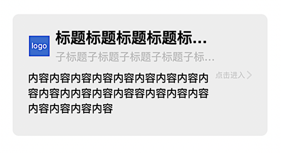
**前端工程师笔试题**

1. 总结描述一下CSS盒模型，如何改变盒模型：

您的回答：

2. 用Flex布局实现如下功能（要求代码精简，布局准确，尺寸与颜色随意即）：  


您的回答：

3. 以下JS代码有什么问题?尽可能多的列出解决办法。  
function setStyle(element) {  
element.style.fontWeight = "bold";   
element.style.textDecoration = "none";   
element.style.color = "#000000";   
}

4. 描述js中new操作符具体进行了什么操作

New 四步：

1.0 创建一个对象

2.0 this 指向这个对象

3.0 给对象赋值

4.0 返回这个对象

5. 请写出以下几个alert语句的输出结果:  
function ClassA (arg) { this.arg = arg; }  
ClassA.prototype.x = 20; function ClassB(x) { this.x = x; }   
ClassB.prototype = new ClassA(30);  
var a = new ClassA('xxxxx'), b = new ClassB(40); alert (a instanceof Object);  
alert(a instanceof ClassA);  
alert(b instanceof Object);  
alert (b instanceof ClassA);  
a.x = 30;  
alert (a.x);  
delete a.x  
alert (a.x);  
alert(b.x);  
delete b.x;  
alert (b.x);

您的回答：

6. 用JS实现一个函数解析网址中的查询值，例如：输入字符串 https://www.qq.com/query?id=news&name=search&ref=website，输出对象 { id: 'news', name: 'search', ref: 'website' }

您的回答：

7. 利用Promise和async/await实现以下功能：  
现有一个接口https://test.com/get?id=1，需要发起id从1~200的GET方式调用200次异步请求，要求分20轮发出，前一轮全部完成后才进行下一轮，每轮同时并发请求为10次。

您的回答：

co*nst axios = require('axios')*

*// 将所有的网络请求封装为一个方法，方法可以得到一个 promise 对象*

*function sendMsg (num) {*

*return axios({*

*url: `https://test.com/get?id=${num}`*

*})*

*}*

*// 发送请求： num 第 num 轮*

*function time (num) { // 1 =>*  1 ~ 10 2 => 11 ~ 20 3 => 21 ~ 30

  var arr = []

  for (var i = 10 \* (num - 1) + 1; i <= 10 \* num; i++) { // 一共遍历 10 次

    arr.push(sendMsg(i))

  }

  return Promise.all(arr) // Promise.all 当这里所有的 Promise 中的异步请求执行完成之后再向后执行

}

async function start () {

  for (var i = 1; i <= 20; i++) {

    var p = await time(i)

    console.log('---------------------')

    p.forEach(item => {

      console.log(item.data)

    })

    console.log('---------------------')

  }

}

start()

8. 链表翻转。给出一个链表和一个数k，比如链表1→2→3→4→5→6，k=2，则翻转后 2→1→4→3→6→5，若k=3,翻转后3→2→1→6→5→4，若k=4，翻转后4→3→2→1→5→6，用程 序实现。可以用任何语言实现。

您的回答：

var str = '1→2→3→4→5→6'

function reveser (num) {

  var arr = str.split('→').join('')

  var res = ''

  for (var i = 0; i < arr.length; i += num) {

    var item = arr.slice(i, i + num)

    if (item.length == num) {

      res += item

        .split('')

        .reverse()

        .join('')

    } else {

      res += item

    }

  }

  console.log(res.split('').join('→'))

}

reveser(5)