**编码与java URLEncoder/URLDecoder**

目录

**[1 基本概念 1](#_Toc10193)**

[1.1 字符(Character) 1](#_Toc23300)

[1.2 编码字符集(Coded Character Set) 1](#_Toc13538)

[1.3 字符编码表(Character Encoding Form) 1](#_Toc32210)

[1.4 码点(Code Point) 2](#_Toc19607)

[1.5 代码单元(Code Unit) 2](#_Toc7164)

[1.6 字节序(Byte Order Mark) 2](#_Toc4444)

**[2 字母编码介绍 2](#_Toc641)**

[2.1 asscii 2](#_Toc1906)

[2.2 ISO-8859/Latin 2](#_Toc19183)

**[3 象形文字编码介绍 2](#_Toc14138)**

[3.1 gb2312 2](#_Toc1284)

[3.2 gbk 3](#_Toc32464)

[3.3 gb18030 3](#_Toc22113)

**[4 unicode介绍 3](#_Toc30077)**

[4.1 unicode 3](#_Toc2219)

[4.2 utf-8,utf-16,utf-32 3](#_Toc6095)

**[5 java URLEncoder/URLDecoder 4](#_Toc457)**

[5.1 Java string 4](#_Toc26966)

[5.2 URLEncoder 4](#_Toc16175)

**[6 群里疑问 5](#_Toc24915)**

[6.1 编码一次问题 5](#_Toc29774)

[6.2 编码两次问题 5](#_Toc25054)

**[7 参考资料 5](#_Toc11672)**

# 基本概念

## 字符(Character)

* 在计算机中，一个字符是一个单位的字形/类字形单位或符号的基本信息
* 单字节编码: 使用1byte表示1个字符，典型的有ascii/eascii
* 双字节编码: 使用2byte表示1个字符，典型的有gb2312/gbk/big5
* 变长多字节编码：1个字符根据实际情况来决定有几个字节表示

## 编码字符集(Coded Character Set)

* 给字符表里面的字符编上一个数字
* Unicode 字符集就是编码字符集

## 字符编码表(Character Encoding Form)

* 将编码字符集里面字符对应的数字根据一定算法转换成二进制
* 以便于存储/传输
* Utf-8,utf-16 就是字符编码表

## 码点(Code Point)

* 字符编码表里面字符对应的数字

## 代码单元(Code Unit)

* 已编码文本中具有最短的比特组合单元
* 比如utf-8的代码单元就是8比特，1字节
* utf-16 代码单元就是16比特，2字节

## 字节序(Byte Order Mark)

* 表示字节的顺序
* 大端字节: 高位字节在前，低位字节在后
* 小端字节: 低位字节在前，高位字节在后

# 字母编码介绍

## asscii

* 全称(American Standard Code Information Interchanger)美国信息交互标准码
* 美国的英语是基于字母的，127就编的够用了
* 一个字节就够编码了，并且还有1bit没用上

## ISO-8859/Latin

* 欧洲那边基于字母的国家，有更多的字母需求，比如à
* asscii正好有1bit没用，他们就把这1bit用上，就是ISO-8859编码系列标准
* 定义了什么段大概用来做什么，至于里面的具体实现，就交由各自字符集决定
* 所以就会有 ISO-8859-1 至 ISO-8859-16 这么多字符集
* ISO-8859 系列标准，也是单字节编码，完全兼容asscii

# 象形文字编码介绍

## gb2312

* 中国国家标准总局发布的信息交换用汉字编码字符集，gb就代表国标
* 中文采用的是象形文字，远比字母多，1个字节无法表示
* 所有中文采用双字节编码
* 由于为了兼容asscii，asscii的编码保持不变
* 规定每个字节最高位为1，则表示为汉字
* 所以gb2312能表示的字符数量为 127\*127
* 第一阶段收录了6K多个，马上发现就不够用，继续把未用上的补上

## gbk

* 但是汉字实在太多了，gb2312的编码放不下了
* gbk标准来了，gbk标准在gb2312的基础上规定
* 如果一个字节最高为1，就表示一个汉字的开始
* 那gbk就能表示的字符数量为127\*256

## gb18030

* gbk只够汉语的编码
* 经济发展很多少数名族也有电脑了
* 这个时候只有再扩展了gb18030就产生了
* 使用的变长多字节编码
* 每个字符可以由1个、2个、或4个字节组成
* 完全兼容gb2312，基本兼容gbk，可以表示的字符数量多达100多万个

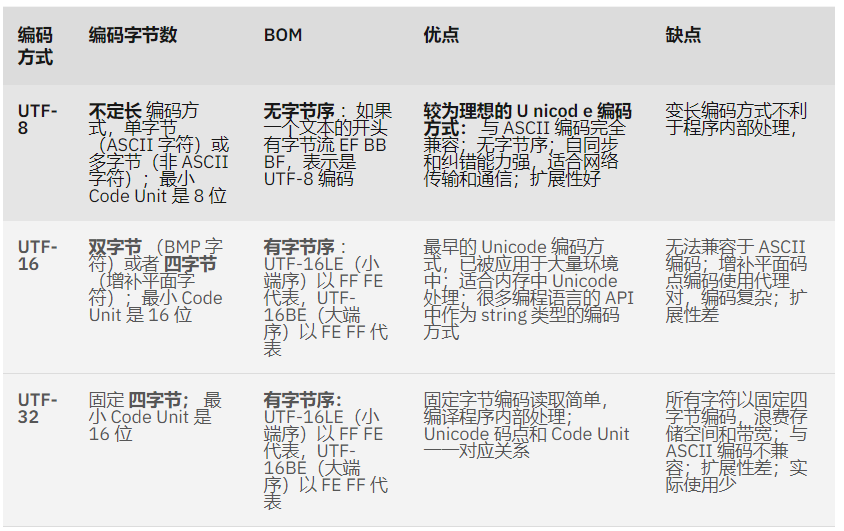
# unicode介绍

## unicode

* 全称(Universal Character Set) ucs、万国码、通用字符集、、
* 我们今天只介绍了英文、中文的编码，就有N种编码字符集了
* 世界上还有其他国家，也有自己的编码字符集
* 每个国家都有自己的标准，互相都不支持对方的编码
* 这个时候unicode编码字符集就出现了
* unicode只管给字符编码，单纯的只是一个编码字符集

## utf-8,utf-16,utf-32

* 就是分别的字符编码表的实现
* 最小使用两个字节表示一个字符称为utf-16
* 两个字节就存在字节序的问题
* 采用大端序号的为 UTF-16BE，java的string就是这种编码
* 采用小端序号的为 UTF-16LE
* 编码方式优缺点



# java URLEncoder/URLDecoder

## Java string

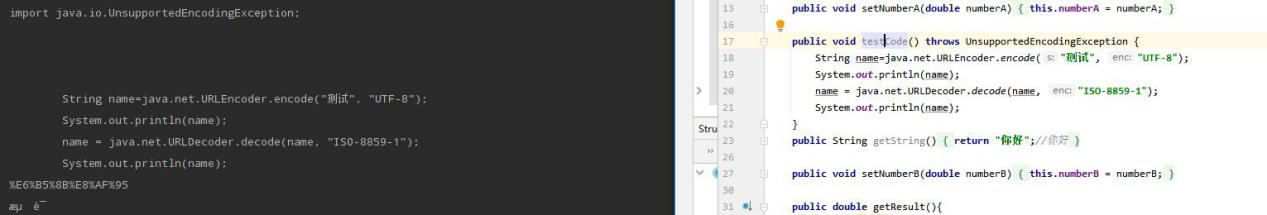
* java所有的字符串，在内存里面都是utf-16的表现形式
* 所以1个char占用2个字节
* 所有的字符串都需要使用原编码格式解码，再转成utf-16格式
* 一个char表示一个代码单元
* 基本平面的字符只需要一个char表示就好
* 增补平面的字符需要两个char
* Chartacter.isSupplementaryCodePoint

## URLEncoder

* URLEncoder的编码规则为
* 将所有非ascii字符按照指定编码格式编码成16进制数据
* 然后在每个16进制数字前增加%

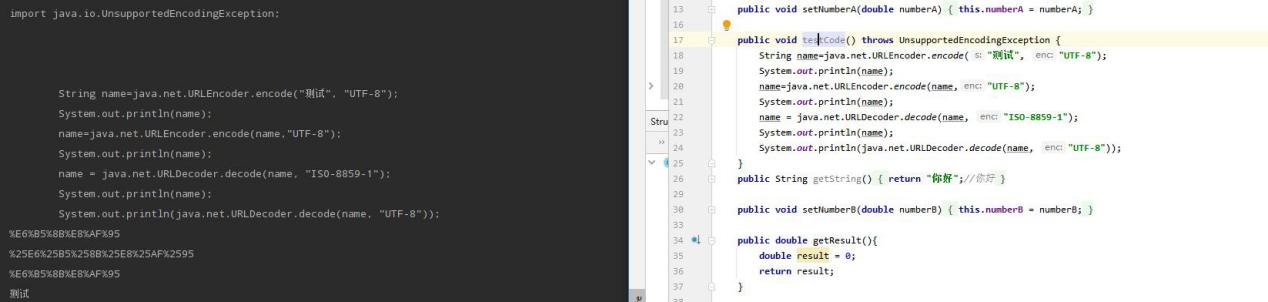
# 群里疑问

## 编码一次问题



* “测试”字符串先使用UTF-8进行编码
* 得到 %E6%B5%8B%E8%AF%95
* 使用ISO-8859-1进行解码
* 查找ISO-8859-1编码表，得出乱码字符串

## 编码两次问题



* 使用ISO-8859-1进行解码的时候，解的是%25
* %25是属于assci范围
* 由于ISO-8859-1/utf-8都兼容assci，所以可以正常解码

# 参考资料

[字符编码笔记：ASCII，Unicode 和 UTF-8](http://www.ruanyifeng.com/blog/2007/10/ascii_unicode_and_utf-8.html)

[字符编码查询](https://graphemica.com/)

[Java字符的编码解码与乱码问题](https://zhuanlan.zhihu.com/p/25435644)

[深入分析 Java 中的中文编码问题](https://developer.ibm.com/zh/articles/j-lo-chinesecoding/)

[Unicod及编码方式概述](https://developer.ibm.com/zh/articles/unicode-programming-language/ )

[刨根究底字符编码](https://www.cnblogs.com/benbenalin/category/1005679.html)

[UniCode编码表](https://www.cnblogs.com/csguo/p/7401874.html )

[深入分析 web 请求响应中的编码问题](https://developer.ibm.com/zh/articles/wa-lo-ecoding-response-problem/)

[理解字节序](http://www.ruanyifeng.com/blog/2016/11/byte-order.html)

[HTML ISO-8859-1 参考手册](https://www.runoob.com/tags/ref-entities.html)

java核心技术卷1(第8版) 第3.6章 字符串