如果不知道问什么: 谁问的越多！谁培训费花的越值！

1. 笔记中或画图中那句话或者那个位置不明白，立刻问！

2. 案例中哪行代码或注释不明白，立刻问！

3. 课下自己做项目，功能不会做，立刻问！

4. 刷笔试题时遇到不会的笔试题，立刻问！——公众号: 前端大全 定期总结大厂笔试题

5. 看小程序问题清单时，遇到看不懂的问题，立刻问！

正课:

1. String类型提供的正则相关函数:

2. RegExp对象

一. String类型提供的正则相关函数:

1. 查找敏感词: 4种需求:

(1). 查找一个固定的敏感词出现的位置: var i=str.indexOf("敏感词",fromi)

找不到返回-1

(2). 用正则查找第一个敏感词出现的位置: var i=str.search(/正则/i)

找不到返回-1

问题: 只能返回位置！无法返回敏感词内容

(3). 返回敏感词的内容: 2种:

a. 只返回第一个敏感词的内容和位置:

1). var arr=str.match(/正则/i)

2). 意为: 在str中查找第一个符合正则要求的敏感词的内容和位置

3). 返回值:

i. 如果找到，返回一个数组:

arr=[

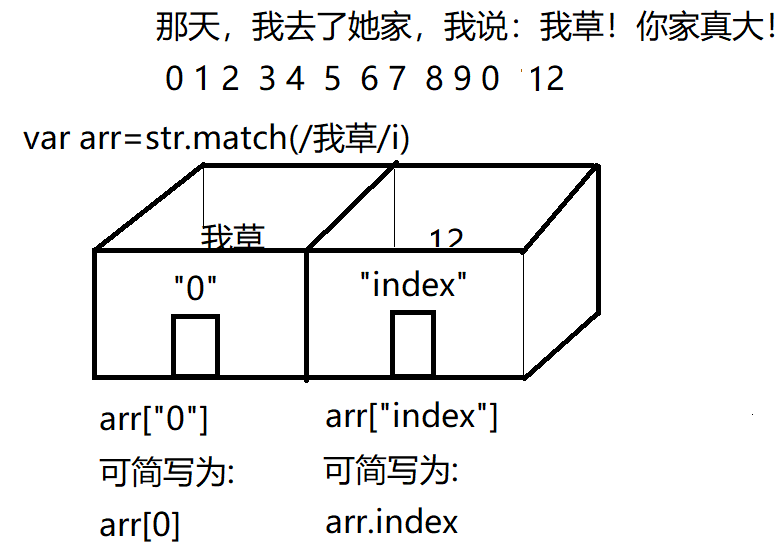
"0": "敏感词内容" ,

"index": 敏感词的位置i

]

如果获取本次找到的敏感词的内容: arr["0"] 可简写为 arr[0]

如果获取本次找到的敏感词的位置: arr["index"] 可简写为 arr.index



ii. 如果找不到: 返回null

4). 如果arr!=null，说明找到了！如果==null，说明没找到！

5). 示例: 使用match，既查找一个敏感词的内容，又查找一个敏感词的位置

|  |
| --- |
| //请用户输入一条消息内容  var input=prompt("请输入消息内容");  //查找消息中是否包含“我草”  var arr=input.match(/([我卧]|wo)\s\*([艹草槽]|cao)/i);  if(arr!=null){//如果找到    //就在网页上输出"包含敏感词，禁止发送"  //(向)网页(中)写(一行文本或一行HTML代码)    //document.write(`<h1 style="color:#555">在位置${arr["index"]}发现敏感词"${arr["0"]}",禁止发送</h1>`);    document.write(`<h1 style="color:#555">在位置${arr.index}发现敏感词"${arr[0]}",禁止发送</h1>`);  }else{//否则    //就在网页上输出"然哥说: xxxx"    document.write(`<h1 style="color:green">然哥说: ${input}</h1>`)  } |

6). 问题: 只能找出第一个敏感词，无法找所有敏感词

b. 查找所有敏感词的内容(不含位置):

1). var arr=str.match(/正则/ig) //g 意为global 全部 i忽略大小写

2). 意为: 在字符串str中查找所有符合正则要求的敏感词，放入数组中返回

3). 返回值:

i. 如果找到，返回包含本次找到的所有敏感词的内容，但是不再包含位置:

arr=[ "敏感词1", "敏感词2", ... ]

ii. 如果找不到: null

4). 如果arr!=null，说明找到了！如果==null，说明没找到！

5). 示例: 使用match加g，查找字符串中所有敏感词

|  |
| --- |
| var str="老师:请用 小红 我的 朋友 造句。小亮:小红是我的朋友。小然:朋友！小红是我的！";  //想找出所有以小字开头的人名  var arr1=str.match(/小[\u4e00-\u9fa5]/);  console.log(arr1);  var arr2=str.match(/小[\u4e00-\u9fa5]/g);  console.log(arr2); |

c. match的问题:

1). 不加g，只能找一个，但是既能获得内容，又能获得位置

2). 加g，可以找所有，但是不知道位置了！

(4). 既查找每个敏感词的内容，又查找每个敏感词的位置！

String类型没有提供该函数

只能用后边要学的reg.exec()函数

总结:

1. 凡是只返回一个下标i值的函数，如果找不到，返回-1

2. 凡是返回一个数组的函数，如果找不到，返回null

强调: 如果一个函数有可能返回null，则使用函数的返回值时必须加着12分的小心！！！

null是造成错误最多的根源之一。因为null后什么都不能加，加()报错，加.也报错。

解决: 凡是有可能返回null的函数的返回值，都要先判断不是null，才能正常使用，否则，必须为null的情况提供专门的解决办法.

补: js中数组的本质:

|  |  |
| --- | --- |
| 第一阶段说: 数组分为索引数组和关联数组  索引数组: 下标都是数字的数组，比如: [ 1,2,3 ] => [ 0:1, 1: 2 , 2:3 ]  关联数组: 下标都是自定义的字符串名称的数组, 比如:  var ym=[];  ym["数学"]=89;  ym["语文"]=69;  ym["英语"]=91;  其实: 其实js中一切数组低层都是关联数组！  所有的下标，不管是数字，还是字符串，低层其实都是字符串  arr=[ 1,2,3 ] => [ "0":1, "1": 2 , "2":3 ]  var ym=[]; ym["数学"]=89; ym["语文"]=69; ym["英语"]=91;  [ "数学":89, "语文":69, "英语":91 ]  所有数组都可用 数组名["房间号"] 来取出某个位置上的值  比如: arr["0"] => 1 ym["数学"] => 89  但是因为""很麻烦，所以可以简写:  ym["数学"] 可简写为 ym.数学  arr["0"] 可简写为 arr[0] ~~arr.0 和程序中的小数点冲突了！~~  示例: 比较所谓的索引数组和关联数组的结构和用法   |  | | --- | | var arr=[ 1,2,3 ];  var ym=[];  ym["数学"]=89;  ym["语文"]=69;  ym["英语"]=91;  console.log(arr);  console.log(ym);  console.log(arr["0"], arr[0]);  //             1         1  console.log(ym["数学"], ym.数学);  //               89      89  console.log(arr["length"],arr.length);  //                3            3  console.log(ym["length"],ym.length);  //                0          0  //因为length只统计数字下标的个数 | |

2. 替换敏感词: 2种:

(1). 简单替换: 将所有找到的敏感词都替换为统一的新值！

a. str=str.replace(/正则/ig, "新值")

b. 取出原字符串str中的内容，替换其中所有符合正则要求的敏感词为统一的新值后，将替换后的新字符串，返回，并保存到str变量中。——才能接住替换后的新字符串

c. 示例: 替换字符串中所有敏感词为统一的新值

|  |
| --- |
| var str="老师:请用 小红 我的 朋友 造句。小亮:小红是我的朋友。小然:朋友！小红是我的！";  // var str="you can you up,no can no bibi";  //想找出所有以小字开头的人名  var arr=str.match(/小[\u4e00-\u9fa5]/g);  console.log(arr);  //想把所有小字开头的人名替换为\*\*  str=str.replace(/小[\u4e00-\u9fa5]/g,"\*\*");  console.log(str);  //一般同学:  // if(arr!=null){  //   console.log(`共替换${arr.length}处`);  // }else{  //   console.log(`共替换0处`);  // }  //基础好的同学:  console.log(`共替换${arr==null?0:arr.length}处`);  var arr=[12,2,123,32,23,1,3];  arr.sort(function(a,b){return a-b});  console.log(arr); |

中午必须看的视频:

小程序->在线->JSCORE->day02 3. 回调函数 callback 亮哥 带饭

(2). 高级替换: 根据每次找到的敏感词的不同，动态选择不同的新值替换！

补:

|  |
| --- |
| 大多数编程语言中: 字符串都是不可变类型！  什么是不可变类型: 一旦创建，值不可改变，只能整体替换  什么是可变类型: 创建后，值还可随时修改的数据类型。  正是因为字符串是不可变类型，所以所有字符串类型提供的函数，都无权修改原字符串！只能返回加工后的新字符串，再用变量接住！  但是因为数组是可变类型，所有几乎所有数组类型提供的函数，都可直接修改原数组！无需用变量接住！ |

3. 切割

总结:

1. 正则:

(1). 一个敏感词正文就是一个固定的写死的规则

(2). 如果规则中一位字符有多种备选字时: 字符集 []

(3). 如果字符集中部分备选字符是连续的: -

(4). 常用字符集:

a. 一位数字: [0-9] 或 \d

b. 一位小写字母: [a-z]

c. 一位大写字母: [A-Z]

d. 一位字母(大小写都行): [A-Za-z]

e. 一位字母或数字都行: [0-9A-Za-z] ~~不能用\w~~

f. 一位字母数字或\_都行: [0-9A-Za-z\_] 或 \w

g. 一位汉字: [\u4e00-\u9fa5]

h. 一位空字符: \s

(5). 数量词:

a. 带有明确数量边界的量词:

(1). 必须出现n次，不能多也不能少: {n}

(2). 至少出现n次，最多m次: {n,m}

(3). 至少出现n次，多了不限: {n,}

b. 没有明确数量边界的量词:

(1). 可有可无，多了不限: \*

(2). 可有可无，最多出现一次: ?

(3). 至少出现一次，多了不限: +

(6). 选择和分组:

a. 两个子规则或多个子规则，二选一或多选一匹配即可: | 或者

b. 正则中局部子规则先匹配，再和整个正则联合匹配: () 分组

(7). 特殊位置:

a. 字符串开头: ^

b. 字符串结尾: $

c. 单词边界: \b

2. String类型提供的正则函数:

(1). 查找敏感词:

a. 查找一个固定的敏感词出现的位置: var i=str.indexOf("敏感词",fromi)

找不到返回-1

b. 用正则查找第一个敏感词出现的位置: var i=str.search(/正则/i)

找不到返回-1

作业:

1. (必须) 复习三目运算:

思维导图: javascript->分支结构

2. (必须) 看小程序视频学习回调函数:

小程序->在线->JSCORE->day02 3. 回调函数 callback 亮哥 带饭