

职业教育联盟系列课程



职业教育联盟

OCCUPATION EDUCATION ALLIANCE

成就百万精英
引领行业发展

常用前端框架 及工具

刘静文 王朝梅 主编

OCCUPATION EDUCATION ALLIANCE COURSE



重庆大学
电子音像出版社
<http://www.cqup.com.cn>

常用前端框架及工具

职业教育联盟系列课程

主 编 刘静文 王朝梅

参 编 刘 彬

重庆大学电子音像出版社

内容提要

本书属于职业教育联盟系列课程使用教材，主要讲解了 ECMAScript 9 简介、let 和 const 命令、ECMAScript 9 基础、ECMAScript 9 语法和类、ECMAScript 9 对象、函数、ECMAScript9 ArrayBuffer、Vue 安装、实例与模板语法、Vue 计算属性和侦听器、Class 与 Style 绑定、Vue 渲染，事件处理、Vue 表单输入绑定，Vue 组件、Vue 过渡动画、可复用性组合、Vue 路由、Vue 脚手架、状态管理、Angular、WebPack 的使用、SVN 介绍与安装使用和 Git 介绍与安装使用。通过本书的学习，学生能掌握 ECMAScript 及 Vue 等常用前端框架及工具的实际应用。

版权信息

书名：常用前端框架及工具

作者：刘静文 王朝梅

重庆大学电子音像出版社

责任编辑：石媛媛

地址：重庆市沙坪坝区大学城西路 21 号

ISBN 978-7-89446-417-0

出版时间：2021 年 5 月

邮编：401331

电话：023-88617080

字数：324 千字

定价：89.00 元

版权所有，侵权必究

前言

欢迎学习职业教育联盟系列课程（OEAC）（新形态教材），职业教育联盟由全国几十所优秀的职业教育学校组成，课程研发团队由教学经验丰富的一线老师、企业中工作经验丰富的行业专家及教育专家组成。我们的纲领是：培养 500 万 IT 行业精英；我们的使命是：引领职业教育发展。

本系列教材作为新形态教材，广泛利用多媒体等新兴技术用于辅助教学，让学生寓教于乐，提高学习的兴趣和效率；同时，充分考虑了读者的阅读习惯和学习习惯，在编排上做了非常科学的安排：

- 本系列教材为作者团队花费了大量的人力、物力和财力倾力打造的新形态教材，全系采用“二维码链接配套资源”的新形态教材模式，每本教材都拥有视频教学资源、评估试题等配套教学资源，可以通过嵌入到教材每章节中的二维码轻松查看配套资源，让学习变得高效、有趣又轻松。
- 整本书包含学习目标、课程内容、总结、作业等，这个编排结构可以让读者更加轻松、高效地学习。一来可以提高理论的应用能力，二来可以巩固所学的理论知识，锻炼读者自己解决问题和举一反三的能力。
- 课程内容中含有实战案例，让读者在提高应用能力的同时获得实战经验，真正体现了学以致用指导方针。
- 采用图文结合的编排方式，宽松的版式让读者可以轻松阅读。

本系列课程由大量的老师及专家给予支持和帮助，由于参与本系列课程研发的人数太多，在这里没有一一列出他们的名字，在此由衷地感谢他们！本课程中使用的图例和片段仅用于教学示范和讲解，不做其他商业用途。在编写过程中，有一些图例和片段无法确定作者与出处，在此也向他们深表感谢，并请原作者与出版社或主编本人联系。同时希望读者和同行人士多提宝贵意见和建议。

本系列课程适合教学使用，也适合自学使用。

编者

2021 年 1 月 15 日



评估试题参考答案



案例资源

目录

第 1 章 ECMAScript 9 简介、let 和 const 命令	1
1.1 ECMAScript 简介	3
1.2 let 命令	4
1.2.1 let 基本用法	4
1.2.2 块级作用域	5
1.3 const 命令	6
1.3.1 基础语法	6
1.3.2 ES9 声明变量的六种方法	8
第 2 章 ECMAScript 9 基础	10
2.1 变量的解构赋值	12
2.2 字符串的扩展	12
2.2.1 字符串的遍历器接口	12
2.2.2 includes()、startsWith()、endsWith()	13
2.2.3 repeat()、padStart()、padEnd()	14
2.3 数值的扩展	14
2.3.1 Number.isFinite()、Number.isNaN()	14
2.3.2 Number.parseInt()、Number.parseFloat()	15
2.3.3 Number.isInteger()	16
2.4 Math 对象的扩展	16
2.4.1 Math.trunc()	16
2.4.2 Math.sign()	17
2.4.3 Math.imul()	17
2.5 函数的扩展	17
2.5.1 与解构赋值默认值结合使用	17
2.5.2 函数的 length 属性	18

2.5.3	rest 参数	19
2.5.4	严格模式	20
2.6	数组的扩展	21
2.6.1	扩展运算符	21
2.6.2	替代函数的 apply()方法	22
2.6.3	扩展运算符的应用	23
2.6.4	Array.from()与 Array.of()	24
第 3 章	ECMAScript 9 语法和类	27
3.1	class、extends、super	29
3.2	箭头函数	30
3.3	模板字符串	32
第 4 章	ECMAScript 9 对象、函数	34
4.1	对象的属性和方法	36
4.1.1	Object.is()	36
4.1.2	Object.assign()	37
4.1.3	Object.keys(), Object.values(), Object.entries()	37
4.2	函数	38
第 5 章	ECMAScript 9 的 ArrayBuffer 对象	47
5.1	ArrayBuffer 对象	49
5.1.1	概述	49
5.1.2	TypedArray 视图	51
5.1.3	概述	51
5.1.4	构造函数	52
5.2	DataView 视图	54
第 6 章	Vue 安装、实例与模板语法	58

6.1	安装介绍.....	60
6.2	实例	61
6.2.1	Vue.js 初步了解.....	61
6.2.2	实例生命周期钩子	62
6.3	Vue.js 模板语法	63
 第 7 章 Vue 计算属性和侦听器、Class 与 Style 绑定		67
7.1	计算属性.....	68
7.1.1	计算属性基础	68
7.1.2	计算属性缓存及方法	69
7.2	侦听器	70
7.3	Class 与 Style 绑定.....	71
7.3.1	class 属性绑定	71
7.3.2	数组语法.....	72
7.3.3	Style（内联样式）	73
 第 8 章 Vue 渲染及事件处理		76
8.1	条件渲染.....	77
8.1.1	v-if.....	77
8.1.2	v-else	77
8.1.3	v-else-if.....	77
8.1.4	v-show	78
8.2	列表渲染（v-for）	79
8.3	事件处理.....	81
8.3.1	监听事件	81
8.3.2	事件处理方法	81
8.3.3	事件修饰符.....	82
8.3.4	按键修饰符.....	83

第 9 章 Vue 表单输入绑定及 Vue 组件	87
9.1 表单输入绑定.....	88
9.1.1 表单.....	88
9.1.2 修饰符.....	90
9.2 组件	90
9.2.1 全局组件.....	91
9.2.2 局部组件.....	91
9.2.3 父组件向子组件传值 (Prop)	92
9.2.4 动态 Prop	93
9.2.5 子组件向父组件传值.....	93
第 10 章 Vue 过渡动画	97
10.1 过渡	98
10.1.1 单元素/组件的过渡.....	98
10.1.2 过渡的类名.....	99
10.1.3 多个元素的过渡.....	100
10.1.4 多个组件的过渡.....	100
10.2 CSS 动画	101
10.3 可复用的过渡.....	102
第 11 章 可复用性组合	104
11.1 混入	105
11.1.1 什么是混入.....	105
11.1.2 选项合并.....	105
11.2 自定义指令.....	107
11.3 过滤器	108
第 12 章 Vue 路由	111

12.1	Vue-router	112
12.1.1	版本对应	112
12.1.2	vue-router 的安装	112
12.1.3	vue-router 的使用	112
12.2	使用 vue-router 的综合实例	114
12.3	路由参数获取	117
12.3.1	JavaScript 控制路由跳转	118
12.3.2	嵌套路由	118
 第 13 章 Vue 脚手架及状态管理		122
13.1	Vue-cli 使用	123
13.1.1	环境搭建	123
13.1.2	vue-cli 构建项目	123
13.2	vuex	126
13.2.1	modules	126
13.2.2	mutations	128
13.2.3	actions	129
13.2.4	getters	130
13.2.5	mapState、mapGetters、mapActions	131
13.3	axios	132
13.3.1	GET 请求	132
13.3.2	POST 请求	132
13.3.3	一次性并发多个请求	133
13.3.4	Vue-cli proxyTable 解决开发环境的跨域问题	134
 第 14 章 Angular		136
14.1	Angular 简介与环境搭建	137
14.1.1	Angular 是什么	137
14.1.2	Angular 有什么特性	137
14.1.3	Angular 提供了哪些功能	137

14.1.4	Angular 与 AngularJS 有什么区别	137
14.1.5	环境搭建	138
14.2	插值表达式	139
14.2.1	绑定普通文本	139
14.2.2	绑定对象属性	139
14.3	自定义组件	140
14.3.1	定义组件的元信息	140
14.3.2	定义组件类	140
14.3.3	接口使用示例	140
14.3.4	自定义组件示例	141
14.4	常用指令简介	142
14.4.1	ngIf 指令简介	142
14.4.2	ngFor 指令简介	142
14.4.3	ngIf 与 ngFor 指令使用示例	142
14.5	事件绑定	143
14.5.1	事件绑定语法	144
14.5.2	事件绑定示例	144
14.6	Http 模块简介	144
14.6.1	使用 Http 服务步骤	144
14.6.2	Http 服务使用示例	145
14.7	路由模块简介	146
14.7.1	导入路由模块	146
14.7.2	配置路由信息	147
14.7.3	routerLink 指令	147
14.7.4	router-outlet 指令	147
14.7.5	路由使用示例	148

第 15 章 Webpack..... 150

15.1	Webpack 简介	151
15.1.1	为什么要使用 Webpack	151
15.1.2	什么是 Webpack	151

15.1.3	Webpack 和 Grunt 以及 Gulp 相比有什么特性	151
15.2	使用 WebPack.....	152
15.2.1	安装.....	152
15.2.2	正式使用 Webpack 前的准备	152
15.2.3	正式使用 Webpack	154
15.3	通过配置文件来使用 Webpack.....	155
15.3.1	更快捷的执行打包任务	156
15.3.2	使用 Webpack 构建本地服务器	157
15.3.3	Babel.....	158
15.4	模块	160
第 16 章	SVN 介绍与安装使用.....	164
16.1	SVN 简介.....	165
16.1.1	什么是 SVN.....	165
16.1.2	SVN 工作原理	165
16.1.3	SVN 能做什么	165
16.1.4	SVN 安装.....	166
16.2	服务端安装（windows）	166
16.3	客户端安装.....	170
16.4	创建资源库和用户.....	172
16.4.1	创建用户	172
16.4.2	创建资源库.....	173
16.5	SVN 操作说明.....	177
第 17 章	Git 介绍与安装使用	180
17.1	Git 介绍与安装	181
17.1.1	Git 简介.....	181
17.1.2	Git 客户端下载.....	182
17.1.3	Git 客户端安装过程.....	182
17.2	Git 操作	188

17.2.1	创建版本库.....	188
17.2.2	版本回退.....	191
17.2.3	工作区与暂存区的区别.....	192
17.3	Git 撤销修改和删除文件.....	193
17.3.1	撤销修改.....	193
17.3.2	删除文件.....	195
17.4	远程仓库.....	197
17.4.1	如何添加远程库.....	198
17.4.2	如何从远程库克隆.....	201

第1章 ECMAScript 9 简介、let 和 const 命令



视频教学资源

⊕ 学习目标

了解 ECMAScript6

掌握 let

掌握 const 命令

⊖ 本章单词

请在预习前完成下列单词，并将单词写在横线上。

① let ([let] 允许, 让): _____

② const(['konst] 常数, 不变的): _____

⊕ 预习任务

了解 ECMAScript 历史并了解 ECMAScript6, 提前了解 let 与 const 的作用。

```
{  
  let a = 10;  
  var b = 1;  
}  
const MAX = 5;
```

1.1 ECMAScript 简介

在介绍 ECMAScript9 之前，先了解一下 ECMAScript6。

ECMAScript6 是 JavaScript 语言的下一代标准，于 2015 年 6 月批准通过。ECMAScript6 可以用 JavaScript 语言来编写复杂的大型应用程序，成为企业级开发语言。让代码更加准确，更易于阅读。

ECMAScript 是 JavaScript 语言的国际标准，JavaScript 是 ECMAScript 的实现（ES 是规范，JS 是实现）。在日常场合，这两个词是可以互换的。

1. ECMAScript6 的历史

1998 年 6 月，ECMAScript 2.0 版发布。

1999 年 12 月，ECMAScript 3.0 版发布，成为 JavaScript 的通行标准，得到了广泛支持。

2007 年 10 月，ECMAScript 4.0 版草案发布，对 3.0 版做了大幅升级，预计次年 8 月发布正式版本。草案发布后，由于 4.0 版的目标过于激进，各方对于是否通过这个标准，发生了严重分歧。以 Yahoo、Microsoft、Google 为首的大公司，反对 JavaScript 的大幅升级，主张小幅改动；以 JavaScript 创造者 Brendan Eich 为首的 Mozilla 公司，则坚持当前的草案。

2008 年 7 月，由于对于下一个版本应该包括哪些功能，各方分歧太大，争论过于激烈，ECMA 开会决定，中止 ECMAScript 4.0 的开发，将其中涉及现有功能改善的一小部分，发布为 ECMAScript 3.1，而将其他激进的设想扩大范围，放入以后的版本，由于会议的气氛，该版本的项目代号起名为 Harmony（和谐）。会后不久，ECMAScript 3.1 就改名为 ECMAScript 5。

2009 年 12 月，ECMAScript 5.0 版正式发布。Harmony 项目则一分为二，一些较为可行的设想定名为 JavaScript.next 继续开发，后来演变成 ECMAScript 6；一些不是很成熟的设想，则被视为 JavaScript.next.next，在更远的将来再考虑推出。

2011 年 6 月，ECMAScript 5.1 版发布，并且成为 ISO 国际标准（ISO/IEC 16262:2011）。

2013 年 3 月，ECMAScript 6 草案冻结，不再添加新功能。新的功能设想将被放到 ECMAScript 7。

2013 年 12 月，ECMAScript 6 草案发布。然后是 12 个月的讨论期，听取各方反馈。

2015 年 6 月 17 日，ECMAScript 6 发布正式版本，即 ECMAScript 2015。

ECMA 的第 39 号技术专家委员会（Technical Committee 39，简称 TC39）负责制订 ECMAScript 标准，成员包括 Microsoft、Mozilla、Google 等大公司。TC39 的总体考虑是，ES5 与 ES3 基本保持兼容，较大的语法修正和新功能加入，将由 JavaScript.next 完成。

2. ECMAScript6 和 ECMAScript2015 的关系

ECMAScript 2015（简称 ES2015）这个词，也是经常可以看到的。它与 ES9 有什么关系呢？

2011 年，ECMAScript 5.1 版发布后，就开始制定 6.0 版了。因此，ES9 这个词的原意，就是指 JavaScript 语言的下一个版本。

但是，因为这个版本引入的语法功能太多，而且制定过程当中，还有很多组织和个人不断提交新功能。事情很快就变得清楚了，不可能在一个版本里面包括所有将要引入的功能。常规的做法是先发布 6.0 版，过一段时间再发 6.1 版，然后是 6.2 版、6.3 版等等。

但是，标准的制定者不想这样做。他们想让标准的升级成为常规流程：任何人在任何时候，都可以向标准委员会提交新语法的提案，然后标准委员会每个月开一次会，评估这些提案是否可以接受，需要哪些改进。如果经过多次会议以后，一个提案足够成熟了，就可以正式进入标准了。这就是说，标准的版本升级变成了一个不断滚动的流程，每个月都会有变动。

标准委员会最终决定，标准在每年的 6 月份正式发布一次，作为当年的正式版本。接下来的时间，就在这个版本的基础上做改动，直到下一年的 6 月份，草案就自然变成了新一年的版本。这样一来，就不需要以前的版本号了，只要用年份标记就可以了。

ES9 的第一个版本，就这样在 2015 年 6 月发布了，正式名称就是《ECMAScript 2015 标准》（简称 ES2015）。2016 年 6 月，小幅修订的《ECMAScript 2016 标准》（简称 ES2016）如期发布，这个版本可以看作是 ES9.1 版，因为两者的差异非常小（只新增了数组实例的 `includes` 方法和指数运算符），基本上是同一个标准。根据计划，2017 年 6 月发布 ES2017 标准。

因此，ES6 既是一个历史名词，也是一个泛指，含义是 5.1 版以后的 JavaScript 的下一代标准，涵盖了 ES2015、ES2016、ES2017 等等，而 ES2015 则是正式名称，特指该年发布的正式版本的语言标准。本书中提到 ES6 的地方，一般是指 ES2015 标准，但有时也是泛指“下一代 JavaScript 语言”。

ES9 是 ECMAScript2018 版本，即是 2018 年发布的新的标准。

1.2 let 命令

1.2.1 let 基本用法

ES9 新增了 `let` 命令，用来声明变量。它的用法类似于 `var`，但是所声明的变量，只在 `let` 命令所在的代码块内有效。

```
{  
  let name = '小明' ;  
}
```