张东

zhangdong@tedu.cn

准备:

1. 犀牛书: JavaScript权威指南

2. 微信公众号: 前端大全

3. 上届笔记/思维导图

不许记笔记！

正课:

1. String:

2. 正则表达式:

什么是字符串: 一串字符组成的字符数组

vs 数组: 相同: 1. 下标, 2. .length, 3. 遍历

4. .slice()

不同: 类型不同, API不通用

API:

所有字符串API都无权修改原字符串，只能返回新字符串

1. 大小写转换:

str.toUpperCase() 将字符串中所有小写字母转为大写

str.toLowerCase() 将字符串中所有大写字母转为小写

何时: 不区分大小写时

2. 获取指定位置字符:

str[i] => str.charAt(i)

获取指定位置字符的unicode号

var unicode=str.charCodeAt(i)

将unicode号反向转回字:

var char=String.fromCharCode(unicode)

3. 选取子字符串:

str.slice(starti,endi+1) 优点: 支持负数参数，表示倒数

str.substring(starti,endi+1)

缺点: 不支持负数参数

解决: str.length-n 代替-n

str.substr(starti,n)

4. 查找关键词: 4种:

1. 查找一个固定的关键词的位置:

var i=str.indexOf("关键词",fromi)

在str中，从fromi位置开始，找下一个"关键词"所在得位置i

如果没有fromi，默认从0位置开始

返回值: 找到的第一个关键词的第一个字的下标位置

如果找不到，返回-1

var i=str.lastIndexOf("关键词")

在str中找最后一个"关键词"的位置

问题: 只能查找一种固定的关键词

解决: 正则表达式

2. 判断是否包含符合正则的关键词:

var i=str.search(/正则/i)

在str中查找第一个符合正则要求的关键词的位置

如果找不到，返回-1

问题: 正则默认区分大小的

解决: 在第二个/后加后缀i ignore

问题: 只能找第一个

解决:

3. 查找所有关键词的内容:

var arr=str.match(/正则/i)

在str中查找符合正则表达式要求的关键词，并将内容放入arr数组中

2种:

1. 默认只找第一个, 但会返回位置

arr: [ 0: 关键词, index: 位置 , ... ]

2. 查找所有关键词的内容: 在第二个/后加后缀g

强调: 如果找不到，返回null

强调: 如果一个函数有可能返回null，都必须先判断不是null，才能使用!

问题: 只能获得关键词内容，无法获得位置

4. 即查找每个关键词的内容，又获得每个关键词的位置

reg.exec()

5. 替换关键词: 2种：

1. 简单替换: 将所有敏感词都替换为统一的新值

str=str.replace(/正则/ig, "新值");

2. 高级替换: 根据关键词的不同，动态选择不同的替换值

str=str.replace(/正则/ig,function(kw){

return 根据kw不同，动态返回新值

});

衍生操作: 删除: 将关键词替换为""

2. 正则表达式regular expression:

什么是: 规定一个字符串中字符出现规律的规则

何时: 2种场景:

1. 模糊查找多种敏感词

2. 验证格式

如何:

1. 一个关键词自身，就是一个最简单的正则表达式

2. 字符集:

什么是: 规定一位字符上多个备选字符列表的集合

何时: 只要一位字符上有多种备选字时

如何: [备选字符列表]

强调: 一个字符集只能匹配一位字符

简写: 如果备选字符列表中部分字符连续，可用-省略中间字符

常用: 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]

一位字母或数字: [0-9A-Za-z]

一位汉字: [\u4e00-\u9fa5]

除了: [^xxx]

问题: 排除之外的范围，太大

车牌号: [\u4e00-\u9fa5][A-Z]·[A-Z0-9]{5}

3. 预定义字符集: 对常用字符集的更简写

4个: \d 一位数字

\w 一位字母,数字或\_ [0-9A-Za-z\_]

\s 一位空字符: 空格, tab, ...

. 通配符

字符集的问题: 只能固定一位字符，无法灵活定义数量

解决: 量词

4. 量词:

什么是: 规定一位字符出现次数的规则

何时: 只要规定字符出现的次数时

如何: 字符集量词

强调: 一个量词默认仅修饰相邻的前一个字符集

包括: 2大类:

1. 有明确数量范围:

{m,n} 至少m个, 最多n个

{m,} 至少m个，多了不限

{m} 必须m个

2. 没有明确数量范围:

? 可有可无，最多一个

\* 可有可无，多了不限

+ 至少一个，多了不限

5. 选择和分组:

1. 选择: 或 |

规则1|规则2 只要满足两个规则中其一即可

2. 分组: (多个规则)

什么是: 将多个规则用()包裹为一组

为什么: 量词默认只修饰相邻的一个字符集

何时: 只要希望一个量词，同时修饰多个字符集时

身份证号: 15位数字 2位数字 一位数字或X

后三位整体，可有可无，最多一次

\d{15}(\d{2}[0-9X])?

手机号:

+86 或 0086 可有可无，最多一次

空字符 可有可无，多了不限

1

3,4,5,6,7,8 任选其一

9位数字

(\+86|0086)?\s\*1[3-8]\d{9}

6. 匹配特殊位置:

字符串开头: ^

字符串结尾: $

何时: 仅希望匹配开头或结尾的内容时

比如: " zhang dong "

^\s+ 仅匹配开头的空字符

\s+$ 仅匹配结尾的空字符

^\s+|\s+$ 即匹配开头，又匹配结尾的空字符

单词边界: \b 包括: 空格，标点，开头，结尾

何时: 2种:

1. 匹配一个独立的单词时 \b[a-z]+\b

2. 仅匹配单词开头或结尾的字母 \b[a-z]