**Python 的各种符号**

**Table 1.1. Python命令行选项**

| **选项** | **作用** |
| --- | --- |
| -c cmd | 在命令行直接执行python代码。如python -c 'print "hello world"'。 |
| -d | 脚本编译后从解释器产生调试信息。同PYTHONDEBUG=1。 |
| -E | 忽略环境变量。 |
| -h | 显示python命令行选项帮助信息。 |
| -i | 脚本执行后马上进入交互命令行模式。同PYTHONINSPECT=1。 |
| -O | 在执行前对解释器产生的字节码进行优化。同 PYTHONOPTIMIZE=1。 |
| -OO | 在执行前对解释器产生的字节码进行优化，并删除优化代码中的嵌入式文档字符串。 |
| -Q arg | 除法规则选项，-Qold(default)，-Qwarn，-Qwarnall，-Qnew。 |
| -S | 解释器不自动导入site.py模块。 |
| -t | 当脚本的tab缩排格式不一致时产生警告。 |
| -u | 不缓冲stdin、stdout和stderr，默认是缓冲的。同PYTHONUNBUFFERED=1。 |
| -v | 产生每个模块的信息。如果两个-v选项，则产生更详细的信息。同PYTHONVERBOSE=x。 |
| -V | 显示Python的版本信息。 |
| -W arg | 出错信息控制。(arg is action:message:category:module:lineno) |
| -x | 忽略源文件的首行。要在多平台上执行脚本时有用。 |
| file | 执行file里的代码。 |
| - | 从stdin里读取执行代码。 |

**Table 2.1. Python运算符列表**

| **运算符** | **描述** |
| --- | --- |
| x+y，x-y | 加、减，“+”号可重载为连接符 |
| x\*y，x\*\*y，x/y，x%y | 相乘、求平方、相除、求余，“\*”号可重载为重复，“%”号可重载为格式化 |
| <，<=，>，>=，==，<>，!= | 比较运算符 |
| +=，-=，\*=，/=，%=，\*\*=，<<=，>>=，&=，^=，|= | 自变运算符 |
| x|y | 按位或 |
| x^y | 按位异或 |
| x&y | 按位与 |
| ~x | 按位取反 |
| x<<，x>>y | x向左或向右移y位 |
| is, is not | 等同测试 |
| in, not in | 是否为成员测试 |
| or，and，not | 逻辑运算符 |
| x[i]，x[i:j]，x.y，x(...) | 索引，分片，限定引用，函数调用 |
| (...)，[...]，{...}，'...' | 元组，列表，字典，转化为字符串 |

**Table 2.2. 运算符优先顺序列表(从最高到最低)**

| **运算符** | **描述** |
| --- | --- |
| 'expr' | 字符串转换 |
| {key:expr,...} | 字典 |
| [expr1,expr2...] | 列表 |
| (expr1,expr2,...) | 元组 |
| function(expr,...) | 函数调用 |
| x[index:index] | 切片 |
| x[index] | 下标索引取值 |
| x.attribute | 属性引用 |
| ~x | 按位取反 |
| +x，-x | 正，负 |
| x\*\*y | 幂 |
| x\*y，x/y，x%y | 乘，除，取模 |
| x+y，x-y | 加，减 |
| x<<y，x>>y | 移位 |
| x&y | 按位与 |
| x^y | 按位异或 |
| x|y | 按位或 |
| x<y，x<=y，x==y，x!=y，x>=y，x>y | 比较 |
| x is y，x is not y | 等同测试 |
| x in y，x not in y | 成员判断 |
| not x | 逻辑否 |
| x and y | 逻辑与 |
| x or y | 逻辑或 |
| lambda arg,...:expr | Lambda匿名函数 |

**Table 2.3.**

| **对象/常量** | **值** |
| --- | --- |
| "" | 假 |
| "string" | 真 |
| 0 | 假 |
| >=1 | 真 |
| <=-1 | 真 |
| ()空元组 | 假 |
| []空列表 | 假 |
| {}空字典 | 假 |
| None | 假 |

**Table 3.1. 字符串格式化代码**

| **格式** | **描述** |
| --- | --- |
| %% | 百分号标记 |
| %c | 字符及其ASCII码 |
| %s | 字符串 |
| %d | 有符号整数(十进制) |
| %u | 无符号整数(十进制) |
| %o | 无符号整数(八进制) |
| %x | 无符号整数(十六进制) |
| %X | 无符号整数(十六进制大写字符) |
| %e | 浮点数字(科学计数法) |
| %E | 浮点数字(科学计数法，用E代替e) |
| %f | 浮点数字(用小数点符号) |
| %g | 浮点数字(根据值的大小采用%e或%f) |
| %G | 浮点数字(类似于%g) |
| %p | 指针(用十六进制打印值的内存地址) |
| %n | 存储输出字符的数量放进参数列表的下一个变量中 |

**Table 3.2. python支持的转义字符表**

| **转义字符** | **描述** |
| --- | --- |
| \(在行尾时) | 续行符 |
| \\ | 反斜杠符号 |
| \' | 单引号 |
| \" | 双引号 |
| \a | 响铃 |
| \b | 退格(Backspace) |
| \e | 转义 |
| \000 | 空 |
| \n | 换行 |
| \v | 纵向制表符 |
| \t | 横向制表符 |
| \r | 回车 |
| \f | 换页 |
| \oyy | 八进制数yy代表的字符，例如：\o12代表换行 |
| \xyy | 十进制数yy代表的字符，例如：\x0a代表换行 |
| \other | 其它的字符以普通格式输出 |

* **Table 3.3. 列表对象支持的方法**

| **方法** | **描述** |
| --- | --- |
| append(x) | 在列表尾部追加单个对象x。使用多个参数会引起异常。 |
| count(x) | 返回对象x在列表中出现的次数。 |
| extend(L) | 将列表L中的表项添加到列表中。返回None。 |
| Index(x) | 返回列表中匹配对象x的第一个列表项的索引。无匹配元素时产生异常。 |
| insert(i,x) | 在索引为i的元素前插入对象x。如list.insert(0,x)在第一项前插入对象。返回None。 |
| pop(x) | 删除列表中索引为x的表项，并返回该表项的值。若未指定索引，pop返回列表最后一项。 |
| remove(x) | 删除列表中匹配对象x的第一个元素。匹配元素时产生异常。返回None。 |
| reverse() | 颠倒列表元素的顺序。 |
| sort() | 对列表排序，返回none。bisect模块可用于排序列表项的添加和删除。 |

**Table 3.4. 字典方法**

| **方法** | **描述** |
| --- | --- |
| has\_key(x) | 如果字典中有键x，则返回真。 |
| keys() | 返回字典中键的列表 |
| values() | 返回字典中值的列表。 |
| items() | 返回tuples的列表。每个tuple由字典的键和相应值组成。 |
| clear() | 删除字典的所有条目。 |
| copy() | 返回字典高层结构的一个拷贝，但不复制嵌入结构，而只复制对那些结构的引用。 |
| update(x) | 用字典x中的键值对更新字典内容。 |
| get(x[,y]) | 返回键x，若未找到该键返回none，若提供y，则未找到x时返回y。 |

**Table 3.5. 数组类型代码**

| **代码** | **等价的C类型** | **以字节为单位的最小尺寸** |
| --- | --- | --- |
| c | char | 1 |
| b(B) | byte(unsigned byte) | 1 |
| h(H) | short(unsigned short) | 2 |
| i(I) | int(unsigned int) | 2 |
| l(L) | long(unsigned long) | 4 |
| f | float | 4 |
| d | double | 8 |

**Table 9.1. mode**

| **模式** | **描述** |
| --- | --- |
| r | 以读方式打开文件，可读取文件信息。 |
| w | 以写方式打开文件，可向文件写入信息。 |
| a | 以追加方式打开文件，文件指针自动移到文件尾。 |
| r+ | 以读写方式打开文件，可对文件进行读和写操作。 |
| w+ | 消除文件内容，然后以读写方式打开文件。 |
| a+ | 以读写方式打开文件，并把文件指针移到文件尾。 |
| b | 以二进制模式打开文件，而不是以文本模式。该模式只对Windows或Dos有效，类Unix的文件是用二进制模式进行操作的。 |

**Table 9.2. bufsize**

| **bufsize取值** | **描述** |
| --- | --- |
| 0 | 禁用缓冲 |
| 1 | 行缓冲 |
| >1 | 指定缓冲区的大小 |
| <1 | 系统默认的缓冲区大小 |

open()函数返回一个文件对象，我们可通过read()或write()函数对文件进行读写操作，下面是一些文件对象方法：

**Table 9.3. 文件对象方法**

| **方法** | **描述** |
| --- | --- |
| f.close() | 关闭文件，记住用open()打开文件后一定要记得关闭它，否则会占用系统的可打开文件句柄数。 |
| f.fileno() | 获得文件描述符 |
| f.flush() | 刷新输出缓存 |
| f.isatty() | 如果文件是一个交互终端，则返回True，否则返回False。 |
| f.read([count]) | 读出文件，如果有count，则读出count个字节。 |
| f.readline() | 读出一行信息。 |
| f.readlines() | 读出所有行，也就是读出整个文件的信息。 |
| f.seek(offset[,where]) | 把文件指针移动到相对于where的offset位置。offset为0表示文件开始处，这是默认值 ；1表示当前位置；2表示文件结尾。 |
| f.tell() | 获得文件指针位置。 |
| f.truncate([size]) | 截取文件，使文件的大小为size。 |
| f.write(string) | 把string字符串写入文件。 |
| f.writelines(list) | 把list中的字符串一行一行地写入文件。 |

**Table 10.1. 正则表达式基本字符**

| **字符** | **描述** |
| --- | --- |
| text | 匹配text字符串 |
| . | 匹配除换行符之外的任意一个单个字符 |
| ^ | 匹配一个字符串的开头 |
| $ | 匹配一个字符串的末尾 |

在正则表达式中，我们还可用匹配限定符来约束匹配的次数。

**Table 10.2. 匹配限定符**

| **最大匹配** | **最小匹配** | **描述** |
| --- | --- | --- |
| \* | \*? | 重复匹配前表达式零次或多次 |
| + | +? | 重复匹配前表达式一次或多次 |
| ? | ?? | 重复匹配前表达式零次或一次 |
| {m} | {m}? | 精确重复匹配前表达式m次 |
| {m,} | {m,}? | 至少重复匹配前表达式m次 |
| {m,n} | {m,n}? | 至少重复匹配前表达式m次，至多重复匹配前表达式n次 |

据上所述，".\*"为最大匹配，能匹配源字符串所有能匹配的字符串。".\*?"为最小匹配，只匹配第一次出现的字符串。如：d.\*g能匹配任意以d开头，以g结尾的字符串，如"debug"和"debugging"，甚至"dog is walking"。而d.\*?g只能匹配"debug"，在"dog is walking"字符串中，则只匹配到"dog "。

在一些更复杂的匹配中，我们可用到组和运算符。

**Table 10.3. 组和运算符**

| **组** | **描述** |
| --- | --- |
| [...] | 匹配集合内的字符，如[a-z],[1-9]或[,./;'] |
| [^...] | 匹配除集合外的所有字符，相当于取反操作 |
| A|B | 匹配表达式A或B，相当于OR操作 |
| (...) | 表达式分组，每对括号为一组，如([a-b]+)([A-Z]+)([1-9]+) |
| \number | 匹配在number表达式组内的文本 |

有一组特殊的字符序列，用来匹配具体的字符类型或字符环境。如\b匹配字符边界，food\b匹配"food"、"zoofood"，而和"foodies"不匹配。

**Table 10.4. 特殊字符序列**

| **字符** | **描述** |
| --- | --- |
| \A | 只匹配字符串的开始 |
| \b | 匹配一个单词边界 |
| \B | 匹配一个单词的非边界 |
| \d | 匹配任意十进制数字字符，等价于r'[0-9]' |
| \D | 匹配任意非十进制数字字符，等价于r'[^0-9]' |
| \s | 匹配任意空格字符（空格符、tab制表符、换行符、回车、换页符、垂直线符号） |
| \S | 匹配任意非空格字符 |
| \w | 匹配任意字母数字字符，等价于[a-zA-Z0-9\_]。注意，包含'\_'字符。 |
| \W | 匹配任意非字母数字字符，等价于[^a-zA-Z0-9\_]。 |
| \Z | 仅匹配字符串的尾部 |
| \\ | 匹配反斜线字符 |

有一套声明(assertion)对具体事件进行声明。

**Table 10.5. 正则表达式声明**

| **声明** | **描述** |
| --- | --- |
| (?iLmsux) | 匹配空字符串，iLmsux字符对应下表的正则表达式修饰符。 |
| (?:...) | 匹配圆括号内定义的表达式，但不填充字符组表。 |
| (?P<name>) | 匹配圆括号内定义的表达式，但匹配的表达式还可用作name标识的符号组。 |
| (?P=name) | 匹配所有与前面命名的字符组相匹配的文本。 |
| (?#...) | 引入注释，忽略圆括号内的内容。 |
| (?=...) | 如果所提供的文本与下一个正则表达式元素匹配，这之间没有多余的文本就匹配。这允许在一个表达式中进行超前操作，而不影响正则表达式其余部分的分析。如"Martin"其后紧跟"Brown"，则"Martin(?=Brown)"就只与"Martin"匹配。 |
| (?!...) | 仅当指定表达式与下一个正则表达式元素不匹配时匹配，是(?=...)的反操作。 |
| (?<=...) | 如果字符串当前位置的前缀字符串是给定文本，就匹配，整个表达式就在当前位置终止。如(?<=abc)def表达式与"abcdef"匹配。这种匹配是对前缀字符数量的精确匹配。 |
| (?<!...) | 如果字符串当前位置的前缀字符串不是给定的正文，就匹配，是(?<=...)的反操作。 |

正则表达式还支持一些处理标志，它会影响正则式的执行方法。

**Table 10.6. 处理标志**

| **标志** | **描述** |
| --- | --- |
| I或IGNORECASE | 忽略表达式的大小写来匹配文本。 |

**Table 10.7. MatchObject对象方法**

| **方法** | **描述** |
| --- | --- |
| expand(template) | 展开模板中用反斜线定义的内容。 |
| m.group([group,...]) | 返回匹配的文本，是个元组。此文本是与给定group或由其索引数字定义的组匹配的文本，如果没有组定组名，则返回所有匹配项。 |
| m.groups([default]) | 返回一个元组，该元组包含模式中与所有组匹配的文本。如果给出default参数，default参数值就是与给定表达式不匹配的组的返回值。default参数的默认取值为None。 |
| m.groupdict([default]) | 返回一个字典，该字典包含匹配的所有子组。如果给出default参数，其值就是那些不匹配组的返回值。default参数的默认取值为None。 |
| m.start([group]) | 返回指定group的开始位置，或返回全部匹配的开始位置。 |
| m.end([group]) | 返回指定group的结束位置，或返回全部匹配的结束位置。 |
| m.span([group]) | 返回两元素组，此元组等价于关于一给定组或一个完整匹配表达式的(m.start(group),m.end(group)))列表 |
| m.pos | 传递给match()或search()函数的pos值。 |
| m.endpos | 传递给match()或search()函数的endpos值。 |
| m.lastindex |  |
| m.lastgroup |  |
| m.re | 创建这个MatchObject对象的正则式对象 |
| m.string | 提供给match()或search()函数的字符串。 |

**Table 10.8. 正则式对象方法/属性**

| **方法/属性** | **描述** |
| --- | --- |
| r.search(string[,pos[,endpos]]) | 同search()函数，但此函数允许指定搜索的起点和终点 |
| r.match(string[,pos[,endpos]]) | 同match()函数，但此函数允许指定搜索的起点和终点 |
| r.split(string[,max]) | 同split()函数 |
| r.findall(string) | 同findall()函数 |
| r.sub(replace,string[,count]) | 同sub()函数 |
| r.subn(replace,string[,count]) | 同subn()函数 |
| r.flags | 创建对象时定义的标志 |
| r.groupindex | 将r'(?Pid)'定义的符号组名字映射为组序号的字典 |
| r.pattern | 在创建对象时使用的模式 |