获取表单提交的数据

- 请求对象：request
  - public string getParameter(String name)

示例

- HTML代码：<input name="username">——数据对应关系
- jsp代码获取该值：<% String name = request.getParameter("usename"); %>

- Get和Post请求区别

	Get	Post
参数在url中	是	否
长度限制	1~2M(不同浏览器不同)	无
安全性	不安全	安全
URL可收藏性	可收藏	否

Get请求速度高于Post请求

- 表单数据获取

一、下拉框数据获取

```
<select>
  <option value="">--请选择--</option>
  <op... value="yueyang">岳阳</op...>
  <... value="zhangsha">长沙</...>
</select>
```

通过 request.getParameter("...")来接收参数

二、单选框

```
<input type='radio' name='sex' value='1'>男
<input ... 'radio' name='sex' value='2'>女
```

通过request.getParameter('...')来接收，若男、女两个选项均未点击，获取值为null

三、多选框!!!

```
<input type='checkbox' name='hobby' value='sing'>唱
<... 'checkbox' name='hobby' value='drup'>跳
<... 'checkbox' name='hobby' value='rap'>rap
```

通过request.getParameterValues()来接收一个String[]返回值

- request对象常用方法

方法名称	说明
String getParameter(String name)	根据表单组件名称获取提交数据
String[] getParameterValues(String name)	获取表单组件对应多个求数据
void setCharacterEncoding(String charset)	指定每个请求的编码
RequestDispatcher getRequestDispatcher(String path)	返回一个RequestDispatcher象，该对象的forword(request,response)用于转发请求

0 内置对象-jsp（9个）

- 请求对象： request
- 输出对象： out
- 响应对象： response
- 应用程序对象： application
- 会话对象： session
- 页面上下文对象： pageContext
- 页面对象： page
- 配置对象： config
- 异常对象： exception

0 中文乱码

- Get 不会乱码（tomcat 8后不会乱码，下会乱码）
- post 中文会乱码

这个编码格式不能任意，比如当前JSP页面指定了编码格式则就要与其一致

```
request.setCharacterEncoding("UTF-8");//要放在获取数据的java语句之前
response.setCharacterEncoding("UTF-8")
```

//放在第一句，不能放在第二句，否则所有数据

解决中文乱码

◆ 设置请求和响应的编码方式

- request.setCharacterEncoding("utf-8");
- response.setCharacterEncoding("utf-8");
- <%@ page language="java" contentType="text/html; charset=utf-8"%>

一致

◆ get 请求出现乱码

- 治标的方法： new String(s.getBytes("iso-8859-1"), "utf-8");
- 治本的方法：配置 tomcat\conf\server.xml 文件
  - URIEncoding="UTF-8"
  - useBodyEncodingForURI="true"

```
<Connector connectionTimeout="20000" port="8080"
protocol="HTTP/1.1" redirectPort="8443" URIEncoding="UTF-8" />
```

○页面跳转

一、转发：request.getRequestDispatcher("jsp").forward(request,response)

二、重定向：response.sendRedirect("url")

转发：同一次请求，服务器内部处理，会保留数据(同一个request，相当于将其从左手拿到右手)

重定向：两次请求，将重定用url返回客户端，客户端再次发起一个请求 地址栏会变

○使用session保存用户名

session.setAttribute("用户名","value");
Object

Object v=session.getAttribute("用户名")

○session.invalidate()方法

用于使当前HTTP会话失效，它会删除整个会话及其所有属性

○ session.setMaxInactiveInterval(0) // 设置会话立即过期，效果类似于invalidate()

○application

代表应用程序作用域，是servletContext接口的实例，是web应用中范围最大、生命周期最长的共享数据存储区域

总结：request、session、application在jsp中都是内置对象，其都能用来存储数据

1. request只在本次请求中有效，若为一个请求，含有多个request数据不共享
2. session在本次会话中有效
3. application在web服务器启动期间有效，不论某个客户端都能访问其中数据。