

一，四大域对象：实际开发

PageContext: pageContext存放的数据在当前页面有效，开发时使用较少

ServletRequest: request存放的数据在一次请求（转发）内有效，使用非常多

HttpSession: session存放的数据在一次会话中有效，使用的比较多，如：存放用户的登录信息，购

购物车功能

ServletContext: application存放的数据在整个应用范围内都有效，因为范围太大，应该尽量少用

二，EL表达式

el表达式: expression language表达式语言

要简化jsp中java代码开发

它是一种开发语言，是jsp中获取数据的一种规范

EL的具体功能

1，获取数据

el表达式只能获取存在4个作用域中的数据

`${u}` 原理: `pageContext.findAttribute("u")`

el获取对于null这样的数据，在页面中表现为空字符串“ ”

`${u.name}` == `u.getName()` 方法

点(.) 运算符相当于调用了getter方法，点后面跟的是属性名

属性导航: `${stu.city.address}`

[]运算符，点能做到的，[]也能做到，它能做到的，点不一定能做到

`${student.name}` == `${student['name']}` == `${student["name"]}`

如：一般括号都好使，当点使用不好用的时候，就使用括号

<%

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

```
list.add("aaa");
```

```
list.add("bbb");
```

```
list.add("ccc");
```

```
request.setAttribute("lists", list);
```

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

```
map.put("A", "aaaa");
```

```

map.put("B", "bbbb");
map.put("C", "cccc");
request.setAttribute("m", map);
%>
${lists[1]} 获取出bbb
<br/>
${m["B"]} 通过m获取出bbbb
<br/>
${m.B } 获取出bbbb

```

2, 简单运算

①empty判断null，空字符串和没有元素的集合（即使集合对象本身不为null）都返回true

只要没有数据，都是null

例如：

使用empty的时候，null或者空字符串或者空的集合都是返回true


```

<%
String string1 = null;
request.setAttribute("string1", string1);
String string2 = "";
request.setAttribute("string2", string2);
String string3 = "abc";
request.setAttribute("string3", string3);
List list1 = new ArrayList();
request.setAttribute("lis1", list1);
List lis2 = new ArrayList();
lis2.add("absx");
request.setAttribute("list2", lis2);
%>
${empty(string1) }----true
<br/>
${empty(string2) }----true
<br/>
${empty(string3) }----false
<br/>

```

```

    ${empty(list1) }----true
<br/>
    ${empty(list2) }---- false
<br/>

```

②三元运算符

`${empty list2 ? "您还没有购买任何商品呢！" : "您购买的商品如下："}`

如：设置那个默认被选中

```

<%
int sex = 1;
request.setAttribute("s", sex);
%>
<input type = "radio" name = "sex" value = "m"
${s==0?"checked='checked' ":""} />男<br/>
<input type = "radio" name = "sex" value = "f"
${s==1?"checked='checked' ":""} />女<br/>

```

3, 11个隐式对象

(其中蓝色字体为重点掌握的)

| EL隐式对象引用名称 | 类型 | jsp内置对象 | 说明 |
|-------------------------------|--|--------------------------|---|
| <code>pageContext</code> | <code>javax.servlet.jsp.PageContext</code> | <code>pageContext</code> | 一样的 |
| <code>pageScope</code> | <code>java.util.Map<String, object></code> | 没有对应的 | <code>pageContext</code> 范围中存放的数据 页面范围 |
| <code>requestScope</code> | <code>java.util.Map<String, Object></code> | 没有对应的 | 请求范围数据 |
| <code>sessionScope</code> | <code>java.util.Map<String, Object></code> | 没有对应的 | 会话范围数据 |
| <code>applicationScope</code> | <code>java.util.Map<String, Object></code> | 没有对应的 | 应用范围数据 |
| <code>param</code> | <code>java.util.Map<String, String></code> | 没有对应的 | 一个请求参数 |
| <code>paramValues</code> | <code>java.util.Map<String, String[]></code> | 没有对应的 | 重名请求参数 |
| <code>header</code> | <code>java.util.Map<String, String></code> | 没有对应的 | 一个请求消息头 |

| | | | |
|------------------------------|---------------------------------|-------|---------|
| headerValues 消息头 | java.util.Map<String, String[]> | 没有对应的 | 重名请求 |
| initParam 中全局参数 | java.util.Map<String, String> | 没有对应的 | web.xml |
| cookie key: cookie对象的name | java.util.Map<String, Cookie> | 没有对应的 | |

前四个:

```
<%
pageContext.setAttribute("p", "pp");
request.setAttribute("p", "request");
session.setAttribute("p", "session");
application.setAttribute("p", "application");
%>
```

指定作用域进行取值


```
${pageScope.p }
<br/>
${requestScope.p }
<br/>
${sessionScope.p }
<br/>
${applicationScope.p }
```

param: (表单数据加上param, 使其变为getParameter, 否则底层代码是getAttribute)

```
${param.userName}===request.getParameter("userName");
```

cookie中取值:

取出cookie中的id值


```
${cookie.JSESSIONID.value }
<br/>
```

pageContext:

```
${pageContext.request.contextPath }得到当前页面中的信息
```

要想得到其他八个对象，需要通过pageContext进行点