

实验 10：基于 Vue.js 进行前端静态开发

一、实验目的

- 1、掌握 Vue.js 组件化开发方法，能够构建复杂的单页面应用；
- 2、理解 Vue Router 路由管理，实现多页面导航效果；
- 3、能够独立设计并实现博客界面，具备完整的前端项目开发能力。

二、实验学时

2 学时

三、实验类型

综合性

四、实验需求

1、硬件

每人配备计算机 1 台，建议优先使用个人计算机开展实验。

2、软件

安装 Visual Studio Code，以及 Edge 浏览器。

3、网络

本地主机能够访问互联网和实验中心网络。

4、工具

无。

五、实验任务

- 1、搭建博客项目基础结构，配置 Vue Router 路由

- 2、实现博客首页布局 and 文章列表组件
- 3、创建博客文章详情页 and 评论功能

六、实验内容及步骤

1、搭建博客项目基础结构，配置 Vue Router 路由

步骤 1：打开终端，使用 Vue CLI 创建新项目。

XML/HTML

```
1 vue create vue-blog
2 cd vue-blog
```

步骤 2：项目结构规划

创建以下目录结构。

XML/HTML

```
1 src/
2 |— components/           # 可复用组件
3 |   |— BlogHeader.vue
4 |   |— ArticleCard.vue
5 |   └─ CommentForm.vue
6 |— views/                # 页面组件
7 |   |— Home.vue
8 |   |— Article.vue
9 |   └─ About.vue
10 |— router/              # 路由配置
11 |   └─ index.js
12 └─ assets/              # 静态资源
```

步骤 3：配置路由

修改 src/router/index.js。

JavaScript

```
1 import { createRouter, createWebHistory } from 'vue-router'
2 import Home from '../views/Home.vue'
3
4 const routes = [
5   {
6     path: '/',
7     name: 'Home',
8     component: Home
9   },
10  {
11    path: '/article/:id',
12    name: 'Article',
13    component: () => import('../views/Article.vue')
14  },
15  {
16    path: '/about',
17    name: 'About',
18    component: () => import('../views/About.vue')
19  }
20 ]
21
22 const router = createRouter({
23   history: createWebHistory(),
24   routes
25 })
26
27 export default router
```

步骤 4：创建基础布局

修改 src/App.vue。

```
1 <template>
2   <div id="app">
3     <BlogHeader />
4     <main class="main-content">
5       <router-view />
6     </main>
7   </div>
8 </template>
9
10 <script>
11 import BlogHeader from './components/BlogHeader.vue'
12
13 export default {
14   name: 'App',
15   components: {
16     BlogHeader
17   }
18 }
19 </script>
20
21 <style>
22 * {
23   margin: 0;
24   padding: 0;
25   box-sizing: border-box;
26 }
27
28 body {
29   font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
30   line-height: 1.6;
31   color: #333;
32   background: #f8f9fa;
33 }
34
35 .main-content {
36   min-height: calc(100vh - 80px);
37   padding-top: 80px;
38 }
39 </style>
```

2、实现博客首页布局和文章列表组件

步骤 1：创建导航栏组件

创建 `src/components/BlogHeader.vue`。

```
1 <template>
2   <header class="blog-header">
3     <nav class="nav-container">
4       <div class="nav-brand">
5         <router-link to="/" class="brand-link">
6           <span class="brand-text">我的技术博客</span>
7         </router-link>
8       </div>
9
10      <div class="nav-links">
11        <router-link to="/" class="nav-link">首页</router-link>
12        <router-link to="/about" class="nav-link">关于</router-link>
13      </div>
14    </nav>
15  </header>
16 </template>
17
18 <script>
19 export default {
20   name: 'BlogHeader'
21 }
22 </script>
23
24 <style scoped>
25 .blog-header {
26   position: fixed;
27   top: 0;
28   left: 0;
29   right: 0;
30   background: rgba(255, 255, 255, 0.95);
31   backdrop-filter: blur(10px);
32   z-index: 1000;
33   box-shadow: 0 2px 20px rgba(0, 0, 0, 0.1);
34 }
35
36 .nav-container {
37   max-width: 1200px;
38   margin: 0 auto;
39   display: flex;
40   align-items: center;
41   justify-content: space-between;
42   padding: 0 20px;
```

```
43   height: 80px;
44 }
45
46 .brand-link {
47   text-decoration: none;
48   color: #333;
49   font-weight: bold;
50   font-size: 1.5rem;
51 }
52
53 .nav-links {
54   display: flex;
55   gap: 30px;
56 }
57
58 .nav-link {
59   text-decoration: none;
60   color: #333;
61   font-weight: 500;
62   padding: 10px 0;
63   position: relative;
64   transition: color 0.3s ease;
65 }
66
67 .nav-link:hover,
68 .nav-link.router-link-active {
69   color: #667eea;
70 }
71
72 .nav-link.router-link-active::after {
73   content: '';
74   position: absolute;
75   bottom: 0;
76   left: 0;
77   right: 0;
78   height: 2px;
79   background: #667eea;
80 }
81 </style>
```

步骤 2: 创建首页组件

创建 src/views/Home.vue。

```
1 <template>
2   <div class="home">
3     <!-- 英雄区域 -->
4     <section class="hero-section">
5       <div class="hero-content">
6         <h1 class="hero-title">探索技术的无限可能</h1>
7         <p class="hero-subtitle">分享前端开发、编程技巧和技术思考</p>
8       </div>
9     </section>
10
11    <!-- 文章列表 -->
12    <section class="articles-section">
13      <div class="container">
14        <h2 class="section-title">最新文章</h2>
15        <div class="articles-grid">
16          <ArticleCard
17            v-for="article in articles"
18            :key="article.id"
19            :article="article"
20            @click="goToArticle(article.id)"
21          />
22        </div>
23      </div>
24    </section>
25  </div>
26 </template>
27
28 <script>
29 import ArticleCard from '@components/ArticleCard.vue'
30
31 export default {
32   name: 'Home',
33   components: {
34     ArticleCard
35   },
36   data() {
37     return {
38       articles: [
39         {
40           id: 1,
41           title: 'Vue.js 3.0 组合式API详解',
42           excerpt: '深入探讨Vue 3的组合式API，学习如何更好地组织组件逻辑...' ,
```

```
43         cover: 'https://picsum.photos/400/250?random=1',
44         category: '前端开发',
45         date: '2024-01-15',
46         readTime: '8分钟',
47         tags: ['Vue.js', 'JavaScript', '前端框架']
48     },
49     {
50         id: 2,
51         title: 'CSS Grid布局完全指南',
52         excerpt: '掌握CSS Grid布局的核心概念和实战技巧, 创建响应式网页...',
53         cover: 'https://picsum.photos/400/250?random=2',
54         category: 'CSS',
55         date: '2024-01-12',
56         readTime: '6分钟',
57         tags: ['CSS', '布局', '响应式设计']
58     },
59     {
60         id: 3,
61         title: 'JavaScript异步编程实践',
62         excerpt: '从回调到async/await, 全面解析JavaScript异步编程的最佳实
        践...',
63         cover: 'https://picsum.photos/400/250?random=3',
64         category: 'JavaScript',
65         date: '2024-01-10',
66         readTime: '10分钟',
67         tags: ['JavaScript', '异步编程', 'Promise']
68     }
69 ]
70 }
71 },
72 methods: {
73     goToArticle(id) {
74         this.$router.push(`/article/${id}`)
75     }
76 }
77 }
78 </script>
79
80 <style scoped>
81 .home {
82     padding: 0;
83 }
84
85 .hero-section {
```

```
86   background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
87   color: white;
88   padding: 100px 0;
89   text-align: center;
90 }
91
92 .hero-content {
93   max-width: 1200px;
94   margin: 0 auto;
95   padding: 0 20px;
96 }
97
98 .hero-title {
99   font-size: 3rem;
100  margin-bottom: 1rem;
101  animation: fadeInUp 1s ease;
102 }
103
104 .hero-subtitle {
105   font-size: 1.3rem;
106   opacity: 0.9;
107   animation: fadeInUp 1s ease 0.2s both;
108 }
109
110 .articles-section {
111   padding: 80px 0;
112 }
113
114 .container {
115   max-width: 1200px;
116   margin: 0 auto;
117   padding: 0 20px;
118 }
119
120 .section-title {
121   text-align: center;
122   font-size: 2.5rem;
123   margin-bottom: 3rem;
124   color: #333;
125 }
126
127 .articles-grid {
128   display: grid;
129   grid-template-columns: repeat(auto-fit, minmax(350px, 1fr));
```

```
130   gap: 30px;
131 }
132
133 @keyframes fadeInUp {
134   from {
135     opacity: 0;
136     transform: translateY(30px);
137   }
138   to {
139     opacity: 1;
140     transform: translateY(0);
141   }
142 }
143
144 @media (max-width: 768px) {
145   .hero-title {
146     font-size: 2.2rem;
147   }
148
149   .articles-grid {
150     grid-template-columns: 1fr;
151   }
152 }
153 </style>
```

步骤 3：创建文章卡片组件

创建 `src/components/ArticleCard.vue`。

```
1 <template>
2   <article class="article-card" @click="$emit('click')">
3     <div class="article-cover">
4       
5       <div class="article-category">{{ article.category }}</div>
6     </div>
7
8     <div class="article-content">
9       <h3 class="article-title">{{ article.title }}</h3>
10      <p class="article-excerpt">{{ article.excerpt }}</p>
11
12      <div class="article-meta">
13        <span class="article-date">{{ formatDate(article.date) }}</span
14      >
15        <span class="article-read-time">{{ article.readTime }}</span>
16      </div>
17
18      <div class="article-tags">
19        <span
20          v-for="tag in article.tags"
21          :key="tag"
22          class="tag"
23        >
24          {{ tag }}
25        </span>
26      </div>
27    </article>
28  </template>
29
30  <script>
31    export default {
32      name: 'ArticleCard',
33      props: {
34        article: {
35          type: Object,
36          required: true
37        }
38      },
39      methods: {
40        formatDate(dateString) {
41          const options = { year: 'numeric', month: 'long', day: 'numeric' }
```

```
    }
42     return new Date(dateString).toLocaleDateString('zh-CN', options)
43   }
44 }
45 }
46 </script>
47
48 <style scoped>
49 .article-card {
50   background: white;
51   border-radius: 10px;
52   overflow: hidden;
53   box-shadow: 0 5px 15px rgba(0,0,0,0.1);
54   cursor: pointer;
55   transition: all 0.3s ease;
56   height: 100%;
57   display: flex;
58   flex-direction: column;
59 }
60
61 .article-card:hover {
62   transform: translateY(-5px);
63   box-shadow: 0 15px 30px rgba(0,0,0,0.15);
64 }
65
66 .article-cover {
67   position: relative;
68   height: 200px;
69   overflow: hidden;
70 }
71
72 .article-cover img {
73   width: 100%;
74   height: 100%;
75   object-fit: cover;
76   transition: transform 0.3s ease;
77 }
78
79 .article-card:hover .article-cover img {
80   transform: scale(1.1);
81 }
82
83 .article-category {
84   position: absolute;
```

```
85   top: 15px;
86   left: 15px;
87   background: #ff6b6b;
88   color: white;
89   padding: 5px 10px;
90   border-radius: 20px;
91   font-size: 0.8rem;
92 }
93
94 .article-content {
95   padding: 20px;
96   flex-grow: 1;
97   display: flex;
98   flex-direction: column;
99 }
100
101 .article-title {
102   font-size: 1.3rem;
103   margin-bottom: 10px;
104   color: #333;
105   line-height: 1.4;
106 }
107
108 .article-excerpt {
109   color: #666;
110   margin-bottom: 15px;
111   flex-grow: 1;
112   line-height: 1.6;
113 }
114
115 .article-meta {
116   display: flex;
117   justify-content: space-between;
118   font-size: 0.9rem;
119   color: #999;
120   margin-bottom: 15px;
121 }
122
123 .article-tags {
124   display: flex;
125   flex-wrap: wrap;
126   gap: 5px;
127 }
128
```

```
129 .tag {
130   background: #e9ecef;
131   color: #495057;
132   padding: 3px 8px;
133   border-radius: 12px;
134   font-size: 0.8rem;
135 }
136
137 .article-card {
138   animation: slideInUp 0.6s ease;
139 }
140
141 @keyframes slideInUp {
142   from {
143     opacity: 0;
144     transform: translateY(30px);
145   }
146   to {
147     opacity: 1;
148     transform: translateY(0);
149   }
150 }
151 </style>
```

3、创建博客文章详情页和评论功能

步骤 1：创建文章详情页

创建 src/views/Article.vue。

```
1 <template>
2   <div class="article-page">
3     <div class="container">
4       <!-- 返回按钮 -->
5       <button class="back-button" @click="$router.go(-1)">
6         ← 返回文章列表
7       </button>
8
9       <!-- 文章内容 -->
10      <article class="article-content" v-if="article">
11        <header class="article-header">
12          <h1 class="article-title">{{ article.title }}</h1>
13          <div class="article-meta">
14            <span class="publish-date">{{ formatDate(article.date) }}</
15            span>
16            <span class="read-time">{{ article.readTime }} 阅读</span>
17            <span class="category">{{ article.category }}</span>
18          </div>
19          <div class="article-tags">
20            <span v-for="tag in article.tags" :key="tag" class="tag">
21              {{ tag }}
22            </span>
23          </div>
24        </header>
25        <div class="article-body">
26          <p>Vue.js 3.0 带来了许多令人兴奋的新特性，让我们一起来探索这些变化。</
27          p>
28          <h2>组合式 API</h2>
29          <p>组合式 API 是 Vue 3 最重要的特性之一，它提供了更好的逻辑复用和代码组
30          织方式。通过 setup 函数，我们可以更灵活地组织组件逻辑。</p>
31          <h2>性能优化</h2>
32          <p>Vue 3 在性能方面做了大量优化，包括更快的虚拟 DOM、更好的 Tree-shak
33          ing 支持等。</p>
34          <h2>更好的 TypeScript 支持</h2>
35          <p>Vue 3 从一开始就使用 TypeScript 重写，提供了更好的类型推断和支持。
36          </p>
37        </div>
```

```
38     <!-- 评论区域 -->
39     <section class="comments-section">
40         <h3>评论 ({{ comments.length }})</h3>
41         <CommentForm @submit="addComment" />
42         <div class="comments-list">
43             <div
44                 v-for="comment in comments"
45                 :key="comment.id"
46                 class="comment-item"
47             >
48                 <div class="comment-avatar">
49                     {{ comment.author.charAt(0) }}
50                 </div>
51                 <div class="comment-content">
52                     <div class="comment-header">
53                         <strong>{{ comment.author }}</strong>
54                         <span class="comment-date">{{ formatDate(comment.date) }}</span>
55                     </div>
56                     <p>{{ comment.content }}</p>
57                 </div>
58             </div>
59         </div>
60     </section>
61 </article>
62
63     <div v-else class="loading">
64         文章加载中...
65     </div>
66 </div>
67 </div>
68 </template>
69
70 <script>
71 import CommentForm from '@components/CommentForm.vue'
72
73 export default {
74     name: 'Article',
75     components: {
76         CommentForm
77     },
78     data() {
79         return {
80             article: null,
```

```
81     comments: []
82   }
83 },
84 computed: {
85   articleId() {
86     return this.$route.params.id
87   }
88 },
89 mounted() {
90   this.fetchArticle()
91   this.fetchComments()
92 },
93 methods: {
94   async fetchArticle() {
95     // 模拟API调用
96     setTimeout(() => {
97       const articlesData = {
98         1: {
99           id: 1,
100           title: 'Vue.js 3.0 组合式API详解',
101           date: '2024-01-15',
102           readTime: '8分钟',
103           category: '前端开发',
104           tags: ['Vue.js', 'JavaScript', '前端框架']
105         },
106         2: {
107           id: 2,
108           title: 'CSS Grid布局完全指南',
109           date: '2024-01-12',
110           readTime: '6分钟',
111           category: 'CSS',
112           tags: ['CSS', '布局', '响应式设计']
113         },
114         3: {
115           id: 3,
116           title: 'JavaScript异步编程实践',
117           date: '2024-01-10',
118           readTime: '10分钟',
119           category: 'JavaScript',
120           tags: ['JavaScript', '异步编程', 'Promise']
121         }
122       }
123       this.article = articlesData[this.articleId]
124     }, 500)
```

```
125     },
126     async fetchComments() {
127         // 模拟获取评论
128         this.comments = [
129             {
130                 id: 1,
131                 author: '技术爱好者',
132                 content: '这篇文章很有帮助, 对Vue 3的新特性讲解得很清楚!',
133                 date: '2024-01-16'
134             },
135             {
136                 id: 2,
137                 author: '前端开发者',
138                 content: '期待更多关于组合式API的实战案例。',
139                 date: '2024-01-15'
140             }
141         ]
142     },
143     addComment(commentData) {
144         const newComment = {
145             id: this.comments.length + 1,
146             ...commentData,
147             date: new Date().toISOString().split('T')[0]
148         }
149         this.comments.unshift(newComment)
150     },
151     formatDate(dateString) {
152         const options = { year: 'numeric', month: 'long', day: 'numeric'
153     }
154         return new Date(dateString).toLocaleDateString('zh-CN', options)
155     }
156 }
157 </script>
158
159 <style scoped>
160 .article-page {
161     padding: 40px 0;
162 }
163
164 .back-button {
165     background: none;
166     border: 2px solid #ddd;
167     padding: 10px 20px;
```

```
168   border-radius: 5px;
169   cursor: pointer;
170   margin-bottom: 30px;
171   transition: all 0.3s ease;
172   font-size: 1rem;
173 }
174
175 .back-button:hover {
176   border-color: #667eea;
177   color: #667eea;
178 }
179
180 .article-content {
181   max-width: 800px;
182   margin: 0 auto;
183 }
184
185 .article-header {
186   text-align: center;
187   margin-bottom: 50px;
188   padding-bottom: 30px;
189   border-bottom: 1px solid #eee;
190 }
191
192 .article-title {
193   font-size: 2.5rem;
194   margin-bottom: 20px;
195   line-height: 1.3;
196   color: #333;
197 }
198
199 .article-meta {
200   display: flex;
201   justify-content: center;
202   gap: 20px;
203   margin-bottom: 15px;
204   color: #666;
205   font-size: 0.9rem;
206   flex-wrap: wrap;
207 }
208
209 .article-tags {
210   display: flex;
211   justify-content: center;
```

```
212   gap: 10px;
213   flex-wrap: wrap;
214 }
215
216 .article-body {
217   line-height: 1.8;
218   font-size: 1.1rem;
219   color: #444;
220 }
221
222 .article-body h2 {
223   margin: 40px 0 20px;
224   color: #333;
225   font-size: 1.8rem;
226 }
227
228 .article-body p {
229   margin-bottom: 20px;
230   text-align: justify;
231 }
232
233 .comments-section {
234   margin-top: 60px;
235   padding-top: 40px;
236   border-top: 1px solid #eee;
237 }
238
239 .comments-section h3 {
240   margin-bottom: 30px;
241   font-size: 1.5rem;
242   color: #333;
243 }
244
245 .comments-list {
246   margin-top: 30px;
247 }
248
249 .comment-item {
250   display: flex;
251   gap: 15px;
252   padding: 20px 0;
253   border-bottom: 1px solid #f0f0f0;
254 }
255
```

```
256 .comment-avatar {
257   width: 40px;
258   height: 40px;
259   background: #667eea;
260   color: white;
261   border-radius: 50%;
262   display: flex;
263   align-items: center;
264   justify-content: center;
265   font-weight: bold;
266   flex-shrink: 0;
267 }
268
269 .comment-header {
270   display: flex;
271   justify-content: space-between;
272   margin-bottom: 5px;
273   align-items: center;
274 }
275
276 .comment-date {
277   color: #999;
278   font-size: 0.9rem;
279 }
280
281 .loading {
282   text-align: center;
283   padding: 100px 0;
284   font-size: 1.2rem;
285   color: #666;
286 }
287
288 .article-content {
289   animation: fadeIn 0.6s ease;
290 }
291
292 @keyframes fadeIn {
293   from {
294     opacity: 0;
295     transform: translateY(20px);
296   }
297   to {
298     opacity: 1;
299     transform: translateY(0);
```

```
300   }
301 }
302
303 @media (max-width: 768px) {
304   .article-title {
305     font-size: 2rem;
306   }
307
308   .article-meta {
309     flex-direction: column;
310     gap: 10px;
311   }
312
313   .comment-header {
314     flex-direction: column;
315     align-items: flex-start;
316     gap: 5px;
317   }
318 }
319 </style>
```

步骤 2：创建评论表单组件

创建 `src/components/CommentForm.vue`。

```
1 <template>
2   <form class="comment-form" @submit.prevent="submitComment">
3     <h4>发表评论</h4>
4     <div class="form-group">
5       <input
6         type="text"
7         v-model="form.author"
8         placeholder="您的姓名"
9         required
10        class="form-input"
11        maxlength="20"
12      >
13    </div>
14    <div class="form-group">
15      <textarea
16        v-model="form.content"
17        placeholder="请输入评论内容..."
18        required
19        rows="4"
20        class="form-textarea"
21        maxlength="500"
22      ></textarea>
23      <div class="char-count">{{ form.content.length }}/500</div>
24    </div>
25    <button type="submit" class="submit-btn" :disabled="!isFormValid">
26      发表评论
27    </button>
28
29    <div v-if="showSuccess" class="success-message">
30      评论发表成功!
31    </div>
32  </form>
33 </template>
34
35 <script>
36 export default {
37   name: 'CommentForm',
38   data() {
39     return {
40       form: {
41         author: '',
42         content: ''
```

```
43     },
44     showSuccess: false
45   }
46 },
47 computed: {
48   isValid() {
49     return this.form.author.trim() && this.form.content.trim()
50   }
51 },
52 methods: {
53   submitComment() {
54     if (this.isValid) {
55       this.$emit('submit', {
56         author: this.form.author.trim(),
57         content: this.form.content.trim()
58       })
59       this.form.author = ''
60       this.form.content = ''
61       this.showSuccess = true
62       setTimeout(() => {
63         this.showSuccess = false
64       }, 3000)
65     }
66   }
67 }
68 }
69 </script>
70
71 <style scoped>
72 .comment-form {
73   background: #f8f9fa;
74   padding: 25px;
75   border-radius: 10px;
76   margin-bottom: 30px;
77   position: relative;
78 }
79
80 .comment-form h4 {
81   margin-bottom: 20px;
82   color: #333;
83   font-size: 1.2rem;
84 }
85
86 .form-group {
```

```
87   margin-bottom: 20px;
88   position: relative;
89 }
90
91 .form-input, .form-textarea {
92   width: 100%;
93   padding: 12px 15px;
94   border: 2px solid #e9ecef;
95   border-radius: 5px;
96   font-size: 1rem;
97   transition: all 0.3s ease;
98   font-family: inherit;
99 }
100
101 .form-input:focus, .form-textarea:focus {
102   outline: none;
103   border-color: #667eea;
104   box-shadow: 0 0 0 3px rgba(102, 126, 234, 0.1);
105 }
106
107 .form-textarea {
108   resize: vertical;
109   min-height: 100px;
110 }
111
112 .char-count {
113   text-align: right;
114   font-size: 0.8rem;
115   color: #999;
116   margin-top: 5px;
117 }
118
119 .submit-btn {
120   background: #667eea;
121   color: white;
122   border: none;
123   padding: 12px 30px;
124   border-radius: 5px;
125   cursor: pointer;
126   font-size: 1rem;
127   transition: all 0.3s ease;
128   width: 100%;
129 }
130
```

```
131 .submit-btn:hover:not(:disabled) {
132   background: #5a6fd8;
133   transform: translateY(-1px);
134 }
135
136 .submit-btn:disabled {
137   background: #ccc;
138   cursor: not-allowed;
139   transform: none;
140 }
141
142 .success-message {
143   background: #d4edda;
144   color: #155724;
145   padding: 10px 15px;
146   border-radius: 5px;
147   margin-top: 15px;
148   text-align: center;
149   animation: slideIn 0.3s ease;
150 }
151
152 @keyframes slideIn {
153   from {
154     opacity: 0;
155     transform: translateY(-10px);
156   }
157   to {
158     opacity: 1;
159     transform: translateY(0);
160   }
161 }
162
163 .comment-form {
164   animation: slideInUp 0.5s ease;
165 }
166
167 @keyframes slideInUp {
168   from {
169     opacity: 0;
170     transform: translateY(20px);
171   }
172   to {
173     opacity: 1;
174     transform: translateY(0);
```

```
175   }  
176 }  
177 </style>
```

4、实验验证与运行

运行项目，并调试完善。

XML/HTML

```
1 npm run serve
```

七、实验考核

本实验考核采用【实验随堂查】方式开展。

每个实验完成后，在实验课上通过现场演示的方式向实验指导教师进行汇报，并完成现场问答交流。

每个实验考核满分 100 分，其中实验成果汇报 60 分，现场提问交流 40 分。

实验考核流程：

- （1）学生演示汇报实验内容的完成情况，实验指导老师现场打分。
- （2）指导老师结合实验内容进行提问，每位学生提问 2-3 个问题，根据回答的情况现场打分。
- （3）实验考核结束后，进行公布成绩。