다.
InputStream	getResourceAsStream(String path)	path로 지정된 경로의 자원을 InputStream 객체로 리턴한다. 자원이 존재하지 않으면 null을 리턴한다.
String	getRealPath(String path)	path로 지정된 경로의 자원을 서버의 실제 파일 시스템상의 경로로 바꾸어 문자열로 리턴한다.
없음	log(String msg)	문자열 msg를 서블릿 로그 파일에 기록한다.
없음	log(String msg, java.lang.Throwable exception)	문자열 msg와 예외의 StackTrace 정보를 로그 파일에 기록한다.

▶ 실습 : applicationTest1.jsp

교재 p.199

JSP 버전	3.1
컨테이너 정보	Apache Tomcat/8.0.36
웹 어플리케이션의 실제 파일시스템 경로	E:\D\jsp2017\metadata\plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps\ch5\

```
<table border="1">
  <tr>
    <td>JSP 버전</td>
    <td>
      <%=application.getMajorVersion() %>.<%=application.getMinorVersion() %>
    </td>
  </tr>
  <tr>
    <td>컨테이너 정보</td>
    <td><%=application.getServerInfo() %></td>
  </tr>
  <tr>
    <td>웹 어플리케이션의 실제 파일시스템 경로</td>
    <td><%=application.getRealPath("/") %></td>
  </tr>
</table>
```



# 내장 객체

## ▶ out 객체

- ▶ 응답 페이지를 만들기 위해 사용하는 출력 스트림 객체
- ▶ `javax.servlet.jsp.JspWriter` 클래스 타입으로 생성

메서드	설명
<code>getBufferSize()</code>	output buffer의 크기를 바이트로 알려준다.
<code>getRemaining()</code>	남아 있는 버퍼의 크기 중 사용 가능한 비율을 알려준다.
<code>clearBuffer()</code>	버퍼에 있는 콘텐츠를 모두 지운다.
<code>flush()</code>	버퍼를 비우고 output stream도 비운다.
<code>close()</code>	output stream을 닫고 버퍼를 비운다.
<code>println(content)</code>	content의 내용을 newline과 함께 출력한다.
<code>print(content)</code>	content의 내용을 출력한다.

▶ 실습 : outTest1.jsp

교재 p.201

```
<body>
<%=
int sum=0;
for(int i=1;i<=100;i++){
    sum+=i;
}
out.print("<h2>" + sum + "입니다.</h2>");
%>
<hr>
<h2>
<%=
int sum1=0;
for(int i=1;i<=100;i++){
    sum1+=i;
}
%>
<%=sum1 %>입니다.
</h2>
</body>
```

5050입니다.

5050입니다.

▶ 실습 : outTest3.jsp

교재 p.203

Out 테스트

autoFlush 여부	true
출력 버퍼의 크기	8192바이트
출력 버퍼의 남은 양	7761바이트

```
<body>
<h2>Out 테스트</h2>
<table border="1">
  <tr>
    <td>autoFlush 여부</td>
    <td><%=out.isAutoFlush() %></td>
  </tr>
  <tr>
    <td>출력 버퍼의 크기</td>
    <td><%=out.getBufferSize() %>바이트</td>
  </tr>
  <tr>
    <td>출력 버퍼의 남은 양</td>
    <td><%=out.getRemaining() %>바이트</td>
  </tr>
</table>
</body>
```