前端测试题

要求:

\* 时长为15min

1. 给定一个二维数组data，实现getter和setter方法。getter方法可以根据输入的编号，获取到二维数组中对应位置的数据；setter方法可以根据输入的编号和值，重置数据。

const data = [

['a','c',1,6,5],

[6,7,'d',9,'3',11],

[12,13,14,15,16,17]

];

getter(data, 3) => 1

getter(data, 8) => 'd'

getter(data, 13) => 13

setter(data, 1, 'T')

=>

[

['T','c',1,6,5],

[6,7,'d',9,'3',11],

[12,13,14,15,16,17]

]

1. 给定一个数组，数组记录了当前数据的ID，和后面紧邻的数据ID。对该数据进行分组，实现以下功能:

功能描述:

对数组进行分类，

如果当前数据没有被其他数据的next指向，则为一个数组的第一个数据，

如果当前数据有被其他数据的next指向，则排在next指向的后面，

其他以此类推。

const data = [

{ id: 1, next: 2 },

{ id: 3, next: 4 },

{ id: 4, next: 5 },

{ id: 5, next: 6 },

{ id: 6, next: 7 },

{ id: 7, next: 8 },

{ id: 8, next: 9 },

{ id: 2, next: 10 },

{ id: 20, next: 30 },

{ id: 30, next: 40 },

{ id: 100, next: 78 }

]

getLinks(data)

=>

[

[{ id: 1, next: 2 }, { id: 2, next: 10 }],

[{ id: 3, next: 4 }, { id: 4, next: 5 }, { id: 5, next: 6 }, { id: 6, next: 7 }, { id: 7, next: 8 }, { id: 8, next: 9 }]

[{ id: 20, next: 30 }, { id: 30, next: 40 }],

[{ id: 100, next: 78 }]

]

1. 实现Animal类，可以同时达到以下效果:

输入:

new Animal('dog');

输出:

This is dog

输入:

new Animal('cat').sleep(5).eat('dinner');

输出：

This is cat

Sleep 5s ...

Wake up after 5s

Eat dinner~

输入:

new Animal(‘cat’).eat(‘dinner’).eat(‘supper’);

输出：

This is cat

Eat dinner~

Eat supper~

输入:

new Animal(‘cat’).sleepFirst(5).eat(‘supper’);

输出：

Sleep 5s ...

Wake up after 5s

Hi This is cat!

Eat supper~

1. 解析加减乘除指令

加法指令:

ADD(1, 2) => 1加2等于3

减法:

SUB(2, 1) => 2减去1等于1

乘法:

MUL(2, 3) => 2乘以3等于6

除法:

DIV(4, 2) => 4除以2等于2

实现解析函数parse，功能如下:

输入:

parse(‘ADD(1,2)’)

输出:

3

输入:

parse(‘SUB(2,1)’)

输出:

1

输入:

parse(‘MUL(2,1)’)

输出:

2

输入:

parse(‘DIV(4,2)’)

输出:

2