



第六十三课 元字符与量词

知识目标

- 元字符
- 量词

项目目标

- 使用正则完成 Minecraft 设置页面的手机号验证和密码验证



元字符

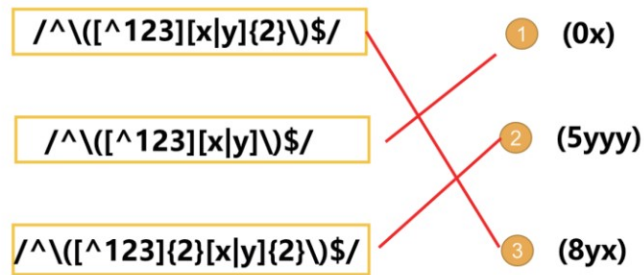
这节课呢我们再来学习几个元字符，如下：

字符	描述
\	将下一个字符标记为一个特殊字符,例如\"(\"匹配\"(\"
x y	匹配 x 或 y
[^abc]	匹配未包含的任意字符(除了abc)
(pattern)	匹配pattern并获取这一匹配

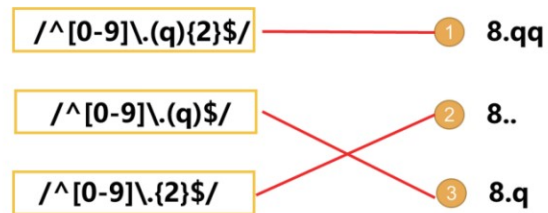


接着我们来做两个小练习：

(1)



(2)



下面我们来看一个例子，用正则表达式来规定 qq 邮箱的格式：

```
var str=prompt(" 请输入邮箱 ");
var reg= /^[0-9]+[a-z]*\\@?$/;
if(reg.test(str)){
    alert(" 格式正确 ");
}else{
    alert(" 格式错误 ");
}
```

效果如图：



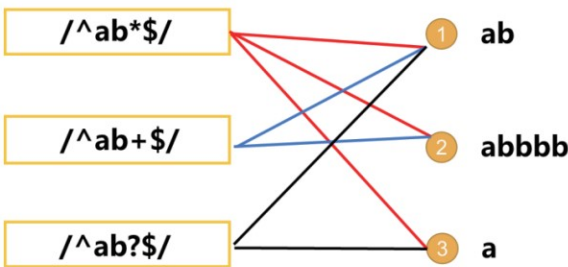


量词

下面我们再来学习几个量词：

量词	描述
*	匹配前面的子表达式零次或多次
+	匹配前面的子表达式一次或多次
?	匹配前面的子表达式零次或一次

来做个练习吧：



我们来看一个例子，代码如下：

```
var str=prompt(" 请输入格式正确的内容 ");
var reg= /^ [0-9]+ [a-z]* \ @? $ /;
if (reg.test(str)){
    alert(" 格式正确 ");
}else{
    alert(" 格式错误 ");
}
```

效果如图：





实现如图所示效果:

```
/^d{3}-d{8}d{4}-d{7}$/
```

符合上述正则表达式的形式有两种，对于不同的形式分别举个例子。

```
123-12345678  
1234-1234567
```

实现如图所示效果:

```
/^(d{3,4}-)?d{7,8}$/
```

符合上述正则表达式的形式有多种，对于不同的形式分别举个例子。
例如123-1234567

```
1234567  
12345678  
123-1234567  
123-12345678  
1234-1234567  
1234-12345678
```



小 tip



完成项目



(1) 手机号验证

```
function validate(i) {  
    ...  
    if (i == 3){  
        var reg = /^1[3|4|5|8]\d{9}$/;  
        var phone = inputsArr[i].value;  
        if (!(reg.test(phone))){  
            alert("手机号输入有误");  
            inputsArr[i].value = "";  
            return false;  
        }  
    }  
}
```



(2) 密码验证

```
function validate(i) {  
    ...  
    if (i == 4){  
        var reg1 = /[A-Z]/;  
        var reg2 = /[a-z]/;  
        var reg3 = /[0-9]/;  
        var password = inputsArr[i].value;  
        if (!(reg1.test(password)&&reg2.test(password)  
            &&reg3.test(password))||password.length<6){  
            inputsArr[i].value = "";  
            alert(" 密码设置有误 ");  
        }  
    }  
}
```

效果如图:



(1) 下列关于元字符说法错误的是:

- A. \ 将下一个字符标记为一个特殊字符
- B. x|y 匹配 x 或 y
- C. (pattern) 匹配 pattern 并获取这一匹配
- D. [^abc] 匹配包含的任意字符

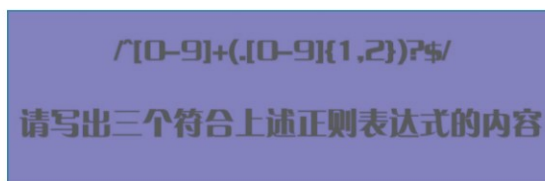


(2) 下列不属于量词的是:

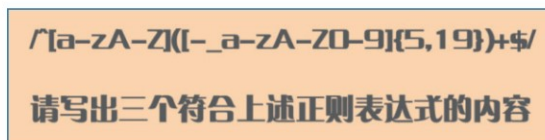
- A. +
- B. *
- C. /
- D. ?



实现如图所示效果:



实现如图所示效果:





课后心得