

내 장 객 체 의 종 류

내장객체 변수명	클래스/ 인터페이스 타입	설 명
request	javax.servlet.HttpServletRequest (javax.servlet.http.HttpServletRequest)	클라이언트의 HTTP 요청정보를 저장 한 객체(HTTP 헤더정보, 파라미터등)
response	javax.servlet.Response (javax.servlet.http.HttpRespnse)	HTTP 요청에 대한 응답정보를 저장한 객체
session	javax.servlet.http.HttpSession	클라이언트의 세션정보를 저장한 객체
pageContext	javax.servlet.jsp.PageContext	응답페이지 실행에 필요한 컨텍스트 정보를 저장한 객체
out	javax.servlet.jsp.JspWriter	응답 페이지 전송을 위한 출력 스트림 객체
application	javax.servlet.ServletContext	동일한 어플리케이션의 컨텍스트 정보를 저장한 객체
config	javax.servlet.ServletConfig	해당 페이지의 서블릿 설정 정보 (조기화 정보)를 저장한 객체
page	java.lang.Object (javax.servlet.jsp.HttpJspPage)	해당 페이지 서블릿 객체(인스턴스)
exception	java.lang.Throwable	예외 처리를 위한 객체

request 객체

(1) 요청 파라미터와 관련된 메소드

리턴타입	메 소 드 명	설 명
String	getParameter(String name)	name이란 이름으로 지정된 파라미터에 할당된 값을 리턴한다. 지정된 이름이 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 때 주로 사용되며 하나의 이름만 가질 때는 위의 getParameter메소드를 사용하자.
Enumeration	getParameterNames()	요청에 포함된 모든 파라미터 이름을 java.util.Enumeration 객체로 리턴한다.

(2) HTTP 헤더 정보와 관련된 메소드들

리턴타입	메 소 드 명	설 명
String	getHeader (String headername)	HTTP요청헤더에 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.Enumeration객체로 리턴한다.
Enumeration	getHeaderNames()	HTTP요청헤더에 포함된 모든 헤더이름을 java.util.Enumeration객체로 리턴한다.
int	getIntHeader (String headerName)	headerName헤더의 값을 int타입으로 리턴한다. 지정된 헤더값을 int로 변환할수 없을 경우는 NumberFormatException이 발생하고, headerName헤더가 없을 경우에는 -1을 리턴

(3) 세션 정보와 관련된 메소드들

리턴타입	메 소 드 명	설 명
HttpSession	getSession()	요청한 클라이언트에 지정된 HttpSession 객체를 얻는다. 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 세션 ID를 문자열로 리턴한다.
boolean	isRequestedSessionIdValid()	요청에 포함된 클라이언트의 세션 ID가 유효하면 true를, 아니면 false를 리턴한다.

(4) , URL/URI, 요청방식과 관련된 메소드들

리턴타입	메 소 드 명	설 명
Cookie[]	getCookies()	HTTP 요청 메시지의 헤더에 포함된 쿠키를 javax.servlet.http.Cookie 배열로 리턴한다.
String	getServerName()	서버의 도메인명을 문자열로 리턴한다.
int	getServerPort()	서버의 포트번호를 int형으로 리턴한다.
StringBuffer	getRequestURL()	요청 URL을 StringBuffer로 리턴한다.
String	getRequestURI()	요청 URI를 문자열로 리턴한다.
String	getQueryString()	요청에 사용된 쿼리 문자열을 문자열로 리턴한다.
String	getRemoteHost()	클라이언트 호스트 이름을 문자열로 리턴한다.
String	getRemoteAddr()	클라이언트의 IP주소를 문자열로 리턴한다.
String	getProtocol()	요청에 사용된 프로토콜 이름을 문자열로 리턴한다.
String	getMethod()	요청에 사용된 요청방식(GET, POST등)을 문자열로 리턴한다.
String	getContextPath()	해당 JSP페이지의 컨텍스트 경로를 문자열로 리턴한다.

response 객체

리턴타입	메 소 드 명	설 명
void	setHeader(String headerName, String value)	응답에 포함될 헤더 정보에 headerName의 이름으로 value값을 설정해 추가한다.
void	addCookie(Cookie cookie)	javax.servlet.http.Cookie형식의 쿠키를 응답헤더에 추가한다.
void	sendRedirect(String url)	지정된 url로 요청을 재 전송한다.
void	setContentType(String type)	응답 페이지의 contentType을 설정한다.

pageContext 객체

리턴타입	메 소 드 명	설 명
ServletRequest	getRequest()	클라이언트의 요청정보를 담고 있는 객체를 리턴한다(request 내장객체를 리턴한다.)
ServletResponse	getResponse()	요청에 대한 응답 객체를 리턴한다. (response내장 객체를 리턴한다.)
JspWriter	getOut()	응답 출력 스트림을 리턴한다 (out객체를 리턴한다)
Object	getPage()	서블릿 인스턴스 객체를 리턴한다. (page 내장객체를 리턴한다.)
Sevletconfig	getServletConfig()	서블릿의 초기 설정 정보를 담고 있는 객체를 리턴한다(config 내장객체를 리턴한다.)
ServletContext	getServletContext()	서블릿의 실행환경정보를 담고 있는 객체를 리턴한다(application 내장객체를 리턴한다)
HttpSession	getSession()	클라이언터의 세션정보를 담고있는 객체를 리턴한다(session 내장객체를 리턴한다)
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.

session 객체

리턴타입	메 소 드 명	설 명
string	string	해당 세션의 세션 ID를 문자열로 리턴한다. 세션ID는 session객체 생성시에 웹 컨테이너에 의해 자동 할당된다.
long	getCreationTime()	1970년 1월 1일 00시 00분 00초(epoch)부터 해당 세션이 생성된 순간까지의 경과시간을 밀리초로 계산하여 long형으로 리턴한다.
long	getLastAccessedTime()	epoch로부터 해당 세션에 마지막으로 접근된 시간까지의 경과시간을 밀리초로 계산하여 long형으로 리턴한다.
int	getMaxInactiveInterval()	클라이언트의 요청이 없을시 서버가 해당 세션을 유지하도록 지정된 시간을 초단위의 정수로 리턴한다.
없음	invalidate()	세션의 속성값으로 저장된 모든 객체를 반납하여 해당 세션을 종료시킨다.
boolean	isNew()	새로운 세션일 경우 true를 리턴하고, 기존의 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.

application 객체

리턴타입	메 소 드 명	설 명
string	getAttributeNames()	application 객체에 저장된 속성 이름 리턴
int	getMajorversion()	컨테이너가 지원하는 Major 버전 정보 리턴
int	getMinorVersion()	컨테이너가 지원하는 Minor 버전 정보 리턴
String	getServerInfo()	JSP/servlet 컨테이너 이름 및 버전 리턴
String	getMimeType (String file)	file에 지정된 파일에 대한 MIME 타입 리턴 (멀티미디어 타입)
java.net.URL	getResource (String path)	path에 지정된 자원을 URL 객체로 리턴
inputStream	getResourceAsStream (String path)	path에 지정된 자원을 InputStream 객체로 리턴
String	getRealPath (String path)	path에 지정된 자원 파일시스템의 실제 경로 리턴
없음	log(String msg)	message 내용을 로그 파일에 기록
없음	log(String msg, Throwable exception)	예외상황과 관련된 정보를 포함하여 로그파일에 기록

out 객체

리턴타입	메 소 드 명	설 명
없음	clear()	출력버퍼에 저장된 내용을 버린다. 만일 이미 버퍼가 다 채워져서 클라이언트로 전송이 되었을 경우에는 예외를 발생시킨다.
없음	clearBuffer()	출력 버퍼에 저장된 내용을 버린다. clear()메소드와는 다르게 버퍼에 담긴 내용이 이미 전송된 이후에도 예외를 발생시키지는 않고 현재 저장되어 있는 버퍼만을 버린다.
없음	flush()	현재 버퍼에 저장되어 있는 내용을 클라이언트로 전송하고 버퍼를 비운다.
없음	close()	출력 버퍼를 클라이언트로 전송하고 출력스트림을 종료한다.
boolean	isAutoFlush()	pageWriter의 pageFlush속성으로 지정된 값을 리턴한다. 즉, 출력 버퍼가 다 채워졌을 때 버퍼내용을 클라이언트로 지정되어 있으면, true를 리턴하고, 출력 버퍼가 다 채워졌을 때 예외가 발생하도록 지정되어 있으면 false를 리턴한다.
int	getBufferSize()	출력버퍼의 크기를 바이트 단위로 계산하여 정수값으로 리턴한다.
int	getRemaining()	출력버퍼의 남은양을 바이트 단위로 계산하여 정수값으로 리턴한다.
없음	print(String str)	출력스트림으로 str 문자열을 출력한다.

config 객체

리턴타입	메 소 드 명	설 명
String	getInitParameter (String init_paramName)	컨테이너의 설정 파일에 저장되어 있는 초기 파라미터 값 중 init_paramName의 이름을 가진 파라미터 값을 리턴한다. 이름이 없으면 null을 리턴한다.
Enumeration	getInitParameterNames()	컨테이너의 설정 파일에 저장되어 있는 모든 초기 파라미터 이름을 Enumeration 타입으로 리턴한다.
String	getServletName()	해당 서블릿의 이름을 문자열로 리턴한다.

exception 객체

리턴타입	메 소 드 명	설 명
String	getMessage()	에러 메시지를 문자열로 리턴한다.
없음	printStackTrace()	해당 에러의 스택 트레이스 정보를 출력한다
없음	printStackTrace (PrintWriter out)	해당 에러의 스택 트레이스 정보를 PrintWriter객체 out으로 출력한다.
없음	printStackTrace (PrintStream out)	해당 에러의 스택 트레이스 정보를 PrintStream객체 out으로 출력한다.