

# 知识小记:

- 1 转发与重定向
- 2
- 3 重定向: 从当前页面定向到另一个页面, 地址改变
- 4 `response.sendRedirect("welcome");`
- 5 转发: 将用户对当前jsp页面或者是servlet的请求转发给另一个jsp、页面或者servlet, 地址不变
- 6 `request.getRequestDispatcher("/welcome").forward(request, response);`

## (1) 登录页面综合编程:

前端:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset="UTF-8">
5     <title>用户登录</title>
6 </head>
7 <body>
8     <!-- 也可以自己写一个label 不过要加上id -->
9     <form action="/login" method="post">
10         <input type="text" name="username" required><br><br>
11         <input type="password" name="password" required><br><br>
12         <button type="submit">登录</button>
13     </form>
14 </body>
15 </html>
```

数据库连接:

```
1 public Connection getConnection() {
2     Connection con = null;
3     try {
4         Class.forName(drive);
5         con=DriverManager.getConnection(url,user,password);
6     }
7     catch(Exception e){
8         e.printStackTrace();
9     }
10    return con;
11 }
12 public boolean checkUser(String username, String password) throws
SQLException {
13     Connection conn = null;
14     PreparedStatement ps = null;
15     ResultSet rs = null;
16     try {
17         conn = getConnection();
```

```

18         String sql = "SELECT * FROM user WHERE username = ? AND password =
        ?";
19         ps = conn.prepareStatement(sql);
20         ps.setString(1, username);
21         ps.setString(2, password);
22         rs = ps.executeQuery();
23         return rs.next();
24     } finally {
25         rs.close();
26         ps.close();
27         conn.close();
28     }
29 }

```

后端接口:

```

1  @PostMapping("/login")
2      public String login(@RequestParam("username") String
    username,@RequestParam("password") String password)
3      {
4          boolean success = userService.checkUser(username, password);
5          if (success) {
6              system.out.println("登录成功");
7              return "success";
8          } else {
9              system.out.println("登录失败");
10             return "failure";
11         }
12     }

```

## (2) 表格练习:

```

1  完成以下代码:
2  宽318个像素, 高167个像素
3  添加标题, 内容为“学生考试成绩单”
4  添加表格第一行, 输出4列, 分别是“姓名”、“语文”、“数学”、“英语”
5  添加第二行内容, 分别是“张三”、“89”、“92”、“87”
6      <table width="318" height="167" border="1" align="center">
7          <caption>学生考试成绩单</caption>
8          <tr>
9              <td align="center" valign="middle">姓名</td>
10             <td align="center" valign="middle">语文</td>
11             <td align="center" valign="middle">数学</td>
12             <td align="center" valign="middle">英语</td>
13         </tr>
14         <tr>
15             <td align="center" valign="middle">张三</td>
16             <td align="center" valign="middle">89</td>
17             <td align="center" valign="middle">92</td>
18             <td align="center" valign="middle">87</td>
19         </tr>
20     </table>

```

### (3) 前端时间js展示:

```
1  请用 HTML 和 JS 实现:页面上有一个按钮,点击后在按钮下方显示当前时间,格
2  式为“当前时间是 HH:MM”。时间应根据用户电脑时间动态变化。
3  <!DOCTYPE html>
4  <html>
5  <head>
6      <meta charset="UTF-8">
7      <title>显示当前时间</title>
8      <script>
9          function displayTime() {
10              const currentDate = new Date();
11              const hours = currentDate.getHours(); // 小时
12              const minutes = currentDate.getMinutes(); // 分钟
13              const currentTime = "当前时间是 " + formatTime(hours) + ":" +
formatTime(minutes);
14              document.getElementById('current-time').innerHTML = currentTime;
15              setInterval(displayTime, 1000); // 每隔一秒更新时间
16          }
17          function formatTime(time) {
18              return (time < 10 ? "0" : "") + time;
19          }
20      </script>
21 </head>
22 <body>
23     <button onclick="displayTime()">点击显示当前时间</button>
24     <p id="current-time"></p>
25     <p style="font-family: 黑体; font-size: 18px; color: red;">湖南科技大学</p>
26 </body>
27 </html>
```

### (4) 鼠标事件及DOM操作:

```
1  请编程实现:红色背景块,鼠标进入显示“welcome”;离开显示“see you later”。
2  红色背景块的 css 指定为: background-color:red;width:100px; height:50px;
padding:20px。
3  <!DOCTYPE html>
4  <html>
5  <head>
6      <style>
7          .box {
8              background-color: red;
9              width: 100px;
10             height: 50px;
11             padding: 20px;
12         }
13     </style>
14 </head>
15 <body>
16     <div class="box" onmouseover="this.innerText='welcome'"
onmouseout="this.innerText='see you later'"></div>
17 </body>
18 </html>
```

## (5) CSS字体:

```
1 请编程实现：将链接“湖南科技大学”的字体设置为宋体、15 像素、红色字体
2 显示。
3 <!DOCTYPE html>
4
5 <html>
6
7 <head>
8
9     <meta charset="utf-8">
10     <style>
11         a{
12
13             font-family:宋体;
14
15             font-size:15px;
16
17             color:red;
18
19         }
20     </style>
21 </head>
22
23 <body>
24
25 <p>湖南</p>
26
27 <p>湘潭</p>
28
29 <a href="https://www.hnust.com">湖南科技大学</a>
30
31 </body>
32
33 </html>
```

## (6) 数据库连接:

```
1 Connection conn = DriverManager.getConnection(
    "jdbc:mysql://127.0.0.1:3306/test", "root", "123456");
```

## (7) html特性:

HTML (Hyper Text Mark-up Language) 即超文本标记语言或超文本链接标示语言，是目前网络上应用最为广泛的语言，也是构成网页文档的主要语言。HTML文本是由HTML命令组成的描述性文本，HTML命令可以说明文字、图形、动画、声音、表格、链接等。HTML的结构包括头部 (Head)、主体 (Body) 两大部分，其中头部描述浏览器所需的信息，而主体则包含所要说明的具体内容。

1、简易性，HTML版本升级采用超集方式，从而更加灵活方便。

2、可扩展性，HTML语言的广泛应用带来了加强功能，增加标识符等要求，HTML采取子类元素的方式，为系统扩展带来保证。

3、平台无关性。虽然PC机大行其道，但使用MAC等其他机器的大有人在，HTML可以使用在广泛的平台上。

## (8) 什么是AOP:

面向切面编程。是OOP的延续，也是Spring框架中的一个重要内容，是函数式编程的一种衍生泛型。简单的说他就是把我们程序重复的代码抽取出来，在需要执行的时候使用动态代理技术在不修改源码的基础上，对我们的已有方法进行增强。可以减少重复代码，提高开发效率，方便维护。

## (9) 什么是springmvc:

1、一种轻量级的、基于MVC的Web层应用框架。通过把模型-视图-控制器分离，将web层进行职责解耦，把复杂的web应用分成逻辑清晰的几部分，简化开发，减少出错，方便组内开发人员之间的配合。

2、Spring框架中的一个重要内容，同时是一个容器，使用IOC核心技术，管理界面层中的控制对象。同时MVC是模型，视图，控制器的简写。

3、springmvc底层就是servlet，以servlet为核心，接收请求，处理请求，显示结果给用户。核心servlet为DispatcherServlet。

## (10) SSM总结:

1、SSM框架是Spring + Spring MVC + MyBatis，是标准的MVC设计模式，将整个系统划分为view层，Controller层，Service层，DAO层四层，使用Spring MVC负责请求的转发和视图管理，spring实现业务对象管理，mybatis作为数据对象的持久化引擎。

2、Spring是一个开源框架，Spring是一个轻量级的控制反转（IoC）和面向切面（AOP）的容器框架，贯彻了整个项目的始终，Spring的核心思想是IoC（控制反转），即不再需要程序员去显式地new一个对象，而是让Spring框架帮你来完成这一切。

3、Spring MVC框架通过实现Model-View-Controller模式来很好地将数据、业务与展现进行分离，主要体现在view层和Controller层。

4、Mybatis作为DAO层的实现者，完成数据库的增删改查功能，是一个基于Java的持久层框架。

## 其他简答:

### (1) cookie练习。

```
1  创建一个名为“mrCookInfo”的cookie，在cookie中写入用户名称、用户生日和用户电子邮箱地址：
2  String name = "username";
3
4  String birthday = "19900101";
5
6  String mail = "tom@126.com";
7
8  Cookie myCook = new Cookie("mrCookInfo", name + "#" + birthday + "#" + mail);
9
10 myCook.setMaxAge(60*60*24*365); //设置cookie有效期
11
12 response.addCookie(myCook);
```

## (2) 什么是cookie。

cookie是小段的文本信息，通过使用cookie可以标识用户身份、记录用户名及密码、跟踪重复用户。cookie在服务器端生成并发送给浏览器，浏览器将cookie的key/value保存到某个指定的目录中，服务器的名称与值可以由服务器端定义。

## (3) Session的基本原理是什么。

Session对象的原理在于，服务器可以为客户端创建并维护一个所谓的Session对象，用于存放数据。在创建Session对象的同时，服务器将会为该Session对象产生一个唯一编号，这个编号称之为SessionID，服务器以Cookie的方式将SessionID存放在客户端。当浏览器再次访问该服务器时，会将SessionID作为Cookie信息带到服务器，服务器可以通过该SessionID检索到以前的Session对象，并对其访问。需要注意的是，此时的Cookie中仅仅保存了一个SessionID，而相对较多的会话数据保存在服务器端对应的Session对象中，由服务器来统一维护，这样一定程度保证了会话数据安全性，但增加了服务器端的内存开销。存放在客户端的用于保存SessionID的Cookie会在浏览器关闭时清除。我们把用户打开一个浏览器访问某个应用开始，到关闭浏览器为止交互过程称为一个“会话”。在一个“会话”过程中，可能会向同一个应用发出了多次请求，这些请求将共享一个Session对象，因为这些请求携带了相同的SessionID信息。Session对象的正常使用要依赖于Cookie。如果考虑到客户端浏览器可能出于安全的考虑禁用了Cookie，应该使用URL重写的方式使Session在客户端禁用Cookie的情况下继续生效。

## (4) Filter接口有哪些方法？这些方法的用途是什么。

`public void init(FilterConfig filterConfig)`过滤器的初始化方法，容器调用此方法完成过滤的初始化。对于每一个Filter实例，此方法只被调用一次。

`public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)` 此方法与Servlet的service()方法相类似，当请求及响应交给过滤器时，过滤器调用此方法进行过滤处理。

`public void destroy()`在过滤器生命周期结束时调用此方法，用于释放过滤器所占用的资源。

## (5) 说明反转控制（IOC）和面向方向编程（AOP）在spring中的应用。

Spring 核心容器（Core）提供Spring框架的基本功能。核心容器的主要组件是BeanFactory，它是工厂模式的实现。BeanFactory使用控制反转（IoC）模式将应用程序的配置和依赖性规范与实际的应用代码程序分开。Spring的声明式事务基于AOP实现，却并不需要程序开发者成为AOP专家，亦可轻易使用Spring的声明式事务管理。

## (6) Java数据库编程包含哪些类？Java数据库编程的基本过程是什么。

用到的类：Connection、ResultSet、PreparedStatement、Statement

Java中访问数据库的步骤如下：

- 1) 注册驱动；
- 2) 建立连接；
- 3) 创建Statement；
- 4) 执行sql语句；
- 5) 处理结果集（若sql语句为查询语句）；

6) 关闭连接。

## (7) spring配置javabean。

```
1 <bean id="valueBean" class="cn.tedu.spring.ValueBean">
2     <property name="username" value="xxx"/>
3     <property name="realname" value="xxx"/>
4     <property name="password" value="xxx"/>
5     <property name="address" ref="xxx"/>
6 </bean>
7 以下为一个例子：
8 <!-- 定义car这个bean，id为myCar -->
9 <bean id="myCar" class="cn.tewuyiang.pojo.Car">
10     <!--
11         为car的属性注入值，因为speed和price都是基本数据类型，所以使用value为属性设置
12         值；
13         注意，这里的name为speed和price，不是因为属性名就是speed和price，
14         而是set方法分别为setSpeed和setPrice，名称是通过将set删除，然后将第一个字母变
15         小写得出；
16         -->
17         <property name="speed" value="100"/>
18         <property name="price" value="99999.9"/>
19     </bean>
20
21 <!-- 定义user这个bean -->
22 <bean id="user" class="cn.tewuyiang.pojo.User">
23     <property name="name" value="aaa" />
24     <property name="age" value="123" />
25     <!-- car是引用类型，所以这里使用ref为其注入值，注入的就是上面定义的myCar
26         基本数据类型或Java包装类型使用value，
27         而引用类型使用ref，引用另外一个bean的id
28         -->
29     <property name="car" ref="myCar" />
30 </bean>
```

## (8) 简述http请求get和post的区别。

Form中的get和post方法，在数据传输过程中分别对应了HTTP协议中的GET和POST方法。

二者主要区别如下：

- 1) Get是用来从服务器上获得数据，而Post是用来向服务器上传数据；
- 2) Get将表单中数据按照variable=value的形式，添加到action所指向的URL后面，并且两者使用“?”连接，而各个变量之间使用“&”连接；Post是将表单中的数据放在form的数据体中，按照变量和值相对应的方式，传递到action所指向URL；
- 3) Get是不安全的，因为在传输过程，数据被放在请求的URL中；Post的所有操作对用户来说都是不可见的；
- 4) Get传输的数据量小，这主要是因为受URL长度限制；而Post可以传输大量的数据，所以在上传文件只能使用Post；
- 5) Get限制Form表单的数据集必须为ASCII字符，而Post支持整个ISO10646字符集；
- 6) Get是Form的默认方法。

## (9) J2EE是什么？它包括哪些技术。

从整体上讲，J2EE是使用Java技术开发企业级应用的工业标准，它是Java技术不断适应和促进企业级应用过程中的产物。适用于企业级应用的J2EE，提供一个平台独立的、可移植的、多用户的、安全的和基于标准的企业级平台，从而简化企业应用的开发、管理和部署。J2EE是一个标准，而不是一个现成的产品。

主要包括以下这些技术：

### Servlet

Servlet是Java平台上的CGI技术。Servlet在服务器端运行，动态地生成Web页面。与传统的CGI和许多其它类似CGI的技术相比，Java Servlet具有更高的效率并更容易使用。对于Servlet，重复的请求不会导致同一程序的多次转载，它是依靠线程的方式来支持并发访问的。

### JSP

JSP(Java Server Page)是一种实现普通静态HTML和动态页面输出混合编码的技术。从这一点来看，非常类似Microsoft ASP、PHP等技术。借助形式上的内容和外观表现的分离，Web页面制作的任务可以比较方便地划分给页面设计人员和程序员，并方便地通过JSP来合成。在运行时态，JSP将会被首先转换成Servlet，并以Servlet的形态编译运行，因此它的效率和功能与Servlet相比没有差别，一样具有很高的效率。

### EJB

EJB定义了一组可重用的组件：Enterprise Beans。开发人员可以利用这些组件，像搭积木一样建立分布式应用。

### JDBC

JDBC(Java Database Connectivity, Java数据库连接)API是一个标准SQL(Structured Query Language, 结构化查询语言)数据库访问接口，它使数据库开发人员能够用标准Java API编写数据库应用程序。JDBC API主要用来连接数据库和直接调用SQL命令执行各种SQL语句。利用JDBC API可以执行一般的SQL语句、动态SQL语句及带IN和OUT参数的存储过程。Java中的JDBC相当于Microsoft平台中的ODBC(Open Database Connectivity)

## (10) html5新特性。

语义特性：html5赋予网页更好的意义和结构，使用新的语义定义标签，可以更好地了解html文档含义，使得创建网站也更简单。

本地存储特性：能够不需要第三方插件而保存数据到用户的浏览器中。

连接特性：具有更有效的连接工作效率，使得基于网页的实时聊天、游戏和在线交流得到更优化实现。同时，html5拥有更有效的服务器推送技术。

网页多媒体特性：支持网页端的Audio、Video等多媒体功能，使访问视频、音频资源更加简单。

三维、图形及特效特性：基于SVG、Canvas、WebGL及CSS3的3D功能，呈现良好的视觉效果。

## JS常用知识点：

```
1 //日期对象
2 var d = new Date();
3 //默认输出Tue Apr 02 2019 09:01:19 GMT+0800 (中国标准时间)
```



方法	描述
<code>getDate()</code>	以数值返回天 (1-31)
<code>getDay()</code>	以数值获取周名 (0-6)
<code>getFullYear()</code>	获取四位的年 (yyyy)
<code>getHours()</code>	获取小时 (0-23)
<code>getMilliseconds()</code>	获取毫秒 (0-999)
<code>getMinutes()</code>	获取分 (0-59)
<code>getMonth()</code>	获取月 (0-11)
<code>getSeconds()</code>	获取秒 (0-59)
<code>getTime()</code>	获取时间 (从 1970 年 1 月 1 日至今)

事件	描述
<code>onchange</code>	HTML 元素已被改变
<code>onclick</code>	用户点击了 HTML 元素
<code>onmouseover</code>	用户把鼠标移动到 HTML 元素上
<code>onmouseout</code>	用户把鼠标移开 HTML 元素
<code>onkeydown</code>	用户按下键盘按键
<code>onload</code>	浏览器已经完成页面加载

## 查找 HTML 元素

方法	描述
<code>document.getElementById(<i>id</i>)</code>	通过元素 <i>id</i> 来查找元素
<code>document.getElementsByTagName(<i>name</i>)</code>	通过标签名来查找元素
<code>document.getElementsByClassName(<i>name</i>)</code>	通过类名来查找元素

## 改变 HTML 元素

方法	描述
<code>element.innerHTML = new html content</code>	改变元素的 inner HTML
<code>element.attribute = new value</code>	改变 HTML 元素的属性值
<code>element.setAttribute(<i>attribute</i>, <i>value</i>)</code>	改变 HTML 元素的属性值
<code>element.style.property = new style</code>	改变 HTML 元素的样式