

# JSP 隐式对象

## JSP 隐式对象

JSP 隐式对象是 JSP 容器为每个页面提供的 Java 对象，开发者可以直接使用它们而不用显式声明。JSP 隐式对象也被称为预定义变量。

JSP 所支持的九大隐式对象：

对象	描述
request	<code>HttpServletRequest</code> 类的实例
response	<code>HttpServletResponse</code> 类的实例
out	<code>PrintWriter</code> 类的实例，用于把结果输出至网页上
session	<code>HttpSession</code> 类的实例
application	<code>ServletContext</code> 类的实例，与应用上下文有关
config	<code>ServletConfig</code> 类的实例
pageContext	<code>PageContext</code> 类的实例，提供对 JSP 页面所有对象以及命名空间的访问
page	类似于 Java 类中的 <code>this</code> 关键字
Exception	<code>Exception</code> 类的对象，代表发生错误的 JSP 页面中对应的异常对象

## request 对象

request 对象是 `javax.servlet.http.HttpServletRequest` 类的实例。每当客户端请求一个 JSP 页面时，JSP 引擎就会制造一个新的 request 对象来代表这个请求。

request 对象提供了一系列方法来获取 HTTP 头信息，cookies，HTTP 方法等等。

---

## response 对象

response 对象是 `javax.servlet.http.HttpServletResponse` 类的实例。当服务器创建 request 对象时会同时创建用于响应这个客户端的 response 对象。

response 对象也定义了处理 HTTP 头模块的接口。通过这个对象，开发者们可以添加新的 cookies，时间戳，HTTP 状态码等等。

---

## out 对象

out 对象是 `javax.servlet.jsp.JspWriter` 类的实例，用来在 response 对象中写入内容。

最初的 JspWriter 类对象根据页面是否有缓存来进行不同的实例化操作。可以在 page 指令中使用 `buffered='false'` 属性来轻松关闭缓存。

JspWriter 类包含了大部分 `java.io.PrintWriter` 类中的方法。不过，JspWriter 新增了一些专为处理缓存而设计的方法。还有就是，JspWriter 类会抛出 `IOExceptions` 异常，而 `PrintWriter` 不会。

下表列出了我们将会用来输出 `boolean`，`char`，`int`，`double`，`String`，`object` 等类型数据的重要方法：

方法	描述
<code>out.print(dataType dt)</code>	输出 Type 类型的值
<code>out.println(dataType dt)</code>	输出 Type 类型的值然后换行
<code>out.flush()</code>	刷新输出流

---

## session 对象

session 对象是 `javax.servlet.http.HttpSession` 类的实例。和 Java Servlets 中的 session 对象有一样的行为。

session 对象用来跟踪在各个客户端请求间的会话。

---

## application 对象

application 对象直接包装了 servlet 的 `ServletContext` 类的对象，是 `javax.servlet.ServletContext` 类的实例。

这个对象在 JSP 页面的整个生命周期中都代表着这个 JSP 页面。这个对象在 JSP 页面初始化时被创建，随着 `jspDestroy()` 方法的调用而被移除。

通过向 application 中添加属性，则所有组成您 web 应用的 JSP 文件都能访问到这些属性。

---

## config 对象

config 对象是 `javax.servlet.ServletConfig` 类的实例，直接包装了 servlet 的 `ServletConfig` 类的对象。

这个对象允许开发者访问 Servlet 或者 JSP 引擎的初始化参数，比如文件路径等。

以下是 config 对象的使用方法，不是很重要，所以不常用：

```
config.getServletName();
```

它返回包含在 <servlet-name> 元素中的 servlet 名字，注意，<servlet-name> 元素在 WEB-INF\web.xml 文件中定义。

---

## pageContext 对象

pageContext 对象是 javax.servlet.jsp.PageContext 类的实例，用来代表整个 JSP 页面。

这个对象主要用来访问页面信息，同时过滤掉大部分实现细节。

这个对象存储了 request 对象和 response 对象的引用。application 对象，config 对象，session 对象，out 对象可以通过访问这个对象的属性来导出。

pageContext 对象也包含了传给 JSP 页面的指令信息，包括缓存信息，ErrorPage URL, 页面 scope 等。

PageContext 类定义了一些字段，包括 PAGE\_SCOPE，REQUEST\_SCOPE，SESSION\_SCOPE，APPLICATION\_SCOPE。它也提供了 40 余种方法，有一半继承自 javax.servlet.jsp.JspContext 类。

其中一个重要的方法就是 removeAttribute()，它可接受一个或两个参数。比如，

pageContext.removeAttribute("attrName") 移除四个 scope 中相关属性，但是下面这种方法只移除特定 scope 中的相关属性：

```
pageContext.removeAttribute("attrName", PAGE_SCOPE);
```

---

## page 对象

这个对象就是页面实例的引用。它可以被看做是整个 JSP 页面的代表。

page 对象就是 this 对象的同义词。

---

## exception 对象

exception 对象包装了从先前页面中抛出的异常信息。它通常被用来产生对出错条件的适当响应。