每个案例或小视频，至少写/看三遍

1. 代码+注释抄一遍

2. 保留注释，删掉代码，试着看着注释把代码填回来

3. 自己试着将注释+代码一起写回来——才是你自己会

肯定不能都记住，但是用到什么就复习什么，用到什么保证可以找到。——思维导图

作业:

1. 微信 正则表达式:

第一个字: (微|w(ei)?)

微 或 wei 其中ei可有可无

中间: \s\*

第二个字: (信|x(in)?)

信 或 xin 其中in可有可无

(微|w(ei)?) \s\*(信|x(in)?)

正课:

1. String类型中的正则函数:

2. RegExp对象

1. String类型中的正则函数:

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")

2. 用正则模糊查找一个敏感词的位置:

var i=str.search(/正则/i)

3. 要查找敏感词的内容: 2种:

1. 只查找第一个敏感词的内容和位置

var arr=str.match(/正则/i)

arr: [ "0": "我草", "index": 9 ]

问题: 只能找一个敏感词！

2. 查找字符串中所有符合条件的敏感词的内容:

var arr=str.match(/正则/ig) global

arr: [ "我艹", "我槽" ].length = 2

0 1

强调: 只包含关键词的内容，不再包含关键词的位置

如果找不到，返回null

强调: 如果一个函数可能返回null，就应该先验证返回值不是null，再使用！

4. 查找所有关键词的内容和位置:

reg.exec()

2. 替换: 2种:

1. 简单替换: 将所有关键词都替换为一致的新值

str=str.replace(/正则/ig, "新值");

将str中所有符合正则要求的关键词都替换为指定的"新值"

强调: 所有字符串都是不可变类型, 所有字符串的函数都无权直接修改原字符串，只能返回新字符串

所以，如果想获得新字符串，必须自己用变量接一下。

比如: str=str.replace(/小[\u4e00-\u9fa5]/,"\*\*")

//将str中 所有 小字开头的人名 都 替换成 \*\*，并将替换后的结果，手动保存回原变量中。

2. 高级替换: 根据搜到的不同敏感词替换成不同的新值

比如: 鄙视题: 将每个单词首字母改为大写！

//找到每个单词首字母: /\b[a-z]/g

//用replace将字符串中的所有首字母，替换为其对应的大写字母:

str="you can you up";

str=str.replace(//2个实参

/\b[a-z]/g, //第一个实参是要查找字符的正则表达式

//第二个实参是一个没有名字的函数

//每找到一个敏感词，就自动调用一次

//每次调用时都自动传入本次找到的敏感词给kword，比如: 本次找到“y”，那么kword=“y”

function(kword){

//1. 必须有一个形参

//2.根据kword接到的值不同，返回不同新的替换值

return y.toUpperCase() -> Y

//return出的新值，会被自动替换到当前找到的敏感词位置。

}

)

衍生: 删除关键词: 其实就是替换为空字符串

str=str.replace(/正则/ig, "");

作业: 鄙视: 删除字符串开头和结尾的空字符

3. 切割：将一个字符串按指定的分隔符，切割为多段子字符串

2种:

1. 分隔符是固定的:

var arr=str.split("分隔符")

将str按指定的分隔符切割为多段子字符串，保存在一个字符串数组中。

比如: var email="zhangdong@tedu.cn";

//将email按中间的@切割

var arr=email.split("@");

//["zhangdong","tedu.cn"]

// 0 1

var uname=arr[0];

var domain=arr[1];

2. 分隔符不确定，但有规律:

var arr=str.split(/正则/i)

比如: //再按</li>空字符<li>切割

var arr=html.split(/<\/li>\s\*<li>/i);

2. RegExp对象:

什么是: 保存一条正则表达式，并包含用正则表达式执行验证和查找功能 的对象

何时: 只要在js中使用正则表达式，都要先创建正则表达式对象，再使用正则。

如何:

1. 创建正则表达式对象:

2种:

1. 创建一个固定的正则表达式对象:

var reg= /正则/ig

问题: //之间不允许写js语句动态生成正则表达式

2. 正则表达式需要动态生成

var reg=new RegExp("正则","ig")

说明: 因为RegExp的第一个形参要求接受一个字符串格式的正则表达式。所以，我们有无数种办法，拼接出我们想要的任何字符串。

比如: var str=kwords.join("|");

"明月|白鹭"

var reg=new RegExp(str);

/明月|白鹭/

2. RegExp对象中包含两个函数:

1. 验证字符串是否符合格式要求

var bool=reg.test(str)

检查

用正则表达式reg，检查str字符串是否符合格式要求

返回值: true/false

坑: test()默认只要能找到符合条件的部分内容，就返回true

解决: 今后凡是验证，都必须前加^，同时后加$

比如: //定义手机号的规则表达式reg

var reg=/^1[3-9]\d{9}$/;

//用规则表达式去验证手机号是否符合格式要求

var result=reg.test(phone);

//如果符合要求

if(result==true){

... ...

2. 第四种查找: 查找所有敏感词的内容和位置