1. 编写一个 People 类，使其的实例具有事件绑定、触发事件、解除绑定功能。（实例可能监听多个不同的事件，也可以去除监听事件）

/\*

\* TODO

\*

\*/

People.prototype.sayHi = function () {

console.log('Hi, I am ' + this.name)

}

// 目标效果：

var jerry = new People('Jerry')

jerry.sayHi() // => 输出：'Hi, I am Jerry'

jerry.on('greeting', say)

jerry.on('greeting', say2)

jerry.emit('greeting', 'Hi') // => 输出：'Hi, nice meeting you.' 和 'Hi, nice meeting you, too'

jerry.off('greeting', say)

jerry.emit('greeting', 'Hi') // => 只输出：'Hi, nice meeting you, too'

function say (greeting) {

console.log(greeting + ', nice meeting you.')

}

function say2 (greeting) {

console.log(greeting + ', nice meeting you, too')

}

1. 找出下面代码的规律并且编写一个函数，转换特定的整数到对应的字符串。

1 => A

2 => B

3 => C

...

26 => Z

27 => AA

28 => AB

29 => AC

...

52 => AZ

53 => BA

...

function convert (num) {

// TODO

}

1. 完成 compose 函数。它接受任意多个单参函数（只接受一个参数的函数）作为参数，并且返回一个函数。它的作为用：使得类似 f(g(h(a)))这样的函数调用可以简写为 compose(f, g, h)(a)。

function compose () {

// TODO

}

// 测试代码

function addOne (a) { return a + 1 }

function multiTwo (a) { return a \* 2 }

function divThree (a) { return a / 3 }

function toString (a) { return a + '' }

function split (a) { return a.split('') }

split(toString(addOne(multiTwo(divThree(666))))) // => ["4", "4", "5"]

var testForCompose = compose(split, toString, addOne, multiTwo, divThree)

testForCompose(666) // => ["4", "4", "5"]

1. 如果现在页面上出现了卡顿现象，例如鼠标移动不流畅，页面滚动不流畅，你会如何进行debug？

答：

1. HTTP是无状态协议，那么网站应该如何记住用户的登录状态？尽量详细回答整个过程。

答：

1. 尽量多地列出你所知道的关于前端的技术（方法、框架等），并且简述其作用。

答：