正课:

1. String的正则函数

2. RegExp对象

1. String的正则函数:

1. 查找敏感词: 4种:

1. 查找一个固定的敏感词出现的位置

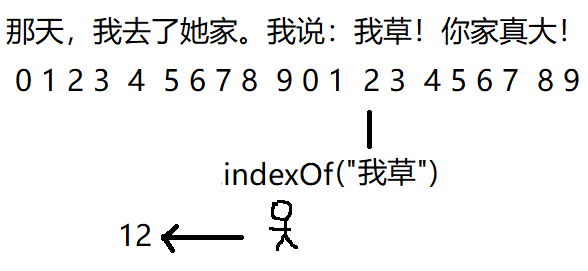
var i=str.indexOf("敏感词",fromi);

位置<-的<-"敏感词"

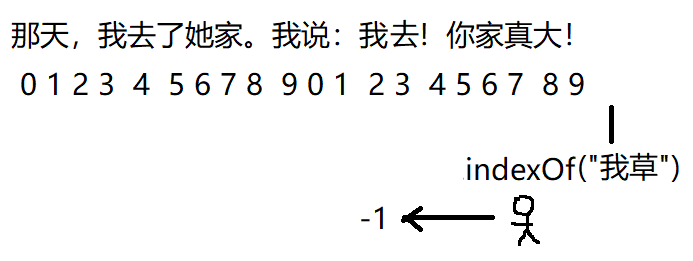
在str中查找fromi位置之后的下一个"敏感词"的位置。

如果没有给fromi参数值，则默认从开头(0位置)开始找。

返回值: 如果找到敏感词，返回敏感词第一个字在字符串中的下标位置.



如果没找到，返回-1



问题: 只能查找一个固定的敏感词，如果换成同音字或拼音，就找不到了！

2. 用正则表达式模糊查找一个敏感词的位置:

var i=str.search(/正则/i)

在str中查找第一个符合正则表达式要求的敏感词的位置。

强调: search不能指定开始查找的位置，只能从头(0位置)开始找。

返回值: 同indexOf

问题: 所有正则默认区分大小写

解决: 在第二个/后加i

问题: 只能返回敏感词的位置，不能返回敏感词的内容。

3. 获取敏感词的内容：2种:

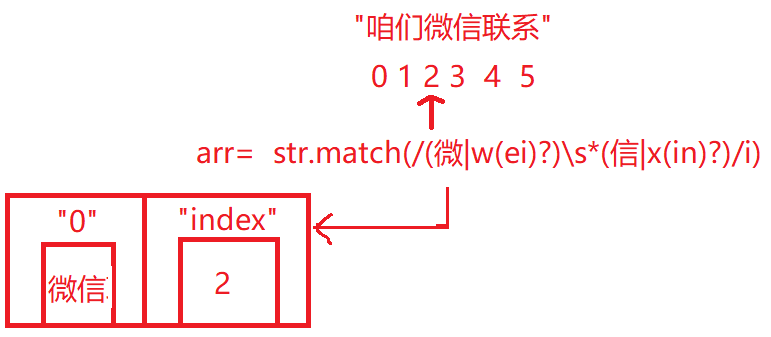
1. 只获取一个敏感词的内容和位置:

var arr=str.match(/正则/i)

匹配

在str中，查找第一个符合正则要求的敏感词的内容和位置

返回值: 如果找到敏感词，返回的是一个数组:



如果想获得本地找到的敏感词的内容：

arr[0]

如果想获得本次找到的敏感词的位置:

arr["index"]

强调: "index"不能改名，因为这个数组时match内部建立好，返回给咱们的。人家建立数组时，用的什么门牌号，咱们访问数组内容时，就要遵循人家match的要求！

如果没找到，返回null

强调: 如果一个函数有可能返回null！则使用前必须先验证不是null，才能使用。因为null之后，什么都不能加！null()报错，null.报错！

问题: 正则表达式默认只能匹配一个敏感词。匹配到之后，就下班了！

解决: 只要在/后加g，global全部

2. 查找所有敏感词的内容:

var arr=str.match(/正则/ig)

在str中查找所有符合正则要求的敏感词的内容。

问题: 如果match加了g，就只能返回敏感词的内容，无法返回位置了！

返回值: 如果找到多个敏感词:

arr: ["敏感词1", "敏感词2",... ...]

如果没找到，返回null

今后，只要只关心所有敏感词的内容，而不关心敏感词的位置时，就用match+g。

4. 即查找每个关键词的内容，又查找每个关键词的位置。

RegExp对象的exec()函数解决

2. 替换：2种:

1. 简单替换: 将所有敏感词都替换为统一的新词

str=str.replace(/正则/ig, "新词")

将str中所有符合正则要求的敏感词，都替换为指定的统一的新词。

坑: 替换后，不报错，也替换不成功！

因为: 字符串都是不可变类型！

无法对原字符串直接修改。

所有字符串函数，都只能返回修改后的新字符串。而保持原字符串不变。

解决: 要想获得新值，必须用=接住新值，并保存到变量中。

2. 高级替换: 根据每次找到的敏感词不同，动态选择不同的新词替换！

str=str.replace(

/正则/ig,

//回调函数callback:

//replace会自动在每个找到的关键词上调用一次这个回调函数。找到几个词，就反复调用几次！每次调用时，会自动将本次找到的关键词传给函数的参数keyword

//2个要求:

funtion(keyword){//1. 必须有一个参数接住本次找到的一个关键词

//2. 必须返回处理后的新词

return keyword.toUpperCase();

}

//replace会将回调函数返回的新词，替换到本次找到的关键词位置！

)

强调: 所有形参，都可以自己定义变量名。

补: 回调函数: 我们自己定义的，但是不是由我们自己调用执行。而是交给其它对象或函数去执行。

衍生: 删除敏感词，其实就是将敏感词替换为""

3. 切割:

什么是: 将一个字符串，根据指定的切割符，切割成多段子字符串。

2种:

1. 简单切割: 切割符是固定的

var arr=str.split("切割符")

将字符串str，按指定的"切割符"，切割成多段子字符串保存在数组中

2. 复杂切割: 切割符是变化的，但是有规律的。

var arr=str.split(/正则/)

将字符串str，按符合正则的切割符，切割成多段子字符串。

衍生: 打散字符串为字符数组

总结:

查找敏感词

1. 查找一个固定的敏感词: indexOf

2. 用正则模糊查找一个敏感词的位置: search

3. 用正则查找敏感词的内容

1. 只查找一个敏感词的内容和位置match

2. 查找所有敏感词的内容，但是无法获得位置

match+g

4. 即查找每个敏感词的内容，又查找每个敏感词的位置。——暂缺

替换敏感词 replace

删除 replace为""

切割 split

打散字符串为字符数组 split("")

验证格式——暂缺

2. RegExp对象:

什么是: 专门保存一条正则表达式，并提供用正则执行查找和验证方法的 对象

何时:

1. 验证

2. 高级查找

如何:

1. 创建对象: 2种:

1. 用//简化版创建:

var reg=/\b[a-z]/ig;

何时: 如果正则表达式是固定不变的！

问题: 不支持动态生成正则表达式

因为: //之间是正则表达式的地盘！写js！人家不认识！

2. 用new创建

var reg=new RegExp("\b[a-z]","ig")

何时: 如果正则表达式需要根据变量或数组内容，动态生成！

因为: 这种方法用js认识的""代替了//。而在js中，有很多种办法。灵活的拼出想要的字符串

2. RegExp提供的函数:

1. 验证格式：

var bool=reg.test(str)

用reg规则，检测str是否符合规则要求。

返回值: true/false

坑: test默认只要在str中找到部分和reg匹配，就返回true！而不要求从头到尾完全匹配！

解决: 今后凡是验证，都要求从头到尾必须完整匹配：前加^，同时后加$，意为从头到尾!

作业:

1. 定义函数去掉字符串开头或结尾的空字符！