



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



3. 익스프레션 언어의 연산자

❖ 산술 연산자, 비교 연산자, 논리 연산자, 조건 연산자

- 자바 연산자와 동일한 기능을 하는 연산자들에 대해 알아보자.

[예제 7-8] 익스프레션 언어의 연산자 사용 예 (1)

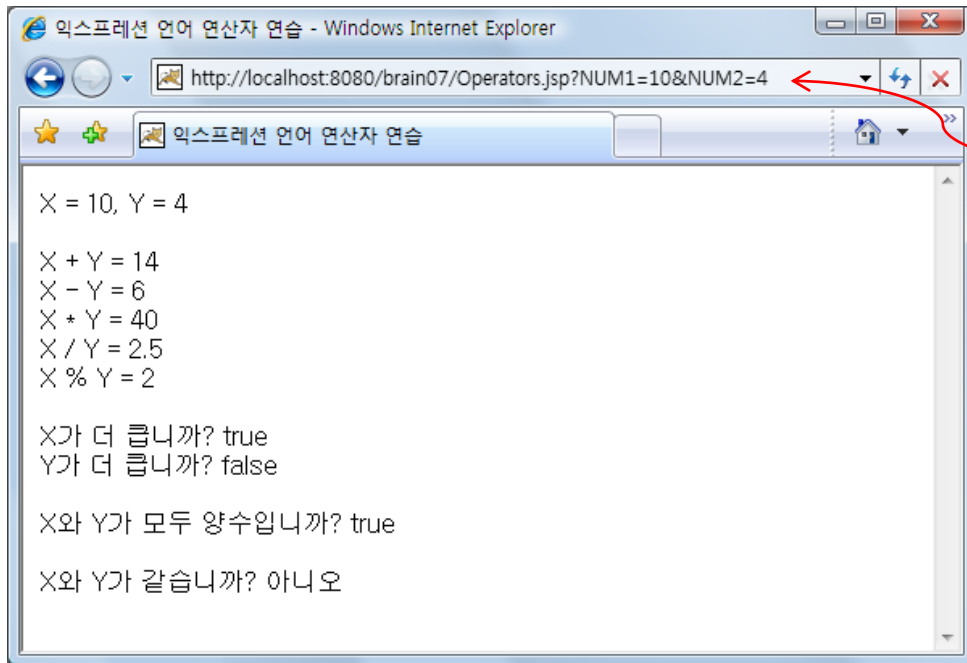
```
<%@page contentType= "text/html; charset=euc-kr" %>
<HTML>
  <HEAD><TITLE>익스프레션 언어 연산자 연습</TITLE></HEAD>
  <BODY>
    X = ${param.NUM1}, Y = ${param.NUM2} <BR><BR>
    X + Y = ${param.NUM1 + param.NUM2} <BR>
    X - Y = ${param.NUM1 - param.NUM2} <BR>
    X * Y = ${param.NUM1 * param.NUM2} <BR>
    X / Y = ${param.NUM1 / param.NUM2} <BR>
    X % Y = ${param.NUM1 % param.NUM2} <BR><BR>
    X가 더 큼니까? ${param.NUM1 - param.NUM2 > 0} <BR>
    Y가 더 큼니까? ${param.NUM1 - param.NUM2 < 0} <BR><BR>
    X와 Y가 모두 양수입니까? ${ (param.NUM1 > 0) && (param.NUM2 > 0) } <BR><BR>
    X와 Y가 같습니까? ${param.NUM1 == param.NUM2? "예" : "아니오" } <BR><BR>
  </BODY>
</HTML>
```



3. 익스프레션 언어의 연산자

❖ 산술 연산자, 비교 연산자, 논리 연산자, 조건 연산자

- 톰캣의 brain07 웹 애플리케이션 디렉터리에 Operators.jsp라는 이름으로 저장 후 실행해 본다.



URL 뒤에 이런 식으로
두 개의 수를 쓰세요.

[그림 7-12] 예제 7-8의 실행 결과

3. 익스프레션 언어의 연산자

❖ 산술 연산자, 비교 연산자, 논리 연산자, 조건 연산자

- 익스프레션 언어의 연산자 중에는 HTML 문법이나 JSP 문법에서도 특별한 의미를 갖는 기호를 사용하는 것이 있다.
- 이런 기호들이 잘못 해석되는 것을 방지하기 위해 익스프레션 언어에서는 기호로 된 연산자들과 똑같은 기능을 하는 영문 단어 연산자들을 추가로 만들어 놓고 있다.

[예제 7-9] 익스프레션 언어의 연산자 사용 예 (2)

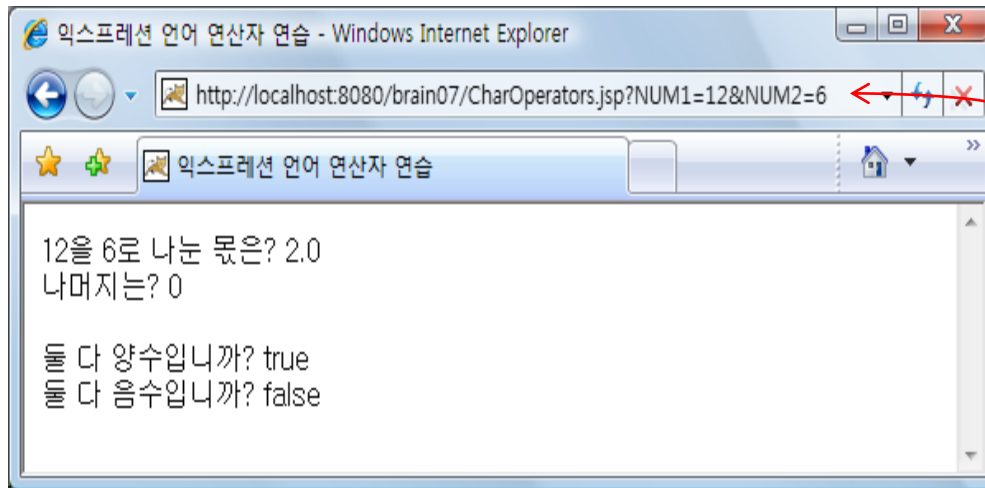
```
<%@page contentType= "text/html; charset=euc-kr " %>
<HTML>
  <HEAD><TITLE>익스프레션 언어 연산자 연습</TITLE></HEAD>
  <BODY>
    ${param.NUM1}을 ${param.NUM2}로 나눈 몫은? ${param.NUM1 div param.NUM2} <BR>
    나머지는? ${param.NUM1 mod param.NUM2} <BR><BR>
    둘 다 양수입니까? ${ (param.NUM1 gt 0) and (param.NUM2 gt 0)} <BR>
    둘 다 음수입니까? ${ (param.NUM1 lt 0) and (param.NUM2 lt 0)} <BR>
  </BODY>
</HTML>
```



3. 익스프레션 언어의 연산자

❖ 산술 연산자, 비교 연산자, 논리 연산자, 조건 연산자

- 앞 예제를 톰캣의 brain07 웹 애플리케이션 디렉터리에 CharOperators.jsp라는 이름으로 저장한 후 실행해 본다.



URL 뒤에 이런 식으로
두 개의 수를 쓰세요.

[그림 7-13] 예제 7-9의 실행 결과

3. 익스프레션 언어의 연산자

❖ 산술 연산자, 비교 연산자, 논리 연산자, 조건 연산자

- 익스프레션 언어의 연산자 중에는 같은 모습으로 다른 기능을 하는 것도 있다.

`${gender == "female"}`

↑
두 값이 같으면 *true* 다르면 *false*

`${"CAR" < "CAT"}`

↑
유니코드에 따른 사전식 비교.
CAR가 먼저이므로 *true*

`${"CAT" > "DOG"}`

↑
유니코드에 따른 사전식 비교.
CAT이 먼저이므로 *false*

[예제 7-10] 익스프레션 언어의 연산자 사용 예 (3)

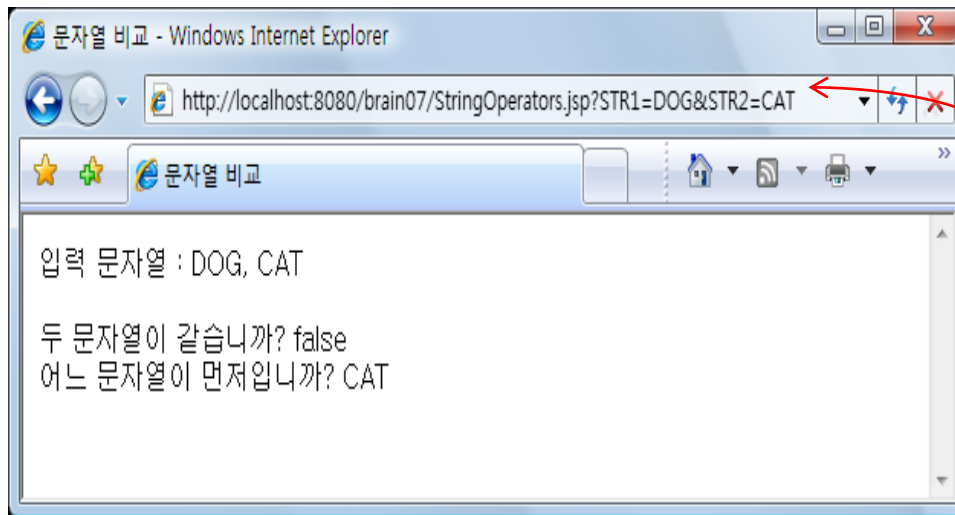
```
<%@page contentType= "text/html; charset=euc-kr" %>
<HTML>
  <HEAD><TITLE>문자열 비교</TITLE></HEAD>
  <BODY>
    입력 문자열 : ${param.STR1}, ${param.STR2} <BR><BR>
    두 문자열이 같습니까? ${param.STR1 == param.STR2} <BR>
    어느 문자열이 먼저입니까? ${param.STR1 < param.STR2 ? param.STR1 : param.STR2}
  </BODY>
</HTML>
```



3. 익스프레션 언어의 연산자

❖ 산술 연산자, 비교 연산자, 논리 연산자, 조건 연산자

- 앞 예제를 톰캣의 brain07 웹 애플리케이션 디렉터리에 StringOperators.jsp라는 이름으로 저장한 후에 실행해 본다.



URL 뒤에 이런 식으로
두 개의 수를 쓰세요.

[그림 7-13] 예제 7-9의 실행 결과



3. 익스프레션 언어의 연산자

❖ 엠프티 연산자

- empty라는 단어 형태의 엠프티 연산자는 데이터의 존재 여부를 확인하는 단항 연산자이다. 피연산자인 데이터 이름은 empty라는 연산자 이름 뒤에 써야 한다.

`${empty NAME}`

↑
데이터 이름

[예제 7-11] 엠프티 연산자의 사용 예

```
<%@page contentType= "text/html; charset=euc-kr" %>
<HTML>
  <HEAD><TITLE>엠프티 연산자</TITLE></HEAD>
  <BODY>
    안녕하세요, ${empty param.ID ? "guest" : param.ID}님
  </BODY>
</HTML>
```



3. 익스프레션 언어의 연산자

❖ 연산자의 우선순위를 바꾸는 괄호 연산자

- 여러 개의 연산자가 포함된 수식식에는 왼쪽에서부터 오른쪽으로 계산되는 것이 순서이지만, 가감승제 연산자가 뒤섞여 있을 때는 곱셈, 나눗셈이 덧셈, 뺄셈보다 먼저 계산된다. 익스프레션 언어의 연산자에도 마찬가지로 우선순위가 있다.
- EL 식 안에 여러 개의 연산자가 있으면 왼쪽부터 오른쪽으로 순서대로 처리되지만, 우선순위가 다른 연산자가 섞여 있으면 높은 우선순위의 연산자가 먼저 처리된다.

$$\$(2 + 3 * 4)$$

↑
곱셈이 먼저 수행됩니다

- 우선순위를 바꾸기 위해서는 수식식과 마찬가지로 괄호를 사용하면 된다.

$$\$((2 + 3) * 4)$$

↑
덧셈이 먼저 수행됩니다.



3. 익스프레션 언어의 연산자

❖ 연산자의 우선순위를 바꾸는 괄호 연산자

- 익스프레션 언어의 연산자 우선 순위

우선순위	연 산 자
↑ 높음	[]
	()
	-(부호) ! not empty
	* / % div mod
	+ -
	< > <= >= it ht le ge
	== != eq ne
	&& or
	or
↓ 낮음	? :



3. 익스프레션 언어의 연산자

❖ 대괄호 연산자와 마침표 연산자

- 대괄호와 마침표 연산자의 용도: 자바에서는 배열 항목과 객체 멤버를 가리키기 위해 사용되지만, 익스프레션 언어에서는 다음과 같은 데이터 항목을 가리키기 위해 사용된다.



- * 배열 항목
- * `java.util.List` 객체의 데이터 항목
- * `java.util.Map` 객체의 데이터 항목
- * 자바빈(JavaBean) 프로퍼티

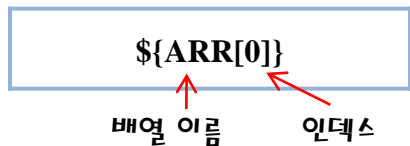
- 배열과 `java.util.List` 객체의 데이터 항목을 가리킬 때는 반드시 대괄호 연산자를 이용해야 한다.
- `java.util.Map` 객체의 데이터 항목과 자바빈 프로퍼티를 가리킬 때는 대괄호 연산자와 마침표 연산자 중 하나를 사용할 수 있다.



3. 익스프레션 언어의 연산자

❖ 대괄호 연산자와 마침표 연산자

- 대괄호 연산자를 이용해서 배열 항목을 가져다가 출력하는 방법 - 자바에서와 마찬가지로 배열 이름 다음에 대괄호로 묶은 인덱스를 표시하면 된다.



- EL 식의 인덱스는 자바에서와 마찬가지로 0부터 시작하고, 0은 첫 번째 데이터 항목을 가리킨다.



3. 익스프레션 언어의 연산자

❖ 대괄호 연산자와 마침표 연산자

[예제 7-12] 배열 항목을 출력하는 EL 식의 예

```
<%@page contentType= "text/html; charset=euc-kr" %>
<%
    String winners[] = new String[3];
    winners[0] = "이수현" ;
    winners[1] = "정세훈" ;
    winners[2] = "김진희" ;
    request.setAttribute( "WINNERS ", winners);
    RequestDispatcher dispatcher = request.getRequestDispatcher( "WinnersView.jsp ");
    dispatcher.forward(request, response);
%>
```

배열을 애트리뷰트 형태로
저장합니다

호 출

```
<%@page contentType= "text/html; charset=euc-kr" %>
<HTML>
  <HEAD><TITLE>우승자 명단</TITLE></HEAD>
  <BODY>
    <H3>우승자 명단</H3>
    1등. ${WINNERS[0]} <BR>
    2등. ${WINNERS[1]} <BR>
    3등. ${WINNERS[2]} <BR>
  </BODY>
</HTML>
```

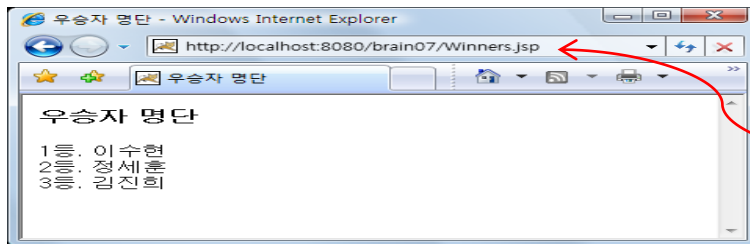
배열 항목을 가져다가 출력하는
EL 식입니다



3. 익스프레션 언어의 연산자

❖ 대괄호 연산자와 마침표 연산자

- 앞의 두 JSP 페이지를 톰캣의 brain07 웹 애플리케이션 디렉터리에 Winners.jsp와 WinnersView.jsp라는 이름으로 저장한다.



(예제 7-12)의 첫 번째 JSP 페이지의 URL을 쓰세요

[그림 7-16] 예제 7-12의 실행 결과

- java.util.List는 자바의 표준 라이브러리에 있는 인터페이스 이름인데, 이 인터페이스에는 여러 개의 데이터 항목을 저장해서 관리할 수 있는 메서드들이 있다.

```
ArrayList<String> items = new ArrayList<String>();  
items.add( "딸기" );  
items.add( "오렌지" );  
items.add( "복숭아" );
```

문자열을 저장할 수 있는
ArrayList 객체를 만듭니다

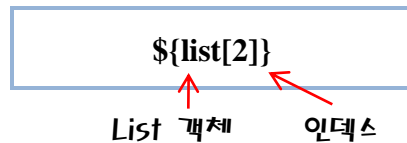
ArrayList 객체에 3개의 문자
열을 저장합니다



3. 익스프레션 언어의 연산자

❖ 대괄호 연산자와 마침표 연산자

- ArrayList 객체에 저장한 데이터 항목을 EL 식을 이용해서 가져오려면 배열의 경우와 마찬가지로 대괄호 연산자를 이용하면 된다.



3. 익스프레션 언어의 연산자

❖ 대괄호 연산자와 마침표 연산자

[예제 7-13] List 객체의 항목을 출력하는 EL 식의 예

```
<%@page contentType= "text/html; charset=euc-kr" %>
<%@page import= "java.util.*" %>
<%
    ArrayList<String> items = new ArrayList<String>();
    items.add( "딸기" );
    items.add( "오렌지" );
    items.add( "복숭아" );
    request.setAttribute( "FRUITS ", items);
    RequestDispatcher dispatcher = request.getRequestDispatcher( "FruitsView.jsp ");
    dispatcher.forward(request, response);
%>
```

List 객체를 애트리뷰트
형태로 저장합니다

호출

```
<%@page contentType= "text/html; charset=euc-kr" %>
<HTML>
  <HEAD><TITLE>인기 상품 목록</TITLE></HEAD>
  <BODY>
    <H3>이달에 가장 많이 팔린 과일입니다.</H3>
    1위. ${FRUITS[0]} <BR>
    2위. ${FRUITS[1]} <BR>
    3위. ${FRUITS[2]} <BR>
  </BODY>
</HTML>
```

List 객체의 항목을 가져다가 출력하는
EL 식입니다



3. 익스프레션 언어의 연산자

❖ 대괄호 연산자와 마침표 연산자

- `java.util.Map`는 `java.util.List`와 마찬가지로 자바의 표준 라이브러리에 있는 인터페이스 이름이며, 이 인터페이스에는 여러 데이터 항목을 <이름, 값> 쌍으로 저장해서 관리할 수 있는 메서드들이 있다.

```
HashMap<String, Integer> map = new HashMap<String, Integer>();  
map.put( "Edgar ", new Integer(95));  
map.put( "Thomas ", new Integer(100));  
map.put( "John ", new Integer(75));
```

이름이 `String` 타입이고, 값이 `Integer` 타입인 데이터를 저장할 수 있는 `HashMap` 객체를 만듭니다

`HashMap` 객체에 3개의 데이터를 저장합니다

- `Map` 객체에 저장한 데이터 값을 가져오기 위해서는 대괄호 연산자나 마침표 연산자를 이용해서 데이터의 이름을 지정해야 한다.

`${MAP["John "]}`

Map 객체 데이터 이름

`${MAP.John}`

Map 객체 데이터 이름



3. 익스프레션 언어의 연산자

❖ 대괄호 연산자와 마침표 연산자

[예제 7-14] Map 객체의 항목을 출력하는 EL 식의 예

```
<%@page contentType= "text/html; charset=euc-kr" %>
<%@page import= "java.util.*" %>
<%
    HashMap<String, String> map = new HashMap<String, String>();
    map.put( "Edgar", "보스턴" );
    map.put( "Thomas", "오하이오" );
    map.put( "John", "워싱턴" );
    request.setAttribute( "ADDRESS", map);
    RequestDispatcher dispatcher = request.getRequestDispatcher( "AddressView.jsp" );
    dispatcher.forward(request, response);
%>
```

Map 객체를 애트리뷰트
형태로 저장합니다

호출

```
<%@page contentType= "text/html; charset=euc-kr" %>
<HTML>
  <HEAD><TITLE>주소록</TITLE></HEAD>
  <BODY>
    ${param.NAME}의 주소는? ${ADDRESS[param.NAME]}
  </BODY>
</HTML>
```

Map 객체에서 데이터 이름에 해당하는
항목을 찾아서 출력하는 EL 식입니다.



3. 익스프레션 언어의 연산자

❖ 대괄호 연산자와 마침표 연산자

- 자바빈(JavaBean)이란 JavaBean 규격서에서 따라 작성된 자바 클래스를 말한다.
- JavaBean 규격서에 따르면 JavaBean 클래스에는 파라미터 없는 생성자가 반드시 있어야 하고, 이 클래스 외부에서 필드(클래스의 멤버 변수)에 접근할 필요가 있을 때는 직접 접근할 수 없고, 반드시 메서드를 통해서 접근해야 한다.

[예제 7-15] 자바빈 클래스의 예

```
package mall;
public class ProductInfo {
    private String name;
    private int value;
    public ProductInfo() {
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getName() {
        return name;
    }
    public void setPrice(int price) {
        value = price;
    }
    public int getPrice() {
        return value;
    }
}
```



3. 익스프레션 언어의 연산자

❖ 대괄호 연산자와 마침표 연산자

- EL 식에서 자바빈 프로퍼티의 값을 가져올 때는 프로퍼티 이름을 사용해야 한다. 즉, 자바빈 객체의 이름 뒤에 마침표를 찍고 그 다음에 프로퍼티 이름을 쓰거나, 자바빈 객체의 이름 뒤에 대괄호를 치고 그 안에 따옴표로 묶은 프로퍼티 이름을 써야 한다.

`${bean.price}`

↑ ↙
자바빈 객체 프로퍼티 이름

`${bean["price"]}`

↑ ↙
자바빈 객체 프로퍼티 이름



3. 익스프레션 언어의 연산자

❖ 대괄호 연산자와 마침표 연산자

[예제 7-16] 자바빈의 프로퍼티를 출력하는 EL 식의 예

```
<%@page contentType= "text/html; charset=euc-kr" %>
<%@page import= "mall.ProductInfo" %>
<%
    ProductInfo product = new ProductInfo();
    product.setName( "초코케이크 3호" );
    product.setPrice(20000);
    request.setAttribute( "PRODUCT", product);
    RequestDispatcher dispatcher = request.getRequestDispatcher( "ProductInfoView.jsp" );
    dispatcher.forward(request, response);
%>
```

자바빈 객체를 애트리뷰트
형태로 저장합니다

호출

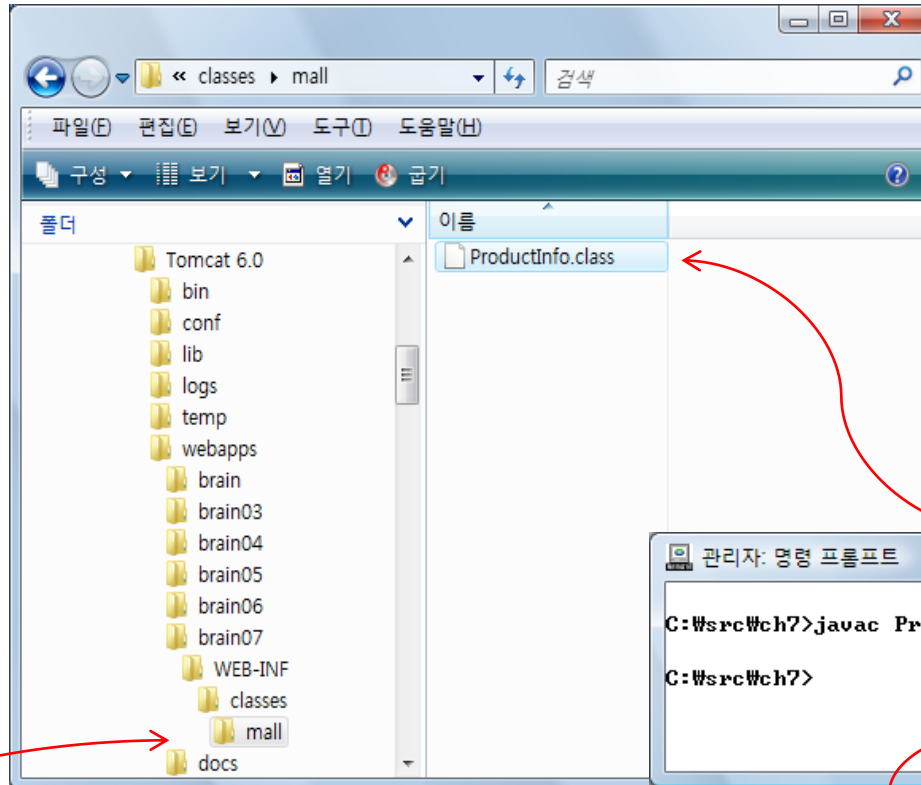
```
<%@page contentType= "text/html; charset=euc-kr" %>
<HTML>
  <HEAD><TITLE>상품 정보</TITLE></HEAD>
  <BODY>
    <H3>상품 정보</H3>
    상품명: ${PRODUCT.name} <BR>
    가격: ${PRODUCT.price} 원 <BR>
  </BODY>
</HTML>
```

자바빈의 프로퍼티를 가져다가
출력하는 EL 식

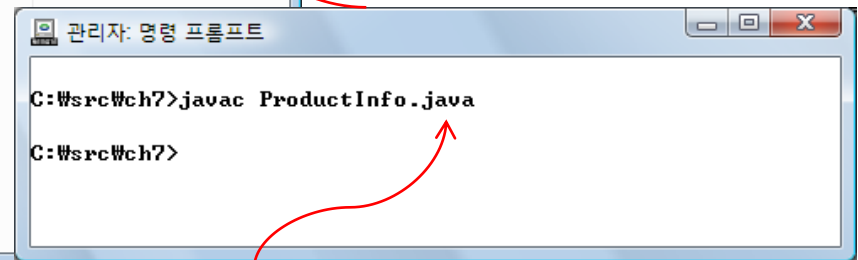


3. 익스프레션 언어의 연산자

❖ 대괄호 연산자와 마침표 연산자



③ (예제 7-15)의 컴파일 결과를 mall 디렉터리에 저장하세요



② (예제 7-15)를 컴파일하세요.

① brain07/WEB-INF 디렉터리 아래에 classes 디렉터를 만들고, 그 아래에 mall 디렉터를 만드세요



Thank You !

뇌를 자극하는 JSP & Servlet