

华中科技大学课程实验报告

##[#]

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 CS 创新 2401

学 号 U202414911

姓 名 刘喆

指导教师 范晔斌

报告日期 2024 年 11 月 24 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
2.1	菜单按钮	3
3	分页面设计	6
3.1	分页一：宗旨的概览	6
3.2	分页二：人物的介绍	7
3.3	分页三：作品内容的浏览	9
3.4	分页四：互动性世界地图	9
4	网页设计小结	11
4.1	问题一：设计效果不知如何实现	11
4.2	问题二：设计结果与目标有出入	11
4.3	问题三：设计不够美观	12
4.4	总结	12
5	课程的收获和建议	13
5.1	计算机基础知识	13
5.2	文档撰写工具 LaTeX	13
5.3	编程工具 Python	14
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	16
5.7	HTML 和 CSS 的基本应用	16
5.8	对网页制作流程的理解	16

1 网页整体框架

本网页的主题灵感来源于日本著名轻小说系列《凉宫春日》，并以此为基础构建，呈现出一种对原作的衍生性阐释与再创作。本网站可视为该系列的一个粉丝向衍生平台，旨在构建一种简单的介绍式网页，向作品致敬，同时为广大粉丝提供一个有趣味的二次创作产物。通过此平台，用户能发掘或再一次感受作品的趣味所在。网页的核心内容页主要包括五个部分，分别是模仿原作中社团网站的主页，介绍团队宗旨的分页一，介绍主要人物的分页二，提供相应小说节选的分页三，提供互动性世界地图的分页四。

网页设计框架设计简洁，主要包含五个页面。框架由一个主页面和多个分页面组成，通过跳转菜单实现页面间的切换。主页面位于框架的顶端，起到导航作用，包含一个跳转菜单，用户可以通过该菜单选择访问不同的分页面。

整体页面框架简略如下图：

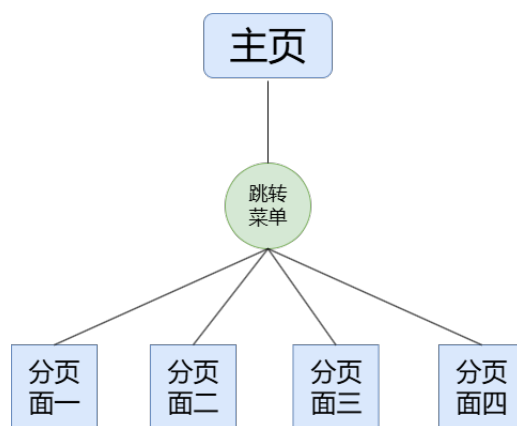


图 1-1 整体页面框架

跳转菜单是整个框架的核心，但并没有单独作为一个网页出现，而是重复地写入各个网页 `html` 文件中，链接通向四个页面：对于每个页面而言，通向五个页面中除了它自身的其他四个页面。每个分页面具有独立的内容，并且循序渐进。用户可以随时从分页面返回主页面，通过菜单继续导航至其他页面。

该设计采用了层次结构清晰的树状结构，访问路径简单直接，同时提升页面可用性和用户体验。

2 主页设计

主页面设计参考了原本动画中的设计（见图 2-1）



图 2-1 动画设计

但在此基础上为了使内容不那么单调，选择加入背景图片。相应地产生了问题：就是文字内容辨识度变低并且图片的留白不那么美观。

为解决这些问题，我选择加入白色圆角边框和半透明的背景底色框住 body 内容。这样一来文字和图片辨识度就变高了。此外，运用 PhotoShop，将图片白色背景去除，使之看上去更自然，于是便得到产物。（见图 2-2）



图 2-2 我的设计

2.1 菜单按钮

菜单按钮的设计在主页初期阶段即已确定，并与整体页面风格紧密融合。确保网站在后续的页面设计中，能够维持视觉与交互体验的一致性。这避免后期出现风格不统一或功能错位的情况。此举不仅增强了网站的视觉连贯性，使得新页面或新功能的添加能够更顺畅地与现有设计进行对接和融合。

一开始浏览其他网页时考虑过以底栏作为跳转菜单，但使用后发现不美观（会遮挡内容），在修改使其固定在页面最底部又发现这不够直接并且由于背景图变得难以辨认。故最后采用如下形式。具体设计与样貌可见图 2-2。

按钮的外貌设计在于 css 文件。

Listing 1 按钮

```
#menu-btn {  
    position: fixed; /* 固定位置，这里也可以不需要 */  
    top: 20px;  
    left: 20px; /* 位于左侧 */  
    background-color: rgba(0, 0, 0, 0.4); // 透明化处理  
    color: #fff;  
    padding: 10px 15px;  
    font-size: 16px;
```

```
border: none;
border-radius: 5px;
cursor: pointer;
z-index: 10;
transition: background-color 0.3s ease;//动画效果更自然
}

#menu-btn:hover {
    background-color: rgba(0, 0, 0, 0.7);
}

.side-menu {
    position: fixed;
    top: 0;
    left: 0;
    height: 100%;
    width: 250px;
    background-color: rgba(255, 255, 255, 0.9);
    padding: 20px;
    transform: translateX(-100%);
    transition: transform 0.3s ease; /* 动画效果使其不那么突兀 */
    z-index: 9; /* 置于最上方防止遮挡 */
}

.side-menu h2 {
    font-size: 24px;
    text-align: center;
    margin-bottom: 20px;
}

.side-menu ul {
    list-style: none;
    padding: 0;
    margin: 0;
}
```

```
.side-menu li {  
    margin: 10px 0;  
}  
  
.side-menu a {  
    text-decoration: none;  
    color: #333;  
    font-size: 18px;  
}  
  
.side-menu a:hover {  
    color: #007acc;  
}
```

其中绝大部分框架摘自网络，在理解涵义后自行修改了部分功能并且加上某些修饰如透明化处理和圆角边框以使页面更为美观。同时，理解按钮的繁杂设计过程，这为运用 css 打下坚实基础，让整体简约风格的规划更为熟练，为后续设计提供不少便利。

JavaScript 部分较为简单，略过。

3 分页面设计

3.1 分页一：宗旨的概览

页面一主要是对团队的一个概述，诸如名称的来历与宗旨是什么，介绍了这是怎样的一个团体，放在第一个分页面显得很合适。以下是具体页面展示。

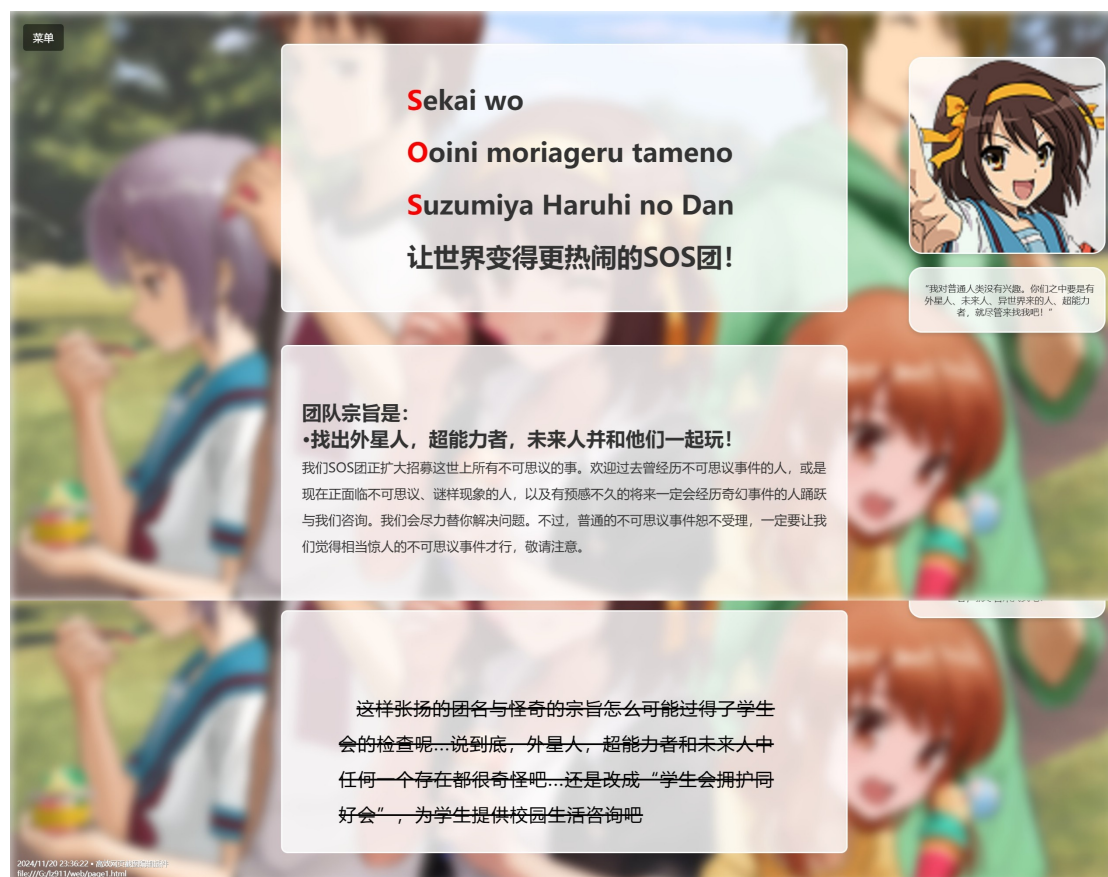


图 3-1 分页面一

页面设计灵感来源于现代网络博客的常见布局，采用了滑动主页面与固定侧分栏的结构。主页面专注于展示核心内容，这里将团队全称与给出，并将首字母标以增强引导性，进而给出更多信息。而侧分栏则承载一些辅助性和装饰性的元素，在这里我选用一张视觉冲击力较强的图片与一段次分栏插入文字，就像是人物在呼唤一样。这种设计不仅让页面内容更加聚焦，而且通过视觉上的分隔，使得每个区域都有明确的功能分配，避免了信息的过度拥挤。

这种设计方式的优点在于它增强了页面的交互感和用户体验。用户可以通

过滚动主页面轻松浏览主要内容，而侧分栏则始终可见，提供即时的信息或导航，无需额外操作。为了进一步提高页面的响应性和视觉层次感，设计上给予主页面和分栏合适的间隙，强调主页面和侧分栏之间的视觉分隔。这种布局不仅符合现代网页设计的趋势，也能有效提升页面的可读性与互动性，信息结构清晰、浏览流畅。

3.2 分页二：人物的介绍

分页面二的设计更为简约与集中，同时由于主页与分页面一中交互功能较为缺乏而设计了更有操作感的设计，这样式源于 mdn 网站中 JavaScript 的教材，即点击图片以切换。这种方式有许多许多优点：

1. **交互性增强**点击图片即可切换内容，为用户提供了直观且易用的操作方式，同时增强了交互趣味性。相比传统的翻页或滚动展示方式，这种设计更具吸引力。
2. **界面简洁**通过模糊背景和中央卡片的设计，界面元素分工明确。背景作为辅景，卡片聚焦于核心内容，保证了整体布局的清晰和整洁。
3. **节省屏幕空间**点击切换方式避免了同时展示多个内容的冗杂，节省页面空间，使得布局更加紧凑，同时也为其他元素（如导航菜单）预留了更多空间。
4. **延展性强**这种设计方便扩展新的角色或内容，只需添加对应的图片和介绍，而无需改变页面结构。
5. **视觉层次分明**背景模糊处理凸显了中央内容区域，同时通过透明卡片和文字对比，使用户视觉焦点集中在信息展示上，避免干扰。



(a) 分页面二示例一



(b) 分页面二示例二

图 3-2 切换示例

具体实现方法并不困难，主要是构建循环队列，定义含有五张图片的数组，

获取点击带来的输入，通过循环实现彼此间切换。

Listing 2 JavaScript 示例

```
// 定义部分
document.addEventListener("DOMContentLoaded", function() {
    var img = document.getElementById("image");
    var description = document.getElementById("image-description");
    var images = [
        { src: ".....", description: "....." },
        { src: ..... },
        { src: .....}
        .....
    ];
    var currentIndex = 0;

    // 这是为了保证如 <br> 命令正常使用
    function updateContent() {
        image.src = images[currentIndex].src;
        description.innerHTML = images[currentIndex].description;
    }
    updateContent();

    // 点击带来索引的变化
    image.addEventListener("click", function() {
        currentIndex = (currentIndex + 1) % images.length;
        updateContent();
    });
});
```

学习这一部分，我先采用 mdn 教程中关于切换图片的框架，再学习多张图片切换的方法，最后在基础上加以润色（带有文字的切换与 description 中
 命令的实现）。一步步地，取得良好成功。

3.3 分页三：作品内容的浏览

分页面三设计采用了简洁直观的风格，就与网站整体基调一样，通过模块化布局划分了内容区域。右侧主干以大图片展示了小说系列的封面，并配有简要的标题介绍，吸引用户注意力。图片下方的“点击查看文档”按钮提供了进一步的互动选项，增强了用户的操作性。设计兼顾了信息传递的效率和用户的视觉体验，是一种功能与美感兼备的网页结构。

此外，左侧的分栏是一种装饰性的有趣味性的文本，这文本选自与出版商的官方网站，但是由于某些原因被迫关闭，让观众与此处网站的资源形成一种有趣的对比。这一页面设计较为美观，因为前面的经验打下坚实基础。



图 3-3 分页面一

此处资源仅供浏览，不提倡下载。

3.4 分页四：互动性世界地图

分页面四是整个设计中最为独特和引人注目的一页。这一页面创新地将动画主题曲舞蹈的视频内容与互动地图结合在一起，兼具趣味性和功能性。首先，此页面设计了一个子页面，其中包含一个关于世界上不同国家地区的粉丝的关于动画主题曲的舞蹈片段。在页面中，嵌入了一块完整的世界地图，用户可以通过点击地图上的不同区域来导航至对应国家或地区来跳转到视频中对应国家地

区的对应部分。每个点击点都精确定位到与该地区相关的视频时间轴，用户无需手动查找即可轻松直达内容。这种设计增强了用户的交互体验，巧妙地将全球动画文化的多样性和魅力呈现出来。实现了“地图即入口，视频即窗口”的设计理念，充满探索的乐趣。

实现这一功能，首先地需要利用世界地图。一开始希望运用 svg 文件特性进行跳转。对应在网站 [https:// simplemaps.com](https://simplemaps.com) 可以找到。然而，实践中发现网站上提供一种更为简便的方法，即它提供一个已经整理完毕的模板，并且在 js 文件中写出了具体不同国家地区跳转的 url。并且地，阅读相关使用规范和许可后，我了解到这一功能免费开放给公众与团体进行非盈利目的，所以使用是合法的。而具体提供的 JavaScript 片段如下所示：

Listing 3 JavaScript 示例

```
UA: {  
  name: "Ukraine",  
  description: "default",  
  color: "default",  
  hover_color: "default",  
  url: "default"  
},
```

其次，使用到视频的时间跳转，通过在 url 后写入 $?t=10$ 便可在当 $t=10$ 存在时跳转到第十秒。这一功能是在通过使用 Chatgpt 寻求并学习的。

综上所述，整体功能便可以实现，下面给出两个页面的概览。



(a) 地图界面



(b) 视频界面

图 3-4 功能展示

在下方，给出了提示信息，因为可供跳转的国家地区仅是部分。故提示跳转国家地区。同时，如果浏览者希望浏览更多的内容，也给出跳转方法。

4 网页设计小结

遇到的问题与解决方案

在进行网页设计学习过程中，我遇到了一些技术性和实现性的问题。这些问题主要集中在功能的实现和代码的结构优化上，而非直接面向用户体验的设计。为了克服这些问题，我通过分析问题的根源，采取了相应的解决方案。以下是我遇到的主要问题的具体描述，以及我为解决这些问题所采取的优化措施。

4.1 问题一：设计效果不知如何实现

这个问题主要源于对设计效果的实现方式缺乏清晰的思路。虽然在设计阶段有了一定的概念，但在如何将设计落到实际网页上时，我发现自己缺乏有效的技术手段。这是实践过程中最大的困难。

解决方案

1. 加强学习和工具使用：加强对前端技术的学习，特别是 HTML、CSS 和 JavaScript 的使用。
2. 寻求参考和灵感：参考一些高质量的网站和开源项目，例如 hexo 网站上一些模板，分析他们的实现方式。网上有很多开源的网页模板和代码库，可以快速找出类似设计的实现方法。

4.2 问题二：设计结果与目标有出入

这个问题发生在设计阶段和开发阶段的过渡过程中。设计图看起来很完美，但当开始实现时，发现代码实现出来的页面与原设计效果相差较大。原因可能在于设计规范不明确或开发时的细节处理不到位。

解决方案

1. 明确设计规范：确保在设计阶段就细化布局、字体、颜色、间距等细节，来保证设计的一致性。
2. 分阶段验证：将设计分成几个阶段逐步验证和实施，避免一次性完成整个页面，逐步改进和修正设计与实际开发之间的差距。

4.3 问题三：设计不够美观

虽然完成了基本的功能性设计，但在美学方面的效果不足。这可能是因为实现时缺乏对细节的打磨，或者设计本身缺乏足够的视觉吸引力。

解决方案

1. 增强细节处理：通过细节的打磨，如按钮的悬浮效果、图标的优化、动画的流畅性等，使得网页看起来更加精致和专业。
2. 用户体验：设计要考虑用户体验。将一些关键细节如网页间跳转方法直接呈现在一目了然的界面，能够流畅使用。
3. 注重排版与配色：优化排版、字体和颜色的搭配，遵循设计美学原则。使用适合的字体和大小、合理的行距和间距，可以大大提升页面的视觉效果。

4.4 总结

网页设计中遇到的这些问题是对于网页设计新手常见的挑战，但通过系统的学习和不断的调整，这些问题是可以逐步解决的。关键在于合理规划设计和开发流程，在每个环节都注重细节与质量。随着实践的深入，相信能够不断提升设计水平，最终实现具备视觉吸引力的网页设计。

5 课程的收获和建议

通过学习新生实践课程，了解现代计算机基本知识，从基础的数据存储与硬件结构，到其他诸如 git 和网页设计。并且在实践作业中进行进一步的实践与学习。但是，课程安排个人认为也有一些亟待改进之处。

5.1 计算机基础知识

5.1.1 对计算机原理的理解更为深入

1. 通过对计算机基础知识的学习，我更加清晰地了解了计算机的基本构成、工作原理以及如何与硬件、操作系统和应用程序互动。
2. 尤其是对计算机硬件（如 CPU、内存、硬盘等）和软件（如操作系统、编程语言）的关系有了更全面的认识。

5.1.2 操作系统基础的掌握

1. 操作系统是计算机的核心组成部分，我学习了操作系统的基本功能，如进程管理、内存管理、文件系统、输入输出控制等。
2. 通过对操作系统内核的学习，我了解了操作系统如何调度资源并提供用户与硬件的接口。

5.1.3 某些专题内容讲解重复

1. 例如，对于数据在计算机中的存储结构和冯·诺伊曼体系，在其他课程中有所介绍，可以适当精简。

5.2 文档撰写工具 LaTeX

5.2.1 对 LaTeX 的基本理解和应用

1. 学习了 LaTeX 的基本语法和结构，包括文档类、章节结构、列表、表格、图片插入等基本操作。
2. 通过编写简单的文章和报告，深刻理解了 LaTeX 在排版复杂文档时的优势，尤其是在数学公式、参考文献管理等方面的能力。

5.2.2 数学公式和符号的排版能力

1. LaTeX 提供了强大的数学公式支持，能够清晰地表达各种数学公式和符号。
2. 学会了如何使用 LaTeX 编写公式，掌握了常用的数学运算符等的排版方式。

5.2.3 自动化和高效的排版处理

1. 学习了如何高效管理文档结构，如自动生成目录、索引、附录等，减少了手动调整页面和内容的工作量。
2. 使用 LaTeX 的命令，使得文档结构更加清晰一致。

5.2.4 讲解时间较短

1. 在映像中，latex 似乎只是一节课的内容，时间过短。而真正的学习更多是在课下思考。
2. 较短的时间带来不充分的实践，这导致某基本功能运用不熟练。
3. 此外，对文件编辑工具，讲解了 latex 且主要运用 latex，可以将 word 实践的部分删除改为 latex 相关。

5.3 编程工具 Python

5.3.1 对 Python 语言的基础掌握

1. 学习了 Python 的基本语法，包括数据类型、运算符、控制结构（如循环、条件语句）等基础知识。
2. 掌握了函数定义、参数传递、模块导入等核心概念，能够编写简单的 Python 脚本和程序。

5.3.2 标准库和第三方库的使用

1. 学习了 Python 的标准库，这是一种功能基本但全面的库，尽管与之前学习的 c/cpp 略有不同，但层面上一些结构能在其他高级语言如 cpp 上找到相同之处。
2. 了解了常用的第三方库，这些库为编写代码提供极大的便利，免去造轮子的过程，简洁高效，运行效率高。

5.3.3 某些 Python 语言基础知识需要更多时间

1. 对于 Python 基础语法的学习某些比较独特的概念比如 python 函数的语法等内容需要更多时间学习。
2. **建议：**可以适当增加基础部分的时间，减少内容广度，但加入更多的实例操作和示范。

5.4 图像设计软件 Photoshop

5.4.1 对图片的二次处理于加工

1. Photoshop 是一个全面强大的图像修改及创造平台。在网页设计中许多需要修改的细节，都可以通过 Photoshop 解决，例如图片的背景色去除，图片大小的调整和模糊化处理等等。
2. 使用 Photoshop，能在短时间内完成对图像的任意加工，无需多方寻找工具。

5.5 版本管理软件 Git

5.5.1 对版本控制的基本理解

1. 学习了 Git 作为分布式版本控制工具的基本概念，理解了版本控制的重要性，尤其是在团队协作中的作用。
2. 掌握了 Git 中的基本命令，如 ‘git init’、‘git clone’、‘git commit’、‘git push’、‘git pull’ 等，能够独立进行版本控制操作。
3. 学会了如何解决 Git 合并冲突，能够通过手动解决冲突或者使用工具来帮助完成冲突的处理。
4. 掌握了 Git 的回滚操作，包括使用 ‘git reset’、‘git revert’ 来撤销提交，灵活控制版本历史。

5.5.2 远程仓库的使用

1. 了解常见的远程仓库操作，如 ‘git remote’、‘git fetch’、‘git merge’ 等命令。

5.5.3 Git 理解不够直观

1. Git 到底在开发中起到什么作用很难真正理解，导致意识不到它的重要性

2. Git 某些命令较为抽象，尤其是在 `reset/revert` 和 `merge` 部分
3. 只是在头歌平台上练习，但并没有在实际项目中理解，可以设计一些小项目进行练习。
4. 在这里，推荐一个可视化学习网站 learngitbranching.js.org

5.6 网页制作 Dreamweaver

5.6.1 使用 Dreamweaver 带来便利

1. 所见即所得的功能展示，让我在进行网页设计时不必先进行编辑再手动跳转到相应界面，提高开发效率
2. 设计视图和代码界面的快捷切换，可以更简便地设计
3. 自动补全功能帮助提升工作效率，减少输入错误

5.6.2 有待改进之处

1. dreamweaver 功能较多，但并未逐个尝试，对这一工具地理解不够
2. snippet 功能不佳，使用 vscode 与相关插件也许能达到更好的效果

5.7 HTML 和 CSS 的基本应用

5.7.1 代码编写

1. 学习了 HTML 结构标签和 CSS 样式的基本用法，能够编写简单的网页页面并进行样式设置。
2. 通过 Dreamweaver 的代码提示和实时预览功能，了解了如何编写 HTML 和 CSS 文件。

5.8 对网页制作流程的理解

5.8.1 基础学习

1. 学习了网页设计与制作的基本流程，从设计、编写代码到最终发布，能规划和实现一个完整的网页项目。