

实验 05：使用 JavaScript 操作 DOM

一、实验目的

- 1、掌握 DOM 树结构模型，熟练使用 JavaScript 进行节点的查询、创建、插入、删除、修改操作；
- 2、理解 JavaScript 事件机制，能处理用户交互；
- 3、学会用 DOM 动态修改元素样式、属性及内容，实现页面交互效果；
- 4、通过具体任务实践，提升用 DOM 操作解决实际问题的能力，培养前端交互设计思维。

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、生成九九乘法表
- 2、实现模态弹出框
- 3、实现表单验证
- 4、实现自动轮播图

六、实验内容及步骤

1、生成九九乘法表

步骤 1：创建 HTML 结构

XML/HTML

```
1 <div id="multiplication-chart"></div>
```

步骤 2：实现 JavaScript 代码，示例如下。

JavaScript

```
1 const chartContainer = document.getElementById('multiplication-chart');
2 const table = document.createElement('table'); // 创建table元素
3
4 // 循环生成行 (i从1到9)
5 for (let i = 1; i <= 9; i++) {
6   const tr = document.createElement('tr'); // 创建行
7   // 循环生成列 (j从1到i)
8   for (let j = 1; j <= i; j++) {
9     const td = document.createElement('td'); // 创建单元格
10    td.textContent = `${j} × ${i} = ${i * j}`; // 设置内容
11    tr.appendChild(td); // 单元格添加到行
12  }
13  table.appendChild(tr); // 行添加到table
14 }
15
16 chartContainer.appendChild(table); // table添加到容器
```

2、实现模态弹出框

步骤 1：创建 HTML 结构

XML/HTML

```
1 <!-- 打开按钮 -->
2 <button id="openModal" class="btn">打开弹出框</button>
3 <!-- 遮罩层（半透明背景） -->
4 <div id="modalOverlay" class="overlay hidden"></div>
5 <!-- 弹出框容器 -->
6 <div id="modalContent" class="modal hidden">
7   <div class="modal-header">
8     <h3>提示</h3>
9     <button id="closeModal" class="close-btn">&times;</button>
10  </div>
11  <div class="modal-body">
12    <p>这是一个模态弹出框！ </p>
13  </div>
14 </div>
```

步骤 2：实现 CSS 样式

CSS

```
1 .overlay {
2   position: fixed;
3   top: 0; left: 0;
4   width: 100%; height: 100%;
5   background: rgba(0,0,0,0.5);
6   z-index: 998;
7 }
8 .modal {
9   position: fixed;
10  top: 50%; left: 50%;
11  transform: translate(-50%, -50%);
12  width: 400px;
13  padding: 20px;
14  background: white;
15  border-radius: 8px;
16  z-index: 999;
17  box-shadow: 0 2px 10px rgba(0,0,0,0.1);
18 }
19 .close-btn {
20   position: absolute;
21   top: 10px; right: 10px;
22   background: none;
23   border: none;
24   font-size: 20px;
25   cursor: pointer;
26 }
27 .hidden { display: none; }
```

步骤3：实现 JavaScript 逻辑

```
1 // 1. 获取DOM元素
2 const openModalBtn = document.getElementById('openModal');
3 const closeModalBtn = document.getElementById('closeModal');
4 const modalOverlay = document.getElementById('modalOverlay');
5 const modalContent = document.getElementById('modalContent');
6
7 // 2. 打开弹出框: 移除hidden类
8 openModalBtn.addEventListener('click', () => {
9   modalOverlay.classList.remove('hidden');
10  modalContent.classList.remove('hidden');
11 });
12
13 // 3. 关闭弹出框 (按钮+遮罩层)
14 const closeFunc = () => {
15   modalOverlay.classList.add('hidden');
16   modalContent.classList.add('hidden');
17 };
18 closeModalBtn.addEventListener('click', closeFunc);
19 modalOverlay.addEventListener('click', closeFunc); // 点击遮罩层也关闭
20
21 // 可选: ESC键关闭
22 document.addEventListener('keydown', (e) => {
23   if (e.key === 'Escape' && !modalContent.classList.contains('hidden')) {
24     closeFunc();
25   }
26 });
```

3、实现表单验证

步骤 1: 创建 HTML 结构

XML/HTML

```
1 <form id="registerForm">
2   <div class="form-group">
3     <label for="username">用户名: </label>
4     <input type="text" id="username" required>
5     <span id="usernameError" class="error-msg hidden">用户名≥3位</span>
6   </div>
7   <div class="form-group">
8     <label for="password">密码: </label>
9     <input type="password" id="password" required>
10    <span id="passwordError" class="error-msg hidden">密码需含字母+数字, ≥
      6位</span>
11  </div>
12  <div class="form-group">
13    <label for="email">邮箱: </label>
14    <input type="email" id="email" required>
15    <span id="emailError" class="error-msg hidden">邮箱格式错误</span>
16  </div>
17  <button type="submit" class="btn">注册</button>
18 </form>
```

步骤 2: 实现 CSS 样式

CSS

```
1 .form-group { margin-bottom: 15px; }
2 .error-msg {
3   color: red; font-size: 12px;
4   margin-left: 10px;
5 }
```

步骤 3: 实现 JavaScript 逻辑

```
1 const form = document.getElementById('registerForm');
2 const username = document.getElementById('username');
3 const password = document.getElementById('password');
4 const email = document.getElementById('email');
5 // 错误提示元素
6 const usernameError = document.getElementById('usernameError');
7 const passwordError = document.getElementById('passwordError');
8 const emailError = document.getElementById('emailError');
9
10 // 提交事件：阻止默认提交，执行验证
11 form.addEventListener('submit', (e) => {
12   e.preventDefault(); // 阻止表单默认提交
13   let isValid = true;
14
15   // 1. 验证用户名
16   if (username.value.length < 3) {
17     usernameError.classList.remove('hidden');
18     isValid = false;
19   } else {
20     usernameError.classList.add('hidden');
21   }
22
23   // 2. 验证密码（正则：含字母+数字，≥6位）
24   const pwdReg = /^(?=.*[A-Za-z])(?=.*\d)[A-Za-z\d]{6,}$/;
25   if (!pwdReg.test(password.value)) {
26     passwordError.classList.remove('hidden');
27     isValid = false;
28   } else {
29     passwordError.classList.add('hidden');
30   }
31
32   // 3. 验证邮箱（简单正则）
33   const emailReg = /^[a-zA-Z0-9_-]+@[a-zA-Z0-9_-]+(\.[a-zA-Z0-9_-]+)+$/;
34   if (!emailReg.test(email.value)) {
35     emailError.classList.remove('hidden');
36     isValid = false;
37   } else {
38     emailError.classList.add('hidden');
39   }
40
41   // 所有验证通过：提示成功
```

```
42  if (isValid) {  
43      alert('注册成功! ');  
44      form.reset(); // 清空表单  
45  }  
46  });
```

4、实现自动轮播图

步骤 1: 创建 HTML 结构

XML/HTML

```
1 <div class="slider">  
2   <div class="slides">  
3       
4       
5       
6   </div>  
7   <!-- 左右箭头 -->  
8   <button class="slider-btn prev">&lt;</button>  
9   <button class="slider-btn next">&gt;</button>  
10  <!-- 指示点 -->  
11  <div class="indicators">  
12    <span class="indicator active" data-index="0"></span>  
13    <span class="indicator" data-index="1"></span>  
14    <span class="indicator" data-index="2"></span>  
15  </div>  
16 </div>
```

步骤 2: 实现 CSS 样式

CSS

```
1 .slider {
2   position: relative;
3   width: 600px; height: 400px;
4   margin: 0 auto; overflow: hidden;
5 }
6 .slides {
7   display: flex;
8   width: 300%; /* 3张图占3倍宽度 */
9   height: 100%;
10  transition: transform 0.5s ease;
11 }
12 .slide {
13   width: 33.333%; /* 每张图占1/3 */
14   height: 100%; object-fit: cover;
15 }
16 .slider-btn {
17   position: absolute;
18   top: 50%; transform: translateY(-50%);
19   background: rgba(0,0,0,0.5); color: white;
20   border: none; padding: 10px; cursor: pointer;
21 }
22 .prev { left: 10px; } .next { right: 10px; }
23 .indicators {
24   position: absolute;
25   bottom: 20px; left: 50%;
26   transform: translateX(-50%);
27   display: flex; gap: 10px;
28 }
29 .indicator {
30   width: 12px; height: 12px;
31   border-radius: 50%; background: rgba(255,255,255,0.5);
32   cursor: pointer;
33 }
34 .indicator.active { background: white; }
```

步骤3：实现 JavaScript 逻辑

JavaScript

```
1 const slides = document.querySelector('.slides');
2 const prevBtn = document.querySelector('.prev');
3 const nextBtn = document.querySelector('.next');
4 const indicators = document.querySelectorAll('.indicator');
5 let currentIndex = 0;
6 const slideCount = document.querySelectorAll('.slide').length;
7
8 // 更新轮播图状态 (切换图片+指示点)
9 const updateSlider = () => {
10   slides.style.transform = `translateX(-${currentIndex * (100 / slideCount)}%)`;
11   // 更新指示点active类
12   indicators.forEach((indicator, index) => {
13     indicator.classList.toggle('active', index === currentIndex);
14   });
15 };
16
17 // 下一张
18 const nextSlide = () => {
19   currentIndex = (currentIndex + 1) % slideCount; // 循环到最后一张后回到第一张
20   updateSlider();
21 };
22
23 // 上一张
24 const prevSlide = () => {
25   currentIndex = (currentIndex - 1 + slideCount) % slideCount;
26   updateSlider();
27 };
28
29 // 绑定箭头事件
30 nextBtn.addEventListener('click', nextSlide);
31 prevBtn.addEventListener('click', prevSlide);
32
33 // 绑定指示点事件
34 indicators.forEach(indicator => {
35   indicator.addEventListener('click', () => {
36     currentIndex = parseInt(indicator.dataset.index);
37     updateSlider();
38   });
39 });
40
```

```
41 // 自动轮播：每3秒切换一次
42 let timer = setInterval(nextSlide, 3000);
43
44 // 鼠标悬停暂停，离开恢复
45 const slider = document.querySelector('.slider');
46 slider.addEventListener('mouseenter', () => clearInterval(timer));
47 slider.addEventListener('mouseleave', () => timer = setInterval(nextSlide, 3000));
```

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。