千锋教育
QIAN FENG EDUCATION

做真实的自己，用良心做教育

正则表达式

H5教学部

目录

做真实的自己，用良心做教育

千锋教育 QIAN FENG EDUCATION

正则表达式介绍

正则表达式介绍

正则表达式(Regular Expression)是一个描述字符模式的对象, 用于对字符串进行匹配, 一般用在有规律的字符串匹配中; 如: 匹配用户名是否正确, 邮箱是否正确等

正则表达式常用于表单验证, 如在HTML表单中填写的用户名、地址、出生日期, 邮箱等信息, 在表单提交到服务器做进一步处理之前, 我们需要先检查表单中的信息是否符合要求, 做表单验证, 以确认用户确实输入了信息并且这些信息是符合要求的.

千锋教育
QIAN FENG EDUCATION

正则表达式的使用

创建正则表达式有两种方式：

1, 使用new

```
var box = new RegExp("box"); //传入非空字符串
console.log(box);    // /box/
console.log(typeof box); //object

var box = new RegExp("box", "gi");  //第二个参数是模式修饰符
console.log(box); // /box/gi
```

2, 采用字面量方式

```
var box = /box/;
console.log(box);    // /box/
console.log(typeof box); //object

var box = /box/gi;
console.log(box); // /box/gi
```

注意: 其中g表示全局匹配，i表示忽略大小写

千锋教育
QIAN FENG EDUCATION

正则表达式的使用

1, 使用正则表达式匹配字符串有两种方式:

1, test() : 返回true则符合, false则不符合

2, exec() : 返回数组则符合, null则不符合

这两种方法使用正则表达式对象去调用, 参数为要匹配的字符串

```
var box = /box/gi;
var str = "This is a Box bOX box";
console.log(box.test(str));
console.log(/box/gi.test(str));

var box = /box/gi;
var str = "This is a Box boX"
console.log(box.exec(str));
console.log(/box/gi.exec(str));
```

千锋教育
QIAN FENG EDUCATION

正则表达式的使用

2, 字符串的正则表达式方法

除了 test()和 exec()方法，String 对象也提供了 4 个使用正则表达式的方法。

```
var str = "This is a Box box BoX";
var matchArr = str.match(/box/gi);
console.log(matchArr); //返回数组或null

//查找并替换, 返回替换后的新字符串
var replaceStr = str.replace(/box/gi, "xxx");
console.log(replaceStr);

//查找并返回匹配的字符串的起始位置,找不到匹配的则返回-1
var searchIndex = str.search(/box/i);
console.log(searchIndex);

//根据指定字符串拆分, 返回拆分后的数组, 否则返回原字符串
var splitArr = str.split(/b/i);
console.log(splitArr.length);
```

千锋教育
QIAN FENG EDUCATION

正则表达式的使用

```
//.号元字符, 代表除了换行之外的所有单个字符
var pattern = /g..gle/;   //一个点.匹配一个任意的字符
var str = "goagle";
console.log(pattern.test(str));

//*号元字符, 配合其他字符使用, 允许其他字符出现任意多次
// 重复多次匹配, 可以出现任意次,
var  pattern = /g.*gle/; //.* 匹配0到多个字符
var  str = "google"
console.log(pattern.test(str));

// [ ] : 表示字符可以出现的范围
//[a-z]*表示任意0到多个a-z的字母
var pattern = /g[a-z]*gle/;
var str = "google";
console.log(pattern.test(str));
```

千锋教育
QIAN FENG EDUCATION

正则表达式的使用

```
//非字符: ^
var pattern = /g[^0-9]*gle/; //可以有任意多个非0-9的字符
var str = "google";
console.log(pattern.test(str));

//+ 表示至少出现1次
//[A-Z]+: 至少出现一个A-Z的字符
var pattern = /[a-z][A-Z]+/;
var str = "gooGle";
console.log(pattern.test(str));

//使用元符号匹配
//\w* :匹配任意多个数字字母下划线，\w：等价于[a-zA-Z0-9_]
var pattern = /g\w*gle/;
var str = "gooA3gle";
console.log(pattern.test(str));
```

千锋教育
QIAN FENG EDUCATION

正则表达式的使用

```
//\d 代表数字, 等价于 [0-9]
//\d* 表示任意多个数字
var pattern = /g\d*gle/;
var str = "g3243gle";
console.log(pattern.test(str));

//\D: 匹配非数字, 相当于[^0-9]
var pattern = /g\Dgle/;
var str = "ga3gle";
console.log(pattern.test(str));

//\D{7,}: 匹配至少7个非数字, 相当于[^0-9]{7,}
var pattern = /\D{7,}/;
var str = "g3243gle";
console.log(pattern.test(str));
```

千锋教育
QIAN FENG EDUCATION

正则表达式的使用

```
//使用锚元字符
// /^ 匹配开始,从头开始匹配
// $/ 匹配结尾,从结尾开始匹配
var pattern = /^google$/;
var str = "google";
console.log(pattern.test(str));

// \s 匹配空格
var pattern = /goo\sgle/;
var str = "goo gle";
console.log(pattern.test(str));
```

千锋教育
QIAN FENG EDUCATION

正则表达式的使用

```
//使用或模式匹配: |
// | 代表或者的意思, 匹配其中一种字符串
var pattern = /google|baidu|bing/;  // | : 匹配三个中的其中一个字符串
var str = "googl2e bai3du;bingl"
console.log(pattern.test(str));

//分组模式匹配: ()
// ()加上小括号, 将内容进行分组, 可以作为一个整体进行多次匹配
var pattern = /(google){4,8}/;   //匹配分组中的字符出现4-8次
var str = "googlegooglegooglegoogle"
console.log(pattern.test(str));

//获取8..8之间的任意字符
var pattern = /8(.*)8/;
var str = "this is 8google8 baab 8ggg8";
console.log(str.match(pattern));
console.log(RegExp.$1); //"google8 baab 8ggg"
```

千锋教育
QIAN FENG EDUCATION

正则表达式的使用

使用exec返回数组

//exec 方法将匹配到的内容返回, 并将()分组的内容也放入数组返回

```
//以字母开头,至少一个, 忽略大小写
var pattern = /^[a-z]+/i;
var str = "google 2016";
console.log(pattern.exec(str)); //"google"
```

```
// /^[a-z]+\s[0-9]{4}$/i : 表示以字母开头, 至少有一个字母,有一个空格,和4个数字,并以数字结尾,忽略大小写
var pattern = /^[a-z]+\s[0-9]{4}$/i;
var str = "google 2016";
console.log(pattern.exec(str)); //"google 2016"
```

千锋教育
QIAN FENG EDUCATION

正则表达式的使用

```
//使用分组
var pattern = /^([a-z]+)\s([0-9]{4})$/i;
var str = "google 2016";
console.log(pattern.exec(str));     //"google 2016,google,2016"
console.log(pattern.exec(str)[0]); //google 2016
console.log(pattern.exec(str)[1]); //google  RegExp.$1
console.log(pattern.exec(str)[2]); //2016    RegExp.$2
console.log(RegExp.$1);  //google  RegExp.$1
console.log(RegExp.$2);  //2016  RegExp.$1
```

千锋教育
QIAN FENG EDUCATION

正则表达式的使用

```
//捕获性分组 和非捕获性分组
//捕获性分组
var pattern = /(\d+)([a-z])/;    //一个或多个数字,和一个a-z的字母
var str = "123abc";
console.log(pattern.exec(str)); //"123a,123,a"

//非捕获性分组(添加?:后不会捕获第二个括号中的内容)
var pattern = /(\d+)(?:[a-z])/;
var str = "123abc";
console.log(pattern.exec(str)); //"123a,123"
```

千锋教育
QIAN FENG EDUCATION

正则表达式的使用

```
//使用特殊字符匹配
// \: 转义符,可以将本来有语义的字符没有语义, 这里的.和[]都是字符, 不代表任何正则匹配的语义
var pattern = /\.\[\/b\]/;
var str = ".[/b]"
console.log(pattern.test(str));

//使用换行模式
//m: 换行模式, 换行后又重新开始匹配
// /^表示以数字开头
var pattern = /^\d+/mg;
var str = "1,baidu\n2,google\n3,bing";
console.log(str.replace(pattern, "#"));
```

千锋教育
QIAN FENG EDUCATION

常用正则表达式

邮政编码(共6位数字, 第一位不能为0)

```
var pattern = /^[1-9]\d{5}$/;
var str = "518000"
console.log(pattern.test(str));
```

电子邮件(xxxx@xxx(.xxx)+)

```
var str = "zh.an.san@1000phone.com"
var pattern = /^(\w)+(\.\w+)*@(\w)+((\.\w+)+)$/;
console.log(pattern.test(str));
```

手机号(13或14或15或18开头的11位数字)

```
var pattern = /^((13[0-9])|(14[5|7])(15([0-3]|[5-9]))|(18[05-9]))\d{8}$/;
var str = "18676273422";
console.log(pattern.test(str));
```

用户名(只能使用数字字母下划线, 且数字不能开头, 长度在6-15位)

```
var pattern = /^[a-zA-Z_]\w{5,14}$/;
```

千锋教育
QIAN FENG EDUCATION

常用正则表达式

删除多余空格

```
var str = "   zhang  san";
var pattern = /\s+/g;
str.replace(pattern,  “”  );  //zhangsan
```

删除首尾空格

```
str.replace(/^\s+/, "");
str.replace(/\s+$/, "");
var str = "  Li    ss  ";
console.log(str.replace(/^\s+/, ""));
console.log(str.replace(/\s+$/, ""));
```

身份证

```
var pattern = /^\d{17}(\d|X)$/;
var str = "42517372848273771X";
console.log(pattern.test(str));
```

千锋教育
QIAN FENG EDUCATION

常用正则表达式

中文

```
var str = "张三";
var pattern = /[\u4e00-\u9fa5]+/;
pattern .test(str);
```

简单日期格式

```
 var str = "2017-11-11";
var pattern = /^\d{4}\-\d{1,2}\-\d{1,2}$/;
pattern .test(str);
```

图片文件格式

```
var str = "aa.jpg";
var pattern = /\.(jpg|jpeg|gif)$/i;
pattern .test(str);
```

千锋教育
QIAN FENG EDUCATION

练习

1, 找出下面字符串中的bag,beg,big,bog, 忽略大小写, 并将其改为bug:

(I am a Big man, I have so mach bag, so veryone call me beg man, bog bog bog, I hate you!)

2, 假设有一个多字符的片断重复出现，把"really"、"really really "，以及任意数量连续出现的"really"字符串换成一个简单的"very ”

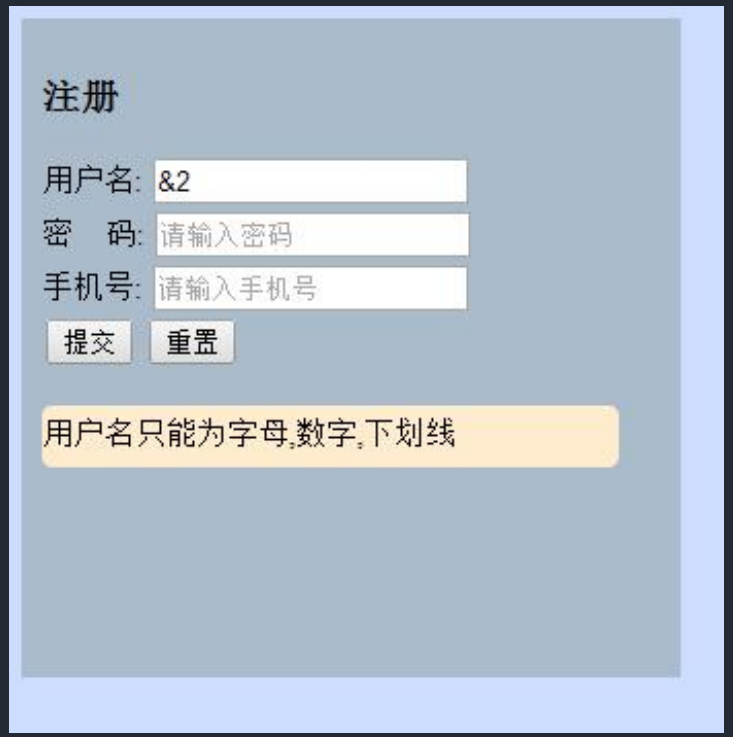
(Billy tried really hard Sally tried really really hard Timmy tried really really really hard Johnny tried really really really really hard)

千锋教育
QIAN FENG EDUCATION

练习

3, 有以下表单, 验证用户名, 密码, 手机号

- 用户名只包含数字,字母,下划线, 且长度不小于6位
- 密码长度在8到16位
- 手机号要合法



千锋教育
QIAN FENG EDUCATION

THANK YOU

千锋教育
QIAN FENG EDUCATION

做真实的自己，用良心做教育