



## 第六十一课 正则表达式、元字符与量词

### 知识目标

- 正则表达式
- 元字符
- 量词
- test()
- prompt()

### 项目目标

- 实现密码验证效果



### 正则表达式

我们在进行账号登录的时候，通常需要输入符合一定规则的账号和密码才可以，像如下图所示的 QQ 注册界面，那怎么才能进行账号和密码的正确检索呢？

欢迎注册QQ  
每一天，乐在沟通。 [免费账号](#)

+86

可通过手机验证码注册

[立即注册](#)

☒ 我已阅读并同意相关服务协议及隐私条款





正则表达式：通常被用来检索、验证符合某一规则的文本。

我们来举一个例子，用正则表达式完全匹配下方这个手机号码，应该如何实现呢？

13284097645

代码如下：

/13284097645/

这样就实现了完全匹配。"/" 是语法规定，正则表达式中必须有。

下面我们再做两个练习，来深度了解下完全匹配：

(1) 在电话号码的开始和结尾处加了数字

121328409764598

/13284097645/

结果为符合完全匹配。

(2) 在电话号码中间插入了一个字母

1328409a7645

/13284097645/

结果不能实现完全匹配。

那如果想只要是 11 位数字就符合规则，应该怎么办呢？代码如下：

/[0-9]{11}/

[0-9] 代表 0-9 之间的数字，{11} 代表 11 位，这些字符都是什么呢？

## 元字符

正则表达式由普通字符、元字符和量词组成。普通字符包括大小写字母与数字。

元字符则具有特殊的含义，量词可以确定匹配字符的次数。

下面我们来认识一些常见的元字符：

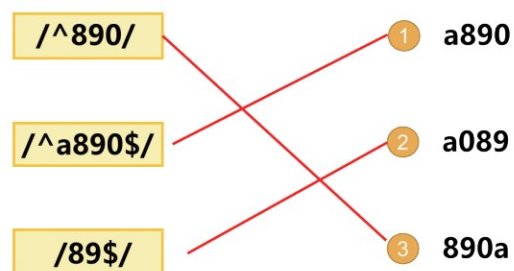
2



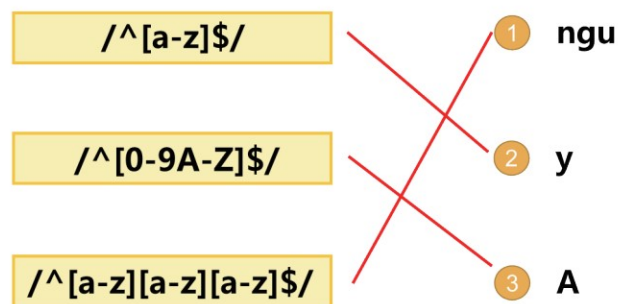
字符	描述
^	匹配输入字符串的开始位置
\$	匹配输入字符串的结束位置
[abc]	匹配所包含的任意一个字符
[a-z]或[A-Z]	匹配范围内的任意小写或大写字母字符
[0-9]	匹配从0到9的数字

来做几个练习吧：

(1)



(2)



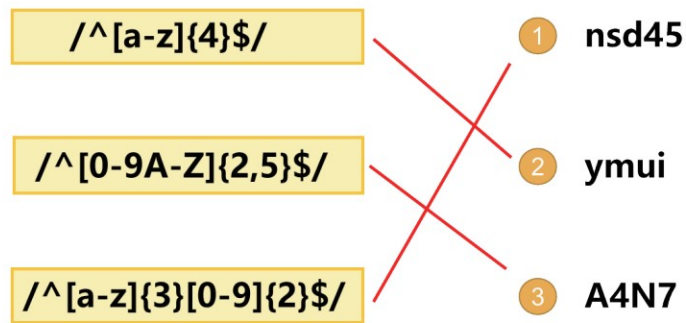
`/^[a-z][a-z][a-z]$/` 有没有简写的方式呢？

### 量词

我们来看几个常用的量词

量词	描述
{n}	匹配的字符出现n次
{n,m}	匹配的字符出现最少n次，最多m次

再来做个练习：



那么如何检测字符串是否匹配某个模式呢？

test()

正则表达式由普通字符、元字符和量词组成。普通字符包括大小写字母与数字。

元字符则具有特殊的含义，量词可以确定匹配字符的次数。

匹配的模式 .test( 要检测的字符 );

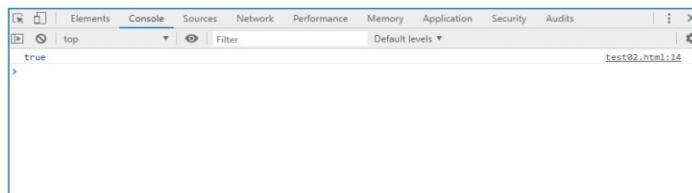
如果字符串中有匹配的值返回 true ， 否则返回 false。

我们来看一个例子：

```
var str="6nAA";  
var reg= /^[0-9a-zA-Z]{2,5}$/;  
var result=reg.test(str);  
console.log(result);
```

正则是可以赋值给变量的。

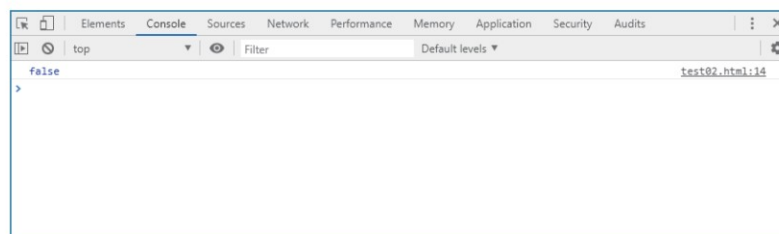
效果如图，该正则匹配此模式：



```
var str="86dKH9";  
var reg= /^[0-9a-zA-Z]{2,5}$/;  
var result=reg.test(str);  
console.log(result);
```



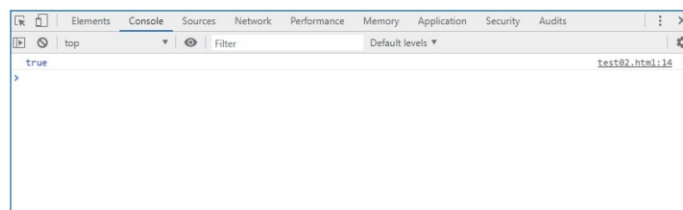
效果如图，该正则不匹配此模式：



其实正则还可以进行检索，代码如下：

```
var str="Hello world";  
var reg= /e/;  
var result=reg.test(str);  
console.log(result);
```

效果如图：



### prompt()

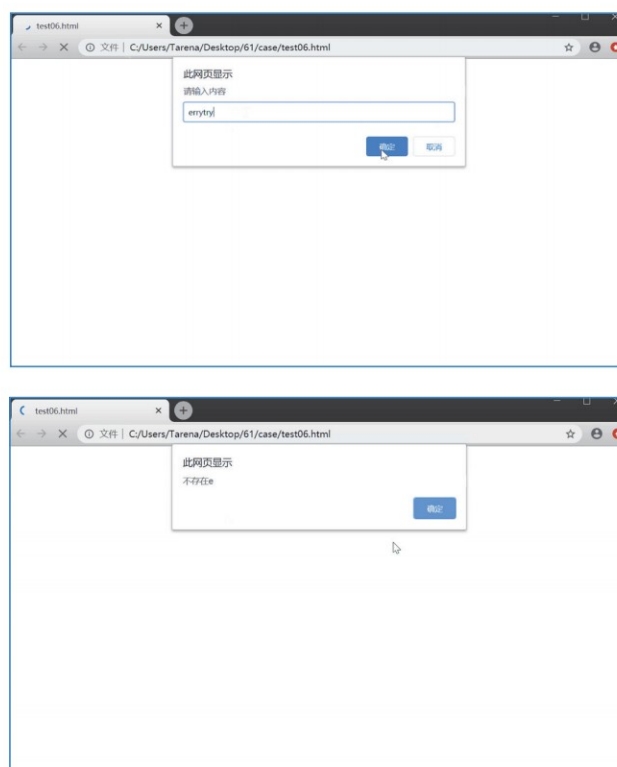
那我们怎样能验证用户输入的内容呢？

prompt()：用于显示可提示用户进行输入的对话框。

```
var str=prompt(" 请输入内容 ");  
var patt= /e/;  
if(patt.test(str)){  
    alert(" 存在 e");  
}else{  
    alert(" 不存在 e");  
}
```



效果如图：



实现如图所示效果：

**使用正则表达式验证用户输入的QQ号，最少5位数字，最多11位数字，符合规则则在警告框中显示格式正确，否则显示格式错误。**



```
var reg=/^[0-9]{5,11}$/;  
if(reg.test(prompt(" 输入 QQ 号码 ")){  
    alert(" 格式正确 ");  
}else{  
    alert(" 格式错误 ");  
}
```



小 tip



小tip:

prompt()提示框中输入的数字为**字符串**类型

完成项目





```
var ps = document.getElementById("ps");
document.onclick = function () {
    var userName = prompt(" 输入密码才能查看答案 ");
    var reg = /^[a-zA-Z0-9]{4,10}$/;
    if(reg.test(userName)){
        ps.style.color="red";
    }else{
        alert(" 密码不正确 ");
    }
}
```

效果如图：



(1) 下列关于元字符说法错误的是：

- A. ^ 匹配输入字符串的开始位置
- B. [0-9] 匹配从 0 到 9 的数字
- C. [a-z] 匹配范围内的任意小写字母字符
- D. & 匹配输入字符串的结束位置





(2) `/^[0-9][a-z]{3,7}$/`

下列选项符合上述正则表达式规则的是：

- A. 466g
- B. fnjh
- C. e4578
- D. 7dyty



实现如图所示效果：

使用正则表达式验证用户输入11位的密码，开头两位必须为小写字母，后9位可为数字、小写或大写字母，符合规则则在警告框中显示密码正确，否则显示密码错误。



实现如图所示效果：

使用正则表达式验证用户输入11位的手机号，第一位数字不能为0，符合规则则在警告框中显示格式正确，否则显示格式错误。



## 课后心得