张东

zhangdong@tedu.cn

准备:

1. 犀牛书: JavaScript权威指南

2. 公众号: 前端大全!

3. 上届笔记 和 思维导图

谁抄的越勤，谁学的越烂！！！

正课:

String:

1. 什么是String:

2. API:

3. \*\*\*正则表达式:

1. 什么是String: 一串字符组成的字符数组

vs 数组:

相同: 1. 下标, 2. .length, 3. for, 4. slice

不同: 类型不同, API不通用!

字符串必须用' '——ESLint

2. API: 所有字符串API无权修改原字符串，只能返回新字符串

大小写转换:

str.toLowerCase() //转为小写

str.toUpperCase() //转为大写

何时: 不区分大小写时

获取指定位置的字符:

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

获取指定位置字符的unicode号:

str.charCodeAt(i) 字->号

将unicode号转回字: String.fromCharCode(号)

选取子字符串:

str.substring(starti,endi+1) => str.slice(starti,endi+1)

差别: substring不支持负数参数

变通: str.length-n

str.substr(starti,n) 从starti开始选取n个字符

\*\*\*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"出现的位置

返回值: 关键词第一个字的位置

找不到，返回-1

var i=str.lastIndexOf("关键词")

查找str中"关键词"最后出现的位置

优: 设置开始位置，找所有关键词的位置

缺: 只能查找一个固定的关键词

2. 判断是否包含符合正则条件的敏感词:

var i=str.search(/正则/i)

查找str中查找第一个符合正则要求的关键词的位置

返回值: 关键词位置

找不到返回-1

问题: 正则默认都区分大小写

解决: 在第二个/后加i ignore

问题: 只能找一个，且只返回位置，无法返回关键词内容

3. 获取所有关键词的内容:

var kwords=str.match(/正则/ig)

查找str中所有关键词的内容

返回值: 关键词数组

如果找不到, 返回null

强调: 如果一个函数可能返回null，就必须先验证不是null，再使用!

问题: 所有正则，默认只查找第一个关键词

解决: 在第二个斜线后加g global

问题: 只能找关键词的内容，无法返回每个关键词的位置

4. 即找所有关键词的内容，又找所有关键词的位置

reg.exec()

替换:

1. 简单替换: 将所有关键词，替换为统一的新内容

str=str.replace(/正则/ig,"替换值")

更简单: 如果关键词是固定的: str.replace("关键词","新值")

2. 高级替换: 根据每个关键词的不同，动态选择不同的新值替换

str=str.replace(

/正则/ig,

function(kword){//每找到一个关键词，就自动调用该function

//kword会自动获得本次找到的一个关键词

return 根据kword的不同，返回不同的新值

}

)

切割:

3. \*\*\*正则表达式:

什么是: 描述一个字符串中字符出现规律的规则!

为什么: 有些关键词会发生变化！

何时: 2种:

1. 查找可变的关键词

2. 验证格式

如何: 正则表达式的语法:

1. 一个固定的关键词原文，也是一种最简单的规则

2. 字符集: 规定一位字符上可用的备选字符列表

何时: 只要一位字符有多种备选字

如何: [备选字列表]

简写: 如果备选字符列表连续，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]

一位字母或数字:[A-Za-z0-9]

一位汉字: [\u4e00-\u9fa5]

排除: [^xxx]

3. 预定义字符集:

对常用字符集的简化:

一位数字 \d

一位字母,数字,\_都行 \w [0-9A-Za-z\_]

一位空字符 \s 包括: 空格, tab, ...

通配符 .

问题: 字符集只能规定一位字符的备选内容

无法定义字符出现的个数

解决:

4. 量词: 专门规定一个字符集出现次数的规则

何时: 只要规定一个字符集出现次数时

如何: 字符集量词

2大类:

1. 有明确数量边界:

字符集{6,8} 至少6个，最多8个

字符集{6,} 至少6个，多了不限

字符集{6} 必须6个，不能多也不能少

2. 没有明确数量边界:

字符集? 可有可无，最多一个

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 |

规则1|规则2 在两个规则之间选其一匹配即可

何时: 只要在多个规则间选其一匹配时

分组: 将多个规则，用()包裹为一个整体

为什么: 量词默认仅修饰相邻的前一个字符集

何时: 只要一个量词，要修饰多个字符集时

如何: (规则1规则2...)量词

身份证号: 15位数字 2位数字 一位数字或X或x

后三位整体可有可无，最多一个

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

手机号: +86或0086 可有可无，最多1次

空字符 可有可无，多了不限

1

[34578]

9位数字

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

微信 wei xin w x

(微|w(ei)?)\s\*(信|x(in)?)

6. 匹配特殊位置:

三个位置:

1. 开头 ^ 比如: 开头的空字符: ^\s+

2. 结尾 $ 比如: 结尾的空字符: \s+$

何时: 如果只匹配紧挨开头或结尾的内容时

即匹配开头，又匹配结尾: ^\s+|\s+$  
 3. 单词边界 \b 包括: 开头, 结尾, 空格, 标点