

CHAP 05.

기본 객체와 영역



CONTENTS

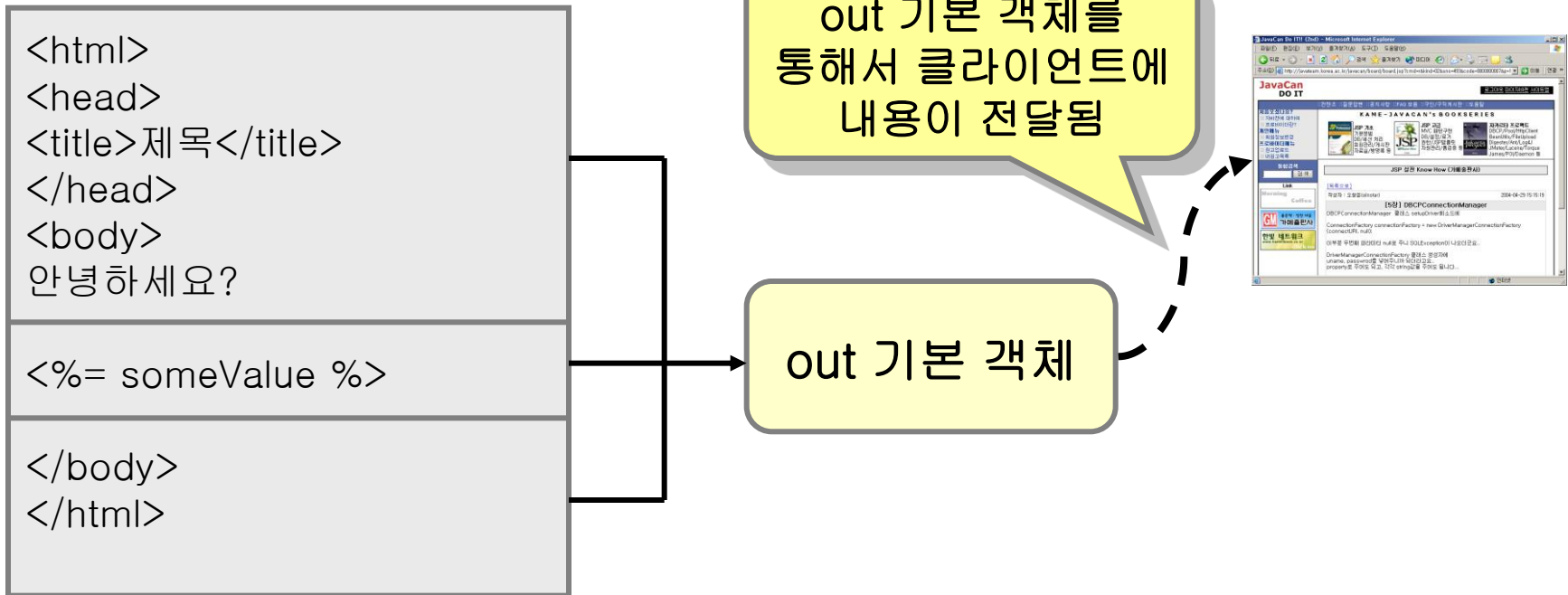
- 기본 객체
- out 기본 객체
- pageContext 기본 객체
- application 기본 객체
- JSP의 네 가지 영역(scope)
- 속성(attribute) 사용

JSP의 9가지 기본 객체

기본객체명	실제 타입	설명
request	javax.servlet.http.HttpServletRequest 또는 javax.servlet.ServletRequest	클라이언트의 요청 정보를 저장한다.
response	javax.servlet.http.HttpServletResponse 또는 javax.servlet.ServletResponse	응답 정보를 저장한다.
pageContext	javax.servlet.jsp.PageContext	JSP 페이지에 대한 정보를 저장한다.
session	javax.servlet.http.HttpSession	HTTP 세션 정보를 저장한다.
application	javax.servlet.ServletContext	웹 어플리케이션에 대한 정보를 저장한다.
out	javax.servlet.jsp.JspWriter	JSP 페이지가 생성하는 결과를 출력할 때 사용되는 출력 스트림이다.
config	javax.servlet.ServletConfig	JSP 페이지에 대한 설정 정보를 저장한다.
page	java.lang.Object	JSP 페이지를 구현한 자바 클래스 인스턴스이다.
exception	java.lang.Throwable	예외 객체. 에러 페이지에서만 사용된다.

out 기본 객체

"JSP 페이지가 생성하는 모든 내용은 out 기본 객체를 통해서 전송된다"

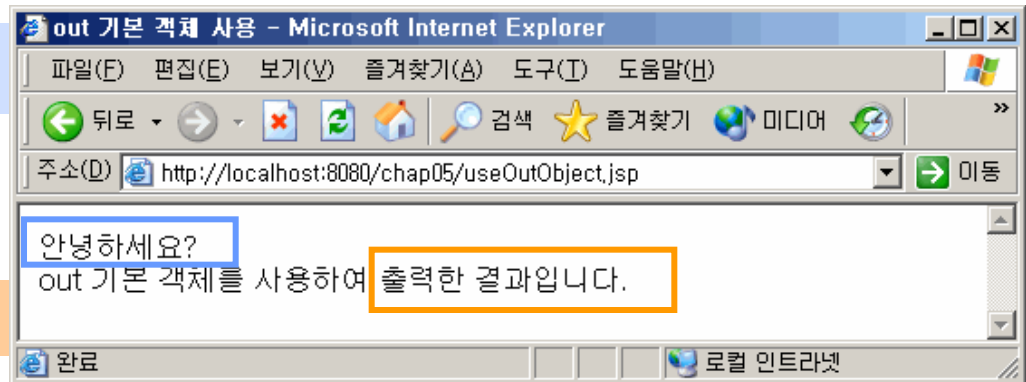


out 기본 객체의 출력 메소드

메소드	설명
print(출력값)	값을 출력한다.
println(출력값)	값을 출력한 뒤 마지막에 WrWn(또는 Wn)을 출력한다.
newLine()	WrWn(또는 Wn)을 출력한다.

Example

```
<%@ page contentType = "text/html; charset=euc-kr" %>
<html>
<head><title>out 기본 객체 사용</title></head>
<body>
<%
    out.println("안녕하세요?");
%>
<br>
out 기본 객체를 사용하여
<%
    out.println("출력한 결과입니다.");
%>
</body></html>
```



pageContext 기본 객체의 주요 기능

"pageContext는 하나의 JSP 페이지와 1대 1로 매핑되는 객체"

- 다른 기본 객체 구하기
- 속성 처리하기
- 페이지의 흐름 제어 (7장, 8장)
- 에러 데이터 구하기 (6장)

pageContext 기본 객체: 기본 객체 접근 메소드

메소드	리턴타입	설명
getRequest()	ServletRequest	request 기본 객체를 구한다.
getResponse()	ServletResponse	response 기본 객체를 구한다.
getSession()	HttpSession	session 기본 객체를 구한다.
getServletContext()	ServletContext	application 기본 객체를 구한다.
getServletConfig()	ServletConfig	config 기본 객체를 구한다.
getOut()	JspWriter	out 기본 객체를 구한다.
getException()	Exception	exception 기본 객체를 구한다.
getPage()	Object	page 기본 객체를 구한다.

application 기본 객체

- 웹 어플리케이션과 관련된 기본 객체이다.
- 하나의 웹 어플리케이션에 포함된 모든 JSP 페이지는 하나의 application 기본 객체를 공유
- 웹 어플리케이션 전반에 걸쳐서 사용되는 정보를 저장

application 기본 객체: 초기화 파라미터 읽기

[어플리케이션 초기화 파라미터를 읽어올 때 사용되는 메소드]

메소드	리턴타입	설명
<code>getInitParameter(String name)</code>	String	이름이 name인 웹 어플리케이션 초기화 파라미터의 값을 읽어온다. 존재하지 않을 경우 null을 리턴한다.
<code>getInitParameterNames()</code>	Enumeration	웹 어플리케이션 초기화 파라미터의 이름 목록을 리턴한다.

web.xml 파일에서 설정한 어플리케이션 파라미터를 application 기본 객체를 통해서 사용한다.

```
<web.xml>
...
<context-param>
  <description>설명</description>
  <param-name>port</param-name>
  <param-value>2005</param-value>
</context-param>
...
</web.xml>
```

```
<%@ page contentType = "..." %>
...
<%= application.getInitParameter("port") %>
...
```

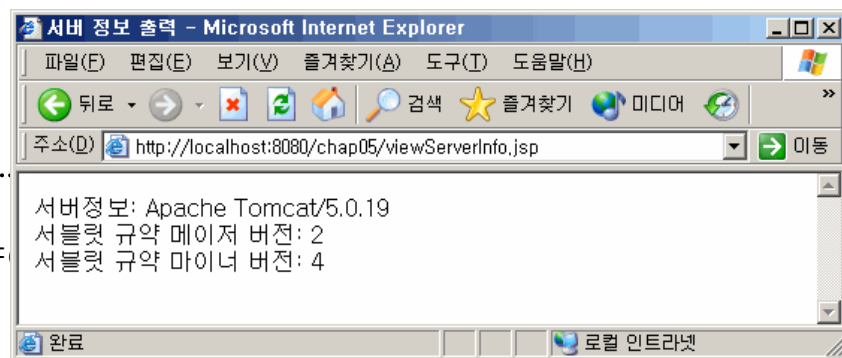
application 기본 객체: 서버 정보 읽기

[어플리케이션 초기화 파라미터를 읽어올 때 사용되는 메소드]

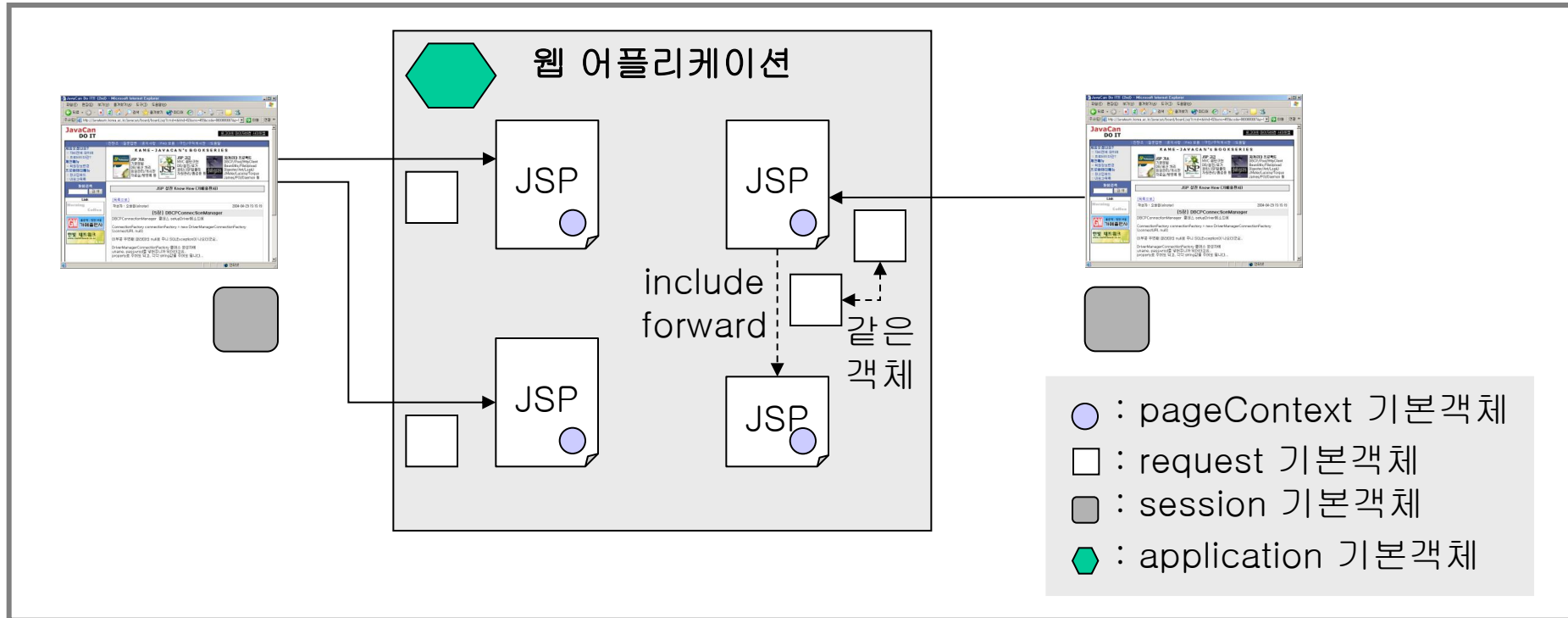
메소드	리턴타입	설명
getServerInfo()	String	서버 정보를 구한다.
getMajorVersion()	String	서버가 지원하는 서블릿 규약의 메이저 버전을 리턴한다. 버전의 정수부분을 리턴한다.
getMinorVersion()	String	서버가 지원하는 서블릿 규약의 마이너 버전을 리턴한다. 버전의 소수 부분을 리턴한다.

Example

```
<%@ page contentType = "text/html; charset=  
<html>  
<head><title>서버 정보 출력</title></head>  
<body>  
서버정보: <%= application.getServerInfo() %><br>  
서블릿 규약 메이저 버전: <%= application.getMajorVersion() %> <br>  
서블릿 규약 마이너 버전: <%= application.getMinorVersion() %>  
</body></html>
```



JSP의 네 가지 영역



- PAGE 영역 – pageContext 기본 객체
- REQUEST 영역 – request 기본 객체
- SESSION 영역 – session 기본 객체
- APPLICATION 영역 – application 기본 객체

JSP 기본 객체의 속성

- 기본 객체 `pageContext`, `request`, `session`, `application`은 속성을 갖는다.
- 속성은 각각의 기본 객체가 존재하는 동안에 사용 가능
- JSP 페이지 사이에서 정보를 주고 받거나 공유하기 위한 목적으로 사용
- 속성은 <속성이름, 값>의 형태를 가짐
- 서로 다른 이름을 가진 속성을 여러 개 포함할 수 있다.

[네 기본 객체의 속성 관련 메소드]

메소드	리턴타입	설명
<code>setAttribute(String name, Object value)</code>	<code>void</code>	이름이 <code>name</code> 인 속성의 값을 <code>value</code> 로 지정한다.
<code>getAttribute(String name)</code>	<code>Object</code>	이름이 <code>name</code> 인 속성의 값을 구한다. 지정한 이름의 속성이 존재하지 않을 경우 <code>null</code> 을 리턴한다.
<code>removeAttribute(String name)</code>	<code>void</code>	이름이 <code>name</code> 인 속성을 삭제한다.
<code>getAttributeNames()</code>	<code>java.util.Enumeration</code>	속성의 이름 목록을 구한다. (<code>pageContext</code> 기본 객체는 이 메소드를 제공하지 않는다.)

☞ 속성의 값 타입은 `Object`이므로 값을 사용할 때에는 알맞은 형변환이 필요

속성의 활용

기본 객체	영역	쓰임새
pageContext	PAGE	(한번의 요청을 처리하는) 하나의 JSP 페이지 내에서 공유될 값을 저장한다. 주로 커스텀 태그에서 새로운 변수를 추가할 때 사용된다.
request	REQUEST	한번의 요청을 처리하는 데 사용되는 모든 JSP 페이지에서 공유될 값을 저장한다. 주로 하나의 요청을 처리하는 데 사용되는 JSP 페이지 사이에서 정보를 전달하기 위해 사용된다.
session	SESSION	한 사용자와 관련된 정보를 JSP 들이 공유하기 위해서 사용된다. 사용자의 로그인 정보와 같은 것들을 저장한다.
application	APPLICATION	모든 사용자와 관련해서 공유할 정보를 저장한다. 임시 디렉토리 경로와 같은 웹 어플리케이션의 설정 정보를 주로 저장한다.