



Web前端开发高级模拟测试（九）

64分

共54道题 总分：200分 形考总分：0分

形考：0分

单选题 多选题 判断题 客观填空题

一、单选题 共30题，60分

1

2分

nodejs的哪个模块可以搭建服务器？

- ☒ A Http
- ☐ B url
- ☐ C path
- ☐ D fs

作答结果：正确

打开解析 □

A

单选题 (30题, 60分)

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30		

多选题 (15题, 30分)

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	

判断题 (5题, 10分)

1	2	3	4	5
---	---	---	---	---

客观填空题 (4题, 100分)

1	2	3	4
---	---	---	---

☒ 正确 ☐ 错误 ☐ 批阅 ☐ 未做

2

2分

关于Iterator遍历器的说法，错误的是：（）

- ☐ A next()方法是Iterator遍历器的核心
- ☐ B 当next()返回对象的done属性为false，遍历结束
- ☒ C 具有Iterator接口的对象视为可遍历对象
- ☐ D 可以自定义一个可遍历对象和其遍历行为

作答结果：错误

打开解析 □

B

3

2分

执行[1,2,3].fill(4)的结果是：（）

- ☐ A [4]
- ☐ B [1,2,3,4]

2

Ⓒ [4,1,2,3]

Ⓓ [4,4,4]

作答结果：正确

打开解析 ☐

D

4

2分

关于canvas不正确的是()

Ⓐ canvas是HTML5新增的标签，允许脚本动态渲染位图像

Ⓑ canvas是作为Flash的替代技术出现

Ⓒ canvas拥有多种绘制路径、矩形、圆形、字符及添加图像的方法

Ⓓ ie9起已经可以支持canvas

作答结果：正确

打开解析 ☐

B

5

2分

表单元素要加上什么类，才能给表单添加圆角属性和阴影效果（）

Ⓐ form-group

Ⓑ form-horizontal

Ⓒ form-inline

Ⓓ form-contorl

作答结果：正确

打开解析 ☐

D

6

2分

关于模板字符串，下列说法不正确的是：（）

Ⓐ 使用反引号标识

Ⓑ 插入变量的时候使用\${ }

Ⓒ 所有的空格和缩进都会被保留在输出中

Ⓓ \${ }中的表达式不能是函数的调用

作答结果：正确

打开解析 

D

7

2分

JavaScript中类的继承使用的关键字是：（）

☒ A extends

☐ B inherit

☐ C extend

☐ D base

作答结果：正确

打开解析 

A

8

2分

下面不属于关键字let的特点的是：（）

☐ A 只在 let 命令所在的代码块内有效

☒ B 会产生变量提升现象

☐ C 同一个作用域，不能重复声明同一个变量

☐ D 不能在函数内部重新声明参数

作答结果：正确

打开解析 

B

9

2分

以下属于vue指令的是（）

☐ A for

☐ B ng-model

☒ C v-bind

☐ D if

作答结果：正确

打开解析 

C

10

2分

下面不属于ECMAScript规范的范围的是：（）

- ☐ (A) 数据类型
- ☐ (B) 语法
- ☒ (C) DOM事件
- ☐ (D) 内置对象和函数的标准库

作答结果：正确

打开解析 

C

11

2分

jqer-mobile中设置图标底部对齐的方法是（）

- ☐ (A) data-icon="arrow-r"
- ☐ (B) data-iconpos="top"
- ☒ (C) data-iconpos="bottom"
- ☐ (D) 以上都不正确

作答结果：正确

打开解析 

C

12

2分

不属于用于数组遍历的函数的是：（）

- ☐ (A) keys()
- ☐ (B) entries()
- ☐ (C) values()
- ☒ (D) find()

作答结果：正确

打开解析 

D

13

2分

在vue中，v-if和v-show的说法正确的是（）

- ☐ (A) v-if隐藏元素后再页面中仍然显示盖元素
- ☒ (B) v-show隐藏元素后再页面中仍然显示盖元素
- ☐ (C) 二者功相同，并无区别

Ⓓ 以上说法都不对

作答结果：正确

打开解析 

B

14

2分

less声明变量的符号是（）

Ⓐ \$

Ⓑ *

Ⓒ @

Ⓓ &

作答结果：正确

打开解析 

C

15

2分

关于vue生命周期说法正确的是（）

Ⓐ Vue 实例从创建到销毁的过程，就是生命周期

Ⓑ 生命周期一共有4个钩子函数

Ⓒ DOM在updated钩子函数后渲染完毕

Ⓓ 以上说法都不对

作答结果：正确

打开解析 

A

16

2分

php 关于类继承说法正确的是

Ⓐ 子类可以直接调用基类的所有方法

Ⓑ 子类可以重写基类的方法

Ⓒ 子类可以直接继承多个基类

Ⓓ 子类可以重写基类 protected 级别方法为 private 级别

作答结果：正确

打开解析 

B

17

2分

XMLHttpRequest对象的status的值，当status==404是指什么（）

- (A) 服务器端错误
- (B) url重定向
- (C) 客户端
- (D) 服务器端不知道客户端要请求哪个资源
- (E) 没有任何意义

作答结果：正确

打开解析 ☐

C

18

2分

以下关于video说法正确的是（）

- (A) 当前，video元素支持三种视频格式，其中WebM=带有Theora视频编码和Vorbis音频编码的WebM文件。
- (B) source元素可以添加多个，具体播放哪个由浏览器决定。
- (C) video内使用img展示有视频封面
- (D) loop属性可以使媒介文件循环播放

作答结果：正确

打开解析 ☐

D

19

2分

关于箭头函数的描述，错误的是：（）

- (A) 使用箭头符号=>定义
- (B) 参数超过1个的话，需要用小括号（）括起来
- (C) 函数体语句超过1条的时候，需要用大括号{ }括起来，用return语句返回
- (D) 函数体内的 this 对象，绑定使用时所在的对象

作答结果：正确

打开解析 ☐

D

20

2分

Nodejs的特点不包括下列哪个？

- ☐ (A) 单线程
- ☒ (B) 多线程
- ☐ (C) 非阻塞I/O
- ☐ (D) 事件驱动

作答结果：正确

打开解析 ☐

B

21

2分

下述有关HTML Doctype的描述，说法错误的是（）

- ☐ (A) 他的责任就是告诉浏览器文档使用哪种HTML或XHTML规范
- ☐ (B) 不同文档模式主要影响CSS内容的呈现，尤其是浏览器对盒模型的解析
- ☒ (C) 不同浏览器在混杂模式下的行为差异非常大，如果不是要某些hack技术，跨浏览器的行为根本就没有一致性可言
- ☐ (D) 如果在文档开始处没有发现文档类型声明，则所有浏览器都会默认开启严格模式

作答结果：错误

打开解析 ☐

D

22

2分

关于 javascript 模块化，下列描述错误的是（）。

- ☐ (A) AMD推崇依赖前置，CMD推崇依赖就近
- ☒ (B) Sea.js遵循AMD规范，RequireJS遵循CMD规范
- ☐ (C) 主流模块化包括CommonJS,AMD,CMD等
- ☐ (D) 模块化有利于管理模块间的依赖，更依赖模块的维护

作答结果：正确

打开解析 ☐

B

23

在对象的解构赋值中，var {a,b,c} = { "c":10, "b":9, "a":8 } 结果中，a、b、c的值分别是：（）

2分

- (A) 10 9 8
- (B) 8 9 10
- (C) undefined 9 undefined
- (D) null 9 null

作答结果：正确

打开解析 ☐

B

24

2分

nodejs引用第三方模块的时候，如果不放入node_modules文件夹中，则必须写相对路径，写法是？

- (A) ./
- (B) ../
- (C) /
- (D) //

作答结果：错误

打开解析 ☐

A

25

2分

关于关键字const，下列说法错误的是：（）

- (A) 用于声明常量，声明后不可修改
- (B) 不会发生变量提升现象
- (C) 不能重复声明同一个变量
- (D) 可以先声明，不赋值。

作答结果：错误

打开解析 ☐

D

26

2分

设有一个数据库mydb中有一个表tb1，表中有六个字段，主键为ID，有十条记录，ID从0到9，以下代码输出结果是？()

```
<?php $link = mysql_connect('localhost', 'user', 'password') or die('Could not connect: ' .mysql_error()); $result = mysql_query("SELECT id, name, age FROM mydb.tb1 WHERE id<'5'") or die('Could not query: ' .mysql_error()); echo
```


mysql_num_fields(\$result); mysql_close(\$link); ?>

- (A) 6
- (B) 5
- (C) 4
- (D) 3

作答结果：正确

打开解析

B

27

2分

在php.ini中，如果想开启PHP中的短标签方式，应该将（
）设置为“On”

- (A) log_errors
- (B) display_errors
- (C) short_open_tag
- (D) date.timezone

作答结果：正确

打开解析

C

28

2分

img-responsive类可以让图片支持响应式布局，它的实现原理是（
）

- (A) 设置了max-width:100%;和height:auto;
- (B) 设置了max-width:100%;和height:100%;
- (C) 设置了width:auto;和height:100%;
- (D) 设置了width:auto;和height:auto;

作答结果：正确

打开解析

A

29

2分

关于数值的扩展，window.isNaN(“abc”) 和Number.isNaN(“abc”)的结果分别是：（
）

- (A) true false

B

false true


C

true true

D

false false

作答结果：正确

打开解析 

A

30

2分

获取Map实例对象的成员数，使用正确的属性是（）

A

size

B

length

C

sum

D

members

作答结果：正确

打开解析 

A

二、多选题 共15题，30分

1

2分

以下属于vue生命周期钩子函数的是（）

A

beforeCreat

B

mounted

C

created

D

updated

作答结果：错误

打开解析 

ABCD

2

2分

关于Symbol，正确的说法是：（）

A

是ES6新增的一种数据类型

B

Symbol() === Symbol() 结果为false

☒ Symbol('same') === Symbol('same') 结果为true

☒ 当symbol值作为对象的属性名的时候，不能用点运算符获取对应的值。

作答结果：错误

打开解析 ☐

ABD

3

2分

以下是canvas的方法是（）

☒ getContext()

☒ fill()

☒ stroke()

☐ controller()

作答结果：正确

打开解析 ☐

ABC

4

2分

以下属于Audio/Video的方法有？

☒ load()

☒ play()

☒ pause()

☒ stop()

作答结果：错误

打开解析 ☐

ABC

5

2分

关于webpack说法正确的是（）

☒ webpack是一个模块打包工具

☒ 可通过plugin扩展，方便、灵活

☐ 只能用于采用模块化开发的项目

☒ 良好的开发体验

作答结果：错误

打开解析 ☐

ABCD

6

2分

关于bootstrap的布局容器说法正确的是（）

- ☒ A container-fluid类用于100%宽度
- ☐ B .container和。container-fluid可以相互嵌套
- ☐ C 布局容器仅提供了.container和。container-fluid两种容器
- ☒ D .container类用于固定宽度并支持响应式的容器

作答结果：错误

打开解析 ☐

ACD

7

2分

关于laravel控制器中redirect跳转使用正确的是（）

- ☒ A return redirect()->to('home/test');
- ☒ B return redirect()->action('HomeController@index');
- ☒ C return redirect()->route(test');
- ☒ D return back();

作答结果：正确

打开解析 ☐

ABCD

8

2分

以下属于vuex属性的是（）？

- ☒ A states
- ☒ B mutations
- ☒ C actions
- ☐ D setters

作答结果：正确

打开解析 ☐

ABC

9

2分

关于Promise对象的状态，下列说法正确的是：（）

- ☒ A 三种状态分别是：pending初始状态、fulfilled成功、rejected失败
- ☐ B pending初始状态可以变成fulfilled成功
- ☐ C rejected失败不可以变成pending初始状态
- ☐ D rejected失败可以变成fulfilled成功

作答结果：正确

打开解析 

ABC

10

2分

函数的参数传递包括()

- ☒ A 按值传递
- ☒ B 按引用传递
- ☒ C 按变量传递
- ☐ D 按作用域传递

作答结果：错误

打开解析 

AB

11

2分

关于set结构，下面说法正确的是：（）

- ☒ A 需要用new关键字创建实例
- ☐ B 结构成员都是唯一的，不允许重复
- ☒ C 初始化的时候接受数组作为参数使用add方法添加已经存在的成员会报错

作答结果：错误

打开解析 

ABC

12

PHP表单的提交方法有:

2分

☒ A post

☐ B request

☒ C get

☐ D querystring

作答结果：错误

打开解析 ☐

ABC

13

2分

在jquery-mobile中关于网格布局说法正确的是（）

☒ A class 为 ui-grid-a（两列布局），您必须指定 ui-block-a 和 ui-block-b 的两个子元素。

☒ B class 为 ui-grid-b（三列布局），则必须添加 ui-block-a、ui-block-b 和 ui-block-c 的三个子元素。

☐ C class为ui-grid-solo,width为100%

☐ D 还有4列布局

作答结果：错误

打开解析 ☐

ABCD

14

2分

vue的生命周期，第一次页面加载会触发那几个钩子

☐ A beforeCreat

☒ B mounted

☒ C created

☐ D updated

☒ E beforeMount

作答结果：错误

打开解析 ☐

ABCE

15

下列数据结构中，能被for...of遍历的是：（）

☒

2分

- ☒ A Array数组
- ☒ B Object对象
- ☒ C String字符串
- ☐ D Set结构

作答结果：错误

打开解析

ACD

三、判断题 共5题，10分

1

2分

vue-router的导航守卫有全局导航守卫和组件内导航守卫两种

- ☒ A 对
- ☐ B 错

作答结果：错误

打开解析

F

2

2分

关于类class,如果类的constructor构造方法有多个，后者会覆盖前者

- ☒ A 对
- ☐ B 错

作答结果：错误

打开解析

F

3

2分

Nodejs是一门后端语言，使用js书写

- ☒ A 对
- ☐ B 错

作答结果：错误

打开解析

F

4

2分

模块热更新是webpack的一个功能，他可以使得代码修改过后不用刷新浏览器就可以更新，是高级版的自动刷新浏览器

A 对

B 错

作答结果：正确

打开解析 ☐

T

5

2分

在jquery-mobile中当用户垂直或水平旋转移动设备时，触发方向改变（orientationchange）事件。

A 对

B 错

作答结果：正确

打开解析 ☐

T

四、客观填空题 共4题，100分

1

20分

阅读下列说明、效果图和代码，填写（1）至（10）代码。

【说明】

公司现要开发一个移动端的新项目，项目为人员管理系统，项目名为hr-manage，项目拟采用vue进行开发。

现在设计阶段给出两个简单模块，分别是底部的首页和人员管理两个切换页，首页仅提供个欢迎语句即可，人员管理能做简单人员操作。项目核心包括assets文件夹、components文件夹、router文件夹、views文件夹、App.vue文件和main.js文件，assets文件夹包含logo.png和css文件夹，css文件夹中是public.css文件，components文件夹包含Hello.vue和footer.vue文件，views文件夹包含index文件夹和manage文件夹，里面各包含index.vue。实际完成的效果图如图5-1和图5-2：


【效果图】



图1



图2

```
【素材】
 素材.rar
【vue代码： App.vue】
<template>
  <div id="app">
    
    <router-view> </router-view>
  </div>
</template>
<script>
export default {
  name: 'app'
}
</script>
<style>
@import './assets/css/public.css';
#app {
  font-family: 'Avenir', Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  text-align: center;
  color: #2c3e50;
  margin-top: 60px;
}
</style>
【vue代码： views/manage/index.vue】
<template>
<template>
  <div class="manage tc">
    <button v-on:click="add">新增</button>
    <div class="input-area" v-show="showAdd">
      <input type="text" placeholder="请输入人员姓名" v-
model="nameValue">
      <button v-on:click="addName">确定</button>
    </div>
    <table>
      <tr>
        <th>姓名</th>
        <th>操作</th>
      </tr>
      <!--vue中for循环的指令 -->
      <tr (          )="(item,index) in peoples">
        <td>
          {{item.name}}
        </td>
        <td v-bind:id="index"><span v-on:click="edit">编辑</span>
<span v-on:click="del">删除</span></td>
      </tr>
    </table>
    <!--vue中元素显示的指令 -->
    <div class="wrap" (          )="showEdit">
      <div class="content">
        <!--vue中数据双向绑定的指令 -->
        <input type="text" placeholder="请输入新姓名" (
) ="newName">
        <button v-on:click="cancel">取消</button>
        <button v-on:click="editName">确定</button>
      </div>
    </div>
    <footer-nav v-bind:class="{ 'isManage':isNowPage}"></footer-
nav>
  </div>
</template>
<style>
/*样式省略*/
</style>
<script>
//引入公共组件底部菜单条
(          ) footerNav from './../components/ (
) .vue'
export default{
  components: {
    footerNav
  },
},
```

```
data(){
  return{
    isNowPage: true,
    showAdd: false,
    showEdit: false,
    peoples: [{name:'Jack'},{'name':'Joe'}],
    nameValue: "",
    newName: "",
    editId: 0
  }
},
//定义方法
(      ) : {
  add(){
    this.showAdd = true
  },
  addName(){
    var v = this.nameValue
    if(v.trim() == ""){
      alert("请输入新增人员姓名")
    }else{
      var data = {}
      data.name = v
      this.peoples.push(data)
    }
  },
  del(e){
    var id = e.target.offsetParent.id
    this.peoples.splice(id,1)
  },
  edit(e){
    var id = e.target.offsetParent.id
    this.showEdit = true
    this.editId = id
    this.newName = this.peoples[id].name
  },
  cancel(){
    this.showEdit = false
  },
}
//编辑调用的方法
(      ) (){
  var v = this.newName
  if(v.trim() == ""){
    alert("请输入姓名")
  }else{
    this.peoples[this.editId].name = this.newName
    this.showEdit = false
  }
}
}
}
</script>
【js代码： router/index.js】
import Vue from 'vue'
import Router from 'vue-router'
import Hello from '@/components/Hello'
import Index from '../views/index/index'
import Manage from '../views/manage/index'
Vue.use(Router)
export default new Router({
//实例化路由
(      ) : [
  {
    path: '/',
    name: 'Index',
    component: Index
  },
  {
    path: '/manage',
    name: 'Manage',
    component: Manage
  }
]
})
【问题】
1、根据项目上下文， 仔细查
```

看App.vue和views/manage/index.vue文件，填写（1） - （7）空。
2、根据您掌握的vue-router知识，填写router/index.js中（8）空。
3、vuex 是一个专门为vue.js应用程序开发的状态管理模式。vuex中有默认的五种基本的对象，分别是state、（
）、mutations、actions和（
）。

1、学生答案：

2、学生答案：

3、学生答案：

4、学生答案：

5、学生答案：

6、学生答案：

7、学生答案：

8、学生答案：

9、学生答案：

10、学生答案：

打开解析 ☐

- (1) v-for

(2) v-show

(3) v-model

(4) import

(5) footer

(6) methods

(7) editName

(8) routes

(9) getters

(10) modules

2

20分

阅读下列说明，根据要求，完成以下题目。

【说明】

使用Node.js和Express框架完成一个简单的留言项目，具体的需求如下所示。

1、实现一个留言列表，默认首页是留言列表页面，留言数据可以先

- 用本地数组存储。
- 2、实现发表留言操作，发表留言可以对留言数组进行新增操作，成功后刷新留言列表。
- 3、项目的核心内容public文件夹、views文件夹和app.js， public中包含index.css和lib文件夹， lib文件夹中包含bootstrap文件夹（bootstrap的静态资源）， views文件夹中包含index.html和post.html。
- 4、实际的效果图如图2-1和图2-2所示：
- 【效果图】



图1

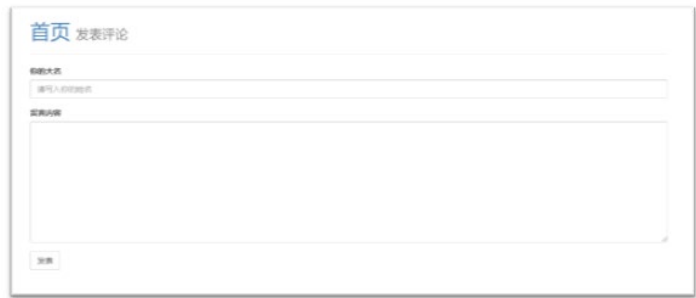


图2

【素材】

 [素材.rar](#)

【html代码： index.html】

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="UTF-8">
  <title>留言本</title>
  <link rel="stylesheet"
href="/public/lib/bootstrap/bootstrap.css">
</head>
```

```
<body>
  <div class="header container">
    <div>
      <h1>留言板 <small>你的声音，我们一直在倾听</small>
    </h1>
    <a class="btn btn-success" href="/post">发表留言</a>
    </div>
  </div>
  <div class="comments container">
    <ul>
      {{each comments}}
        <li>{{ $value.name }}说： {{ $value.message }} <span>{{
$value.dateTime }}</span> </li>
      {{/each}}
    </ul>
  </div>
</body>
</html>
```

【html代码： post.html】

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
```

```
<title>Document</title>
<link rel="stylesheet"
href="/public/lib/bootstrap/bootstrap.css">
</head>
<body>
  <div class="header container">
    <div>
      <h1><a href="/"> 首页</a> <small>发表评论</small></h1>
    </div>
  </div>
  <div class="comments container">
    <form action="/post" method="post">
      <div>
        <label for="input_name">你的大名</label>
        <input type="text" required minlength="2" maxlength="10"
id="input_name" name="name" placeholder="请写入你的姓名">
      </div>
      <div>
        <label for="textarea_message">留言内容</label>
        <textarea name="message" id="textarea_message"
cols="30" rows="10" required minlength="5" maxlength="20">
      </textarea>
      </div>
      <button type="submit" class="btn btn-default">发表
    </button>
    </form>
  </div>
</body>
</html>
```

```
【js代码： app.js】
const express = require('express');
const app = express();
const bodyParser = require('body-parser');
var comments = [{
  name: '张三',
  message: '今天天气不错! ',
  dateTime: '2015-10-16'
},
{
  name: '张三2',
  message: '今天天气不错! ',
  dateTime: '2019-10-16'
},
{
  name: '张三3',
  message: '今天天气不错! ',
  dateTime: '2019-10-17'
},
{
  name: '张三4',
  message: '今天天气不错! ',
  dateTime: '2019-11-16'
},
{
  name: '张三5',
  message: '今天天气不错! ',
  dateTime: '2019-11-17'
}
];
//配置express的静态资源路径
app.use('/public/',express. ( ) ('./public'));
// 配置 body-parser 中间件（插件，专门用来解析表单 POST 请求体）
// parse application/x-www-form-urlencoded
app.use(bodyParser.urlencoded({
  extended: false
}))
app.use(bodyParser.json())
app.get('/',function (req,res) {
  res.render('index.html',{
    //将已知的留言数据传入页面
    comments: ( )
  });
});
app.get('/post',function (req,res) {
  res. ( ) ('post.html');//切换到post.html页面
```

```
});
app.post('/post',function (req,res) {
    var myDate = new Date();
    var year = myDate.getFullYear(); //获取完整的年份(4位,1970-
    ???)
    var month = myDate.getMonth() + 1; //获取当前月份(0-11,0代
    表1月)
    var date = myDate.getDate(); //获取当前日(1-31)
    let comment = req. (      ) ;//获取请求的内容体
    comment.dateTime = year + "-" + month + "-" + date;
    comments.unshift(comment);
    //发布完成重定向到首页
    res. (      ) ('/');
})
//监听启动3000端口
app. (      ) ( (      ) ,function () {
//控制台打印信息
    console. (      ) ('3000');
});
```

【问题】
1、基于Express框架，在app.js中填写（1） - （8）空。
2、假如你当前处于该项目的根目录下，应该在命令行执行什么命令来启动基于node的该项目（默认已经安装node）。
（ ） （ ）

1、学生答案：

2、学生答案：

3、学生答案：

4、学生答案：

5、学生答案：

6、学生答案：

7、学生答案：

8、学生答案：

9、学生答案：

10、学生答案：

打开解析 ☐

- (1) static

(2) comments

(3) render

(4) body

(5) redirect

(6) listen

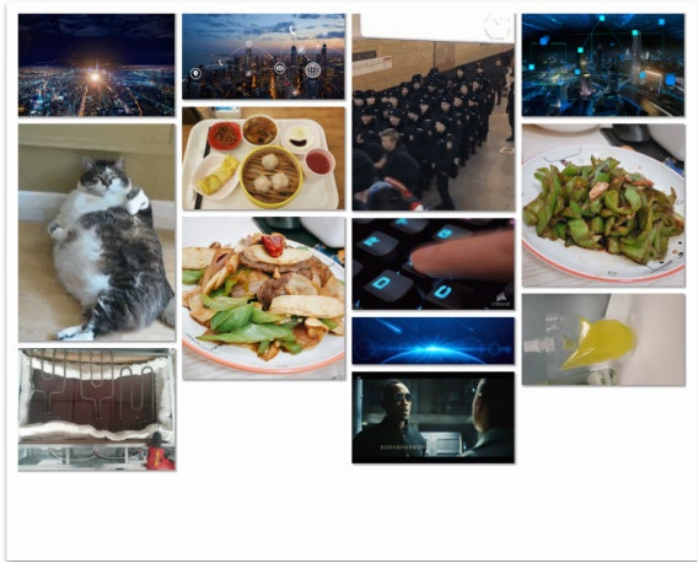
- (7) 3000
- (8) log
- (9) node
- (10) app.js

3

20分

原生JAVASCRIPT一般比较多出现在插件中，以下给出了使用原生JAVASCRIPT制作的瀑布流的小案例，效果图如3-1所示，通过对原生JAVASCRIPT的知识掌握，回答问题1至问题4。

【效果图】



【HTML代码：】

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>瀑布流</title>
    <link rel="stylesheet" type="text/css"
href="css/index4.css">
  </head>
  <body>
    <div id="box">
      <div class="item">
    </div>
      <div class="item">
    </div>
      <div class="item">
    </div>
      <div class="item">
    </div>
      <div class="item">
    </div>
      <div class="item">
    </div>
      <div class="item">
    </div>
      <div class="item">
    </div>
      <div class="item">
    </div>
      <div class="item">
    </div>
  </div>
</body>
</html>
```



```
<div class="item">
</div>
<div class="item">
</div>
<div class="item">
</div>
</div>
<script src="js/index4.js"></script>
</body>
</html>
```

【JS代码：】

```
var box = document.getElementById('box');
var items = box.children;
// 定义每一列之间的间隙 为10像素
var gap = 10;
window.onload = function() {
    // 一进来就调用一次
    ();
    // 封装成一个函数
    function waterFall() {
        // 获取页面宽度
        var pageWidth = getClient().width;
        // 声明元素宽度为items数组第一个的宽度
        var itemWidth = items[0].offsetWidth;
        // 1- 确定列数 = 页面的宽度 / 图片的宽度
        // 转换成int型
        var columns = Math.floor(pageWidth / (itemWidth + gap));
        var arr = [];
        for (var i = 0; i < items.length; i++) {
            if (i < columns) {
                // 2- 确定第一行
                items[i].style.top = 0;
                items[i].style.left = (itemWidth + gap) * i + 'px';
                // 添加到数组arr中
                arr.push(items[i].offsetHeight);
            } else {
                // 其他行
                // 3- 找到数组中最小高度 和 它的索引
                var minHeight = arr[0];
                var index = 0;
                for (var j = 0; j < arr.length; j++) {
                    if (minHeight > arr[j]) {
                        minHeight = arr[j];
                        index = j;
                    }
                }
                // 4- 设置下一行的第一个盒子位置
                // top值就是最小列的高度 + gap
                items[i].style.top = arr[index] + gap + 'px';
                // left值就是最小列距离左边的距离
                items[i].style.left = items[index].offsetLeft + 'px';
                // 5- 修改最小列的高度
                // 最小列的高度 = 当前自己的高度 + 拼接过来的高度 + 间隙的高度
                arr[index] = arr[index] + items[i].offsetHeight + gap;
            }
        }
    }
    // 页面尺寸改变时实时触发
```

```
//监听页面改变大小事件
window.      = function() {
    waterFall();
};
// 当加载到第30张的时候
//监听页面鼠标滚动事件
window.      = function() {
    if (getClient().height + getScrollTop() >= items[items.length
- 1].offsetTop) {
        // 模拟 ajax 获取数据
        var datas = [
            "../image/瀑布流/001.jpg",
            "../image/瀑布流/002.jpg",
            "../image/瀑布流/003.jpg",
            "../image/瀑布流/004.jpg",
            "../image/瀑布流/005.jpg",
            "../image/瀑布流/006.jpg",
            "../image/瀑布流/007.jpg",
            "../image/瀑布流/008.jpg",
            "../image/瀑布流/009.jpg",
            "../image/瀑布流/010.jpg",
            "../image/瀑布流/011.jpg",
            "../image/瀑布流/012.jpg",
            "../image/瀑布流/030.jpg"
        ];
        for (var i = 0; i < datas.length; i++) {
            //创建一个div元素
            var div = document.      ("div");
            div.className = "item";
            div.innerHTML = '';
            //将div添加到box中
            box.      (div);
        }
        waterFall();
    }
};
// clientWidth 处理兼容性
function getClient() {
    return {
        width:window.innerWidth ||
        document.documentElement.clientWidth ||
        document.body.clientWidth,
        height: window.innerHeight ||
        document.documentElement.clientHeight ||
        document.body.clientHeight
    }
}
// scrollTop兼容性处理
function getScrollTop() {
    return window.pageYOffset ||
    document.documentElement.scrollTop;
}
```

【素材】

 [素材.zip](#)

【问题】

- 1、分析上面JAVASCRIPT代码，填写（1）至（8）。
- 2、像l6/7/8的版本数组不支持indexOf方法，可以在Array原型链上

- 扩展indexOf方法，使用到的关键字是 ？
- A. proto
 - B. prototype;
 - C. _proto_
 - D. _prototype_
- 3、在原生JAVASCRIPT中给元素绑定点击事件IE写法的是 ？
- A.attachEvent
 - B.addEventListener
 - C.addEvent
 - D.attachEventListene

1、 学生答案：

2、 学生答案：

3、 学生答案：

4、 学生答案：

5、 学生答案：

6、 学生答案：


7、 学生答案：

8、 学生答案：

9、 学生答案：

10、 学生答案：

打开解析 

- 
- (1) waterFall
 - (2) 0
 - (3) parseInt
 - (4) push
 - (5) onresize
 - (6) onscroll
 - (7) createElement
 - (8) appendChild
 - (9) B
 - (10) A

4

40分

阅读下列说明、效果图和代码，进行动态网页开发，回答问题1至问题20。

【说明】

该程序为一个快餐收银系统，前端使用Vue.js框架，使用Vue CLI构建项目,项目名为manage。前端核心文件全部位于src目录，包括路由配置文件router/index.js，入口文件main.js,根组件App.vue,存放组件目录components，在components目录

下common/leftNav.vue是左侧菜单，page/pos.vue是商品管理组件。

Vue.js框架采用Axios组件实现Ajax请求。

本题素材来源于网络，代码源于<http://jspang.com/>，版权归作者所有。

【效果图】



【入口文件代码：main.js】

```
// 在 (1) 处补全代码，导入vue
import      from 'vue'
import App from './App'
// 在 (2) 处补全代码，导入router
import      from './router'
import ElementUI from 'element-ui'
import 'element-ui/lib/theme-default/index.css'
Vue.config.productionTip = false;
// 在 (3) 处补全代码，应用第三方ui库
Vue.use(      );
new Vue({
  el: '#app',
  router,
  template: '<App/>',
  components: { App }
})
```

【根组件代码：App.vue】

```
<template>
  <div id="app">
    <!--在 (4) 处使用双标签实现左侧导航-->

    <!--操作区域-->
    <div class="main">
      <!-- 在 (5) 处使用双标签实现路由视图 -->

    </div>
  </div>
</template>
<script>
// 在 (6) 处填写导入关键字
    leftNav from '@components/common/leftNav'
// 在 (7) 处填写导出关键字
    default {
      name: 'app',
      components:{
        leftNav
      }
    }
  </script>
```

```
</script>
<style>
#app {
  font-family: 'Microsoft YaHei','Avenir', Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  text-align: left;
  color: #2c3e50;
  height:100%;
}
.main{
  float:left;
  width:95%;
  background-color: #EFF2F7;
  height:100%;
  overflow: auto;
}
</style>
【路由代码: index.js】
import Vue from 'vue'
import Router from 'vue-router'
import Pos from '@/components/page/Pos'
Vue.use(Router)
// 在 (8) 处填写路由类
export default new      ({}
  routes: [
    {
      // 在 (9) 处填写路径属性
      : '/',
      name: 'Pos',
      // 在 (10) 处填写组件属性
      : Pos
    }
  ]
})
【Pos组件代码: Pos.vue】
<template>
  <div class="pos">
    <div>
      <el-row>
        <el-col :span="7" class="pos-order" id="order-list">
          <el-tabs>
            <el-tab-pane label="点餐">
              <el-table :data="tableData" border
style="width: 100%">
                <el-table-column prop="goodsName"
label="商品"></el-table-column>
                <el-table-column prop="count" label="量"
width="50"></el-table-column>
                <el-table-column prop="price" label="金额"
width="70"></el-table-column>
                <el-table-column label="操作" width="100"
fixed="right">
                  <template scope="scope">
                    <!-- 在 (11) 处填写删除单个商品的方法 -->
                    <el-button type="text" size="small"
@click="      (scope.row)">删除</el-button>
                    <!-- 在 (12) 处填写增加商品的方法 -->
                    <el-button type="text" size="small"
```

```
@click="
      (scope.row)">增加</el-button>
    </template>
  </el-table-column>
</el-table>
<div class="totalDiv">
  <small>数量: </small>
  <!-- 在 (13) 处填写总数量-->
  <strong>{{
    }}</strong>
  <small>总计: </small>
  <!-- 在 (14) 处填写总钱数 -->
  <strong>{{
    }}</strong> 元
</div>
<div class="order-btn">
  <el-button type="warning">挂单</el-
button>
    <el-button type="danger"
@click="delAllGoods()">删除</el-button>
    <el-button type="success"
@click="checkout()"> 结账</el-button>
  </div>
</el-tab-pane>
<el-tab-pane label="其他">
  其他
</el-tab-pane>
</el-tabs>
</el-col>
<!--商品展示-->
<el-col :span="17">
  <div class="often-goods">
    <div class="title">常用商品</div>
    <div class="often-goods-list">
      <ul>
        <!-- 在(15)处填写常用商品 -->
        <li v-for="goods in
          "
@click="addOrderList(goods)" :key="goods.goodsId">
          <span>{{goods.goodsName}}</span>
          <span class="o-
price"> ¥ {{goods.price}}元</span>
        </li>
      </ul>
    </div>
  </div>
  <div class="goods-type">
    <el-tabs>
      <el-tab-pane label="汉堡">
        <ul class='cookList'>
          <li v-for="goods in type0Goods"
@click="addOrderList(goods)" :key="goods.goodsId">
            <span class="foodImg">
              
            </span>
            <span class="foodName">
              {{goods.goodsName}}</span>
            <span
class="foodPrice"> ¥ {{goods.price}}元</span>
          </li>
        </ul>
      </el-tab-pane>
      <el-tab-pane label="小食">
        <ul class='cookList'>
```

```

        <li v-for="goods in type1Goods"
@click="addOrderList(goods)" :key="goods.goodsId">
            <span class="foodImg">
                
            </span>
            <span class="foodName">
{{goods.goodsName}}</span>
            <span
class="foodPrice"> ¥ {{goods.price}}元</span>
        </li>
    </ul>
</el-tab-pane>
<el-tab-pane label="饮料">
    <ul class='cookList'>
        <li v-for="goods in type2Goods"
@click="addOrderList(goods)" :key="goods.goodsId">
            <span class="foodImg">
                
            </span>
            <span class="foodName">
{{goods.goodsName}}</span>
            <span
class="foodPrice"> ¥ {{goods.price}}元</span>
        </li>
    </ul>
</el-tab-pane>
<el-tab-pane label="套餐">
    <ul class='cookList'>
        <li v-for="goods in type3Goods"
@click="addOrderList(goods)" :key="goods.goodsId">
            <span class="foodImg">
                
            </span>
            <span class="foodName">
{{goods.goodsName}}</span>
            <span
class="foodPrice"> ¥ {{goods.price}}元</span>
        </li>
    </ul>
</el-tab-pane>
</el-tabs>
</div>
</el-col>
</el-row>
</div>
</div>
</template>
```

```

<script>
import axios from 'axios';
export default {
  name: 'Pos',
  // 在(16)处填写数据属性
  () {
    return {
      tableData: [], //订单列表的值
      oftenGoods: [],
```

```

        type0Goods: [],
        type1Goods: [],
        type2Goods: [],
        type3Goods: [],
        totalMoney: 0, //订单总价格
        totalCount: 0 //订单商品总数量
    }
},
// 在(17)处填写html加载完成的生命周期函数
    : function () {
        var orderHeight = document.body.clientHeight;
        document.getElementById("order-list").style.height =
orderHeight + 'px';
        // 让左边的order-list选中商品的列表的高度和页面的高度一
致。
        // console.log(orderHeight)
    },
    created() {
        //在(18)处补全代码,发起网络请求,读取常用商品列表


        .get('https://www.fastmock.site/mock/fcd3afd8873fff8c2aa4381c17c831e8/data/api/goodslist')
        .then(response => {
            // 在(19)处补全代码,填写常用商品
            this.          = response.data;
        })
        // 在(20)处补全代码,当网络请失败时
        .          (error => {
            console.log(error);
            alert('网络错误, 不能访问');
        })
        //读取分类商品列表
        // 二维数组 第二维代表分类

        axios.get('https://www.fastmock.site/mock/fcd3afd8873fff8c2aa4381c17c831e8/data/api/goodstype')
        .then(response => {
            // console.log(response);
            //this.oftenGoods=response.data;
            // 汉堡
            this.type0Goods = response.data[0];
            // 小食品
            this.type1Goods = response.data[1];
            // 饮料
            this.type2Goods = response.data[2];
            // 套餐
            this.type3Goods = response.data[3];
        })
        .catch(error => {
            console.log(error);
            alert('网络错误, 不能访问');
        })
    },

    methods: {
        //添加订单列表的方法
        addOrderList(goods) {
            //goods传过来的当前的商品
            this.totalCount = 0; //汇总数量清0
            this.totalMoney = 0;//总钱数清0
            let isHave = false;
            //判断是否这个商品已经存在于订单列表
        }
    }
}
```



```
for (let i = 0; i < this.tableData.length; i++) {
  // console.log(this.tableData[i].goodsId);
  if (this.tableData[i].goodsId == goods.goodsId) {
    isHave = true; //存在
  }
}
//根据isHave的值判断订单列表中是否已经有此商品
if (isHave) {
  //存在就进行数量添加
  // filter里边是一个箭头函数判断的条件
  // let arr = this.tableData.filter(o => o.goodsId ==
goods.goodsId);
  let arr = this.tableData.filter(function(o){return
o.goodsId==goods.goodsId});
  console.log(arr)
  arr[0].count++;
} else {
  //不存在就推入数组
  let newGoods = { goodsId: goods.goodsId,
goodsName: goods.goodsName, price: goods.price, count: 1 };
  this.tableData.push(newGoods);
}
this.getAllMoney();
},
//删除单个商品
delSingleGoods(goods) {
  console.log(goods);
  this.tableData = this.tableData.filter(o => o.goodsId !=
goods.goodsId);
  this.getAllMoney();
},
//删除所有商品
delAllGoods() {
  this.tableData = [];
  this.totalCount = 0;
  this.totalMoney = 0;
},
//汇总数量和金额
getAllMoney() {
  this.totalCount = 0;
  this.totalMoney = 0;
  if (this.tableData) {
    this.tableData.forEach((element) => {
      this.totalCount += element.count;
      this.totalMoney = this.totalMoney + (element.price *
element.count);
    });
  }
},
//结账方法模拟
checkout() {
  if (this.totalCount!=0) {
    this.tableData = [];
    this.totalCount = 0;
    this.totalMoney = 0;
    this.$message({
      message: '结账成功',
      type: 'success'
    });
  }else{
    this.$message.error('不能空结');
```

```
        }  
      }  
    }  
  }  
</script>  
<style scoped>  
//此处省略样式代码  
</style>  
【问题】根据vue知识，正确填写所缺代码。  
 素材.zip
```

1、 学生答案：

2、 学生答案：

3、 学生答案：

4、 学生答案：

5、 学生答案：

6、 学生答案：

7、 学生答案：

8、 学生答案：

9、 学生答案：

10、 学生答案：

11、 学生答案：

12、 学生答案：

13、 学生答案：

14、 学生答案：

15、 学生答案：

16、 学生答案：


17、学生答案：

18、学生答案：

19、学生答案：

20、学生答案：

打开解析 



```
(1)Vue
(2)router
(3)ElementUI
(4)<leftNav></leftNav>
(5)<router-view></router-view>
(6)import
(7)export
(8)Router
(9)path
(10)component
(11)delSingleGoods
(12)addOrderList
(13)totalCount
(14)totalMoney
(15)oftenGoods
(16)data
(17)mounted
(18)axios
(19)oftenGoods
(20)catch
```

