**技术选型**

**Scrapy VS requests+beautifulsoup**

1. requests和bs都是库，Scrapy是框架。
2. scrapy框架可以加入requests和bs。
3. scrapy是基于twisted，性能是最大的优势。
4. scrapy方便扩展，提供了很多内置的功能。
5. scrapy内置的css和xpath selector非常方便，bs最大的缺点就是慢

[回到顶部](http://www.cnblogs.com/jinxiao-pu/p/6707830.html#_labelTop)

**爬虫能做什么？**

1. 搜索引擎----百度、google、垂直领域搜索引擎
2. 推荐引擎----今日头条、一点资讯
3. 机器学习的数据样本
4. 数据分析（如金融数据分析）、舆情分析。

[回到顶部](http://www.cnblogs.com/jinxiao-pu/p/6707830.html#_labelTop)

**正则表达式**

1. 特殊字符 （原始字符串'booby123'）
2. ^ 开头 '^b.\*'----以b开头的任意字符
3. $ 结尾 '^b.\*3$'----以b开头，3结尾的任意字符
4. \* 任意长度（次数），≥0
5. ? 非贪婪模式，非贪婪模式尽可能少的匹配所搜索的字符串 '.\*?(b.\*?b).\*'----从左至右第一个b和的二个b之间的内容（包含b）
6. + 一次或多次
7. {2} 指定出现次数2次
8. {2,} 出现次数≥2次
9. {2,5} 出现次数2≤x≤5
10. | 或 例如，“z|food”能匹配“z”或“food”(此处请谨慎)。“[z|f]ood”则匹配“zood”或“food”或"zood"。

[] 中括号中任意一个符合即可（中括号里面没有分转义字符） '[abc]ooby123'----只要开头符合[]中任意一个即可

[^] 只要不出现[]的即可

[a-Z] 从小a到大Z '1[48357][0-9]{9}'----电话号码

. 任意字符

\s 匹配不可见字符 \n \t '你\s好'----可以匹配‘你 好’

\S 匹配可见字符，即普通字符

\w 匹配下划线在内的任何单词字符

\W 和上一个相反

[\u4E00-\u9FA5] 只能匹配汉字

() 要取出的信息就用括号括起来

\d 数字

**网页分类**

1. 静态网页
2. 动态网页
3. webservice（restapi）

[回到顶部](http://www.cnblogs.com/jinxiao-pu/p/6707830.html#_labelTop)

**深度优先和广度优先**

1. 网站的树结构
2. 深度优先算法和实现----递归
3. 广度优先算法和实现----队列

深度优先：一般采用递归的方式（函数体内调用自身）。要注意堆栈溢出（终止循环的条件）。

过程：

def depth\_tree(tree\_node)：

if tree\_node is not None:

print (tree\_node\_data)

if tree\_node.left is not None:

return depth\_tree(tree\_node.left.data)

if tree\_node.right is not None:

return depth\_tree(tree\_node.right.data)

广度优先：

def level\_queue(root):

if root is None:

return

my\_queue = []

node = root

my\_queue.append(node)

while my\_queue:

node = my\_queue.pop(0)

print （node.elem）

if no.lchild is not None:

my\_queue.append(node.lchild)

if no.rchild is not None:

my\_queue.append(node.rchild)

[回到顶部](http://www.cnblogs.com/jinxiao-pu/p/6707830.html#_labelTop)

**爬虫去重策略**

1. 将访问过的url保存到数据库中。
2. 将访问过的url保存到set中，只要o(1)【这是常数阶时间复杂度】的代价就可以查询url

100000000x2bytex50个字符/1024/1024/1024≈9G 一亿条数据，一条50字符

1. **url经过md5等方法哈希后保存到set中**（比较常用）
2. ........

[回到顶部](http://www.cnblogs.com/jinxiao-pu/p/6707830.html#_labelTop)

**字符串编码**

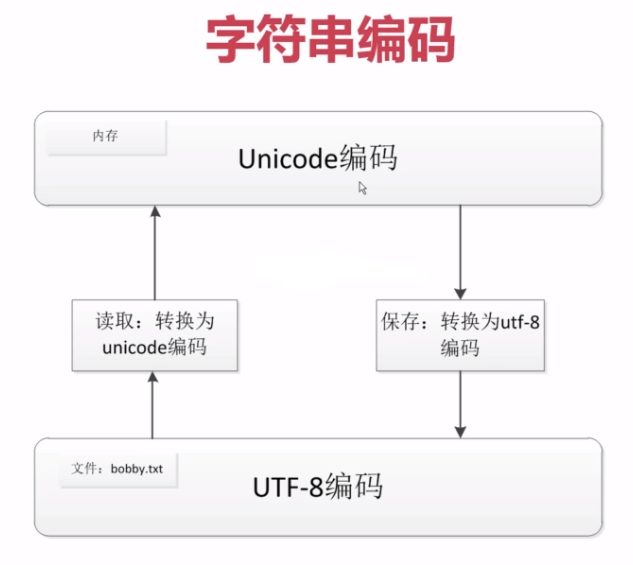
1. 计算机只能识别数字，文本转换为数字才能处理。计算机中8个bit作为一个字节，所以一个字节能表示最大的数字就是255.
2. 计算机是美国人发明的，所以一个字节可以表示所有字符了，所以ASCII（一个字节）编码就成为美国人标准编码。
3. 但是ASCII处理中文明显是不够的，中文不止255个汉字，所以中国制定了GB2312编码，用两个字节表示一个汉字。GB2312还把ASCII包含进去了，同理，日文，韩文等等上百个国家为了解决这个问题就都发展了一套字节的编码，标准就越来越多，如果出现多种语言混合就一定会出现乱码。
4. 于是unicode出现了，将所有语言统一到一套编码里。
5. ASCII和unicode编码：

（1）字母A用ASCII编码十进制65，二进制01000001

（2）汉字‘中’已超多ASCII编码的范围，用unicode编码是20013，二进制01001110 00101101

（3）A用unicode编码中只需要前面补0，二进制是 00000000 01000001

1. 乱码问题解决可，但是如果内容全是英文，unicode编码比ASCII需要多一倍的存储空间，同时如果传输需要多一倍的传输。
2. 所以出现了可变长的编码“utf-8”，把英文变长一个字节，汉字3个字节。特别生僻的变成4-6字节。如果传输大量的英文，utf-8作用就很明显。



**python 中encode与decode**

encode将Unicode字符串转换为想要的格式,换句话理解encode的字符串必须先转换为Unicode编码。

decode 是将字符串转换为 Unicode

在Python3中所有字符类型统一为Unicode编码

在Python2中：

windows: 中文字符串为gb2312编码格式

linux: 中文字符串为 utf8编码格式

例如将下面的字符串转换为utf8格式：

s = “测试字符串”

su = u“宇宙中心”

Windows:

s.decode(‘gb2312’).encode(‘utf-8’)

su.encode(‘utf8’)

Linux：

s.decode(‘utf-8).encode(‘utf-8’)

su.encode(‘utf8’)