

华中科技大学

课程实验报告

课程名称： 新生实践课

专业班级 CS2409

学 号 U202490042

姓 名 马俊豪

指导教师 李榕

报告日期 2024 年 11 月 18 日

计算机科学与技术学院

目 录

1	网页整体框架	1
2	主页设计	2
3	分页面设计	8
3.1	页面 1 (自创小游戏)	8
3.2	页面 2 (我的相册)	9
3.3	页面 3 (我的动画作品)	11
3.4	页面 4 (学习心得)	12
4	网页设计小结	13
5	课程的收获和建议	14
5.1	计算机基础知识	14
5.2	文档撰写工具 LaTeX.....	14
5.3	编程工具 Python.....	15
5.4	图像设计软件 Photoshop.....	15
5.5	版本管理软件 Git	15
5.6	网页制作 Dreamweaver.....	15

1 网页整体框架

图1-1是我网站的主要框架，它是由我的主页面为中心，导航到不同的网站，页面简洁。我的个人主页采用了模块化布局，将主要功能分为多个区域，并以垂直对齐的形式展示在页面中。各模块独立但又相互关联，用户可以通过点击不同模块进入对应的页面，满足日常浏览与操作的需求。

主页中心位置是导航的核心，其他功能模块围绕其布局，构建了一个清晰的导航体系。通过简洁的设计风格，增强用户体验，便于访问不同内容。使用的编程语言有如下：

- 1) JavaScript
- 2) HTML

功能模块方面，我的主页是整个网站的核心，起到导航主入口的作用。用户无论在哪个页面，都可以通过返回主页快速定位到其他模块。而其它网站是我的分页面，它涵盖了我的生活、娱乐、以及学习等方面的信息。我认为它作为一个自我介绍网站，它发挥出了所有它该有的功能。



图 1-1 网页整体框架展示

2 主页设计

请见图2-1为了方便我的内容创作，我从 github 的开源项目中上找到了这个网站模板，并对其加以了删改。为了能够加深浏览者对我的第一印象，我采用了开门见山的形式，讲述了我的性格形象。同时页面设计以动态效果为主，增强了用户的体验感以及与网站的交互效果，也增强了页面活力。而且我的兴趣爱好也以技能条的形式展现出来，更加直白明了。

同时，我还大量运用了 github 中开源的仓库（如 Typed.js、MetingJS、APlayer）简化了复杂功能的实现，让我能够专注于内容设计。我认为我的网页比较突出的地方在于主页面的音乐播放器，我认为可以一定程度上，加长浏览者在此页面的停留时间。主页的设计语言上除了运用 html 外，还有 javascript 语言。这也是我第一次尝试写 javascript，因此我也参考了一些百度上的代码，让我收获颇丰，同时以一个更好的页面效果展现给浏览者。



图 2-1 主页展示

页面的算法及思路介绍

1. 页面结构设计

(1) HTML 结构:

用了语义化的 HTML5 标签如 `<main>` 和 `<article>`, 提升了代码的可读性和 SEO 效果。包括有:

- `nav`: 定义导航栏, 包含背景图和动态标题。
- `aside`: 侧边栏, 展示个人信息、格言、站点数据等模块。
- `main`:
 - 音乐播放器 (通过 MetingJS 库实现)。
 - 技能展示模块 (带颜色的进度条)。
 - 链接区域, 用于导航至相关内容。
 - 联系方式。
- `footer`: 页脚, 展示版权信息。

(2) 动态元素:

- **Typed.js**: 动态展示标题标语, 增强了用户体验。
- **雪花特效**: 利用自定义的 jQuery 插件生成雪花下落动画。
- **浏览器标题变化**: 通过监听 `visibilitychange` 事件, 修改页面崩溃和恢复的互动语言。

2. 效果设计

色彩:

- 用渐变色 (`linear-gradient`) 渲染技能条, 突出层次。
- 背景图通过链接 `ixiaowai.cn` 加载。

字体和图标:

- 引入 FontAwesome 提供精美的图标。
- 通过 CSS 动画展示。

3. 交互设计

用户体验提升:

- **音乐播放器:** 嵌入了第三方音乐列表。
- **雪花特效:** 为页面增添互动感。
- **鼠标点击效果和动态标题:** 增强互动感。

页面算法分析

1. 动态标题思路

JavaScript 实现代码:

```
1 var OriginTitle = document.title;
2 var titleTime;
3 document.addEventListener('visibilitychange', function () {
4     if (document.hidden) {
5         document.title = ' (°A°`´) 页面崩溃啦~';
6         clearTimeout(titleTime);
7     } else {
8         document.title = ' (> <*) 噫又好啦~' + OriginTitle;
9         titleTime = setTimeout(function () {
10             document.title = OriginTitle;
11         }, 2000);
12     }
13 });
```

Listing 1 动态标题切换算法

工作原理:

- 利用 `document.hidden` 判断页面是否可见。
- 页面不可见时, 改变标题为“页面崩溃啦”。
- 页面恢复时, 则显示“页面又好啦”, 然后又在两秒后恢复原始标题。

2. 雪花效果思路

JavaScript 实现代码:

```
1 $.fn.snow = function (options) {
2     var $flake = $('<div id="snowbox">').css({
3         'position': 'absolute',
4         'z-index': '9999',
5         'top': '-50px'
6     }).html('&#10052;'),
7     documentHeight = $(document).height(),
8     documentWidth = $(document).width(),
9     defaults = {
```

```
10     minSize: 10,
11     maxSize: 20,
12     newOn: 1000,
13     flakeColor: "#AFDAEF"
14 };
15 options = $.extend({}, defaults, options);
16 var interval = setInterval(function () {
17     var startPositionLeft = Math.random() * documentWidth - 100,
18         startOpacity = 0.5 + Math.random(),
19         sizeFlake = options.minSize + Math.random() * options.maxSize,
20         endPositionTop = documentHeight - 200,
21         endPositionLeft = startPositionLeft - 500 + Math.random() * 500,
22         durationFall = documentHeight * 10 + Math.random() * 5000;
23     $flake.clone().appendTo('body').css({
24         left: startPositionLeft,
25         opacity: startOpacity,
26         'font-size': sizeFlake,
27         color: options.flakeColor
28     }).animate({
29         top: endPositionTop,
30         left: endPositionLeft,
31         opacity: 0.2
32     }, durationFall, 'linear', function () {
33         $(this).remove();
34     });
35 }, options.newOn);
36 };
```

Listing 2 雪花动画效果

工作原理:

- 利用 `Math.random()` 生成雪花的随机起始位置、大小、透明度。
- 每隔 `options.newOn` 多少毫秒生成一片雪花，调用 `animate()` 动画从顶部移动到底部，结束时移除 DOM。
- 使用 `setInterval` 不断地生成雪花。

3. 技能条渲染思路

HTML 实现代码:


```
1 <div class="skillbar-title" style="background: linear-gradient(to right, #FF0066 0%, #FF00CC 100%); width: 75%">
2   <span>网球</span>
3 </div>
```

Listing 3 技能条效果

工作原理:

- **宽度控制:** 动态设置 `style="width: X%"`, 然后再由百分比数字决定技能熟练度。
- **渐变背景:** 使用 `linear-gradient` 填充对应百分比的颜色。

* 注: 7, 8, 9 页经过了 chatGPT 的润色以及在结构上的优化, 旨在方便读者阅览。*

3 分页面设计

我的分页面一共有五个，分别是自创小游戏、我的相册、我的动画作品以及学习心得。我设计这些分页面的思路是为了能够让浏览者在了解我的个性爱好的同时，通过清晰的页面结构、友好的交互设计和多媒体形式（图片、文字、动画、游戏等）的结合，让他们体验到个性化且吸引人的内容空间。

3.1 页面 1 (自创小游戏)

该页面（如图3-1）可以包含我自主设计和开发的小游戏展示区，核心是增强互动性和趣味性。我的设计意图是为了能够展现我的创作能力和编程技巧，并且提供娱乐性内容，吸引访问者互动。这也是我第一个自己做的小游戏，因此可能还有一些不完美的地方。大致的算法思路：

1. 初始化设置:

目标：设置画布、游戏元素以及初始状态。

逻辑：设置 canvas 的宽高为浏览器窗口大小，确保全屏展示。定义玩家飞船 (spaceship) 的初始位置、大小、速度等属性。使用数组存储陨石对象 (asteroids)，并动态增加。设置 score 和 difficulty 用于分数和难度调整。

2. 玩家飞船的移动算法:

目标：根据按键状态控制飞船在上下左右方向移动。

逻辑：通过监听 keydown 和 keyup 事件记录按键状态 (keys 数组)。在每帧更新中，根据 ArrowLeft、ArrowRight 等按键状态更新飞船的坐标。

6. 游戏循环算法:

目标：不断更新游戏状态并绘制画面，直到游戏结束。

逻辑：使用 requestAnimationFrame 实现动画更新

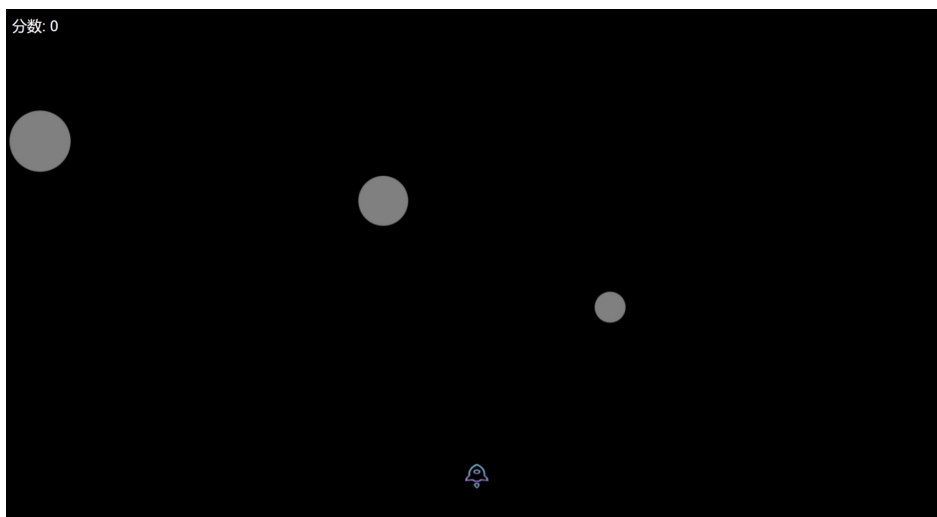


图 3-1 分页面 1 展示

3.2 页面 2 (我的相册)

我的相册 (如图3-2) 这个页面主要是展示了我在外面旅游的照片, 让浏览者加深对我的印象, 同时方便我分享个人拍摄或收藏的照片。这个页面的设计难度不是特别大, 运用的算法也不是特别多。

大致的算法思路:

1. 目标: 初始化 HTML 基本结构:

逻辑:

- header: 页面标题和描述
- gallery: 容纳若干图片卡片
- footer: 页面版权信息

2. 目标: 设置 CSS 样式:

逻辑:

- 定义网格布局, 适应屏幕宽度变化。
- 为每张图片卡片设置样式 (阴影、边框圆角)。
- 增加鼠标悬停的放大效果。

3. 目标: 图片布局逻辑:

逻辑:

- 通过 grid 自动排列图片。
- 根据屏幕大小调整列数，确保每列最小宽度为 400px。

4. 目标：响应式支持：

逻辑：

- 使用 viewport 设置缩放。
- 确保图片比例自适应。

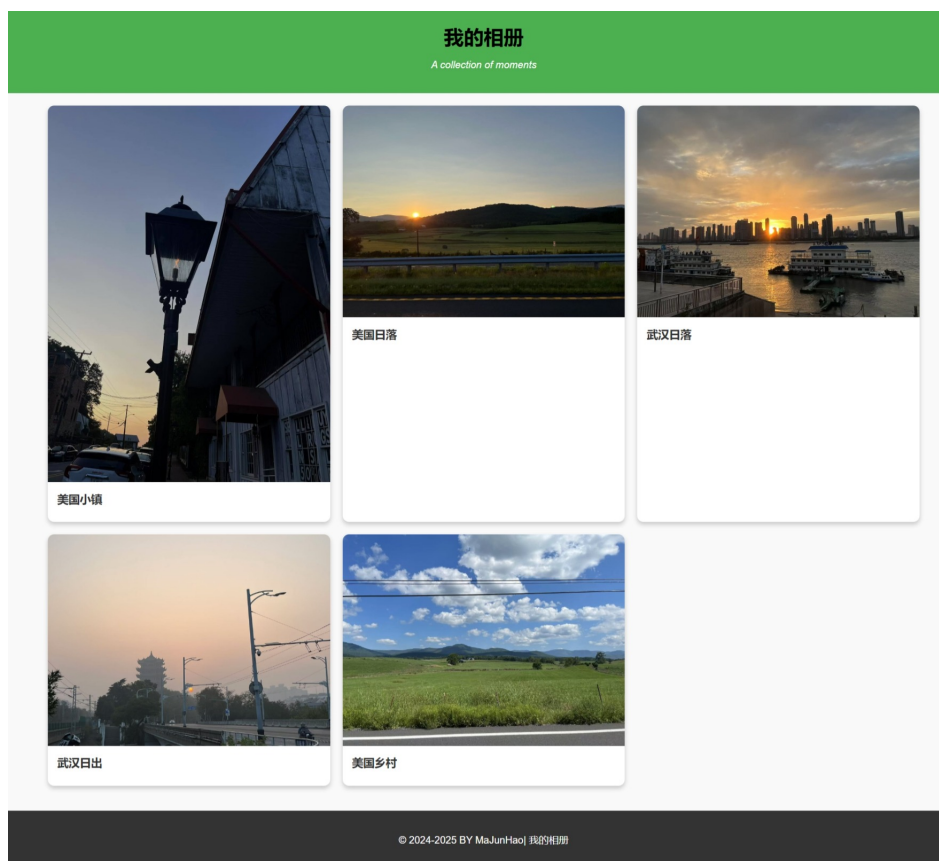


图 3-2 分页面 2 展示

3.3 页面 3 (我的动画作品)

我的动画作品（如图3-3）的页面中，存放了我在 YouTube 上收藏的影片，供浏览者欣赏，由于 flash 已不再支持最新版的浏览器，因此我才想到用 mp4 文件来存放动画作品。同时我也比较细心地编辑了一下知识产权信息的显示，使之能够随着浏览者的滑动，他会一直在页面下方，会一直被看到。其他方面的算法难度不是特别大。大致的算法思路：

1. 目标：存放频频。

逻辑：

-将每个视频展示单独封装为一个容器标题用 <h3> 标识，视频用 <video> 标签，带有播放控制功能。每个容器边框圆角和阴影设计突出层次



图 3-3 分页面 3 展示

3.4 页面 4 (学习心得)

我的大学学习心得（如图3-4）中存放了我的学习经验、心得体会。这个页面主要以内容创作为主，没有过多的编程特色，因此我使用了 chatGPT 对内容进行了润色。同时网站的编写难度不是特别大，因此我加了一些小功能，让其能够导航到想看的内容上。大致的设计算法：

1. 目标：实现导航栏链接功能。

逻辑：

-每个导航链接通过锚点跳转到页面特定部分，链接通过 HTML 锚点（如“自主学习”）绑定到相应的模块标题。



图 3-4 分页面 4 展示

4 网页设计小结

这个网站耗时两周终于做完了，通过这次的网站设计，我有很多收获，在网站的设计上，我运用了多种编程语言，因此我也从中学到了许多知识。同时在设计网站的过程中，我也学到了如何优化网站设计，增强用户体验感与交互感。在学到新语言的同时，也培养了我对网站设计的审美感。在网站设计中，我认识到用户体验（UX）和用户界面（UI）优化的重要性。从用户的角度出发，我可以更好地优化网站布局，调整配色，设计更加直白的导航，并注重页面的交互效果。这些举措显著地提升了网站的易用性和吸引力，让用户更轻松地找到信息并享受其中的操作体验。

虽然这个网站已经完成，但我也发现了一些可以改进的地方。例如在代码优化上，还有一些提升的空间；在用户体验感方面，应该更深入地了解用户需求，从而实现更有针对性的功能设计。未来，我希望能够继续学习更高效的开发工具与框架，同时多关注行业动态，为自己积累更多设计灵感。

5 课程的收获和建议

我认为这个课程让我受益匪浅，它不仅让我了解并掌握如何使用不同的软件，而且我能够学到新的编程语言——python 语言。同时每节课上都会有不同的实验操作，我从中也提升了自己的动手能力。在获得丰富的知识的同时，我也希望提出几点建议，首先，我注意到课堂教师非常大，但是课堂只有一个大屏幕，因此坐在后排的同学几乎看不见老师的投影内容，希望能够多添加几个屏幕，让后排的同学也能看到老师课件。然后，我也注意到在课堂中，老师讲课的节奏偏快，大部分同学宁愿自己操作也不会听老师，因此我建议老师调整讲课节奏。

以上就是我的收获与建议，希望能对课程设计有帮助。

5.1 计算机基础知识

通过对计算机基础知识的学习，我认识到计算机的历史，以及物联网发展脉络，同时我也认识到目前我国与西方国家在技术上的差异。因此我也了解到我们应该培养科学家精神。

在学习计算机知识的过程中，我了解到计算机技术从最初的基础理论的研究到如今的广泛应用，背后凝聚了许多科学家的智慧与努力。从阿兰·图灵奠定现代计算机科学的基础，到冯·诺依曼提出存储程序的计算机体系结构，再到当今人工智能和量子计算的突破，这些都离不开科学家们对基础研究的投入和对技术前沿的不断探索。作为当代青年，我们应主动肩负起历史的使命，积极参与科技创新！

5.2 文档撰写工具 LaTeX

LaTeX 文档撰写工具是论文创作的主要工具，他极大的帮助了人们，为论文创作提供了极大便利。而此实验报告的撰写也是在 LaTeX 上完成的。通过对 LaTeX 的学习，我能够掌握如何使用 LaTeX。而在学习过程中，我也遇到了一些新的阻碍，如新的代码学习，以及文章布局的优化。

5.3 编程工具 Python

我认为 python 语言的学习是本课程中第二难学习的课题，python 对于我来说是一个全新的语言，因此我也遇到了不同的难题，比如说因为 python 与 C 语言的有着截然不同区别，我也需要转换观念，适应新的编程语言。我认为 python 中，比较难学习的是自定义函数以及组合数据类型，不过在我解决问题的同时我也培养了寻找问题的能力以及自主解决问题的能力。

5.4 图像设计软件 Photoshop

photoshop 是我们的自学内容，在下载 photoshop 的时候我就遇到了问题，不过成功下载后，我就没有遇到什么大的问题了，这个软件能够让我编辑图片，同时也为设计网站提供便利，因此我也非常重视这个软件的学习。

5.5 版本管理软件 Git

Git 的使用能够大大的提升团队的工作与沟通效率，在课堂中我们学习了简单的 Git 的指令，方便我们在网站的制作过程中，能够协调其他成员的工作，并且方便我们更新代码，同时，我认为可以适当增加一些内容，让学生能够更好的使用并掌握 Git, 因为 Git 是一个比较专业的上传代码的语言，掌握 Git 能够显得更加专业。

5.6 网页制作 Dreamweaver

通过学习对 Dreamweaver 的使用，我能够更加容易地对网站布局进行修改。减少我手写代码的时间，大幅提升我的工作效率，我认为学生可以自学 Dreamweaver，这样可以将更多的时间放在 python 及 Html 的学习上。