

答题中

Web前端开发高级模拟测试（五）

共55道题 总分：200分 形考总分：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



在es6中新增的模板字符串关键字是（）

- ☐ (A) "
- ☐ (B) \$
- ☐ (C) &
- ☐ (D) `

2

2分



vue中，使用什么指令进行数据双向绑定（）

- ☐ (A) v-bind
- ☐ (B) v-mode
- ☐ (C) v-modle
- ☐ (D) {{}}

3

2分



不是MySQL约束的类型的是（）

- ☐ (A) not null
- ☐ (B) unique key
- ☐ (C) primary key
- ☐ (D) sort

vue中实例是在哪个阶段创建完的（）

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
---	---	---	---

标记

已做

正做

未做

2

4

2分



- (A) updated
- (B) mounted
- (C) destored
- (D) beforeDestroy

5

2分



以下哪个标签用于定义SVG的路径（）

- (A) <path>
- (B) <polyline>
- (C) <rect>
- (D) <ellipse>

6

2分



在webpack 配置文件中，下列哪个是正确的入口配置项（）

- (A) entry
- (B) enter
- (C) input
- (D) join

7

2分



下列不是ES6中SET的方法的是（）

- (A) has
- (B) add
- (C) put
- (D) delete

8

2分

以下关computed， methods， watch的描述不正确的是（）

- (A) computed是计算属性， methods是点击事件触发， watch是监听
- (B) methods有getter、setter属性， setter接受， getter返回



- ③ methods存放的是函数方法
- ④ watch有newQuestion,, oldQuestion两个参数, 一个新值, 一个旧值

9

2分



- 首屏加载速度快的加载技术是 ()
- ① 预加载
 - ② 懒加载
 - ③ 先加载
 - ④ 快加载

10

2分



- 以下属于vue中常用的声明周期是 ()
- ① created阶段
 - ② started阶段
 - ③ prepared阶段
 - ④ kill阶段

11

2分



- 关于Promise对象的状态, 下列说法错误的是 ()
- ① 三种状态分别是: pending初始状态、fulfilled成功、rejected失败
 - ② pending初始状态可以变成fulfilled成功
 - ③ rejected失败不可以变成pending初始状态
 - ④ rejected失败可以变成fulfilled成功

12

2分



- 以下HTML中的规范不正确的是 ()
- ① HTML代码所有的标签名和属性应该都为小写
 - ② 属性值使用双引号进行闭合
 - ③ 给HTML代码块添加必要注释
 - ④ 代码全都向左靠齐

13

2分



下列符号是ES6中的箭头函数的是 ()

- Ⓐ -->
- Ⓑ =>
- Ⓒ ->
- Ⓓ ==>

14

2分



创建表的语句正确的是 ()

- Ⓐ create table mytable(id int(12),username varchar(20));
- Ⓑ createtable mytable(id int(12),username varchar(20));
- Ⓒ create table mytable(id(12),username(20) varchar);
- Ⓓ create table mytable(id int(12);uername varchar(20));

15

2分



以下不是vue中的通信的是 ()

- Ⓐ vuex
- Ⓑ 父子通信
- Ⓒ 兄弟通信
- Ⓓ 前后通信

16

2分



vue中只执行一次的指令是 ()

- Ⓐ v-one
- Ⓑ v-once
- Ⓒ v-1
- Ⓓ v-model

17

2分



下列是ES6数组扩展的方法是 ()

- Ⓐ push
- Ⓑ shift
- Ⓒ find
- Ⓓ indexOf

18

2分



bootstrap的栅格系统描述错误的是（）

- Ⓐ 系统默认每行划分12列
- Ⓑ 网格不可以嵌套
- Ⓒ 默认列间距是30px
- Ⓓ 手机屏幕的列类前缀是.col-sm-x

19

2分



下列CSS3属性选择器中表示匹配属性值以指定值开头的是（）

- Ⓐ [attribute|=value]
- Ⓑ [attribute^=value] [attribute\$=value]
- Ⓒ [attribute*=value]

20

2分



如何判断变量是数组（）

- Ⓐ Array.isArray()
- Ⓑ Array.sort()
- Ⓒ Array.array()
- Ⓓ Array.fromArray()

21

2分



雪碧图的核心实现原理是依赖以下哪个样式（）

- Ⓐ font-size
- Ⓑ background
- Ⓒ color
- Ⓓ font-family

22

2分

以下哪种属性可以设置Canvas里填充背景的颜色（）

- Ⓐ fillStyle
- Ⓑ strokeStyle
- Ⓒ



- ☐ (C) .shadowColor
- ☐ (D) shadowBlur

23

2分



下列不是HTML5新增的标签的是 ()

- ☐ (A) video
- ☐ (B) music
- ☐ (C) audio
- ☐ (D) canvas

24

2分



下列关于DOM描述正确的是 ()

- ☐ (A) DOM是浏览器对象模型的简称
- ☐ (B) DOM操作非常耗性能
- ☐ (C) CSS中存在DOM操作
- ☐ (D) DOM操作和浏览器没有关系

25

2分



vue中怎么发送请求 ()

- ☐ (A) axios
- ☐ (B) ajax
- ☐ (C) xhr
- ☐ (D) send

26

2分



以下关于箭头函数特点错误的是 ()

- ☐ (A) 不需要function关键字来创建函数
- ☐ (B) 省略return关键字
- ☐ (C) 继承当前上下文的 this 关键字
- ☐ (D) .箭头的方向会随着变量类型变化

27

以下关于jQuery Mobile的说法正确的是 ()

- ☐ (A) jQuery Mobile和jQuery没有关系

2分



- ☐
- ☐ (B) jQuery Mobile是针对触屏智能手机与平板电脑的网页开发框架
- ☐ (C) jQuery Mobile不能在iphone上使用
- ☐ (D) jQuery Mobile只是javascript层的框架

28

2分



- 关于javascript模块化编程，下列说法正确的是（）
- ☐ (A) javascript模块越多越好
 - ☐ (B) .AMD是一种javascript模块化编程的思想
 - ☐ (C) BMD是一种javascript模块化编程的思想
 - ☐ (D) 系统规模越小， javascript模块化越有优势

29

2分



- 下列不是ES6中新增的关键字（）
- ☐ (A) var
 - ☐ (B) let
 - ☐ (C) const
 - ☐ (D) import

30

2分



- PHP类中只能内部所访问的权限是（）
- ☐ (A) public
 - ☐ (B) protected
 - ☐ (C) default
 - ☐ (D) private

二、多选题 共15题， 30分

1

2分

- 以下对webpack的描述，正确的有（）
- ☐ (A) webpack是一个js文件，不需要安装
 - ☐ (B) webpack依赖于node环境
 - ☐ (C) webpack可以完成前端项目的打包工作



☐ D webpack是国产插件

2

2分

对于前端优化，下列说法正确的有（）

- ☐ A 前端优化主要从HTML、CSS、JAVASCRIPT和图片等静态资源做优化
- ☐ B 前端无法对图片做出优化
- ☐ C 前端优化能显著提升页面访问速度
- ☐ D 前端优化能提升用户体验



3

2分

vue中，第一次加载页面会触发哪几个钩子（）

- ☐ A beforeCreated
- ☐ B beforeMount
- ☐ C .afterCreated
- ☐ D afterMount



4

2分

下列哪些属于Vue指令（）

- ☐ A v-show
- ☐ B v-for
- ☐ C v-if
- ☐ D v-on



5

2分

下列是虚拟 DOM 的优点（）

- ☐ A 保证性能下限
- ☐ B 无需手动操作DOM
- ☐ C 跨平台
- ☐ D 可以完全替代DOM



6

2分



除了webpack，还有哪些打包插件是用的比较多的（）

- ☐ A grunt
- ☐ B gulp
- ☐ C zip
- ☐ D bower

7

2分



下列关于MVC模式的说法，正确的有（）

- ☐ A M指Model（模型），数据保存
- ☐ B V指View（视图），用户界面
- ☐ C C指Controller（控制器），业务逻辑
- ☐ D MVC和MVVM都是指开发模式

8

2分



PHP中只引入一次外部文件的方法是（）

- ☐ A require_once("filename");
- ☐ B include_once("filename");
- ☐ C require("filename");
- ☐ D include("filename");

9

2分



vue-router有哪些导航钩子（）

- ☐ A 全局导航钩子
- ☐ B 组件内导航钩子
- ☐ C 单路由共享组件导航钩子
- ☐ D 内外导航钩子

10

2分

vue常用的修饰符有哪些（）

- ☐ A prevent:提交事件不再重载页面
- ☐ B stop:阻止单击事件冒泡



- ☐ C self:当事件发生在该元素本身而不是子元素的时候会触发
- ☐ D capture:事件侦听，事件发生的时候会调用

11

2分



module模块中，对下列语句的描述，错误的是（）

- ☐ A export default 默认导出
- ☐ B import导入
- ☐ C export、export default能直接导出变量表达式
- ☐ D import*as重命名

12

2分



下列对jsonp说法正确的是（）

- ☐ A jsonp只能用于get请求
- ☐ B .jsonp是json的扩展
- ☐ C jsonp是一种前端跨域处理方案
- ☐ D jsonp原理是html中标签的src属性可以跨域

13

2分



下列是jQuery Mobile的表单控件的有（）

- ☐ A 加载条
- ☐ B 文本输入框
- ☐ C 复选框
- ☐ D 滑动条

14

2分



以下查询语句正确的有（）

- ☐ A select * from tableA where name='张三';
- ☐ B select * from tableA where name='张%' && age>20;
- ☐ C select * from tableA desc;
- ☐ D select * from tableA group by age;

15

2分



vue-router有哪些组件 ()

- ☐ A router-link
- ☐ B router-view
- ☐ C .vue-href
- ☐ D vue-page

三、判断题 共5题, 10分

1

2分



canvas能兼容ie8 ()

- ☐ A 对
- ☐ B 错

2

2分



使用webpack可以对后台代码进行打包 ()

- ☐ A 对
- ☐ B 错

3

2分



active-class 是vue-router 模块的 router-link的属性 ()

- ☐ A 对
- ☐ B 错

4

2分



使用vue中axios发送请求可以不考虑跨域问题 ()

- ☐ A 对
- ☐ B 错

5

2分



vue-router可以控制页面局部刷新 ()

- ☐ A 对
- ☐ B 错

四、客观填空题 共5题, 100分

1

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

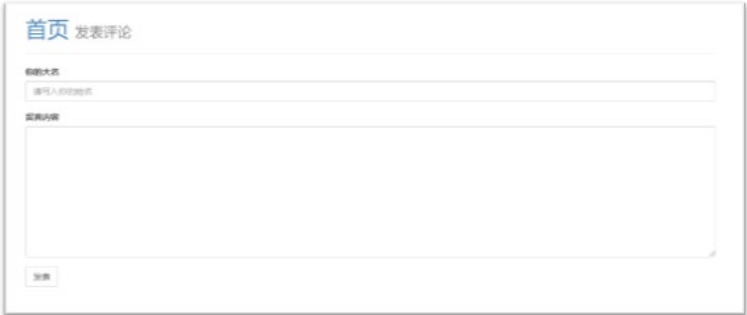


图2

【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>
      </div>
    </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.static({root: './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.render('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.body.comment; //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.redirect('/');
});
//监听启动3000端口
app.listen(3000,function () {
//控制台打印信息
  console.log('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） （2） （3） （4） （5） （6） （7） （8）

- 1、
- 2、
- 3、
- 4、
- 5、
- 6、
- 7、
- 8、



9、
10、

2

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、
- 2、
- 3、
- 4、
- 5、
- 6、

- 7、
- 8、
- 9、
- 10、



3

20分

阅读下列说明，效果图和代码，进行网站需求开发，回答问题1至问题4。

【说明】

很多互联网网站现在都做的很花里胡哨，领导提出公司网站也要加上比较绚丽的特效，由于公司网站是深色背景，美工提出来随着鼠标出现彩色小球效果比较好，作为前端的你发现这种效果可以用代码实现。

项目名称为小球快动moveball，项目包括index.html、css文件夹和js文件夹，css文件夹中包含index.css文件，js文件夹包含jquery.min.js、underscore.js和index.js文件。完成的效果图如图4-1所示：

【效果图】

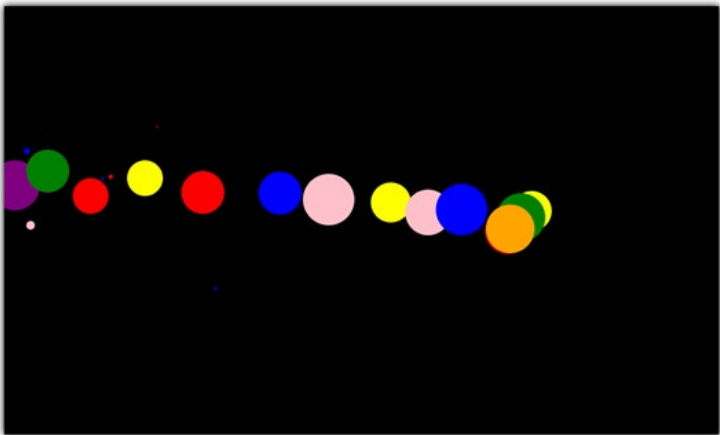


图1

【素材】

素材.rar

【html代码：主页index.html】

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>小球快动</title>
    <link rel="stylesheet" type="text/css"
href="css/index.css">
  </head>
  <body>
    <canvas id="canvas">
      当前的浏览器不支持该版本!
    </canvas>
    <script type="text/javascript" src="js/jquery.min.js" >
</script>
    <script type="text/javascript" src="js/underscore.js" >
</script>
```

```

        <script src="js/index.js"> </script>
    </body>
</html>
【js代码： index.js】
//1.获取当前画布
//es6中通常用于定义常量的声明关键字
(          ) canvas=document.getElementById("canvas"),
//获取上下文
//获取2d的上下文
ctx=canvas. (          ) ('2d');
//设置画布大小
canvas.width=1000;
canvas.height=600;
canvas.style.backgroundColor='#000';
//2.创建小球类
class Ball{
    constructor(x,y,color){
        this.x=x;
        this.y=y;
        this.color=color;
        this.r=40
    }
    /*
    * 绘制小球
    */
    render(){
        ctx.save();
        ctx.beginPath();
        ctx.arc(this.x,this.y,this.r,0,Math.PI*2);
        ctx.fillStyle=this.color;
        ctx.fill();
        ctx.restore();
    }
}
//3.会移动的小球类（继承Ball）
//es6中新增的继承关键字
class MoveBall (          ) Ball{
    constructor(x,y,color){
        //继承父类的构造方法
        (          ) (x,y,color);
        //量的变化
        this.dx=_random(-5,5);
        this.dy=_random(-5,5);
        this.dr=_random(1,3)
    }
    /*
    * 动态更新数据
    */
    upDate(){
        this.x += this.dx;
        this.y += this.dy;
        this.r -= this.dr;
        if(this.r < 0){
            this.r=0;
        }
    }
}
//4.实例化小球
//es6中新增的代替var声明变量的关键字
//使用该关键字声明了变量， 那这个变量就有一个块级作用域范围
(          ) ballArr=[],
colorArr=['red','green','blue','yellow','purple','pink','orange'];
//5.监听鼠标移动
//es6中箭头函数的使用
canvas.addEventListener('mousemove', (          ) {
    ballArr.push(new
MoveBall(e.offsetX,e.offsetY,colorArr[_random(0,colorArr.length-
1)]));
    console.log(ballArr);
});
```

```
//6.开启定时器
setInterval(function(){
  //清屏
  ctx.clearRect(0,0,canvas.width,canvas.height);
  //绘制
  for(let i=0;i<ballArr.length;i++){
    ballArr[i].render();
    ballArr[i].upDate();
  }
},50);
//实例化Ball这个类
let ball= (      ) Ball(50,50,'red');
/绘制小球
(      ) .render();
```

- 【问题】
- 1、分析JavaScript代码，回答下列问题，填写（1）至（8）。
- 2、ES6是不能直接在浏览器中执行，现今流行的转换插件有（ ）？

- A. jquery
- B. bootstrap
- C. liferay
- D. babel
- 3、根据您对ES6中对解构赋值的学习，下列代码打印的值为（ ）？

```
let [x,y='b'] = ['a'];
console.log(y);
```

A. b

B. a

C. undefined

D. NaN

- 1、
- 2、
- 3、
- 4、
- 5、
- 6、
- 7、
- 8、
- 9、
- 10、



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PC端的新项目，项目为出差管理系统，项目名为business-out，项目前端技术路线为vue+elementUI。

现在你需要使用Vue.js框架搭建一个整体架构，项目布局为左右结构，左边是功能菜单，右边是对应展示和操作区域，点击左边的菜

单通过切换vue组件相应的替换右边的内容。项目核心包括assets文件夹、components文件夹、router文件夹、App.vue文件和main.js文件， assets文件夹包
含logo.png和triangle.png， components文件夹包
含home.vue， router文件夹包含index.vue， 其他例如node模块和package.json配置文件类的这里不做描述。
实际完成的效果图如图：

【效果图】



【vue代码： App.vue】

```
<template>
<template>
  <div id="app">
    <el-container>
      <el-aside
        :width="nav_width"
        style="background: rgb(0, 21, 41);transition: 0.2s;overflow-x:hidden"
      >
        <div id="nav_title">
          
          <span style="position: relative; top: 3px; font-size:
17px;">出差管理系统</span>
        </div>
        <div id="nav">
          <el-menu
            :default-active="url"
            class="el-menu-vertical-demo"
            @open="handleOpen"
            @close="handleClose"
            :collapse="isCollapse"
            background-color="rgb(0, 21, 41)"
            text-color="#fff"
            router>
            <el-menu-item index="/Home">
              <i class="el-icon-menu"> </i>
              <span slot="title">主页（HOME） </span>
            </el-menu-item>
            <el-menu-item index=" ">
              <i class="el-icon-document"> </i>
              <span slot="title">我要出差</span>
            </el-menu-item>
            <el-menu-item index=" ">
              <i class="el-icon-share"> </i>
              <span slot="title">我要报销</span>
            </el-menu-item>
            <el-menu-item index=" ">
```

```

        <i class="el-icon-setting"></i>
        <span slot="title">我的审批</span>
      </el-menu-item>
      <el-menu-item index=" ">
        <i class="el-icon-picture-outline"></i>
        <span slot="title">系统管理</span>
      </el-menu-item>
      <el-menu-item index="">
        <i class="el-icon-upload"></i>
        <span slot="title">我的资料</span>
      </el-menu-item>
    </el-menu>
  </div>
</el-aside>
<el-container>
  <el-header style="border-bottom: solid 1px #e6e6e6;line-
height:68px;">
    <i
      :class="isCollapse?'el-icon-d-arrow-right':'el-icon-d-
arrow-left'"
      style="font-size:20px;cursor: pointer;"
      @click="nav_switch"
    ></i>
  </el-header>
  <el-main >
    <!-- <transition name="fade"> -->
    <router-view/>
    <!-- </transition> -->
  </el-main>
</el-container>
</el-container>
</div>
</template>
<script>
export default {
  : "App",//绑定容器
  data() {
    return {
      isCollapse: false,
      nav_width: "255px",
      url:"/Home",
    };
  },
  :{ //监听路由变化
    $ ( to , from ){//配置路由
      console.log( to , from );
      //将要去的路径赋值给当前url
      this.url = .path
    }
  },
  (){//vue创建周期完成后
    //往路由的数组中添加首页地址
    this.$router. ("/Home");
  },
  //定义方法
  : {
```

```
handleOpen(key, keyPath) {
  //导航展开/收起
  console.log(key, keyPath);
},
handleClose(key, keyPath) {
  //导航展开/收起
  console.log(key, keyPath);
},
nav_switch() {
  //导航左右放大缩小
  this.isCollapse = !this.isCollapse;
  if (this.isCollapse) {
    this.nav_width = "64px";
  } else {
    this.nav_width = "255px";
  }
}
};
</script>
<style>
样式这里就不贴出来了
</style>
【js代码： router/index.js】
import Vue from 'vue'
import Router from 'vue-router'
import Home from '@/components/home/home'
```

```
Vue.use(Router)

export default new Router({
  routes: [
    { //首页路由配置
      path: '/Home', //路径
      name: 'Home', //名称
      component: Home //绑定组件
    }
  ]
})
```

- 【问题】（20分）
- 1、根据项目上下文，仔细查看App.vue文件，填写（1） - （7）空。
 - 2、根据掌握的vue-router知识，填写router/index.js中（8） - （10）空。

 [素材.zip](#)

- 1、
- 2、
- 3、
- 4、

- 5、
- 6、
- 7、
- 8、
- 9、
- 10、

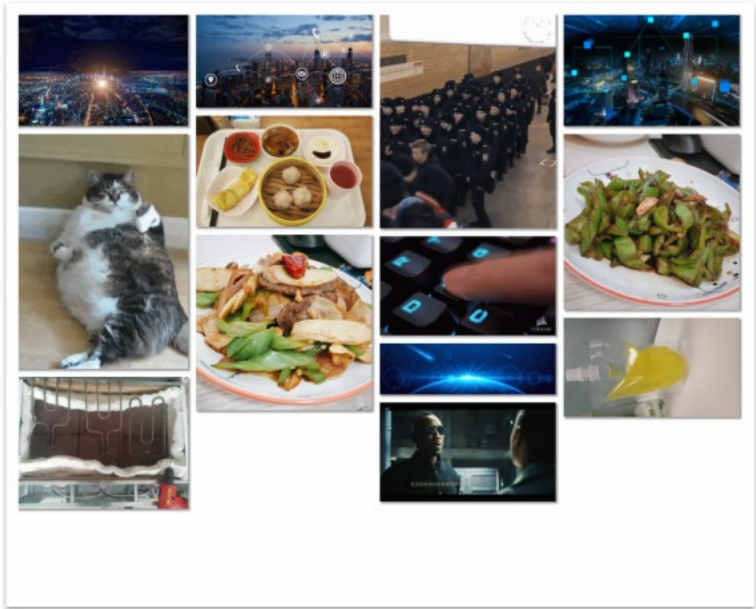


5

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>
  </body>
</html>
```

```
<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>
<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、
- 2、
- 3、
- 4、
- 5、
- 6、
- 7、
- 8、
- 9、
- 10、



