# 正则表达式

### 初始化的方式

+ 字面量 /partten/flags

+ 构造函数 new Regexp(partten, flags)

### 元字符

+ `.` 全字符 用于匹配单个的全部符号(\n和\t)除外

+ `a|b` 或字符 匹配a或者b

+ `\` 转义符 用来转义某些特定的符号 形成特定的功能 或者 让特定符号变成普通字符串

  + `\d` 所有的数字

  + `\w` 所有的单词

  + `\s` 所有的空白

  + `\b` 单词的边界

  + `\n` 换行符

  + `\t` 制表符

  + `\D` 非数字

  + `\W` 非单词

  + `\S` 非空白

  + `\B` 非单词边界 也就是字母和字母之间间隙

+ `^`

  + 行首 在正则的头部他表示行首

  + 非 在非正则的头部他表示取反

+ `$` 行尾

+ `[abc]` a或者b或者c

+ `[^abc]` 非a且非b且非c

+ `[a-c]` a或者b或者c

+ `[^a-c]` a到c以外的字符

+ `{n}` 前面那个语法匹配n次

+ `{n,}` 前面那个语法至少匹配n次

+ `{n,m}` 前面那个语法至少匹配n次至多匹配m次

+ `+`  {1,}前一个语法出现至少1次 等价于{1,} 有多少

+ `?` {0,1}前一个语法出现至少0次 至多1次 等价于 {0,1} 有没有

+ `\*` {0,}前一个语法出现至少0次 至多正无穷次 等价于 {0,}