* 正则的概念
* 正则表达式(regular expression)是一个描述**字符规则**的对象。ECMAScript 的RegExp 类（构造函数）表示正则表达式，而String 和 RegExp 都定义了使用正则表达式进行强大的模式匹配和文本检索与替换的方法。
* 正则表达式主要用来验证客户端的输入数据。 用户填写完表单单击提交按钮之后， 表单就会被发送到服务器，在服务器端通常会用 PHP、JSP 等服务器语言对其进行进一步处理 。因为客户端验证，可以节约大量的服务器端的系统资源，并且提供更好的用户体验。
* 创建方式
  + 使用RegExp构造函数创建

|  |
| --- |
| var reg= new RegExp( 'study' ，'igm' ); |

第一个参数就是我们的模式字符串（变量）

第二个参数可选，模式修饰符（i:忽略大小写、g:全局匹配、m:多行匹配）

* 用字面量方式创建(如果正则表达式中有变量则不可用此方式)

|  |
| --- |
| var reg= /study/ igm; |

* 测试正则表达式
* test(str)方法在字符串中查找是否存在指定的正则表达式并返回布尔值，如果存在则返回 true，不存在则返回 false。
  + exec()方法在一个字符串中执行匹配检索。如果它没有找到任何匹配返回null，如果找到一个匹配，它将返回一个数组。
* 使用正则的字符串方法
  + 使用 replace 替换匹配到的数据,2个参数：1代表正则，2代表用来s替换的字符
  + 使用 split 拆分成字符串数组
  + 使用 match 方法获取匹配内容，返回数组
  + 使用 search 来查找匹配数据，和全局没有关系，查找第一个匹配值的位置。(-1)
* 获取控制（常用）

字符类：单个字符和数字

|  |  |
| --- | --- |
| 元字符/元符号 | 匹配情况 |
| . | 匹配除换行符外的任意字符 |
| [a-z0-9] | 匹配括号中的字符集中的任意字符（小写字母和数字） |
| [^a-z0-9] | 匹配任意不在括号中的字符集中的字符 |
| \d | 匹配数字 |
| \D | 匹配非数字，同[^0-9]相同 |
| \w | 匹配字母和数字及\_ |
| \W | 匹配非字母和数字及\_ |

字符类：空白字符

|  |  |
| --- | --- |
| 元字符/元符号 | 匹配情况 |
| \0 | 匹配null字符 |
| \b | 匹配单词边界 |
| \n | 匹配换行符 |
| \s | 匹配空白字符、空格、制表符和换行符 |
| \S | 匹配非空白字符 |

字符类：锚字符

|  |  |
| --- | --- |
| 元字符/元符号 | 匹配情况 |
| ^ | 行首匹配 |
| $ | 行尾匹配 |

字符类：重复字符

|  |  |
| --- | --- |
| 元字符/元符号 | 匹配情况 |
| x? | 匹配0个或1个x |
| x\* | 匹配0个或任意多个x |
| x+ | 匹配至少一个x |
| (xyz)+ | 匹配至少一个(xyz) |
| x{m,n} | 匹配最少m个、最多n个x |
| x{m} | 匹配m个 |
| x{m,} | 匹配至少m个 |

字符类：替代字符

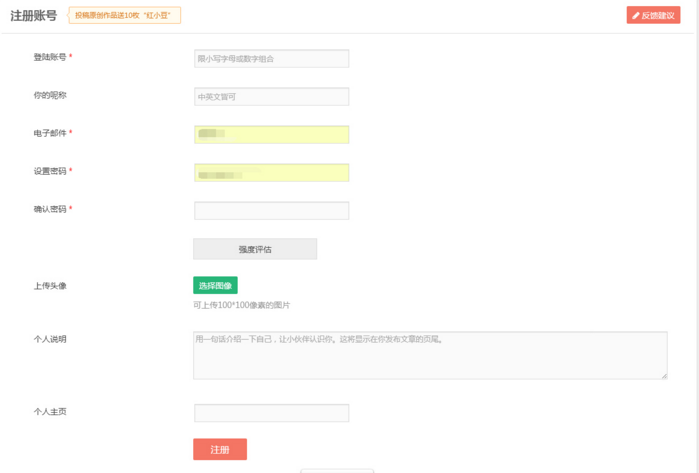
|  |  |
| --- | --- |
| 元字符/元符号 | 匹配情况 |
| this|where|logo | 匹配this或where或logo中任意一个 |

* 应用

1. 掌握正则的创建方式
2. 掌握正则表达式语言
3. 邮政编码检测(^[1-9][\d]{5}$)
4. 字符串首尾去空格
5. 邮件格式检测
6. 手机号检测
7. 身份证检测
8. 日期检测
9. 中文检测[\u4e00-\uf9a5]
10. 用户名检测
11. 字符串过滤
12. 密码检测

* 应用

1. 表单及AJAX的综合应用



* 综合应用

1. 掌握正则的封装
2. 掌握综合表单验证