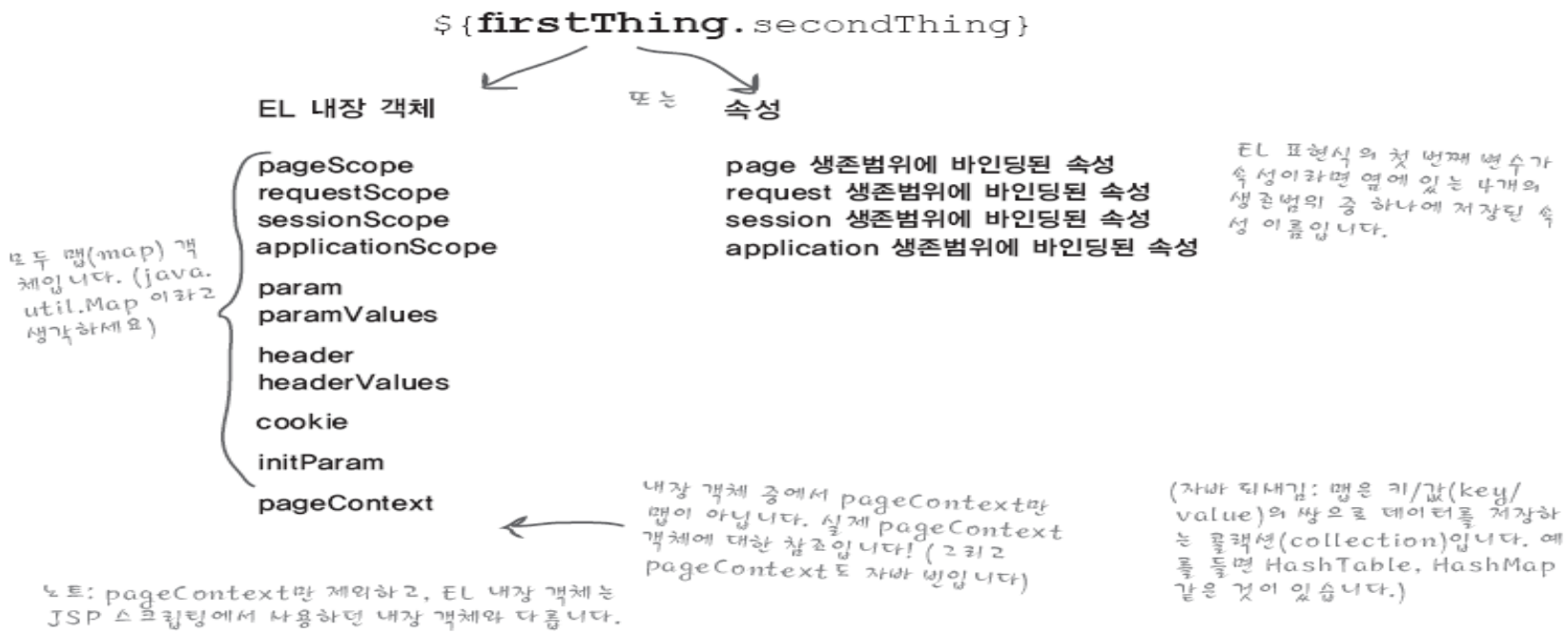
10. EL

EL

- Expression Language
- JSP2.0 스펙에 추가
- 빈즈,맵,리스트,배열 유형의 객체의 속성이나 원소를 접근하여 화면에 출력할수록 있도록 하는 기능을 제공한다.
- 스크립팅 코드를 대신할 수 있다.
- JSTL과 함께 쓸 경우 많은 스크립팅 코드를 대신 할 수 있어서 유용하다.

EL

- SYNTAX
  ${ 표현식 }



EL 내장객체

pageScope
requestScope
sessionScope
applicationScope

생존범위 속성 맵

param
paramValues

요청 파라미터 맵

header
headerValues

요청 헤더 맵

cookie

오~우, 좀 어려운 놈이군요. 이것도 쿠키 맵인가요?

initParam

컨텍스트 초기화 파라미터 맵(서블릿 초기화 파라미터 아님)

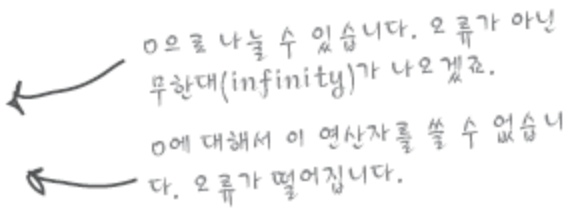
pageContext

맵이 아닌 유일한 녀석. 얘만 실제 pageContext 객체에 대한 참조입니다. 그리고 이건 빈입니다. API 문서에서 pageContext 접근자(Getter)가 어떤 것이 있는지 한번 보세요.*

EL 기본 연산자

산술 연산자 (5)

더하기:	+
빼기:	-
곱하기:	*
나누기:	/와 div
나머지:	%와 mod



논리 연산자 (3)

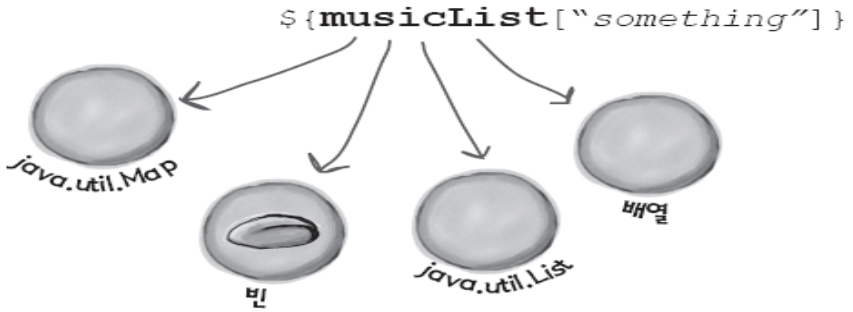
AND:	&&와 and
OR:	||와 or
NOT:	!와 not

관계 연산자 (6)

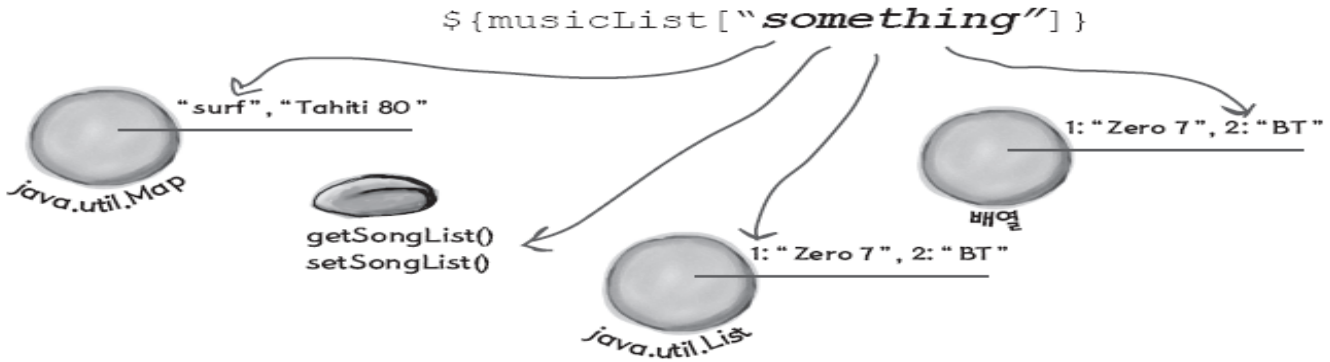
등호:	==와 eq
부등호:	!=와 ne
~보다 작다:	<와 lt
~보다 크다:	>와 gt
~보다 작거나 같다:	<=와 le
~보다 크거나 같다:	>=와 ge

브래킷([]) 연산자

① [ ] 연산자의 왼편에는 맵, 빈, 배열, 리스트 변수가 올 수 있습니다.

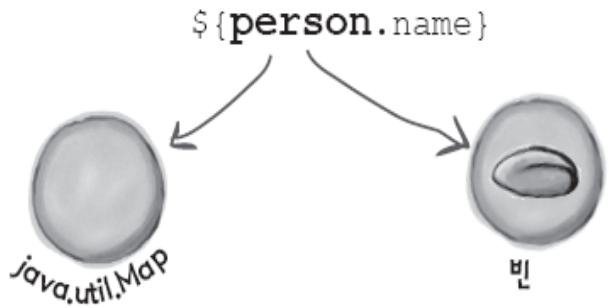


② [ ] 연산자 안의 값이 문자열(따옴표로 묶여 있다면)이라면, 이것은 맵 키가 될 수 있고, 빈 프로퍼티 또는 리스트나 배열 인덱스가 될 수 있습니다.

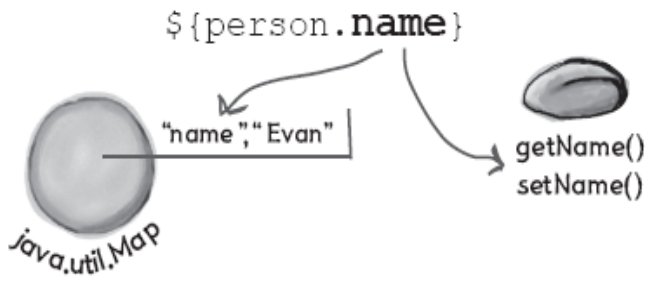


도트(.) 연산자

① 표현식에서 도트 연산자 왼쪽은 반드시 맵 또는 빈이여야 합니다.



② 표현식에서 도트 연산자 오른쪽은 반드시 맵의 키이거나 빈 프로퍼티여야 합니다.



③ 오른쪽에 오는 값은 식별자로서 일반적인 자바 명명 규칙을 따라야 합니다.

${person.name}

* 문자, _, $로 시작해야 합니다.
* 두 번째 글자부터 숫자를 써도 무방합니다.
* 자바 예약어(키워드)는 사용할 수 없습니다.

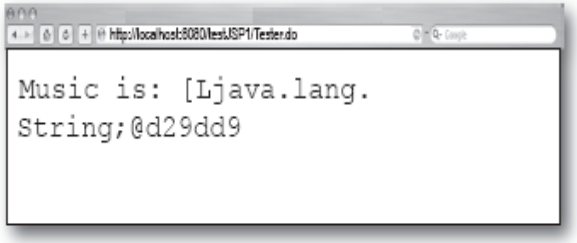
티로 배열 다루기

서블릿 코드

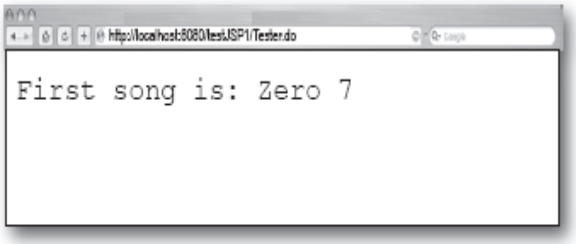
```
String[] favoriteMusic = {"Zero 7", "Tahiti 80", "BT", "Frou Frou"};
request.setAttribute("musicList", favoriteMusic);
```

JSP 코드

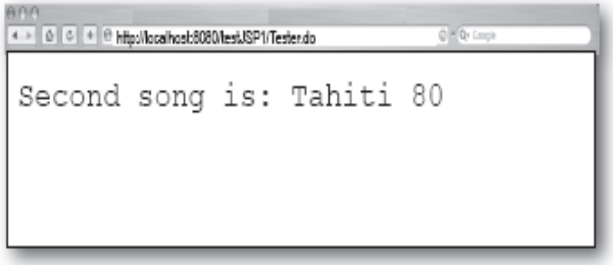
```
Music is:  ${musicList}
```




```
First song is: ${musicList[0]}
```




```
Second song is: ${musicList["1"]}
```



티로 Map 다루기

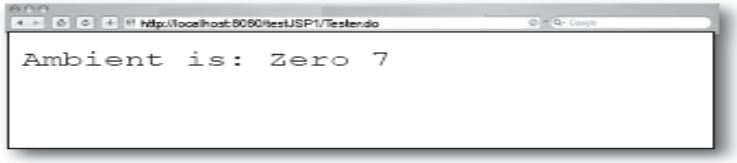
서블릿 코드

```
java.util.Map musicMap = new java.util.HashMap();
musicMap.put("Ambient", "Zero 7");
musicMap.put("Surf", "Tahiti 80");
musicMap.put("DJ", "BT");
musicMap.put("Indie", "Travis");
request.setAttribute("musicMap", musicMap);
```

맵을 만든 다음, String인 키와 객체를 설정합니다. 마지막으로 이를 request 속성에 넣어 두는 거죠.

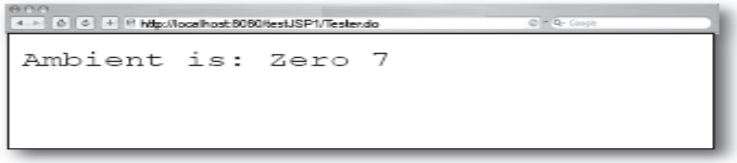
JSP 코드

```
Ambient is: ${musicMap.Ambient}
```



두 표현식 모두 "Ambient"를 맵 키로 처리합니다(musicMap은 맵입니다!)

```
Ambient is: ${musicMap["Ambient"]}
```



EL null처리

EL	출력값
${foo}	
${foo[bar]}	
${bar[foo]}	
${foo.bar}	

출력값이 없습니다. 예를 들어 "출력값은: ${foo} 입니다"라고 코딩하면, "출력값은: 입니다"라고 나오겠죠

${7 + foo}	7
${7 / foo}	무한대
${7 - foo}	7
${7 % foo}	예외가 떨어집니다

EL 산술 연산에서는 널(null)을 0으로 취급합니다.

EL	출력값
${7 < foo}	false
${7 == foo}	false
${foo == foo}	true
${7 != foo}	true
${true and foo}	false
${true or foo}	true
${not foo}	true

EL 논리 연산에서는 정의되지 않은 값은 "거짓(false)"으로 처리합니다