

Lab6

李晓伟 19302010006

1.

设置 num=1 count 计数器=0 初始值

if 的判定：剩余的秒数小于 5 直接终止循环 或者 判断计数器 大于 9 次也中止循环

2.

###telephone 部分

1 表示首位为 1

(3|4|5|6|7|8|9)表示第二位为 3, 4, 5, 6, 7, 8, 9 中的一位

\d{9}表示后九位是 0-9 中任意数字

###mail 部分

([a-zA-Z0-9]+[_\|\.\?])表示 a-z 或者 A-Z 或者 0-9 中任意一位+下划线或者点或者不加（? 表示出现一次或者零次）

*表示前面的字符串出现多次或者零次

后面同理

[a-zA-Z]{2,3}表示 a-zA-Z 中最少取两位，最多取三位

3.

先用 str.split(" ")将这句话分成一个个单词

循环中 RegExp(words[i], "i")表示第 i 的单词 以及"i"表示对大小写不敏感的匹配

if 进行正则判定 并且 确保 set 内组数小于 10 组

将两组一样的单词放入容器中 再存入 set

4.

先用 toUpperCase 将 want 的字全部换成大写字母 因为键盘上都是大写

将 actual 转换成单个字母的数组

for 循环中先将数组遍历并且通过正则表达式将 want 中能够打出来的字删除

剩下的即键盘上坏掉的字母

RegExp(actual[i], "ig")即对实际打出数组中的单个字与 want 进行全局大小写不敏感的匹配（“ig”）

5.

replace(/(^\s*)|(\s*\$)/g, "")表示将字符串头部和尾部零次或者多次出现的空格都去掉

replace(/\s+/ig, " ")表示将两个空格变成一个空格

6.

先将 nums 中的数通过循环存入 map 中 key: 数字 value: index 同时也存入这个加数位于第几位

用 get 方法判断最后的和减去一个加数 剩下的加数是否在 map 中

若 num 存在于 map 数组中 则 num 就表示另一个加数是第几位

用 push 将两个加数的位数就放进 arr 中

后将 arr 清空 寻找是否还有下一组加数

7.

map 存放字符串中的每一个数字以及位数

for 循环中 其实是判断两个一样的字母中间相差的个数

i: 表示前一位相同字母的位数 j 是后一位

然后用 $j-i+1$ 算出这一段字符串的长度 和 answer 中原有的纪录的长度取最大值

用 “bacdbadf” 举例 map 中实际存的是 b1 a2.....存到 b5 的时候

i=1 (b 先前的位置) j=4 (b 在字符串中的位置)

那这一段包含不同字符串的子字符串长度就为 $4-1+1=4$

8.

构造函数最后调用的时候要注意需要 new 出一个实例

而原型链之中包含了构造函数这一步骤 因此调用的时候只需要用 let var 即可

Object.create 则比原型链更进一步 甚至不需要声明可以直接调用

###Map, Set, Array

Map: 由于存在 key 和 value 一一对应的关系 在成对出现的变量处使用可以方便很多

Set: 与 Array 相类似 但有不会存入重复值的特性 因此在只看变量不看位置和数量的时候可以使用

Array: 普普通通的数组 但在看重顺序并且单变量的时候可以使用

三者之间可以互相转换



