正则表达式

# 正则表达式概述

**正则表达式**，又称**正规表示法**、**常规表示法**（英语：Regular Expression，在代码中常简写为regex、regexp或RE），计算机科学的一个概念。正则表达式使用单个字符串来描述、匹配一系列符合某个句法规则的字符串。 主要应用在前段技术。

# 正则的定义语法

在JavaScript中定义一个正则表达式使用：

第一种：var reg = /^…..$/

|  |
| --- |
| <script type=*"text/javascript"*>  **function** valid(){  //定义一个正则表达式  **var** reg = /^[a]$/;  alert(**typeof** reg);  }  </script> |

第二种创建正则的方式

|  |
| --- |
| <script type=*"text/javascript"*>  **function** valid(){  //定义一个正则表达式  //var reg = /^[a]$/;  //定义一个正则表达式  **var** reg = **new** RegExp('^[a]$');  alert(**typeof** reg);  }  </script> |

# 正则表达式的使用

|  |
| --- |
| <script type=*"text/javascript"*>  **function** valid(){  //定义一个正则表达式  //var reg = /^[a]$/;  //定义一个正则表达式  **var** reg = **new** RegExp('^[a]$');  //alert(typeof reg);  **var** uname = document.getElementById('username').value;  //学习正则表达式，只学习test方法  **if**(reg.test(uname)){  alert("合法");  }**else**{  alert("不合法");  }  }  </script>  </head>  <body>  用户名：<input type=*"text"* name=*"username"* id=*"username"*>  <input value=*"点击"* type=*"button"* onclick="valid()">  </body> |

# 正则表达式的写法

## 字符类匹配

[abc] a、b 或 c 中 一位

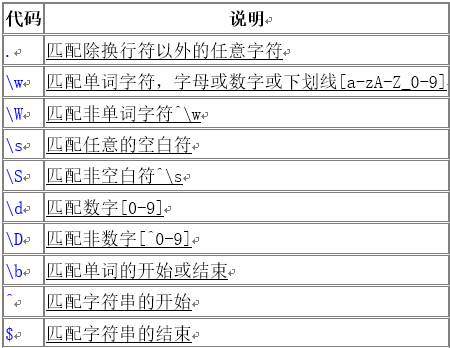
[^abc] 非a、b 或 c 中 一位

[a-zA-Z] a 到 z 或 A 到 Z，两头的字母包括在内

|  |
| --- |
| **function** valid(){  //定义一个正则表达式  //var reg = /^[a]$/;  //定义一个正则表达式  //var reg = new RegExp('^[^abc]$');  //表示从小a到z大A到Z任意的一位  //var reg = new RegExp('^[a-zA-Z]$');  //var reg = new RegExp('^[a-zA-Z0-9]$');  //var reg = new RegExp('^[a-zA-Z0-9\_]$');  **var** reg = **new** RegExp('^[0-9\_]$');  //alert(typeof reg);  **var** uname = document.getElementById('username').value;  //学习正则表达式，只学习test方法  **if**(reg.test(uname)){  alert("合法");  }**else**{  alert("不合法");  }  } |

## 元字符匹配

默认都是一位字符



## 数量词

X? 一次或一次也没有

X\* 零次或多次

X+ 一次或多次

X{n} 恰好 n 次

X{n,} 至少 n 次

X{n,m} 至少 n 次，但是不超过 m 次

## 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：你没法指定它们，因为它们会被解释成其它的意思。这时你就必须使用\来取消这些字符的特殊意义。因此，你应该使用\.和\\*，

注意：在[]之内的特殊字符自动转义。