# 1复习JavaScript

## 1.1 创建函数类型对象

方式一：标准函数声明方式

function 函数类型对象名（）{

}

方式二：命令行函数声明方式

var 函数类型对象名 = function (){ };

function fun1(){

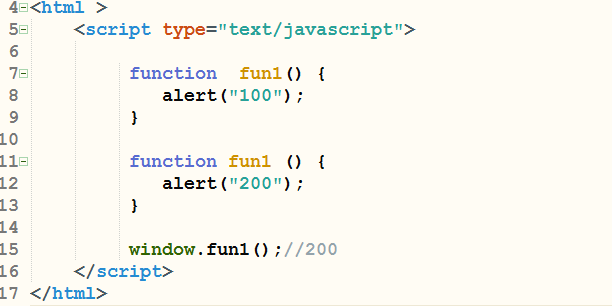
alert("123");

}

## 1.2 在JS中是否存在函数重载

不存在，原因JavaScript5.0是弱类型语言，没有类文件概念。

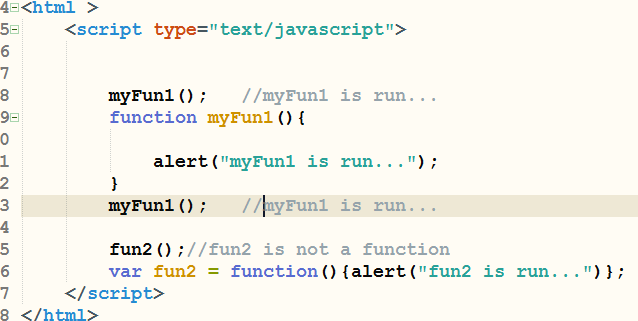
函数名不允许重名，第二个重名函数会将第二个函数覆盖掉。

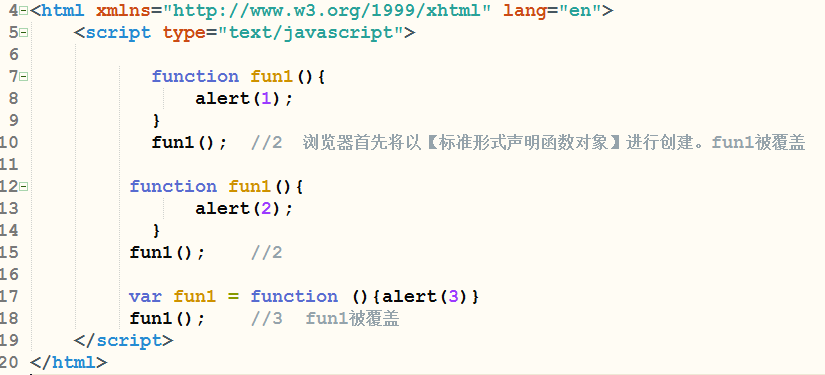


## 1.3 浏览器何时创建函数类型对象

浏览器在加载网页时，首先会将当前网页中所有以【标准形式声明函数对象】进行创建。

然后，才会【自上而下执行命令行】



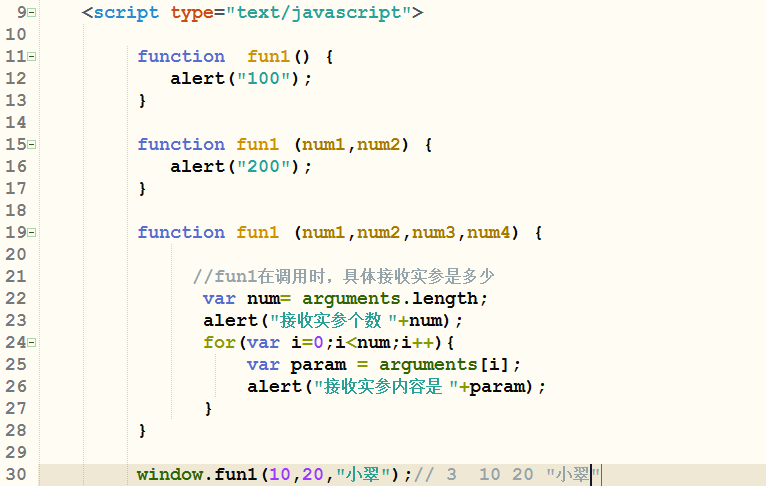


## 1.4 在JS中函数的参数管理方式

1.sun公司将Java中参数管理方式引入到JavaScript

2.JavaScript中自动为每一个函数类型对象匹配一个【arguments】属性

这个属性是一个数组，用于接收来自于外部传递的实参



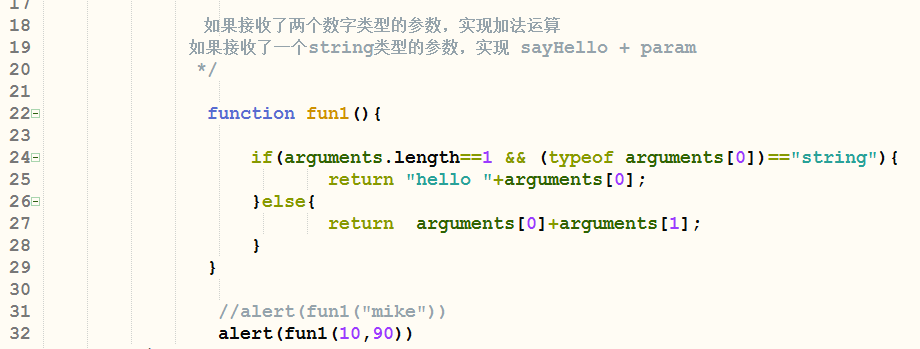
最后一个fun1 将前边的fun1 覆盖掉了。

## 1.5 在JS中实现函数的重载

函数重载本质：

是【封装理念体现】；可以使得函数在接收不同参数时，可以体现不同的功能

使用函数的参数属性arguments 对接收的参数和类型进行判断。



## 1.6 Object类型的创建方式

object类型对象创建方式：

1. var obj = new Object();

2. var stuObj = {sid:10,sname:'mike'};

3. 自定义构造函数

function Student(param1,param2){

this.sid = param1;

this.sname=param2;

}

var stuObj2 = new Student(10,"allen")

1. Var a = [];

## 1.7 Object类型对象的特点

由于JavaScript是弱类型编程语言，因此object类型对象不受类型约束。

所以可以任意为object类型对象【添加和移除】属性和函数



## 1.8 如何为Object类型对象添加属性

obj.属性名 = 值；

obj["属性名"] =值；（重点）

## 1.9 JavaScript中数组是什么类型

object类型

## 1.10 JS中数组的下标是否可以是string

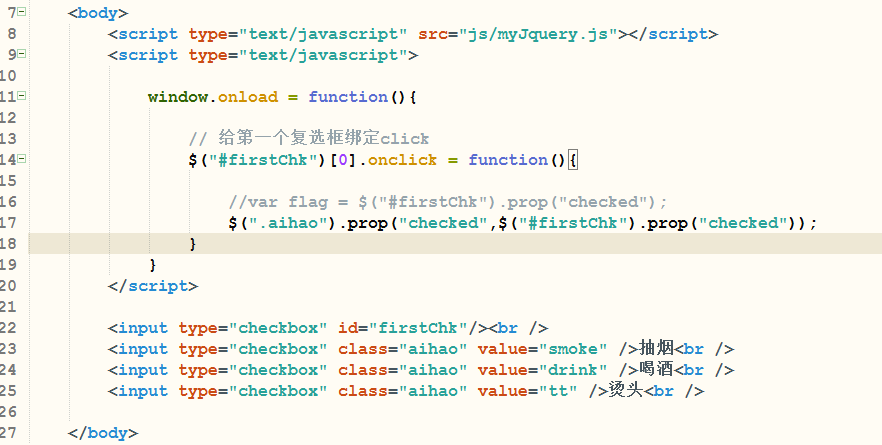
可以



## 1.11 JS缺陷，复选框全选



## 1.12 Jquery原理





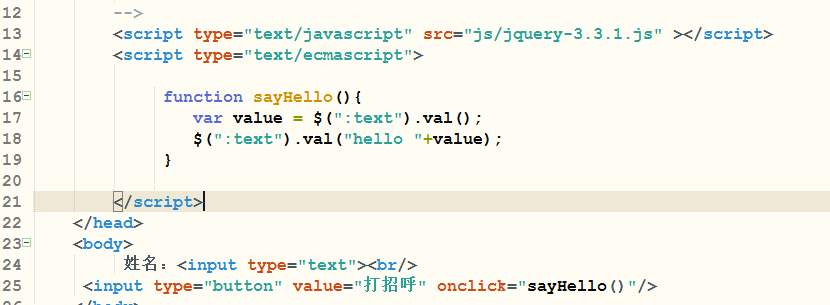
# 2 Jquery

## 2.1 jquery初识

1.JQUERY重点：

1） 掌握jquery工具包定位Dom对象条件

2） 掌握【jquery对象】的功能函数使用



## 2.2 Jquery对象和Dom对象

1）【DOM对象】： 是浏览器创建的对象。

一个Dom对象用于管理网页中一个标签

开发人员是通过对Dom对象下达命令，来间接的控制html标签

02）【jquery对象】：是【jquery函数】负责创建的对象 $()

【jquery对象】就是一个数组。

【jquery对象】保存的是本次定位的所有Dom对象

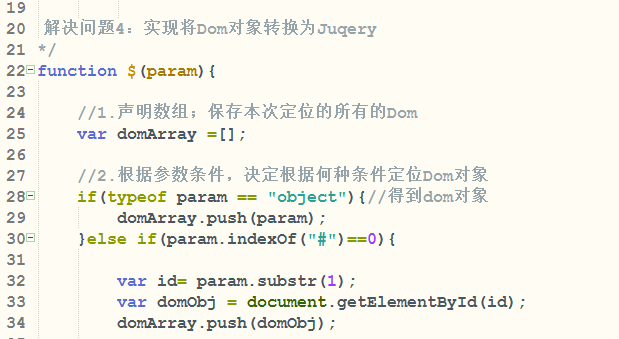
## 2.3 DOM对象转换为Jquery对象

本质：如何将一个Dom对象保存到一个数组中

实现： var domObj = document.getElmentById("one");

var $obj =$(domObj); //obj[0]=domObj





## 2.4 Jquery对象转换为 DOM对象

本质： 从数组提取Dom对象

var domObj = $obj[下标]；

var domObj = $obj.get(下标)；



## 2.5基本选择器

1.定位条件：主要根据【id编号】， getElementById

【class样式选择器】， getElementsByClassName

【标签类型名】 getElementsByTagName

2.具体使用：

$("#id编号")：找到页面中第一个满足条件的Dom对象。并封装到一个数组来返回。

$(".样式选择器"):定位页面中所有拥有指定的【样式选择器】的标签，将标签对应的Dom对象返回。

$("标签类型名"):

$("\*")

$("条件1,条件2")：页面中某一个Dom对象如果满足其中一种条件就会被定位

相当于MySql where job='manager' or sal>=3000



## 2.6 层级选择器

1.定位条件：根据标签之间关系定位

2.标签之间关系： 父子关系，兄弟关系

<tr>

<td>

<input type="text">

</td>

</tr>

[td] 是【tr】的【直接子标签】

【input】是【td】的【直接子标签】

【input】是[tr]的【间接子标签】

<body>

<div>1</div> 大哥

<p>2</p> 二哥

<span>3</span> 三弟

</body>

3.具体使用：

父找子

$("定位父标签条件>定位子标签条件")：定位当前父标签下，所有满足条件的【直接子标签】

$("定位父标签条件 定位子标签条件")：定位当前父标签下，所有满足条件的【直接子标签】和【间接子标签】

找兄弟:

$("定位当前标签条件~定位兄弟标签条件")：定位当前标签后面，所有满足条件的兄弟标签

$("定位当前标签条件+定位兄弟标签条件"): 定位当前标签后面与之紧邻并且满足条件的兄弟标签

$("定位当前标签条件").siblings("定位兄弟标签条件"):定位当前标签,所有满足条件的兄弟标签



## 2.7 表单域选择器

1.HTML中表单域标签分类:

1) input系列：<input type="text"/>

<input type="password"/>

<input type="radio"/>

<input type="checkbox"/>

<input type="file"/>

<input type="button"/>

<input type="submit"/>

<input type="reset"/>

<input type="hiden"/>

2)多行文本框: <textarea></textarea>

1. 下拉列表框：

<select>

<option></option>

</select>

2.定位条件:

根据input标签中type属性内容定位

1. 具体用法：

$(":type属性内容")

$(":text"):定位当前页面中所有的文本框

$(":checkbox"):定位当前页面中所有的多选框



## 2.8 过滤器--表单域工作状态属性过滤器

1.HTML标签中属性分类:

1)基本属性： id ,name,title.

2)样式属性： border,background-color

3)工作状态属性：

disabled: 表示标签是否处于可用状态

readOnly: 表示标签内容是否处于只读状态

checked : 只存在 【radio,checkbox】，表示标签是否处于选中状态

selected: 只存在【option】，表示option标签是否处于选中状态

1. 监听属性：

onclick

onchange

1. 具体用法：

$("选择器:enabled"):定位满足条件的并且处于【可用状态】的所有标签

$("选择器:disabled"):定位满足条件的并且处于【不可用状态】的所有标签

$("选择器:checked"):定位满足条件的并且处于[被选中的状态]的所有标签 raido,checkbox

$("选择器:selected")：定位满足条件的并且处于[被选中的状态]的所有标签 option



## 2.9 Jquery中函数操作html标签属性

1) value 属性： val(值)：对value赋值

val():对value属性读取

2）基本属性【id,name,title】:

attr("基本属性名","值")：对jquery对象中包含的所有Dom对象中指定的

【基本属性】进行统一赋值

attr("基本属性名")： 对jquery对象中第一个Dom对象中指定的

【基本属性】进行取值。

3）样式属性： css("样式属性名","值")

4）工作状态属性：

prop("工作状态属性",值)：对jquery对象中包含的所有Dom对象中指定的

【工作状态属性】进行统一赋值

prop（"工作状态属性"）: 对jquery对象中第一个Dom对象中指定的

【工作状态属性】进行取值。

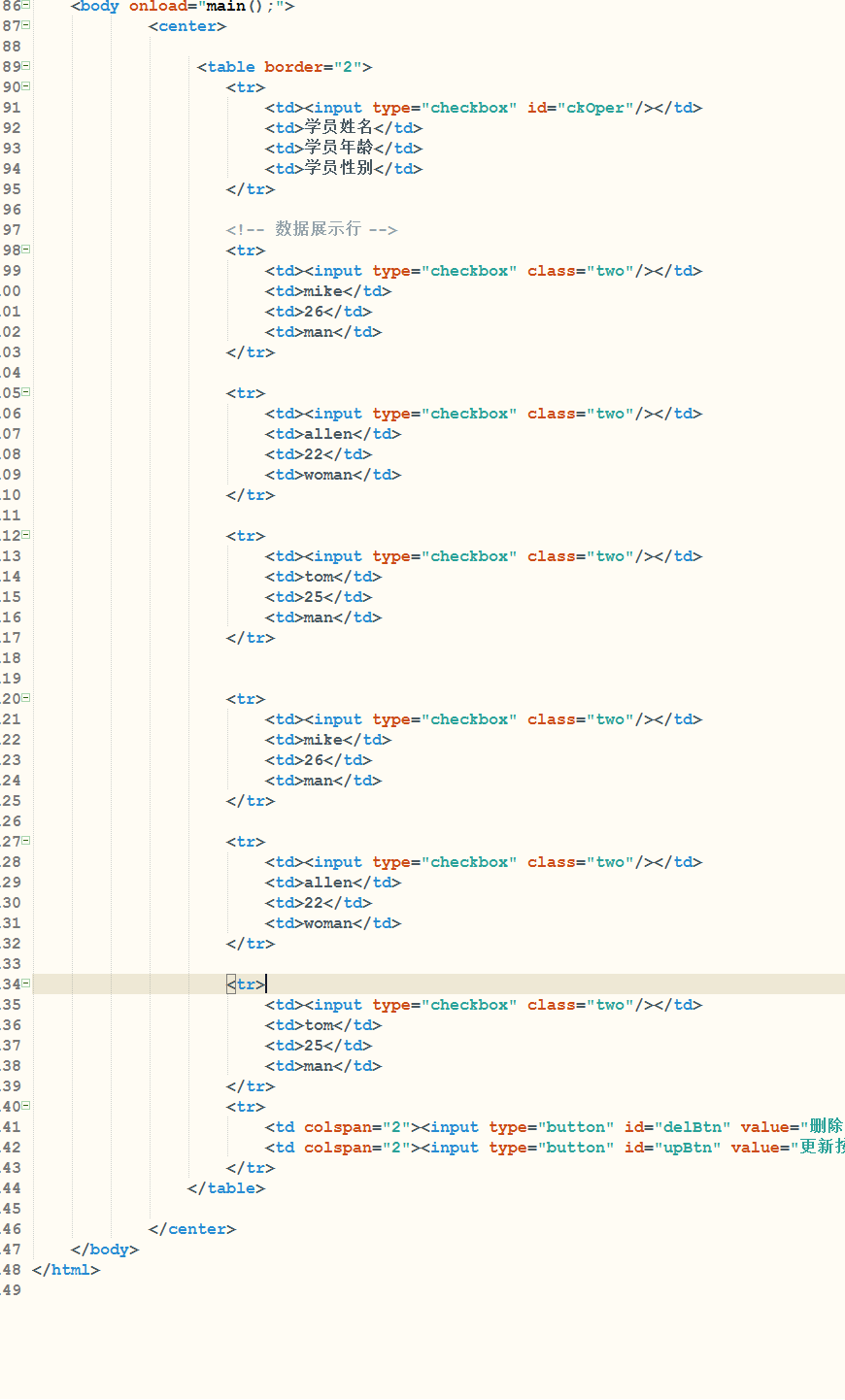
5）文字显示内容

text("文字内容")：对jquery对象中包含的所有Dom对象的文字显示内容进行统一赋值

Text(): 对jquery对象中第一个Dom对象中文字显示内容读取

## 2.10 基本功





## 2.11 过滤器--基本过滤器

1.过滤条件： 定位的Dom对象在jquery对象存在的位置进行过滤器

2.Dom对象在JQUERY对象的位置：Dom对象在jquery对象位置与关联的标签声明位置有关系。

<body>

<p>p1</p>

<div>div1</div> dom\_1

<div>div2</div> dom\_2

</body>

var $obj = $("div"); [dom\_1,dom\_2]

1. 具体使用：

$("选择器:first"):定位页面中第一个满足条件的标签

$("选择器:last"): 定位页面中最后一个满足条件的标签

$("选择器:eq(2)"):定位页面中第三个满足条件的标签

$("选择器:lt(3)"):定位页面中前三个满足条件的标签

$("选择器:gt(0)"):定位页面中除了第一个满足条件的标签之外的标签

$("选择器:gt(4)"):定位页面中除了01234的标签之外的标签

<!--标题行-->

<input type="checkbox">

<!--数据行-->

<input type="checkbox">

<input type="checkbox">

<input type="checkbox">

$(":checkbox:gt(0)") 定位数据行checkbox

## 2.11 过滤器--基本属性过滤器

1）根据标签中，是否【手动声明】了指定的基本属性进行过滤

<div>1</div> 是存在了name属性，但是没有手动声明。name属性 值是默认值 空字符串

<div name="two">2</div> 手动声明了name属性

$("选择器[name]"):定位的标签，首先要满足选择器定位条件，同时 必须手动声明了name

("div[name]"):定位当前页面中，所有手动声明了name属性div标签 <div name="two">2</div

2）根据标签中，基本属性内容是否与【指定内容相同】进行过滤

$(选择器[基本属性名='值'])：定位标签中的指定的【属性内容】与【指 定内容】必须完全相同

<div>1</div>

<div name="two">2</div>

<div>3</div>

$("div[name='']"):<div>1</div> <div>3</div>

3)$(选择器[基本属性名 ^='值']):定位标签中的 指定的【属性内容】以【指定内容】为开头

<div name="one">1</div>

<div name="two">2</div>

<div name="three">3</div>

$("div[name ^='t']"): <div name="two">2</div> <div name="three">3</div>

4)$(选择器[基本属性名 $='值']): 定位标签中的 指定的【属性内容】以【指定内容】为结尾

$("div[name $='e']"): <div name="one">1</div> <div name="three">3</div>

5)$(选择器[基本属性名 \*='值'])：定位标签中的 指定的【属性内容】 必须包含 【指定内容】

相当于 where name like '%a%'

$("div[name\*='e']"): <div name="one">1</div> <div name="three">3</div>

\*\*\*\*\*允许同时使用多个基本属性过滤器进行筛选

$("选择器[id][name $='t'][title='ok']"):要求定位标签必须同时满足这三个条件

## 2.12 Jquery中监听事件绑定

方式1：jquery对象.jquery监听函数（处理函数名）：将指定监听事件绑定到jquery对象中 所有的Dom

HTML 监听属性 jquery监听函数

onclick click()

onchange change()

onblur blur()

\*\*\*\*\*\*\*\*\*\* jquery监听函数名称，就是Html中监听属性去掉on的名字

function fun1(){

alert("处理函数")

}

$(":button").click(fun1):为页面中所有的button标签上绑定onclick监听事件，对应 处理方法fun1

方式2：jquery对象.bind("jquery监听函数名",处理函数名)

解除标签上绑定事件

jquery对象.unbind("jquery监听函数名");

$(":button:first").unbind("click");解除页面中第一个button按钮上onclick监听事件

jquery对象.unbind(); 将当前标签身上所有的监听事件进行解除

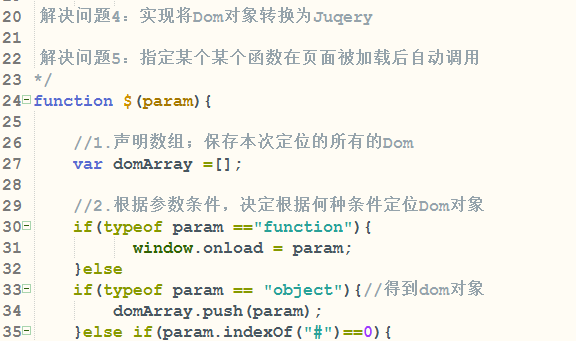
## 2.13 通过jquery指定页面加载完毕后自动调用函数

window.onload = 函数对象；

替换

$(函数对象)

实现原理如下：



## 2.14 $()重载现象总结

$("选择器:过滤器")：定位满足条件的Dom对象，并返回一个jquery对象

$(dom对象) ： 将dom对象添加到一个数组中，并返回这个数组。这个数组就是 jquery对象

$(函数对象)： 这个函数将会在页面被浏览器加载完毕后，被调用



# 3 Jquery实战

## 表格(全选、全不选、变红、删除更新按钮)

见2.0 基本功！

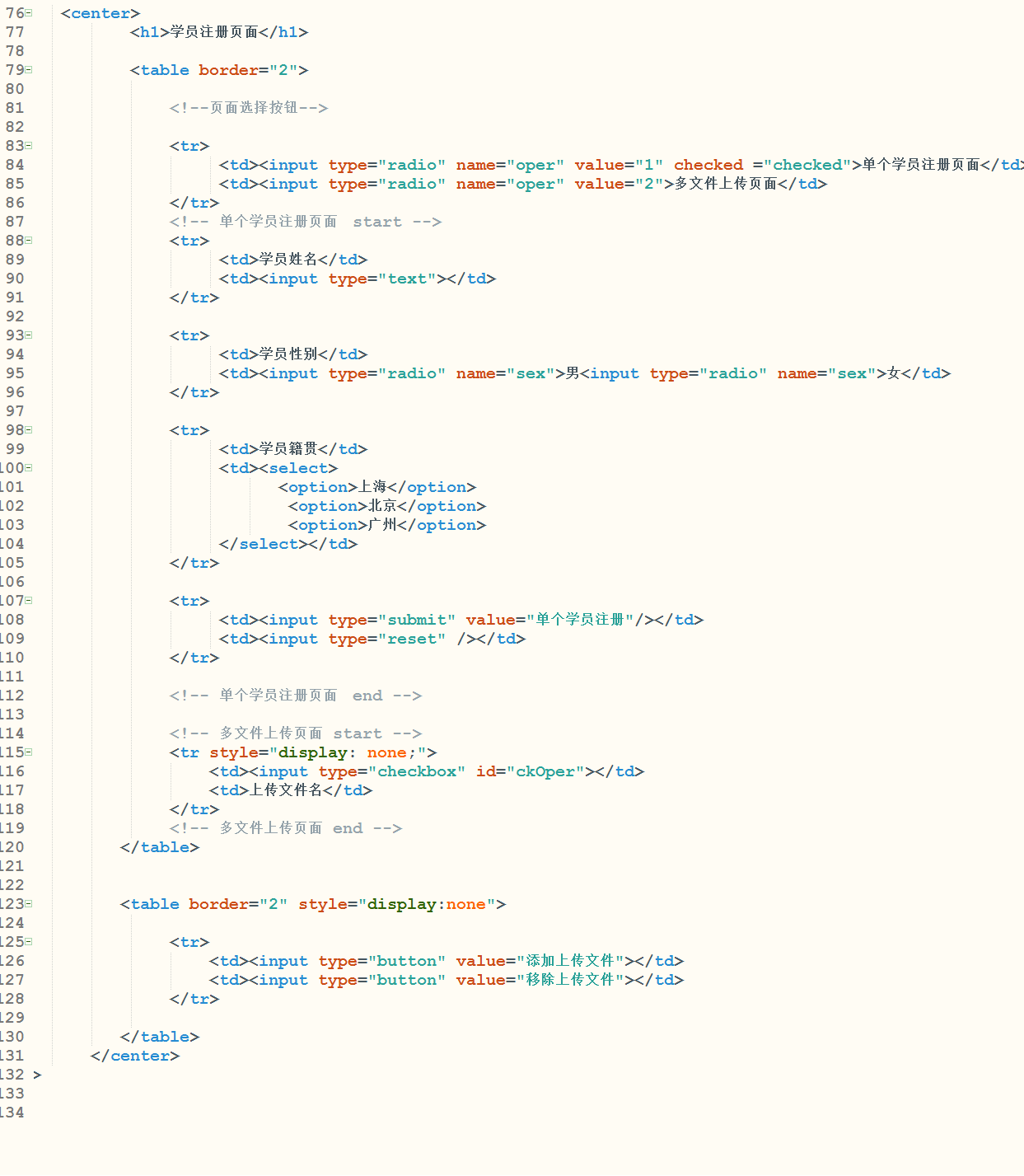


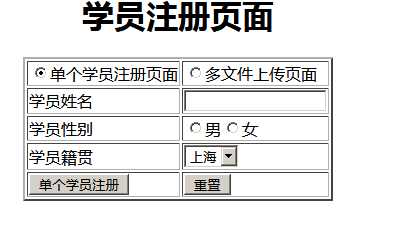
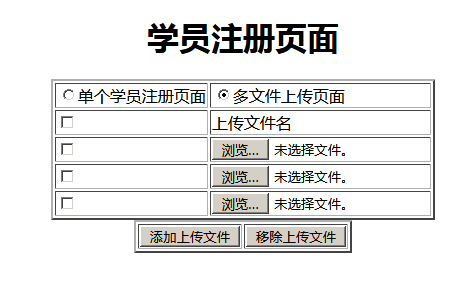
## 下拉列表级联



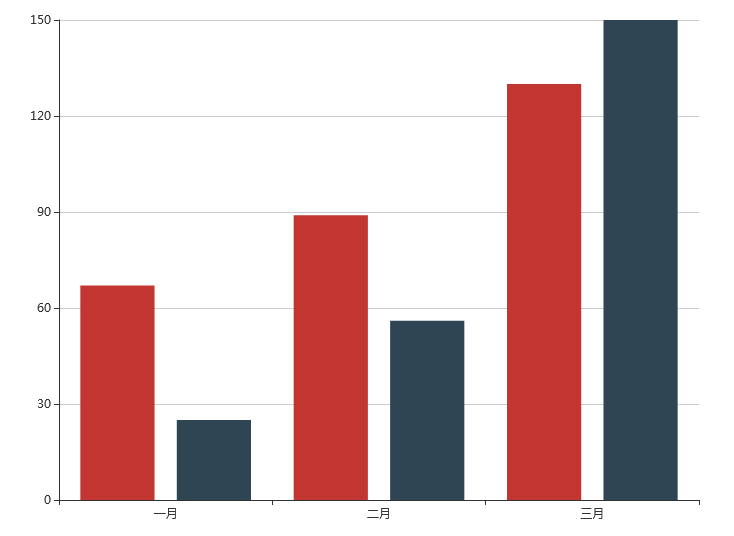
## 动态表格







## echart动态的数据

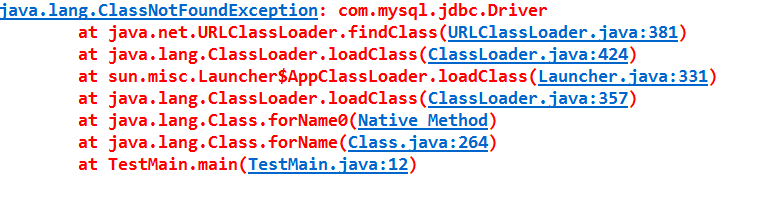


## 下拉列表级联的柱状图



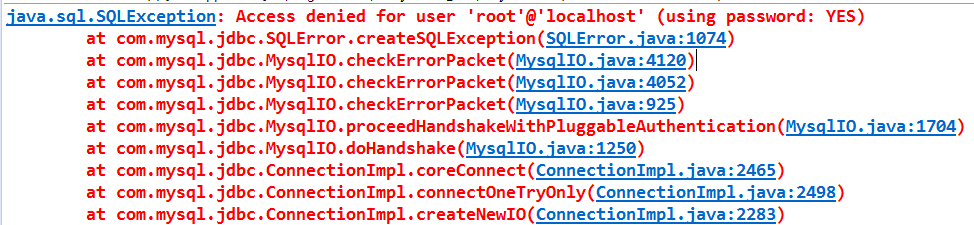
# 4 JDBC中的异常

## 4.1没有引入mysql驱动包



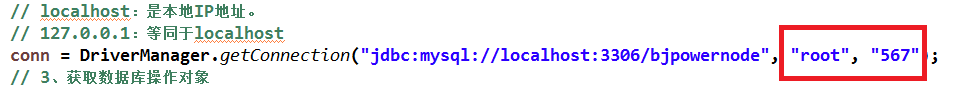
原因：当前工程没有导入 mysql驱动包

## 4.2 密码输入错误

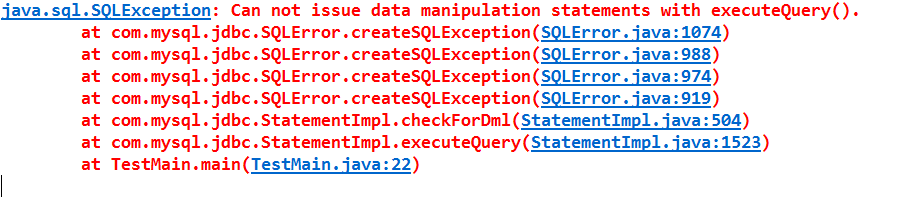


原因：使用登录密码错误

修改地方：



## 4.3 执行器使用不正确

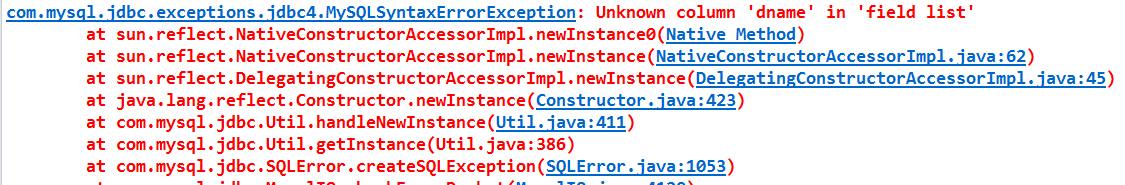


原因： DML(INSERT,UPDATE,DELETE) 不适合使用executeQuery执行器进行处理

修改：

TestMain类中22行，将executeQuery变更为executeUpdate

## 4.4字段在表中不存在异常



异常公式：

Unknown column 字段名 in ‘field list’

未知字段出现在Select 未知字段

未知字段出现在 update 中 set 未知字段=值

Unknown column 字段名 in ‘WHERE clause’

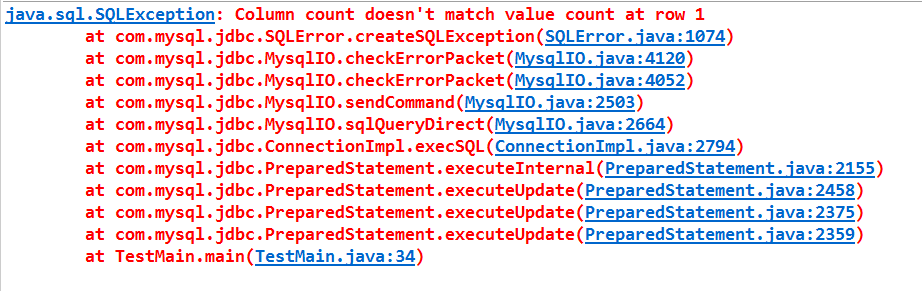
在where过滤条件中，出现未知字段。

修改公式：

当前向数据库推送SQL语句执行器

找到执行器推送的【SQL语句】

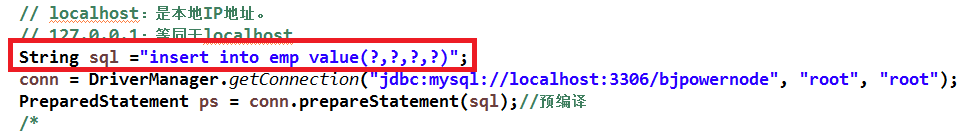
第五章:字段个数与赋值内容不匹配



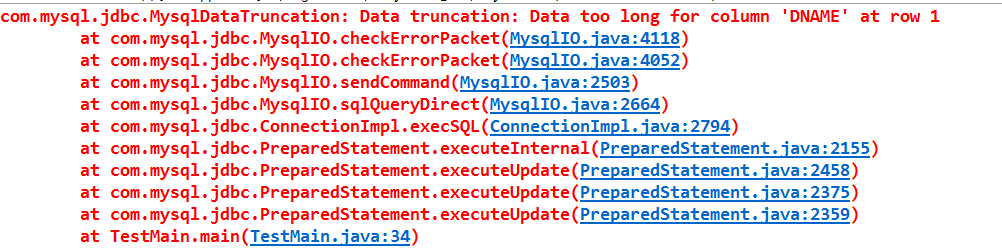
原因：

在预编译过程中，SQL语句中需要赋值的字段个数与实际赋值内容不匹配。

出错地方



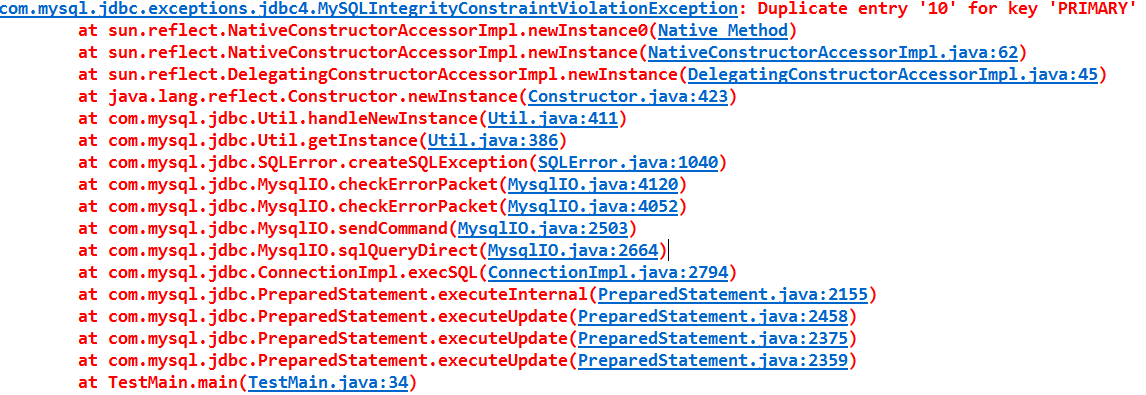
## 4.5 赋值内容超过了字段可以接受最大范围



提供赋值内容，对于字段【dname】来说，太长了。

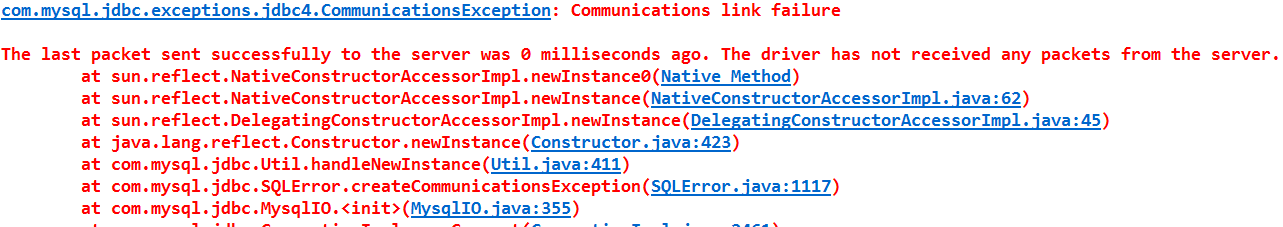
修改地方：到数据库中，将【dname】字段接受字符串长度扩大

## 4.6 插入数据主键值已经存在了



不需要修改程序，只需将插入主键值变为数据库不存在的主键值。

## 4.7 数据库为开通异常



服务窗口，将MySql启动。