# JavaWeb

## HTML

1. HTML的所有操作都是通过标记实现的。标记就是标签。<标签名称>
2. HTML文件的后缀名为.html
3. <标签名称 属性=” ” 属性=” ” >xxxxx</标签名称>

多个属性用空格分开即可

1. HTML规范：
2. 有开始和结束标签，成对出现；
3. html包含两部分：<head>设置相关信息</head> 和<body>所有显示在网页上的内容</body>
4. html不区分大小写
5. 有些标签只有开始标签，没有结束标签，如换行<br/>，表示在标签内结束。
6. HTML常用标签

|  |  |  |
| --- | --- | --- |
| **标签名称** | **含义** | **属性** |
| <font></font> | 修改字体 | size: 字体大小，范围为1-7；  color: 字体颜色 |
| <!--这里是注释--> | | |
| <h1>标题1</h1>  <h2>标题2</h2>  …..  <h6>标题2</h6> | 显示标题，会自动换行，从h1到h6字体大小会自动变小 |  |
| <hr/>  <hr size=”7” /> | 插入水平线，在标签内结束 | size: 表示水平线的粗线  color: 颜色 |
| &nbsp; | 代表一个空格 |  |
| <ol></ol> | 有序列表的范围 |  |
| <ol>  <li>这里是列表内容</li>  </ol> |  |  |
| <ul type = “clrcle”><li></li></ul> | ul: 无序列表的范围 |  |
| <img src=”图片的路径”/> | src: 图片的路径 width:图片的宽度 height: 图片的高度  alt: 鼠标放到图片上显示的文字，或者当图片不存在时也会显示这些文字 | |
| <a href=”链接到资源的路径” target=\_blank>显示在页面上的内容</a>  target =blank 表示在新页面打开，默认为在当前页面打开  href=”#” 表示不链接到任何内容 | | |
| <pre> </pre> : 原样输出里面的内容，不做换行等操作 | | |
| <a href="name">顶部</a>  <a href="#name">返回顶部</a>  功能：单机“返回顶部”将会会到“顶部”这一位置 | | |
| <table></table> 表示表格的范围，属性有：  border=”1” 表示有表格线 bordercolor: 表格线颜色 cellspacing = "0" 表示单元格之间没有线 width: 宽度 | | |
| <table>  <tr>  <td>一</td><td>二</td>  </tr>  <tr>  <td>三</td><td>四</td>  </tr>  <tr>  <td>五</td><td>六</td>  </tr>  </table>  <tr></tr> 代表一行 <td></td> 代表一个单元格 | | |
| <tr>和<td> 的属性： align="center" 表示居中(left、right) | | |
|  | | |

1. 表单标签

<form></form> : 定义一个表单的范围，属性有：

action: 提交地址

method: 表单的提交方式，有get和post，默认是get. get与post的区别：get提交会携带参数，post不会；get的安全级别较低；get请求数据的大小有限制

<input/> 输入框，其属性有：

type: 输入框的类型（text：文本；password：密码；radio：单选框，注意，有几个单选项就写几个<input type=”radio”/>，并且，需要写上name属性，几个单选项的name属性必须一样，要有一个value属性，用以区分选项；checkbox：多选框，需要写上name属性，几个单选项的name属性必须一样，有几个多选项就写几个<input…../> ，要有一个value属性，用以区分选项；file：文件框）

<input type="submit" value="提交">: 生成一个按钮，名字叫“提交”，点击这个按钮，网页地址栏会发生变化

<input type="image" src="图片的地址"/> 点击图片也可以进行提交

<input type="reset"/> 重置表单中的所有信息

<select></select> 下拉框

<textarea cols="50" rows="10">默认内容</textarea> 文本域

注意：所有的<input> 都需要加入一个name属性

form的提交方式：

1）使用input: <input type="submit" value="提交">: 生成一个按钮，名字叫“提交”，点击这个按钮，网页地址栏会发生变化

2）给一个按钮设置方法：

        <form id ="form1">

            <input type="text" name = "user"/><br/>

            <input type="button" value = "提交" onclick="goto()"/>

        </form>

    <script>

        function goto(){

            var obj = document.getElementById("form1");

            form1.action = "http://www.baidu.com"; //这里更改提交地址

            form1.submit();//会把form的参数进行提交

        }

    </script>

1. 常用标签

|  |  |
| --- | --- |
| <div></div> | 自动换行 |
| <span></span> | 在一行显示 |
| <p></p> | 段落标签，也会换行，比<br/>会多一空行 |
|  |  |

1. html的头标签的使用：在head里面的标签就是头标签

|  |
| --- |
| <title></title> 显示网页的标题 |
| <meta、> 提供页面的基本信息  <meta http-equiv="refresh" content="3;url=http://www.baidu.com"></meta>  表示3秒后，自动跳转到百度首页 |
| <base/> 可以设置超链接的形式  <base target="\_blank"/> 意味这所有的超链接将会在新的页面打开，不用在每一个<a></a>标签进行设置 |

## 二．CSS

1. CSS：层叠样式表(英文全称：Cascading Style Sheets)。CSS将页面内容与显示样式进行了分离，提高了显示功能。

2. CSS和html结合的方式：

1）每个html标签都有一个style属性，通过这个属性可以和html进行联系，如<div style="background-color:blue;color: #000;">内容</div>

属性与属性值之间用 : 隔开，属性与属性之间用 ; 隔开

2) 使用<style>标签进行实现，写在<head>标签里面

<style type="text/css">

div{

background-color:

blue;color:red;

}

</style>

<style type="text/css"></style>是固定不变的，中间写css代码，对哪个进行设置就用{}进行标注

3） <style type="text/css">

@import url(css文件的路径)

</style>

1. 使用头标签<link>

<link rel="stylesheet" type="text/css" href="css文件的路径"/>

3. CSS的优先级：后加载的CSS会覆盖之前加载的

4. CSS的选择器（三种）

1）标签选择器，使用标签名称作为选择器的名称

2）使用class属性的值，每个标签都有一个属性class，比如<div class=”hh”></div>

通过div.hh{} 来表达CSS样式

.hh{}表示所有class值为hh的标签

3）使用id属性的值，每个标签都有一个属性class，比如<div id=”hh”></div>

通过div#hh{} 来表达CSS样式

#hh{}表示所有id值为hh的标签

三种选择器优先级：style>id>class>标签

5. CSS的扩展选择器：

1）关联选择器：如有<div><p>需要设置的样式</p></div>，可表示为div p {}

2）组合选择器：如有<div></div> <p></p>，可表示为div,p {} 这样div和p中的内容可以被设置成一样的样式

6. CSS盒子模型

在进行布局前需要把数据进行封装到一块一块的区域内（一般是div），这个区域的专业术语叫做盒子。

边框（border）：分为上（border-top）、下（border-bottom）、左（border-left）、右（border-right）

内边距（padding）

外边距（margin）

7. CSS布局定位属性 position，其值有

absolute: 将对象从文档流中拖出，使用top, left等进行绝对定位

relative: 不会把对象从文档流中拖出，使用top, left等进行定位

8. Bootstrap: 概念：一个前端开发的框架，定义好了许多css样式和js插件

\*使用方法：将下载好的Bootstrap三个文件夹复制到项目中；在html文件中引入。

<head>

<meta charset=*"utf-8"*>

<meta http-equiv=*"X-UA-Compatible"* content=*"IE=edge"*>

<meta name=*"viewport"* content=*"width=device-width, initial-scale=1"*>

<title>用户管理信息系统</title>

<link href=*"https://cdn.jsdelivr.net/npm/bootstrap@3.3.7/dist/css/bootstrap.min.css"* rel=*"stylesheet"*>

<script src=*"https://cdn.jsdelivr.net/npm/jquery@1.12.4/dist/jquery.min.js"*></script>

<script src=*"https://cdn.jsdelivr.net/npm/bootstrap@3.3.7/dist/js/bootstrap.min.js"*></script>

<script type=*"text/javascript"*></script>

</head>

## 三．JavaScript（js）

1. js是基于对象和事件驱动的脚本语言，主要应用于客户端。客户端主要是值浏览器。

2. js的特点：

1）交互性：信息的动态交互

2）安全性：js不能访问本地磁盘的文件

3）跨平台性

3. JacvaScript和Java的区别：

js基于对象，java面向对象，二者没有任何联系；js只需解析就能执行，java需要先编译再执行

4. js的组成：三部分

1）ECMAScript：js的基本语法

2）BOM：browser object model，浏览器对象模型

3）DOM：document object model，文档对象模型

5. js和html的结合方式（两种）

第一种，使用标签<script>，具体为

<script type="text/javascript"> js代码 </script>

第二种，使用标签<script>，引入js文件

<script type="text/javascript" src="1.js">

6. js语法

|  |
| --- |
| alter(“这里是内容”) 在网页上弹出一个消息框 |
| 定义变量，都要使用var  var str = “abc” ; var num = 10 |
| typeof(xxx) 查看类型 |
| if、while语句，与java中的相同 |
| for(var i=1;i<10;i++){} 注意var，其他的与java相同 |
| js不区分整数和小数 |
| js中，如果一个一个字符串是”123”,那么+表示字符串的拼接，-表示数字的相减，类似于先把“123”强制转换成了数字 |
| ==进行比较时，比较的是值，比如a=”5”,a==5返回true;===比较的是值和类型，a===5返回false |
| document.write("hahhh") 直接在网页上输出内容，也可以执行html代码，如  document.write("<hr/>") 会输出一条水平线 |
| js定义数组的三种方式：var arr = [1,2,3] ; var arr = new Array(5) //定义一个长度为5的数组，数组的索引从0开始; var arr = new Array(1,2,3)  js数组可以存放不同的类型 |

7. js的函数

|  |
| --- |
| 自定义函数：  function 方法名(参数列表){方法体; 返回值可以省略} 参数列表里不用写var  var add = function(参数列表){方法体; 返回值可以省略} |
| 在一个<scrip></scrip>中定义的一个变量 var a = 10，可以在这个scrip中使用，也可以在其他的scrip中使用，这成为全局变量 |
| js中不存在方法重载，同一方法名后面的方法会覆盖前面的方法 |

8. scrip标签的位置

scrip标签放到任何位置都能执行，一般放到</body>后面，因为代码是从上往下执行，scrip有可能会用到body标签中的内容

9. String对象常用方法：

|  |
| --- |
| var a ="hello";  document.write(a.link("http://www.baidu.com")); // 在页面上显示hello, 单击会链接到百度，（String.link(url)方法） |

10. js 的全局函数

|  |
| --- |
| eval(str); str是js代码，此函数可以直接执行这些js代码 |
| encodeURI("xx"); 对内容进行编码，特别是包括中文，返回string |
| decodeURI(“ ”); 对内容进行解码, 返回string |
| isNaN() 判断是否是一个数字, 返回布尔值，是数字返回false |

11. bom对象

|  |
| --- |
| location: 方法有  location.href 获得当前页面的地址  document.write(location.href) 显示当前页面的网页地址  location.href = url //设置指定页面，可以结合按钮点击事件，例如  <body>  <input type="button" value="跳转" onclick="goto()" />  </body>  <script type="text/javascript">  function goto(){  location.href="http://www.baidu.com";  }  </script>  //onclick跟着按钮单击事件，后跟方法名，方法定义为js方法 |
| window: 窗口对象，顶层对象。所有的BOM对象都是在window里面操作的。 |
| var flag = window.confirm("是否删除"); //出现一个window.confirm()会在网页上弹出一个有“确定”和“取消”按钮的消息框，单击“确定”会返回true |
| var str = window.prompt("请输入内容"); 弹出一个输入框 |
| window.open(url); 打开一个新窗口 |

12. dom对象

想要对标记型文档进行操作，首先需要把标记型文档中的所有内容封装成对象。对标记型文档进行操作，需要对标记型文档进行解析。解析后的对象有: document(整个文档)，element(标签对象)，属性，文本对象, Node对象

13. document对象常用方法

|  |
| --- |
| write() 向页面输出值或执行html代码 |
| getElementById() 通过标签的id值来获取标签，返回值是一个object, 可以获取该标签的一些属性值,如object.name. 可以理解为id值唯一 |
| getElementsName() 通过标签的name值来获取标签，返回值是一个数组，访问该标签属性值的方法为arr[i].id. 可以理解为name值并不为唯一，多个标签有相同的name值 |
| getElementsTagName() 通过标签名称来获取标签，返回一个数组 |
| document.createElement(标签名称) //创建一个标签 |
| document.createTextNode(文本内容) //创建文本 |
| appendChild() 属于标签的一个方法，把一个标签或文本添加到该标签下 |
|  |

14. element(标签对象)

要想操作element对象，首先需要获取element对象，如使用document .getElementById()方法来获取

假设有 var obj = document .getElementById(“xxx”), 则obj可以代表获取了一个标签，其中的element方法有：

|  |
| --- |
| obj.getAttribute(属性名称) 获取属性的属性值 |
| obj.setAttribute(属性名称，属性值) 给标签的属性添加属性值 |
| obj.removeAttribute(属性名称) 删除属性及其属性值，但无法删除value |
| obj.getElementsByTagName(标签名称) 获取该标签下的指定子标签，返回一个集合 |
| obj.innerHTML; 返回该标签的文本内容，可以obj.innerHTML=xxxx 为标签添加内容 |

## 四. xml

1. xml: 可扩展标记型语言，也是使用标签就行操作，其标签可以自己定义，xml可以显示数据，但主要功能是存储数据。

2. xml的应用：不同系统之间传输数据；用于表示生活中有关系的数据；作为配置文件

3. xml的语法：

（1）xml的文档声明

\*创建一个文件，后缀名是.xml

\*如果写xml，第一步一定要有一个文档声明，写在第一行第一列

<?xml version="1.0" encoding="utf-8"?>

（2）定义元素（标签）

\*标签的定义有开始必须要有结束，如果标签内没有内容，则可以在标签内结束<xxx/>

\*一个xml必须有且仅有一个根标签，其他标签都在这个标签下面

\*xml中空格和换行都被当成内容来解析

\*xml区分大小写

\*标签的名字不能以数字和下划线开头，不能以xml、XML开头

\*标签中不能包含空格和冒号

\*标签可以是中文

（3）定义属性

\*一个标签上可以有多个不同名称的属性，属性的命名规范与标签相同，属性值可以用双引号或者单引号括起来  
（4）注释

\*xml的注释与html相同，xml的注释不能嵌套

（5）特殊字符

（6）CDATA区

可以解决多个字符都需要转移的操作

（7）PI指令

4. xml的约束技术：dtd和schema约束

5. xml的解析：dom和sax

## 五. 反射

1. 反射的原理：在配置文件中配置了类，可以通过反射获得类中的所有内容。

首先把java文件保存到本地硬盘为.java文件；编译java文件，成.class文件；使用jvm把class文件加载到内存中；class文件在内存中可以用Class类来表示。

当使用反射后，首先需要获取到Class类，进而得到class类中的所有内容，包括属性(Field类来表示)，构造方法(Constructor类来表示)，普通方法(Method类来表示)。

2. 使用反射操作类中的属性：

假设有一个Student类

Class c1 = Student.class; //方式1

Class c2 = new Student().getClass(); //方式2

Class c3 = Class.forName("com.chinacreator.student.Student"); //方式3，需要填入完整的类名，包括包

## 六．注解

web应用

1. 静态页面：

在webapps目录下创建一个项目目录，目录名不能有空格和中文，在该目录下创建一个html文件

2. 动态页面：

在webapps目录下创建一个项目目录，在该项目目录下创建如下内容：

WEB-INF目录，在这个目录下创建web.xml文件

创建动态或静态页面

WEB-INF目录下的文件不能被客户端访问

一. 注解

public @interface 注解名 {

属性方法，如：

String value() default "";

}

1.元注解：用于描述注解的注解

@Target() 描述注解能够作用的位置

@Target({ElementType.TYPE} 表示注解只能作用于类上

ElementType有三个取值： TYPE可以作用于类上，METHOD 可以作用于方法上 FIELD 可以作用于成员变量上

@Retention() 描述注解被保留的阶段

RetentionPolicy.RUNTIME 表示该注解可以被jvm读取到

@Documented 描述注释是否被抽取到API文档中

@inherited 描述注解是否被子类继承

<servlet>

<servlet-name>demo</servlet-name>

<servlet-class>com.servlet.ServletDemo</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>demo</servlet-name>

<url-pattern>/demo</url-pattern>

</servlet-mapping>

## 七．HTTP

1. GET 与POST请求的区别：

GET: 请求参数在请求行中，在url后；url的长度有限制

POST: 请求参数在请求体中，url的长度无限制

2. Request 和Response的对象的原理：

\* request和response对象是服务器创建的，程序员来使用；

\*request对象获取请求消息，response对象来设置相应消息

3. request 功能：

1）获取请求行数据

方法：获取请求方式 String getMethod()

获取虚拟目录 String getContextPath()

获取Servlet路径 String getServletPath()

获取get方式请求参数 String getQueryString()

获取请求URI String getRequestURI()

获取协议和版本号 String getProtocol()

获取客户机的IP地址 String getRemoteAddr()

request.setCharacterEncoding("utf-8"); //解决中文乱码问题

2）获取请求头数据

String getHeader(String name) 通过请求头的名称获取请求头的值

3）获取请求体数据

只有post请求方式才有请求体，需要先获取流对象，再从流对象中获取数据。

BufferedReader getReader()

其他方式：

1）获取请求参数通用方式

String getParameter(String name) 根据参数名称获取参数值

String[] getParameterValues(String name) 根据参数名称获取参数值的数组

Enumeration<String> getParameterNames() 获取所有请求的参数名称

Map<String, String[]> getParameterMap() 获取所有参数的map集合

2）请求转发：一种在服务器内部的资源跳转方式

request.getRequestDispatcher(path).forward(request,response)

path是需要跳转的路径，如"/RequestDemo2"

浏览器的地址栏不会发生变化；只能跳转到当前服务器的内部资源中；只发送一次请求。

3）共享数据

4）获取ServletContext

4. 响应消息：服务器端发送给浏览器端的数据，分为响应行，响应头，响应空行，响应体。

5. 响应行：组成：协议 / 版本，响应状态码 状态码描述

状态码：1xx：服务器接收客户端消息，但没有接收完成。2xx：成功，如200。3xx：重定向，如302。4xx：客户端错误，如404，请求路径错误；405请求方式没有对应的方法。5xx：服务器错误

6. Response对象：设置响应消息。

1）设置响应行 设置状态码：setStatus(int sc)

2）重定向

response.sendRedirect("/response\_study/ResponseDemo2"); //需要加上项目路径。

重定向与转发的区别：重定向会使地址栏发生变化，可以访问其他站点的资源，重定向是两次请求。

7. ServletContext对象：代表整个web应用，可以个程序的容器（服务器）来通信

获取方式：request.getServletContext(); 或者：this.getServletContext();

功能：1）获取MIME类型 String getMimeType(String file)

2）域对象：共享数据 3）获取文件的真实（服务器）路径 String getRealPath(String path)

## 八. JDBC

1. JDBC操作流程

1) 导入jar包

2) 注册驱动 Class.forname(“com.mysql.jdbc.Driver”);

3) 获取数据库连接对象

Connection con = DriverManager.getConnection(url,user,password)

4) 定义 sql语句 sql = “Select ……”

5) 获取执行sql对象的Statement

Statement statement = con.createStatement();

6) 执行sql

ResultSet rs = statement.executeQuery(sql);

7) 处理结果

8) 释放资源

2. 数据库连接池：Druid使用步骤

1. 导入jar包 Druid-1.0.9.jar

2. 定义配置文件：properties后缀名文件

3. 加载配置文件

4. 获取数据库连接池

5. 获取连接

## 九．会话技术

1. 会话技术：一次会话中包含多次请求和响应。浏览器第一次给服务器资源发送请求，会话建立直到有一方断开为止

1）会话的功能：在一次会话的范围内的多次请求间，共享数据

2）方式：客户端会话技术(Cookie)和服务器端会话技术(Session)

2. Cookie

1) 概念：客户端会话技术，将数据保存到客户端

2) 使用方法：创建Cookie对象，绑定数据；new Cookie(String name, String value)

发送Cookie对象；response.addCookie(Cookie,cookie)

获取Cookie，拿到数据 Cookie[] request.getCookies()

3) 实现原理：基于响应头set-cookie和请求头cookie实现

3. Cookie的细节

1)一次可以发送多个Cookie，方式是创建多个Cookie对象，使用多次response.addCookie(Cookie,cookie)方法

2)默认情况下，浏览器关闭后，Cookie数据被销毁。可以持久化存储：setMaxAge(int Seconds) Seconds 为正数时表示存活的时间，负数表示关闭浏览器就删除Cookie，是默认情况；为0表示立即删除Cookie.

3)Tomcat 8 之后可以存储中文

4. Cookie的特点和作用：特点：cookie存储数据在客户端浏览器；浏览器对于单个Cookie的大小有限制，对于单个域名下的Cookie数量也有限制。

作用：用于存储少量的不太敏感的数据；在不登录的情况下，完成服务器对客户端的身份识别。

5. JSP: Java Serve Pages，java服务端页面，可以理解为一个特殊的页面，既可以定义html页面，也可以定义java代码。JSP本质上是一个Servlet。

6. JSP脚本：JSP定义java的方式

方式1：<% java代码 %> 定义的java代码在Service方法中。

方式2：<%! java代码 %> 这里可以定义成员变量和成员方法

方式：<%= java代码 %> 这里的java代码会输出到页面上

7. Session 对象

概念：服务端会话技术，在一次会话的多次请求间共享数据，将数据保存在服务器端的对象中(HttpSession). Session的实现依赖于Cookie.

特定：数据存在服务器端；Session可以存储任意类型，任意大小的数据；

## 十. JSP详解

1. JSP: Java Serve Pages，java服务端页面，可以理解为一个特殊的页面，既可以定义html页面，也可以定义java代码。JSP本质上是一个Servlet。

2. JSP脚本：JSP定义java的方式

方式1：<% java代码 %> 定义的java代码在Service方法中。

方式2：<%! java代码 %> 这里可以定义成员变量和成员方法

方式：<%= java代码 %> 这里的java代码会输出到页面上

1. JSP指令

\*作用：用于配置JSP文件，导入资源文件

\*格式 <%@ 指令名称 属性名1=属性值1 属性名2=属性值...2%>

\*分类：

1. page 用于配置JSP页面，其属性有

|  |  |
| --- | --- |
| contentType | 等同于response.setContentType(), 设置响应体的mime类型以及字符集，设置当前jsp页面的编码 |
| pageEncoding | 设置当前jsp页面的编码 |
| import | 导入java包 |
| errorPage | 发生异常后跳转到指定页面 |
|  |  |

2.include 导入页面的资源文件

3.taglib 导入资源，常用于导入标签库

4. JSP注释

\*html注释 <!-- -->

\*jsp特有注释 <%-- --%> 可以注释所有

5. JSP内置对象：在JSP页面不需要创建，直接使用的对象，一共有9个

6.MVC开发模式：是指把程序分为三个部分，分别为Model(模型，javaBean，完成具体的业务操作，如查询数据库，封装对象)，View(视图，JSP，展示数据)，Controller(控制器，Servlet，获取请求参数，调用模型，把数据交给View进行展示)。

7. EL 表达式：Expression Language, 表达式语言。可以替换和简化JSP中java代码的编写。语法 ${表达式}。JSP默认支持EL表达式， \${表达式}表示忽略这个EL表达式。

\*用法：${域名称.键名} ：获取指定键名的值，当键名唯一时，可以省略域名称，举例：

<%

request.setAttribute("name", "Tom");

session.setAttribute("age", 13);

%>

${requestScope.name}

${sessionScope.age}

\*empty 判断字符串，集合，数组是否为空和长度是否为0

${empty list}

${not empty list}

8. JSTL标签：概念：JavaServer Pages Tag Library, JSP标准标签库，用于简化和替换jsp页面上的java代码

\*使用步骤：

1. 导入jstl相关jar包

2. 引入标签库：taglib指令

<%@ taglib %>

3. 使用标签库

9. 常用的JSTL标签

\*if 相当于java中的if

<%@ taglib prefix = "c" uri="http://java.sun.com/jsp/jstl/core" %>

<c:if test="true">

<h1>为真时则执行这里</h1>

</c:if>

\*choose 相当于java中的switch

<c:choose>

<c:when test="">xxxx</c:when>

<c:when test="">xxxx</c:when>

<c:when test="">xxxx</c:when>

<c:otherwise>xxxxx</c:otherwise>

</c:choose>

\*foreach 相当于java中的for

<c:forEach begin="1" end = "10" var = "i" step = "1">

循环体

</c:forEach>

10. 三层架构：软件设计架构

\*界面层（表示层/web层）：用户看到的界面，用于接收用户参数，封装数据，调用业务逻辑层，转发jsp页面完成展示

\*业务逻辑层（service层）：处理业务逻辑，即业务功能

\*数据访问层（dao层）：操作数据存储文件

## 十一. Filter

1. 概念：web中的过滤器，当浏览器等访问服务器的资源时，过滤器可以将请求拦截下来，完成一些特殊的功能。

2. 过滤器的作用：一般用于完成通过的操作，如登陆验证、统一编码处理、敏感字符过滤。

3. 使用步骤：

1）定义一个类，实现接口Filter

2）复写方法

3）配置拦截路径：两种方式，分别为：

\*web.xml

\*@WebFilter("/\*") 表示访问所有资源前，都会执行该过滤器

4. Filter生命周期：

public void init() 服务器启动后执行，只执行一次，用于加载资源

public void doFilter() 每一次请求拦截资源时都会执行，执行多次

public void destroy() 服务器正常关闭后执行，只执行一次，用于释放资源

5. 过滤器配置详解：

1）拦截路径配置：

/index.jsp 只有访问具体的路径才会拦截

/user/\* 拦截指定目录下的所有资源

\*.jsp 拦截指定后缀名的资源

/\* 拦截所有资源

## 十二. JQuery

1. 概念：一个JavaScript（js）框架，简化js的开发。JQuery对象和js对象方法不通用。

2. 引入方法：

<script src="js/jquery-3.1.1.min.js"></script>

要么使用绝对路径；相对路径的前提是js文件夹与这个html在同一个目录下，一般为WebContent文件夹下

3. JQuery与js的转换：

JQuery--->js：JQuery对象[索引]

js--->JQuery：$(js对象)

4. 选择器：筛选具有相似功能的元素（标签）

事件绑定：<input type="button" value="点击" id = "b1"/>

<script>

$("#b1").click(function(){

alert("abc");

});

</script>

<script>

$(function(){

//这里表示入口函数，dom文档加载完成之后执行这里的代码

});

</script>

5.基本选择器：

(1)

$("#b1").click(function(){

//#b1 表示标注了一个id为b1的标签，#后面跟要查找的id 值

#(“#one”).css(“属性名”,”属性值”) 设置css样式

});

(2) #(“div”).css() #()里面直接跟字符串表示标签名

(3) #(“.class的名字”) “.”后面跟一个class的名字表示按class进行查找，即查找class = “xxxx”的标签

(4) $("A, B,....") 选中多个元素

6.层级选择器

(1)$("A B") 选择A内部的所有B元素，即使B中又有B 元素，也会被选中

(2)$("A >B") 选择A内部的所有B元素，如果B中又有B 元素，不会被选中

7.属性选择器

(1) $("A[属性名]") 含有指定属性名的A元素

(2) $("A[属性名='属性值']")

8. 循环 .each()

$("button").click(function(){

$("li").each(function(){

alert($(this).text())

});

});

## 十三. AJAX

1. 概念：Ajax 即“Asynchronous Javascript And XML”（异步 JavaScript 和 XML），是指一种创建交互式、快速动态网页应用的网页开发技术，无需重新加载整个网页的情况下，能够更新部分网页的技术。

2. 发送异步请求：使用$.ajax()

$.ajax({

url:"xxx", //请求路径

type:"POST", //请求方式

data:{"username":"tom","age":18}, //设置参数

success:function(){

}, //设置响应成功后执行的函数

error:function(){

}, //设置响应失败后执行的函数

dataType:"text" //设置接收到的响应数据格式

})

3. 发送异步请求：使用$.get()或$.post()

$.get(url,[data][fn][type])

url:请求路径

data:请求参数

fn:回调函数

type:接收到的响应数据格式

## 十四. JSON

1. 概念：JSON(JavaScript Object Notation, JS 对象表示法) 是一种轻量级的数据交换格式。

2. 基本语法：

数据在名称/值对中

数据由逗号分隔

花括号保存对象

方括号保存数组

3. JSON对象与java对象的相互转换：

（1）JSON转为java

\* 1. 导入jackson的jar包 （jackson-databind，jackson-core，jackson-annotations）；

2. 创建jackson核心对象ObjectMapper；

ObjectMapper mapper = new ObjectMapper();

3. 调用ObjectMapper的相关方法进行转换。

ClassType ct = mapper.readValue(String json, ClassType)

（2）java对象转为JSON

\*步骤：1. 导入jackson的jar包 （jackson-databind，jackson-core，jackson-annotations）；

2. 创建jackson核心对象ObjectMapper；

ObjectMapper mapper = new ObjectMapper();

3. 调用ObjectMapper的相关方法进行转换。

String json=mapper.writeValueAsString(args);

4. JSON注解

@JsonFormat(pattern = "yyyy-MM-dd")放在定义的变量之上，用于设置转换后的格式

## 十五. redis

1. redis的数据结构：key-value格式，其中，key为字符串，value有5中数据结构，分别为：字符串(String)，哈希类型(hash)，列表(list)，集合(set)，有序集合(sortedset)

2.字符串类型

1. 存储 set key value

2. 获取 get key

3. 删除 del key

3.哈希类型

1. 存储 hset key field value

hset myhash username zhangsan

hset myhash password 123

2. 获取 hget key field

hgetall key 获取hash中的所有

3. 删除 hdel key field

4. 列表类型：可以添加元素到列表的头部或尾部

1. 添加

lpush key value: 将元素加入到左边

rpush key value: 将元素加入到右边

2. 获取

lrange key start end: 范围获取

lrange key 0 -1 获取所有

3. 删除

lpop key : 删除最左边的元素，并将元素返回

rpop key : 删除最右边的元素，并将元素返回

5. 集合(set)，不允许重复

1. 存储 sad key value

2. 获取 smembers key 获取set集合中所有元素

3. srem key value 删除set中的一个value

6. 有序集合 sortedset，不允许重复元素，且元素有序

1. 存储 zadd key score value: 存储数据和分数，数据按score进行排序

2. 获取 zrange key start value

zrange key start value withscore 获取数据和分数

3. 删除 zren key value

7. 通用命令

1. keys \* 查询所有的key

2. type xxx 获取某key的类型

3. del xxx 删除某一个key value

8. redis持久化

方式1：RDB，默认方式，不需要进行配置，默认使用这种机制。在一定的间隔时间中，检测key的变化情况，然后持久化数据。

方式2：AOF，日志记录方式，可以记录每一条命令的操作。

9. Jedis: java操作redis的工具，使用步骤：

1. 导入jar包：jedis-2.9.0，commons-pool2-2.4.2

2. 获取连接

Jedis jedis = new Jedis("localhost",6379);

Jedis jedis = new Jedis(); //空参数某认本机

3. 操作

jedis.set("username", "zhangsan");

4. 关闭资源

jedis.close();

10. jedis连接池：JedisPool. 使用方法：

1. 创建配置对象；

JedisPoolConfig config = new JedisPoolConfig();

config.setMaxTotal(50);

config.setMaxIdle(10);

2. 创建连接对象、调用方法 getSource() 方法获取Jedis对象连接

JedisPool jedispool = new JedisPool(config,"localhost",6379);

Jedis jedis = jedispool.getResource();

3. 使用

4. 关闭

jedis.close();

## 十六. Maven

1. 修改setting.xml文件：

<localRepository>D:\program\apache-maven-3.6.3\maven-repository</localRepository> ：jar包本地下载地址

<mirrors>

<mirror>

<id>nexus-aliyun</id>

<mirrorOf>central</mirrorOf>

<name>Nexus aliyun</name>

<url>http://maven.aliyun.com/nexus/content/groups/public</url>

</mirror>

</mirrors>

2. groupId: 组织名称，可以写公司域名

atifactId: 模块名称

version: 版本号

## 十七. SpringBoot

1. 在pom.xml中添加相关依赖：

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>1.5.9.RELEASE</version>

</parent>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

</dependencies>

2. 在主函数添加注解：@SpringBootApplication

告诉函数此函数为SpringBoot函数

## End