

答题中

Web前端开发高级模拟测试（二）

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

2分



以下属于vue指令的是（）

- (A) for
- (B) ng-model
- (C) v-bind
- (D) if

2

2分



jquery-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

3

2分



关于模板字符串，下列说法不正确的是：（）

- (A) 使用反引号标识
- (B) 插入变量的时候使用\${}
- (C) 所有的空格和缩进都会被保留在输出中
- (D) \${}中的表达式不能是函数的调用

4

关于箭头函数的描述，错误的是：（）

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分



- Ⓐ 使用箭头符号=>定义
- Ⓑ 参数超过1个的话，需要用小括号（）括起来
- Ⓒ 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
- Ⓓ 函数体内的 this 对象，绑定使用时所在的对象

5

2分



关于 javascript 模块化，下列描述错误的是（）。

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

6

2分



关于Iterator遍历器的说法，错误的是：（）

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

7

2分



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

- Ⓐ 服务器端错误
- Ⓑ url重定向
- Ⓒ 客户端
- Ⓓ 服务器端不知道客户端要请求哪个资源

8

2分



JavaScript中类的继承使用的关键字是：（）

- Ⓐ extends
- Ⓑ inherit
- Ⓒ extend
- Ⓓ

(D) base

9

2分



php 关于类继承说法正确的是

- (A) 子类可以直接调用基类的所有方法
- (B) 子类可以重写基类的方法，但是final修饰的类除外
- (C) 子类可以直接继承多个基类
- (D) 子类可以重写基类 protected 级别方法为 private 级别

10

2分



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：（）

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

11

2分



nodejs的哪个模块可以搭建服务器？

- (A) Http
- (B) url
- (C) path
- (D) fs

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()

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该元素



- Ⓐ v-show隐藏元素后再页面中仍然显示该元素
- Ⓑ 二者功相同，并无区别
- Ⓒ 以上说法都不对

14

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？

- Ⓐ ./
- Ⓑ ../
- Ⓒ /
- Ⓓ //

15

2分



下述有关HTML Doctype的描述，说法错误的是（）

- Ⓐ 他的责任就是告诉浏览器文档使用哪种HTML或XHTML规范
- Ⓑ 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
- Ⓒ 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
- Ⓓ 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

16

2分



less声明变量的符号是（）

- Ⓐ \$
- Ⓑ \*
- Ⓒ @
- Ⓓ &

17

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

18

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

19

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

20

2分



在bootstrap4中，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（ ）

- (A) form-group
- (B) form-horizontal
- (C) form-inline
- (D) form-control

21

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



④ 可以先声明，不赋值。

22

2分



Nodejs的特点不包括下列哪个？

- ① 单线程
- ② 多线程
- ③ 非阻塞I/O
- ④ 事件驱动

23

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”

- ① log\_errors
- ② display\_errors
- ③ short\_open\_tag
- ④ date.timezone

24

2分



关于vue生命周期说法正确的是（）

- ① Vue 实例从创建到销毁的过程，就是生命周期
- ② 生命周期一共有4个钩子函数
- ③ DOM在updated钩子函数后渲染完毕
- ④ 以上说法都不对

25

2分



关于canvas不正确的是()

- ① canvas是HTML5新增的标签，允许脚本动态渲染位图像
- ② canvas是作为Flash的替代技术出现
- ③ canvas拥有多种绘制路径、矩形、圆形、字符及添加图像的方法
- ④ ie9起已经可以支持canvas

26

下面不属于关键字let的特点的是：（）



2分



- Ⓐ 只在 let 命令所在的代码块内有效
- Ⓑ 会产生变量提升现象
- Ⓒ 同一个作用域，不能重复声明同一个变量
- Ⓓ 不能在函数内部重新声明参数

27

2分



img-responsive类可以让图片支持响应式布局，它的实现原理是（）

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

28

2分



执行[1,2,3].fill(4)的结果是：（）

- Ⓐ [4]
- Ⓑ [1,2,3,4]
- Ⓒ [4,1,2,3]
- Ⓓ [4,4,4]

29

2分



在javascript中，以下表达式转布尔值为true的是（）

- Ⓐ 0
- Ⓑ undefined
- Ⓒ []
- Ⓓ null

30

2分

在javascript中，以下代码的执行结果是（）  
`<ul> <li>1</li>  
> <li>2</li> <li>3</li> </ul> <script> var li=document.  
getElementsByTagName("li"); for(var i=0;i<li.length;i++){ l  
i[i].onclick=function(){ console.log(i); } } </script>`

- Ⓐ 0 1 2 3
- Ⓑ 0 1 2
- Ⓒ



- (C) 3 3 3
- (D) 2 2 2

二、多选题      共15题，30分

1

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

2

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

3

2分



下列数据结构中, 能被for...of遍历的是: ()

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

4

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();

5

2分

关于webpack说法正确的是（）

- ☐ A webpack是一个模块打包工具
- ☐ B 可通过plugin扩展，方便、灵活
- ☐ C 只能用于采用模块化开发的项目
- ☐ D 良好的开发体验



6

2分

以下属于Audio/Video的方法有？

- ☐ A load()
- ☐ B play()
- ☐ C pause()
- ☐ D stop()



7

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
- ☐ C Symbol('same') === Symbol('same') 结果为true
- ☐ D 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。



8

2分

以下是canvas的方法是（）

- ☐ A getContext()
- ☐ B fill()
- ☐ C stroke()
- ☐ D controller()



9

2分



以下属于vuex属性的是（）？

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

10

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

11

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初始状态可以状变成fulfilled成功
- ☐ C rejected失败不可以状变成pending初始状态
- ☐ D rejected失败可以状变成fulfilled成功

12

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

13

2分

PHP表单的提交方法有:

- ☐ A post
- ☐



- ☐ B request
- ☐ C get
- ☐ D querystring

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

15

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
- ☐ D 使用add方法添加已经存在的成员会报错

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

2

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

3

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

4

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

5

2分



Nodejs是一门后端语言，使用js书写

- ☐ (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，进行网站性能优化，回答问题1至问题17。

【说明】  
本项目使用jquery mobile开发的移动端项目，项目名称是blog，本项目下包含index.html、js目录和css目录，其中js目录包括jquery.min.js 和jQuery.mobile.min.js，css目录中包含jquery.mobile.min.css。

【效果】

页面一效果



页面二效果



【素材】

 素材.rar

【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>
  <!-- 分别在 (1) (2) (3) 处补全代码，导入jQuery Mobile样式文件和js文件，以./相对路径开始 -->
  <link rel="stylesheet" href="( )">
  <script src=" ( ) "></script>
  <script src=" ( ) "></script>
</head>
<body>
  <!-- 在 (4) 处补全代码，在浏览器中显示的页面，根据下面代码，补齐 (5) 处代码 -->
```

```
<div data-role=" ( ) " id=" ( ) ">
<!-- 在 (6) 处补全代码，在页面顶部创建工具条 -->
  <div data-role=" ( ) ">
    <h1>欢迎来到我的主页</h1>
  </div>
<!-- 在 (7) 处补齐代码，定义页面内容 -->
  <div data-role=" ( ) ">
    <p>欢迎! 点击以下链接跳转到第二个页面。 </p>
    <!-- 在 (8) 处根据下文补全代码 -->
    <a href=" ( ) ">跳转到第二个页面</a>
  </div>
<!-- 在 (9) 处补全代码，用于创建页面底部工具栏，固定在页面底部，全屏模式 -->
  <div data-role=" ( ) " data-position=" ( ) " data-fullscreen=" ( ) ">
    <h1>底部文本</h1>
  </div>
</div>
<div data-role="page" id="pagetwo">
  <div data-role="header" data-position="fixed" data-fullscreen="true">
    <h1>欢迎来到我的主页</h1>
  </div>
  <div data-role="main" style="padding-top:40px;">
    <!-- 在 (12) 处补全代码，可折叠集合 -->
    <div data-role=" ( ) ">
      <!-- 在 (13) (14) 处补全代码，可折叠内容块，箭头向右-->
      <div data-role=" ( ) " data-collapsed-icon=" ( ) " data-mini="true">
        <h1>特别关心</h1>
        <!-- 在 (15) 处补全代码，显示视图列表 -->
        <ul data-role=" ( ) ">
          <li><a href="#">张三</a></li>
          <li><a href="#">李四</a></li>
          <li><a href="#">王二</a></li>
        </ul>
      </div>
      <div>
        <!-- 在 (16) (17) 处补全代码，页面从左向右滑动。 -->
        <a href="#pageone" data-transition=" ( ) " data-direction=" ( ) ">点击跳转页面</a>
      </div>
    </div>
  </div>
  <div data-role="footer" data-position="fixed" data-fullscreen="true">
    <h1>底部文本</h1>
  </div>
</div>
<div data-role="page" id="class-page">
  <div data-role="content">
    <ul data-role="listview" id="listview2">
      <li>向右滑动切换页面</li>
      <li>向右滑动切换页面</li>
      <li>向右滑动切换页面</li>
    </ul>
  </div>
</div>
</div>
</div>
</div>
</body>
</html>
```

【问题】请根据代码注释，回答问题，把答案填入对应位置。

1、

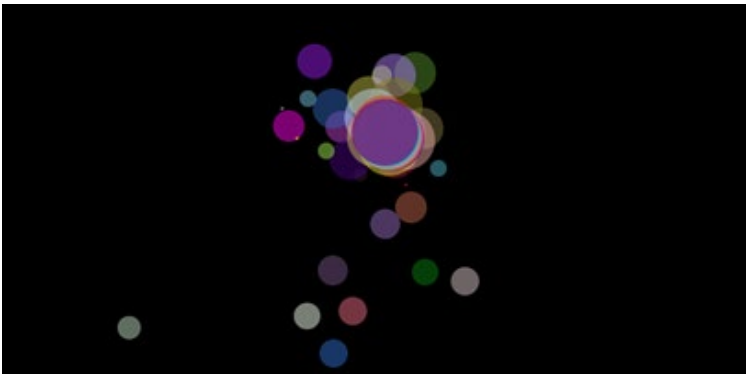
- 2、
- 3、
- 4、
- 5、
- 6、
- 7、
- 8、
- 9、
- 10、
- 11、
- 12、
- 13、
- 14、
- 15、
- 16、
- 17、




2

25分

阅读下列说明、效果图和代码，填写（1）至（10）代码。  
【说明】  
项目一为使用es6语法实现的页面效果，项目名为ball。项目下有index.html和js目录，js目录下面有index.js文件。



【素材】  
 素材.rar  
【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>
  <!-- 在 (1) (2) 处补齐合适的标签 -->
  <      id="canvas"></      >
  <!-- 在 (3) 处补齐代码，使用./开始的相对路径 -->
  <script src="      "></script>
</body>
</html>
【代码： index.js】
// 定义一个取两个数之间的随机数的方法
let newMath = (x, y) => {
  let tmp = y - x;
  return Math.floor(Math.random() * tmp + x);
}
//生成一个随机的色值
let randomRgbColor = () => { //随机生成RGBA颜色
  let r = Math.floor(Math.random() * 256); //随机生成256以内r值
  let g = Math.floor(Math.random() * 256); //随机生成256以
内g值
  let b = Math.floor(Math.random() * 256); //随机生成256以内b值
  return `rgba(${r},${g},${b},.5)`; //返回rgb(r,g,b,a)格式颜色
}
//获取当前的画布
const canvas = document.getElementById('canvas');
// 在 (4) 处填写方法，返回一个用于在画布上绘图的环境
const ctx = canvas.      ('2d');
canvas.width = 1000;
canvas.height = 600;
canvas.style.backgroundColor = '#000';
//在 (5) 处补齐代码， 创建小球的类
(      ) Ball {
  //在 (6) 处补齐代码， 创建Ball类的构造函数
  (      ) (x, y, color) {
    this.x = x;
    this.y = y;
    this.color = color;
    this.r = 40;
  }
  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();
  }
}
// 在 (7) (8) 处补齐代码， MoveBall类继承Ball类
(      ) MoveBall (      ) Ball {
  constructor(x, y, color) {
    //在 (9) 处补齐代码， 继承父类Ball的方法
    (      ) (x, y, color);
    this.dX = newMath(-5, 5);
    this.dY = newMath(-5, 5);
    this.dR = newMath(1, 3);
  }
  update() {
    this.x += this.dX;
    this.y += this.dY;
    this.r -= this.dR;
    if (this.r < 0) {
      this.r = 0;
    }
  }
}
```



```
    }  
  }  
}  
//储存变化的小球  
let ballArr = [];  
//监听鼠标的移动  
canvas.addEventListener('mousemove', function (e) {  
  // 在 (10) 处补齐代码，构造一个移动小球实例。  
  ballArr.push( (          ) (e.offsetX, e.offsetY,  
    randomRgbColor()));  
})  
// 开启定时器,并开始画图  
setInterval(function () {  
  ctx.clearRect(0, 0, canvas.width, canvas.height);  
  for (let i = 0, l = ballArr.length; i < l; i++) {  
    ballArr[i].render();  
    ballArr[i].update();  
  }  
}, 50  
)
```

【问题】根据说明，代码提示补全代码。

- 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Vue项目开发，回答问题1至问题18。

【说明】

该程序为一个vue演示项目，使用vue-cli自动搭建，使用路由实现单页面应用，使用mint-ui和mui第三方的UI库，界面更加友好，使用axios发起网络请求，并返回数据。

其中的App.vue是根组件，Home组件使用循环遍历data上的数据；User组件和Child组件用来演示父子组件传递数据；List组件用来演示axios请求以及数据展示在页面上。

【效果图】图3-1 展示Home组件 3-2展示User组件 3-3展示List组件



图1



图2



图3

【素材】

 [素材.7z](#)

【路由页面代码：index.js】

```
import Vue from 'vue'
import Router from 'vue-router'
import Home from '@components/Home'
import User from '@components/User'
import List from '@components/List'
import Setting from '@components/Setting'
// 在 (1) 处补齐代码，使用路由
Vue.use( ( ) )
export default new Router({
  routes: [
    {
      path: '/',
      name: 'home',
      component: Home
    }, {
      path: '/user',
      name: 'user',
      component: User
    }, {
      path: '/list',
      name: 'list',
      component: List
    }, {
      path: '/setting',
      name: 'setting',
      component: Setting
    }
  ],
  mode:"history",
  linkActiveClass:"mui-active"
})
```

【App组件代码：App.vue】

```
<template>
  <div id="app">
<!-- 在 (2) 处补齐代码，使用路由链接代替a标签 -->
    <mt-header fixed top>
      <( ) to="/" slot="left">
        <mt-button>首页</mt-button>
```

```

    </router-link>
  </mt-header>
  <router-view />
  <nav class="mui-bar mui-bar-tab">
    <router-link to="/" class="mui-tab-item mui-active">
      <span class="mui-icon mui-icon-home"> </span>
      <span class="mui-tab-label">首页</span>
    </router-link>
    <!-- 在 (3) 处补齐代码，根据路由规则，点击加载user组件 -->
    <router-link to="/" ( ) " class="mui-tab-item">
      <span class="mui-icon mui-icon-person"> </span>
      <span class="mui-tab-label">用户</span>
    </router-link>
    <router-link to="/list" class="mui-tab-item">
      <span class="mui-icon mui-icon-bars"> </span>
      <span class="mui-tab-label">列表</span>
    </router-link>
    <router-link to="/setting" class="mui-tab-item">
      <span class="mui-icon mui-icon-gear"> </span>
      <span class="mui-tab-label">设置</span>
    </router-link>
  </nav>
</div>
</template>
<script>
import "../mui/css/mui.min.css";
export default {
  name: "App"
};
</script>
<style>
#app {
  margin-top: 60px;
}
</style>
【Home组件代码：Home.vue】
<template>
  <div id="home">
    <h2>循环遍历数据</h2>
    <ol>
      <!-- 在 (4) 处补齐代码，使用内置指令遍历data上的数据 -->
      <li ( ) ="site in sites" :key="site.id">
        <h3>{{ site.name }}</h3>
        <span>
          <!-- 在 (5) 处补齐代码，绑定点击事件 -->
          <a ( ) ="goto(site.url)">{{ site.url }}</a>
        </span>
      </li>
    </ol>
  </div>
</template>
<script>
export default {
  name: "home",
  // 在 (6) 处补齐代码，内置的属性，存放自定义方法对象。
  ( ) : {
    goto: function(url) {
      window.open(url, "_blank");
    }
  },
  // 在 (7) 处补齐代码，内置的属性，存放数据。
  ( ) () {
    return {
      sites: [
        { id: "1", name: "百度", url: "http://www.baidu.com" },
        { id: "2", name: "谷歌", url: "http://www.google.com" },
        { id: "3", name: "淘宝", url: "http://www.taobao.com" }
      ]
    };
  }
};

```

```
};
</script>
<!-- 在（8）处补齐代码，规定样式的作用范围只是本文件。-->
<style（      ）>
#home {
padding: 20px;
}
li {
font-size: 20px;
list-style: none;
border: 1px solid #ccc;
margin: 10px auto;
padding: 4px;
}
li a {
font-size: 14px;
}
</style>
【User组件代码：User.vue】
<template>
<div id="user">
<h2>父子组件传值</h2>
<div class="line"></div>
<div>
<h3>在子组件的实例上，给子组件添加自定义属性。</h3>
<!--在（9）处补全代码，使用子组件，在（10）处补全代码，
通过事件获取子组件传递是数据-->
<（      ）:parentToChild="father.info"
@sendValueToParent = "（      ）"/>
</div>
</div>
</template>
<script>
import Child from "./Child";
export default {
name: "user",
methods: {
getValueFromChild: function(res) {
alert(res);
}
},
data() {
return {
father: { info: "父组件上的数据" }
};
},
components: { Child }
};
</script>
<style scoped>
.line {
border: 1px solid #ccc;
height: 0;
width: 100%;
margin: 20px auto;
}
</style>
【Child组件代码：Child.vue】
<template>
<div id="child">
<h4>子组件展示父组件上的数据</h4>
<!-- 在（11）处补全代码，把父组件传递的数据展示在子组件上
-->
<p>{{（      ）}}</p>
<button @click="clickEvent">通过事件，子组件向父组件传
值</button>
</div>
</template>
<script>
export default {
```

```
name: "child",
props: ["parentToChild"],
methods: {
  clickEvent: function() {
    // 在 (12) 处补全代码，通过事件向父组件传递数据
    this. (      ) ("sendValueToParent", this.value1);
  }
},
data() {
  return {
    value1: "这是子组件上的数据"
  };
}
};
</script>
<style scoped>
#child {
  border: 1px solid red;
  padding: 10px;
}
</style>
【List组件代码：List.vue】
<template>
  <div class="list">
    <h4 (      ) ="obj in info" :key='obj["album"]'
["albumID"]'>
      {{obj["album"]["name"]}}
    </h4>
  </div>
</template>
<script>
// 在 (14) (15) 处补全代码，导出默认对象
(      ) (      ) {
  name: "list",
  data() {
    return {
      info: null
    };
  },
  // 在 (16) 处补全代码，html加载完成后执行。
  (      ) () {
    this.axios
      .get("https://api.imjad.cn/qqfm/v1/?
type=album&page_size=10&id=4&page=1")
    // 在 (17) 处补全代码，成功回调函数
    . (      ) (response => (this.info =
response.data.data.albumInfoList))
    // 在 (18) 处补全代码，失败回调函数
    . (      ) (function(error) {
      // 请求失败处理
      console.log(error);
    });
  }
};
</script>
<style scoped>
</style>
【问题】请根据代码注释，回答问题，把答案填入对应位置。
```

- 1、
- 2、
- 3、
- 4、

- 5、
- 6、
- 7、
- 8、
- 9、
- 10、
- 11、
- 12、
- 13、
- 14、
- 15、
- 16、
- 17、
- 18、



4

25分

阅读下列说明，根据要求，完成以下题目。

【说明】

本项目为node框架express项目，项目名称为expressDemo,其中包括index.js项目入口文件，package.json配置文件，和mode\_modules目录，此目录存放项目依赖的各种模块。

【素材】

 素材.rar

【项目入口文件代码：index.js】

```
// 在（1）处补齐代码，导入node的express框架
var express=__（      ）__("express");
// 在（2）处补齐代码，创建应用
var app=__（      ）();
app.get("/user",function(req,res){
    res.send("user");
});
//通过正则的分组形式匹配参数
app.get(/^\/student\/(\d{3})$/ ,function(req,res){
    // 在（3）处补齐代码，获取url地址栏传入的第一个参数
    res.send("student"+__（      ）__);
});
// 通过key-value形式获取传递过来的参数
```

```
app.get("/teacher/:id",function(req,res){
    //在（4）处补齐代码，获取url地址栏传入的id值
    res.send("teacher的id是"+__);
})
// 匹配根目录
app.get("/",function(req,res){
    res.send("hello");
});
// 在（5）处补全代码，监听3000端口
app.__(3000);
```

【问题】根据代码提示填空

- 1、
- 2、
- 3、
- 4、
- 5、

