**Base58**

**参考文章：**<https://blog.csdn.net/idwtwt/article/details/80740474>

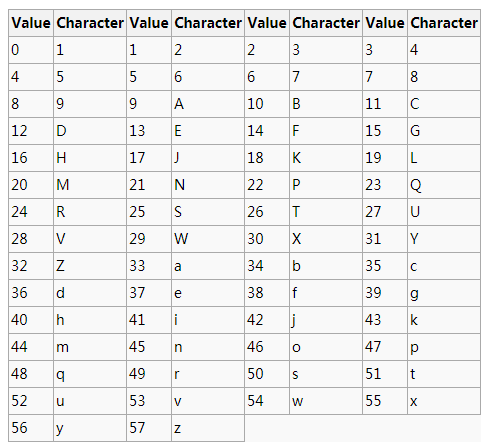
<https://blog.csdn.net/weixin_28717683/article/details/114881432>

https://the\_itach1.gitee.io/2021/01/18/%E4%B8%80%E4%BA%9B%E5%B8%B8%E7%94%A8%E7%AE%97%E6%B3%95/

简介：

Base58是用于Bitcoin中使用的一种独特的编码方式，主要用于产生Bitcoin的钱包地址。相比Base64，Base58不使用数字"0"，字母大写"O"，字母大写"I"，和字母小写"l"，以及"+“和”/"符号。  
设计Base58主要的目的是：  
 1.避免混淆。在某些字体下，数字0和字母大写O，以及字母大写I和字母小写l会非常相似。  
 2.不使用"+“和”/"的原因是非字母或数字的字符串作为帐号较难被接受。  
 3.没有标点符号，通常不会被从中间分行。  
 4.大部分的软件支持双击选择整个字符串。

而且因为58不是2的整次幂，所以没有使用类似base64编码中使用直接截取3个字符转4个字符（3\*8=4\*6，2的6次方刚好64）的方法进行转换，而是采用我们数学上经常使用的进制转换方法——辗转相除法（**本质上，base64编码是64进制，base58是58进制**）。编码表：



也就是字符1代表0，字符2代表1,字符3代表2...字符z代表57。然后回一下辗转相除法。  
如要将1234转换为58进制；  
第一步：1234除于58，商21，余数为16，查表得H  
第二步：21除于58，商0，余数为21，查表得N  
所以得到base58编码为：NH  
如果待转换的数前面有0怎么办？直接附加编码1来代表，有多少个就附加多少个（编码表中1代表0）。

**字符串编码流程:**

将字符串的每个字节换算成ASCII(0-255) (字符串实际上就是256进制的数字组合)

源字符串为：ABD

换算后： 65 66 68

将256进制的数字转换成10进制数字

256进制数：65 66 68

转成10进制：(65 \* 256 + 66) \* 256 + 68 = 4276804

将10进制数字转换成58进制数字（辗转相除法）

10进制数：4276804

58进制数： 21 53 20 0（注意：从后往前放，4276804 % 58 = 0，所以最后一位是0）

将58进制数字的每一位按照表格转换成对应的字符

58进制数：21 53 20 0

码表：123456789abcdefghijkmnopqrstuvwxyzABCDEFGHJKLMNPQRSTUVWXYZ

转换后的字符：nVm1

**字符串解码流程:**

字符串每一位按码表替换成数字 (58进制)

源字符： nVm1

58进制数：21 53 20 0

58进制转10进制

58进制数：21 53 20 0

10进制数：((21 \* 58 + 53) \* 58 + 20) \* 58 + 0 = 4276804

10进制转256进制（注意：辗转相除法，从后往前放，4276804 % 256 = 68，所以最后一个是68）

10进制数：4276804

256进制： 65 66 68

256进制转字符(按ascii转)

256进制： 65 66 68

字符：ABD

小结一下：这个所谓的256进制和58进制算是我们自己自定义出来的，它的每一位都是按照ascii码和base58编码表填上去的，就像16进制的A对应10进制的11，在58进制里面就是C，在ascii是一个不可见字符。这两个在这里是为了进行编码的转化所提供的一种工具（至少我现在这么觉得）。总之，搞懂了这个base58的编码原理还是挺有意义的。