正则表达式

概念:

一个用于规范字符串的表达式。

简单来说,就是写一个规则,然后通过这个规则可以从一个杂乱的字符串中找出符合规则的部分,也就可以用于判断一个字符串是否符合某个规则。还可以将一个字符串中符合规则的部分进行替换。

初体验:

```
var reg = /aaa/; // 写了一个规则,这个规则的意思是说3个a字符连在一起
var str = 'sdafaaaaa123wer'; // 一个杂乱的字符串
var arr = str.match(reg); // 从字符串中提取出符合规则的部分
console.log(arr); // ["aaa", index: 4, input: "sdafaaaaaa123wer"]
```

正则表达式其实就是一种规则,其实把正则称作规则表达式更为恰当。正则的语法基于一种古老的perl语言。

正则表达式的作用

1. 提取字符串: 从指定的字符串中将符合规则部分提取出来

2. 匹配正则:判断一个字符串是否跟指定的规则匹配

3. 替换字符串:将一个字符串中跟规则匹配的部分替换掉

正则表达式的创建

语法: /正则表达式主体/修饰符(可选)

```
var reg = new RegExp() // 构造函数方式
var reg = /hello/i; // 字面量方式
```

使用说明:正则表达式比较特殊,不是字符串,不用加引号,但是要有定界符,两边都是斜杠。在斜杠后面可以加一些特定的修饰符。

修饰符

标志	说明
i	忽略大小写
g	全局匹配,贪婪模式
gi	全局匹配+忽略大小写

正则表达式的组成

• 普通字符

• 特殊字符(元字符):正则表达式中有特殊意义的字符

元字符

常用元字符串

元字符	说明
\d	匹配数字
\D	匹配任意非数字的字符
\w	匹配字母或数字或下划线
\W	匹配任意不是字母,数字,下划线
\s	匹配任意的空白符
\S	匹配任意不是空白符的字符
	匹配除换行符以外的任意单个字符
۸	表示匹配行首的文本(以谁开始)
\$	表示匹配行尾的文本(以谁结束)

限定符

限定符	说明
*	重复零次或更多次
+	重复一次或更多次
?	重复零次或一次
{n}	重复n次
{n,}	重复n次或更多次
{n,m}	重复n到m次

其它

- [] 字符串用中括号括起来,表示匹配其中的任一字符,相当于或的意思
- [^] 匹配除中括号以内的内容

\ 转义符

- | 或者,选择两者中的一个。注意 | 将左右两边分为两部分,而不管左右两边有多长多乱
- () 从两个直接量中选择一个,分组 eg:gr(a|e)y匹配gray和grey [\u4e00-\u9fa5] 匹配汉字

匹配字符串

字符串的匹配有两个方法:

• 正则表达式.test(字符串):可以匹配为true,否则为false

```
/\d/.test('13522336699') //true
```

search() 匹配成功返回下标位置,不成功返回-1
 其实可以理解为查找字符串中符合正则表达式的子字符串的位置

```
var str = 'www.baidu.com'
str.search(/baidu/) // 4
```

提取字符串

语法: match(正则表达式)

例:

```
var reg = /aaa/g; // 贪婪匹配
var str = 'sdafaaaaaa123wer';
var arr = str.match(reg);
console.log(arr); // ["aaa", "aaa"]
```

语法: 正则表达式.exec(字符串)

```
var reg = /(a)aa/g; // 贪婪匹配
var str = 'sdafaa aaaa123wer';
var arr = reg.exec(str);
console.log(arr); // ["aaa", "a", index: 7, input: "sdafaa aaaa123wer"]
```

先匹配整体,再匹配小括号

案例

验证手机号:

```
^\d{11}$
```

验证邮编:

```
^\d{6}$
```

验证邮箱 xxx@1000phone.com:

```
^\w+@\w+\.\w+$
```

```
// 1. 提取工资
var str = "张三:1000,李四:5000,王五:8000。";
var array = str.match(/\d+/g);
console.log(array);

// 2. 提取email地址
var str = "123123@xx.com,fangfang@valuedopinions.cn 286669312@qq.com 2、
emailenglish@emailenglish.englishtown.com 286669312@qq.com...";
var array = str.match(/\w+@\w+\.\w+(\.\w+)?/g);
console.log(array);
```

替换字符串

语法:字符串.replace(正则表达式,新内容)

```
var str = 'i love html very much and html is very good';
var reg = /html/g;
var res = str.replace(reg,'js');
console.log(res); // i love js very much and js is very good
```

引用替换:

```
var str = '8/1/2019';
var reg = /(\d)\/(\d)\/(\d+)/;
var res = str.replace(reg,"$3/$1/$2");
console.log(res);
```

案例

1. 手机号检测

```
var reg = /^1[345789]\d{9}$/;
```

1. 身份证检测

```
var reg = /^[1-9]\d{16}([0-9]|X)$/;
```

- 1. 日期检测
- 2. 中文检测

```
var reg = /[\u4e00-\u9fa5]/;
```

- 1. 用户名检测
- 2. 字符串过滤
- 3. 正则\$符的应用