**达内郑州嵩山路中心java方向各阶段重点知识点(三)** 2017.8.29

**该文档由达内郑州嵩山路中心java项目经理何老师进行整理,供学员学习使用,其他中心或机构使用请保留该出处,您身边如有想要学习技术的请联系嵩山路java项目经理:何老师,电话:13838585543**

**注:文档中带有 "面" 标识的为重点面试题,未标注的为各阶段重点知识点**

1. **JDBC的工作原理？(面)**

JDBC是java应用程序访问数据库的解决方案

JDBC访问数据库的工作过程:

1)注册加载驱动:

Class.forName("oracle.jdbc.driver.OracleDriver");

2)建立连接:

Connection conn =DriverManager.getConnection(url,username,password);

3)创建执行SQL语句对象及SQL语句:

Statement stmt=conn.createStatement();

String sql="SELECT \* FROM emp";

4)执行SQL语句:

ResultSetrs = stmt.executeQuery(sql);

5)处理结果集

while (rs.next()) {

rs.getInt(...);

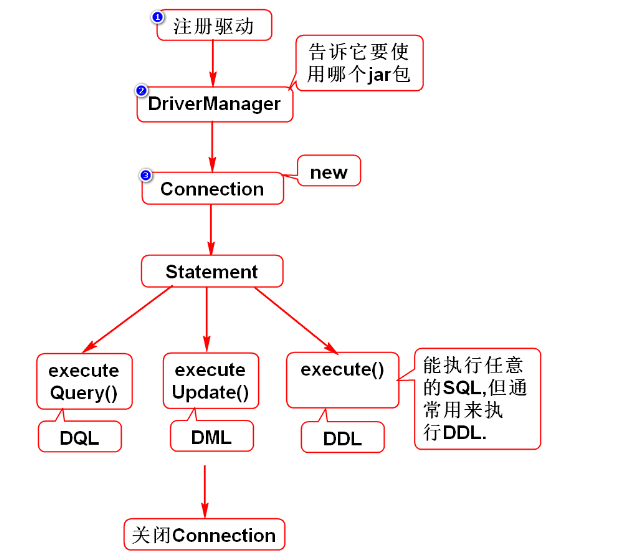
rs.getString(...);

rs.getDate(...);

}

6)关闭连接

conn.close();



1. **JDBC中定义了那些接口和类?**

驱动接口: Driver

驱动注册类: DriverManager

连接接口: Connection

DatabasemetaData

语句对象接口: Statement

PreparedStatement

结果集接口: ResultSet

1. **Statement接口下有哪些执行SQL语句的方法?(面)**

int executeUpdate(String sql): 执行DML语句,返回执行后影响表中的int值记录数.

ResultSet executeQuery(String sql): 执行DQL语句,返回值为查询的结果集.

boolean execute(String sql): 任何SQL都可执行,但由于DML,DQL有专门的方法,所以该方法常用来执行DDL语句.返回值为true时说明指定SQL语句后有查询结果集.

1. **连接池是什么?为什么要使用连接池?(面)**

使用DriverManager每次访问数据库都会重新去加载驱动,建立连接,访问完毕又关闭连接,频繁建立和关闭当并发量大时会导致数据库崩溃.为了避免这种情况可以数据库连接池.

连接池会在初始化一些缓存连接放在一个池子里,每次访问数据库会从去找连接池内空闲的连接进行访问,将该连接的状态有空闲改变为占用,当连接池内连接都在使用时会动态创建一些新的连接放在池子中,访问完毕将连接归还给连接池,将连接的状态由占用改变为空闲,而不是关闭连接.

**怎么使用连接池?**

1)在config.properties文件中设置连接参数和驱动信息

driver=com.mysql.jdbc.Driver

url=jdbc:mysql://localhost:3306/cloud\_note

username=root

password=root

initSize=1

maxSize=3

2)创建Properties对象读取config.properties配置文件中的连接信息

Properties prop=**new** Properties();

prop.load(DBUtil.**class**.getClassLoader().

getResourceAsStream("config.properties"));

driverName=prop.getProperty("driver");

url=prop.getProperty("url");

username=prop.getProperty("username");

password=prop.getProperty("password");

3)创建数据库连接池对象,并将读取到的连接参数设置给数据库连接池

BasicDataSource ds=**new** BasicDataSource();

ds.setDriverClassName(driverName);

ds.setUrl(url);

ds.setUsername(username);

ds.setPassword(password);

4)创建数据库连接池的连接

Connection conn=ds.getConnection();

5)创建SQL语句执行对象

6)执行SQL语句

7)处理结果集

8)归还连接

注:在JDBC中一般会将连接池封装成一个DButil工具类使用.

1. **Statement和PreparedStatement的区别? (面)**

JDBC执行SQL必须编译成执行计划才能运行,创建执行计划很耗效率,执行计划缓存在数据库上,相同的SQL复用一个执行计划.

**Statement**执行两条相同结构的语句,只是参数不同时被认定2个不同的SQL,需要创建两个执行计划,如:

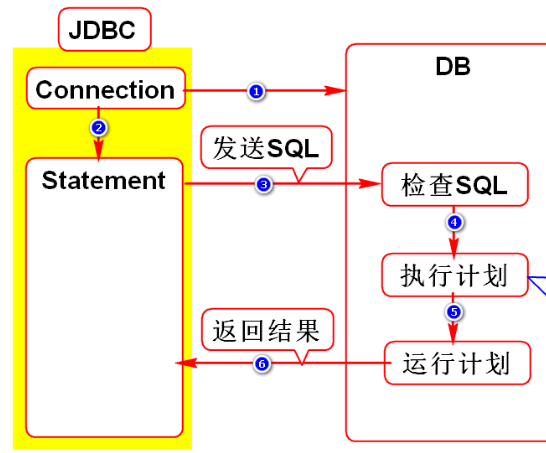
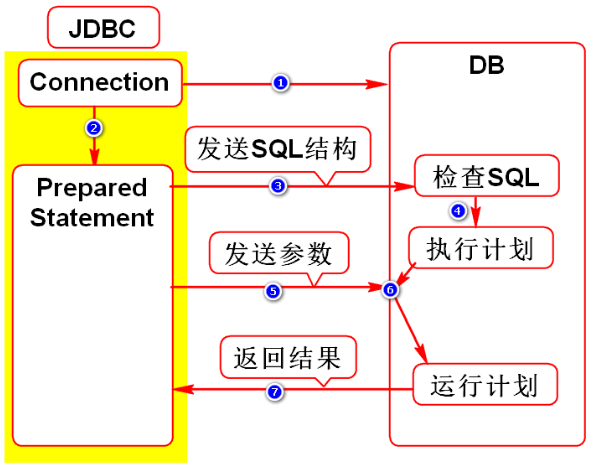
select \* from emps where empno=1

select \* from emps where empno=2

所以Statement执行有条件的语句时效率低,更适合执行不带条件的静态语句(DDL)

**PreparedStatement**执行有条件的语句时,条件参数使用?代替,无论参数怎么改变,对应的SQL都一样,执行计划一样,可以进行复原,更适合执行这种动态参数的SQL.另外还能避免SQL注入攻击,如:

select \* from emps where empno=?

1. **什么事务以及事务的特性? (面)**

事务（Transaction）：数据库中保证交易可靠的机制。

原子性:事务是一个完整的过程,要么都成功,要么都失败

一致性:事务前后的数据要保持一致,即收支平衡

隔离性:事务过程中的数据不能被别人访问

持久性:事务完成后对数据库内数据永久有效

1. **JDBC对事务的支持?**

JDBC默认自动提交事务,在executeUpdate()方法之后自动提交

也可以进行手动管理事务:

conn.getAutoCommit() :获得当前事务的提交方式，默认为true

conn.setAutoCommit(false);取消自动提交事务,true自动提交,false手动提交

conn.commit();提交事务

conn.rollback();回滚事务

一般一项业务中包含一次DML操作就自动管理事务,一项业务中包含多次DML操作需要手动管理事务.

1. **HTML,XML,JavaScript都是什么语言? (面)**

HTML:超文本标记语言

XML:可扩展标记语言

JavaScript:嵌入在HTML中的解释型脚本语言

1. **Web三要素? (面)**

浏览器:向服务器发起请求,下载服务器中的HTML页面,然后将HTML网页中的内容显示出来.

服务器:接收浏览器的请求,发送相应的页面.

HTTP协议:浏览器与服务器的通讯协议.

1. **HTML中块级元素和行内元素的区别?**

块级元素:默认情况下,块级元素会独占一行,两个元素之间会换行,比如<p>,<div>

行内元素:元素之间不会换行,在同一列显示,比如:<span>,<a>等,行内元素可以使用<br>手动换行

1. **HTML中表单有哪些表单控件? (面)(能说出至少10个以上)**

表单是浏览器向服务器传送数据的手段,使用<form></form>标签将表单元素及书写包裹在内.

有三个主要属性:

action:表单提交的URL地址路径

method: 表单的提交方式,默认为get请求方式,可设置为post请求方式.

enctype:表单数据的编码方式

表单控件分为input元素和其他元素:

input元素:

文本框:<input type="text"/>

密码框:<input type="password"/>

主要属性:value:用户输入的文本数据 maxlength:最大输入字符数 readonly:只读

单选框:<input type="radio" checked name="sex" value="男"/>男性

<input type="radio" name="sex" value='女'/>女性

复选框:<input type="checkbox" checked name="like" value="1"/>篮球

<input type="checkbox" name="like" value="2"/>足球

主要属性:value:单/复选框文本 name:分组,一组单/复选框name属性必须相同

checked:设置选中

按钮元素:

提交按钮:<input type="submit" value="提交"/>:自带表单提交效果

重置按钮:<input type="reset" value="重置"/>:重置表单内容

普通按钮:<input type="button" value="点我"/>:普通按钮

主要属性:value:按钮名字

隐藏框,文件选择框元素:

隐藏框:<input type="hidden"/>:一个用户看不到的框体

文件框:<input type="file"/>:选择要上传的文件

其他元素:

文本域:<textarea>文本<textarea/>

主要属性:cols:列数 rows:行数 readonly:只读

下拉选: <select>

<option >请选择</option>

<option selected value="1">北京</option>

<option value="2">上海</option>

<option value="3">广州</option>

</select>

主要属性: selected: 设置默认选中,如果设置多个,位置最下的生效

value: select的value值，主要是用作表单提交的时候将用户选中的option

的value值的传递给服务器

注:input元素中的name属性规定 input 元素的名称, 只有设置了 name 属性的表

单元素才能在提交表单时传递它们的value值.

比如在servlet中可以通过request.getParameter(name 属性的值)来获取浏览器提

交表单时传递给服务器的value值数据.

对于不同的input 元素，value 属性的用法也不同：

type="button", "reset", "submit" -按钮上的显示的文本

type="text", "password", "hidden" -框体内的初始值

type="checkbox", "radio" -与单/复选框相关联的值

注意：<input type="checkbox"> 和 <input type="radio"> 中必须设置 value 属

性,在表单提交时会将chekcd(选中的)单选和复选框的value属性值发送给服务器.

比如在Jquery中可以通过选择器以及val()获取浏览器表单中标签元素的value值

$('#id').val();

1. **在html中使用CSS样式有几种方式? (面)**
2. 内联样式:在元素的style属性中写样式,该样式是无法被其他元素复用的

<h1 style="color:red">CSS</h1>

1. 内部样式:在head中的子元素style中写样式,当前页面的元素可以复用

<style>

h2{color:blue;}

</style>

1. 外部样式:在css文件中写样式,用link标签引入,这些样式可以被所有的网页所复用

<link rel="stylesheet" href="x.css"/>

1. **CSS样式在使用中有哪些特性?**
2. 继承性:父元素的样式可以被子元素继承,如:颜色,字体
3. 层叠性:多次给同一个元素设不同的样式,其效果会叠加
4. 优先级:多次给同一个元素设置相同的样式,效果以后者为准,即后者覆盖前者,也叫作就近原则(离元素近的样式生效 )
5. **CSS中的常用选择器有哪些? (面)**

类选择器:根据class属性选择具有某一类共同特点的元素,使用点加class属性名选择.

*.girl*{color:*pink*;}

id选择器:根据id属性选择唯一的元素,使用#号加id属性名

*#p4*{color:*blue*;}

选择器组:选中每个选择器所对应目标的并集(合),两个选择器之间用逗号隔开.

*.girl***,***#p4*{font-weight:*bold*;}

派生选择器:选择父元素下符合条件的子元素,父元素空格子元素

*#p5* **b**{color:*red*;}

伪类选择器:根据状态选择元素,元素冒号加上状态

**img***:hover*{width:*250px*;height:*250px*;}

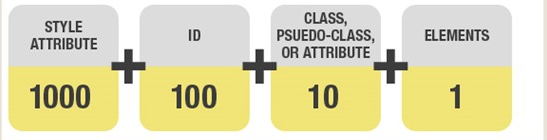
元素选择器:根据标签元素选择元素,比如div

**div**{width:*250px*;height:*250px*;}

1. **CSS样式选择器的优先级?**

当两个或者以上的CSS样式同时操作同一个html元素时,并且属性有重复时该用哪个样式,CSS有一套自己的优先级规范.

想要弄明白CSS样式选择器的优先级需要先搞明白权值:



上图:将不同选择器分为4个等级

第一等：代表内联样式，如: style=””，权值为1000。

第二等：代表ID选择器，如：#content，权值为100。

第三等：代表类，伪类和属性选择器，如.content，权值为10。

第四等：代表元素选择器，如div p，权值为1。

默认情况单独使用时各类型选择器就按如上权值划分优先级,但组合使用时如下图:



有UL, #NAV, LI, .ACTIVE, A等选择器,他们的组合权值为:

1+100+1+10+1=113

当组合使用使用可以通过以上计算方式算出权值,就能知道执行的是哪个样式.

注意:的是通用选择器（\*），子选择器（>）和相邻同胞选择器（+）并不在这四个等级中，所以他们的权值都为0。

1. **在html中引入CSS外部样式和引入外部JS文件分别使用什么方式? (面)**

**引入CSS外部样式:**

<link rel="stylesheet" href="x.css"/>

**引入外部JS文件:**

<script src="a.js"></script>

1. **在JS中有哪些值可以表示空值? (面)**

表示空的值有:null/""/undefined/0/NaN,一些表示空的值都是false

注: JS中任何值都可以做条件,哪怕不是布尔值,当使用其他值做条件时,代表空的值等价于false,代表非空的值等价于true,boolean值做计算时true代表1,false代表0,两个true相加等于2.

1. **JS中的eval()函数和isNaN()函数怎么用? (面)**

eval()函数用于计算字符串表达式或者用于执行字符串中的JS代码:

比如:

var s1="2+5";

eval(s1); //7

或者

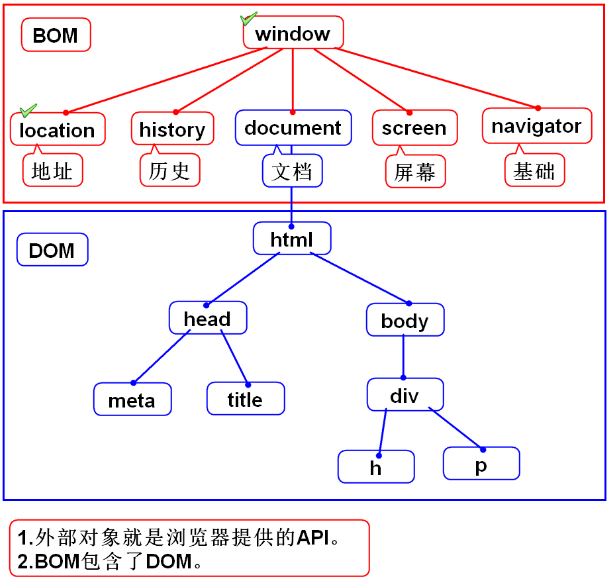
eval("2+6"); //8

isNaN()函数是判断一个值是不是不是一个数字,不是一个数字返回true,是数字返回false.

1. **javascript中的BOM对象和DOM对象? (面)**

BOM对象:浏览器对象模型,用来操作浏览器窗口的,让javascript与浏览器实现对话,包含DOM对象.比如:浏览器历史记录,改变浏览器窗口等操作

DOM对象:文档对象模型,用来操作和访问HTML中的文档标签.比如:增删节点操作



1. **javascript中的弹出框有几种?分别怎么实现? (面)**

常用的有三种:

提示对话框:alert():弹出并显示字符串文本信息

**function** f1(){

alert("你好");

}

确认对话框:confirm():弹出确认框,点击确定返回true,点击取消返回false.

**function** f2(){

**var** b=confirm("你吃了吗");

console.log(b);

}

输入对话框: prompt():弹出输入框,让用户输入信息

**function** f3(){

**var** r=prompt("你吃了吗?");

console.log(r);

}

1. **JavaScript中的定时器有哪2种? (面)**

在javaScript中有两种定时器:

setInterval(exp,time):周期性定时器,exp为执行语句,time为时间周期,单位毫秒

启动周期性定时器,相当于启动了一个线程,他会每隔N秒调用一次函数,直到停止为止.

停止需要调用clearInterval()函数并传入要停止的定时器id

**function** f4(){

**var** n=5;

**var** id=setInterval(**function** (){

console.log(n--);

**if**(n==0){

//停止定时器需要指定ID

clearInterval(id);

console.log("爆炸");

}

}, 1000);

}

setTimeout(exp,time):一次性定时器,exp为执行语句,time为间隔时间,单位毫秒

启动一次性定时器,相当于启动了一个线程他会延迟N毫秒执行函数,并且当执行过一次后就会自动停止, 在一次性定时器未执行前可以调用clearTimeout()并传入要停止的定时器id停止定时器,若已经执行则停止无效

**var** id;

**function** f5(){

id=setTimeout(**function**(){

console.log("起床啦");

},3000);

}

**function** f6(){

//在一次性定时器未执行前可以停止,若已经执行则停止无效

clearTimeout(id);

}

1. **在JavaScript和JQery中怎么实现页面加载完毕立即执行函数? (面)**

在JavaScript中可以使用onload事件:该事件是页面加载事件,会在页面加载完毕后自动触发执行函数,如下:

window.onload=**function**(){

alert("页面加载完毕,已执行该函数");

}

在JQery有更简单的方式,如下:

$(**function**(){

alert("页面加载完毕,已执行该函数");

})

1. **JQery对象和DOM对象的关系,怎么实现互相转换? (面)**

jQuery对象只能调用jQuery方法, DOM对象只能调用DOM方法, 二者可以互相转换来调用对方的方法.

DOM对象转JQery对象:

假设有DOM对象img,在执行函数时传入this,this指的是img自身,是一个DOM对象,将其转换成JQery只需要将其使用$()转换,之后就可以使用JQery对象的width()方法:

<img src=*"../images/01.jpg"*onclick="chg(this);"/>

<sciprt>

**function** chg(img){

$(img).width();

}

</sciprt>

JQery对象转DOM对象:

两种转换方式将一个jQuery对象转换成DOM对象：[index]和.get(index);

1) JQery对象本质上是一个DOM数组,内部的元素其实都是DOM对象，可以通过

[index] 的方法，来得到相应的DOM对象。

**var** $v =$("#v") ; //jQuery对象

**var** v=$v[0]; //DOM对象

alert(p.innerHTML); //使用DOM对象的方法

2)jQuery本身提供了通过.get(index)方法，得到相应的DOM对象

**var** $v =$("#v") ; //jQuery对象

**var** v=$v.get(0); //DOM对象

alert(p.innerHTML); //使用DOM对象的方法

1. **JQery选择器有哪些? (面)(只需要说出有哪些选择器以及他们的作用和语法即可)**

选择器是为了定位节点,对节点进行增删改操作,在JQery中常用选择器有:

--基本选择器:依据id,class,元素,选择器组定位元素.

比如:$('#id'),$('.class'),$('div'),$('div,#id');

--层次选择器:类似CSS中的派生选择器,在父节点下找符合条件的子元素

比如:$("div p"):选中父元素div下的所有的子元素p

--过滤选择器:依据元素的基本特征,文本内容,元素可见性,元素的属性和状态定位元素

基本过滤选择器:依据元素的基本特征定位元素,一般用在表格和列表中.

:first 第一个元素

:last 最后一个元素

:eq(index) 下标等于index的元素

比如:$("tr:first"):选择tr下的第一个元素

内容过滤选择器:根据文本内容定位元素

:contains(text) 匹配包含text的元素

:empty 匹配所有不包含子元素或文本的空的元素

比如:$("p:contains('java')"):匹配元素中包含有java字符串内容的元素

可见性过滤选择器:根据元素的可见性定位元素

:hidden 匹配所有不可见或type为hidden的元素

:visible 匹配所有的可见性元素

比如:$('input:hideen'):配备input标签下所有不可见的元素

属性过滤选择器:根据属性定位元素

[attribute] 匹配所有带有attribut属性的元素

[attribute=value] 匹配属性等于value的元素

[attribute!=value] 匹配属性不等于value的元素

比如:$("input[value='你好']"):在input标签中匹配value属性等于'你好'的元素

状态过滤选择器:根据状态定位元素

:checked 在checkbox(复选框)中匹配checked(选中)的元素

:selected 在select(下拉选)中匹配option被选中的元素

比如:$("input:selected"):在input标签中匹配下拉选中option是选中状态的元素.

--表单选择器:通过form表单控件的类型来匹配元素

:text 匹配文本框

:password 匹配密码框

:button 匹配普通按钮

:summit 匹配提交按钮

:reset 匹配重置按钮

:radio 匹配单选框

:checkbox 匹配复选框

:file 匹配文件框

:hidden 匹配隐藏框

比如:$(":text"):匹配文本框

1. **怎么通过JQery读写节点? (面)**

通过JQery可以读写节点的HTML,文本,属性值等

HTML的内容: 读: obj.html() 写: obj.html("<p>写点什么呢?</p>")

文本内容: 读: obj.text() 写: obj.text("随便写点啥!");

value属性值: 读: obj.val() 写: obj.val("随便是啥?");

节点属性值: 读: obj.attr("属性名"); 写: obj.attr("属性名","属性值");

注:html()和text()区别在于:

html()读写时会带有文本内容的样式,text()读写时只会读写文本内容.

html()和val()的区别在于:

html()一般是对非表单元素的双标签元素进行读写操作

val()则针对表单元素或单标签带有value属性的元素,比如input的不同类型的标签

读写时的区别:读的时候()内不加任何内容,写的时候()需要加入要写入的内容

1. **怎么通过JQery创建,增加,删除节点? (面)**

--创建节点:$("节点的内容") 比如:$("<p>你好</p>")

--增加节点:根据要增加的DOM节点位置不同,使用的方法也不同

parent.append(obj):向父节点的最后一个子节点后追加

parent.prepend(obj):向父节点的第一个字节点前增加

brother.after(obj):向当前兄弟节点之后增加

brother.before(obj):向当前兄弟节点之前增加

--删除节点:obj.remove()

--清空节点:obj.empty()

比如: <tr id=*"id1"*>

<td>姓名</td>

<td>性别</td>

<td>年龄</td>

</tr>

向tr中最后一个子节点后追加节点:

$('#id1').append($("<td>工资</td>"));

删除第一个子节点:

$('#id1').eq(0).remove();

追加节点完毕后结果为:

<tr id=*"id1"*>

<td>性别</td>

<td>年龄</td>

<td>工资</td>

</tr>

1. **怎么通过JQery进行CSS样式操作? (面)**

追加指定样式:addClass(" ")

移除指定样式:removeClass(" ")

移出所有样式:removeClass( )

切 换 样 式:toggleClass(" ")

判断是否有某个样式:hasClass(" ")

1. **怎么通过JQery进行节点遍历操作? (面)**

查找当前节点的直接子节点:children()

查找满足选择器的所有后代:find(selector)

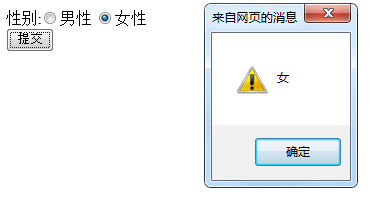
查找所有的兄弟节点:siblings()

下一个兄弟节点:next()

上一个兄弟节点:prev()

查找父节点:parent()

1. **通过JQery获取和修改单选框(radio)中被选中的value? (面)**

****

<script>

$(**function**(){

$('#btn').click(getChecked);

})

**function** getChecked(){

**var** sex=$('#good input:checked').val();

alert(sex);

**var** sex1=$('#good input:checked').val("无法区分性别");

alert(sex1.val());

}

</script>

<body>

<form id=*"good"*>

性别:<input type=*"radio"* name=*"sex"* value=*"男"*>男性

<input type=*"radio"* name=*"sex"* value=*"女"*>女性<br>

<input type=*"button"* value=*"提交"* id=*"btn"*>

</form>

</body>

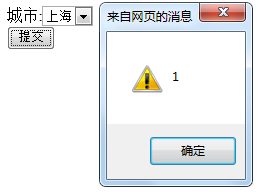
以上案例,可以使用层次选择器+过滤选择器,获取id为good的父元素下的符合状态过滤选择器是checked(选中状态)的元素.第二个红色框选中的是获取选中的radio并修改其value属性的值.

另外还以使用如下方式获取:

var sex=$("input[type='radio']:checked").val();

var sex=$('input:radio[name="sex"]:checked').val();

1. **通过JQery获取和修改下拉选(select)中被选中的option的value和文本? (面)**

<script>

$(**function**(){

$('#btn').click(getSelected);})

**function** getSelected(){

**var** city=$('#city option:selected').val();

alert(city);

city=$('#city option:selected').text();

alert(city);

**var** city1=$('#city option:selected').text("广州");

alert(city1.text());

city1=$('#city option:selected').val("4");

alert(city1.val());}

</script>

<body>

<form id=*"good"*>

城市:<select id=*"city"*>

<option value=*"1"*>上海</option>

<option value=*"2"*>北京</option>

<option value=*"3"*>郑州</option>

</select><br>

<input type=*"button"* value=*"提交"* id=*"btn"*>

</form>

</body>

以上案例:可以使用层次选择器+过滤选择器,获取id为city的父元素下的符合状态过滤选择器是selected(选中状态)的option元素.

第一个红色框选中的是获取选中的option的 value属性的值.

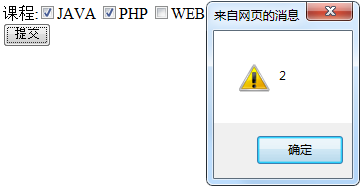
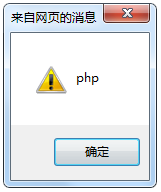
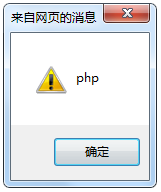
第二个红色框选中的是获取选中的option的文本值(双标签内的文本,如上海)

第三个红色框选中的是获取选中的option的并修改其文本值为广州.

第四个红色框选中的是获取选中的option并修改其value属性的值为4.

弹窗的结果为:1 上海 广州 4

1. **通过JQery获取和修改复选框(checkbox)中选中的option的value和文本? (面)**

<script>

$(**function**(){

$('#btn').click(getChecked);})

**function** getChecked(){

**var** box=[];

$('#good :checkbox:checked').each(**function**(){

box.push($(**this**).val());

})

alert(box.length);

**for**(i=0;i<box.length;i++){

alert(box[i]);

}

alert($('#good :checkbox:checked').eq(1).val());

**var** box1=$('#good :checkbox:checked').

eq(1).val("oracle");

alert(box1.val());

}

</script>

</head>

<body>

<form id=*"good"*>

课程:<input type=*"checkbox"* value=*"java"*>JAVA

<input type=*"checkbox"* value=*"php"*>PHP

<input type=*"checkbox"* value=*"web"*>WEB<br>

<input type=*"button"* value=*"提交"* id=*"btn"*>

</form>

</body>

以上案例: 使用层次选择器+过滤选择器,获取id为good的父元素下的符合表单元素选择器:checkbox的元素再次使用状态过滤选择器:checked定位到被选中的复选框.

注意:复选框通过选择器定位到2个以上选中的元素时获得的是一个数组,可以使用each()方法遍历元素,再使用push()方法将元素存入到数组中,想要修改某个指定选中元素的value属性值时可以使用eq()来指定下标进行修改,下标从0开始.

以上三个案例获取和修改被选中的下拉选,复选框,单选框是比较常见的功能和操作,具体实现可以根据不同的选择器有很多种方法,以上几个案例只是其中之一,在实际工作中请自行搭配选择器使用.

1. **JQery常用的事件及其他相关函数有哪些? (面)(能说出来至少10种以上)**

绑定事件函数:

$obj.click():单击事件

$.obj. dblclick():双击事件

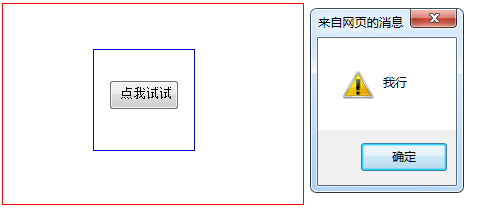
$.obj. focus():输入框获得焦点事件

$.obj. blur():输入域失去焦点

$.obj.toggle() :绑定两个或多个事件函数,单击时切换执行,常用来执行show()和hide()

$obj.on():选中的元素及其子元素绑定对应事件

**案例:div标签下有p标签,p标签下有button按钮:**



<script>

$(**function**(){

$("div").on('click',**function**(){

alert("我行");

})

});

</script>

<body>

<div>

<p>

<input type=*"button"* id=*"btn"* value=*"点我试试"*>

</p>

</div>

</body>

对div绑定单击事件时,点击div下范围内的任意后代元素都会触发绑定的事件

如果只想让父标签下某些子标签生效可以设置指定子元素,比如指定button按钮标签,这是点击p元素范围时将不再触发事件:

$("div").on('click','#btn',**function**(){

alert("我行");

});

鼠标事件函数:

$.obj.mousedown():鼠标按键按下事件

$.obj. mouseup():鼠标按键松开事件

$.obj. mouseover():鼠标指针位于元素上方悬停时触发事件

$.obj. mouseout():鼠标指针离开元素时触发事件

动画函数:

$.obj.show():显示元素

$.obj.hide():隐藏元素

$.obj. fadeIn():用于淡入已隐藏的元素

$.obj. fadeOut():用于淡出可见元素

$.obj. fadeTo():将被选元素的不透明度逐渐地改变为指定的值。

$.obj. fadeToggle():可在 fadeIn() 与 fadeOut() 方法之间进行切换.

$.obj. slideDown():用于向下滑动显示关闭的元素

$.obj. slideUp():用于向上滑动关闭显示的元素

$.obj. slideToggle():在 slideDown() 与 slideUp() 方法之间进行切换。

数据操作函数:

$.obj. data("绑定名",绑定值):向被选元素绑定数据.

$.obj. data("绑定名"):向被选元素中获取绑定数据

$.obj. removeData("绑定名"):删除之前通过 data() 方法绑定的数据.

注:data()相当于setAttribute()和getAttribute()的用法,只能向html标签元素上绑定数据.

键盘事件函数:

keydown()：在键盘按下时触发事件。

keyup()：在按下键盘起来后触发事件。

keypress()：在敲击按键时触发，我们可以理解为按下并抬起同一个按键。

如果想要指定某个键触发事件, 每个按键都有对应的ascII码,可以获得按键对应的码值,通过码值进行判断来执行,event(事件)下有两个属性可以获得:

event.keyCode: 获取按下的键盘按键的码值

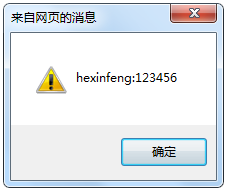
**event.** **which:** 获取按下的键盘按键的码值,作用于keyCode一样,用哪个都行.

另外在组合键中ctrl按键使用会很多,还有一个属性可以监测ctrl键是否被按下并保持:

event.ctrlKey: 监测ctrl键是否被按下并保持,按下并保持返回true,反之false

**案例:登录页面不点击提交,使用Enter(回车键)提交登录请求,Enter的码值是13.**

****

****

<script>

$(**function**(){

$("#pwd").keyup(**function**(e){

**if**(e.keyCode===13){

alert(e.keyCode);

$('#btn').click(getTest());

}

**function** getTest(){

**var** name=$('#username').val();

**var** pwd=$('#pwd').val();

alert(name+":"+pwd);}

});

});

</script>

<body>

<form id=*"good"*>

账号:<input type=*"text"* id=*"username"*>

密码:<input type=*"password"* id=*"pwd"*><br>

<input type=*"button"* value=*"提交"* id=*"btn"*>

</form>

</body>

首先在密码框上绑定键盘按下事件keyup,传入事件监听函数,通过event(事件)监听在

密码框按下的按钮是否是Enter,比对码值,如果是13则代表Enter被按下,进行if语句

执行alert弹窗输出Enter的码值13,然后在按钮上绑定单击事件并直接执行getTest()

函数,获取用户输入的账号和密码,然后将账号密码通过弹窗显示.

注:单击事件内传入的方法函数如果不带括号,比如传入getTest那么则是在点击时触发,

如果传入的方法函数带有括号,比如getTest()则是立即触发执行不经过单击.

另:为什么要将监听键盘按钮事件绑定在密码框上,用户习惯问题,一般都是输入完密码

后直接敲回车提交登录请求,如果绑定在账号框里假设用户无输入了Enter键,密码还没

有输入则进行提交会提交失败,另绑定在按钮上的原因:如果绑定在按钮上需要先点击选

择按钮,然后按Enter才会生效,所以存在技术实现问题,所以绑定在密码框上最合适的,

如果有验证码框,那么则需要绑定在验证码框上合适一些.

**案例:演示Ctrl+c组合按键,按键c的码值是67.**





<script>

$(**function**(){

$("#test").keyup(**function**(e){

alert(e.ctrlKey);

**if**(e.ctrlKey && e.keyCode===67){

alert("你按下了Ctrl+c");}

});

});

</script>

<body>

<form id=*"good"*>

请输入信息:<textarea id=*"test"*>

2018年8月29日 雨天

</textarea>

</form>

</body>

在文本域内使用Ctrl+c先被event(事件)监测到Ctrl已被按下弹出框显示true,然后进

入if语句进行与运算,e.ctrlKey返回的是true并且监测到c的码值等于67,执行alert

弹出框弹出显示你按下了Ctrl+c.

以下是键盘事件相关按键对应的码值:





