

내장객체

**JSP**



# JSP 내장객체

- ❖ JSP 내장 객체(기본 객체)
- ❖ JSP컨테이너에서 제공해주는 객체(9가지)
- ❖ session, exception은 page지시어로 핸들링 가능(그 외 자동 생성)

분류	내장 객체
입출력 관련 객체	request response out
서블릿 관련 객체	page config
외부 환경 정보를 제공하는 객체	session application pageContext
예외 관련 객체	exception

# request 내장객체(EX01.jsp)

❖ 클라이언트와 서버 사이의 요청에 관련된 정보를 관리하는 객체

메서드	설명
getContextPath()	웹 애플리케이션의 컨텍스트 패스를 구한다.
getMethod( )	요청방식이 GET 방식인지 POST 방식인지를 알려준다.
getRequestURL( )	요청 URL을 구한다.
getRequestURI( )	요청 URL에서 쿼리 스트링을 제외한 부분을 구한다.
getQueryString( )	요청 URL 다음에 오는 쿼리 스트링을 구한다.
getSession(flag)	요청 관련된 세션 객체를 구한다.
getRequestDispatcher(path)	지정 로컬 URL에 대한 RequestDispatcher 객체를 구한다.
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( )	사용 중인 프로토콜을 알려준다.

# request 파라미터 관련 메서드

## ❖ 실습

메서드	설명
getParameter(String name)	지정한 이름의 파라미터를 구한다. 지정한 이름의 파라미터가 여러 개 있을 경우에는 첫 번째 파라미터의 값을 구한다.
getParameterNames( )	모든 파라미터의 이름을 구한다.
getParameterValues(String name)	지정한 이름의 파라미터가 여러 개 있을 경우 사용하며 지정한 이름을 가진 파라미터의 모든 값을 String[]으로 구한다.
getParameterMap( )	웹 브라우저가 전송한 파라미터의 맵을 구한다.

# request 헤더 관련 메서드

## ❖ 실습

메서드	설명
<code>getHeader(String name)</code>	지정한 이름의 헤더 값을 구한다.
<code>getHeaders(String name )</code>	지정한 이름의 헤더 목록을 구한다.
<code>getHeaderNames( )</code>	모든 헤더의 이름을 구한다.
<code>getIntHeader(String name)</code>	지정한 헤더의 값을 정수 값으로 읽어온다.
<code>getDateHeader(String name)</code>	지정한 헤더의 값을 시간 값으로 읽어온다. 1970년 1월1일 기준 1/1000초 단위

# response 내장 객체

## ❖ 클라이언트에게 응답할 내용을 처리하는 객체

- 요청에 대한 실행 결과를 클라이언트 측의 브라우저로 되돌려주는 기능
- `sendRedirect()` 메서드를 사용하여 페이지를 이동 시키는 기능을 자주 사용

메서드	설명
<code>setContentType(String type)</code>	문자열 형태의 MIME type으로 <code>contentType</code> 을 설정한다.
<code>setHeader(String name, String value)</code>	헤더를 Name이라는 이름에 value의 값으로 세팅한다.
<code>setDateHeader(String name, String value)</code>	헤더를 name이라는 이름에 date에 설정된 ms시간 값으로 세팅
<code>sendError(int status, String message)</code>	에러코드를 세팅하고 메시지를 보냄
<code>sendRedirect(String URL)</code>	클라이언트 요청을 URL로 보냄

# pageContext 내장 객체

❖ JSP 페이지에서 다른 기본객체에 접근할 수 있는 메서드를 제공한다

메서드	설명
getRequest( )	Request 내장 객체를 구한다.
getResponse( )	Response 내장 객체를 구한다.
getSession( )	Session 내장 객체를 구한다.
getServletContext( )	Application 내장 객체를 구한다.
getServletConfig( )	Config 내장 객체를 구한다.
getOut( )	Out 내장 객체를 구한다.
getException( )	Exception 내장 객체를 구한다.
getPage( )	Page 내장 객체를 구한다.

# Application 내장 객체

- ❖ JSP페이지의 서블릿 컨텍스트
- ❖ 웹 어플리케이션당 하나씩 존재

메서드	설명
<code>getInitParameter(String name)</code>	웹 어플리케이션의 초기화 파라미터에서 Name이라는 이름의 파라미터 값을 읽어 온다. 값이 존재하지 않으면 null값을 반환
<code>getInitParameterNames()</code>	웹 어플리케이션의 초기화 파라미터 이름목록 반환
<code>getServerInfo()</code>	서버의 정보를 반환
<code>getMajorVersion()</code>	서버가 지원하는 서블릿 규약의 메이저 버전을 반환(버전의 정수부분)
<code>getMinorVersion()</code>	서버가 지원하는 서블릿 규약의 마이너 버전을 반환(버전의 소수부분)
<code>log(String msg)</code>	로그 메시지 msg를 기록한다.
<code>log(String msg, Throwable throwable)</code>	로그 메시지 msg를 기록한다. 예외정보도 함께 기록
<code>getReadPath(String path)</code>	웹 어플리케이션 내에서 지정한 경로에 해당하는 자원의 시스템상의 경자원로를 반환
<code>getResource(String path)</code>	웹 어플리케이션 내에서 지정한 경로에 해당하는 자원에 접근할 수 있는 URL객체 반환
<code>getReasourceAsStream(String path)</code>	웹 어플리케이션 내에서 지정한 경로에 해당하는 자원에서 데이터를 읽을 수 있는 InputStream을 반환



# Application 내장 객체

## ❖ 초기화 파라미터 불러오기

- web.xml 에 초기화 파라미터 값 입력

```
<context-param>
```

```
  <param-name>파라미터1</param-name>
```

```
  <param-value>파라미터 값1</param-value>
```

```
</context-param>
```

```
<context-param>
```

```
  <param-name>파라미터2</param-name>
```

```
  <param-value>파라미터 값2</param-value>
```

```
</context-param>
```

## ❖ 서버 정보 불러오기

# Out 내장 객체

❖ JSP페이지의 출력을 위한 객체

메서드	설명
getBufferSize()	버퍼의 크기를 반환
getRemaining()	남은 버퍼의 크기를 반환
clear()	버퍼를 비운다.(비어있었다면 IOException발생)
clearBuffer()	버퍼를 비운다.(IOException 발생 안 함)
flush()	버퍼를 내보낸다.
isAutoFlush()	버퍼가 다 찼을 때 자동으로 flush할지 결정(true/false)

# 그 외 내장 객체

## ❖ Session

- 세션을 저장하는 객체

## ❖ Config

- JSP페이지에서 사용하는 초기 파라미터를 저장하고 있는 객체

## ❖ Page

- jsp페이지 자체를 의미(JAVA class의 this 같은 의미)

## ❖ Exception

- JSP페이지에서 발생한 에러(예외)를 처리할 때 사용하는 객체
- Page Directive에서 isErrorPage=true가 설정 되어 있어야 사용가능

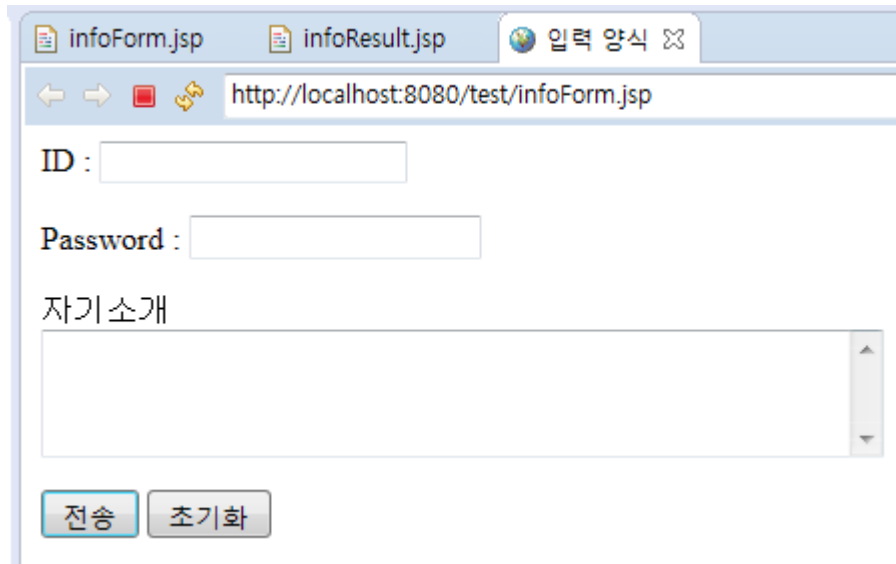
# 속성(Attribute)과 범위(Scope)

- `pageContext`, `request`, `session`, `application` 객체에 새로운 속성을 추가한 후, `secondPage.jsp`로 포워딩한다.
  - 이 페이지에서 설정한 속성 값은 브라우저에 출력할 수 없기 때문에 4개의 값이 콘솔 창에 출력된다.
- 
- `firstPage.jsp`에서 설정한 속성 값을 브라우저에 출력한다.
  - 페이지 속성은 제외한 3개의 값이 출력된다.
- 
- `secondPage.jsp`의 [또 다른 페이지]를 클릭하여 리다이렉트 방식으로 요청되는 페이지로 `firstPage.jsp`에서 설정한 속성 값을 브라우저에 출력한다.
  - 페이지 속성과 요청 속성이 제외된 2개의 값이 출력된다.

# 예제[1]

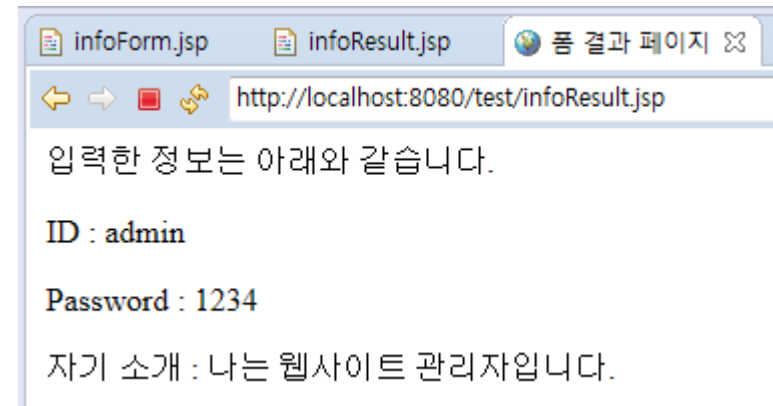
- 2개의 페이지를 작성하여 아래의 그림과 같이 출력되도록 하세요.
- 결과값은 request.getParameter()를 사용하세요.

## <infoForm.jsp : 입력 양식 페이지>



A screenshot of a web browser displaying the 'infoForm.jsp' page. The browser's address bar shows 'http://localhost:8080/test/infoForm.jsp'. The page contains three input fields: 'ID :', 'Password :', and a larger text area labeled '자기소개' (Self-introduction). At the bottom, there are two buttons: '전송' (Submit) and '초기화' (Reset).

## <infoResult.jsp : 결과 페이지>



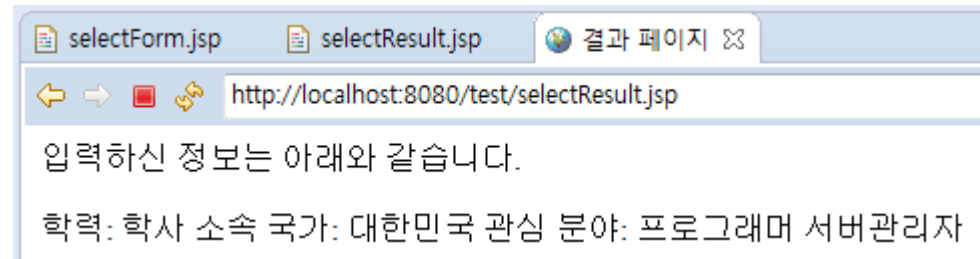
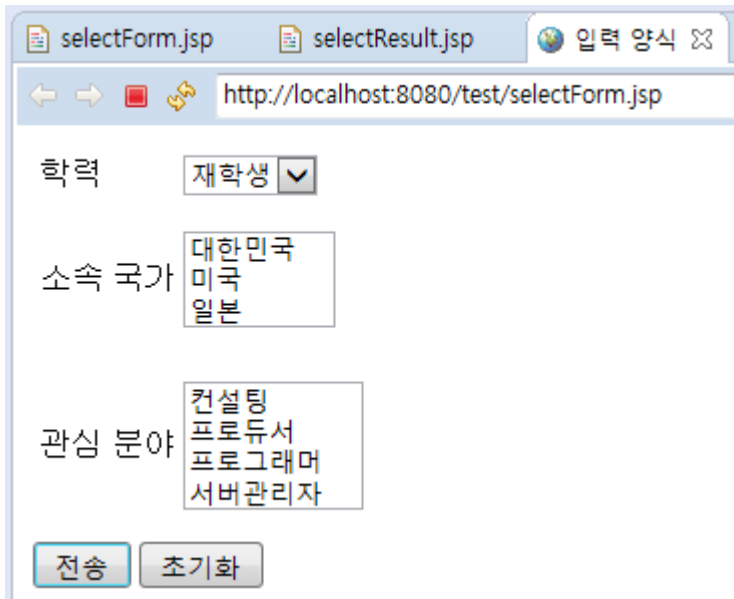
A screenshot of a web browser displaying the 'infoResult.jsp' page. The browser's address bar shows 'http://localhost:8080/test/infoResult.jsp'. The page displays the submitted information: '입력한 정보는 아래와 같습니다.' (The entered information is as follows.), followed by 'ID : admin', 'Password : 1234', and '자기 소개 : 나는 웹사이트 관리자입니다.' (Self-introduction : I am a website administrator.).

## 예제[2]

- 2개의 페이지를 작성하여 아래의 그림과 같이 출력되도록 하세요.
- 결과값은 request.getParameter()와 request.getParameterValues(여러 개의 값)를 사용하세요.
- 입력 페이지에서 관심분야는 다중 선택(mutiple)이 가능하도록 하고, 결과 페이지에서 관심분야를 출력할 때는 배열 변수와 for문을 사용하세요.

<selectForm.jsp : 입력 양식 페이지>

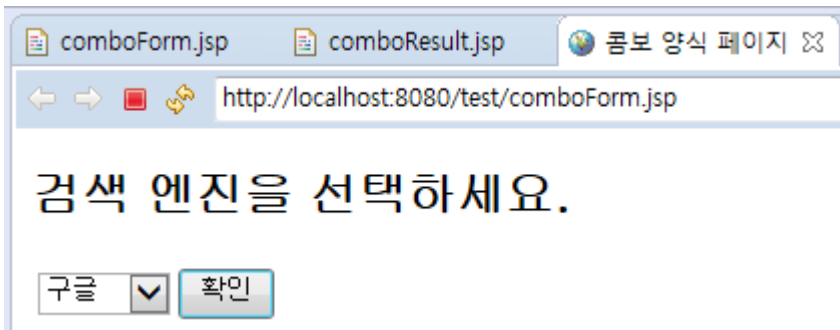
<selectResult.jsp : 결과 페이지>



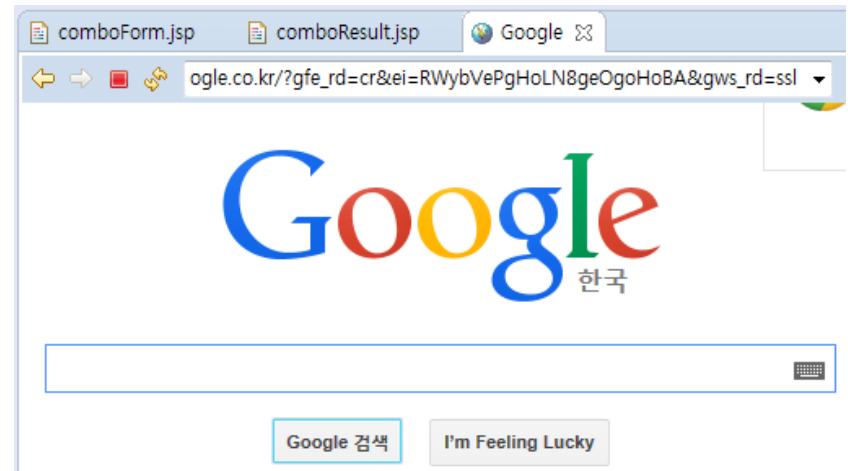
## 예제(3)

- 2개의 페이지를 작성하여 아래의 그림과 같이 출력되도록 하세요.
- 콤보 박스에서 항목을 선택하면 해당 사이트로 이동되도록 만드세요.
- 결과값은 request.getParameter()를 사용하세요.

<comboForm.jsp : 콤보 양식 페이지>



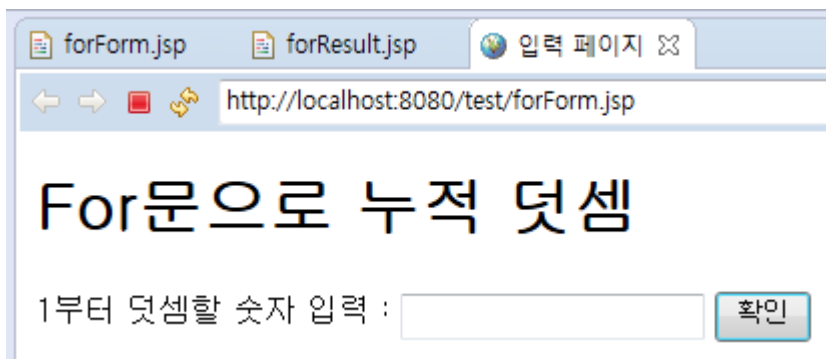
<comboResult.jsp : 결과 페이지>



## 예제[4]

- 2개의 페이지를 작성하여 아래의 그림과 같이 출력되도록 하세요.
- 입력 페이지에서 정수를 입력하고 [확인] 버튼을 누르면, 결과 페이지에서는 1부터 입력한 숫자까지의 합계가 출력되도록 만드세요.
- 결과 페이지에서 합계를 구하는 함수를 선언하여 사용하고, 결과값은 `request.getParameter()`와 `for`문을 사용하여 출력하세요.

### <forForm.jsp : 입력 양식 페이지>



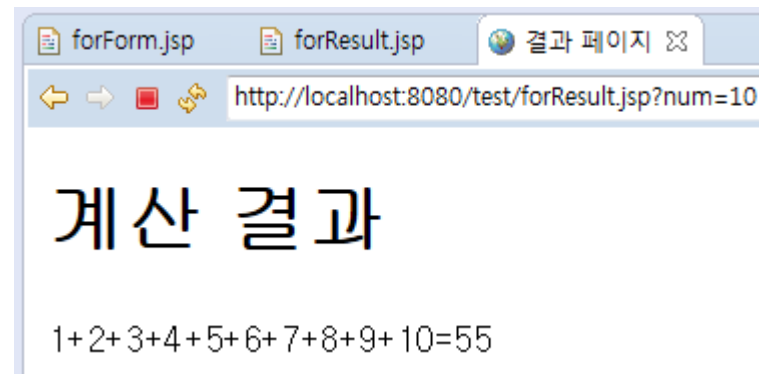
forForm.jsp forResult.jsp 입력 페이지

← → http://localhost:8080/test/forForm.jsp

## For문으로 누적 덧셈

1부터 덧셈할 숫자 입력 :

### <forResult.jsp : 결과 페이지>



forForm.jsp forResult.jsp 결과 페이지

← → http://localhost:8080/test/forResult.jsp?num=10

## 계산 결과

1+2+3+4+5+6+7+8+9+10=55