

1. EL: **Expression Language**——表达式语言，
2. 语法格式: **`${EL Expression}`** 典型标记: **\$** 和 **{ }**，两者之间不能空格分家
3. 读取方式使用 **.** 和 **[]**，多用来做读取、运算等操作。
 点和中括号使用的相同点：大多数情况下通用。区别：
对于连字符（-）或者其它特殊字符的字符串只能用 “[]” 操作符，而不能用 “.” 操作符。
 不建议用来设定四个属性范围的属性值, 如果确实要使用，可以在 EL 表达式中使用赋值符号，即可！但是仍然不能为 Java 的变量赋值。
4. JSP EL 表达式用于以下情形：
 - A. 静态文本
 - B. 标准标签和自定义标签，eg：


```
<body bgcolor=${pageScope.color}>
<h1>测试</ht>
</body>
```
 - C. **EL 不能在脚本元素中使用**——不可以在 Java 的代码段中使用 EL 表达式。
5. 禁用 EL 计算的语法为：有时为了兼顾旧版本 JSP
`<%@ page isELIgnored = "true" %>`
6. EL 表达式的操作内容可以是

常量： eg: `${true}` `${10}` `${10.5}` `${ "hello" }` `${null}`

变量： `${object.attribute}` `${object.attribute.attribute}`

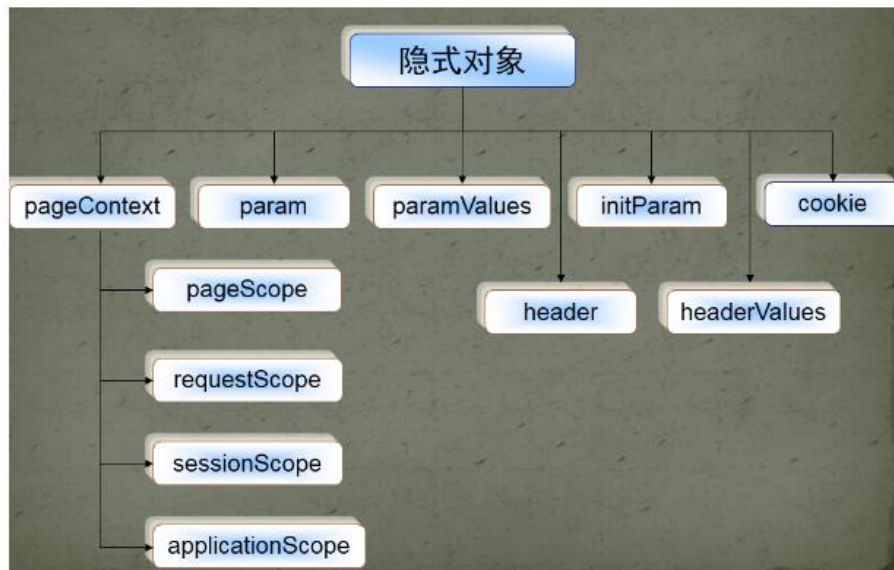
当 EL 在对表达式中的变量进行操作的时候，它通过 `pageContext.findAttribute(“ ”)` 的方式来查找变量，查找的范围从 `page`, `request`, `session`, `application`，如果这几个范围都没有找到则返回 `null`。
 也可以通过以下的内容来指定范围：
`pageScope`, `requestScope`, `sessionScope`, `applicationScope`.

属性范围	在 EL 中的名称
Page	PageScope
Request	RequestScope
Session	SessionScope
Application	ApplicationScope

点运算符 (.) 或方括号 [] 访问变量的值.

变量类项的自动转换，转换不成功则报错。

jsp 隐含对像：



7. EL 表达式的运算:

A. 基础运算:

基础运算	基础运算
<code><p>基础运算</p></code>	
<code><hr/></code>	
<code>5+3: \${5+3}</code>	
<code>
</code>	
<code>5-3: \${5-3}</code>	5+3: 8
<code>
</code>	5-3: 2
<code>5*3: \${5*3}</code>	5*3: 15
<code>
</code>	5/3: 1.6666666666666667
<code>5 / 3: \${5 div 3}</code>	5 div 3: 1.6666666666666667
<code>
</code>	
<code>5%3: \${5%3}</code>	5%3: 2
<code>
</code>	
<code>5mod3: \${5 mod 3}</code>	5mod3: 2

B. 关系运算:

<p>关系运算</p>

```
<span>7 == 5 :</span>${ 7==5}<br/>
<span>7 eq 5 :</span>${ 7 eq 5}<br/>
<span>7 ne 5 :</span>${ 7 ne 5}<br/>
<span>7 > 5 :</span>${ 7>5}<br/>
<span>7 gt 5 :</span>${ 7 gt 5}<br/>
<span>7 &ge; 5 :</span>${ 7 gt 5}<span>****</span><br/>
<span>7 >= 5 :</span>${ 7 >= 5}<br/>
<span>7 ge 5 :</span>${ 7 ge 5}<br/>
<span>7 < 5 :</span>${ 7 < 5}<br/>
<span>7 &lt; 5 :</span>${ 7 < 5}<br/>
<span>7 lt 5 :</span>${ 7 lt 5}<br/>
<span>7 <= 5 :</span>${ 7 <= 5}<br/>
<span>7 &le; 5 :</span>${ 7 <= 5}<br/>
<span>7 le 5 :</span>${ 7 le 5}<br/>
```

关系运算

```
7 == 5 :false
7 eq 5 :false
7 ne 5 :true
7 > 5 :true
7 gt 5 :true
7 &ge; 5 :true****
7 >= 5 :true
7 ge 5 :true
7 < 5 :false
7 &lt; 5 :false
7 lt 5 :false
7 <= 5 :false
7 &le; 5 :false
7 le 5 :false
```

C. 逻辑运算:

<p>逻辑运算</p>

<hr/>

```
<span>true and true:</span> ${true and true}<br/>
<span>true && true:</span> ${true && true}<br/>
<span>true || true:</span> ${true || true}<br/>
<span>true or true:</span> ${true or true}<br/>
<span>true:</span> ${true}<br/>
<span>not true:</span> ${not true}<br/>
<span>! true:</span> ${! true}<br/>
<span>true and false:</span> ${true and false}<br/>
<span>true && false:</span> ${true && false}<br/>
<span>true || false:</span> ${true || false}<br/>
<span>true or false:</span> ${true or false}<br/>
```

逻辑运算

```
true and true: true
true && true: true
true || true: true
true or true: true
true: true
not true: false
! true: false
true and false: false
true && false: false
true || false: true
true or false: true
```

D. 三目运算:

<p>三目运算符: </p>

```
<span>${ '{' }|A? 0:1 }</span>${ |A? 0:1 }<br/>
<span>${ '{' }A? 0:1 }</span>${ A? 0:1 }<br/>
<span>${ '{' }A==true? 0:1 }</span>${ A==true ? 0:1 }<br/>
<span>${ '{' }A!=true? 0:1 }</span>${ A!=true ? 0:1 }<br/>
<span>${ '{' }A=true? 0:1 }</span>${ A=true ? 0:1 }<br/>
<span>${ '{' }A=(true? 0:1) }</span>${ A=(true ? 0:1) }<br/>
<span>${ '{' }A=0? 0:1 }</span>${ A=0 ? 0:1 }<br/>
<span>${ '{' }A==0? 0:1 }</span>${ A==0 ? 0:1 }<br/>
```

三目运算符:

```
${ !A? 0:1 }0
${ A? 0:1 }1
${ A==true? 0:1 }1
${ A!=true? 0:1 }0
${ A=true? 0:1 }0
${ A=(true? 0:1) }0
${ A!=0? 0:1 }1
${ A==0? 0:1 }0
```

E. 判空运算:

<p>空判断: </p>

```
<span>${ '{' } empty '' }</span>${ empty '' }<br/>
<span>${ '{' } empty 'abc' }</span>${ empty 'abc' }<br/>
<span>${ '{' } empty "abc" }</span>${ empty "abc" }<br/>
```

空判断:

```
${ empty '' }true
${ empty 'abc' }false
${ empty "abc" }false
```

8. EL 表达式的操作结果会自动输出到输出流对象。

9. 用来取得用户请求的参数, 如:

`${param.name}` ----`request.getParameter(String name)`

`${paramValues.name}` ----`request.getParameterValues(String name)`

10. 读取 Cookie:

A. 设置 Cookie:

`<title>Insert title here2</title>`

`<%`

`Cookie cookie1 = new Cookie("country","chn");`

`cookie1.setMaxAge(120);`

`response.addCookie(cookie1);`

`%>`

`</head>`

B. 读取 Cookie:

`<hr />`

`${cookie.country.name}
`

`${cookie.country.value}
`

`${cookie['country'].name }
`

`${cookie['country'].value }
`

`<hr />`

11. 不要使用 EL 中的保留字作为变量名称!

12. EL 对容器元素的读取:

A. 数组----数组的取值, 只能使用[]和索引来取值!

eg: `${strArray[0]}`

B. LIST----list 只能通过索引获取

eg: `${list[0]}` `${list.get(0)}`

C. MAP----map 获取元素是通过键 Key!

eg: `${map["11"]} ${map.get("13')}`

13. Web 项目发现显示与自己预期不一样，从下面几方面入手

- A. F5，刷新页面；
- B. Ctrl + F5；强制浏览器刷新；
- C. Ctrl + Shift +Delet，清空浏览器的缓存；
- D. 重启 Tomcat/Jetty 等服务器，重新加载配置文件；
- E. Clean Project，让 Eclipse 对项目使用源码重新编译；
- F. 若项目是 Maven 项目，重编编译 Maven 项目的源码；
- G. 怀疑自己的代码单词拼写或逻辑等错误！