

# 20230922

---

## 1.正则表达式

---

定义：使用一种由规则的字符串 来检验 目标字符串是否匹配

特点：可以检查是否包含，以及特点类型字符的位置

曾泽表达式也是一种特使的字符，该字符可以转换为某种算法，使用该算法来匹配文本

正则表达式与js无关，但可以在js中调用

使用场景：

数据验证，验证email，往事，电话，出生年月，字符串长度.....合法性/正确性

替换文本：可以识别特定字符，删除或替换该字符

提取子字符串：如网址中获取域名

相关对象：RegExp对象--用于执行或预编译正则表达式对象（js内置对象）

## 创建正则表达式

### 正则表达式字符

\d数字

\D非数字

\w字母数字下划线

\W非字母数字下划线

范围：

/[0-5]/

选择匹配：

/123|321/

数量匹配：

n+ 匹配至少一个n

n\*匹配0个或多个n

n? 匹配0个或一个n

n{x} 匹配x个n

n{x, y}匹配x到y个n

n{x, }匹配至少x个n

边界匹配

^n

n\$

## 正则表达式的其他方法

校验格式：

校验结果 = 正则对象.test (待校验的字符串)

### 另一种校验格式：match

校验结果=待校验的字符串.match (正则表达式)

### 文本搜索：search ()

格式：

带校验的字符串.search

### 文本替换

replace ()

格式：

代替换的字符串.replace (正则表达式, 替的文本) ;

作用：使用替换的文本 代替源字符串中 符合正则表达式的字符

case:

1.手机号码验证

2.保留三位小数验证

3.长度6-8位包含字母和数字

```
<input id="id_input_01"><span>先输入手机号</span>
<script>
    var ipt = document.getElementById("id_input_01");

    ipt.addEventListener("change", function () {
        var str = ipt.value;
        var reg = /^1[35689]\d{9}$/;
        document.getElementsByTagName("span")[0].innerText = reg.test(str)? "手机
号正确": "有误";
    })
</script>
```

```
<input id="id_input_01"><span>先输入一串小数</span>
<script>
    var ipt = document.getElementById("id_input_01");

    ipt.addEventListener("change", function () {
        var str = ipt.value;
        var reg = /^(([1-9]\d?|0)\.\d{3})/;
        document.getElementsByTagName("span")[0].innerText = reg.test(str)? "正
确": "有误";
    })
</script>
```

## 表单验证

js可用来在数据被送往服务器之前, 对html表单中的数据进行验证(是否为空?两次密码是否一致?输入是否满足条件?)

onsubmit: 表单提交事件

特点:

表单提交时触发(提交之前触发)

可以通过return+方法返回值 true/false 来控制表单是否能提交

## 本地存储

同一个变量在不同html之间不能互通传值,如果想实现这个功能就需要本地存储.

本地存储: 保存前端页面数据到浏览器中

### 1 sessionStorage

会话存储/临时存储: 用于临时保存当前页面的数据 当关闭页面/关闭窗口/关闭浏览器 数据会被删除

方法:

存: setItem(key,value); key 为该存储数据起的名字 value需要保存的数据

取: getItem(key); 根据key值 获得value;

```
<body>
  <div id="id_div_01">

  </div>
<script>
  var count = sessionStorage.getItem("counts");
  if(count == 0 || count == null){
    count = 1;
  }else {
    ++count;
  }
  sessionStorage.setItem("counts",count);
  document.getElementById("id_div_01").innerText = count
</script>
</body>
```

### 2 localStorage

本地存储: 用于长久保存页面数据. 保存的数据没有过期时间, 直到永久删除

方法: 同sessionStorage一样

可以永远计数, 直到清除浏览数据

### 3 cookie存储

cookie是由服务端(访问的网站) 创建的文件, 用于存储浏览信息, 如个人资料

cookie必须基于服务器环境运行

写入格式:

第一种

`document.cookie = "key = value(字符串类型的键值对)"`,自动将新写入的cookie键值对, 以;分割

第二种cookie(设置过期时间)

`key=value;expires=GMT时间格式的失效时间`

读取格式:

`var a = document.cookie`