正课:

内置对象/包装类型

\*\*\*String

\*\*\*\*正则表达式

内置对象:

什么是: ES标准中规定的，浏览器已经实现的对象

包括: 11个:

String Number Boolean ——包装类型

Array Date RegExp Math

Error

Function Object

Global——在浏览器中被window替换

包装类型:

什么是: 专门封装原始类型的值并提供操作原始类型值的API

为什么: 原始类型的值本身不具有任何的功能

才需要包装类型对象的帮助，完成功能

何时:只要试图用原始类型的值调用函数时，都会自动创建对应包装类型的对象，封装要操作的值，并使用函数执行功能。

如何: 不用手动使用

比如: n.toFixed(2) => new Number(n).toFixed(2);

"张".charCodeAt() => new String("张").charCodeAt();

\*\*\*String

什么是: 由多个字符组成的只读字符数组！

vs 数组: 相同: 1. 下标; 2. .length; 3. .slice()

不同: 类型不同: API不通用

API:

大小写转换: 将str中的字母都转为统一的小写/大写

str.toLowerCase()

str.toUpperCase()

何时: 不区分大小写时，都要先转为一致的大小写，再比较或判断

比如: 验证码, 用户名, 电子邮件

获得指定位置的字符: str.charAt(i) => str[i]

获得指定位置字符的unicode号:

str.charCodeAt(i) 获得str中i位置的unicode号

将unicode号转回字:

String.fromCharCode(unicode号)

获取子字符串: 3种:

str.slice(starti,endi+1)

str.substring(starti,endi+1) 用法同slice，只不过不支持负数参数

str.substr(starti,n) 从starti开始，获取n个字符

查找关键词: 4种:

1. 查找一个固定的关键词:

var i=str.indexOf("关键词",fromi);

在str中fromi位置之后，查找下一个"关键词"的位置

返回: 如果找到，返回关键词的下标位置i

如果没找到，返回-1

var i=str.lastIndexOf("关键词");

在str中找最后一个关键词的位置i

问题: 每次只能查找一种关键词。

解决: 正则表达式模糊匹配多种关键词:

2. 判断是否包含敏感词:

var i=str.search(/正则/i);

在str中找第一个符合正则要求的敏感词的位置！

返回值: 如果找到，返回下标位置

如果没找到，返回-1

问题: 默认正则表达式都是区分大小写！

解决: 在第二个/后加后缀i, ignore,表示忽略大小写

问题: 永远只能找第一个，无法找所有

仅返回位置，无法返回内容

3. 查找所有敏感词的内容:

var kwords=str.match(/正则/ig);

返回值: kwords中保存所有找到的敏感词

如果找不到，返回null

问题: 默认正则表达式只匹配第一个符合条件的敏感词

解决: 在第二个/后加g, global 全部

问题: 只能获得内容，无法获得每个敏感词的位置

4. 即查找每个敏感词的内容，又查找位置: regExp.exec()

替换:

什么是: 将字符串中符合规则的敏感词替换为新内容

如何: 2种:

1. 简单替换: 将所有敏感词替换为同一种新值

str=str.replace(/正则/ig,替换的新值)

强调: 所有字符串API都无权修改原字符串，只能返回新字符串

2. 高级替换: 根据敏感词的不同，动态选择替换为不同的值

str=str.replace(/正则/ig,function(kw){

kw //自动获得本次找到的敏感词

return 根据不同的kw返回不同的值

})

衍生: 删除: 将关键词替换为""

切割:

什么是: 将字符串按指定的关键词切割为多段子字符串

如何: 2种:

1. 简单: 分隔符是固定的:

var subStrs=str.split("分隔符")

返回: 多个子字符串的数组

切割后的结果中，不包含分隔符的

2. 复杂: 分隔符不是固定的:

var subStrs=str.split(/正则/)

固定套路: 将字符串打散为字符数组:

var chars=str.split("");

\*\*\*\*正则表达式:

什么是: 规定字符串中字符出现规律的规则

何时: 2大用途:

1. 通过规则，模糊查找多种关键词

2. 验证用户输入的格式

如何:

1. 关键词原文本身就是最简单的正则表达式

2. 字符集:

什么是: 规定一位字符备选字符列表的集合

何时: 只要一位字符有多种可能备选时

如何: [备选字符列表]

强调: 不要用逗号分隔

每个[]只规定一位字符的备选字

简写:

1. 如果部分备选字符的unicode是连续的，可用-省略中间字符

比如: 一位数字: [0-9]

一位小写字母:[a-z]

一位大写字母:[A-Z]

一位字母: [A-Za-z]

一位汉字:[\u4e00-\u9fa5]

2. 除了xxx:

[^xxx] 比如: 除了4和7都行!

3. 预定义字符集:

什么是: 对常用特定字符集的更简化写法:

包括: 4个:

\d => [0-9]

\w => [A-Za-z0-9\_]

\s => 空字符: 空格, Tab...

. 匹配任意字符

4. 量词:

什么是: 规定一位字符集出现次数的规则

何时: 只要规定一位字符集出现的次数

如何: 量词紧跟在修饰的字符集之后

2大类:

1. 有明确数量边界的:

字符集{n,m} 最少n个,最多m个

字符集{n,} 至少n个,多了不限

字符集{n} 必须n个

2. 没有明确数量边界:

字符集? 可有可无，最多1次

字符集\* 可有可无，多了不限

字符集+ 至少一个，多了不限

5. 选择和分组:

选择: 或

何时: 只要在多个规则中任选其一匹配时

如何: 规则1|规则2

只要满足两个规则之一即可！

分组: ()

何时: 只要希望一个量词同时修饰多个字符集时

就要将要修饰的字符集放入一个()中作为一组。再用量词修饰()

为什么: 默认一个量词只能修饰相邻的前一个字符集

比如: 身份证号: 15位数字 2位数字 1位数字或Xx

整体可有可无,最多一次

\d{15} ( \d\d [0-9Xx] ) ?

手机号:

+86或0086 可有可无,最多1次

空格 可有可无，多了不限

1

3,4,5,7,8中挑一个

9位数字

(\+86|0086)?\s\*1[34578]\d{9}

(微|w(ei)?)\s\*(信|x(in)?)

6. 指定匹配位置: 3个:

开头: ^ 比如: 匹配开头的空字符: ^\s+

结尾: $ 比如: 匹配结尾的空字符: \s+$

比如: 开头或结尾的空字符: ^\s+|\s+$

单词边界: \b 包括: 空格, 标点, 开头, 结尾

比如: 匹配单词no: \bno\b