目录

[javascript 3](#_Toc389776688)

[Javascript数字相加的时候要做的处理:(否者就是字符串相加) 3](#_Toc389776689)

[javascript设计模式: 3](#_Toc389776690)

[Javascript 日期格式化 4](#_Toc389776691)

[JSON.parse()和JSON.stringify() 6](#_Toc389776692)

[windows 8](#_Toc389776693)

[sc create命令行添加/创建/修改服务 8](#_Toc389776694)

[服务器:tomcat 10](#_Toc389776695)

[java.lang.OutOfMemoryError：PermGen space 或者Java heap space的产生原因以及解决办法 10](#_Toc389776696)

[**Tomcat version 6.0 only supports J2EE 1.2, 1.3, 1.4, and Java EE 5 Web modules** 12](#_Toc389776697)

[关于The APR based Apache Tomcat Native library警告 13](#_Toc389776698)

[eclipse下启动tomcat出现Setting property 'source' to 'org.eclipse.jst.jee.server: '错误的解决办法 14](#_Toc389776699)

[Java 14](#_Toc389776700)

[微软面试题——反转字符串 14](#_Toc389776701)

[J ava字符串反转的三种实现方法 17](#_Toc389776702)

[Java中的instanceof关键字 18](#_Toc389776703)

[Java的native关键字 20](#_Toc389776704)

[乱码处理: 23](#_Toc389776705)

[String intern()方法 30](#_Toc389776706)

[JVM发生OOM时，自动生成DUMP文件。 30](#_Toc389776707)

[技术必知点 31](#_Toc389776708)

[MIME笔记 31](#_Toc389776709)

[数据库 35](#_Toc389776710)

[Oracle 35](#_Toc389776711)

[查看oracel是32位还是64位的 35](#_Toc389776712)

[方法一：使用sqlplus 35](#_Toc389776713)

[方法二： 查看v$version 视图 36](#_Toc389776714)

[方法三：查看v$sql 视图 37](#_Toc389776715)

[Mysql 39](#_Toc389776716)

[Mysql之SQL语句优化: 39](#_Toc389776717)

[查看数据库的编码: 40](#_Toc389776718)

[查看数据库详细信息找到占用时间长的Show full processlist 40](#_Toc389776719)

[绿色版mysql注册卸载服务方法 41](#_Toc389776720)

[mysql5.6.16绿色版配置、运行：包括修改root用户的密码 42](#_Toc389776721)

[SQLSERVE问题总结： 45](#_Toc389776722)

[总结JDBC连接SQLServer的错误Error establishing socket 45](#_Toc389776723)

[SQL 46](#_Toc389776724)

[SQL查询null和””的用法: 46](#_Toc389776725)

[Eclipse:使用总结： 47](#_Toc389776726)

[修改窗口的背景色 47](#_Toc389776727)

[Svn： 48](#_Toc389776728)

[解决subclipse同步冲突问题A conflict in the working copy obstructs the current operation 48](#_Toc389776729)

[办公技巧 49](#_Toc389776730)

[word 49](#_Toc389776731)

[Word自动生成页码: 49](#_Toc389776732)

[Excel 50](#_Toc389776733)

[Powerpoint 50](#_Toc389776734)

# javascript

## Javascript数字相加的时候要做的处理:(否者就是字符串相加)

|  |
| --- |
| var   a   =   3;      var   b   =   98           c   =   a   +   b      想得到c=101      确变成了字符串拼接，得到了398           我该则么做呢？  c   =   parseInt(a)   +   parseInt(b) |

## javascript设计模式:

|  |
| --- |
| Javascript是一种弱类型的语言,其基本类型就三种:字符串,数值型.布尔型,:相互之间的转换:toString()方法可以把数值型,Boolean型转换为字符串:parseFloat(),parseInt可以把字符串转化为数值型双重非可以把字符串或者数值型变成boolean:  Var Boolean=!!num;  函数是最高等的对象:  Javascript的闭包:  三、闭包的概念  闭包就是能够读取其他函数内部变量的函数。  由于在Javascript语言中，只有函数内部的子函数才能读取局部变量，因此可以把闭包简单理解成"定义在一个函数内部的函数"。  所以，在本质上，闭包就是将函数内部和函数外部连接起来的一座桥梁。 |

## Javascript 日期格式化

|  |
| --- |
| /\*\*  \* 时间格式化 返回格式化的时间  \* @param date {object} 可选参数，要格式化的data对象，没有则为当前时间  \* @param fomat {string} 格式化字符串，例如：'YYYY年MM月DD日 hh时mm分ss秒 星期' 'YYYY/MM/DD week' (中文为星期，英文为week)  \* @return {string} 返回格式化的字符串  \*  \* 例子:  \* formatDate(new Date("january 01,2012"));  \* formatDate(new Date());  \* formatDate('YYYY年MM月DD日 hh时mm分ss秒 星期 YYYY-MM-DD week');  \* formatDate(new Date("january 01,2012"),'YYYY年MM月DD日 hh时mm分ss秒 星期 YYYY/MM/DD week');  \*  \* 格式：  \* YYYY：4位年,如1993  　　 \*　　YY：2位年,如93  　　 \*　　MM：月份  　　 \*　　DD：日期  　　 \*　　hh：小时  　　 \*　　mm：分钟  　　 \*　　ss：秒钟  　　 \*　　星期：星期，返回如 星期二  　　 \*　　周：返回如 周二  　　 \*　　week：英文星期全称，返回如 Saturday  　　 \*　　www：三位英文星期，返回如 Sat  \*/  function formatDate(date, format) {  if (arguments.length < 2 && !date.getTime) {  format = date;  date = new Date();  }  typeof format != 'string' && (format = 'YYYY年MM月DD日 hh时mm分ss秒');  var week = ['Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', '日', '一', '二', '三', '四', '五', '六'];  return format.replace(/YYYY|YY|MM|DD|hh|mm|ss|星期|周|www|week/g, function(a) {  switch (a) {  case "YYYY": return date.getFullYear();  case "YY": return (date.getFullYear()+"").slice(2);  case "MM": return date.getMonth() + 1;  case "DD": return date.getDate();  case "hh": return date.getHours();  case "mm": return date.getMinutes();  case "ss": return date.getSeconds();  case "星期": return "星期" + week[date.getDay() + 7];  case "周": return "周" + week[date.getDay() + 7];  case "week": return week[date.getDay()];  case "www": return week[date.getDay()].slice(0,3);  }  });  } |
| 用法：第一个参数传一个Date对象（没有则使用当前时间），第二个参数是格式化字符串，格式如下：  　　　　YYYY：4位年,如1993  　　　　YY：2位年,如93  　　　　MM：月份  　　　　DD：日期  　　　　hh：小时  　　　　mm：分钟  　　　　ss：秒钟  　　　　星期：星期，返回如 星期二  　　　　周：返回如 周二  　　　　week：英文星期全称，返回如 Saturday  　　　　www：三位英文星期，返回如 Sat  　　示例：  \* formatDate(new Date("january 01,2012"));  \* formatDate(new Date());  \* formatDate('YYYY年MM月DD日 hh时mm分ss秒 星期 YYYY-MM-DD YY年 week');  \* formatDate(new Date("january 01,2012"),'YYYY年MM月DD日 hh时mm分ss秒 星期/周 www YYYY/MM/DD week'); |

## JSON.parse()和JSON.stringify()

|  |
| --- |
| parse用于从一个字符串中解析出json对象,如  var str = '{"name":"huangxiaojian","age":"23"}'  结果：  JSON.parse(str)  Object   1. age: "23" 2. name: "huangxiaojian" 3. \_\_proto\_\_: Object   注意：单引号写在{}外，每个属性名都必须用双引号，否则会抛出异常。    stringify()用于从一个对象解析出字符串，如  var a = {a:1,b:2}  结果：  JSON.stringify(a)  "{"a":1,"b":2}" |

## 判断Javascript变量是否为空

|  |
| --- |
| 在不确定变量类型的情况下，判断Javascript变量是否为空不要使用：  if(tmp){}  上述判断的问题在于，例如：当tmp的值为数字0时，也会返回false。  所以尽量使用：  if(tmp != null){}    下面是if(tmp){}判断的结果：  if(tmp) if(tmp != null)  tmp未定义时 //false //false  var tmp; //false //false  var tmp = 0; //false //true  var tmp = 10; //true //true  var tmp = ""; //false //true  var tmp = "ddd"; //true //true  var tmp = 0.0; //false //true  var tmp = new String(); //true //true  结论：要判断tmp未定义或者为null可以使用if(tmp != null)，不能使用if(tmp) |

## javascript数组使用调用方法汇总

|  |
| --- |
| JS数组方法汇总  shift：删除原数组第一项，并返回删除元素的值；如果数组为空则返回undefined  var a = [1,2,3,4,5];  var b = a.shift(); //a：[2,3,4,5] b：1  unshift：将参数添加到原数组开头，并返回数组的长度  var a = [1,2,3,4,5];  var b = a.unshift(-2,-1); //a：[-2,-1,1,2,3,4,5] b：7  注：在IE6.0下测试返回值总为undefined，FF2.0下测试返回值为7，所以这个方法的返回值不可靠，需要用返回值时可用splice代替本方法来使用。  pop：删除原数组最后一项，并返回删除元素的值；如果数组为空则返回undefined  var a = [1,2,3,4,5];  var b = a.pop(); //a：[1,2,3,4] b：5  push：将参数添加到原数组末尾，并返回数组的长度  var a = [1,2,3,4,5];  var b = a.push(6,7); //a：[1,2,3,4,5,6,7] b：7  concat：返回一个新数组，是将参数添加到原数组中构成的  var a = [1,2,3,4,5];  var b = a.concat(6,7); //a：[1,2,3,4,5] b：[1,2,3,4,5,6,7]  splice(start,deleteCount,val1,val2,...)：从start位置开始删除deleteCount项，并从该位置起插入val1,val2,...  var a = [1,2,3,4,5];  var b = a.splice(2,2,7,8,9); //a：[1,2,7,8,9,5] b：[3,4]  var b = a.splice(0,1); //同shift  a.splice(0,0,-2,-1); var b = a.length; //同unshift  var b = a.splice(a.length-1,1); //同pop  a.splice(a.length,0,6,7); var b = a.length; //同push  reverse：将数组反序  var a = [1,2,3,4,5];  var b = a.reverse(); //a：[5,4,3,2,1] b：[5,4,3,2,1]  sort(orderfunction)：按指定的参数对数组进行排序  var a = [1,2,3,4,5];  var b = a.sort(); //a：[1,2,3,4,5] b：[1,2,3,4,5]  slice(start,end)：返回从原数组中指定开始下标到结束下标之间的项组成的新数组  var a = [1,2,3,4,5];  var b = a.slice(2,5); //a：[1,2,3,4,5] b：[3,4,5]  join(separator)：将数组的元素组起一个字符串，以separator为分隔符，省略的话则用默认用逗号为分隔符  var a = [1,2,3,4,5];  var b = a.join("|"); //a：[1,2,3,4,5] b："1|2|3|4|5" |

## javascript arguments

|  |
| --- |
| **什么是arguments** **arguments** **是是JavaScript里的一个内置对象，它很古怪，也经常被人所忽视，但实际上是很重要的。所有主要的js函数库都利用了arguments对象。所以agruments对象对于javascript程序员来说是必需熟悉的。**  所有的函数都有属于自己的一个arguments对象，它包括了函所要调用的参数。他不是一个数组，如果用typeof arguments，返回的是’object’。虽然我们可以用调用数据的方法来调用arguments。比如length,还有index方法。但是数 组的push和pop对象是不适用的。 **创建一个灵活的函数** 看起来貌似argument对象使用起来十分有限，但是实际上它是一个非常有用的对象。你可以通过使用argument对象让函数能够调用数量不定 的参数。在Dean Edwards的base2库里有个格式化的函数，展示了这个灵活性。  function format(string) {    var args = arguments;    var pattern = new RegExp(“%([1-" + arguments.length + "])”, ”g”);    return String(string).replace(pattern, function(match, index) {      return args[index];    });  };  我们提供了一个模板字符串，你可以用”%1”到“%9”给返回值添加一个占位符。然后提供给九个其他参数插入。  format(“And the %1 want to know whose %2 you %3″, ”papers”, ”shirt”, ”wear”);  上面的代码将返回：And the papers want to know whose shirt you wear" .  有件事情我们需要注意下，在定义函数的时候，我们只规定了一个参数，string。Javascript允许我们传递任何数量的参数到一个函数里，不管怎么定义这个函数。Arguments对象对这些都是允许的。 **把arguments对象转换成一个真正的数组** 虽然arguments对象不是一个真正的javascript数组，但是我们还是可以轻易的把它转换成标准的数据 ，然后进行数组操作。  var args = Array.prototype.slice.call(arguments);  那么现在这个变量args就含有一个含有函数所有参数的标准javascript数组对象。 **通过预置的arguments对象创建函数** Arguments对象允许我们去执行所有类型的javascript方法。这边附上一个makeFunc函数的定义。这个函数允许我们去提供一个函数引用和这个函数的所有参数。他将返回一个匿名函数去调用你规定的函数，也提供了匿名函数调用时所附带的参数。  function makeFunc() {    var args = Array.prototype.slice.call(arguments);    var func = args.shift();    return function() {      return func.apply(null, args.concat(Array.prototype.slice.call(arguments)));    };  }  第一个argument对象给makeFunc提供了你想调用的函数的引用。他是从arguments数组里移除的。然后makeFunc返回了一个匿名函数去运行规定的方法。  第一个应用的argument指向了函数调用的范围，主要是函数内部关键部分所指向的。我们先保持这个为null。第二个arguments是一个 数组，会为这个函数转变为arguments对象。makeFunc把原始的数组值串联到arguments对象里提供给匿名函数和所调用函数的数组。  你需要输出一个模板总是相同的位置，这样就可以不用总是在每次引用模板的时候调用format函数。你可以使用makeFunc的通用功能去返回可以调用format然后自动补充模板的函数。  var majorTom = makeFunc(format, ”This is Major Tom to ground control. I’m %1.”);  你可以像这样调用majorTom函数：  majorTom(“stepping through the door”);  majorTom(“floating in a most peculiar way”);  每一次你调用majorTom，它会同时调用format函数和第一个argument，已经写好的模板。那么将返回  “This is Major Tom to ground control. I’m stepping through the door.”  “This is Major Tom to ground control. I’m floating in a most peculiar way.” **创建引用自身的函数** 你可能觉得这样很酷，不过，arguments还有更多的惊喜。他还有其他有用的特性：callee方法 。Arguments.callee包括了一个函数的引用去创建一个argument对象。那么该如何使用呢？  Arguments.callee方法能让一个匿名函数很方便的指向本身。  Repeat是一个承载了一个函数引用和两个数字的函数。第一个数字是函数调用几次，第二个数字是每个调用的间隔时间，单位是毫秒。  function repeat(fn, times, delay) {    return function() {      if(times– > 0) {        fn.apply(null, arguments);        var args = Array.prototype.slice.call(arguments);        var self = arguments.callee;        setTimeout(function(){self.apply(null,args)}, delay);      }    };  }  Repeat函数使用了arguments.callee方法从变量self去获取一个引用，指向运行原始指令的函数。这样，匿名函数就可以再次调用本身。  我有一段超级简介的函数，承载了一个字符串和执行alert方法。  function comms(s) {    alert(s);  }  然而，我想创建一个特殊的版本，通过这个版本我可以重复这个动作三次，每次间隔2秒钟。那么，我们可以  var somethingWrong = repeat(comms, 3, 2000);  somethingWrong(“Can you hear me, major tom?”);  调用了somethingWrong函数的结果就是复这个动作三次，每次alert间隔2秒钟。  Arguments虽然不是经常用到，有点古怪，但是，它充满了惊喜，非常值得我们去了解 。 |

## javascript中等于（==）与全等（===）的区别说明

|  |
| --- |
| javascript中等于（==）与全等（===）的区别说明  等于（==）的情况下 只要值相同就返回True。而全等（===）的时候需要值和类型都要匹配才能返回True |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

# Extjs

## 搭建开发环境不出效果在控制台看到的错: TypeError: r.prototype.styleHooks is not an object or null

|  |
| --- |
| 分析可能的原因:没有正确导入需要的js文件:(extjs提供了demo可以从里面查看怎样正确引用  解决:  <link href="extjs/resources/css/ext-all.css" rel="stylesheet" />  <script type="text/javascript" src="extjs/ext-all.js"></script>  <script type="text/javascript" src="extjs/locale/ext-lang-zh\_CN.js"></script>   引这3个就行了 其余俩的重复了 冲突了  所以才会照成这样的错误 |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

# windows

## sc create命令行添加/创建/修改服务

|  |
| --- |
| sc create命令行添加/创建/修改服务  添加服务： sc create BITS binpath= "C:\WINDOWS\system32\svchost.exe -k netsvcs" type= share start= auto displayname= "Background Intelligent Transfer Service" depend= RpcSs/EventSystem 修改描述： sc description BITS "利用空闲的网络带宽在后台传输文件。如果服务被停用，例如 Windows Update 和 MSN Explorer 的功能将无法自动下载程序和其他信息。如果此服务被禁用，任何依赖它的服务如果没有容错技术以直接通过 IE 传输文件，一旦 BITS 被禁用，就可能无法传输文件。"  添加服务： sc create PolicyAgent binpath= "C:\WINDOWS\system32\lsass.exe" type= share start= auto displayname= "IPSEC Services" depend= RPCSS/Tcpip/IPSec 修改描述： sc description PolicyAgent "提供 TCP/IP 网络上客户端和服务器之间端对端的安全。如果此服务被停用，网络上客户端和服务器之间的 TCP/IP 安全将不稳定。如果此服务被禁用，任何依赖它的服务将无法启动。"  SC命令的更多用法请在查看help sc  (注册表修改前请做好备份)，  SvcHost共享服务的特别之处：或许重启后还需要到注册表位置：[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SvcHost]，查看"C:\WINDOWS\system32\svchost.exe -k netsvcs"以红色字为名的键(这里以netsvcs为例)，键值里是否定义有sc create BITS蓝色字的服务(这里以BITS为例)，如果没有就在键值把服务名(这里以BITS为例)添加上去,再重启一次。  @echo off copy internat.exe %windir%\system32 sc create Internat binpath= %windir%\system32\internat.exe type= own start= auto DisplayName= "Internat Aionegirl" sc description Internat "输入法" sc start Internat pause  @echo off del %windir%\system32\internat.exe /s /q sc stop Internat sc delete Internat pause sc create MyService binpath= C:\\winnt\\sys tem32\\NewServ.exe type= own  start= auto sc create MyService binpath= F:\\clear.bat type= own start= auto  type=----own, share, interact, kernel, filesys  关于建立服务的类型，选项值包括驱动程序使用的类型，默认是share。  start=----boot, sys tem, auto, demand, disabled  关于启动服务的类型，选项值包括驱动程序使用的类型，默认是demand（手动）。  添加服务：  sc create BITS binpath= "C:/WINDOWS/system32/svchost.exe -k netsvcs" type= share start= auto displayname= "Background Intelligent Transfer Service" depend= RpcSs/[EVE](http://www.txwb.com/Article/wyzl/GCYX/200703/5561.html)ntSystem  修改描述：  sc description BITS "利用空闲的网络带宽在后台传输文件。如果服务被停用，例如 Windows Update 和 [MSN](http://www.jztop.com/e/search/result/?searchid=498) Explorer 的功能将无法自动下载程序和其他信息。如果此服务被禁用，任何依赖它的服务如果没有容错技术以直接通过 IE 传输文件，一旦 BITS 被禁用，就可能无法传输文件。"   添加服务：  sc create PolicyAgent binpath= "C:/WINDOWS/system32/lsass.exe" type= share start= auto displayname= "IPSEC Services" depend= RPCSS/Tcpip/IPSec  修改描述：  sc description PolicyAgent "提供 TCP/IP 网络上客户端和[服务器](http://www.jztop.com/systems/)之间端对端的安全。如果此服务被停用，网络上客户端和[服务器](http://www.jztop.com/systems/)之间的 TCP/IP 安全将不稳定。如果此服务被禁用，任何依赖它的服务将无法启动。"   SC命令的更多用法请在查看help sc   ([注册表](http://www.jztop.com/os/regedit/)修改前请做好备份)，   SvcHost共享服务的特别之处：或许重启后还需要到[注册表](http://www.jztop.com/os/regedit/)位置：[HKEY\_LOCAL\_MACHINE/SOFTWARE/Microsoft/Windows NT/CurrentVersion/SvcHost]，查看"C:/WINDOWS/system32/svchost.exe -k netsvcs"以红色字为名的键(这里以netsvcs为例)，键值里是否定义有sc create BITS蓝色字的服务(这里以BITS为例)，如果没有就在键值把服务名(这里以BITS为例)添加上去,再重启一次。  Creates a service entry in the registry and Service Database.  SYNTAX:  sc create [service name] [binPath= ] <option1> <option2>...  CREATE OPTIONS: NOTE: The option name includes the equal sign. type= <own|share|interact|kernel|filesys|rec>        (default = own) start= <boot|system|auto|demand|disabled>        (default = demand) error= <normal|severe|critical|ignore>        (default = normal) binPath= <BinaryPathName> group= <LoadOrderGroup> tag= <yes|no> depend= <Dependencies(separated by / (forward slash))> obj= <AccountName|ObjectName>        (default = LocalSystem) DisplayName= <display name> password= <password>  ***提示大家：sc cretate命令后面带的参数比如[binPath= ]的等号后面必须有个空格，没空格你怎么折腾都不行的，哈哈。微软还真搞笑啊，搞这么个功能迷惑人。同样，后面的其它参数比如group= , displayname=的等号后面都需要空格，不然sc create命令不会成功创建服务，只会返回给你帮助信息。***  ***sc create nlsvc binpath= "F:\Program Files\NetLimiter 2 Lite\NLClient.exe" type= own start= demand DisplayName= NetLimiter*** |

# 服务器:tomcat

## java.lang.OutOfMemoryError：PermGen space 或者Java heap space的产生原因以及解决办法

|  |
| --- |
| 这里以tomcat环境为例，其它WEB服务器如jboss,weblogic等是同一个道理。  一、java.lang.OutOfMemoryError: PermGen space  PermGen space的全称是Permanent Generation space,是指内存的永久保存区域,  这块内存主要是被JVM存放Class和Meta信息的,Class在被Loader时就会被放到PermGen space中,  它和存放类实例(Instance)的Heap区域不同,GC(Garbage Collection)不会在主程序运行期对  PermGen space进行清理，所以如果你的应用中有很多CLASS的话,就很可能出现PermGen space错误,  这种错误常见在web服务器对JSP进行pre compile的时候。如果你的WEB APP下都用了大量的第三方jar, 其大小  超过了jvm默认的大小(4M)那么就会产生此错误信息了。  解决方法： 手动设置MaxPermSize大小  修改TOMCAT\_HOME/bin/catalina.sh  在“echo "Using CATALINA\_BASE: $CATALINA\_BASE"”上面加入以下行：  JAVA\_OPTS="-server -XX:PermSize=64M -XX:MaxPermSize=128m  建议：将相同的第三方jar文件移置到tomcat/shared/lib目录下，这样可以达到减少jar 文档重复占用内存的目的。  二、java.lang.OutOfMemoryError: Java heap space  Heap size 设置  JVM堆的设置是指java程序运行过程中JVM可以调配使用的内存空间的设置.JVM在启动的时候会自动设置Heap size的值，  其初始空间(即-Xms)是物理内存的1/64，最大空间(-Xmx)是物理内存的1/4。可以利用JVM提供的-Xmn -Xms -Xmx等选项可  进行设置。Heap size 的大小是Young Generation 和Tenured Generaion 之和。  提示：在JVM中如果98％的时间是用于GC且可用的Heap size 不足2％的时候将抛出此异常信息。  提示：Heap Size 最大不要超过可用物理内存的80％，一般的要将-Xms和-Xmx选项设置为相同，而-Xmn为1/4的-Xmx值。  解决方法：手动设置Heap size  修改TOMCAT\_HOME/bin/catalina.sh  在“echo "Using CATALINA\_BASE: $CATALINA\_BASE"”上面加入以下行：  JAVA\_OPTS="-server -Xms800m -Xmx800m -XX:MaxNewSize=256m"  三、实例，以下给出1G内存环境下java jvm 的参数设置参考：  JAVA\_OPTS="-server -Xms800m -Xmx800m -XX:PermSize=64M -XX:MaxNewSize=256m -XX:MaxPermSize=128m -Djava.awt.headless=true "  大家都知道，JAVA程序启动时都会JVM都会分配一个初始内存和最大内存给这个应用程序。这个初始内存和最大内存在一定程度都会影响程序的性能。比如说在应用程序用到最大内存的时候，JVM是要先去做垃圾回收的动作，释放被占用的一些内存。  所以想调整Tomcat的启动时初始内存和最大内存就需要向JVM声明，一般的JAVA程序在运行都可以通过中-Xms -Xmx来调整应用程序的初始内存和最大内存:  如：java -Xms64m -Xmx128m a.jar.  tomcat的启动程序是包装过的，不能直接使用java -X..... tomcat.\*来改变内存的设置。在Tomcat在改变这个设置  有两种方法：  1. 就需要在环境变量中加上TOMCAT\_OPTS, CATALINA\_OPTS两个属性，  如 SET CATALINA\_OPTS= -Xms64m -Xmx512m;  ms是最小的，mx是最大，64m, 512m分别是指内存的容量.  2. 修改Catalina.bat文件  在166行“rem Execute Java with the applicable properties ”以下每行  %\_EXECJAVA% %JAVA\_OPTS% %CATALINA\_OPTS% %DEBUG\_OPTS% -Djava.endorsed.dirs="%JAVA\_ENDORSED\_DIRS%" -classpath "%CLASSPATH%" -Dcatalina.base="%CATALINA\_BASE%" -Dcatalina.home="%CATALINA\_HOME%" -Djava.io.tmpdir="%CATALINA\_TMPDIR%" %MAINCLASS% %CMD\_LINE\_ARGS% %ACTION% 中的%CATALINA\_OPTS% 替换成  -Xms64m -Xmx512m |

## [**Tomcat version 6.0 only supports J2EE 1.2, 1.3, 1.4, and Java EE 5 Web modules**](http://www.cnblogs.com/chanedi/articles/2112477.html)

最近在用eclipse做项目，新建项目时什么都贪新，用最新的版本，在Dynamic web module version栏里选了最新的3.0版本，布署项目的时候就出现了如期的错误，在网上看到[http://hi.baidu.com/yolanda441/b ... 41d36dd1164e2e.html](http://hi.baidu.com/yolanda441/blog/item/124b4cd9ea41d36dd1164e2e.html)这位兄弟的博客,解决了问题，现将方法贴于此，希望对别人有所帮助：  
  
《通过search，发现在project的.setting folder下面，有个名为org.eclipse.wst.common.project.facet.core.xml的文件，里面配置有各种版本信息。此时，按照本机配置修改这个文件，问题就解决了。  
  
<?xml version="1.0" encoding="UTF-8"?>  
<faceted-project>  
<runtime name="Apache Tomcat v5.5"/>  
<fixed facet="jst.web"/>  
<fixed facet="jst.java"/>  
<installed facet="jst.java" version="5.0"/>  
<installed facet="jst.web" version="2.4"/>  
<installed facet="wst.jsdt.web" version="1.0"/>  
</faceted-project>  
》  
  
像我这样的错误，可以下载Tomcat 7。0解决，不过我闲麻烦，干脆就在配置文件中把<installed facet="jst.web" version="3.0"/> 改成低些的版本version="2.5" 就好了

## [关于The APR based Apache Tomcat Native library警告](http://jasonhan-sh-hotmail-com.iteye.com/blog/1472911)

  今天在Eclipse上配置Tomcat7，启动时看到如下警告信息：

The APR based Apache Tomcat Native library which allows optimal performance in production environments was not found on the java.library.path

    感到很不舒服，于是在网上找了些解决办法。

    以下内容来自：http://cooliron.blog.163.com/blog/static/124703138201112211571276/

    在eclipse中，启动tomcat时会出现如此信息：The APR based Apache Tomcat Native library which allows optimal performance in production environments was not found on the java.library.path，它其实不影响程序的正常运行，但看到了，感觉实在别扭。其解决办法如下：

<http://tomcat.heanet.ie/native/> 下载tcnative-1.dll，然后放到你的JDK路径下的bin目录下。重新启动tomcat，就没有问题了。

    这个tcnative-1.dll 他的作用如下：

    tomcat整合本地apr会使效率提升：  
    一是，处理静态资源的时候速度更快，（注：有人用jmeter，对使用apr前后的tomcat进行压力测试，结果显示，性能是有一定的提升，但是没有想象中多，只有一点点，当然，这与运行的逻辑有关，但是我个人认为在大多数情况下这部分性能提升甚微）。  
    二是，我认为提升比较多的是对ssl的处理效率，当tomcat处理https的请求是，如果使用本地的openssl库，肯定会比前面提升的效率高。（虽然没有实际测试数据，但我推测这里的性能提升比例上要比第一点多得多）。

    原因上面已经说得很清楚了，下面说下我的解决办法。

    首先，tcnative-1.dll不需要下载，至少对于Tomcat7来说是这样，我们可以在其bin目录下找到这个dll；

    其次，tcnative-1.dll放在哪儿不要紧，只要在系统的PATH中可以找到就行。

   解决：

   在系统的环境变量中添加CATALINA\_HOME，它指向你的Tomcat根目录，接着在PATH路径中添加%CATALINA\_HOME%\bin;

   这样就可以解决了，对了，设置完成后，可能需要重启下Eclipse。

# [eclipse下启动tomcat出现Setting property 'source' to 'org.eclipse.jst.jee.server: '错误的解决办法](http://www.cnblogs.com/tonyzzp/archive/2010/04/11/eclipse_tomcat_setting_property_source_solution.html)

|  |
| --- |
| 在eclipse中启动tomcat时出现Setting property 'source' to 'org.eclipse.jst.jee.server:你的站点名'   did not find a matching property错误    解决办法：  1、在server控制台内，在服务器上点右键--属性  2、general选项卡中点switch location  这时，location变为:/servers/tomcat6.0 server at localhost.server  3、在project explore中找到tomcat项目--Tomcat v6.0 Server at localhost.server  这个文件 ，双击打开  4、在最下面的server option里选中publis module context to separate xml file.保存。（保存时要先停止服务器） |

# Java

## [微软面试题——反转字符串](http://www.cnblogs.com/k-eckel/articles/195931.html)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 这是网络流传的Microsoft的面试题目之一：“编写反转字符串的程序，要求优化速度、优化空间”。因为最近一直很多关注算法方面的实践和研究，因此对这个问题进行了一些思考，给出了5种实现方法（有两种解法相关性比较大）。  解法一：第一次看到这题目，想到最简单、最直觉的解法就是：遍历字符串，将第一个字符和最后一个交换，第二个和倒数第二个交换，依次循环，即可，于是有了第一个解法：   |  | | --- | | char\* strrev1(const char\* str)  {         int len = strlen(str);         char\* tmp = new char[len + 1];         strcpy(tmp,str);           for (int i = 0; i < len/2; ++i)         {                char c = tmp[i];          tmp[i] = tmp[len – i - 1];          tmp[len – i - 1] = c;         }           return tmp;  } |   这里是通过数组的下标方式访问字符串的字符，实际上用指针直接操作即可。解法二正是基于此，实现代码为：   |  | | --- | | char\* strrev2(const char\* str)  {         char\* tmp = new char[strlen(str) + 1];         strcpy(tmp,str);         char\* ret = tmp;           char\* p = tmp + strlen(str) - 1;           while (p > tmp)         {                char t = \*tmp;                \*tmp = \*p;                \*p = t;                  --p;                ++tmp;         }           return ret;  } |   显然上面的两个解法中没有考虑时间和空间的优化，一个典型的优化策略就是两个字符交换的算法优化，我们可以完全不使用任何外部变量即完成两个字符（或者整数）的交换，这也是一个很经典的面试题目。特别是一些嵌入式硬件相关编程中经常要考虑寄存器的使用，因此经常有不使用任何第三个寄存器即完成两个寄存器数据的交换的题目。一般有两个解法，对应这里的解法三和解法四。  解法三的实现代码为：   |  | | --- | | char\* strrev3(const char\* str)  {         char\* tmp = new char[strlen(str) + 1];         strcpy(tmp,str);         char\* ret = tmp;           char\* p = tmp + strlen(str) - 1;           while (p > tmp)         {                \*p ^= \*tmp;                \*tmp ^= \*p;                \*p ^= \*tmp;                  --p;                ++tmp;         }           return ret;  } |   解法四的实现代码为：   |  | | --- | | char\* strrev4(const char\* str)  {         char\* tmp = new char[strlen(str) + 1];         strcpy(tmp,str);         char\* ret = tmp;           char\* p = tmp + strlen(str) - 1;           while (p > tmp)         {                \*p = \*p + \*tmp;                \*tmp = \*p - \*tmp;                \*p = \*p - \*tmp;                  --p;                ++tmp;         }           return ret;  } |   实际上我们还可以通过递归的思想来解决这个问题，思想很简单：每次交换首尾两个字符，中间部分则又变为和原来字符串同样的问题，因此可以通过递归的思想来解决这个问题，对应解法五的实现代码为：   |  | | --- | | char\* strrev5(/\*const \*/char\* str,int len)  {         if (len <= 1)                return str;           char t = \*str;         \*str = \*(str + len -1);         \*(str + len -1) = t;           return (strrev5(str + 1,len - 2) - 1);  } |   以下给出一个测试程序：   |  | | --- | | int main(int argc,char\* argv[])  {         char\* str = "hello";         P(str);           char\* str2 = strrev1(str);         P(str2);           char\* str3 = strrev2(str2);         P(str3);           char\* str4 = strrev3(str3);         P(str4);           char\* str5 = strrev4(str4);         P(str5);           char\* str6 = strrev5(str5,strlen(str5));         P(str6);           return 0;  } |     你就可以看到字符串"hello"和"olleh"交替输出了。    说明：1）这里解法中没有认真考虑输入字符串的合法性和特殊长度（如NULL、一个字符等）字符串的处理；2）前4个算法不改变输入字符串的值，解法五修改了输入字符串。 |

## J ava字符串反转的三种实现方法

|  |
| --- |
| public class StringTest{  public void convert1(String s){    for(int i= s.length()-1;i>=0;i--){     System.out.print(s.charAt(i));    }    System.out.println (); }  public void convert2(String s){    char[] str = s.toCharArray();    for(int i=str.length;i>0;i--){      System.out.print(s.charAt(i-1));    }      System.out.println ();  }  public void convert3(String s){    StringBuffer sb = new StringBuffer(s);    System.out.println (sb.reverse().toString());     }  public static void main(String[] args){    String s = "农民工集体被袭,谁来保障他们.";    StringTest st = new StringTest();    System.out.println ("发转前的字符串是：");     System.out.println (s);     System.out.println ("发转后的字符串是：");     st.convert1(s);    st.convert2(s);    st.convert3(s); } } |

## Java中的instanceof关键字

|  |
| --- |
| instanceof是Java的一个二元操作符，和==，>，<是同一类东东。由于它是由字母组成的，所以也是Java的保留关键字。它的作用是测试它左边的对象是否是它右边的类的实例，返回boolean类型的数据。举个例子：  　　String s = "I AM an Object!"; 　　boolean isObject = s instanceof Object;  　　我们声明了一个String对象引用，指向一个String对象，然后用instancof来测试它所指向的对象是否是Object类的一个实例，显然，这是真的，所以返回true，也就是isObject的值为True。 　　instanceof有一些用处。比如我们写了一个处理账单的系统，其中有这样三个类：  　　public class Bill {//省略细节} 　　public class PhoneBill extends Bill {//省略细节} 　　public class GasBill extends Bill {//省略细节}  　　在处理程序里有一个方法，接受一个Bill类型的对象，计算金额。假设两种账单计算方法不同，而传入的Bill对象可能是两种中的任何一种，所以要用instanceof来判断：  　　public double calculate(Bill bill) { 　　if (bill instanceof PhoneBill) { 　　//计算电话账单 　　} 　　if (bill instanceof GasBill) { 　　//计算燃气账单 　　} 　　... 　　} 　　这样就可以用一个方法处理两种子类。  　　然而，这种做法通常被认为是没有好好利用面向对象中的多态性。其实上面的功能要求用方法重载完全可以实现，这是面向对象变成应有的做法，避免回到结构化编程模式。只要提供两个名字和返回值都相同，接受参数类型不同的方法就可以了：  　　public double calculate(PhoneBill bill) { 　　//计算电话账单 　　}  　　public double calculate(GasBill bill) { 　　//计算燃气账单 　　}  　　所以，使用instanceof在绝大多数情况下并不是推荐的做法，应当好好利用多态。 |

## [Java的native关键字](http://blog.csdn.net/hlantian/article/details/4268666)

|  |
| --- |
| java native关键字   一. 什么是Native Method    简单地讲，一个Native Method就是一个java调用非java代码的接口。一个Native Method是这样一个java的方法：该方法的实现由非java语言实现，比如C。这个特征并非java所特有，很多其它的编程语言都有这一机制，比如在C＋＋中，你可以用extern "C"告知C＋＋编译器去调用一个C的函数。    "A native method is a Java method whose implementation is provided by non-java code."    在定义一个native method时，并不提供实现体（有些像定义一个java interface），因为其实现体是由非java语言在外面实现的。，下面给了一个示例：  package java.lang;  public class Object {      ......      public final native Class<?> getClass();      public native int hashCode();      protected native Object clone() throws CloneNotSupportedException;      public final native void notify();      public final native void notifyAll();      public final native void wait(long timeout) throws InterruptedException;      ......  }          标识符native可以与所有其它的java标识符连用，但是abstract除外。这是合理的，因为native暗示这些方法是有实现体的，只不过这些实现体是非java的，但是abstract却显然的指明这些方法无实现体。native与其它java标识符连用时，其意义同非Native Method并无差别。      一个native method方法可以返回任何java类型，包括非基本类型，而且同样可以进行异常控制。这些方法的实现体可以自制一个异常并且将其抛出，这一点与java的方法非常相似。     native method的存在并不会对其他类调用这些本地方法产生任何影响，实际上调用这些方法的其他类甚至不知道它所调用的是一个本地方法。JVM将控制调用本地方法的所有细节。      如果一个含有本地方法的类被继承，子类会继承这个本地方法并且可以用java语言重写这个方法（这个似乎看起来有些奇怪），同样的如果一个本地方法被fianl标识，它被继承后不能被重写。    本地方法非常有用，因为它有效地扩充了jvm.事实上，我们所写的java代码已经用到了本地方法，在sun的java的并发（多线程）的机制实现中，许多与操作系统的接触点都用到了本地方法，这使得java程序能够超越java运行时的界限。有了本地方法，java程序可以做任何应用层次的任务。  二.为什么要使用Native Method    java使用起来非常方便，然而有些层次的任务用java实现起来不容易，或者我们对程序的效率很在意时，问题就来了。    与java环境外交互：    有时java应用需要与java外面的环境交互。这是本地方法存在的主要原因，你可以想想java需要与一些底层系统如操作系统或某些硬件交换信息时的情况。本地方法正是这样一种交流机制：它为我们提供了一个非常简洁的接口，而且我们无需去了解java应用之外的繁琐的细节。    与操作系统交互：    JVM支持着java语言本身和运行时库，它是java程序赖以生存的平台，它由一个解释器（解释字节码）和一些连接到本地代码的库组成。然而不管怎样，它毕竟不是一个完整的系统，它经常依赖于一些底层（underneath在下面的）系统的支持。这些底层系统常常是强大的操作系统。通过使用本地方法，我们得以用java实现了jre的与底层系统的交互，甚至JVM的一些部分就是用C写的，还有，如果我们要使用一些java语言本身没有提供封装的操作系统的特性时，我们也需要使用本地方法。     Sun's Java     Sun的解释器是用C实现的，这使得它能像一些普通的C一样与外部交互。jre大部分是用java实现的，它也通过一些本地方法与外界交互。例如：类java.lang.Thread 的 setPriority()方法是用java实现的，但是它实现调用的是该类里的本地方法setPriority0()。这个本地方法是用C实现的，并被植入JVM内部，在Windows 95的平台上，这个本地方法最终将调用Win32 SetPriority() API。这是一个本地方法的具体实现由JVM直接提供，更多的情况是本地方法由外部的动态链接库（external dynamic link library）提供，然后被JVM调用。  三.JVM怎样使Native Method跑起来：     我们知道，当一个类第一次被使用到时，这个类的字节码会被加载到内存，并且只会回载一次。在这个被加载的字节码的入口维持着一个该类所有方法描述符的list，这些方法描述符包含这样一些信息：方法代码存于何处，它有哪些参数，方法的描述符（public之类）等等。     如果一个方法描述符内有native，这个描述符块将有一个指向该方法的实现的指针。这些实现在一些DLL文件内，但是它们会被操作系统加载到java程序的地址空间。当一个带有本地方法的类被加载时，其相关的DLL并未被加载，因此指向方法实现的指针并不会被设置。当本地方法被调用之前，这些DLL才会被加载，这是通过调用java.system.loadLibrary()实现的。       最后需要提示的是，使用本地方法是有开销的，它丧失了java的很多好处。如果别无选择，我们可以选择使用本地方法。    可以将native方法比作Java程序同Ｃ程序的接口，其实现步骤： １、在Java中声明native()方法，然后编译； ２、用javah产生一个.h文件； ３、写一个.cpp文件实现native导出方法，其中需要包含第二步产生的.h文件（注意其中又包含了JDK带的jni.h文件）； ４、将第三步的.cpp文件编译成动态链接库文件； ５、在Java中用System.loadLibrary()方法加载第四步产生的动态链接库文件，这个native()方法就可以在Java中被访问了。    具体实现方法可以从网上查找，这里就不写了  不过，又引出两个东西：javah.exe命令和JNI    自己写了一个调用VB的DLL的例子，不过还没写完，调用函数部分不会写，只写了加载DLL的过程  JAVA部分：  public class TestNativeMothod {      public static void main(String[] args) {                    //PROCESS LOGIC                }            public native String callNativeMothod();            static {           System.loadLibrary("TestNativeMethodProj");      }  } VB部分：  Option Explicit  Public Function callNativeMothod() As String      callNativeMothod = "This is a method implemented by VB"  End Function |

## 乱码处理:

|  |
| --- |
| 1.JSP里CHARSET要为GB2312    contentType="text/html;charset=GB2312"  2.SERVERLET类里要有    request.setCharacterEncoding(gb2312);  3.字符集的重新格式化  java.net.URLEncoder  java.net.URLDecoder 对应的encode 与decode进行编码解码。  在传输数据时候浏览器会自动对要表单传输的数据进行url编码，所使用的编码方式取决于当前网页显示时候使用的编码方式。  对于http请求消息的url地址后的参数，getparameter等方法进行自动url解码时采用的编码方式取决servlet引擎；tomcat默认用ISO8859-1进行解码。  重新格式化语句如下：  String str1 = new String(request.getParameter("name").getBytes("ISO-8859-1"),"gb2312");  4.TOMCAT的CONF文件夹里的SERVER.XML中大约92行左右.(如果你没改过)   找到connector区块，加入如下一行：  URIEncoding="GBK" 或 URIEncoding="GB2312" 或 URIEncoding="UTF-8"    完整的应如下：  <Connector     port="80" maxThreads="150" minSpareThreads="25" maxSpareThreads="75"     enableLooku[ps教程](http://www.111cn.net/fw/photo.html)="false" redirectPort="8443" acceptCount="100"     debug="0" connectionTimeout="20000"     disableUploadTimeout="true"     URIEncoding="GB2312"    />  5.Tomcat的conf文件夹下的web.xml文件的<servlet></servlet>标签之间添加如下代码  <init-param>    <param-name>encoding</param-name>    <param-value>GB2312</param-value>  </init-param>  6.使用Servlet规范中的过虑器指定编码  <1>.新创建一个servlet时，Interfaces接口：继承javax.servlet.Filter        options中只需选中：create Inherited Methods; create Constructors;        建好后在其doFilter()方法中：  arg0.setCharacterEncoding("gb2312"); //实现请求乱码的处理  arg1.setCharacterEncoding("gb2312"); //实现响应乱码的处理  7.解决[数据库教程](http://www.111cn.net/database/database.html)的乱码  在jdbc.url中明确字符集 (不适用于sqlserver)  在连接数据库时：con=DriverManager.getConnection(url,"sa","");  url="jdbc:协议：子协议：//IP：端口; 库 ? useUnicode=true & characterEncoding=gb2312"; |
| 我遇到的乱码我的解决:  1. String encoding = System.getProperty("file.encoding");  // System.out.println("系统默认的字符集是:" + encoding);  String getCon= dn.get("content").toString();//获取请求的字符  // String content=new String(getCon.getBytes("ISO-8859-1"),encoding); |
| 2.使用过滤器:  a.过滤器的配置：  <filter>  <description>charset to utf8</description>  <display-name>charset to utf8</display-name>  <filter-name>CharSetE</filter-name>  <filter-class>com.hygj.filter.CharSetE</filter-class>  <init-param>  <description>encode to UTF-8</description><!-- //UTF-8表示中文编码 -->  <param-name>encoding</param-name>  <param-value>UTF-8</param-value>  </init-param>  </filter>  <filter-mapping>  <filter-name>CharSetE</filter-name>  <url-pattern>/\*</url-pattern>  </filter-mapping>  B》package com.hygj.filter;  import java.io.IOException;  import java.io.PrintWriter;  import javax.servlet.Filter;  import javax.servlet.FilterChain;  import javax.servlet.FilterConfig;  import javax.servlet.ServletException;  import javax.servlet.ServletRequest;  import javax.servlet.ServletResponse;  import javax.servlet.http.HttpServlet;  import javax.servlet.http.HttpServletRequest;  import javax.servlet.http.HttpServletResponse;  public class CharSetE implements Filter {    private FilterConfig filterConfig;  private String trargetEncoding=null;  public FilterConfig getFilterConfig() {  return filterConfig;  }  public void setFilterConfig(FilterConfig filterConfig) {  this.filterConfig = filterConfig;  }  public String getTrargetEncoding() {  return trargetEncoding;  }  public void setTrargetEncoding(String trargetEncoding) {  this.trargetEncoding = trargetEncoding;  }  @Override  public void destroy() {  this.filterConfig=null;  this.trargetEncoding=null;//清空资源  }  @Override  public void doFilter(ServletRequest request, ServletResponse response,  FilterChain chain) throws IOException, ServletException {  String encodes=System.getProperty("file.encoding");//获得系统编码  try {  System.out.println("目标编码： "+trargetEncoding);//在部署描述符中的编码//可以在部署描述符中修改：  request.setCharacterEncoding(encodes);//这里选择设置系统默认的编码  chain.doFilter(request, response);  } catch (Exception e) {  filterConfig.getServletContext().log(e.getMessage());  e.printStackTrace();  }      }  @Override  public void init(FilterConfig filterConfig) throws ServletException {  // TODO Auto-generated method stub  this.filterConfig=filterConfig;  this.trargetEncoding=this.filterConfig.getInitParameter("encoding");  }  } |

## String intern()方法

|  |
| --- |
| 1. new产生的string在堆中，而不在字符串池中，即CONSTANT\_String\_info中。  2. intern()去字符串池中查找是否存在该内容为指定字符串的string对象，如果有，则返回一个已有的引用，否则，创建在字符串池中创建新的引用，并返回该引用。  3. 采用new 创建的字符串对象不进入字符串池，字符串相加的时候，都是静态字符串的结果会添加到字符串池，如果其中含有变量不会进入字符串池中。但是字符串一旦进入字符串池中，就会先查找池中有无此对象。如果有此对象，则让对象引用指向此对象。如果无此对象，则先创建此对象，再让对象引用指向此对象。  注意理解字符串池和堆中字符串的区别。 |

## JVM发生OOM时，自动生成DUMP文件。

|  |
| --- |
| 1、配置方法  在JAVA\_OPTIONS变量中增加  -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=${目录}。// -XX:HeapDumpPath=E:\Java\dump  2、参数说明  （1）-XX:+HeapDumpOnOutOfMemoryError参数表示当JVM发生OOM时，自动生成DUMP文件。  （2）-XX:HeapDumpPath=${目录}参数表示生成DUMP文件的路径，也可以指定文件名称，例如：-XX:HeapDumpPath=${目录}/java\_heapdump.hprof。如果不指定文件名，默认为：java\_<pid>\_<date>\_<time>\_heapDump.hprof。 |

Java类型转换String→date

|  |
| --- |
| 用SimpleDateFormat来转换  SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");  Date date = sdf.parse("2008-08-08 12:10:12"); |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

# 技术必知点

## MIME笔记

|  |
| --- |
| MIME笔记  1.  MIME的全称是"Multipurpose Internet Mail Extensions"，中译为"多用途互联网邮件扩展"，指的是一系列的电子邮件技术规范，主要包括RFC 2045、RFC 2046、RFC 2047、RFC 4288、RFC 4289和RFC 2077。  顾名思义，MIME是对传统电子邮件的一个扩展，现在已经成为电子邮件实际上的标准。  2.  传统的电子邮件是1982年定下技术规范的，文件是RFC 822。  它的一个重要特点，就是规定电子邮件只能使用ASCII字符。这导致了三个结果：1）非英语字符都不能在电子邮件中使用；2）电子邮件中不能插入二进制文件（如图片）；3）电子邮件不能有附件。  这实际上无法接受的，因此到了1992年，工程师们决定扩展电子邮件的技术规范，提出一系列补充规范，这就是MIME的由来。  3.  下面是一封传统的电子邮件。  From: "Tommy Lee" <lee@example.com>  To: "Jack Zhang" <zhang@example.com>  Subject: Test  Date: Wed, 17 May 2000 19:08:29 -0400  Message-ID: <NDBBIAKOPKHFGPLCODIGIEKBCHAA.lee@example.com>  Hello World.  从上面可以看出，这封信的发信人地址是lee@example.com，收信人地址是zhang@example.com，邮件主题是Test，发送时间是2000年5月17日，邮件内容是"Hello World."。  在结构上，这封信分为三个部分：首先是信件头，然后是一个空行，最后是信件内容。收信人的客户端软件只会显示最后一部分，要查看全信，必须使用"查看原始邮件"功能。  4.  MIME对传统电子邮件的扩展，表现在它在信件头部分添加了几条语句，主要有三条。  第一条是：  MIME-Version: 1.0  这条语句是必须的，而且1.0这个版本值是不变的，即使MIME本身已经升级了好几次。  有了这条语句，收信端就知道这封信使用了MIME规范。  5.  第二条语句是：  Content-Type: text/plain; charset="ISO-8859-1"  这一行是极端重要的，它表明传递的信息类型和采用的编码。  Content-Type表明信息类型，缺省值为" text/plain"。它包含了主要类型（primary type）和次要类型（subtype）两个部分，两者之间用"/"分割。主要类型有9种，分别是application、audio、example、image、message、model、multipart、text、video。  每一种主要类型下面又有许多种次要类型，常见的有：  text/plain：纯文本，文件扩展名.txt  text/html：HTML文本，文件扩展名.htm和.html  image/jpeg：jpeg格式的图片，文件扩展名.jpg  image/gif：GIF格式的图片，文件扩展名.gif  audio/x-wave：WAVE格式的音频，文件扩展名.wav  audio/mpeg：MP3格式的音频，文件扩展名.mp3  video/mpeg：MPEG格式的视频，文件扩展名.mpg  application/zip：PK-ZIP格式的压缩文件，文件扩展名.zip  详细的Content-Type列表，可以查看这里和这里。  如果信息的主要类型是"text"，那么还必须指明编码类型"charset"，缺省值是ASCII，其他可能值有"ISO-8859-1"、"UTF-8"、"GB2312"等等。  整个Content-Type这一行，不仅使用在电子邮件，后来也被移植到了HTTP协议中，所以现在只要是在网上传播的HTTP信息，都带有Content-Type头，以表明信息类型。  6.  前面已经说过，电子邮件的传统格式不支持非ASCII编码和二进制数据。因此MIME规定了第三条语句：  Content-transfer-encoding: base64  这条语句指明了编码转换的方式。Content-transfer-encoding的值有5种----"7bit"、"8bit"、"binary"、"quoted-printable"和"base64"----其中"7bit"是缺省值，即不用转化的ASCII字符。真正常用是"quoted-printable"和"base64"两种，它们的详细用法，我在明天的笔记中会详细介绍。  7.  下面是一封我收到的邮件的源码：  Date: Wed, 18 Jun 2008 18:07:51 +0800 (CST)  From: xxx <xxx@163.com>  To: yifeng.ruan@gmail.com  Message-ID: <14410503.1073611213783671983.JavaMail.coremail@bj163app54.163.com>  Subject: =?gbk?B?xOO6ww==?=  MIME-Version: 1.0  Content-Type: multipart/alternative;  boundary=&quot;----=\_Part\_287491\_22998031.1213783671982&quot;  ------=\_Part\_287491\_22998031.1213783671982  Content-Type: text/plain; charset=gbk  Content-Transfer-Encoding: base64  IAq4+b7dsr+209PQudi55raoo6yyu7XD1Nq12Le9yM66zs341b7Jz7nSz+DTprXEtqvO96Osx+vE  49TaxOO1xLKpv83W0AogIArW0Ln6yr2x6tPvIC0gyO7Su7fltcTN+MLnyNXWvgoKtcS12jEy1cXN  vMasyb6z/aOst/HU8s7Sw8fXt76/xOO1xM/gudjU8MjOoaPQu9C7us/X96OhtMvNvMas1Nq4vbz+  wO/D5g==  ------=\_Part\_287491\_22998031.1213783671982  Content-Type: text/html; charset=gbk  Content-Transfer-Encoding: quoted-printable  <DIV>&amp;nbsp;</DIV>  <DIV>=B8=F9=BE=DD=B2=BF=B6=D3=D3=D0=B9=D8=B9=E6=B6=A8=A3=AC=B2=BB=B5=C3=D4=  =DA=B5=D8=B7=BD=C8=CE=BA=CE=CD=F8=D5=BE=C9=CF=B9=D2=CF=E0=D3=A6=B5=C4=B6=AB=  =CE=F7=A3=AC=C7=EB=C4=E3=D4=DA=C4=E3=B5=C4=B2=A9=BF=CD=D6=D0</DIV>  <DIV>&amp;nbsp;  ......  可以看到这封信的MIME语句是：  MIME-Version: 1.0  Content-Type: multipart/alternative;  boundary="----=\_Part\_287491\_22998031.1213783671982"  "Content-Type: multipart/alternative;"表明这封信的内容，是纯文本和HTML文本的混合。另两个可能的值是multipart/mixed和multipart/related，分别表示"信件内容中有二进制内容"和"信件带有附件"。  "boundary="----=\_Part\_287491\_22998031.1213783671982"  "表明不同信件内容的分割线是"----=\_Part\_287491\_22998031.1213783671982"，它通常是一个很长的随机字符串。  信件内容部分又有两个子信件头：  Content-Type: text/plain; charset=gbk  Content-Transfer-Encoding: base64  和  Content-Type: text/html; charset=gbk  Content-Transfer-Encoding: quoted-printable  它们表明，第一个部分是gbk编码的纯文本，编码转换格式是base64。第二个部分是gbk编码的HTML文本，编码转化格式是quoted-printable。 |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

# 数据库

## Oracle

### 查看oracel是32位还是64位的

|  |
| --- |
| 方法一：使用sqlplus 64位：  [oracle@qs-wg-db2 ~]$ sqlplus / as sysdba;    SQL\*Plus: Release 10.2.0.5.0 - Productionon Sun Sep 25 08:57:22 2011    Copyright (c) 1982, 2010, Oracle.  All Rights Reserved.      Connected to:  Oracle Database 10g Enterprise EditionRelease 10.2.0.5.0 - 64bit Production  With the Partitioning, OLAP, Data Miningand Real Application Testing options    SQL>    如果是64位，用sqlplus 连上之后会显示具体的位数信息，32位则不会显示。      32位：  C:\Users\Administrator.DavidDai>sqlplus/ as sysdba;    SQL\*Plus: Release 11.2.0.1.0 Production onSun Sep 25 08:55:48 2011    Copyright (c) 1982, 2010, Oracle.  All rights reserved.      Connected to:  Oracle Database 11g Enterprise EditionRelease 11.2.0.1.0 - Production  With the Partitioning, OLAP, Data Miningand Real Application Testing options     方法二： 查看v$version 视图 32位：  SQL> select \* from v$version;    BANNER  --------------------------------------------------------------------------  Oracle Database 11g Enterprise EditionRelease 11.2.0.1.0 - Production  PL/SQL Release 11.2.0.1.0 - Production  CORE   11.2.0.1.0      Production  TNS for 32-bit Windows: Version 11.2.0.1.0- Production  NLSRTL Version 11.2.0.1.0 - Production    64位：  SQL> select \* from v$version;    BANNER  ----------------------------------------------------------------  Oracle Database 10g Enterprise EditionRelease 10.2.0.5.0 - 64bi  PL/SQL Release 10.2.0.5.0 - Production  CORE   10.2.0.5.0      Production  TNS for Linux: Version 10.2.0.5.0 -Production  NLSRTL Version 10.2.0.5.0 – Production    和第一种方法一样，64位会显示具体的位数信息，32位不会显示。   方法三：查看v$sql 视图   32位：输出为8位16进制数  SQL> select address from v$sql whererownum<2;    ADDRESS  --------  B50ACCAC    64位：输出为16位16进制数  SQL> select address from v$sql whererownum<2;    ADDRESS  ----------------  0000000196FDF7D8 |

### 关于navicat连接oracle 报 ORA-12737 set CHS16GBK错误的解决方案

|  |
| --- |
| 昨天下载安装了navicat 9.0 Premium（高级英文版），连接mysql正常，看到他同时支持oracle、sqlite和postgresql，也想来连接一下oracle，进行集中管理。 1、但是连接时候先是报出监听错误、不能解析oracle SID标识符，我用sqlplus和plsql连接却是正常的！！！这说明我的配置文件是没有问题的。 2、重新将连接方式由TNS改为BASIC，然后依次填写连接信息，最后却爆出了“ORA-12737: Instant Client Light: unsupported server character set CHS16GBK”的错误。 3、确定是字符集问题后，google上查到类似的解决方案：   ORA-12737: Instant Client Light: unsupported server character set CHS16GBK  Google，解释如下： ORA-12737: Instant Client Light: unsupported server character set string Cause:     The character set specified is not allowed for this operation or is invalid. Instant Client Light has only minimal character sets. Action:     Do not use Instant Client Light for this character set  4、原因明确了之后，就要怎么解决了。修改服务器的字符集不太现实。后来研究了OCI，知道Instance Client 10是不支持除英文以外的编码的（如果你下载的是中文版的navicat可能不会有这个问题）。  5、去oracle官网下载最新版的OCI：http://www.oracle.com/technetwork/topics/winsoft-085727.html 6、同时在Oracle官网上发现Instance Client 11要比10大很多，下载完成后将解压出来的核心的oci11的3个文件替换掉原先的oci10的3个文件即可。(记得做个备份) 替换方法：在Navicat里，工具->选项->Miscellaneous->OCI。为oci.dll重新指定位置，然后重启navicat就OK了。 重启Navicat最后证明Instance Client 11没有字符集问题。 |

### oracle建立用户与授权

|  |
| --- |
| **创建表空间及用户：**  **create tablespace 表空间名 datafile 'd:/seal.dbf' size 10M autoextend on; create user username identified by password default tablespace 表空间名; grant connect,resource,dba to username;**    登录超级用户sqlplus sys@orcl as sysdba  #################################### # 建立用户和对用户的管理 ####################################  # 合法的Oracle标识符号       1、不能是保留关键字       2、必须使用1到30个字符。数据库的名称最多是8个字符。 3、必须以数据库字符集中的字母字符开头 4、只能够包含数据库字符集中的字母字符，以及以下字符：#、$、\_，另外数据库链接可以包含               @符号和'.'(句号) 5、不能包含引号  # 建立用户 create user angeos identified by angeos; 建立了用户：angeos，密码为：angeos  # 对用户授权 grant connect,resource to angeos; 对用户angeos授予了连接数据库和访问资源的权限  # 对用户授权 grant create session,dba to angeos; CREATE SESSION是一个系统特权，它可以为用户提供连接数据库的能力。 DBA是一个具有超过120个系统特权的角色，所以它可以让用户在数据库中完成几乎任何工作。  # 改变用户的密码 alter user angeos identified by oracle; 将用户angeos的密码改变为：oracle  # 锁定帐号以及解锁 alter user oe account unlock; 然后用用户oe登录数据库服务器，密码为oe 注意：用户解锁后，要重启服务。  # 修改表空间的设置 默认情况下，它会使用表空间SYSTEM和TEMP(用于存放临时数据)。 不推荐采用这种方法。所以我们需要改变表空间。  通过系统用户连接数据库服务器 conn [sys/sysadmin@hostname\_services](mailto:sys/sysadmin@hostname_services) as sysdba; 然后查看表空间 select tablespace\_name,contents from dba\_tablespaces order by tablespace\_name; 使用USER表空间代替SYSTEM表空间 alter user angeos default tablespace users temporary tablespace temp;  尝试建立一张表 create table table1 (      fieldA varchar2(10) )  # 删除用户angeos drop user angeos; 由于用户angeos有一张表table1,所以删除用户时，我们需要指定关键字CASCADE drop user angeos cascade;  # 解释术语模式Schema 数据库的模式定义为数据库对象的集合，而模式的名称就是拥有或者控制这个数据库对象集合的用户名称。 所有数据库对象，例如表、视图、索引、触发器、Java存储过程、PL/SQL程序包，函数等，都是由Oracle 数据库中的一个用户所拥有。甚至Oracle的数据词典，系统编目也是名为SYS的模式的一部分。  用户的传统解释就是能够唯一标识一组信任凭证的名称和密码组合。  # 解释系统特权 在Oracle数据库中有两类特权： （1）对象级别特权是由用户赋予的访问或者操作数据库对象的特权。 （2）系统特权不是控制对指定数据库对象的访问，而是用来许可对各种特性的访问，或者许可Oracle数据库中的          特定任务。       查询特权      connect system/sysadmin      desc dba\_sys\_privs;      查询             select distinct privilege from dba\_sys\_privs order by privilege       赋予系统特权的语句      GRANT system\_priviege to username [with admin option];      撤销系统特权的语句      REVOKE system\_priviege from username;       注意：在任何用户能够连接到Oracle数据库之前，都需要向它们赋予CREATE SESSION特权，为他们提供连接许可。  ################################## # 建立数据表 ##################################  # 语法规则 CREATE TABLE [SCHEMA.] <tablename>(      <coloumn\_name> <datatype> [default <expression>] [<constraint>]      [,<coloumn\_name> <datatype> [default <expression>] [<constraint>]]      [,...] );  # 建表的例子 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CREATE TABLE CD\_COLLECTION (      ALBUM\_TITLE     VARCHAR2(100),      ARTIST      VARCHAR2(100),      COUNTRY      VARCHAR2(25),      RELEASE\_DATE     DATE,      LABEL      VARCHAR2(25),      PRIMARY KEY (ALBUM\_TITLE, ARTIST) ); \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CREATE TABLE seagal.SONGS (      SONG\_TITLE      VARCHAR2(100),      COMPOSER     VARCHAR2(100),      LENGTH      NUMBER,      TRACK      NUMBER,      ARTIST      VARCHAR2(100),      ALBUM\_TITLE     VARCHAR2(100),      FOREIGN KEY (ARTIST, ALBUM\_TITLE) REFERENCES seagal.CD\_COLLECTION(ARTIST,         ALBUM\_TITLE),      PRIMARY KEY (SONG\_TITLE, ARTIST, ALBUM\_TITLE) );  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* select \* from seagal.cd\_collection; select \* from seagal.songs; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* INSERT INTO seagal.CD\_COLLECTION   VALUES('Black Sheets of Rain', 'Bob Mould', 'USA',to\_date('01-01-92','DD-MM-YY'), 'Virgin');  INSERT INTO seagal.CD\_COLLECTION  VALUES('Candy Apple Grey', 'Husker Du', 'USA',to\_date('01/01/86','DD/MM/YY'), 'Warner Brothers');  INSERT INTO seagal.SONGS  VALUES('Black Sheets of Rain','Mould', NULL,1, 'Bob Mould', 'Black Sheets of Rain');  INSERT INTO seagal.SONGS  VALUES('Crystal','Mould', 3.28, 1, 'Husker Du', 'Candy Apple Grey');  INSERT INTO seagal.SONGS  VALUES('Don''t want to know if you are lonely ','Hart', 3.28, 2, 'Husker Du', 'Candy Apple Grey');  INSERT INTO seagal.SONGS  VALUES('I don''t know for sure','Mould', 3.28, 3, 'Husker Du', 'Candy Apple Grey');  INSERT INTO SONGS VALUES('Black Sheets of Rain','Mould', NULL,1, 'Bob Mould', 'Black Sheets of Rain');  COMMIT; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* DROP TABLE FRUITS;  CREATE TABLE FRUITS ( FRUIT VARCHAR2(12), COLOR VARCHAR2(12), QUANTITY NUMBER, PRICE NUMBER, PICKED DATE );  INSERT INTO FRUITS VALUES('Apple', 'Green', 12, 0.5, '12-Sep-2002'); INSERT INTO FRUITS VALUES('Apple', 'Red', 12, 0.5, '15-Sep-2002'); INSERT INTO FRUITS VALUES('Mango', 'Yellow', 10, 1.5, '22-Sep-2002'); INSERT INTO FRUITS VALUES('Mangosteen', 'Purple', 5, 2, '25-Sep-2002'); INSERT INTO FRUITS VALUES('Durian', 'NULL', 2, 15, NULL); INSERT INTO FRUITS VALUES('Orange', 'Orange', 10, 1.5, '28-Aug-2002');  COMMIT; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* COMMIT;  # CREATE TABLE AS SELECT的语句 通过查询一个表，并且将查询结果集物化到一个常规表中来建立表。 复制表的结构，但是约束、索引和触发器等对象不会被放入新的表中。  例子： create table emp\_copy as select \* from scott.emp;  # 数据词典 每一个数据库都有一个数据词典，任何管理Oracle数据库，或者使用Oracle构建应用的用户都需       要使用数据词典。 数据词典是Oracle数据库的编目。 当建立用户、表、约束和其它数据库对象的时候，Oracle都会自动维护一个在数据库中存储的项        目编目。  例如：USER\_TABLES视图可以展示当前用户所拥有的所有表的信息。      使用DESCRIBE命令查看USER\_TABLES视图的结构信息。  具有DBA权限的用户可以查看DBA\_TABLES select owner ,table\_name,tablespace\_name from dba\_tables where owner in('SCOTT','HR') order by owner,tablesapce\_name,table\_name |

### Oracle创建表空间 创建用户 授权

|  |
| --- |
| 1.创建表空间 |
| create tablespace test datafile 'D:\cache\sqlcache.dbf' size 50M default storage (initial 500k next 500k minextents 1 maxextents unlimited   pctincrease 0); |
| 2.创建用户并设置默认的表空间 |
| create user liting identified by liting123 default tablespace test; |
| 3.授权给新建的用户 |
| grant resource,connect,dba to test; |
| 删除表空间 |
| drop tablespace "空间名" including contents and datafiles |
| 删除用户 |
| drop user "lxg" cascade  alter tablespace chinawater add datafile 'c:\oracle\oradata\orcl9\ADDCHINAWATER.dbf' size 200M  创建用户  1. create user userName identified by password;  创建用户 userName,密码为 password  2  给用户授权  1. grant dba to lxg;--授予DBA权限  2. grant unlimited tablespace to lxg;--授予不限制的表空间  3. grant select any table to lxg;--授予查询任何表  4. grant select any dictionary to lxg;--授予 查询 任何字典  1. grant dba to lxg;  2. grant unlimited tablespace to lxg;  3. grant select any table to lxg;  4. grant select any dictionary to lxg; |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

## Mysql

### Mysql之SQL语句优化:

### 查看数据库的编码:

|  |
| --- |
| 查看Mysql的编码：\s 要保持server characterset ，db characterset,client characterset ,conn characterset保持一致：最好四个都是utf-8的  :查看编码：mysql> show variables like 'character%'; |

### 查看数据库详细信息找到占用时间长的Show full processlist

|  |
| --- |
| 同样的查询语句之后加入\G也可以看到格式化之后的 |

### 绿色版mysql注册卸载服务方法

|  |
| --- |
| 如果直接用绿色版的mysql，则下载后解压，只需对目录下的my.ini文件的basedir（mysql的基本目录）和datadir（mysql数据目录）指定就可以，如下所示。  #Path to installation directory. All paths are usually resolved relative to this. basedir="D:\MySQL Server 5.1/" #Path to the database root datadir="D:\MySQL Server 5.1/Data/"  但是因为免安装，所以不会生成服务，只需通过简单的命令就可以实现：  打开命令窗口切换到mysql的bin目录下。  1.注册服务：  D:\MySQL Server 5.1\bin>mysqld --install mysql51  回车执行，会提示创建成功，打开服务可以找到刚创建的mysql51服务  http://files.jb51.net/file_images/article/201306/2013630152256089.png  2.卸载服务：  D:\MySQL Server 5.1\bin>mysqld --remove mysql51 |

### [mysql5.6.16绿色版配置、运行](http://blog.csdn.net/zhuxiaowei716/article/details/19405955)：包括修改root用户的密码

|  |
| --- |
| 1、从该地址http://dev.mysql.com/downloads/mysql/中选择windows的版本，选择下载。  2、将下载的压缩包解压。  3、将根目录下的my-default.ini复制重命名为my.ini。  4、打开my.ini文件，将下面的源码添加修改到my.ini中  [client] default-character-set=utf8  [mysqld]  # 设置mysql服务器编码 character\_set\_server=utf8  #basedir设置mysql的安装目录，如果目录中带空格则必须加引号  #datadir设置mysql数据库的数据的存放目录，文件夹名必须是data basedir=D:\Tool\mysql-5.6.16-win32     datadir=D:\Tool\mysql-5.6.16-win32\data  完成后保存。  5、安装mysql服务 从MS-DOS窗口进入目录E:\myserver\mysql-5.0.37-win32\bin，运行如下命令： mysqld --install mysql --defaults-file= “D:\Tool\mysql-5.6.16-win32\my.ini”  当显示Service successfully installed.时即表示mysql注册成功。  6、启动mysql数据库服务 还在上面的命令窗口里面，输入命令：net start mysql 这样就启动了mysql服务。  7、停止mysql数据库服务  还在上面的命令窗口里面，输入命令：net stop mysql 这样就停止了mysql服务。  8、删除mysql数据库服务  还在上面的命令窗口里面，输入命令：mysqld remove mysql 这样就删除了mysql服务。  9、修改mysql数据库的root密码（初始为空）  方法1： 用SET PASSWORD命令  　　mysql -u root  　　mysql> SET PASSWORD FOR ['root'@'localhost'](mailto:'root'@'localhost') = PASSWORD('newpass');  方法2：用mysqladmin  　　mysqladmin -u root password "newpass"  　　如果root已经设置过密码，采用如下方法  　　mysqladmin -u root password oldpass "newpass"  方法3： 用UPDATE直接编辑user表  　　mysql -u root  　　mysql> use mysql;  　　mysql> UPDATE user SET Password = PASSWORD('newpass') WHERE user = 'root';  　　mysql> FLUSH PRIVILEGES;  在丢失root密码的时候，可以这样  　　mysqld\_safe --skip-grant-tables&  　　mysql -u root mysql  　　mysql> UPDATE user SET password=PASSWORD("new password") WHERE user='root';  　　mysql> FLUSH PRIVILEGES;  10、设置root权限  GRANT ALL PRIVILEGES ON \*.\* TO ['root'@'%'](mailto:'root'@'%25') IDENTIFIED BY 'yourpassword'; |

### MySQL创建用户与授权方法

|  |
| --- |
| 一, 创建用户:  命令:CREATE USER 'username'@'host' IDENTIFIED BY 'password';  说明:username - 你将创建的用户名, host - 指定该用户在哪个主机上可以登陆,如果是本地用户可用localhost, 如果想让该用户可以从任意远程主机登陆,可以使用通配符%. password - 该用户的登陆密码,密码可以为空,如果为空则该用户可以不需要密码登陆服务器.  例子: CREATE USER 'dog'@'localhost' IDENTIFIED BY '123456';  CREATE USER 'pig'@'192.168.1.101\_' IDENDIFIED BY '123456';  CREATE USER 'pig'@'%' IDENTIFIED BY '123456';  CREATE USER 'pig'@'%' IDENTIFIED BY '';  CREATE USER 'pig'@'%';  二,授权:  命令:GRANT privileges ON databasename.tablename TO 'username'@'host'  说明: privileges - 用户的操作权限,如SELECT , INSERT , UPDATE 等(详细列表见该文最后面).如果要授予所的权限则使用ALL.;databasename - 数据库名,tablename-表名,如果要授予该用户对所有数据库和表的相应操作权限则可用\*表示, 如\*.\*.  例子: GRANT SELECT, INSERT ON test.user TO 'pig'@'%';  GRANT ALL ON \*.\* TO 'pig'@'%';  注意:用以上命令授权的用户不能给其它用户授权,如果想让该用户可以授权,用以下命令:  GRANT privileges ON databasename.tablename TO 'username'@'host' WITH GRANT OPTION;  三.设置与更改用户密码  命令:SET PASSWORD FOR 'username'@'host' = PASSWORD('newpassword');如果是当前登陆用户用SET PASSWORD = PASSWORD("newpassword");  例子: SET PASSWORD FOR 'pig'@'%' = PASSWORD("123456");  四.撤销用户权限  命令: REVOKE privilege ON databasename.tablename FROM 'username'@'host';  说明: privilege, databasename, tablename - 同授权部分.  例子: REVOKE SELECT ON \*.\* FROM 'pig'@'%';  注意: 假如你在给用户'pig'@'%'授权的时候是这样的(或类似的):GRANT SELECT ON test.user TO 'pig'@'%', 则在使用REVOKE SELECT ON \*.\* FROM 'pig'@'%';命令并不能撤销该用户对test数据库中user表的SELECT 操作.相反,如果授权使用的是GRANT SELECT ON \*.\* TO 'pig'@'%';则REVOKE SELECT ON test.user FROM 'pig'@'%';命令也不能撤销该用户对test数据库中user表的Select 权限.  具体信息可以用命令SHOW GRANTS FOR 'pig'@'%'; 查看.  五.删除用户  命令: DROP USER 'username'@'host';  附表:在MySQL中的操作权限  ALTER Allows use of ALTER TABLE.  ALTER ROUTINE Alters or drops stored routines.  CREATE Allows use of CREATE TABLE.  CREATE ROUTINE Creates stored routines.  CREATE TEMPORARY TABLE Allows use of CREATE TEMPORARY TABLE.  CREATE USER Allows use of CREATE USER, DROP USER, RENAME USER, and REVOKE ALL PRIVILEGES.  CREATE VIEW Allows use of CREATE VIEW.  DELETE Allows use of DELETE.  DROP Allows use of DROP TABLE.  EXECUTE Allows the user to run stored routines.  FILE Allows use of SELECT... INTO OUTFILE and LOAD DATA INFILE.  INDEX Allows use of CREATE INDEX and DROP INDEX.  INSERT Allows use of INSERT.  LOCK TABLES Allows use of LOCK TABLES on tables for which the user also has SELECT privileges.  PROCESS Allows use of SHOW FULL PROCESSLIST.  RELOAD Allows use of FLUSH.  REPLICATION Allows the user to ask where slave or master  CLIENT servers are.  REPLICATION SLAVE Needed for replication slaves.  SELECT Allows use of SELECT.  SHOW DATABASES Allows use of SHOW DATABASES.  SHOW VIEW Allows use of SHOW CREATE VIEW.  SHUTDOWN Allows use of mysqladmin shutdown.  SUPER Allows use of CHANGE MASTER, KILL, PURGE MASTER LOGS, and SET GLOBAL SQL statements. Allows mysqladmin debug command. Allows one extra connection to be made if maximum connections are reached.  UPDATE Allows use of UPDATE.  USAGE Allows connection without any specific privileges.  详细出处参考：http://www.jb51.net/article/31850.htm |

|  |
| --- |
|  |

## SQLSERVE问题总结：

### 总结JDBC连接SQLServer的错误Error establishing socket

|  |
| --- |
| 在用MSSQL连接数据库的过程中，经常会遇见“Error establishing socket.”的错误，大概得错误日志信息如下：  Sql代码   1. java.sql.SQLException : [Microsoft][SQLServer 2000 Driver **for** JDBC]Error establishing socket. 2. **at** com.microsoft.jdbc.base.BaseExceptions.createException(Unknown Source) 3. **at** com.microsoft.jdbc.base.BaseExceptions.getException(Unknown Source) 4. **at** com.microsoft.jdbc.base.BaseExceptions.getException(Unknown Source) 5. **at** com.microsoft.jdbc.sqlserver.tds.TDSConnection.<init>(Unknown Source) 6. **at** com.microsoft.jdbc.sqlserver.SQLServerImplConnection.**open**(Unknown Source) 7. **at** com.microsoft.jdbc.base.BaseConnection.getNewImplConnection(Unknown Source) 8. **at** com.microsoft.jdbc.base.BaseConnection.**open**(Unknown Source) 9. **at** com.microsoft.jdbc.base.BaseDriver.**connect**(Unknown Source) 10. **at** java.sql.DriverManager.getConnection( DriverManager.java:523 ) 11. **at** java.sql.DriverManager.getConnection( DriverManager.java:171 ) 12. **at** test.SqlServerTest.main( SqlServerTest.java:29 )   一　配置错误  jdbc配置语句为: jdbc:microsoft:sqlserver://server\_name:1433  其中的sqlserver、server\_name和端口号都要看仔细了。  二　1433端口问题  可以通过命令“netstat -an”，来查看是否开了１４３３端口，如果列表中没有，要增开端口。  要保持jdbc配置语句（ jdbc:microsoft:sqlserver://server\_name:1433）中的端口与数据库传输协议默认的端口相同。  MSSQL设置端口号步骤如下：  1． 打开企业管理器，依次在控制台根目录 ->Microsoft SQL Servers->SQL Server 组，列出一部分数据库服务器。  2． 右击我们要连接的数据库服务器，选择属性，在常规选项卡点击网络配置弹出新窗口。  3． 在启用的协议区域我们选择 TCP/IP ，点击属性按钮，弹出新窗口。在这个窗口有网络协议默认值设置，一般默认端口是：1433。 如果你在ｕｒｌ中用的端口号与这个不同，就会出现错误。应保持二者相同。  三 没打补丁  可以用如下的方式查看MSSQL的版本：  打开企业管理器-〉工具-〉SQL查询分析器-〉帮助-〉关于  查看MSSQL的详细版本号  如果 是 8.00.194 就是还没打补丁  8.00.760 就是SP3  8.00.2039 就是SP4   1、下载MSSQL　2000 Service Pack 3a并安装，SQL请选用混和安装模式！！！  http://www.microsoft.com/downloads/details.aspx?FamilyId=90DCD52C-0488-4E46-AFBF-ACACE5369FA3&displaylang=zh-cn  解压缩sp3的升级包后,还要点击setup安装.才能成功升级到sp3。  2、下载SQL Server 2000 Driver for JDBC Service Pack 3  http://www.microsoft.com/downloads/details.aspx?FamilyId=07287B11-0502-461A-B138-2AA54BFDC03A&displaylang=en  四　系统防火墙或者杀毒软件  如果系统开防火墙或者杀毒软件，在有些情况下也可以出现类似的问题。  没有遇见过，有人这么说。 |

|  |
| --- |
|  |

## SQL

### SQL查询null和””的用法:

|  |
| --- |
| 空 如果是null 就是select name,age,成绩 from AA where 成绩 is  null 如果是"" 就是select name,age,成绩 from AA where 成绩=""  不空 如果是null 就是select name,age,成绩 from AA where 成绩 is not null 如果是"" 就是select name,age,成绩 from AA where 成绩<>"" |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

# Eclipse:使用总结：

## 修改窗口的背景色

|  |
| --- |
|  |

# Svn：

## 解决subclipse同步冲突问题A conflict in the working copy obstructs the current operation

|  |
| --- |
| 服务端版本控制软件subversion，客户端是eclipse的插件subclipse，个人感觉还是cvs比较简单方便。明明没什么冲突，当删除一个东西的时候老是提示错误，说冲突  A conflict in the working copy obstructs the current operation org.tigris.subversion.javahl.ClientException: A conflict in the working copy obstructs the current operation svn: Commit failed (details follow): svn: Aborting commit: 'D:eclipse-java-ganymede-SR1-win32workspacemyProjectdatabasesrc' remains in conflict  org.tigris.subversion.javahl.ClientException: A conflict in the working copy obstructs the current operation svn: Commit failed (details follow): svn: Aborting commit: 'D:eclipse-java-ganymede-SR1-win32workspacemyProjectdatabasesrc' remains in conflict  解决办法  右击你要删除的目录-->team-->Show Tree Conflict-->在Eclpise中会出现一个Tab视图显示Tree Conflict的内容，一个一个选择列表中的冲突右击将其标志为已解决，然后回到项目再提交更改就可以了。  注意：我用的版本subclipse是<http://subclipse.tigris.org/update_1.6.x>  这样的冲突会照成无法更新~~~更新的时候会跳过这个冲突的文件~~~提交的时候提交不了~~这个时候应该还原一下 就可以解决了 ~~所谓的还原就是还原服务器上的上一个版本~~~ |

|  |
| --- |
|  |

|  |
| --- |
|  |

# 办公技巧

## word

### Word自动生成页码:

|  |
| --- |
| 1. 在开始的时候定好样式: 3. 编写文档: 5. 这样目录就生产了(当编辑完成之后可以选择更新目录) |

|  |
| --- |
|  |

## Excel

|  |
| --- |
|  |

|  |
| --- |
|  |

## Powerpoint

|  |
| --- |
|  |

|  |
| --- |
|  |

# SSH

## Struts

|  |
| --- |
|  |

|  |
| --- |
|  |

## spring

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

## hibernate

id to load is required for loading

|  |
| --- |
| 错误原因应该是你的hibernate的配置文件把有些字段设置为not null但是你实际的对象中存在null值，所以就会出现这个错误 |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |