1. **使用bootstrap修饰表格**

导入样式：<link rel=”stylesheet” href=”css/bootstrap.min.css”/>

## 组件： （1）Table：

## Table（基础表格占据100%宽度，有水平的分隔线，行高）

## table-bordered（水平，垂直都有线）

## table-striped(斑马线表格)

## table-hover(带有鼠标移动效果的表格)

## table-condensed(紧缩的表格)

## 表格状态类：(tr,td,th)

## .info

## .warning

## .danger

## .success

1. **分页组件（pagination）**

<nav aria-label="Page navigation">

<ul class="pagination">

<li>

<a href="#" aria-label="Previous">

<span aria-hidden="true">&laquo;</span>

</a>

</li>

<li><a href="#">1</a></li>

<li><a href="#">2</a></li>

<li><a href="#">3</a></li>

<li><a href="#">4</a></li>

<li><a href="#">5</a></li>

<li>

<a href="#" aria-label="Next">

<span aria-hidden="true">&raquo;</span>

</a>

</li>

</ul>

</nav>

1. **bootstrap栅格系统**

## 栅格系统会将网页分成行和列，一行最多允许12列

## <div class=”container”>//占据屏幕的90%，居中。

## <div class=”row”>//放行

## <div class=”col-xx-xx”></div>//放列

## </div>

</div>

1. **面包屑导航**

是一种线性的[导航](https://baike.so.com/doc/4678902-4892702.html" \t "_blank)方式。用来表达内容归属关系的界面元素，也就是我们经常看到的"主分类>一级分类>二级分类>三级分类>……>。添加面包屑直接为ul列表添加class=breadcrumb。

## 下拉菜单

## 引入两个脚本：

## <script src=”js/jquery-3.2.1.min.js” type=”text/javascript” charset=”utf-8”></script>

## <script src=”js/bootstrap.min.js” type=”text/javascript” charset=”utf-8”></script>

## 格式：

## <div class="dropdown">

## <button data-toggle="dropdown" >

## Dropdown//菜单名

## <span class="caret"></span>

## </button>

## //下拉菜单

## <ul class="dropdown-menu" aria-labelledby="dropdownMenu1">

## <li><a href="#">Action</a></li>

## <li><a href="#">action</a></li>

## <li><a href="#"> action </a></li>

## <li><a href="#"> action</a></li>

## </ul>

## </div>

## 导航栏

## 向nav标签添加.navbar navbar-default:可以完成一个默认的导航栏。

## 导航栏的元素：

## Logo

## navbar-header：logo

## navbar-brand:商标

## 菜单

## nav navbar-nav

## 退出按钮之类的

## 轮播图

## 轮播（Carousel）插件是一种灵活的响应式的向站点添加滑块的方式。除此之外，内容也是足够灵活的，可以是图像、内嵌框架、视频或者其他想要放置的任何类型的内容。

<div id="myCarousel" class="carousel slide">

<!-- 轮播（Carousel）指标 -->

<ol class="carousel-indicators">

<li data-target="#myCarousel" data-slide-to="0" class="active"></li>

<li data-target="#myCarousel" data-slide-to="1"></li>

<li data-target="#myCarousel" data-slide-to="2"></li>

</ol>

<!-- 轮播（Carousel）项目 -->

<div class="carousel-inner">

<div class="item active">

<img src="img/banner\_1.jpg" alt="First slide">

</div>

<div class="item">

<img src="img/banner\_2.jpg" alt="Second slide">

</div>

<div class="item">

<img src="img/banner\_3.jpg" alt="Third slide"> </div>

</div>

<!-- 轮播（Carousel）导航 -->

<a class="carousel-control left" href="#myCarousel"

data-slide="prev"> <span \_ngcontent-c3="" aria-hidden="true" class="glyphicon glyphicon-chevron-left"></span></a>

<a class="carousel-control right" href="#myCarousel"

data-slide="next">

<span \_ngcontent-c3="" aria-hidden="true" class="glyphicon glyphicon-chevron-right"></span>

</a>

## </div>

## 缩略图

## 缩略图:代表网页上或计算机中图片经压缩方式处理后的小图 ，其中通常会包含指向完整大小的图片的[超链接](https://baike.so.com/doc/5335020-5570458.html" \t "_blank)。缩略图用于在 Web 浏览器中更加迅速地装入图形或图片较多的网页。因其小巧，加载速度非常快，故用于快速浏览。相当于图片文件预览及目录的作用。

## <div class="row">

## <div class="col-xs-6 col-md-3">

## <a href="#" class="thumbnail">

## <img src="..." alt="...">

## </a>

## </div>

## ...

## </div>

## 列表

## <ul class="list-group">

## <li class="list-group-item">…</li>

## <li class="list-group-item">…</li>

## <li class="list-group-item">…</li>

## <li class="list-group-item">…</li>

## <li class="list-group-item">…</li>

## </ul>

## 表单

## 搜集客户数据

## //action：到哪里去

## //method：怎么去post或get方法

## <form action=“ ” method= “post”>

## …

## </form>

## 用bootstrap修饰表单

## <form>

## <div class="form-group">

## <label for="exampleInputEmail1">Email address</label>

## <input type="email" class="form-control" id="exampleInputEmail1" placeholder="Email">

## </div>

## </form>

## js dom操作

## DOM：文档对象模型，document object model。是一种编程接口，通过DOM编程我们就可以通过js来操作网页。

## 网页操作：增删改查。

## HTML网页其实就是一棵DOM树。

## 网页操作

## 1 查找某个元素。

## Document.getElementById(元素的ID)，通过元素ID来获取某个元素。

## Document.getElementByTagName(元素的标签名)，通过元素标签名来获取某个元素。

## Document.getElementByName(元素的名称)，通过元素名称来获取某个元素。

## 2 删除某个元素

## 父元素.removeChild(子元素)

## 3 新增元素

## 创建一个新元素追加到父元素中。

## Ducument.createElement(“元素名”),appendChild(新元素)。

## 4 修改元素

## Ducument.createElement(“元素名”)

## Json

JSON 指的是 JavaScript 对象表示法（JavaScript Object Notation），是轻量级的文本数据交换格式。JSON 独立于语言：JSON 使用 Javascript语法来描述数据对象，但是 JSON 仍然独立于语言和平台。JSON 解析器和 JSON 库支持许多不同的编程语言。 目前非常多的动态（PHP，JSP，.NET）编程语言都支持JSON。JSON 具有自我描述性，更易理解。

## 13 表单验证

## 先用bootstrap栅格系统写好表单，再在JavaScript里编写函数通过正则表达式，来验证输入的内容。

## 定时器

js 定时器有以下两个方法：

setInterval() ：按照指定的周期（以毫秒计）来调用函数或计算表达式。方法会不停地调用函数，直到 clearInterval() 被调用或窗口被关闭。

setTimeout() ：在指定的毫秒数后调用函数或计算表达式。

## 漂浮广告

## 

## 

1. 静态页面开发：

一个main，一个table\_jz，然后分别用设计他们的盒子模型，设计他们的宽高、边距调整。

盒子模型的设置：

.main{

width:1024px;

height:768px;

background: url(../img/);

}

.table\_jz{

width:1024px;

height:768px;

background: url(../img/);

margin:0 auto;

padding-top: 280px;

margin-left:100px ;

}

Margin（外边距）:0 auto //0:用于上下；auto:用于左右

Padding-top(内边距)

1. 游戏核心代码：土地状态的变化
2. 手动控制变化

一个表格通过手动点击变化

<td><imgsrc="img/xxx.png" id="img\_0"onclick="changeState(this.id)"/></td>

变化的脚本：

<script type="text/javascript">

var status=1;

function changeState(id){

document.getElementById(imgId).src='img/seed.png';

}

var timer=20;

function grow(imgId){

timer--;

if(timer==10){

document.getElementById(imgId).src='img/bud.png';

}

if(timer==0){

document.getElementById(imgId).src='img/flower.png';

return;

}

setTimeout("grow("+id+")",1000)

}

function changeStatu(sta){

if(sta=="播种"){

statu=1;

}

if(sta=="发芽"){

statu=1;

}

if(sta=="结果"){

statu=1;

}

if(sta=="采摘"){

statu=1;

}

if(sta=="翻墙"){

statu=1;

}

}

</script>

用一个变量来控制状态变化









Spring:

Web前端通过json/ajax进行数据交换到后端，后端会给前端做出响应。

1. spring框架

核心功能：

IOC：inverse of control(控制反转)

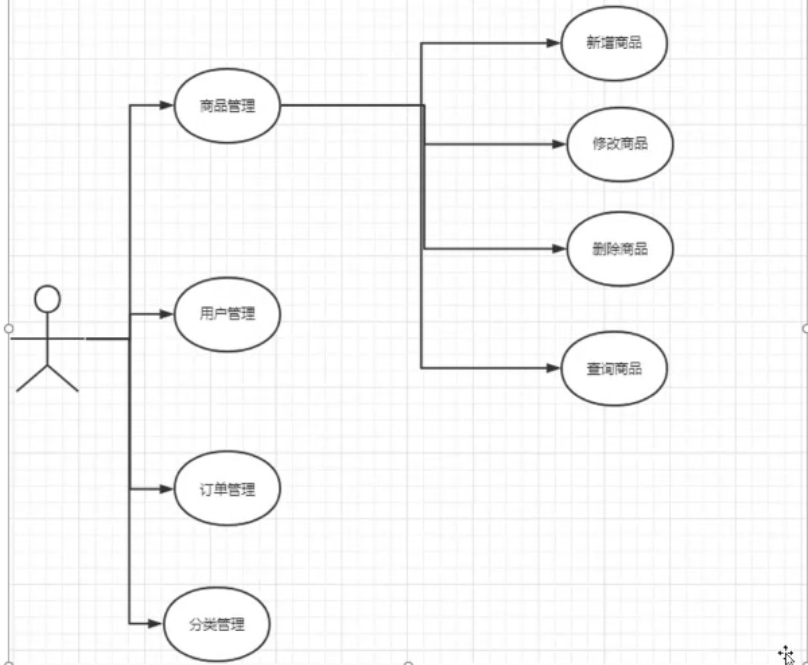
AOP:aspected object program（面向切面编程）

ORM:object relation mapping（对象关系映射）

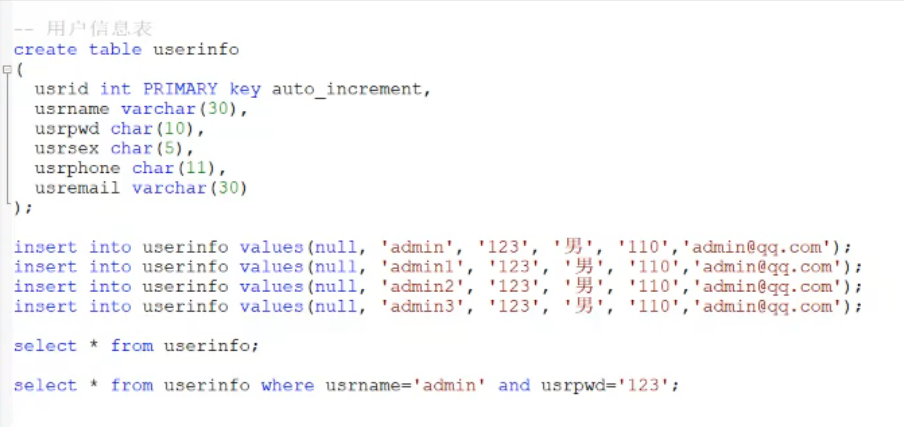
WEB:web

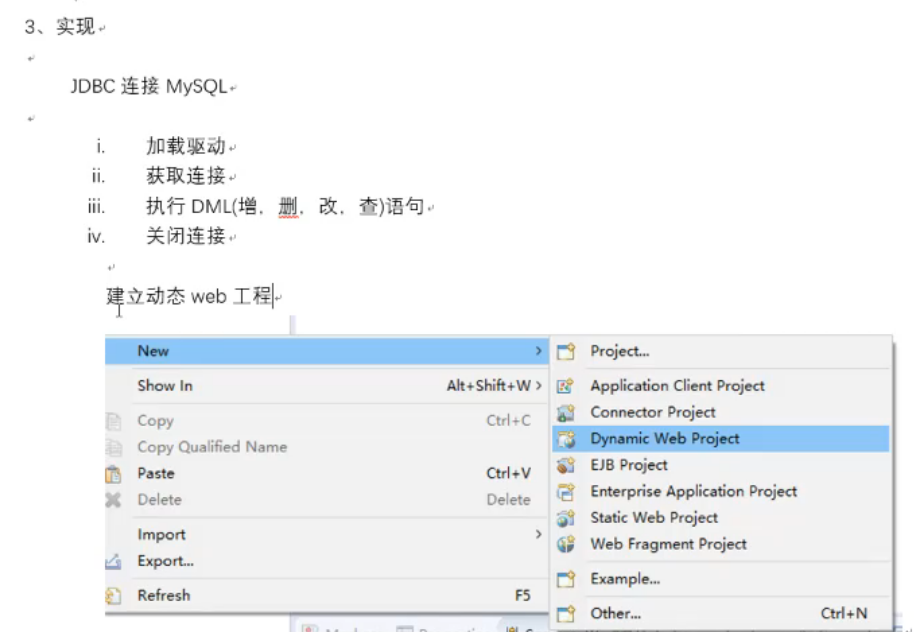


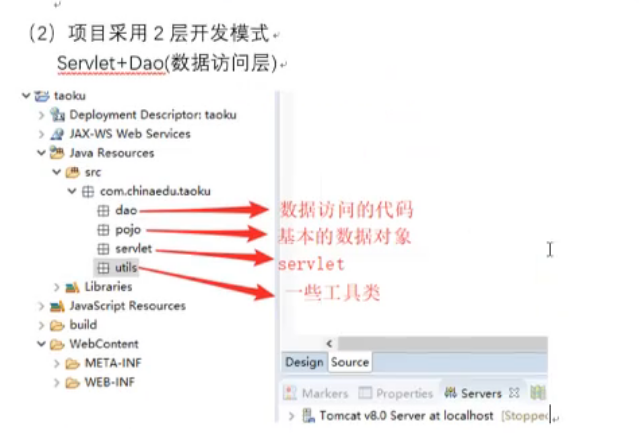
用例图：



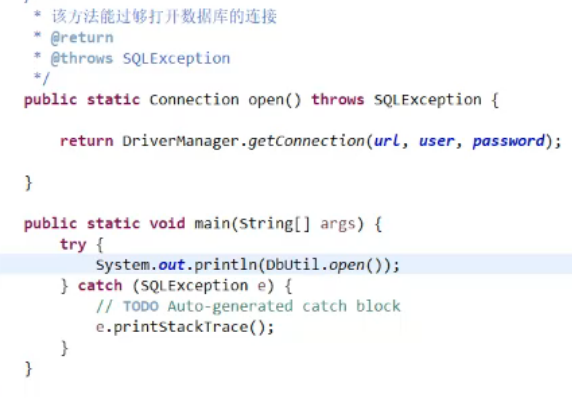
用户表：

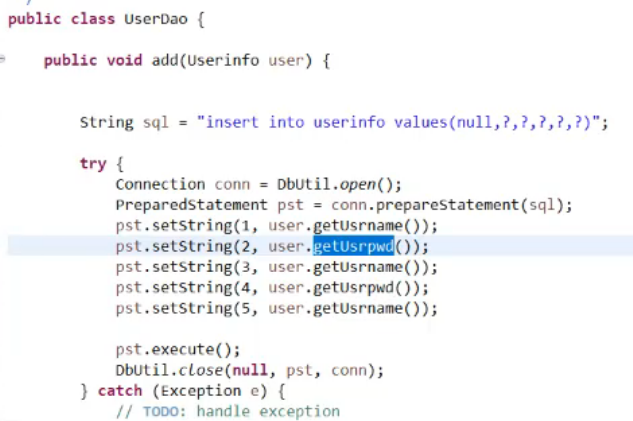














数字格式化异常：就是Integer.parselnt(“abc”)

项目没有显示图片：1. 打开网页 2.右键—点击检查（点击复制图片地址，将其粘贴在任一文本文件中，查看路径）

ProductServelt的作用：查询数据 转到下一个页面（从当前文档跳转到下一个文档，在代码中写一个循环，在其中添加路径会继续跳转页面）

商品详细信息查看流程：

1. 点击图片的时候，将请求发送到ProductDetail并将商品id传过去

获取id：String id = request.getParameter(“id”);

1. 在ProductDetail中获取传入的商品ID同时去数据库中查询商品的详细信息

查询数据库：ProductDao dao = new ProductDao();

Product p = dao.findProductByPk(Integer.parseInt(id));

ProductDao：建立连接：Connection conn = DbUtil.open()

创建命令对象：PreparedStatement pst = conn.prepareStatement(sq1);

pst.setInt(1,id);

执行命令：ResultSet set = pst.executeQuery();

Product p = null;

关闭连接：DbUtil.close(null,pst,conn);

return p;

1. 将查询到的商品信息放入请求作用域中

将数据放入到作用域中，目的是将数据带入到页面中去：request.SetAttribute(“p”,p);

1. 再转到product\_info.jsp这个页面，在该页面显示这个商品的详细信息

跳转页面：request.getRequestDispatcher(“product\_info.jsp”).forward(request,response);

购物车购买流程：

1. 点击购买按钮（用表单提交）

<a href = “ShopServlet?id=${p.pid}”></a>

1. 将请求发送到服务器的ShopServlet（需要传递商品编号）



1. ShopServlet：接收客户传递的商品编号，去数据库中查询商品的详细信息（商品对象）

String id = req.getParameter(“id”);

ProductDao dao = new ProductDao();

Product product = dao.findProductByPk(Integer.parseInt(id));

1. ShopServlet中将商品的对象信息放入到购物车中（购物车其实就是一个商品的集合ArrayList<Product>）

如果是第一次买商品，那么此时应该去创建一个购物车，如果是第二次那么应该直接去用原来的购物车

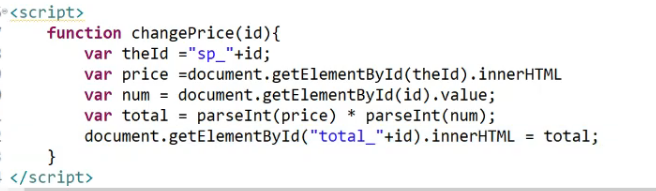


1. 购物车存在的时长是你的整个购物过程中（session适合存放购物车）

查看购物车：

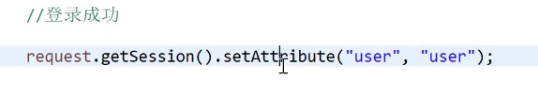


修改小计：



检查用户是否登录：





提交订单流程：

1. 点击提交订单
2. 提交到OrderServelt
3. 在OrderServelt需要处理：购物车中的商品的信息，订单信息（订单的编号，订购人（外键），订购时间）

提交一个订单对象，附带多个商品信息（属于一对多的关系）

1. 在数据库中建立两个表，一个订单表，一个订单明细表

事务：一个原子操作，两个或两个以上的数据库操作，要么都执行，要么都不执行

事务特点：一致性，原子性，持久性，隔离性

jdbc：可以控制事务，通过spring框架也可以控制事务

try{

开始事务

执行订单的插入

执行商品的插入

事务提交

}catch（Exception ex）{

事务回滚

}