

EL 표현식



EL에서 알아야 할 4가지

- EL 이란?
- EL 에서 사용하는 연산자
- EL에서 객체 찾기
- EL에서의 객체 표현 이해



EL이란?

- Expression Language
- 표현식(<%= %>) 대신 사용
- 자바의 클래스 메소드 호출 및 집합 객체에 대한 접근 방법을 제공
- JSP 에서 제공하는 내장객체 중 4가지 기본 영역에 대한 속성 사용
- 구문
: \$ { expr }
- 사용 예
: 당신의 이름은 \${ name } 입니다.

EL에서 사용하는 연산자

- 연산자

연산자	종류
산술 연산자	+, -, *, /, div, %, mod
비교 연산자	==, eq, !=, ne, <, lt, <=, le, >, gt, >=, ge
논리 연산자	&&, and, , or, !, not
empty 연산자	empty 값이 null 이면 true 비어있는 문자열이면 true 배열이나 컬렉션 일 경우 길이가 0이면 true
삼항 연산자	(조건식) ? 값1 : 값2 조건식이 참이면 값1 선택 거짓이면 값2 선택

EL에서 객체 찾기

- `${ 객체이름 }` → `${ id }`

- 객체 찾는 순서

1. EL에서 기본적으로 제공하는 객체에서 동일한 객체이름을 찾는다.

기본 객체		설명
pageContext		JSP의 page 기본 객체와 동일하다.
영역	pageScope	pageContext 기본객체 영역 접근
	requestScope	request 기본객체 영역 접근
	sessionScope	session 기본객체 영역 접근
	applicationScope	application 기본객체 영역 접근
param		파라미터 매핑 객체, request.getParameter("id") 대체
paramValues		파라미터 매핑 객체, request.getParameterValues("id") 대체
cookie		Cookie 접근 객체

EL에서 객체 찾기

2. EL에서 기본적으로 제공하는 객체에서 찾지 못한다면,
4가지 공유영역에서 객체를 찾는다.

`${ id }`

name

홍길동

id

hong1234

name

임꺽정

userInfo

sbc:1234

JSP

pageContext

request

session

application

EL

pageScope

requestScope

sessionScope

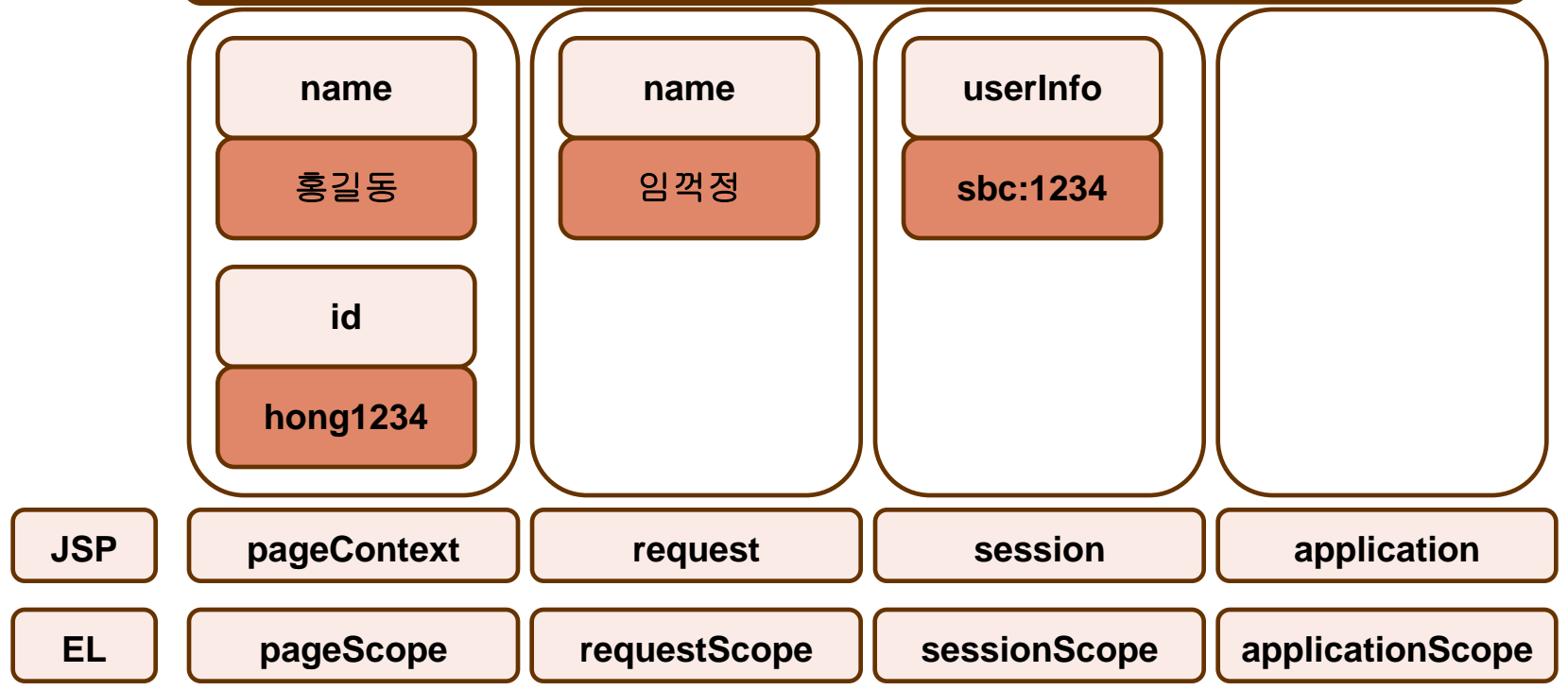
applicationScope

EL에서 객체 찾기

3. 만약, 공유영역에 찾는 객체이름이 중복되어 있다면,

pageContext → request → session → application 순으로 찾는다.

만약, 임꺽정이 나와야 한다면?? → `${requestScope.name}`



EL에서 객체 찾기

4. 공유영역에 객체 등록하기

```
request.setAttribute ( "name", "임꺽정" );  
session.setAttribute ( "userInfo", "sbc:1234" );
```

name

홍길동

id

hong1234

name

임꺽정

userInfo

sbc:1234

JSP

pageContext

request

session

application



EL에서의 객체 표현 이해

1. \$ { 객체 . 이름 }
 - 객체 자리에 올 수 있는 타입은 2가지만 가능
 - 자바빈즈 클래스, 맵 객체

EL에서의 객체 표현 이해

* 자바빈즈 클래스 형태

- 1) 패키지로 선언
- 2) 클래스는 public 선언
- 3) 매개변수 없는 생성자 선언
- 3) 멤버변수는 private 선언
- 4) 멤버변수에 대한 setter, getter

```
package test;  
  
public class Board {  
    private int no;  
    public void setNo (int no) {  
        this.no = no;  
    }  
    public int getNo ( ) {  
        return no;  
    }  
}
```

EL에서의 객체 표현 이해

1. \$ { 객체 . 이름 }

- 객체 타입이 자바빈즈인 경우
- \$ { board . no } → no는 board 의 get 메소드를 의미

```
package test;  
public class Board {  
    private int no;  
    public int getNo ( ) {  
        return no;  
    }  
}
```

```
Board b = new Board();  
pageContext.setAttribute("board", b);
```

\$ { board . no }



EL에서의 객체 표현 이해

1. \$ { 객체 . 이름 }

- 객체의 타입이 맵인 경우
- \$ { board . no } → no는 맵의 key 가 no를 의미

Map

key와 value 로 묶인 구조

key	value
no	123
title	EL

```
Map b = new HashMap();
```

```
b.put("no", "123");
```

```
b.put("title", "EL");
```

```
pageContext.setAttribute("board", b);
```

```
$ { board . no }
```

123

EL에서의 객체 표현 이해

2. `${ 객체 [이름] }`

- 객체 자리에 올 수 타입은 4가지만 가능
 - 자바빈즈 클래스, 맵 객체, **배열**, 컬렉션

```
String [] names = { “홍길동”, “임꺽정”, “강감찬” };  
pageContext.setAttribute(“names”, names);
```

`${ names [0] }` → 홍길동

`${ names [1] }` → 임꺽정

`${ names [2] }` → 강감찬

EL에서의 객체 표현 이해

2. `${ 객체 [이름] }`

- 객체 자리에 올 수 있는 타입은 4가지만 가능
 - 자바빈즈 클래스, 맵 객체, 배열, **컬렉션**

```
List nameList = new ArrayList();
```

```
nameList.add( “홍길동” );
```

```
nameList.add( “임꺽정” );
```

```
pageContext.setAttribute ( “names”, nameList );
```

```
${ names [ 0 ] }   ➔   홍길동
```

```
${ names [ 1 ] }   ➔   임꺽정
```