

答题中

Web前端开发高级模拟测试（四）

共54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

调用Canvas对象的哪个方法来获取绘图环境（）

- 2分
- (A) getCanvas
- (B) getContent
- (C) getContext
- (D) getClient

2

哪项不是webpack的常用命令（）

- 2分
- (A) webpack --output
- (B) webpack --config
- (C) webpack --mode
- (D) webpack --test

3

在数组的解构赋值中，var [a,b,c] = [1,2]结果中，a、b、c的值分别是（）

- 2分
- (A) 1 2 undefined
- (B) 1 2 null
- (C) 1 2 2
- (D) 抛出异常

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

4

2分



在Less中，表示上一层选择器的是（）

- ☐ (A) *
- ☐ (B) \$
- ☐ (C) @
- ☐ (D) &

5

2分



关于AJAX的说法错误的是（）

- ☐ (A) 数据可以使用JSON格式
- ☐ (B) 可以实现跨域通信
- ☐ (C) 可以使用GET或者POST请求
- ☐ (D) 不能解决不同域名的跨域问题

6

2分



下列哪项不是jQuery Mobile中表示过渡的data-transition属性的值（）

- ☐ (A) fade
- ☐ (B) turn
- ☐ (C) slide
- ☐ (D) down

7

2分



在jQuery Mobile弹出框中，在弹出框内的元素上使用以下哪一个正确的属性，点击该元素弹出框会关闭（）

- ☐ (A) data-rel="back"
- ☐ (B) data-rel="close"
- ☐ (C) data-rel="home"
- ☐ (D) data-rel="off"

8

2分

在Less中，表示导入的关键字是（）

- ☐ (A) import
- ☐ (B) export
- ☐ (C)



- ☐ include
- ☐ link

9

2分



在ES6中，增加对数组扩展，[1,2,3].fill(4)的结果是（）

- ☐ [4]
- ☐ [1,2,3,4]
- ☐ [4,1,2,3]
- ☐ [4,4,4]

10

2分



关于webpack配置描述不正确的是（）

- ☐ entry是入口配置，是指页面中的入口文件
- ☐ output出口文件，是指生成的文件
- ☐ 默认入口文件是./src/main.js
- ☐ 在出口文件的配置中可以指定path和filename

11

2分



实现一个元素的旋转效果可以使用CSS3里的什么方法（）

- ☐ translateY
- ☐ scaleY
- ☐ rotate
- ☐ skew

12

2分



为了实现响应式开发，可以使用相对单位。下列哪项不是相对单位（）

- ☐ px
- ☐ em
- ☐ rem
- ☐ %

13

在Less中，定义变量的符号是（）

2分



- (A) *
- (B) \$
- (C) @
- (D) &

14

2分



- 下列哪项是Vue.js的状态管理工具（）
- (A) vuex
 - (B) vue-router
 - (C) axios
 - (D) mint-ui

15

2分



- 关于html优化描述错误的是（）
- (A) 正确闭合html标签
 - (B) 结构、样式、行为有效分离
 - (C) 不使用table标签
 - (D) 使用语义化标签

16

2分



- 在Vue.js内置的指令错误的是（）
- (A) v-if
 - (B) v-bind
 - (C) v-on
 - (D) v-elseif

17

2分



- 在Vue.js的表单中，使用什么指令进行数据双向绑定（）
- (A) v-bind
 - (B) v-model
 - (C) v-html
 - (D) {{}}

18

2分



在ES6语法中创建变量的关键字是（）

- ☐ (A) var
- ☐ (B) let
- ☐ (C) const
- ☐ (D) class

19

2分



let [a, ...b] = [1, 2, 3];运行代码b的值是（）

- ☐ (A) 错误
- ☐ (B) 2
- ☐ (C) [2]
- ☐ (D) [2,3]

20

2分



下列哪项不是audio和video的事件（）

- ☐ (A) onclick
- ☐ (B) onplay
- ☐ (C) onerror
- ☐ (D) onstop

21

2分



对webpack描述不正确的是（）

- ☐ (A) webpack是一个模块构建工具
- ☐ (B) webpack可以很好的应用于单页面
- ☐ (C) webpack是唯一的前端构建工具
- ☐ (D) webpack同时支持require（）和import模块语法

22

2分

根据所学知识分析，在下列数据结构中，不能被for...of遍历的是（）

- ☐ (A) String字符串
- ☐ (B) Object对象
- ☐ (C) Array数组



—

(D) Set结构

23

2分



下列对css权重表述错误的是 ()

- (A) ID选择器权重100
- (B) 类选择器权重10
- (C) 标签选择器权重1
- (D) !important权重1000

24

2分



HTML5为了克服cookie带来的问题，推出了 ()

- (A) canvas
- (B) svg
- (C) Web Storage
- (D) audio和video

25

2分



webpack中用于解析css语法的loader是 ()

- (A) css-loader
- (B) url-loader
- (C) file-loader
- (D) less-loader

26

2分



下列哪项不是jQuery Mobile中的图标 ()

- (A) plus
- (B) info
- (C) test
- (D) home

27

2分

关于Node.js的说法错误的是 ()

- (A) Node.js 用户写的代码是单线程的



- ☐ B Node.js 使用js语法
- ☐ C Node.js 基于Chrome的V8引擎
- ☐ D Node.js是重要的前端技术

28

2分



下列是jQuery Mobile中Grid网格布局容器正确的使用方法是（）

- ☐ A data-role="ui-grid-a"
- ☐ B class="ui-block-a"
- ☐ C class="ui-grid-a"
- ☐ D data-role="ui-block-a"

29

2分



在ES6语法中对module描述错误的是（）

- ☐ A export导出
- ☐ B import导入
- ☐ C export default默认导出
- ☐ D import * as重命名

30

2分



let string = "apple,banana,orange"; string.includes("banana");运行代码结果是（）

- ☐ A true
- ☐ B false
- ☐ C banana
- ☐ D 抛出错误

二、多选题 共15题，30分

1

2分

在jQuery Mobile中，如果使用data-role属性定义页面，值可以是（）

- ☐ A page
- ☐ B header



- ☐ content
- ☐ footer

2

2分



对于Express框架，下列描述正确的有（）

- ☐ Express是nodejs web应用框架
- ☐ Express可以设置中间件来响应HTTP请求
- ☐ 定义了路由表用于执行不同的HTTP请求动作
- ☐ 可以快速的搭建一个完整功能网站

3

2分



数组的扩展中，用于数组遍历的函数的有（）

- ☐ keys()
- ☐ entries()
- ☐ values()
- ☐ find()

4

2分



webpack中常用loader有哪些（）

- ☐ css-loader
- ☐ url-loader
- ☐ style-loader
- ☐ less-loader

5

2分



对于css样式的重置，说法正确的有（）

- ☐ 去除标签的默认样式，如input、li等
- ☐ HTML5的新标签设置display: block
- ☐ 重置一些元素的样式，如超链接等
- ☐ 把行内元素显示方式设置为block

6

2分



下列哪些是Web Storage提供的方法（）

- ☐ A key()
- ☐ B getItem()
- ☐ C setItem()
- ☐ D removeItem()

7

2分



关于对象的扩展，正确的是（）

- ☐ A ES6中对象的表示法更加简洁
- ☐ B 对象的属性名可以是表达式
- ☐ C 对象的方法名不可以是表达式
- ☐ D 当属性名为表达式的时候，需要使用[]标识

8

2分



在webpack 配置文件中配置 output 属性，下列哪些属于output的配置项是（）

- ☐ A path
- ☐ B staticPath
- ☐ C filename
- ☐ D file

9

2分



下列对于箭头函数的应用，正确的有（）

- ☐ A let f = a => a;
- ☐ B let f = a,b => a+b;
- ☐ C let f = () => 1+1;
- ☐ D let f = (id,name) => {id: id, name: name};

10

2分

下列对SVG矢量图形说法正确的是（）

- ☐ A SVG 意为可缩放矢量图形（Scalable Vector Graphics）



- ☐ B SVG使用XML格式定义图像
- ☐ C SVG可以使用一些绘图软件生成
- ☐ D SVG是万维网联盟的标准

11

2分



在jQuery Mobile中，可以使用data-position属性来设置页眉页脚的定位，可以使用的值（）

- ☐ A Inline
- ☐ B Fixed
- ☐ C Top
- ☐ D Bottom

12

2分



常用的JavaScript代码调试语句有（）

- ☐ A console.log()
- ☐ B alert()
- ☐ C debugger
- ☐ D try...catch

13

2分



关于ES6语法中Symbol对象描述正确的是（）

- ☐ A Symbol对象是唯一的标识符
- ☐ B Symbol对象可用作对象的唯一属性名
- ☐ C 使用typeof检查Symbol对象，返回Object
- ☐ D 使用全局函数Symbol()创建Symbol对象

14

2分



下列关于Vue指令正确的是（）

- ☐ A v-for循环遍历数据
- ☐ B v-show和v-if作用完全一样
- ☐ C v-blind可以简写成：
- ☐

☐ D v-on可以简写成@

15

2分



下列哪些是ES6中对象的扩展（）

- ☐ A Object.is()
- ☐ B Object.assign()
- ☐ C Object.getOwnPropertyDescriptor
- ☐ D Object.keys()

三、判断题 共5题，10分

1

2分



ES6新增了两种数据结构Map和Set（）

- ☐ A 对
- ☐ B 错

2

2分



Vue框架的核心思想是组件化开发和数据驱动（）

- ☐ A 对
- ☐ B 错

3

2分



使用图片精灵技术可以减少前端向服务端的请求次数，减轻服务器压力（）

- ☐ A 对
- ☐ B 错

4

2分



Axios 是一个基于 promise 的 HTTP 库（）

- ☐ A 对
- ☐ B 错

5

webpack是前端唯一的模块化构建工具（）



2分

- (A) 对
- (B) 错

四、客观填空题 共4题，100分

1

25分

阅读下列说明，效果图和代码，填写1到10所空位置。

【说明】

本项目名称listpage，使用jQuery mobile布局的移动端项目。项目目录中有index.html,css目录和js目录。其中css目录存放样式文件，js目录存放jQuery.min.js和jQuery.mobile.min.js。

本项目实现首页、通讯录页面和搜索页面切换功能。



(2) listProduct_server项目目录结构如下图所示：



【项目1效果图index.html】



```
【项目1代码：主页index.html】
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width, initial-
scale=1">
  <link rel="stylesheet" href="/css/jquery.mobile-1.4.5.min.css">
  <script src="/js/jquery.min.js"> </script>
  <script src="/js/jquery.mobile-1.4.5.min.js"> </script>
</head>
<body>
<!-- 第一页 -->
<!-- 在（1）处补全代码，作用是页 -->
  <div data-role="          " id="pageone">
    <!--在（2）处补全代码，页面头部 -->
    <div data-role="          ">
      <h1>欢迎使用jQuery Mobile</h1>
      <!-- 在（3）处补全代码，导航栏 -->
      <div data-role="          ">
        <ul>
          <li><a href="#pageone" data-transition="slide"
data-direction="reverse" class="ui-btn-active"
data-icon="home">首页</a>
          </li>
          <li><a href="#pagetwo" data-transition="slide"
data-direction="reverse" data-icon="bars">通讯录</a>
          </li>
          <li><a href="#pagethree" data-transition="slide"
data-direction="reverse" data-icon="search">搜索</a> </li>
        </ul>
      </div>
    </div>
    <!-- 在（4）处补全代码，页面主体 -->
    <div data-role="          " class="ui-content">
      <!-- 在（5）（6）处补全代码，分别设置折叠框集合和折
叠框 -->
      <div data-role="          ">
        <div data-role="          ">
          <h1>新闻标题1</h1>
          <p>新闻1的内容，新闻1的内容，新闻1的内容，新
闻1的内容。</p>
          <p>新闻1的内容，新闻1的内容，新闻1的内容，新
闻1的内容。</p>
          <p>新闻1的内容，新闻1的内容，新闻1的内容，新
闻1的内容。</p>
        </div>
        <div data-role="collapsible">
          <h1>新闻标题2</h1>
          <p>新闻2的内容，新闻2的内容，新闻2的内容，新
闻2的内容。</p>
          <p>新闻2的内容，新闻2的内容，新闻2的内容，新
闻2的内容。</p>
          <p>新闻2的内容，新闻2的内容，新闻2的内容，新
闻2的内容。</p>
          <p>新闻2的内容，新闻2的内容，新闻2的内容，新
闻2的内容。</p>
        </div>
      </div>
    </div>
  </div>
```

```

        </div>
    </div>
</div>
<div data-role="footer" data-position="fixed">
    <h1>底部文本</h1>
</div>
</div>
<!-- 第二页 -->
<div data-role="page" id="pagetwo">
    <div data-role="header">
        <h1>通讯录</h1>
        <div data-role="navbar">
            <ul>
                <li><a href="#pageone" data-transition="slide"
data-direction="reverse" data-icon="home">首页</a>
                </li>
                <li><a href="#pagetwo" data-transition="slide"
data-direction="reverse" data-icon="bars"
class="ui-btn-active">通讯录</a> </li>
                <li><a href="#pagethree" data-transition="slide"
data-direction="reverse" data-icon="search">搜索</a> </li>
            </ul>
        </div>
    </div>
<div data-role=" main" class="ui-content">
    <!-- 在 (7) 处补全代码， 列表视图 -->
    <ul data-role=" "
        <li><a href="#">列表项</a> </li>
        <li><a href="#">列表项</a> </li>
        <li><a href="#">列表项</a> </li>
    </ul>
</div>
    <!-- 在 (8) 处补全代码， 固定在页面底部 -->
    <div data-role="footer" data-position=" ">
        <h1>底部文本</h1>
    </div>
</div>
<!-- 第三页 -->
<div data-role="page" id="pagethree">
    <div data-role="header">
        <h1>搜索</h1>
        <div data-role="navbar">
            <ul>
                <li><a href="#pageone" data-transition="slide"
data-direction="reverse"
data-icon="home">首页</a>
                </li>
                <li><a href="#pagetwo" data-transition="slide"
data-direction="reverse"
data-icon="bars">通讯录</a> </li>
                <!-- 在 (9) 处补全代码， 图标是search -->
                <li><a href="#pagethree" data-transition="slide"
data-direction="reverse"
="search"
class="ui-btn-active">搜索</a> </li>
            </ul>
        </div>
    </div>
</div>
```

```
</div>
<div data-role="main" class="ui-content">
  <form method="post" action="">
    <label for="fname">请输入搜索内容</label>
    <input type="text" name="fname" id="fname">
    <!-- 在（10）处填写input类型为提交按钮 -->
    <input type="          " value="提交按钮">
  </form>
</div>
<div data-role="footer" data-position="fixed">
  <h1>底部文本</h1>
</div>
</div>
</body>
</html>
```

【问题】根据jQuery Mobile知识，正确填写所缺代码。

 [素材.zip](#)

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



2

25分

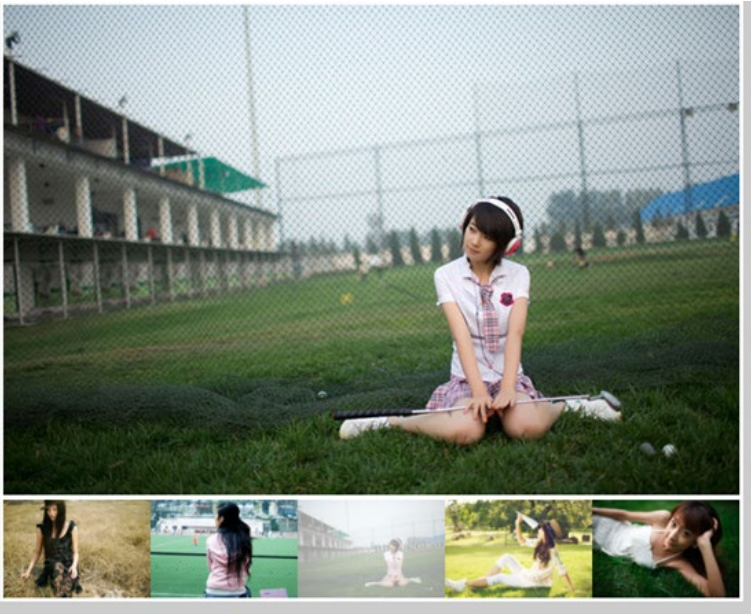
阅读下列说明、效果图和代码，填写（1）至（10）代码。

【说明】

现在有一个使用es6语法实现的效果。如图所示，点击小图，大图实现切换。被点击小图有半透明效果。

项目名称为changeImg。项目目录下有index.html文件和img目录，img目录中存放使用到的图片。

【效果图】



点击图片切换效果首页

```
【代码： index.html】
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>点击展示大图</title>
  <style>
    *{margin:0;padding:0}
    body{background: #ccc;}
    .box {
      width:600px;
      height:auto;;
      margin: auto;
      border: 3px solid #fff;
      background: #fff;
      display:flex;
      flex-direction: column;
      flex-wrap: nowrap;
      justify-content:space-around;
    }
    .smlmg{display: flex;flex-direction: row;justify-content:
space-around;}
    .smlmg img{width:120px;flex:0 1 auto;}
    @keyframes fadeIn {
      to {
        opacity: 1;
      }
    }
    .fadeIn {
      opacity: 0;
      animation: fadeIn 0.5s ease-in 1 forwards;
    }
  </style>
</head>
```



```
<body>
  <div>
    <!-- 展示大图 -->
    <div>
      
    </div>
    <!-- 展示小图 -->
    <div>
      
      
      
      
      
    </div>
  </div>
  <script>
    //在(1)处填写修饰常量的关键字。
    [bigImg, imgs] =
[document.querySelector("#bigImg"),
document.querySelectorAll(".smlImg img")];
    // 给第一张图片添加默认选中效果
    imgs[0].style.opacity = 0.4;
    // 在 (2) 处补全代码，循环遍历imgs，在 (3) 处补全代
码，设置箭头函数的形参
    imgs.      (      => {
      // 在 (4) 处补全代码，给当前的img添加事件监听
      img.      ("click", function(e){
        // 在 (5) 处补全代码，遍历imgs，每一个img透明度
为1
        imgs.      (img => {
          img.style.opacity = 1;
        });
        //在 (6) 处补全代码，拿到点击图片的路径，赋值给大
图的图片路径
        bigImg.src = e.target.      ;
        // 在 (7) 处补全代码，添加类名fadeIn
        bigImg.classList.      ("fadeIn");
        //在 (8) 处补全代码，设置超时计时器，在 (9) 处补全
代码，移除fadeIn类
        () => {
          bigImg.classList.      ("fadeIn");
        }, 500);
        //在 (10) 处补全代码，给点击图片一个透明度
        e.target.style.      = 0.4;
      });
    });
  </script>
</body>
</html>
```

【问题】根据es6语法，填写 (1) 至 (10) 。

 素材.zip

1、

2、

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



3

25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>
<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-row>
    </div>
  </div>
</template>

```

```
<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,当网络请求失败时
            .          (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、
- 2、
- 3、
- 4、
- 5、
- 6、
- 7、
- 8、
- 9、
- 10、
- 11、
- 12、
- 13、
- 14、
- 15、
- 16、
- 17、
- 18、
- 19、
- 20、



4

25分

阅读下列说明，根据要求，完成以下题目。

【说明】
本项目是一个webpack模块化构建工具的演示项目，项目名称为webpackDemo。
项目根目录下的webpack.config.js文件为webpack的配置文
件。src目录为工程目录，其中的index.js为入口文件，css存放构建

之前的样式文件，images存放图片，fun.js普通的js文件。
根目录下的dist文件夹为构建之后的文件夹，其中的index.js是构建之后的输入文件，index.html文件引用构建之后的index.js文件，展示最终效果。
【效果】



【webpack配置文件代码：webpack.config.js】

```
var path = require('path');
module.exports = {
  // 在（1）处补全代码，填写入口关键字
  entry: './src/index.js',
  // 在（2）处补全代码，填写出口文件关键字
  output: {
    path: path.resolve(__dirname, 'dist/'),
    filename: 'index.js'
  },
  // 在（3）处补全代码，填写模块关键字
  module: {
    rules: [
      // 在（4）（5）处填写处理css样式文件的loader
      { test: /\.css$/, use: ['style-loader', 'css-loader'] },
      // 在（6）处填写处理图片文件的loader
      { test: /\.(jpg|png|gif)$/, use: [{loader: 'url-loader', options: {limit:18192}} ] },
      {
        test: /\.js$/,
        exclude: /(node_modules|bower_components)/,
        use: {
          loader: 'babel-loader',
          options: {
            presets: ['@babel/preset-env']
          }
        }
      }
    ]
  }
};
```

【webpack入口文件代码：index.js】

// 在（7）处补全代码，使用对象解构获取对象

```
import {      ,foo} from './fun'
// 在（8）处补全代码，引入css
      "./css/index.css"

//在（9）处填写html文件的合适的类名
document.getElementsByClassName("      ")
[0].innerHTML=bar('100000');
【webpack项目html文件代码：index.html】
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
</head>
<body>
  <div id="box">
    <h3>webpack</h3>
    <div></div>
    <p>
      本质上，webpack 是一个现代 JavaScript 应用程序的静态
      模块打包器(module bundler)。当 webpack 处理应用程序时，它会
      递归地构建一个依赖关系图(dependency graph)，其中包含应用程
      序需要的每个模块，然后将所有这些模块打包成一个或多个
      bundle。
    </p>
    <p>
      今日访问量<b></b>
    </p>
  </div>
  <!-- 在（10）处填写输出的js文件，以./相对路径开始 -->
  <script src="      "></script>
</body>
</html>
```

【问题】根据webpack知识，填写（1）至（10）。

 [素材.zip](#)

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



9、

10、



技术支持：奥鹏教育

京公网安备 11010802029644号