

每日一题汇总

12312312

- Question 1? sdafdsf +

Answer 1

- Question 2? +

Answer 2

- Day1:请写出下面代码执行的的结果 +

```
console.log(1);
setTimeout(() => {
  console.log(2);
  process.nextTick(() => {
    console.log(3);
  });
  new Promise((resolve) => {
    console.log(4);
    resolve();
  }).then(() => {
    console.log(5);
  });
});
new Promise((resolve) => {
  console.log(7);
  resolve();
}).then(() => {
  console.log(8);
});
process.nextTick(() => {
  console.log(6);
});
setTimeout(() => {
  console.log(9);
  process.nextTick(() => {
    console.log(10);
  });
  new Promise((resolve) => {
    console.log(11);
    resolve();
  }).then(() => {
    console.log(12);
  });
});
```

分类：JavaScript

[答案&解析](#)

Day2:写出执行结果

```
function side(arr) {  
  arr[0] = arr[2];  
}  
function a(a, b, c = 3) {  
  c = 10;  
  side(arguments);  
  return a + b + c;  
}  
a(1, 1, 1);
```

分类：JavaScript

[答案&解析](#)

Day3:写出执行结果

```
var min = Math.min();  
max = Math.max();  
console.log(min < max);
```

分类：JavaScript

[答案&解析](#)

Day4:写出执行结果,并解释原因

```
var a = 1;  
(function a() {  
  a = 2;  
  console.log(a);  
})();
```

分类：JavaScript

[答案&解析](#)

Day5:写出执行结果,并解释原因

```
var a = [0];
if (a) {
  console.log(a == true);
} else {
  console.log(a);
}
```

分类: JavaScript

[答案&解析](#)

Day6:写出执行结果,并解释原因

```
(function () {
  var a = (b = 5);
})();

console.log(b);
console.log(a);
```

分类: JavaScript

[答案&解析](#)

Day7:写出执行结果,并解释原因

```
var fullname = "a";
var obj = {
  fullname: "b",
  prop: {
    fullname: "c",
    getFullname: function () {
      return this.fullname;
    },
  },
};

console.log(obj.prop.getFullname()); // c
var test = obj.prop.getFullname;
console.log(test()); // a
```

分类： JavaScript

[答案&解析](#)

Day8:写出执行结果,并解释原因

```
var company = {  
  address: "beijing",  
};  
var yideng = Object.create(company);  
delete yideng.address;  
console.log(yideng.address);
```

分类： JavaScript

[答案&解析](#)

Day9:写出执行结果,并解释原因

```
var foo = function bar() {  
  return 12;  
};  
console.log(typeof bar());
```

分类： JavaScript

[答案&解析](#)

Day10:写出执行结果,并解释原因

```
var x = 1;  
if (function f() {}) {  
  x += typeof f;  
}  
console.log(x);
```

分类： JavaScript

[答案&解析](#)

Day11:写出执行结果,并解释原因

```
function f() {  
    return f;  
}  
console.log(new f() instanceof f);
```

分类: JavaScript

[答案&解析](#)

Day12:写出执行结果,并解释原因

```
var foo = {  
    bar: function () {  
        return this.baz;  
    },  
    baz: 1,  
};  
console.log(typeof (f = foo.bar)());
```

分类: JavaScript

[答案&解析](#)

Day13:关于 AMD、CMD 规范区别说法正确的是? (多选)

关于AMD、CMD规范区别说法正确的是? (多选)

- A.AMD规范: 是 RequireJS在推广过程中对模块定义的规范化产出的
- B.CMD规范: 是SeaJS 在推广过程中对模块定义的规范化产出的
- C.CMD 推崇依赖前置;AMD 推崇依赖就近
- D.CMD 是提前执行;AMD 是延迟执行
- E.AMD性能好,因为只有用户需要的时候才执行;CMD用户体验好,因为没有延迟,依赖模块提前执行了

分类: JavaScript

[答案&解析](#)

Day14:关于 SPA 单页页面的理解正确的是?

关于SPA单页面页面的理解正确的是？

- A. 用户体验好、快，但是内容的改变需要重新加载整个页面，会造成不必要的跳转和重复渲染；
- B. 前后端职责分离，架构清晰，前端进行交互逻辑，后端负责数据处理；
- C. 初次加载耗时多：为实现单页 Web 应用功能及显示效果，需要在加载页面的时候将 JavaScript、CSS 统一加载，部分页面按需加载；
- D. 前进后退路由管理需要使用浏览器的前进后退功能
- E. SEO 难度较大：由于所有的内容都在一个页面中动态替换显示，所以在 SEO 上其有着天然的弱势。

分类：Vue、React

[答案&解析](#)

Day15:下面对 Vue.js 中 keep-alive 的理解正确的是？（多选）

下面对Vue.js中keep-alive的理解正确的是？（多选）

- A. 一般结合路由和动态组件一起使用，用于缓存组件；
- B. 提供 include 和 exclude 属性，两者都支持字符串或正则表达式，include 表示只有名称匹配的组件会被缓存，exclude 表示任何名称匹配的组件都不会被缓存，其中 include 的优先级比 exclude 高；
- C. 对应两个钩子函数 activated 和 deactivated，当组件被激活时，触发钩子函数 activated，当组件被移除时，触发钩子函数 deactivated。
- D. keep-alive 是 Vue 内置的一个组件，可以使被包含的组件保留状态，但是不能避免重新渲染

分类：Vue

[答案&解析](#)

Day16:关于 Vue.js 虚拟 DOM 的优缺点说法正确的是？（多选）

关于Vue.js虚拟DOM的优缺点说法正确的是？（多选）

- A. 可以保证性能下限，比起粗暴的 DOM 操作性能要好很多，因此框架的虚拟 DOM 至少可以保证在你不需要手动优化的情况下，依然可以提供还不错的性能，即保证性能的下限；
- B. 无需手动操作DOM，不再需要手动去操作 DOM，只需要写好 View-Model 的代码逻辑，框架会根据虚拟 DOM 和 数据双向绑定，帮我们以可预期的方式更新视图，极大提高我们的开发效率；
- C. 可以进行极致优化：虚拟 DOM + 合理的优化，可以使性能达到极致
- D. 可以跨平台，虚拟 DOM 本质上是 JavaScript 对象，而 DOM 与平台强相关，相比之下虚拟 DOM 可以进行更方便地跨平台操作，例如服务器渲染、weex 开发等等。

分类: Vue

[答案&解析](#)

Day17:下面代码输出什么?

```
for (let i = 0; i < 3; i++) {  
  setTimeout(() => console.log(i), 1);  
}
```

分类: JavaScript

[答案&解析](#)

Day18:写出执行结果,并解释原因

```
const num = {  
  a: 10,  
  add() {  
    return this.a + 2;  
  },  
  reduce: () => this.a - 2;  
};  
console.log(num.add());  
console.log(num.reduce());
```

分类: JavaScript

[答案&解析](#)

Day19:写出执行结果,并解释原因

```
const person = { name: "yideng" };  
  
function sayHi(age) {  
  return `${this.name} is ${age}`;  
}  
  
console.log(sayHi.call(person, 21));  
console.log(sayHi.bind(person, 21));
```

分类: JavaScript

[答案&解析](#)

Day20:写出执行结果,并解释原因

```
["1", "2", "3"].map(parseInt);
```

分类: JavaScript

[答案&解析](#)

Day21:写出执行结果,并解释原因

```
[typeof null, null instanceof Object];
```

分类: JavaScript

[答案&解析](#)

Day22:写出执行结果,并解释原因

```
function f() {}  
const a = f.prototype,  
      b = Object.getPrototypeOf(f);  
console.log(a === b);
```

分类: JavaScript

[答案&解析](#)

Day23:写出执行结果,并解释原因

```
function showCase(value) {  
  switch (value) {  
    case "A":  
      console.log("Case A");  
      break;  
    case "B":  
      console.log("Case B");  
  }  
}
```



```
        break;
    case undefined:
        console.log("undefined");
        break;
    default:
        console.log("Do not know!");
    }
}
showCase(new String("A"));
```

分类： JavaScript

[答案&解析](#)

Day24:选择正确的答案

```
console.log([2,1,0].reduce(Math.pow));
console.log([].reduce(Math.pow));

/ *
A. 2 报错
B. 2 NaN
C. 1 报错
D. 1 NaN
*/
```

分类： JavaScript

[答案&解析](#)

Day25:请问变量 a 会被 GC 吗

```
function test() {
    var a = 1;
    return function () {
        eval("");
    };
}
test();
```

分类： JavaScript

[答案&解析](#)

Day26:写出执行结果,并解释原因

```
const value = "Value is" + !!Number(["0"]) ? "yideng" : "undefined";
console.log(value);
```

分类: JavaScript

[答案&解析](#)

Day27:写出执行结果,并解释原因

```
var arr = [0, 1];
arr[5] = 5;
newArr = arr.filter(function (x) {
  return x === undefined;
});
console.log(newArr.length);
```

分类: JavaScript

[答案&解析](#)

Day28:写出执行结果,并解释原因(以最新谷歌浏览器为准)

```
async function async1() {
  console.log("async1 start");
  await async2();
  console.log("async1 end");
}
async function async2() {
  console.log("async2");
}
console.log("script start");
setTimeout(function () {
  console.log("setTimeout");
}, 0);
async1();
new Promise(function (resolve) {
  console.log("promise1");
  resolve();
}).then(function () {
  console.log("promise2");
});
console.log("script end");
```

分类： JavaScript

[答案&解析](#)

Day29:下面代码中 a 在什么情况下会打印 1

```
var a = ?;  
if(a == 1 && a== 2 && a== 3){  
    console.log(1);  
}
```

分类： JavaScript

[答案&解析](#)

Day30:写出执行结果,并解释原因

```
const obj = {  
    2: 3,  
    3: 4,  
    length: 2,  
    splice: Array.prototype.splice,  
    push: Array.prototype.push,  
};  
obj.push(1);  
obj.push(2);  
console.log(obj);
```

分类： JavaScript

[答案&解析](#)

Day31:写出执行结果,并解释原因

```
let a = { n: 1 };  
let b = a;  
a.x = a = { n: 2 };  
console.log(a.x);  
console.log(b.x);
```

分类: JavaScript

[答案&解析](#)

Day32:写出执行结果,并解释原因

```
var a1 = {},
    b1 = "123",
    c1 = 123;
a1[b1] = "b";
a1[c1] = "c";
console.log(a1[b1]);
var a2 = {},
    b2 = Symbol("123"),
    c2 = Symbol("123");
a2[b2] = "b";
a2[c2] = "c";
console.log(a2[b2]);
var a3 = {},
    b3 = { key: "123" },
    c3 = { key: "456" };
a3[b3] = "b";
a3[c3] = "c";
console.log(a3[b3]);
```

分类: JavaScript

[答案&解析](#)

Day33:写出执行结果,并解释原因

```
function Foo() {
  Foo.a = function () {
    console.log(1);
  };
  this.a = function () {
    console.log(2);
  };
}
Foo.prototype.a = function () {
  console.log(3);
};
Foo.a = function () {
  console.log(4);
};
Foo.a();
```

```
let obj = new Foo();  
obj.a();  
Foo.a();
```

分类：JavaScript

[答案&解析](#)

Day34:写出执行结果,并解释原因

```
function user(obj) {  
  obj.name = "京程一灯";  
  obj = new Object();  
  obj.name = "精英班";  
}  
let person = new Object();  
user(person);  
console.log(person.name);
```

分类：JavaScript

[答案&解析](#)

Day35:写出执行结果,并解释原因

```
let x, y;  
try {  
  throw new Error();  
} catch (x) {  
  x = 1;  
  y = 2;  
  console.log(x);  
}  
console.log(x);  
console.log(y);
```

分类：JavaScript

[答案&解析](#)

Day36:写出执行结果,并解释原因

```
function fn() {
  getValue = function () {
    console.log(1);
  };
  return this;
}
fn.getValue = function () {
  console.log(2);
};
fn.prototype.getValue = function () {
  console.log(3);
};
var getValue = function () {
  console.log(4);
};
function getValue() {
  console.log(5);
}

//请写出以下输出结果:
getValue();
fn().getValue();
getValue();
new fn().getValue();
new fn().getValue();
```

分类: JavaScript

[答案&解析](#)

Day37:写出执行结果,并解释原因

```
let length = 10;
function fn() {
  console.log(this.length);
}
var obj = {
  length: 5,
  method: function (fn) {
    fn();
    arguments[0]();
  },
};
obj.method(fn, 1);
```

分类: JavaScript

[答案&解析](#)

Day38:写出执行结果,并解释原因

```
var a = 10;
var foo = {
  a: 20,
  bar: function () {
    var a = 30;
    return this.a;
  },
};
console.log(foo.bar());
console.log(foo.bar());
console.log((foo.bar = foo.bar)());
console.log((foo.bar, foo.bar)());
```

分类: JavaScript

[答案&解析](#)

Day39:写出执行结果,并解释原因

```
function getName() {
  for (let i = 0; i < 5; i++) {
    setTimeout(function () {
      console.log(i);
    }, i * 1000);
  }
  return;
  {
    name: "京程一灯";
  }
}
console.log(getName());
```

分类: JavaScript

[答案&解析](#)

Day40:写出执行结果,并解释原因

```
const num = parseInt("2*4", 10);
console.log(num);
```

分类： JavaScript

[答案&解析](#)

Day41:选择正确答案，并解释为什么

```
const company = { name: "京程一灯" };
Object.defineProperty(company, "address", { value: "北京" });
console.log(company);
console.log(Object.keys(company));
/*
A. {name:"京程一灯",address:"北京"}, ["name", "age"]
B. {name:"京程一灯",address:"北京"}, ["name"]
C. {name:"京程一灯"}, ["name", "age"]
D. {name:"京程一灯"}, ["name", "age"]
*/
```

分类： JavaScript

[答案&解析](#)

Day42:写出执行结果,并解释原因

```
let num = 10;
const increaseNumber = () => num++;
const increasePassedNumber = (number) => number++;
const num1 = increaseNumber();
const num2 = increasePassedNumber(num1);
console.log(num1);
console.log(num2);
```

分类： JavaScript

[答案&解析](#)

Day43:写出执行结果,并解释原因

```
const value = { number: 10 };
const multiply = (x = { ...value }) => {
  console.log((x.number *= 2));
};
```



```
multiply();  
multiply();  
multiply(value);  
multiply(value);
```

分类: JavaScript

[答案&解析](#)

Day44:写出执行结果,并解释原因

```
[1, 2, 3, 4].reduce((x, y) => console.log(x, y));
```

分类: JavaScript

[答案&解析](#)

Day45:写出执行结果,并解释原因

```
// index.js  
console.log("running index.js");  
import { sum } from "./sum.js";  
console.log(sum(1, 2));  
  
// sum.js  
console.log("running sum.js");  
export const sum = (a, b) => a + b;
```

分类: JavaScript

[答案&解析](#)

Day46:写出执行结果,并解释原因

```
function addToList(item, list) {  
  return list.push(item);  
}  
const result = addToList("company", ["yideng"]);  
console.log(result);
```

分类: JavaScript

[答案&解析](#)

Day47:写出执行结果,并解释原因

```
var a = 0;
if (true) {
  a = 10;
  console.log(a, window.a);
  function a() {}
  console.log(a, window.a);
  a = 20;
  console.log(a, window.a);
}
console.log(a);
```

分类: JavaScript

[答案&解析](#)

Day48:能否以某种方式为下面的语句使用展开运算而不导致类型错误

```
var obj = { x: 1, y: 2, z: 3 };
[...obj]; // TypeError
// 能否以某种方式为上面的语句使用展开运算而不导致类型错误
// 如果可以, 写出解决方式
```

分类: JavaScript

[答案&解析](#)

Day49:请你完成一个 safeGet 函数, 可以安全的获取无限多层次的数据

```
// 请你完成一个safeGet函数, 可以安全的获取无限多层次的数据, 一旦数据不存在不会报错, 会返回 undefined, 例如
var data = { a: { b: { c: "yideng" } } };
safeGet(data, "a.b.c"); // => yideng
safeGet(data, "a.b.c.d"); // => undefined
safeGet(data, "a.b.c.d.e.f.g"); // => undefined
```

分类：JavaScript、编程题

[答案&解析](#)

Day50:写一个 isPrime()函数

写一个isPrime()函数，当其为质数时返回true，否则返回false。
提示：质数是指在大于1的自然数中，除了1和它本身以外不再有其他因数的自然数。

分类：算法

[答案&解析](#)

Day52:请实现一个 flattenDeep 函数，把嵌套的数组扁平化~~

```
flattenDeep([1, [2, [3, [4]], 5]]); //[1, 2, 3, 4, 5]  
// 请实现一个flattenDeep函数，把嵌套的数组扁平化
```

分类：JavaScript、编程题

[答案&解析](#)

Day53:请实现一个 uniq 函数，实现数组去重~~

```
uniq([1, 2, 3, 5, 3, 2]); //[1, 2, 3, 5]  
// 请实现一个 uniq 函数，实现数组去重
```

分类：算法

[答案&解析](#)

Day54:new 操作符都做了什么，并手动实现一下

分类：JavaScript、编程题

[答案&解析](#)

Day55:实现 (5).add(3).minus(2) 功能

```
// 实现 (5).add(3).minus(2) 功能
console.log((5).add(3).minus(2)); // 6
```

分类: JavaScript

[答案&解析](#)

Day56:介绍下 Set、Map、WeakSet 和 WeakMap 的区别

分类: JavaScript

[答案&解析](#)

Day57:如何在不使用%模运算符的情况下检查一个数字是否是偶数

分类: JavaScript

[答案&解析](#)

Day58:Object.seal 和 Object.freeze 方法之间有什么区别

分类: JavaScript

[答案&解析](#)

Day59:完成 plus 函数，通过全部的测试用例

```
"use strict";
function plus(n) {}
module.exports = plus;
// 测试用例如下
("use strict");
var assert = require("assert");
var plus = require("../lib/assign-4");
describe("测试用例", function () {
  it("plus(0) === 0", function () {
    assert.equal(0, plus(0).toString());
  });
  it("plus(1)(1)(2)(3)(5) === 12", function () {
    assert.equal(12, plus(1)(1)(2)(3)(5).toString());
  });
  it("plus(1)(4)(2)(3) === 10", function () {
    assert.equal(10, plus(1)(4)(2)(3).toString());
  });
});
```

```
});  
it("plus(1,1)(2,2)(3)(4) === 13", function () {  
    assert.equal(13, plus(1, 1)(2, 2)(3)(4).toString());  
});  
});
```

分类: JavaScript

[答案&解析](#)

Day60:解释下这段代码的意思以及用到的技术点

```
[].forEach.call($$("*"), function (a) {  
    a.style.outline =  
        "1px solid #" + (~(Math.random() * (1 << 24))).toString(16);  
});
```

分类: JavaScript

[答案&解析](#)

Day61:写出执行结果,并解释原因

```
var yideng_a = Function.length;  
var yideng_b = new Function().length;  
console.log(yideng_a === yideng_b);
```

分类: JavaScript

[答案&解析](#)

Day62:不借助中间变量交换两个变量的值

不借助中间变量交换两个变量的值
比如 `let a = 1, b = 2;` 交换a, b的值

分类: JavaScript

[答案&解析](#)

Day63:实现一个 isNegativeZero 函数，只检查+0 和-0， -0 则返回 true,+0 返回 false

```
// 实现一个isNegativeZero函数，只检查+0和-0，-0则返回true,+0返回false
function isNegativeZero(num) {
  // 代码实现
}
```

分类：JavaScript、编程题

[答案&解析](#)

Day64:补全代码

```
/*
  说明：该文件名未知，位于当前项目内的dist/scripts文件夹内
  要求：一句话补全代码，获取它的完整位置:http://xx.com/dis/scripts/xx.js
  注：非node环境，node可以使用__dirname
*/
const url = 代码书写处;
export default url;
```

分类：JavaScript

[答案&解析](#)

Day65:选择正确的选项

```
class YiDeng {
  static str = "京程一灯";
  sayStr = () => {
    throw new Error("Need to implement");
  };
}
class Student extends YiDeng() {
  constructor() {
    super();
  }
  sayStr() {
    console.log(Student.str);
  }
}
const laoyuan = new Student();
```

```
console.log(Student.str);  
laoyuan.sayStr();  
  
//A.undefiend, 报错Need to implement  
// B. undefiend, 京程一灯  
// C. undefined, undefined  
// D.京程一灯, 报错Need to implement  
// E. 京程一灯, 京程一灯  
// F.京程一灯, undefined  
// G. str is not defined, 京程一灯
```

分类: JavaScript

[答案&解析](#)

Day66:一个简单的算法题目

给定一个整数数组 `nums` 和一个目标值 `target`, 请你在该数组中找出和为目标值的那两个整数, 这两个整数可能有多种组合, 找出其中一组组合即可, 并返回他们的数组下标。

示例:

给定 `nums = [2, 7, 11, 15]`, `target = 9`
因为 `nums[0] + nums[1] = 2 + 7 = 9`
所以返回 `[0, 1]`

分类: JavaScript

[答案&解析](#)

Day67:写出执行结果,并解释原因

```
3.toString()  
3..toString()  
3...toString()
```

分类: JavaScript

[答案&解析](#)

Day68:写出执行结果,并解释原因

```
function yideng() {}  
const a = {},  
      b = Object.prototype;  
console.log(a.prototype === b);  
console.log(Object.getPrototypeOf(a) === b);  
console.log(yideng.prototype === Object.getPrototypeOf(yideng));
```

分类：JavaScript

[答案&解析](#)

Day69:写出执行结果,并解释原因

```
const lowerCaseOnly = /^[a-z]+$/;  
console.log(lowerCaseOnly.test("yideng"));  
console.log(lowerCaseOnly.test(null));  
console.log(lowerCaseOnly.test());
```

分类：JavaScript

[答案&解析](#)

Day70:写出执行结果,并解释原因

```
function captureOne(re, str) {  
  var match = re.exec(str);  
  return match && match[1];  
}  
var numRe = /num=(\d+)/gi,  
    wordRe = /yideng=(\w+)/i,  
    a1 = captureOne(numRe, "num=1"),  
    a2 = captureOne(wordRe, "yideng=1"),  
    a3 = captureOne(numRe, "NUM=2"),  
    a4 = captureOne(wordRe, "YIDENG=2"),  
    a5 = captureOne(numRe, "Num=3"),  
    a6 = captureOne(wordRe, "YiDeng=3");  
console.log(a1 === a2);  
console.log(a3 === a4);  
console.log(a5 === a6);
```

分类：JavaScript

[答案&解析](#)

Day71:[手写代码]实现 Promise.all 方法

分类：JavaScript、编程题

[答案&解析](#)

Day72:有效括号算法题

```
/*
    给定一个只包括 '(' , ')' , '{' , '}' , '[' , ']' 的字符串，判断字符串是否有效
    有效字符串需满足：
        1. 左括号必须用相同类型的右括号闭合。
        2. 左括号必须以正确的顺序闭合。
    注意空字符串可被认为是有效字符串。
    示例1：
        输入："()"
        输出：true
    示例2：
        输入："() [] {}"
        输出：true
    示例 3：
        输入："]"
        输出：false
    示例 4：
        输入："([)]"
        输出：false
    示例 5：
        输入："{}[]"
        输出：true
*/
```

分类：算法

[答案&解析](#)

Day73:写出执行结果,并解释原因

```
function yideng(n, o) {
  console.log(o); // ?
  return {
    yideng: function (m) {
      return yideng(m, n);
    },
  };
};
```

```
}  
const a = yideng(0);  
a.yideng(1);  
a.yideng(2);  
a.yideng(3);  
const b = yideng(0).yideng(1).yideng(2).yideng(3);  
const c = yideng(0).yideng(1);  
c.yideng(2);  
c.yideng(3);
```

分类： JavaScript

[答案&解析](#)

Day74:写出执行结果,并解释原因

```
var arr1 = "ab".split("");  
var arr2 = arr1.reverse();  
var arr3 = "abc".split("");  
arr2.push(arr3);  
console.log(arr1.length);  
console.log(arr1.slice(-1));  
console.log(arr2.length);  
console.log(arr2.slice(-1));
```

分类： JavaScript

[答案&解析](#)

Day75:写出执行结果,并解释原因

```
var F = function () {};  
Object.prototype.a = function () {  
  console.log("yideng");  
};  
Function.prototype.b = function () {  
  console.log("xuetaang");  
};  
var f = new F();  
F.a();  
F.b();  
f.a();  
f.b();
```

分类: JavaScript

[答案&解析](#)

Day76:写出执行结果,并解释原因

```
const a = [1, 2, 3],  
      b = [1, 2, 3],  
      c = [1, 2, 4],  
      d = "2",  
      e = "11";  
console.log([a == b, a === b, a > c, a < c, d > e]);
```

分类: JavaScript

[答案&解析](#)

Day77:补充代码, 使代码可以正确执行

```
const str = "1234567890";  
function formatNumber(str) {  
  // your code  
}  
console.log(formatNumber(str)); //1,234,567,890  
// 补充代码, 使代码可以正确执行
```

分类: JavaScript

[答案&解析](#)

Day78:Script 放在底部还会影响 dom 的解析和渲染吗? Script 内部的代码执行会等待 css 加载完吗? css 加载会影响 DOMContentLoaded 么?

```
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    <meta charset="UTF-8" />  
    <title>京程一灯, 每日一题</title>  
  </head>  
  <body>  
    <!--
```

1.Script放在底部还会影响dom的解析和渲染吗?

2. Script内部的代码执行会等待css加载完吗?

3. css加载会影响DOMContentLoaded么?

-->

```
<link href="https://cdn/css/bootstrap.css" ref="stylesheet" />
<h1>京程一灯</h1>
<script>
  console.log("DOMContentLoaded");
</script>
</body>
</html>
```

分类: JavaScript

[答案&解析](#)

Day79:写出下面代码 null 和 0 进行比较的代码执行结果, 并解释原因

```
console.log(null == 0);
console.log(null <= 0);
console.log(null < 0);
```

分类: JavaScript

[答案&解析](#)

Day80:关于数组 sort, 下面代码的正确打印结果是什么, 并解释原因

```
const arr1 = ["a", "b", "c"];
const arr2 = ["b", "c", "a"];
console.log(
  arr1.sort() === arr1,
  arr2.sort() === arr2,
  arr1.sort() === arr2.sort()
);
```

分类: JavaScript

[答案&解析](#)

Day81:介绍防抖与节流的原理, 并动手实现

```
// 京程一灯，每日一题
const debounce = (fn, delay) => {
  // 介绍防抖函数原理，并实现
  // your code
};
const throttle = (fn, delay = 500) => {
  // 介绍节流函数原理，并实现
  // your code
};
```

分类：JavaScript

[答案&解析](#)

Day82:关于隐式转换，下面代码的执行结果是什么？并解释原因

```
let a = [];
let b = "0";
console.log(a == 0);
console.log(a == !a);
console.log(b == 0);
console.log(a == b);
```

分类：JavaScript

[答案&解析](#)

Day83:请写出如下代码的打印结果

```
var obj = {};
var x = +obj.yideng?.name ?? "京程一灯";
console.log(x);
```

分类：JavaScript

[答案&解析](#)

Day84:对于 length 下面代码的输出结果是什么？并解释原因

```
function foo() {
  console.log(length);
}
```

```
}  
function bar() {  
  var length = "京程一灯";  
  foo();  
}  
bar();
```

分类： JavaScript

[答案&解析](#)

Day85:对于扩展运算符，下面代码的执行结果是什么？并解释原因

```
let ydObject = { ...null, ...undefined };  
console.log(ydObject);  
let ydArray = [...null, ...undefined];  
console.log(ydArray);
```

分类： JavaScript

[答案&解析](#)

Day86:写出类数组转换结果，并解释原因

```
const arrLike = {  
  length: 4,  
  0: 0,  
  1: 1,  
  "-1": 2,  
  3: 3,  
  4: 4,  
};  
console.log(Array.from(arrLike));  
console.log(Array.prototype.slice.call(arrLike));
```

分类： JavaScript

[答案&解析](#)

Day87:写出下面代码 1, 2, 3 的大小判断结果

```
console.log(1 < 2 < 3);  
console.log(3 > 2 > 1);
```

分类：JavaScript

[答案&解析](#)

Day88:以下两段代码会抛出异常吗？解释原因？

```
let yd = { x: 1, y: 2 };  
// 以下两段代码会抛出异常吗？  
let ydWithXGetter1 = {  
  ...yd,  
  get x() {  
    throw new Error();  
  },  
};  
  
let ydWithXGetter2 = {  
  ...yd,  
  ...{  
    get x() {  
      throw new Error();  
    },  
  },  
};
```

分类：JavaScript

[答案&解析](#)

Day89:请问 React 调用机制一共对任务设置了几种优先级别？每种优先级都代表的具体含义是什么？在你开发过程中如果遇到影响主 UI 渲染卡顿的任务，你又是如何利用这些优先级的？

分类：React

[答案&解析](#)

Day90:Vue 父组件可以监听到子组件的生命周期吗？如果能请写出你的实现方法。

分类：Vue

[答案&解析](#)

Day91:Vue 为什么要用 vm.\$set() 解决对象新增属性不能响应的问题？你能说说如下代码的实现原理么？

```
Vue.set(object, propertyName, value);  
vm.$set(object, propertyName, value);
```

分类：Vue

[答案&解析](#)

Day92:既然 Vue 通过数据劫持可以精准探测数据在具体 dom 上的变化,为什么还需要虚拟 DOM diff 呢？

分类：Vue

[答案&解析](#)

Day93:Vue 组件中写 name 选项有除了搭配 keep-alive 还有其他作用么？你能谈谈你对 keep-alive 了解么？（平时使用和源码实现方面）

分类：Vue

[答案&解析](#)

Day94:说一下 React Hooks 在平时开发中需要注意的问题和原因？

分类：React

[答案&解析](#)

Day95:Promise.all 中任何一个 Promise 出现错误的时候都会执行 reject，导致其它正常返回的数据也无法使用。你有什么解决办法么？

分类：JavaScript

[答案&解析](#)

Day96:请能尽可能多的说出 Vue 组件间通信方式？在组件的通信中 EventBus 非常经典，你能手写实现下 EventBus 么？

分类: Vue

[答案&解析](#)

Day97:请讲一下 react-redux 的实现原理?

分类: React

[答案&解析](#)

Day98:写出下面代码的执行结果, 并解释原因

```
Object.prototype.yideng = "京程一灯";  
var a = 123;  
a.b = 456;  
console.log(a.yideng);  
console.log(a.b);
```

分类: React

[答案&解析](#)

Day99:React 中 setState 后发生了什么? setState 为什么默认是异步? setState 什么时候是同步?

分类: React

[答案&解析](#)

Day100:哪些方法会触发 react 重新渲染? 重新渲染 render 会做些什么?

分类: React

[答案&解析](#)

Day101:Vue v-model 是如何实现的, 语法糖实际是什么

分类: Vue

[答案&解析](#)

Day102:说一下减少 dom 数量的办法？一次性给你大量的 dom 怎么优化？

分类：Html

[答案&解析](#)

Day103:多个 tab 只对应一个内容框，点击每个 tab 都会请求接口并渲染到内容框，怎么确保频繁点击 tab 但能够确保数据正常显示？

分类：JavaScript

[答案&解析](#)

Day104:项目中如何进行异常捕获

分类：JavaScript

[答案&解析](#)

Day105:JavaScript 中如何模拟实现方法的重载,动手实现下

分类：JavaScript

[答案&解析](#)

Day106:Webpack 里面的插件是怎么实现的？

分类：工程化

[答案&解析](#)

Day107:对虚拟 DOM 的理解？虚拟 DOM 主要做了什么？虚拟 DOM 本身是什么？

分类：React、Vue

[答案&解析](#)

Day108:Webpack 为什么慢，如何进行优化

分类：工程化

[答案&解析](#)

Day109:动画性能如何检测

分类: JavaScript

[答案&解析](#)

Day110:客户端缓存有几种方式? 浏览器出现 from disk、from memory 的策略是啥

分类: 网络&安全

[答案&解析](#)

Day111:原生 JavaScript 实现图片懒加载的思路

分类: JavaScript

[答案&解析](#)

Day112:数组里面有 10 万个数据, 取第一个元素和第 10 万个元素的时间相差多少

分类: JavaScript

[答案&解析](#)

Day113:Import 和 CommonJS 在 webpack 打包过程中有什么不同

分类: 工程化

[答案&解析](#)

Day114:说一下 Webpack 热更新的原理

分类: 工程化

[答案&解析](#)

Day115:说一下 vue-router 的原理

分类: Vue

[答案&解析](#)

Day116:商城的列表页跳转到商品的详情页，详情页数据接口很慢，前端可以怎么优化用户体验？

分类：JavaScript

[答案&解析](#)

Day117:React SSR 实现过程？原理是什么？有什么注意事项？

分类：React

[答案&解析](#)

Day118:说下单点登录实现原理

分类：JavaScript

[答案&解析](#)

Day119:怎样判断一个对象是否是数组，如何处理类数组对象

分类：JavaScript

[答案&解析](#)

Day120:说一下 CORS 的简单请求和复杂请求的区别

分类：网络&安全

[答案&解析](#)

Day121:说一下 在 map 中和 for 中调用异步函数的区别

分类：JavaScript

[答案&解析](#)

Day122:说一下 import 的原理，与 require 有什么不同？

分类：JavaScript

[答案&解析](#)

Day123:说下 webpack 的 loader 和 plugin 的区别，都使用过哪些 loader 和 plugin

分类：工程化

[答案&解析](#)

Day124:说一下 Https 加密握手的过程

分类：网络&安全

[答案&解析](#)

Day125:按要求实现代码

分类：算法

[答案&解析](#)

Day126:请描述 CSRF、XSS 的基本概念、攻击原理和防御措施？

分类：网络&安全

[答案&解析](#)

Day127:按要求完成 mergePromise 代码

```
const timeout = (ms) =>
  new Promise((resolve, reject) => {
    setTimeout(() => {
      resolve();
    }, ms);
  });
const ajax1 = () =>
  timeout(2000).then(() => {
    console.log("1");
    return 1;
  });
const ajax2 = () =>
  timeout(1000).then(() => {
    console.log("2");
    return 2;
  });
```

```
});  
const ajax3 = () =>  
  timeout(2000).then(() => {  
    console.log("3");  
    return 3;  
  });  
const mergePromise = (ajaxArray) => {  
  // 1,2,3 done [1,2,3] 此处写代码 请写出ES6、ES3 2中解法  
};  
mergePromise([ajax1, ajax2, ajax3]).then((data) => {  
  console.log("done");  
  console.log(data); // data 为[1,2,3]  
});  
// 执行结果为: 1 2 3 done [1,2,3]
```

公司：阿里

分类：JavaScript、编程题

[答案&解析](#)

Day128:说一下 React.Component 和 React.PureComponent 的区别

公司：高德、有赞、酷家乐

分类：React

[答案&解析](#)

Day129:React 高阶组件 、Render props 和 hooks 有什么区别，为什么要不断迭代

分类：React

[答案&解析](#)

Day130:什么是深拷贝，和浅拷贝有什么区别，动手实现一个深拷贝

分类：JavaScript

[答案&解析](#)

Day131:介绍下 webpack，并说下 Webpack 的构建流程

公司：头条、挖财

分类：工程化

[答案&解析](#)

Day132:说下 tree-shaking 的原理

公司：头条

分类：工程化

[答案&解析](#)

Day133:说一下 mobx 和 redux 有什么区别？

公司：洋葱学院

分类：React

[答案&解析](#)

Day134:Typescript 里有哪些 JavaScript 没有的类型？

公司：滴滴

分类：JavaScript

[答案&解析](#)

Day135:说一下单元测试、E2E 测试？它们有什么区别？

公司：头条

分类：工程化

[答案&解析](#)

Day136:BFC 是什么？触发 BFC 的条件是什么？有哪些应用场景？

公司：快手、网易、伴鱼

分类：Css

[答案&解析](#)

Day137:说说浏览器渲染流程，分层之后在什么时候合成，什么是重排、重绘，怎样避免

公司：老虎

分类：Css

[答案&解析](#)

Day138:有 1000 个 dom，需要更新其中的 100 个，如何操作才能减少 dom 的操作？

公司：爱范儿

分类：JavaScript

[答案&解析](#)

Day139:Vue data 中某一个属性的值发生改变后，视图会立即同步执行重新渲染吗？为什么？

分类：Vue

[答案&解析](#)

Day140:Fiber 算法原理，相比之前的 stack 算法哪些方面做了优化？

分类：React

[答案&解析](#)

Day141:请描述下 JavaScript 中 Scope、Closure、Prototype 概念，并说明 JavaScript 封装、继承实现原理。

公司：玄武科技

分类：JavaScript

[答案&解析](#)

Day142:请列出目前主流的 JavaScript 模块化实现的技术有哪些？说出它们的区别？

公司：玄武科技

分类：JavaScript

[答案&解析](#)

Day143:说一下 Vue3 与 Vue2 的对比

公司：高德

分类：Vue

[答案&解析](#)

Day144:说一下 Vue 单页与多页的区别

分类：Vue

[答案&解析](#)

Day145:介绍下 pm2, pm2 依据什么重启服务

公司：阿里

分类：Node

[答案&解析](#)

Day146:原生 JavaScript 实现图片懒加载的思路

公司：安居客

分类：JavaScript

[答案&解析](#)

Day147:在浏览器执行以下代码，写出打印结果

```
console.log("start");
setTimeout(() => {
  console.log("children2");
  Promise.resolve().then(() => {
    console.log("children3");
  });
}, 0);
new Promise(function (resolve, reject) {
  console.log("children4");
  setTimeout(function () {
    console.log("children5");
    resolve("children6");
  }, 0);
}).then((res) => {
```

```
console.log("children7");
setTimeout(() => {
  console.log(res);
}, 0);
});
```

公司：新东方

分类：JavaScript

[答案&解析](#)

Day148:请实现一个 cacheRequest 方法，保证发出多次同一个 ajax 请求时都能拿到数据，而实际上只发出一次请求

公司：快手

分类：JavaScript、编程题

[答案&解析](#)

Day149:一个人每次只能走一层楼梯或者两层楼梯，问走到第 80 层楼梯一共有多少种方法

公司：快手

分类：算法

[答案&解析](#)

Day150:说一下对 React context 的理解

公司：脉脉、酷家乐

分类：React

[答案&解析](#)

Day151:for..of 和 for...in 是否可以直接遍历对象？为什么？如果不可以有什么解决方案？

分类：JavaScript

[答案&解析](#)

Day152:for..in 和 Object.keys 的区别

分类：JavaScript

[答案&解析](#)

Day153:Promise.resolve(obj), obj 有几种可能

分类: JavaScript

[答案&解析](#)

Day154:给定起止日期, 返回中间的所有月份

```
// 输入两个字符串 2018-08 2018-12  
// 输出他们中间的月份 [2018-9,2018-10, 2018-11]
```

分类: JavaScript

[答案&解析](#)

Day155:动手实现一个 repeat 方法

```
function repeat(func, times, wait) {  
  // TODO  
}  
const repeatFunc = repeat(alert, 4, 3000);  
// 调用这个 repeatFunc ("hellworld"), 会alert4次 hellworld, 每次间隔3秒
```

分类: JavaScript

[答案&解析](#)

Day156:输入一个数字, 找到对应的字母

```
/*  
  如输入1 返回a  
  输入26返回z  
  输入27返回aa  
  输入28返回ab  
  输入53返回aaa  
*/
```

分类: JavaScript

[答案&解析](#)

Day157:输入一个数字, 找到对应的字母

公司: 微软

分类: 算法

[答案&解析](#)

Day158:要求用不同方式对 A 进行改造实现 A.name 发生变化时立即执行 A.getName

```
/*  
已知对象A = {name: 'sfd', getName: function(){console.log(this.name)}},  
现要求用不同方式对A进行改造实现A.name发生变化时立即执行A.getName  
*/
```

公司: 新东方

分类: JavaScript

[答案&解析](#)

Day159:如何把真实 dom 转变为虚拟 dom, 代码实现一下

公司: 高德

分类: JavaScript

[答案&解析](#)

Day160:如何把真实 dom 转变为虚拟 dom, 代码实现一下

公司: 头条、自如

分类: Css

[答案&解析](#)

Day161:说一下 koa2 和 express 区别

公司：菜鸟网络、海风教育

分类：Node

[答案&解析](#)

Day162:说一下 RN、Weex、Flutter 的优缺点

公司：腾讯应用宝、寺库

分类：其它

[答案&解析](#)

Day163:什么是微服务，微服务跟单体应用的区别是啥，用微服务有啥好处？

分类：其它

[答案&解析](#)

Day164:说一下 koa-body 的原理

公司：阿里

分类：Node

[答案&解析](#)

Day165:Node 如何和 MySQL 进行通信

公司：阿里

分类：Node

[答案&解析](#)

Day166:项目中如何处理安全问题

公司：阿里、喜马拉雅

分类：其它

[答案&解析](#)

Day167:修改代码不造成死循环

```
while (1) {  
  console.log(Math.random());  
}
```

分类：JavaScript

[答案&解析](#)

Day168:写出页面渲染值，并解释为什么？

```
<template>  
  <div>{{a.b}}</div>  
</template>  
  
<script>  
export default {  
  data () {  
    return {  
      a: {}  
    }  
  },  
  created () {  
    this.a.b = 1  
  },  
  mounted () {  
    this.a.b = 2  
  }  
}  
</script>
```

公司：头条

分类：Vue

[答案&解析](#)

Day169:怎么给 Vue 定义全局方法

分类：Vue

[答案&解析](#)

分类: JavaScript

Day171:请实现一个 JSON.stringify

分类: JavaScript

答案&解析

Day172:用 JavaScript 实现一个标准的排序算法(快排、冒泡、选择排序), 对某个数字数组进行由低到高的排序。

分类： 算法

答案&解析

Day173:说一下 HTTP2.0 多路复用有哪些优势?

分类：网络&安全

答案&解析

Day174:versions 是一个项目的版本号列表，因多人维护，不规则，动手实现一个版本号处理函数

```
var versions = ["1.45.0", "1.5", "6", "3.3.3.3.3.3"];  
// 要求从小到大排序，注意'1.45'比'1.5'大  
function sortVersion(versions) {  
    // TODO  
}  
// => ['1.5', '1.45.0', '3.3.3.3.3.3', '6']
```

公司：头条

分类: JavaScript

答案&解析

Day175:从输入 URL 到页面加载全过程

公司：头条、边锋、菜鸟网络、爱范儿、心娱

分类：JavaScript

[答案&解析](#)

Day176:怎么理解 to B 和 to C 的业务

公司：顺丰

分类：其它

[答案&解析](#)

Day177:说一下对 React Hook 的理解，它的实现原理，和生命周期有哪些区别？

公司：高德

分类：React

[答案&解析](#)

Day178:React 挂载的时候有 3 个组件，textComponent、composeComponent、domComponent，说一下它们的区别和关系，Dom 结构发生变化时怎么区分 data 的变化，怎么更新，更新怎么调度，如果更新的时候还有其他任务存在怎么处理

公司：滴滴

分类：React

[答案&解析](#)

Day179:Promise.allSettled 了解吗？动手实现一下 Promise.allSettled

公司：快手

分类：JavaScript

[答案&解析](#)

Day180:说一下平时项目是怎么优化的？优化之后是怎么度量的？首屏时间是怎么计算的？

公司：快手

分类：其它

[答案&解析](#)

Day181:说一下组件库设计遵循的一些原则?

公司：编程猫

分类：JavaScript

[答案&解析](#)

Day182:怎么计算组件在视口内出现了几次？IntersectionObserver 怎么使用的？怎么知道一个 DOM 节点出现在视口内？

公司：快手

分类：JavaScript

[答案&解析](#)

Day183：有 100 瓶水，其中有一瓶有毒，小白鼠只要尝一点带毒的水 3 天后就会死亡，至少要多少只小白鼠才能在 3 天内鉴别出哪瓶水有毒？

公司：富途

分类：其它

[答案&解析](#)

Day184:写一个方法，把 16 进制颜色转成 10 进制。(#fff => rgb(255,255,255))

公司：快手

分类：JavaScript

[答案&解析](#)

Day185:按要求完成代码

```
function lottery(whiteList, participant) {  
  // your code  
}  
// whiteList:类型字符串数组,意义是表示从其他系统中计算出来的活跃用户，如果这批用户参与  
抽奖，则必定让他中奖。长度不超过1万
```

```
// participant:类型字符串数组,意义是表示此次活动中真正参与抽奖的用户,长度约是10万。  
  
// 函数希望从participant返回 2 万个用户,表示中奖用户,优先选取whiteList上的用户,若  
// 不在whiteList上,对participant 剩余的随机 选取即可。
```

公司：斗鱼

分类：JavaScript

[答案&解析](#)

Day186:如何获取 html 元素实际的样式值？

公司：京东

分类：JavaScript

[答案&解析](#)

Day187:浏览器为什么要阻止跨域请求？如何解决跨域？每次跨域请求都需要到达服务端吗？

公司：快手

分类：网络&安全

[答案&解析](#)

Day188:DOM 树中，如何判断节点 A 是否包含于节点 B？

公司：京东

分类：JavaScript

[答案&解析](#)

Day189 写出代码正确执行结果，并解释原因

```
function test() {  
  console.log("out");  
}  
(function () {  
  if (false) {  
    function test() {  
      console.log("in");  
    }  
  }  
})
```

```
    }  
    test();  
  })();
```

分类：JavaScript

[答案&解析](#)

Day190:怎么获取相交链表的第一个相交点？

公司：斗鱼

分类：算法

[答案&解析](#)

Day191:移动端适配方案具体实现以及对比

公司：头条

分类：css

[答案&解析](#)

Day192:将一个按照升序排列的有序数组，转换为一棵高度平衡二叉搜索树。本题中，一个高度平衡二叉树是指一个二叉树每个节点的左右两个子树的高度差的绝对值不超过 1。

公司：头条

分类：算法

[答案&解析](#)

Day193：完善下面函数，实现图片的加载

```
function createImg(url) {}  
createImg(url).then((value) => {  
  document.body.appendChild(value);  
});
```

公司：头条

分类：JavaScript

答案&解析

Day194: `var arr = [['A','B'], ['a','b'], [1,2]]` 求二维数组的全排列组合 结果:
Aa1,Aa2,Ab1,Ab2,Ba1,Ba2,Bb1,Bb2

公司：美团

分类：算法

答案&解析

Day195:实现一个简单的仓储系统，可以不断转入和转出货物，货物最多有两层子类目，数字代表该子类目转入/转出的数量。转出时不能出现爆仓情况。

```
/*
 * cargo 说明:
 * key代表类目/子类目名称
 * value 为 number时，代表这个类目的数量，为object 时，代表下一层货物的集合，最多嵌套
两层
 * {
 *   productA:{ // 代表货物的类目名称
 *     a:1, // 1 代表子类目 a 的数量
 *     b:2,
 *     c:{ // c 代表货物的子类名称
 *       c1:1, // c1代表货物的子类名称
 *     }
 *   },
 *   productB:{
 *     e:6
 *   }
 * }
 *
 * 爆仓情况：如转入 {productA:{a:3,c:1}} 转出 {productA:{a:4}},就会发生子类目a爆
仓，此时要返回报错。
 */

class Depository {
  /**
   * @param options
   */
  constructor(options) {}
  // 转入货物
  transferIn(cargo) {}
  // 转出货物
  transferOut(cargo) {}
}
```

公司：支付宝

分类：JavaScript、编程题

[答案&解析](#)

Day196:有看过 V8 源码吗？说下 `v8::Isolate`？

公司：爱奇艺

分类：网络&安全

[答案&解析](#)

Day197:对 git flow 了解吗，应该如何使用

公司：360

分类：工程化

[答案&解析](#)

Day198:说一下 IoC 是啥，应用场景是啥？

分类：JavaScript

[答案&解析](#)

Day199:你有封装过 axios 吗？主要是封装哪些方面？如何中断 axios 请求？

分类：Vue

[答案&解析](#)

Day200:Git 的快照是指什么？Git 的工作区域由哪三部分组成？在 Git 中，如何为提交的版本打标签？

分类：工程化

[答案&解析](#)

Day201:说一下从获得 HTML 到页面渲染的全部流程？为什么栅格线程使用 GPU 计算而不是 CPU 计算？

公司： 淘宝

分类： 网络&安全

[答案&解析](#)

Day202:说下 v8 引擎解释执行 js 代码的大致流程

公司： 淘宝

分类： JavaScript

[答案&解析](#)

Day203:Vue-router history 模式部署的时候要注意什么？server 端用 nginx 和 node 时候分别怎么处理？

公司： 百度

分类： Vue

[答案&解析](#)

Day204:说一下 Vue3 的 Composition API

公司： 京东、好未来

分类： Vue

[答案&解析](#)

Day205:这里有个约定规则，实现一个方法 decodeStr，输入一个字符串，根据约定规则输出编码结果。约定规则如下：

```
//str = "2[a]1[bc]", 返回 "aabc".  
// str = "2[e2[d]]", 返回 "eddedd".  
// str = "3[abc]2[cd]ff", 返回 "abcabcabccdcdff".  
// 可以看出：N[string]，表示string 正好重复 N 次。假设字符串一定是有效正确的字符串；  
// 但是你需要考虑其他可能出现的入参错误场景。
```

公司： 腾讯微视

分类： JavaScript

[答案&解析](#)

Day206:秒 转换为 时分秒格式, 不能使用 js API

公司: 京东

分类: JavaScript

[答案&解析](#)

Day207:说一下 OAuth2.0 的登录详细流程

公司: 360 金融

分类: 网络&安全

[答案&解析](#)

Day208:模拟实现 Object.freeze

分类: JavaScript

[答案&解析](#)

Day209:介绍一下 node 核心模块, 并且详细介绍下 Stream

公司: 菜鸟网络、自如

分类: Node

[答案&解析](#)

Day210:项目中有没有涉及到 Cluster, 说一下你的理解

公司: 阿里

分类: Node

[答案&解析](#)

Day211:有没有写过 Koa 中间件, 说下中间件原理, 介绍下自己写过的中间件

公司: 阿里

分类: Node

[答案&解析](#)

Day212:说下对 AST（Abstract Syntax Tree）抽象语法树的理解？都有哪些用途？

公司: 滴滴

分类: JavaScript

[答案&解析](#)

Day213:如何设计一个 localStorage，保证数据的时效性？

公司: 有赞

分类: JavaScript

[答案&解析](#)

Day214:写出下面代码的输出结果，并解释为什么？

```
//counter.js
let counter = 10;
const add = () => {
  console.log(counter);
};
export { counter, add };

//index.js
add();
import { counter, add } from "./counter";
counter += 1;
console.log(counter);
```

公司:快手

分类: JavaScript

[答案&解析](#)

Day215 下面代码的输出是什么？为什么？


```
const person = { name: "yd" };
Object.defineProperty(person, "age", { value: 21 });
person.age = 18;
console.log(person);
console.log(Object.keys(person));
```

分类: JavaScript

[答案&解析](#)

Day216:说一下 escape、encodeURIComponent 有什么区别? 具体的使用场景?

公司: 水滴筹

分类: JavaScript

[答案&解析](#)

Day217:写出下面代码输出结果, 并解释为什么?

```
const value = { number: 10 };
let number = 2;
const multiply = (x = { ...value }, y = number) => {
  y++;
  console.log((x.number *= y));
};

multiply();
multiply();
multiply(value, number);
multiply(value, number);
```

[答案&解析](#)

Day218:说下下面两段代码执行情况

```
// 代码片段一: 是否存在堆栈溢出错误?
function foo() {
  setTimeout(foo, 0);
}
foo();
```

```
// 代码片段二:如果在控制台中运行以下函数, 页面(选项卡)的 UI 是否仍然响应?  
function foo() {  
  return Promise.resolve().then(foo);  
}
```

[答案&解析](#)

Day219:说下 prototype 和 __proto__ 的关系

分类: JavaScript

[答案&解析](#)

Day220:了解函数式编程中的 compose 吗? 动手实现一下?

分类: JavaScript

[答案&解析](#)

Day221:说下 Webpack Runtime 和 Manifest 代码的作用?

分类: 工程化

[答案&解析](#)

Day222:说一下 ES6 对 Object 类型做了哪些优化更新?

分类: JavaScript

[答案&解析](#)

Day223:使用队列来实现击鼓传花

击鼓传花的规则:

几个朋友一起玩一个游戏, 围成一圈, 开始数数(数数时数字都是挨着的, 第一个人数1, 之后的人数2), 数到某个数字的人自动淘汰。汰的人后面再从1开始数, 重复上面的游戏, 直到就剩下一个人, 请问留下了谁

```
// 测试例子  
var names = ["John", "Jack", "Camila", "Ingrid", "Carl"];
```

```
var endName = passGame(names, 8); // 数到 8 的人淘汰
console.log("最终留下:" + endName); // 'john'
```

分类：算法

[答案&解析](#)

Day224:多种方式实现圆形可点击区域(至少三种)

分类：JavaScript

[答案&解析](#)

Day225:介绍 instanceof 原理并实现

分类：JavaScript

[答案&解析](#)

Day226:用 reduce 实现 map 方法

分类：JavaScript

[答案&解析](#)

Day227:Webpack 如果使用了 hash 命名，是否每次都会重新生成 hash?简单说下 Webpack 的几种 hash 策略?

分类：工程化

[答案&解析](#)

Day228:Token 一般是存放在哪里?Token 放在 cookie 和放在 localStorage、sessionStorage 中有什么不同?

分类：网络&安全

[答案&解析](#)

Day229:说一下网页扫码登录的原理

分类： 其它

[答案&解析](#)

Day230:说一下 Vue 开发如何针对搜索引擎做 SEO 优化

分类： Vue

[答案&解析](#)

Day231:说一下你的 vue 项目目录结构，如果是大型项目你该怎么划分结构和划分组件呢？

分类： 工程化

[答案&解析](#)

Day232:详细说一下 JSON.stringify 的一些特性？和遍历相比，哪个性能高？

分类： JavaScript

[答案&解析](#)

Day233:Vuex 的 action 和 mutation 的特性是什么？有什么区别？Vuex 怎么知道 state 是通过 mutation 修改还是外部直接修改的？

分类： Vue

[答案&解析](#)

Day234:Vue-cli 默认是单页面的，如果要开发多页面应该怎么办？

分类： Vue

[答案&解析](#)

Day235:动手实现一下数组 Reduce 方法的 polyfill

分类： JavaScript

[答案&解析](#)

Day236:了解 HTTP3.0 吗? 简要说一下 HTTP 的一个发展历程?

分类: 网络&安全

[答案&解析](#)

Day237:说一下什么是运营商劫持? 有什么预防措施?

分类: 网络&安全

[答案&解析](#)

Day238:请介绍一下装饰者模式, 并实现

分类: JavaScript

[答案&解析](#)

Day239:按要求实现 go 函数

```
// 示例
go("l"); //gol
go()("l"); //gool
go()()()("l"); //返回gool
```

分类: JavaScript

公司: 深信服

[答案&解析](#)

Day240:写出输出值并解释为什么?

```
let a = 1,
    b = 2,
    head = { next: { next: 1 } };
[a, b] = [b, a];
[head.next, head.next.next] = [head.next.next, head.next];
console.log(a, b, head);
```

分类: JavaScript

[答案&解析](#)

Day241:说一下深拷贝如何解决循环引用问题？

公司：极光推送

分类：JavaScript

[答案&解析](#)

Day242:实现一个函数 `findLastIndex()`, 返回指定数在“有序”数组中最后一次出现位置的索引。如 `findLastIndex([1,2,3,3,3,4,5], 3)`, 返回 4。时间复杂度是多少？什么情况下时间复杂度最高？

公司：快手

分类：算法

[答案&解析](#)

Day243:WebWorker 的缺点是什么？在 worker 线程怎么获取主线程上下文？解决卡顿的问题除了使用 WebWorker 还有其他的解决方案吗？

公司：快手

分类：JavaScript

[答案&解析](#)

Day244:虚拟列表是什么？说一下它的实现原理？

公司：快手

分类：JavaScript

[答案&解析](#)

Day245:说一下请求是异步的为什么会造成阻塞？

公司：易车

分类：网络&安全

[答案&解析](#)

Day246:详细说一下你对 cookie、session、token 的理解

分类：网络&安全

[答案&解析](#)

Day247:说一下对纯函数的理解

公司：有赞

分类：JavaScript

[答案&解析](#)

Day248:说一下 MySQL 和 MongoDB 区别？MongoDB 有哪些缺点和优点？

分类：其它

[答案&解析](#)

Day249:说一下 Generator 的实现原理

分类：JavaScript

[答案&解析](#)

Day250:说一下 let、const 的实现，使用 ES5 模拟实现一下

分类：JavaScript

[答案&解析](#)

Day251:详细说一下对 JavaScript 执行上下文的理解

分类：JavaScript

[答案&解析](#)

Day252:说一下递归和迭代的区别是什么，各有什么优缺点？

分类：JavaScript

[答案&解析](#)

Day253:说一下 JavaScript 的宿主对象和原生对象的区别？为什么扩展 JavaScript 内置对象不是好的做法？

分类：JavaScript

[答案&解析](#)

Day254:请解释下面代码为什么不能用作 IIFE？需要作出哪些修改才能使其成为 IIFE？

```
// 代码一
function (){}()
// 代码二
function foo(){}()
```

分类：JavaScript

[答案&解析](#)

Day255:将输入的数组组装成一颗树状的数据结构，要求程序具有侦测错误输入的能力

```
let dataArr = [
  { id: 1, name: "i1" },
  { id: 2, name: "i2", parentId: 1 },
  { id: 4, name: "i4", parentId: 3 },
  { id: 3, name: "i3", parentId: 2 },
  { id: 8, name: "i8", parentId: 7 },
];
```

公司：阿里

分类：算法

[答案&解析](#)

Day256:将二维数组转成一维数组的方法有哪些？

分类：JavaScript

公司：微店

[答案&解析](#)

Day257:在不改变原数组的前提下，添加或删除某个元素，获得一个新数组(多种方式实现)

分类：JavaScript

公司：有赞

[答案&解析](#)

Day258:聊聊你对 Orinoco 认知

分类：JavaScript

[答案&解析](#)

Day259:按要求实现代码

```
var items = [{ name: "item1" }, { name: "item2" }];
var str = '<div ali-for="item in items">{{item.name}}</div>';

ParseDom(str);
// <div>item1</div><div>item2</div>

function ParseDom(str) {
  // your code
}
```

分类：JavaScript

公司：阿里

[答案&解析](#)

Day260:说下什么是 Reflect？有什么作用？你对元编程是如何理解的？有没有实际应用的例子？

分类：JavaScript

[答案&解析](#)

Day261:说一下 webpack4.x 模块化加载原理?(CommonJS、ES6 module、动态加载)

分类：工程化

[答案&解析](#)

Day262:为什么 WeakMap 和 WeakSet 的键只能使用对象？在什么场景下使用？

分类：JavaScript

[答案&解析](#)

Day263:请写出以下代码结果，并解释为什么？

```
let { ...yideng } = Object.create({ x: 1 });  
console.log(yideng.x);
```

分类：JavaScript

[答案&解析](#)

Day264:请实现 uncurrying 完成函数柯里化

```
var yideng = Array.prototype.push.uncurrying();  
(function () {  
  yideng(arguments, 4);  
  console.log(arguments); // [1,2,3,4]  
})(1, 2, 3);
```

分类：JavaScript

[答案&解析](#)

Day265:了解过 RPC 吗，简单说一下？

分类：其它

[答案&解析](#)

Day266:写出下面代码结果，并解释为什么？

```
function test() {}  
console.log(test.prototype.constructor.constructor);
```

分类：JavaScript

[答案&解析](#)

Day267:请写出执行结果,并解释原因

```
var yideng = {  
  n: 1,  
};  
yideng.x = yideng = {  
  n: 2,  
};  
console.log(yideng.x);
```

分类：JavaScript

[答案&解析](#)

Day268:请修改代码跳出死循环

```
while (1) {  
  switch ("yideng") {  
    case "yideng":  
      // 禁止只写一句break;  
    }  
}
```

分类：JavaScript

[答案&解析](#)

Day269:给你两个有序整数数组 nums1 和 nums2，请你将 nums2 合并到 nums1 中，使 nums1 成为一个有序数组。

```
// 说明：初始化 nums1 和 nums2 的元素数量分别为 m 和 n 。 你可以假设 nums1 有足够的  
// 空间（空间大小大于或等于 m + n）来保存 nums2 中的元素。
```

```
// 输入:
var nums1 = [1, 2, 3],
    m = 3;
var nums2 = [2, 5, 6],
    n = 3;
// 输出: [1,2,2,3,5,6]
```

分类：算法

[答案&解析](#)

Day270:给你一个包含 n 个整数的数组 `nums`，判断 `nums` 中是否存在三个元素 a, b, c ，使得 $a + b + c = 0$ ？请你找出所有满足条件且不重复的三元组。

```
// 示例
给定数组 nums = [-1, 0, 1, 2, -1, -4]， 满足要求的三元组集合为： [ [-1, 0, 1],
[-1, -1, 2] ]
```

分类：算法

[答案&解析](#)

Day271:给定一个字符串 `s`，最多删除一个字符。判断是否能成为回文字符串。

```
/*
输入: "aba"
输出: True
输入: "abca"
输出: True
注意：字符串只包含从 a-z 的小写字母。字符串的最大长度是50000。
*/
```

分类：算法

[答案&解析](#)

Day272:设计一个支持两种操作的数据结构

```
addWord(word) // 添加字符串
search(word)   // 返回布尔值，是否存在
//search可以搜索文字或正则表达式字符串，字符串只包含字母 。 或 a-z 。
```

```
// . 可以表示任何一个字母。

// 示例
addWord("bad")
addWord("dad")
addWord("mad")
search("pad") -> false
search("bad") -> true
search(".ad") -> true
search("b..") -> true
// 可以假设所有单词都是由小写字母 a-z 组成的。
```

分类：算法

[答案&解析](#)

Day273:写出执行结果，并解释为什么？

```
<script>
  // 使用未定义的变量
  yideng;
  console.log(1);
</script>
<script>
  console.log(2);
</script>
```

分类：JavaScript

[答案&解析](#)

Day274:设计 AutoComplete 组件(又叫搜索组件、自动补全组件等)时，需要考虑什么问题？

公司：头条

[答案&解析](#)

Day275:输入两个递增排序的链表，合并这两个链表并使新链表中的节点仍然是递增排序的。

```
//输入：1->2->4, 1->3->4
//输出：1->1->2->3->4->4
```

分类：算法

[答案&解析](#)

Day276:给定一个排序链表，删除所有含有重复数字的结点，只保留原始链表中没有重复出现的数字。

```
/*
输入：1->2->3->3->4->4->5
输出：1->2->5

输入：1->1->1->2->3
输出：2->3
*/
```

分类：算法

[答案&解析](#)

Day277:给定一个链表，删除链表的倒数第 n 个结点，并且返回链表的头结点。

```
// 给定一个链表：1->2->3->4->5，和 n = 2。
// 当删除了倒数第二个结点后，链表变为 1->2->3->5。
// 说明：给定的n保证是有效的
```

分类：算法

[答案&解析](#)

Day278:定义一个函数，输入一个链表的头结点，反转该链表并输出反转后链表的头结点。

```
//输入：1->2->3->4->5->NULL
//输出：5->4->3->2->1->NULL
```

分类：算法

[答案&解析](#)

Day279:反转从位置 m 到 n 的链表。请使用一趟扫描完成反转。

```
// 说明:  $1 \leq m \leq n \leq$  链表长度。  
// 示例:  
// 输入: 1->2->3->4->5->NULL, m = 2, n = 4  
// 输出: 1->4->3->2->5->NULL
```

分类: 算法

[答案&解析](#)

Day280:给定一个链表, 如何判断链表是否有环?

```
// 如果链表中有某个节点, 可以通过连续跟踪 next 指针再次到达, 则链表中存在环。  
// 如果链表中存在环, 则返回 true 。 否则, 返回 false 。
```

分类: 算法

[答案&解析](#)

Day281:给定一个链表, 返回链表开始入环的第一个结点。 如果链表无环, 则返回 null。

分类: 算法

[答案&解析](#)

Day282:根据每日气温列表, 请重新生成一个列表, 对应位置的输出是需要再等待多久温度才会升高超过该日的天数。如果之后都不会升高, 请在该位置用 0 来代替。

```
// 例如, 给定一个列表 temperatures = [73, 74, 75, 71, 69, 72, 76, 73], 你的输出  
// 应该是 [1, 1, 4, 2, 1, 1, 0, 0]。  
// 提示: 气温 列表长度的范围是 [1, 30000]。每个气温的值的均为华氏度, 都是在 [30, 100]  
// 范围内的整数。
```

分类: 算法

[答案&解析](#)

Day283:设计一个支持 push , pop , top 操作, 并能在常数时间内检索到最小元素的栈。

```
push(x); //将元素 x 推入栈中。
pop(); // 删除栈顶元素
top(); // 获取栈顶元素
getMin(); // 检索栈中最小的元素

// 示例
MinStack minStack = new MinStack();
minStack.push(-2);
minStack.push(0);
minStack.push(-3);
minStack.getMin(); // --> 返回 -3
minStack.pop();
minStack.top(); // --> 返回 0
minStack.getMin(); // --> 返回 -2
```

分类：算法

[答案&解析](#)

Day284:如何用栈实现一个队列

```
push(x) // 将一个元素放入队列的尾部。
pop() // 队列首部移除元素。
peek() // 返回队列首部的元素。
empty() // 返回队列是否为空。

// 示例
MyQueue queue = new MyQueue();
queue.push(1);
queue.push(2);
queue.peek(); // 返回 1
queue.pop(); // 返回 1
queue.empty(); // 返回 false

// 说明
// 只能使用标准的栈操作 -- 也就是只有 push to top, peek/pop from top, size, 和
is empty 操作是合法的。
// 假设所有操作都是有效的 （例如，一个空的队列不会调用 pop 或者 peek 操作）。
```

分类：算法

[答案&解析](#)

Day285:给定一个数组 nums 和滑动窗口的大小 k，请找出所有滑动窗口里的最大值。


```
// 示例
// 输入:
// nums = [1, 3, -1, -3, 5, 3, 6, 7], k = 3;
// 输出:
// [3, 3, 5, 5, 6, 7];
```

分类：算法

[答案&解析](#)

Day286: 给定一个没有重复数字的序列，返回其所有可能的全排列。

```
/**
示例
输入: [1,2,3]
输出: [
  [1,2,3],
  [1,3,2],
  [2,1,3],
  [2,3,1],
  [3,1,2],
  [3,2,1]
]
*/
```

分类：算法

[答案&解析](#)

Day287: 给定一组不含重复元素的整数数组 nums，返回该数组所有可能的子集（幂集）。说明：解集不能包含重复的子集。

```
// 输入:  nums = [1,2,3]
/*
输出
[
  [3],
  [1],
  [2],
  [1,2,3],
  [1,3],
  [2,3],
  [1,2],
  []
]
```

```
]
*/
```

分类： 算法

[答案&解析](#)

Day288:给定两个整数 n 和 k，返回 1 ... n 中所有可能的 k 个数的组合。

```
/*
输入： n = 4,k = 2;
输出：
[
  [2,4],
  [3,4],
  [2,3],
  [1,2],
  [1,3],
  [1,4],
]
*/
```

分类： 算法

[答案&解析](#)

Day289:项目中如何应用数据结构？

公司： 挖财

分类： JavaScript

[答案&解析](#)

Day290:node 如何做错误监控？ 如何生成日志？ 日志等级如何划分？

公司： 自如

分类： Node

[答案&解析](#)

Day291:说一下 JavaScript 的执行流程

分类：JavaScript

[答案&解析](#)

Day292:如何设计状态树

公司：挖财

分类：React

[答案&解析](#)

Day293:说下在项目开发你是怎么组织 CSS 的？（Webpack）

分类：工程化

[答案&解析](#)

Day294：说一下进程和线程的区别？

分类：其它

[答案&解析](#)

Day295:详细描述一个 http 请求从发起请求到收到响应的全部过程(越细越好)

分类：网络&安全

[答案&解析](#)

Day296:下面代码是否会导致堆栈溢出错误

```
function foo() {  
  setTimeout(foo, 0);  
}  
foo();  
// 是否存在堆栈溢出错误
```

分类：JavaScript

[答案&解析](#)

Day297:说下 Node 文件查找的优先级以及 Require 方法的文件查找策略

分类：Node

分类：JavaScript

[答案&解析](#)

Day298:手写一个 Ajax

分类：JavaScript

[答案&解析](#)

Day299:说一下什么是功能检测（feature detection）、功能推断（feature inference）、UA 字符串？

分类：JavaScript

[答案&解析](#)

Day300 Get 请求传参长度存在限制，是 HTTP 协议限制的吗？

分类：网络&安全

[答案&解析](#)

Day301:鼠标滚动的时候，会触发很多次事件，你是如何解决的？

分类：JavaScript

[答案&解析](#)

Day302:实现 10 进制转换为 36 进制

分类：JavaScript

公司：头条

[答案&解析](#)

Day303:什么时候用 TCP，什么时候适合 UDP？

分类：网络&安全

公司：腾讯

[答案&解析](#)

Day304:什么情况下会触发 options 请求

分类：网络&安全

[答案&解析](#)

Day305:说一下 Taro 编译的原理？

分类：其它

公司：阿里

[答案&解析](#)

Day306:Vue2.x 和 Vue3.x 渲染器的 diff 算法分别说一下？

分类：Vue

[答案&解析](#)

Day307:如果一个页面突然出现了一段广告，可能是什么原因，怎么解决？

分类：网络&安全

公司：腾讯

[答案&解析](#)

Day308:说下你对 ESLint 的了解？ 如何使用？ 它的工作原理？

分类：工程化

公司：美团

[答案&解析](#)

Day309:说下你对 Node.js 支持高并发的理解?

分类: Node

[答案&解析](#)

Day310:了解 WebSocket 吗? WebSocket 是如何进行握手的?

公司: 腾讯

分类: 网络&安全

[答案&解析](#)

Day311:介绍下 npm 模块安装机制? 输入 npm install 命令敲下回车后它的一个执行流程是怎样的?

分类: 工程化

[答案&解析](#)

Day312:接口怎么防刷

分类: 网络&安全

[答案&解析](#)

Day313:说下你对前端工程化的理解

分类: 工程化

[答案&解析](#)

Day314:ES6 代码转成 ES5 代码的实现思路是什么? 大致说一下 babel 原理?

分类: JavaScript

[答案&解析](#)

Day315:请说说你对 Sparkplug 的理解?

分类: JavaScript

[答案&解析](#)

Day316:说一下实现骨架屏的方案？具体思路？

分类：工程化

[答案&解析](#)

Day317:完成一个转换函数，将数字转换成对应的大写字母，满足下面的对应关系

```
// 1 => A
// 2 => B
// 3 => C
// ...
// 26 => Z
// 27 => AA
// 28 => AB
// ...
// 52 => AZ
// 53 => BA
// 54 => BB

// 实现下方函数
function convert(){
  // todo
}

// 测试代码
const output1 = convert(1);
console.log(output1); //A
const output2 = convert(26);
console.log(output2); // Z
```

公司：腾讯微视

分类：JavaScript

[答案&解析](#)

Day318:WebSocket 如何断开重连

分类：JavaScript

[答案&解析](#)

Day319:为什么说利用多个域名来存储网站资源会更有效?

分类: 网络&安全

公司: 字节跳动

[答案&解析](#)

Day320:文件上传如何实现? 除了 input 还有哪些别的方法?

公司: 字节跳动

分类: JavaScript

[答案&解析](#)

Day321:请设计一个方案: 有 a、b、c 三个 npm 插件, 它们会经常更新, 在前端项目 npm run start 启动后, 要求 a、b、c 三个 npm 插件自动更新到最新版本

分类: 工程化

[答案&解析](#)

Day322:手写发布订阅模式

分类: JavaScript

[答案&解析](#)

Day323:大数计算如何实现

分类: JavaScript

[答案&解析](#)

Day324:说一下网络通信中引入滑动窗口的作用, 它的机制原理是怎样的?

公司: 淘宝

分类: 网络&安全

[答案&解析](#)

Day325:请求在客户端报 413 是什么错误，怎么解决？

公司：淘宝

分类：网络&安全

[答案&解析](#)

Day326:请用 ES5 模拟实现 Number 的 isInteger 方法

分类：JavaScript

[答案&解析](#)

Day327:谈一下微信小程序的架构以及为什么要使用双线程？

分类：其它

[答案&解析](#)

Day328:给定一个数组，按找到每个元素右侧第一个比它大的数字，没有的话返回-1 规则返回一个数组

分类：算法

[答案&解析](#)

Day329:说一说 promise，有几个状态，通过 catch 捕获到 reject 之后，在 catch 后面还能继续执行 then 方法嘛，如果能执行执行的是第几个回调函数？

公司：伴鱼、喜马拉雅

分类：JavaScript

[答案&解析](#)

Day330:为什么 React useState 要使用数组而不是对象？

分类：React

[答案&解析](#)

Day331:说一下对称加密和非对称加密的区别和用处?

分类: 网络&安全

[答案&解析](#)

Day332:手写代码实现`kuai-shou-front-end=>KuaiShouFrontEnd`

分类: JavaScript

[答案&解析](#)

Day333:用尽量短的代码实现一个 array 的链式操作, 将数组中的大于 10 的值进行一个累加

分类: JavaScript

[答案&解析](#)

Day334:按要求实现 rightView 函数

```
function TreeNode(val){
  this.val = val;
  this.left = null;
  this.right = null;
}
function rightView(root){
  // 请你实现
}
// => [1,4,3]
    1      => 1
    2  4    => 4
    7  3    => 3
```

公司: 头条

分类: 算法

[答案&解析](#)

Day335: 动手实现一下 Node utils 模块中的 Promisify 函数

分类: Node

[答案&解析](#)

Day336:说一下 jQuery ajax、axios、fetch 的区别

分类：网络&安全

[答案&解析](#)

Day337:输入一个日期,返回几秒前、几小时前、几天前、几月前

分类：JavaScript

[答案&解析](#)

Day338:说一下你对函数式编程的理解

分类：JavaScript

[答案&解析](#)

Day339：说出前端框架设计模式(MVVM 或 MVP 又或 MVC)的含义以及原理

公司：玄武科技

分类：工程化

[答案&解析](#)

Day340:动手实现一个简单的 mvvm 模型

分类：JavaScript

[答案&解析](#)

Day341:说下 LRU 算法的原理并手写实现？一般有哪些优化方式？

公司：字节跳动、京东

分类：算法

[答案&解析](#)

Day342:有没有搭建过 CI/CD？说下 CI/CD 搭建的流程

公司：爱奇艺

分类：工程化

[答案&解析](#)

Day343:说一下 TypeScript 中 Partial、Pick、Record 作用？

分类：其它

[答案&解析](#)

Day344:封装公共组件需要注意什么？

分类：工程化

[答案&解析](#)

Day345:Prerender 预渲染是什么原理？预渲染 prerender-spa-plugin 能详细讲讲吗？

公司：盒马

分类：Vue、React

[答案&解析](#)

Day346: Node 中怎么开启一个子线程

分类：Node

[答案&解析](#)

Day347:说一下小程序页面的通讯方式

分类：其它

[答案&解析](#)

Day348:class 的继承和 prototype 继承 是完全一样的吗？

分类：JavaScript

[答案&解析](#)

Day349:说一下 vite 和 webpack 对比, 为什么 vite 在 dev 模式下运行速度快很多?

分类: 工程化

[答案&解析](#)

Day350:Reflect.ownKeys 与 Object.keys 的区别

分类: JavaScript

[答案&解析](#)

Day351:说一下对事件流的理解,如果 addEventListener 的第三个参数设置为 true 会发生什么?

分类: JavaScript

[答案&解析](#)

Day352: React Hooks 的 useState 相对于有状态组件 state 区别是什么?

分类: React

[答案&解析](#)

Day353:React Hooks 解决了什么问题? 其原理是什么(以 useEffect 为例)? useMemo 和 useCallback 差别?

分类: React

[答案&解析](#)

Day354:模拟实现 Symbol

分类: JavaScript

[答案&解析](#)

Day355:下面代码在 vue 中输出什么? 为什么?

```
for (let i = 0; i < 10; i++) {  
  this.a = i;  
  this.$nextTick(() => {  
    console.log(this.a);  
  });  
}
```

公司：头条

分类：Vue

[答案&解析](#)

Day356:说说 Context 有哪些属性？怎么使用 Context 开发组件？除了实例的属性可以获取 Context 外哪些地方还能直接获取 Context？为什么 React 并不推荐我们优先考虑使用 Context？

分类：React

[答案&解析](#)

Day357:说一下你觉得的 vue 开发规范有哪些？

分类：工程化

[答案&解析](#)

Day358:说一下你对 vue keep-alive 的理解？以及使用过程中需要注意的地方？

分类：Vue

[答案&解析](#)

Day359:你知道 Vue 的模板语法用的是哪个 web 模板引擎的吗？说说你对这模板引擎的理解？

分类：Vue

[答案&解析](#)

Day360:Vue2.x 为什么要求组件模板只能有一个根元素？

分类：Vue

[答案&解析](#)

Day361:Vue 切换路由时，需要保存草稿的功能，怎么实现？

分类：Vue

[答案&解析](#)

Day362:说一下 Vue 中 EventBus 的使用？在使用过程中有没有遇到重复触发的问题？如何解决的？

分类：Vue

[答案&解析](#)

Day363:举例说明 React 的插槽有哪些运用场景？你有用过 React 的插槽(Portals)吗？怎么用？

分类：React

[答案&解析](#)

Day364：为什么说 React 中 props 是只读的？

分类：React

[答案&解析](#)

Day365:如何解决 Vue 打包 vendor 过大的问题？Webpack 打包 vue 速度慢怎么办？

分类：Vue

[答案&解析](#)

Day366:什么是 CSP？

分类：网络&安全

[答案&解析](#)

Day367:说下 URL 和 URI 的区别？

分类：网络&安全

[答案&解析](#)

Day368:说一下你对 Get 请求传参长度限制的理解？

分类：网络&安全

[答案&解析](#)

Day369:说下你对 Reflect 的理解？为什么会有 Reflect 的出现？Proxy 也简单的说一下？

分类：JavaScript

[答案&解析](#)

Day370:为什么使用 setTimeout 实现 setInterval？怎么模拟？

分类：JavaScript

[答案&解析](#)

Day371:使用 Symbol 函数都有哪些要注意的点？

分类：JavaScript

[答案&解析](#)

Day372:说一下你对 Get 和 Post 请求在缓存方面理解

分类：网络&安全

[答案&解析](#)

Day373：说下 Node 中 require 模块引入的查找方式？

分类：Node

[答案&解析](#)

Day374:说一下开发中常用的几种 Content-Type ?

分类：网络&安全

[答案&解析](#)

Day375:setTimeout/setInterval 实现倒计时如何解决时间偏差的问题?

分类：JavaScript

[答案&解析](#)

Day376:说一下你对进程和线程的了解? Node 中进程间是如何进行通信的?

分类：Node

[答案&解析](#)

Day377:说一下你了解的 WebSocket 鉴权授权方案?

分类：其它

[答案&解析](#)

Day378:说一下什么是 Http 协议无状态?怎么解决 Http 协议无状态?

分类：网络&安全

[答案&解析](#)

Day379:如果让你来实现一个前端监控系统，应该考虑什么？如何去实现？

分类：工程化

[答案&解析](#)

Day380:实现一个系统，统计前端页面性能、页面 JS 报错、用户操作行为、PV/UV、用户设备等消息，并进行必要的监控报警。方案如何设计，用什么技术点，什么样的系统架构，难点会在哪里？

公司： 阿里云

分类： 工程化

[答案&解析](#)

Day381:说一下 JavaScript 严格模式下有哪些不同？

分类： JavaScript

[答案&解析](#)

Day382:说一下 setTimeout 和 setInterval 的区别，包含内存方面的分析？

分类： JavaScript

[答案&解析](#)

Day383:说下 React 的 useEffect、useCallback、useMemo

分类： React

[答案&解析](#)

Day384:JavaScript 中如何实现一个类？怎么实例化这个类？

分类： JavaScript

[答案&解析](#)

Day385:在一个 DOM 上同时绑定两个点击事件：一个用捕获，一个用冒泡。事件会执行几次？先执行冒泡还是捕获？

分类： JavaScript

[答案&解析](#)

Day386:写一个通用的事件侦听器函数

分类： JavaScript

[答案&解析](#)

