张东

zhangdong@tedu.cn

正课:

1. String

2. 内置对象和包装类型

3. String API

4. 正则表达式

1. String字符串

什么是String: 一串字符组成的只读字符数组

vs 数组:

相同: 1. 下标,

2. length

固定套路:

1. 最后一个元素: arr[arr.length-1]

倒数第n个元素: arr[arr.length-n]

2. 追加一个新元素: arr[arr.length]=值;

3. 删除最后一个元素: arr.length--

清空数组: arr.length=0

3. for遍历

4. slice:

arr.slice(starti,endi+1) 含头不含尾

只知道要获取的个数:

arr.slice(starti,starti+n)

简写:

1. 负数参数: 表示倒数第n个位置

2. 省略第二个参数: 一直选取到结尾

3. 两个参数都省略: 完整复制数组

不同: 类型不同！API不通用！

2. 内置对象和包装类型:

鄙视: 请列举js中所有内置对象: 11个

什么是内置对象: ES标准中规定的，浏览器厂商已经实现的，程序员可直接使用的对象

包含: String Number Boolean ——包装类型

Array Date RegExp Math

Error

Function Object

Global(在浏览器中被window代替)

鄙视: 对包装类型的理解

什么是包装类型: 专门封装原始类型的值，并提供操作原始类型值的API 的类型

为什么: 原始类型的值，本身没有任何功能！

何时: 试图用原始类型的值调用API时，程序会自动创建包装类型对象

特点: 调用后，自动释放！

3. String API

强调: 所有String API都无权修改原字符串，只能返回新字符串！

大小写转换:

str.toUpperCase() 转为大写

str.toLowerCase() 转为小写

何时: 只要不区分大小写时，都要先转为一致的大小写，再判断和比较

比如: 验证码，用户名和邮箱地址

获得指定位置的字符:

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

str.charCodeAt(i) 获得i位置的字符的unicode号

String.fromCharCode(unicode) 将unicode转回字

选取子字符串:

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

问题:substring不支持负数参数！

解决: str.substring(starti,str.length-n+1)

如果只知道个数:

str.substr(starti,n) <==> str.slice(starti,starti+n)

强调: 只有两个参数都是下标的函数，才考虑含头不含尾。所以，substr不用考虑含头不含尾，n不用+1。

查找关键词: 4种:

1. 查找一个固定的关键词出现的位置

var i=str.indexOf("关键词",fromi)

在str中，从fromi位置开始，找下一个"关键词"的位置。

返回值: 本次找到的一个关键词的下标！

如果找不到，返回-1

如果直接找最后一个关键词:

var i=str.lastIndexOf("关键词");

问题: 只能查找一种固定的关键词

解决: 用正则表达式模糊查找多种关键词

2. 判断是否包含符合规则的敏感词:

var i=str.search(/正则表达式/i);

问题1: 所有正则表达式，默认区分大小写！

解决: 在第二个/后加i (ignore忽略)

问题2: 只能找第一个，无法找所有！

3. 查找所有关键词的内容:

var kwords=str.match(/正则/ig)

返回值: 包含所有关键词的数组

如果找不到，返回null！

问题1: 默认只找第一个

解决: 在第二个/后加g (global全部)

问题2: 只能返回内容，无法返回每个关键词的位置

4. 即找每个关键词的内容，又找每个关键词的位置:

4. 正则表达式（Regular Expression）:

什么是: 规定字符串中，字符出现规律的规则

何时: 2种:

1. 用规则模糊查找多种关键词

2. 验证字符串的格式

如何:

1. 最简单的规则，是关键词原文

2. 字符集:

什么是: 规定1位字符，备选字符列表的集合

何时: 只要字符串中1位字符，有多种备选时

如何: [备选字符列表]

强调: 1. 不是数组，不加逗号！

2. 无论备选字符列表，有多少字，一个字符集只能匹配1位字符！

当备选字符列表部分连续时，可用-省略中间的字符

常见: 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]

3. 预定义字符集: 4个

其实，就是对常用字符集的简写

\d 一位数字 [0-9]

\w 一位字母，数字或下划线 [0-9A-Za-z\_]

\s 一位空字符: 空格 换行 Tab

. 除换行外的任意字符

问题: 仅使用字符集，无法灵活定义字符的个数

4. 量词:

什么是: 专门规定一个字符集，出现次数的规则

何时: 只要固定一个字符集出现的次数，就用量词！

如何: 字符集量词

包括: 2大类:

1. 有明确数量限制:

字符集{m,n} 至少m个，最多n个

字符集{m, } 至少m个，多了不限！

字符集{m} 必须m个，不能多也不能少

比如: 手机号: 1[34578]\d{9}

QQ号: \d{4,11}

2. 没有明确数量限制:

字符集? 可有可无，最多1个

字符集\* 可有可无，数量不限

字符集+ 至少一个，多了不限

5. 选择和分组:

选择: 或 在多个规则中，匹配任意一个即可！

如何: 规则1|规则2

分组: () 将多个规则分为一组

为什么: 量词，默认只修饰相邻的前一个字符集

何时: 如果希望量词修饰多个字符集时，就用分组

如何: (多个字符集)

比如: 身份证号:

15位数字 2位数字 数字或Xx

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

后三位整体可有可无，最多一次

比如: 完整手机号:

+86或0086 整体可有可无，最多一次

空字符 可有可无，数量不限

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

比如: 微信 weixin wx

(微|w(ei)?)\s\*(信|x(in)?)

6. 匹配特殊位置: 3个位置

何时: 只要匹配字符串特殊位置上的关键词时

包括:

开头 ^，结尾 $，

比如: ^\s+ 开头的空字符

\s+$ 结尾的空字符

^\s+|\s+$ 开头的和结尾的空字符

单词边界 \b: 包括 空格，标点...

何时: 只要匹配一个专门的单词时