EL操作符

1.使用点号”.” 访问性质和映射值

${person.name}

第一个变量可以是一个隐式对象，也可以是一个属性

如果EL表达式中变量后边有点号：

（1）点号左边必须是一个映射值（某范围的属性）或一个bean的性质

（2）点号右边必须是一个映射值或一个bean性质

右边如果是映射值，那么会从Map中取到对应映射值，如果是bean性质，那么会调用bean的getXXX方法取到bean性质。（EL解析程序中进行了复杂的判断）

右边的值遵循Java命名规范。

**2. [] 是更好的点号**

当点号右边是左边的一个bean性质或者属性值时，点操作符才能工作，但[]右边可以是一个变量，可以灵活赋值，复用代码

${ person[“name”] } 等价于 ${ person.name }

（1） 如果EL使用[]，那么左边的变量不限定于**bean**和**Map**，还可以是**List**或**数组**。

（2）如果[]中是一个String直接量（即用引号引起的字符串），那么这可以是一个Map的key，或是一个bean的性质，或是list或数组的index

${ musicList[“something”] }

something可以是index，很关键兄弟，这样就可通过index取元素。

但是，这里[]并不是数组访问符，它只是提供了相同的方法，要知道我们放入的加引号的可是String，数组和List使用[]会把这String强转为int。

也因此，使用数组或list时，index可以不加引号

**当musicList是数组或List时， ${ musicList[0] } 和 ${ musicList[“0”] } 都是对的**

**3.如果[]中不是String直接量，在JSP转Servlet的过程中间，就会进行计算（跟普通的计算不是一个意思）**

比如，我们在request.setAttribute(“gen”, “Amber”)

计算为

${ musicMap[Gen] } ---------------🡪 ${ musicMap[“Amber”] }

即自动找到属性的值，这个值再作为musicMap的key

而 不变

${ musicMap[“Gen”] } -------------🡪 ${ musicMap[“Gen”] }

语法没问题，但这和我们预想的不一样，不能这样写

因此，如果要查找Amber，就不能${ musicMap[Amber] }，因为并没有Amber这个属性，它只是属性值（注意，这里的属性是指四大作用域的属性）

4.[]可以嵌套，会从最里层开始计算

5.点号后面必须是合法的Java变量名，不能以数字开头(可以是字母、\_或$)，不能是Java标识符

${ person.1 }是非法的