# 1\b

## 解释：表着单词的开头或结尾，也就是单词的分界处。虽然通常英文的单词是由空格，标点符号或者换行来分隔的，但是\b并不匹配这些单词分隔字符中的任何一个，它**只匹配一个位置**。

### 1.例1，精确地查找hi这个单词的话，我们应该使用

#### \bhi\b。

### 2.例2，hi后面不远处跟着一个Lucy，你应该用

#### \bhi\b.\*\bLucy\b。

# 2\d

## 解释：匹配一位数字(0，或1，或2，或……)

## 1，例1，以0开头，然后是两个数字，然后是一个连字号“-”，最后是8个数字(也就是中国的电话号码。当然，这个例子只能匹配区号为3位的情形)。

#### 0\d\d-\d\d\d\d\d\d\d\d

### 例2.这里\d后面的{2}({8})的意思是前面\d必须连续重复匹配2次(8次)。

#### 0\d{2}-\d{8}。

### 例3. 匹配1个或更多连续的数字。这里的+是和\*类似的元字符，不同的是\*匹配重复任意次(可能是0次)，而+则匹配重复1次或更多次。

#### \d+

### 例4 QQ号必须为5位到12位数字时

#### ^\d{5,12}$

### 例5.匹配三位数字加上一个英文句号(这个整体也就是这个分组)重复3次

#### (\d{1,3}\.){3}

# 3. –

## 解释：不是元字符，只匹配它本身

# 4. \w

## 解释：匹配字母或数字或下划线或汉字

### 例1：以字母a开头的单词——先是某个单词开始处(\b)，然后是字母a,然后是任意数量的字母或数字(\w\*)，最后是单词结束处(\b)。

#### \ba\w\*\b

### 例2： 匹配刚好6个字符的单词。

#### \b\w{6}\b

# 5. \s

## 解释：匹配任意的空白符，包括空格，制表符(Tab)，换行符，中文全角空格等

# 6. ^

## 解释:匹配字符串的开始，除非在方括号表达式中使用，此时它表示不接受该字符集合

### 例1匹配除了x以外的任意字符

#### [^x]

### 匹配除了aeiou这几个字母以外的任意字符

#### [^aeiou]

# 7. $

## 解释：匹配字符串的结束

# 8. \

## 解释;转义字符，如果你想查找元字符本身的话，比如你查找.,或者\*,就出现了问题：你没办法指定它们，因为它们会被解释成别的意思。这时你就得使用\来取消这些字符的特殊意义。因此，你应该使用\.和\\*。当然，要查找\本身，你也得用\\.例如：deerchao\.net匹配deerchao.net，C:\\Windows匹配C:\Windows

# 9. ?

## 解释：重复零次或一次

# 10 []

## 解释：字符范围，像[0-9]代表的含意与\d就是完全一致的：一位数字；同理[a-z0-9A-Z\_]也完全等同于\w（如果只考虑英文的话）。

# 11 |

## 解释：0\d{2}-\d{8}|0\d{3}-\d{7}这个表达式能匹配两种以连字号分隔的电话号码：一种是三位区号，8位本地号(如010-12345678)，一种是4位区号，7位本地号(0376-2233445)。

# 12 （）

## 解释: \(?0\d{2}\)?[- ]?\d{8}|0\d{2}[- ]?\d{8}这个表达式匹配3位区号的电话号码，其中区号可以用小括号括起来，也可以不用，区号与本地号间可以用连字号或空格间隔，也可以没有间隔

# 13. \D

## 解释：匹配任意非数字的字符

# 14 \B

## 解释：匹配不是单词开头或结束的位置

# 15.其他的知识点

|  |  |
| --- | --- |
| \a | 报警字符(打印它的效果是电脑嘀一声) |
| \b | 通常是单词分界位置，但如果在字符类里使用代表退格 |
| \t | 制表符，Tab |
| \r | 回车 |
| \v | 竖向制表符 |
| \f | 换页符 |
| \n | 换行符 |
| \e | Escape |
| \0nn | ASCII代码中八进制代码为nn的字符 |
| \xnn | ASCII代码中十六进制代码为nn的字符 |
| \unnnn | Unicode代码中十六进制代码为nnnn的字符 |
| \cN | ASCII控制字符。比如\cC代表Ctrl+C |
| \A | 字符串开头(类似^，但不受处理多行选项的影响) |
| \Z | 字符串结尾或行尾(不受处理多行选项的影响) |
| \z | 字符串结尾(类似$，但不受处理多行选项的影响) |
| \G | 当前搜索的开头 |
| \p{name} | Unicode中命名为name的字符类，例如\p{IsGreek} |
| (?>exp) | 贪婪子表达式 |
| (?<x>-<y>exp) | 平衡组 |
| (?im-nsx:exp) | 在子表达式exp中改变处理选项 |
| (?im-nsx) | 为表达式后面的部分改变处理选项 |
| (?(exp)yes|no) | 把exp当作零宽正向先行断言，如果在这个位置能匹配，使用yes作为此组的表达式；否则使用no |
| (?(exp)yes) | 同上，只是使用空表达式作为no |
| (?(name)yes|no) | 如果命名为name的组捕获到了内容，使用yes作为表达式；否则使用no |
| (?(name)yes) | 同上，只是使用空表达式作为no |