

实验 04：使用 CSS 实现网页动画

一、实验目的

- 1、掌握 CSS 动画的核心概念；
- 2、理解动画的时间函数、延迟、迭代次数等参数的作用；
- 3、能独立实现常见网页动画效果，提升视觉交互设计能力。

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、实现加载动画；
- 2、实现悬浮卡片效果；

- 4、实现 3D 立方体动画；
- 5、实现 3D 卡片翻转动画。

六、实验内容及步骤

1、加载动画

步骤 1：创建 HTML 结构

XML/HTML

```
1 <div class="loader"></div>
```

步骤 2：实现 CSS 样式。

CSS

```
1 /* 全局重置 */
2 * {
3     margin: 0;
4     padding: 0;
5     box-sizing: border-box;
6 }
7
8 body {
9     display: flex;
10    justify-content: center;
11    align-items: center;
12    min-height: 100vh;
13    background: #f0f2f5;
14 }
15
16 /* 加载动画容器 */
17 .loader {
18     width: 80px;
19     height: 80px;
20     border: 8px solid #e3e3e3; /* 底部浅色环 */
21     border-top: 8px solid #2b6cb0; /* 顶部蓝色环（形成缺口） */
22     border-radius: 50%; /* 圆形 */
23     /* 动画：旋转1.5秒，线性速度，无限循环 */
24     animation: spin 1.5s linear infinite;
25 }
26
27 /* 定义旋转关键帧 */
28 @keyframes spin {
29     from { transform: rotate(0deg); } /* 初始状态：0度 */
30     to { transform: rotate(360deg); } /* 结束状态：360度（一圈） */
31 }
```

2、悬浮卡片效果

步骤 1：创建 HTML 结构

XML/HTML

```
1 <div class="card-container">
2     <article class="card">
3         
5         <div class="card-content">
6             <h3>卡片标题</h3>
7             <p>这是一段卡片描述文字，hover时会有缩放效果。</p>
8         </div>
9     </article>
10 </div>
```

步骤 2：实现 CSS 样式

CSS

```
1 .card-container {
2     display: flex;
3     justify-content: center;
4     margin-top: 2rem;
5 }
6
7 .card {
8     width: 300px;
9     background: #fff;
10    border-radius: 10px;
11    overflow: hidden;
12    box-shadow: 0 4px 8px rgba(0,0,0,0.1); /* 初始阴影 */
13    /* 过渡效果: transform和box-shadow变化, 时长0.3秒, 缓动曲线 */
14    transition: transform 0.3s ease, box-shadow 0.3s ease;
15 }
16
17 .card img {
18     width: 100%;
19     height: 200px;
20     object-fit: cover;
21 }
22
23 .card-content {
24     padding: 1rem;
25 }
26
27 .card-content h3 {
28     margin-bottom: 0.5rem;
29     color: #2d3748;
30 }
31
32 .card-content p {
33     color: #666;
34     font-size: 0.9rem;
35 }
36
37 /* hover时触发动画 */
38 .card:hover {
39     transform: scale(1.05); /* 放大1.05倍 */
40     box-shadow: 0 8px 16px rgba(0,0,0,0.2); /* 阴影增大 */
41 }
```

3、3D 立方体动画

步骤 1：创建 HTML 结构

XML/HTML

```
1 <div class="cube-wrapper">
2   <div class="cube">
3     <div class="face front">前</div>
4     <div class="face back">后</div>
5     <div class="face left">左</div>
6     <div class="face right">右</div>
7     <div class="face top">上</div>
8     <div class="face bottom">下</div>
9   </div>
10 </div>
```

步骤 2：实现 CSS 样式

CSS

```
1 .cube-wrapper {
2     perspective: 800px; /* 透视距离：越小，3D效果越明显 */
3     margin-top: 3rem;
4 }
5
6 .cube {
7     width: 200px;
8     height: 200px;
9     position: relative;
10    transform-style: preserve-3d; /* 保留子元素的3D变换 */
11    /* 动画：绕Y轴和X轴旋转，时长4秒，无限循环 */
12    animation: rotateCube 4s linear infinite;
13 }
14
15 /* 定义立方体旋转关键帧 */
16 @keyframes rotateCube {
17     from { transform: rotateY(0deg) rotateX(0deg); }
18     to { transform: rotateY(360deg) rotateX(360deg); } /* 绕Y轴和X轴各转一
    圈 */
19 }
20
21 /* 立方体的面：通用样式 */
22 .face {
23     position: absolute;
24     width: 200px;
25     height: 200px;
26     background: rgba(43, 108, 176, 0.8); /* 半透明蓝色 */
27     border: 2px solid #fff;
28     display: flex;
29     justify-content: center;
30     align-items: center;
31     font-size: 2rem;
32     color: #fff;
33 }
34
35 /* 每个面的位置（3D变换） */
36 .front { transform: translateZ(100px); } /* 前面：沿Z轴前进100px */
37 .back { transform: rotateY(180deg) translateZ(100px); } /* 后面：绕Y轴转1
    80度，再前进100px */
38 .left { transform: rotateY(-90deg) translateZ(100px); } /* 左面：绕Y轴转-
    90度，前进100px */
39 .right { transform: rotateY(90deg) translateZ(100px); } /* 右面：绕Y轴转9
```

```
0度, 前进100px */
40 .top { transform: rotateX(90deg) translateZ(100px); } /* 上面: 绕X轴转90
    度, 前进100px */
41 .bottom { transform: rotateX(-90deg) translateZ(100px); } /* 下面: 绕X轴
    转-90度, 前进100px */
```

4、3D 卡片翻转

步骤 1: 创建 HTML 结构

XML/HTML

```
1 <div class="flip-card">
2   <div class="card">
3     <!-- 正面 -->
4     <div class="side front">
5       
6       <h3>卡片正面</h3>
7     </div>
8     <!-- 背面 -->
9     <div class="side back">
10      <h3>卡片背面</h3>
11      <p>这是卡片的背面内容, hover时翻转显示。</p>
12    </div>
13  </div>
14 </div>
```

步骤 2: 实现 CSS 样式

CSS

```
1 .flip-card {
2     perspective: 800px; /* 透视距离 */
3     margin-top: 2rem;
4 }
5
6 .card {
7     width: 300px;
8     height: 400px;
9     position: relative;
10    transform-style: preserve-3d; /* 保留3D变换 */
11    /* hover时翻转：绕Y轴转180度，时长0.8秒，缓动曲线 */
12    transition: transform 0.8s ease;
13 }
14
15 /* 正面和背面的通用样式 */
16 .side {
17     position: absolute;
18     width: 100%;
19     height: 100%;
20     backface-visibility: hidden; /* 隐藏背面（初始时背面朝后，不可见） */
21     border-radius: 10px;
22     padding: 1rem;
23     box-sizing: border-box;
24 }
25
26 .front {
27     background: #fff;
28     display: flex;
29     flex-direction: column;
30     align-items: center;
31 }
32
33 .front img {
34     width: 200px;
35     height: 200px;
36     object-fit: cover;
37     margin-bottom: 1rem;
38 }
39
40 .back {
41     background: #2b6cb0;
42     color: #fff;
```

```
43     transform: rotateY(180deg); /* 初始时背面绕Y轴转180度（朝后） */
44     display: flex;
45     flex-direction: column;
46     justify-content: center;
47 }
48
49 /* hover时卡片翻转：显示背面 */
50 .card:hover {
51     transform: rotateY(180deg);
52 }
```

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。