1. 什么是正则表达式：正则表达式也可以称为规则表达式，就是用来验证字符串是否符合格式要求或者验证字符串中是否包含符合要求的子串。正则表达式有普通字符和特殊字符构成。
2. 正则表达式的创建
   1. 利用构造方法创建
      1. var 表达式名称 = new RegExp(表达式);
   2. 利用字面量形式创建
      1. var 表达式名称 = /表达式/修饰符
3. 正则表达式对象常用方法——test方法
   1. 作用：验证字符串是否符合要求
   2. 格式：正则表达式.test(字符串);
   3. 返回值：如果符合要求返回值为true，如果不符合要求返回值为false
4. 正则表达式的特殊字符（元字符）
   1. ^：表示字符串起始位置，即以某个字符开头，如/^a/表示以a开头
   2. $：表示字符串结束位置，即以某个字符结尾，如/a$/表示以a结尾
   3. \*：表示它前面的字符最少0个最多任意个，如/a\*/表示字符串中包含0到多个字符a
   4. +：表示它前面的字符最少1个最多任意个，如/a+/表示字符串中包含1到多个字符a
   5. ？：表示它前面的字符最少0个，最多1个，如/a？/表示字符串中包含0到1个字符a
   6. {n}：表示它前面的字符有n个，如/a{3}/表示字符串中包含3个字符a
   7. {n，}：表示它前面的字符最少n个，最多无限，如/a{3，}/表示字符串中包含3到任意多个字符a
   8. {n,m}：表示它前面的字符最少为n个，最多为m个，如/a{3，6}/表示字符串中包含3到6个字符a
   9. \：表示转义字符
   10. \d：表示所有数字，注意使用构造方法创建正则表达式时，转义字符的反斜线应该写两条即new RegExp(“\\d”)表示包含任意数字
   11. \D：表示所有非数字
   12. \w：表示字母、数字、下划线
   13. \W：表示除字母、数字、下划线以外的内容
   14. \s：表示空白符（空格、制表符等）
   15. \S：表示非空白符
   16. .：代表除\n外的所有字符
   17. []：包含如下常用形式
       1. [值1值2值3]表示值1，值2，值3中的任意一个字符，如[12345]表示12345中的任意字符
       2. [值1-值2]表示值1到值2间的任意一个字符，[1-5]表示数字1-5间的任意一个字符
       3. [^值1值2值3]表示值1，值2，值3以外的字符
       4. [\u4e00-\u9fa5] 表示任意一个汉字
   18. |：表示或，如/a|b|c/表示字符串中包含a或b或c
   19. （）：表示分组，可以使用$1/$2等取得组中的内容，格式RegExp.$1
   20. 常用正则

|  |  |
| --- | --- |
| **用户名** | /^[a-z0-9\_-]{3,16}$/ |
| **密码** | /^[a-z0-9\_-]{6,18}$/ |
| **十六进制值** | /^#?([a-f0-9]{6}|[a-f0-9]{3})$/ |
| **电子邮箱** | /^([a-z0-9\_\.-]+)@([\da-z\.-]+)\.([a-z\.]{2,6})$/ /^[a-z\d]+(\.[a-z\d]+)\*@([\da-z](-[\da-z])?)+(\.{1,2}[a-z]+)+$/ |
| **URL** | /^(https?:\/\/)?([\da-z\.-]+)\.([a-z\.]{2,6})([\/\w \.-]\*)\*\/?$/ |
| **IP 地址** | /((2[0-4]\d|25[0-5]|[01]?\d\d?)\.){3}(2[0-4]\d|25[0-5]|[01]?\d\d?)/ /^(?:(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$/ |
| **HTML 标签** | /^<([a-z]+)([^<]+)\*(?:>(.\*)<\/\1>|\s+\/>)$/ |
| **删除代码\\注释** | (?<!http:|\S)//.\*$ |
| **Unicode编码中的汉字范围** | /^[\u2E80-\u9FFF]+$/ |