



7 익스프레션 언어

뇌를 자극하는 JSP & Servlet



❖ 학습목표

- 익스프레션 언어는 JSP 규격서가 처음 만들어질 때부터 있던 문법이 아니라 새롭게 추가된 문법이다. 이 문법을 이용하면 데이터를 출력하는 코드를 함축적이고 간결하게 구사할 수 있어 코드의 가독성과 유지보수 용이성에 큰 도움이 된다.

❖ 내용

- 익스프레션 언어란?
- 익스프레션 언어의 기초 문법
- 익스프레션 언어의 연산자
- 익스프레션 언어로 자바의 정적 메서드 호출하기



1. 익스프레션 언어란?

- 익스프레션 언어(expression language)란 식(expression)을 중심으로 코드를 기술하는 언어이다.
- 연산자와 피연산자의 조합을 `${와 }`로 둘러싸서 표현한다.

`${cnt+1}`

익스프레션 언어의 식(EL 식)

`<%= cnt+1 %>`

익스프레션의 식

- 위에 사용된 `cnt` 데이터 이름의 의미는 서로 다르다. 익스프레션에서 사용된 `cnt`는 자바 프로그래밍 언어의 변수 이름이며, EL 식에서 사용된 `cnt`는 애트리뷰트의 이름으로 해석된다.
- 애트리뷰트란 `setAttribute`, `getAttribute`, `removeAttribute` 메서드를 통해 저장되고, 관리되는 데이터를 의미한다.



1. 익스프레션 언어란?

<%

```
int sum = 0;
```

```
for (int cnt = 1; cnt <= 100; cnt++)
```

```
    sum += cnt;
```

```
    request.setAttribute( "RESULT ", new Integer(sum));
```

```
    RequestDispatcher dispatcher = request.getRequestDispatcher( "HundredResult.jsp ");
```

```
    dispatcher.forward(request, response);
```

%>

덧셈의 결과를 애트리뷰트로 저장합니다

호출

```
<%@page contentType= "text/html; charset=euc-kr" %>
```

```
<HTML>
```

```
    <HEAD><TITLE>1부터 100까지의 합</TITLE></HEAD>
```

```
    <BODY>
```

```
        1부터 100까지 더한 결과는? <%= request.getAttribute( "RESULT " ) %>
```

```
    </BODY>
```

```
</HTML>
```

[그림 7-1] 애트리뷰트 형태로 전달되는 데이터

애트리뷰트 값을 가져다가 출력합니다.



1. 익스프레션 언어란?

```
<%
  int sum = 0;
  for (int cnt = 1; cnt <= 100; cnt++)
    sum += cnt;
  request.setAttribute( "RESULT ", new Integer(sum));
  RequestDispatcher dispatcher = request.getRequestDispatcher( "HundredResult.jsp ");
  dispatcher.forward(request, response);
%>
```

덧셈의 결과를 애트리뷰트로 저장합니다

호출

```
<%@page contentType= "text/html; charset=euc-kr "%>
<HTML>
  <HEAD><TITLE>1부터 100까지의 합</TITLE></HEAD>
  <BODY>
    1부터 100까지 더한 결과는? ${RESULT}
  </BODY>
</HTML>
```

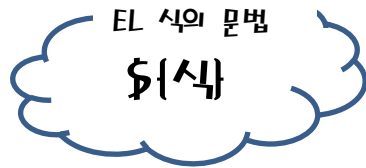
[그림 7-2] 애트리뷰트 값을 출력하는 EL 식

애트리뷰트 값을 가져다가 출력합니다.



2. 익스프레션 언어의 기초 문법

- 익스프레션 언어의 유일한 목적은 식을 계산해서 그 결과를 출력하는 것이므로 다음과 같은 하나의 문법으로 표현할 수 있으며 이 문법을 EL 식이라고 부른다.



- ‘식’ 위치에는 데이터 이름 하나로만 구성된 식이 들어갈 수도 있고, 연산자를 포함하는 식이 들어갈 수도 있으며, 자바의 정적 메서드를 호출하는 식이 들어갈 수도 있다.

$\$ \{ \text{RESULT} \}$

데이터 이름 하나로만 구성된
EL 식

$\$ \{ \text{RESULT} + 101 \}$

연산자를 포함하는
EL 식

$\$ \{ \text{m:sqrt}(100) \}$

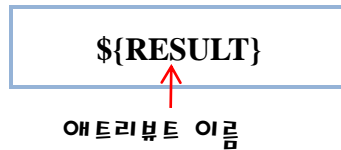
자바의 정적 메서드를 호출하는
EL 식



2. 익스프레션 언어의 기초 문법

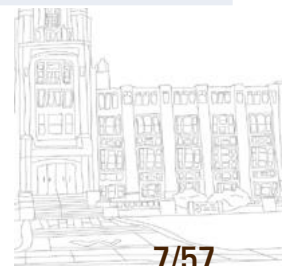
❖ 데이터 이름 하나로만 구성된 EL 식

- 데이터 이름 하나로만 구성된 EL 식은 가장 간단한 형태의 EL 식이다.
- EL 식 안에 기술되는 데이터 이름은 애트리뷰트 이름으로 해석된다.



[표 7-1] JSP/서블릿 기술에서 사용되는 네 종류의 애트리뷰트

애트리뷰트의 종류	호출할 때 사용하는 내장 변수	메서드의 소속
page 애트리뷰트	pageContext 내장 변수	javax.servlet.jsp.JspContext 클래스
request 애트리뷰트	request 내장 변수	javax.servlet.ServletRequest 인터페이스
session 애트리뷰트	session 내장 변수	javax.servlet.http.HttpSession 인터페이스
application 애트리뷰트	application 내장 변수	javax.servlet.ServletContext 인터페이스



2. 익스프레션 언어의 기초 문법

❖ 데이터 이름 하나로만 구성된 EL 식

[예제 7-1] 1부터 100까지의 합을 구하는 JSP 페이지

1부터 100까지의 합을 구하는 JSP 페이지

```
<%  
    int sum = 0;  
    for (int cnt = 1; cnt <= 100; cnt++)  
        sum += cnt;  
    request.setAttribute( "RESULT", new Integer(sum));  
    RequestDispatcher dispatcher = request.getRequestDispatcher( "HundredResult.jsp ");  
    dispatcher.forward(request, response);  
%>
```

request 데이터 영역에
애트리뷰트를 저장합니다

1부터 100까지의 합을 출력하는 JSP 페이지

```
<%@page contentType="text/html; charset=euc-kr"%>  
<HTML>  
    <HEAD><TITLE>1부터 100까지의 합</TITLE></HEAD>  
    <BODY>  
        1부터 100까지 더한 결과는? ${RESULT}  
    </BODY>  
</HTML>
```

request 데이터 영역에 있는
애트리뷰트 값을 가져다가 출
력합니다



2. 익스프레션 언어의 기초 문법

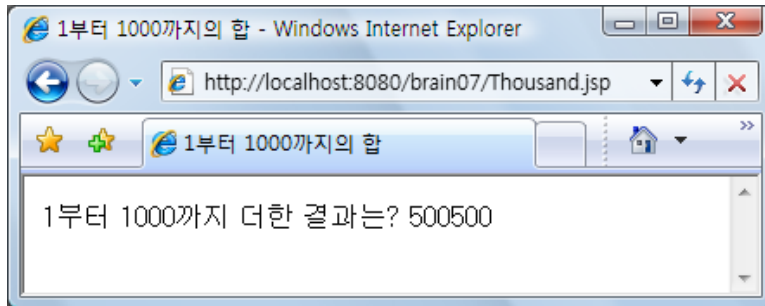
❖ 데이터 이름 하나로만 구성된 EL 식

[예제 7-2] 1부터 1000까지의 합을 구하는 JSP 페이지

```
<%@page contentType= "text/html; charset=euc-kr" %>
<%
    int sum = 0;
    for (int cnt = 1; cnt <= 1000; cnt++)
        sum += cnt;
    pageContext.setAttribute( "RESULT", new Integer(sum));
%>
<HTML>
    <HEAD><TITLE>1부터 1000까지의 합</TITLE></HEAD>
    <BODY>
        1부터 1000까지 더한 결과는? ${RESULT}
    </BODY>
</HTML>
```

page 데이터 영역에
애트리뷰트를 저장합니다

page 데이터 영역에 있는
애트리뷰트 값을 가져다가 출력합니다.



[그림 7-4] 예제 7-2의 실행 결과



2. 익스프레션 언어의 기초 문법

❖ 데이터 이름 하나로만 구성된 EL 식

- EL 식에 있는 데이터 이름을 해석하는 순서는 사용 범위가 좁은 애트리뷰트부터 점점 더 사용 범위가 넓은 애트리뷰트 순으로 진행 된다.



- 순서에 상관없이 특정한 종류의 애트리뷰트를 짚어서 출력하고 싶을 때는 다음과 같이 표시하면 된다.

`${pageScope.SUM}`
↑
page 애트리뷰트임을 표시

`${requestScope.RESULT}`
↑
request 애트리뷰트임을 표시

`${sessionScope.CART}`
↑
session 애트리뷰트임을 표시

`${applicationScope.DB_NAME}`
↑
application 애트리뷰트임을 표시



2. 익스프레션 언어의 기초 문법

❖ 익스프레션 언어의 내장 객체

[표 7-2] 익스프레션 언어의 내장 객체

내장 객체 이름	표현하는 데이터	객체의 타입
pageScope	page 애트리뷰트의 집합	Map
requestScope	request 애트리뷰트의 집합	Map
sessionScope	session 애트리뷰트의 집합	Map
applicationScope	application 애트리뷰트의 집합	Map
param	웹 브라우저로부터 입력된 데이터의 집합	Map
paramValues	웹 브라우저로부터 입력된 데이터의 집합 (똑같은 이름의 데이터가 여럿일 때 사용)	Map
header	HTTP 요청 메시지에 있는 HTTP 헤더의 집합	Map
headerValues	HTTP 요청 메시지에 있는 HTTP 헤더의 집합 (똑같은 이름의 HTTP 헤더가 여럿일 때 사용)	Map
cookie	웹 브라우저로부터 전송된 쿠키의 집합	Map
initParam	웹 애플리케이션의 초기화 파라미터의 집합	Map
pageContext	JSP 페이지의 환경 정보의 집합	PageContext



2. 익스프레션 언어의 기초 문법

❖ 익스프레션 언어의 내장 객체

- param은 웹 브라우저에서 <FORM> 엘리먼트를 통해 입력된 데이터를 가져올 때 사용하는 내장 객체이다.
- param 객체의 사용 방법은 두 가지
 - param 뒤에 마침표를 찍고 해당 데이터 이름을 쓰는 방법
 - param 뒤에 대괄호를 치고, 그 안에 작은따옴표나 큰 따옴표로 묶은 데이터 이름을 쓰는 방법

`${param.NUM}`

↑
입력 데이터의 이름

`${param["COLOR "]}`

↑
입력 데이터의 이름

- <FORM> 엘리먼트를 통해 똑같은 이름의 데이터가 여러 개 입력되는 경우도 있는데, 그럴 때는 paramValuse 내장 객체를 사용하면 된다.



2. 익스프레션 언어의 기초 문법

❖ 익스프레션 언어의 내장 객체

- paramValues 내장 객체를 이용해서 데이터를 가져오는 방법은 두 가지이다.
- 하나는 객체 이름 뒤에 마침표를 찍고, 그 다음에 데이터 이름을 쓰고, 그 다음에 데이터 값의 인덱스를 대괄호로 묶어서 표시하는 것이고, 다른 하나는 객체 이름 뒤에 두 개의 대괄호를 치고 그 안에 각각 따옴표로 묶은 데이터 이름과 인덱스를 쓰는 것이다.

`${paramValues.ANIMAL[0]}`

입력 데이터 이름 인덱스

`${paramValues["ANIMAL"][1]}`

입력 데이터 이름 인덱스

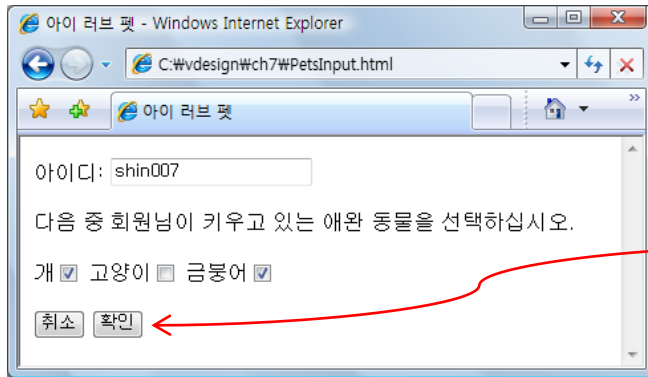
- EL 식의 인덱스가 0부터 시작하므로, 첫 번째 데이터 값을 가져오기 위해서는 인덱스를 0이라고 써야 하고, 두 번째 데이터 값을 가져오기 위해서는 1이라고 써야 한다.



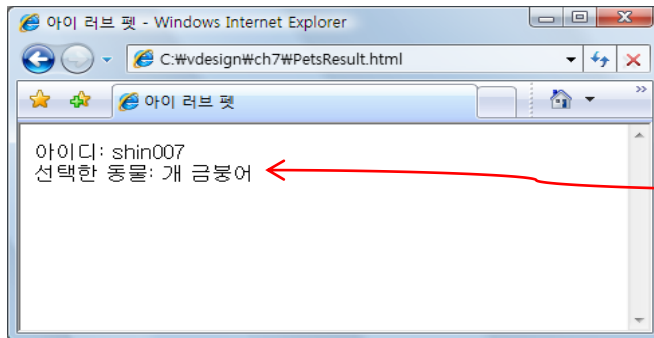
2. 익스프레션 언어의 기초 문법

❖ 익스프레션 언어의 내장 객체

- param과 paramValues 내장 객체를 사용하는 웹 애플리케이션을 작성해보자.



① 데이터를 선택하고 '확인' 버튼을 누르면



② 선택한 데이터가 나타납니다.

[그림 7-5] 애완동물 웹 애플리케이션의 화면 설계



2. 익스프레션 언어의 기초 문법

❖ 익스프레션 언어의 내장 객체

- 앞 두 화면을 HTML문서와 JSP 페이지로 구현하고, URL을 각각 다음과 같이 정한다.

`http://localhost:8080/brain07/PetsInput.html`

← 위쪽 화면의 URL

`http://localhost:8080/brain07/PetsResult.jsp`

← 아래쪽 화면의 URL

[예제 7-3] 데이터를 입력받는 HTML 문서

```
<HTML>
<HEAD>
  <META http-equiv= "Content-Type " content= "text/html; charset=euc-kr ">
  <TITLE>아이 러브 펫</TITLE>
</HEAD>
<BODY>
  <FORM ACTION=PetsResult.jsp>
    아이디: <INPUT TYPE=TEXT NAME=ID><BR><BR>
    다음 중 회원님이 키우고 있는 애완동물을 선택하십시오.<BR><BR>
    개<INPUT TYPE=CHECKBOX NAME=ANIMAL VALUE= "개" >
    고양이<INPUT TYPE=CHECKBOX NAME=ANIMAL VALUE= "고양이" >
    금붕어<INPUT TYPE=CHECKBOX NAME=ANIMAL VALUE= "금붕어" ><BR><BR>
    <INPUT TYPE=RESET VALUE= "취소" >
    <INPUT TYPE=SUBMIT VALUE= "확인" >
  </FORM>
</BODY>
</HTML>
```

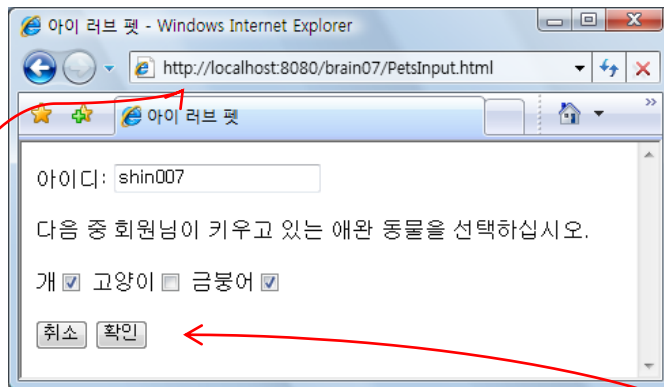


2. 익스프레션 언어의 기초 문법

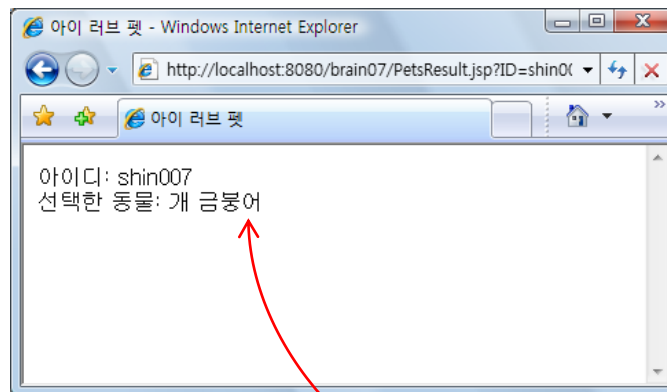
❖ 익스프레션 언어의 내장 객체

[예제 7-4] 입력된 데이터를 출력하는 JSP 페이지

```
<%@page contentType= "text/html; charset=euc-kr"%>
<HTML>
  <HEAD><TITLE>아이 러브 펫</TITLE></HEAD>
  <BODY>
    아이디: ${param.ID} <BR>
    선택한 동물: ${paramValues.ANIMAL[0]}
               ${paramValues.ANIMAL[1]}
               ${paramValues.ANIMAL[2]}
  </BODY>
</HTML>
```



① (예제 7-3)의 URL을 입력하세요.



② 데이터를 선택하고 '확인' 버튼을 누르면

③ 결과 화면이 나타날 것입니다.

2. 익스프레션 언어의 기초 문법

❖ 익스프레션 언어의 내장 객체

- header 내장 객체는 HTTP 요청 메시지에 포함된 HTTP 헤더 값을 가져올 때 사용한다.
- header 내장 객체를 이용해서 HTTP 헤더 값을 가져오는 방법은 두 가지이다. 하나는 이 객체의 이름 뒤에 마침표를 찍고 그 다음에 해당 헤더 이름을 쓰는 것이며, 또 하나는 객체의 이름 뒤에 대괄호를 치고 그 안에 작은 따옴표나 큰 따옴표로 묶은 헤더 이름을 쓰는 것이다.

`${header.Host}`

↑
HTTP 헤더 이름

`${header["User-Agent "]}`

↑
HTTP 헤더 이름

- 첫 번째 사용 방법의 제약 사항 - HTTP 헤더 이름이 자바의 식별자 명명 규칙을 따르지 않을 때는 사용할 수 없다.

`${header.User-Agent}`

↑
잘못된 EL 식

`${header["User-Agent "]}`

↑
올바른 EL 식



2. 익스프레션 언어의 기초 문법

❖ 익스프레션 언어의 내장 객체

- HTTP 요청 메시지 안에 똑같은 이름의 HTTP 헤더가 둘 이상 있을 때는 header 내장 객체 대신 headerValues 내장 객체를 사용해야 한다.
- headerValues 내장 객체의 이름 뒤에 마침표나 대괄호를 이용해서 헤더 이름을 표시하고, 그 다음에 대괄호로 묶은 인덱스를 표시해야 한다.

`${headerValues.Accept[0]}`

HTTP 헤더 이름 인덱스

`${headerValues["User-data"][1]}`

HTTP 헤더 이름 인덱스



2. 익스프레션 언어의 기초 문법

❖ 익스프레션 언어의 내장 객체

- cookie 내장 객체는 웹 브라우저가 웹 서버로 보낸 쿠키를 가져올 때 사용하며, 마침표와 대괄호를 이용하는 사용 방법 두 가지가 있다.

`${cookie.CART}`

↑
쿠키의 이름

`${cookie["USER_NAME"]}`

↑
쿠키의 이름

- 위 EL 식이 가져오는 것은 쿠키의 값이 아니라 쿠키 객체이므로, JSP 페이지 안에 이런 EL 식을 써 놓으면 사용자에게 아무 의미 없는 쿠키 객체의 참조 값만 출력될 것이다.
- 쿠키를 가져오는 식 뒤에 마침표를 찍고 value라고 쓰거나, 대괄호를 치고 그 안에 'value' 또는 "value" 라고 쓰면 쿠키 값이 출력된다.

`${cookie.CART.value}`

↑
쿠키의 값을 가져오라는 표시

`${cookie["CART"]["value"]}`

↑
쿠키의 값을 가져오라는 표시

`${cookie.CART["value"]}`

↑
쿠키의 값을 가져오라는 표시

`${cookie["CART"].value}`

↑
쿠키의 값을 가져오라는 표시



2. 익스프레션 언어의 기초 문법

❖ 익스프레션 언어의 내장 객체

- 쿠키 객체 안에는 쿠키 값 외에도 쿠키가 속하는 도메인 이름, URL 경로명, 쿠키의 수명 같은 중요한 정보들이 들어 있다.
- 그런 정보를 출력하기 위해서는 앞 페이지와 같은 형식의 EL 식에서 `value`라는 이름을 빼고 대신 `domain`, `path`, `maxAge`라는 이름을 써 넣으면 된다.

`${cookie.CART.domain}`

쿠키의 도메인 이름을 가져오라는 표시

`${cookie.CART["path"]}`

쿠키의 URL 경로명을 가져오라는 표시

`${cookie["CART"]["maxAge"]}`

쿠키의 수명을 가져오라는 표시



2. 익스프레션 언어의 기초 문법

❖ 익스프레션 언어의 내장 객체

[예제 7-5] 쿠키 데이터를 가져다가 출력하는 EL 식의 예

쿠키 데이터를 저장하는 JSP 페이지

```
<%@page contentType="text/html; charset=euc-kr" %>
<%
    Cookie cookie = new Cookie("NAME", "John");
    response.addCookie(cookie);
%>
<HTML>
  <HEAD><TITLE>쿠키 데이터 저장 프로그램</TITLE></HEAD>
  <BODY>
    쿠키 값이 설정되었습니다.
  </BODY>
</HTML>
```

쿠키 데이터를 웹 브라우저
쪽에 저장합니다

쿠키 데이터 값을 출력하는 JSP 페이지

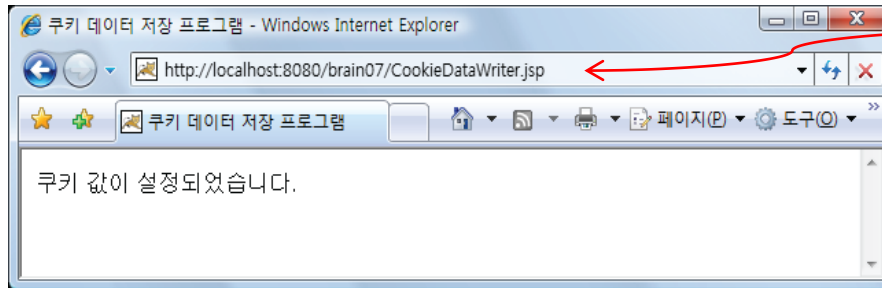
```
<%@page contentType="text/html; charset=euc-kr" %>
<HTML>
  <HEAD><TITLE>쿠키 데이터 출력 프로그램</TITLE></HEAD>
  <BODY>
    NAME 쿠키 데이터의 값은? ${cookie.NAME.value}
  </BODY>
</HTML>
```

쿠키 데이터의 값을 가져와서
출력합니다

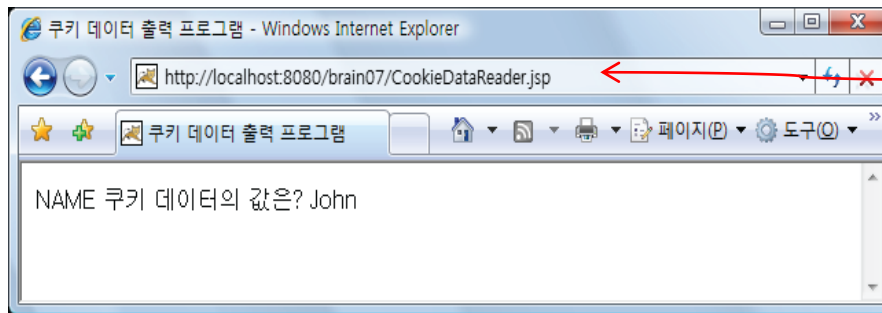


2. 익스프레스션 언어의 기초 문법

❖ 익스프레스션 언어의 내장 객체



쿠키 데이터를 저장하는 JSP 페이지의 URL을 입력하세요



쿠키 데이터를 출력하는 JSP 페이지의 URL을 입력하세요

[그림 7-7] 예제 7-5의 실행 결과

2. 익스프레션 언어의 기초 문법

❖ 익스프레션 언어의 내장 객체

- `initParam`은 웹 애플리케이션의 초기화 파라미터 값을 가져다가 출력할 때 사용하는 내장 객체이다.
- `initParam` 객체의 이름 뒤에 마침표나 대괄호를 이용해서 해당 초기화 파라미터의 이름을 표시한다.

`${initParam.DB_NAME}`

웹 애플리케이션의 초기화 파라미터 이름

`${initParam.["DB_NAME"]}`

웹 애플리케이션의 초기화 파라미터 이름

[예제 7-6] 웹 애플리케이션의 초기화 파라미터 값을 출력하는 JSP 페이지

```
<%@page contentType= "text/html; charset=euc-kr" %>
<HTML>
  <HEAD><TITLE>애플리케이션 초기화 파라미터 예제</TITLE></HEAD>
  <BODY>
    DB_NAME 초기화 파라미터의 값은? ${initParam.DB_NAME}
  </BODY>
</HTML>
```



2. 익스프레션 언어의 기초 문법

❖ 익스프레션 언어의 내장 객체

- `pageContext` 내장 객체는 JSP 페이지의 주변 환경에 대한 정보를 제공한다. 이 내장 객체의 사용 방법은 다소 독특하다.
- `pageContext` 내장 객체의 타입은 `PageContext`라고 되어 있는데 이것은 `java.servlet.jsp` 패키지에 속하는 클래스 이름이다. 이 객체를 이용하면 `PageContext` 클래스에 속하는 `get`으로 시작하는 이름의 메서드를 호출 할 수 있다.
- `pageContext` 클래스에는 8개의 `get`-메서드가 있으며, EL 식을 이용해서 이 메서드들을 호출할 때는 메서드 이름 제일 앞에 있는 `get`이라는 단어를 떼고, 그 다음에 있는 첫 문자를 소문자로 고친 이름을 사용하면 된다.

`${pageContext.request}`

`getRequest` 메서드를 가리키는 단어

`${pageContext["request"]}`

`getRequest` 메서드를 가리키는 단어

- 이 EL 식은 `getRequest` 메서드의 리턴값을 출력하는데, 그 값은 JSP 페이지의 `request` 내장 변수의 값과 동일한 객체이므로 사용자에게 아무 의미도 없는 참조값만 출력된다.



2. 익스프레션 언어의 기초 문법

❖ 익스프레션 언어의 내장 객체

- 다음과 같은 EL 식을 이용하면 JSP 페이지의 URL 경로명을 출력할 수 있다.

```
${pageContext.request.requestURI}
```

↑
getRequestURI 메서드의 리턴값을 가져오는 단어

```
${pageContext["request"]["requestURI"]}
```

↑
getRequestURI 메서드의 리턴값을 가져오는 단어

```
${pageContext.request["requestURI"]}
```

↑
getRequestURI 메서드의 리턴값을 가져오는 단어

```
${pageContext["request"].requestURI}
```

↑
getRequestURI 메서드의 리턴값을 가져오는 단어

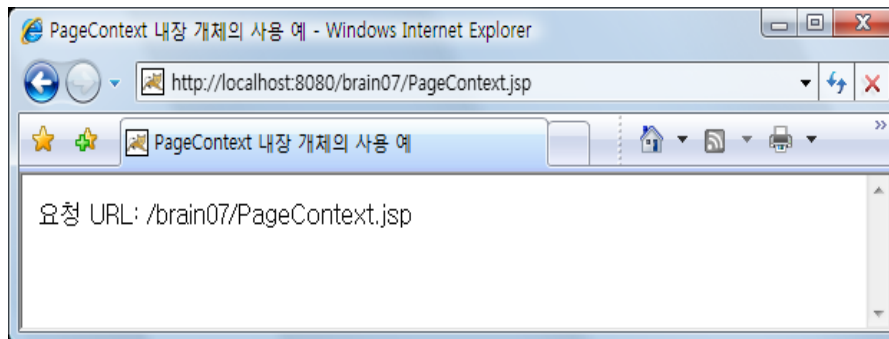


2. 익스프레션 언어의 기초 문법

❖ 익스프레션 언어의 내장 객체

[예제 7-7] pageContext 내장 객체의 사용 예

```
<%@page contentType= "text/html; charset=euc-kr " %>
<HTML>
  <HEAD><TITLE>PageContext 내장 객체의 사용 예</TITLE></HEAD>
  <BODY>
    요청 URL: ${pageContext.request.requestURI} <BR>
  </BODY>
</HTML>
```



[그림 7-11] 예제 7-7의 실행 결과





Thank You !

뇌를 자극하는 JSP & Servlet