正则表达式

1.正则：正确的规则

\* 什么是正则表达式

\* 为什么要使用正则表达式

　　正则表达式(regular expression)是一个描述字符规则的对象。如：只能出现字母，只能出现数字，前三个必须是数字等。

　　前端往往有大量的表单数据校验的工作，采用正则表达式会使得数据校验的工作量大大减轻，如邮箱验证，手机号码，等等。比起用字符串的函数来判断简单，易用。

正则表达式的组成：普通字符 或者 特殊字符

定义:

 1.字面量

var reg = /普通字符或者特殊字符/修饰符

 2.构造方法

 var reg = new RegExp("a");

 console.log(reg.test("a"));

 var reg = new RegExp("\\d");

 console.log(reg.test("1"));

正则中的特殊字符:

        单个字符:

        ^：正则开始

        $  : 正则结束

        .  : 元字符, 表示任意一个字符

        \. : 表示转义字符       \.表示.

        +： 表示其前面紧挨着的字符至少出现1次 等价{1,}

        \* ：表示其前面出现的字符至少出现过0次  等价{0,}

        ?:  表示其前面出现的字符至少出现过0次，至多1次  等价{0,1}

    | : 表示或者

        组合字符:

    \d : 0-9之间的任意一个数字 \d只占一个位置

    \D : 除了\d

    \w : 数字，字母 ，下划线 0-9 a-z A-Z \_

    \W : 除了\w

    \s : 空格或者空白等

    \S : 除了\s

        括号:

       {m,n}表示括号前面紧挨着的字符至少出现m个，至多出现n个 : 以b开头  至少3个a  至多5个a       /^ba{3,5}&/

    {m}表示括号前面紧挨着的字符只能出现m个

    {m,}表示括号前面紧挨着的字符至少出现m个

    [] 表示括号内的任意一个字符

    [wd3h]

    [a-z]表示任意一个小写字母 [a-zA-z0-9]

    [^  ]表示非括号内的任意一个字符

    ()一般与或连用 表示优先级

    [\u4e00-\u9fa5] 任意一个中文字符

<!DOCTYPE html>

<html>

    <head>

        <meta charset="utf-8" />

        <title></title>

    </head>

    <body>

        <input type="text" name="" id="txt" value="" />

        <button >点击</button>

    </body>

</html>

<script>

    var t = document.getElementById("txt");

    var b = document.getElementsByTagName("button")[0];

    b.onclick = function(){

        var str = t.value;

        //要求 文本框输入字母a

//        var reg = /a/;//包含a

//        var reg = /^aaaa$/;//只包含一个a

//        var reg = /^ba{3,5}$/;//以b开头  至少3个a  至多5个a

//        var reg = /^5{6}$/; //  6个5 /^5{6}$/;

//        var reg = /^\d{6}$/; //邮编   6位数字

//        var reg = /^\w{6,18}$/; //定义一个 由 字母或数字 或下划线组成的 用户名   范围在6,18之间

        // 定义一个 由 字母或数字 或下划线组成的 用户名  开头不能是数字   范围在6,18之间

//        var reg = /^\D\w{5,17}$/;//^[a-zA-Z\_]\w{5,17}$/;

//        var reg =/^.{6,}$/; //定义一个 密码  至少6位

//        var reg =/^www\.baidu\.com$/; //  www.baidu.com

//        var reg =/^3\+5$/; //写一个  3+5   /^5+3$/X

//        var reg =/^1[35]\d{9}$/; //以 13 或 15 开头的手机号

//        var reg =/^1(3|5)\d{9}$/;//以 13 或 15 开头的手机号

//        var reg =/^(13|15)\d{9}$/; //以 13 或 15 开头的手机号

//        var reg =/^[\u4e00-\u9fa5]{2}$/; //两个中文

        //test()  正则表达式方法 ：  验证某个字符串是否符合某个正则规范

        if( reg.test( str ) ){

            alert("正确");

        }else{

            alert("不正确");

        }

    }

</script>

2.密码强弱判断

一类字符 是   弱         纯数字  弱              纯字母  弱                  纯 特殊字符 弱

两类字符 是   中

三类字符      强    包含数字  字母 特殊字符

<!DOCTYPE html>

<html>

    <head>

        <meta charset="utf-8">

        <title></title>

    </head>

    <body>

        <input type="text" id="txt"/>

        <input type="button" id="btn" value="验证"/>

    </body>

</html>

<script>

    var txt = document.getElementById("txt");

    var btn = document.getElementById("btn");

    //纯数字

    var regNum = /^\d+$/;

    //纯字母

    var regLetter = /^[a-zA-Z]+$/;

    //纯特殊字符

    var regChar = /^[!@#$%^]+$/;

    var \_regNum = /\d+/;

    var \_regLetter = /[a-zA-Z]+/;

    var \_regChar = /[!@#$%^]+/;

    btn.onclick = function(){

        var str = txt.value;

        if(regNum.test(str) || regLetter.test(str) || regChar.test(str)){

            alert("弱");

        }else if(\_regNum.test(str) && \_regLetter.test(str) && \_regChar.test(str)){

            alert("强");

        }else{

            alert("中");

        }

    }

</script>

3.表单验证

<!DOCTYPE html>

<html>

    <head>

        <meta charset="utf-8">

        <title></title>

    </head>

    <body>

         //action提交的位置   method提交的方式

        <form action="ok.html" method="get" id="myForm">

            姓名:<input type="text" name="username" id="name"/></br>

            密码:<input type="password" name="pwd" id="pwd"/></br>

            <input type="submit" name="" id="" value="提交" />

        </form>

    </body>

</html>

<script>

    var myForm = document.getElementById("myForm");

        //表单提交的事件   该方法返回值为boolen类型   true为提交 false为不提交

    myForm.onsubmit = function(){

        //验证每一个表单项是否满足正则 只要有一个不满足 直接返回false

        //验证用户名

        var name = document.getElementById("name");

        var pwd = document.getElementById("pwd");

        var regName = /^dahuang$/;

        var strName = name.value;

        if(!regName.test(strName)){

            alert("用户名不合法");

            return false;

        }

        //验证密码

        var regPwd = /^123666$/;

        var strPwd = pwd.value;

        if(!regPwd.test(strPwd)){

            alert("密码不合法");

            return false;

        }

        return true;

    }

</script>

4.表单验证强化版：焦点验证

通常一个表单内容特别多，如果每次全部填写完成才能表单提交，这样对用户的体验非常不好。

<!DOCTYPE html>

<html>

    <head>

        <meta charset="utf-8">

        <title></title>

    </head>

    <body>

        <form id="\_form" action="ok.html">

            <input type="text" id="\_name"/></br>

            <input type="password" id="\_pwd"/></br>

            <input type="submit"></br>

        </form>

    </body>

</html>

<script>

    var userName = document.getElementById("\_name");

    var userPwd = document.getElementById("\_pwd");

    var userform = document.getElementById("\_form");

    var flagName = null;

    var flagPwd = null;

    userName.onblur = function(){

        var str = this.value;

        var regName = /^dahuang$/;

        if(regName.test(str)){

            flagName = true;

        }else{

            alert("姓名不合法");

            flagName = false;

        }

    }

    userPwd.onblur = function(){

        var str = this.value;

        var regPwd = /^123666$/;

        if(regPwd.test(str)){

            flagPwd = true;

        }else{

            alert("密码不合法");

            flagPwd = false;

        }

    }

    userform.onsubmit = function(){

        if(flagName && flagPwd){

            return true;

        }else{

            return false;

        }

    }

</script>

5.正则中的相关方法及属性

正则中的相关方法:

test() 检验一个字符串是否符合某一个正则规范，如果符合返回true,否则返回false

用法:  reg.test(str)

exec 根据正则表达式查找，结果满足返回一个长度为1的数组（数组只有一个值）

用法:reg.exec(str)

var reg = /\d/;///\d/g

    var str = "123456789";

    //var arr = reg.exec(str);

    console.log(reg.exec(str)[0]);

    console.log(reg.exec(str)[0]);

    console.log(reg.exec(str)[0]);

    console.log(reg.exec(str)[0]);

lastIndex ：这个属性记录的就是下一次捕获从哪个索引开始。

    var reg = /\d/g;

    var str = "92a345a678a9";

    console.log(reg.exec(str));

    console.log(reg.lastIndex);

    console.log(reg.exec(str));

    console.log(reg.lastIndex);

    console.log(reg.exec(str));

    console.log(reg.lastIndex);

    console.log(reg.exec(str));

    console.log(reg.lastIndex);

g---------------全局

i---------------忽略字母大小写

----------------------------------------------------------------------------------------------

正则中相关字符串的方法:

search方法  返回与正则表达式查找内容匹配的第一个子字符串的位置

用法:str.search(reg)

说明：如果找到则返回子字符至开始处的偏移位，否则返回-1。

    var r, re; // 声明变量。

    var s = "The rain in Spain falls mainly in the plain.";

    re = /falls/i; // 创建正则表达式模式。

    re2 = /tom/i;

    r = s.search(re); // 查找字符串。

    r2 = s.search(re2);

    document.write("r：" + r + "；r2：" + r2);// 返回 Boolean 结果。

match 方法

使用正则表达式模式对字符串执行查找，并将包含查找的结果作为数组返回。

str.match(rgExp)

    var reg = /\d+/g;

    var str = "123a4356a987";

    console.log(str.match(reg));

replace 方法

返回根据正则表达式进行文字替换后的字符串的复制。

stringObj.replace(rgExp, replaceText)

    var reg = /laowang/g;

    var str = "laowang的隔壁也住着laowang";

    console.log(str.replace(reg,"dahuang"));

作业:

1.正则的练习

只能用数字开头，长度在6-18位之间

以字母开头，数字结尾，中间任意一个字符

密码不能少于6位的字符

以a开头   b字符至少出现2个，至多出现6个

变量的命名正则表达式(不能用数字开头 由字母、数字、下划线 、$组成)

以010开头的座机号(后面是8位数字)

手机号以13开头，以8结尾

密码只能用6个

第一位是数字，第二位是A或a，后面至多出现6个字符

第一位是数字，第二位是任意一个字符,后面只能由字母、数字、下划线组成，共8位

写出中国人姓名正则    2--4个中文

写一个qq号的正则，至少5位  至多12位数字

邮编检验 共6位数字 第一位不能是0

检验压缩包  xxx.zip或xxx.rar或xxx.tar 三个格式

电子邮件 (@符号前后使用单词字符即可)   232323@qq.com

手机号 1 3|5|8|7

身份证 （18位  考虑最后一位可能为x）

账户名只能使用数字字母下划线，不能数字开头，长度在6--18之间

2.综合表单验证