· Day1 js

1. 使用发布订阅模式，实现vue得$on和$emit、$off 方法
2. 实现instanceOf方法
3. 红绿灯问题，绿灯3秒，红灯2秒，黄灯1秒，每隔一秒打印一条记录 这样循环，要求：可以在控制台可以运行的代码，并且正确输出

// 绿灯 3

// 绿灯 2

// 绿灯 1

// 红灯 2

// 红灯 1

// 黄灯 1

// 绿灯 3

· Day2 js

1. 请封装一个数组reduce方法
2. 请使用冒泡和快速两种方式实现数组排序
3. 封装Storage对象

Storage.set('name', 哈哈哈') // 设置 name 字段存储的值为'哈哈哈'。

Storage.set('age', 2, 30);

Storage.set('people', ['Oli', 'Aman', 'Dante'], 60)

Storage.get('name') // ‘前端一万小时’

Storage.get('age') // 如果不超过 30 秒，返回数字类型的 2；如果超过 30 秒，返回 undefined，并且 localStorage 里清除 age 字段。

Storage.get('people') // 如果不超过 60 秒，返回数组； 如果超过 60 秒，返回 undefined。

1. 可以将数字转换成中文大写的表示，处理到万级别，例如 toChineseNum(12345)，返回 一万二千三百四十五

·Day3 js

1. 接受一个仅包含数字的 多维数组 ，返回一个迭代器，可以遍历得到它拍平以后的结果

**const** numbers = flatten2([1, [[2], 3, 4], 5])

numbers.next().value // => 1

numbers.next().value // => 2

numbers.next().value // => 3

numbers.next().value // => 4

numbers.next().value // => 5

补齐flatten2 这个函数

1. 复习数组方法 封装实现 数组 map、filter、every 方法

·Day4 js

1. 用自己的话描述js垃圾回收机制

3.深入理解this执行

function Foo() {

getName = function() { alert(1); }

return this

}

Foo.getName = function() { alert(2); }

Foo.prototype.getName = function() { alert(3); }

var getName = function () { alert(4); }

function getName() { alert(5); }

// 输出值？ 请写出原因

Foo.getName();

getName();

Foo().getName();

getName();

new Foo.getName()

new Foo().getName()

new new Foo().getName()

4. 重绘和回流（重排）的区别和关系 用自己的话描述

·Day5 js

1. 总结http协议 、同源策略、跨域问题
2. 中间件模式（middleware）是一种很常见、也很强大的模式，被广泛应用在 Express、Koa、Redux 等类库和框架当中。模拟一个中间件模式。

·Day 6 css

1. 请实现未知宽高的盒子在页面中水平垂直居中
2. 简述盒模型、怎么修改盒模型
3. src和href的区别
4. 简述8种position属性值
5. 简述css hack
6. 简述px、rem、em 的区别 怎么实现适配

·Day 7 webpack

1. 简述webpack模块打包原理
2. 简述webpack loader
3. Webpack一些常用的自定义配置
4. 开发和生产环境的构建配置差异，如何优化webpack构建速度

·Day 10 react&vue

1. 简述react和vue的区别
2. 简述spa，spa实现原理