

## 目录

基于 Web 客户端技术的个性化 UI 的设计和编程 .....	2
(Customized UI design and Programming based on Web client technology) .....	2
1. 前言 .....	2
1.1 毕设任务分析 .....	4
1.2 研学计划 .....	4
1.3 研究方法 .....	5
2. 技术总结和文献综述 .....	7
2.1 Web 平台和客户端技术概述 .....	7
2.2 项目的增量式迭代开发模式 .....	9
3. 内容设计概要 .....	9
3.1 分析和设计 .....	10
3.2 项目的实现和编程 .....	10
3.3 项目的运行和测试 .....	11
3.4 项目的代码提交和版本管理 .....	12
4. 移动互联时代的 UI 开发初步——窄屏终端的响应式设计 .....	13
5. 应用响应式设计技术开发可适配窄屏和宽屏的 UI .....	16
6. 个性化 UI 设计中对鼠标交互的设计开发 .....	19
7. 对触屏和鼠标的通用交互操作的设计开发 .....	21
8. UI 的个性化键盘交互控制的设计开发 .....	24
9. 谈谈本项目中的高质量代码 .....	28
10. 用 gitBash 工具管理项目的代码仓库和 http 服务器 .....	29
10.1 经典 Bash 工具介绍 .....	29
10.2 通过 gitHub 平台实现本项目的全球域名 .....	29
10.3 创建一个空的远程代码仓库 .....	29
10.4 设置本地仓库和远程代码仓库的链接 .....	30
参考文献: .....	34
写作指导: .....	34

# 基于 Web 客户端技术的个性化 UI 的设计和编程

(Customized UI design and Programming based on Web client technology)

科师大元宇宙产业学院 2021 级 XXX

## 【摘要】

经过深入研究和广泛参考相关技术书籍、开发者论坛及学术文献，我成功设计并开发了一款个性化的用户界面（UI）应用程序。在开发过程中，我充分利用了 HTML 语言来构建内容框架，CSS 语言来精心打造 UI 的视觉效果，以及 JavaScript 语言来实现用户界面的交互功能。值得一提的是，本项目的所有代码均出自我的人工编写，没有直接采用或导入任何现成的代码框架或库，从而确保了代码的原创性和独特性。此外，该项目还融入了响应式设计编程的理念，使应用程序能够智能地适应不同屏幕尺寸和分辨率，满足移动互联网时代用户多样化的需求。在编程实践中，我大量运用了面向对象的程序设计思想，其中一个显著的亮点是我创建了一个通用的 pointer 模型。这个模型通过一套代码，实现了对鼠标和触屏的兼容控制，展现了高质量编程的技巧和水平。

**关键词：**UI;JavaScript; UI 设计; CSS; 鼠标和触屏

## 1. 前言

为了顺利完成毕业设计项目和毕业论文，我制定了以下学习计划，

首先，我将明确毕业设计项目和毕业论文的具体要求、目标和任务，确保自己对整个项目的期望结果有清晰的认识。根据项目需求，我将系统地学习相关的基础理论知识，如算法原理、编程语言、数据分析方法等，为项目的实施打下坚实的理论基础。通过查阅国内外相关领域的文献资料，了解当前的研究动态和前沿技术，为自己的项目找到创新点和切入点。根据项目需求和理论基础，制定详细的技术路线和实施方案，明确项目的各个阶段和关键节点。：按照技术路线和实施方案，分阶段进行项目的实施工作，包括数据收集、实验设计、模型构建等。在项目实施的过程中，及时整理实验数据和结果，撰写实验报告和阶段性论文，为毕业论文的撰写积累素材。完成初稿后，反复修改和完善论文，确保论文的逻辑性、严谨性和创新性。

针对毕业设计项目，我制定了以下技术路线：

通过市场调研和用户访谈等方式，明确项目的需求点和痛点，为项目的实施提供方向。根据项目需求和技术现状，选择合适的技术框架和工具，确保项目的

顺利实施。数据收集与处理模型构建与训练：根据项目的需求点，选择合适的算法或模型进行构建和训练，如深度学习模型、机器学习算法等，确保模型的有效性和准确性。通过交叉验证、混淆矩阵等方式对模型进行评估和优化，提高模型的性能和泛化能力。将训练好的模型进行系统集成和部署，实现项目的实际应用和商业化。

为了完成毕业设计项目和毕业论文，我将参考以下资料：

阅读相关领域的专业书籍，了解项目的理论基础和技术细节。查阅国内外相关领域的学术论文，了解当前的研究动态和前沿技术。参考相关领域的开源项目，学习其技术实现和代码风格。阅读相关技术框架和工具的技术文档，了解其使用方法和最佳实践。

在毕业设计项目和毕业论文中，我将采用以下研究方法：

通过查阅相关领域的文献资料，了解当前的研究动态和前沿技术，为自己的项目找到创新点和切入点。通过设计实验来验证项目的可行性和有效性，收集实验数据并进行分析和解读。将实验结果与已有研究成果进行对比分析，评估项目的性能和优势。选择典型案例进行深入剖析和研究，了解项目的实际应用和效果。在项目实施和论文撰写的过程中，及时总结经验和教训，形成自己的研究思路和方法论。

## 1.1 毕设任务分析

毕业设计任务是对学生所学知识进行综合运用、提高实践能力、培养创新思维的重要环节。本次毕业设计任务旨在解决某一特定领域的实际问题，通过深入研究和系统实践，提出一套创新且有效的解决方案。以下是对本次毕设任务的分析。

首先，我们需要明确毕设任务的背景和意义。这包括了解任务所在领域的现状、存在的问题以及发展趋势。通过对背景的深入剖析，我们能够更清晰地认识到任务的必要性和紧迫性，从而激发研究的动力。同时，明确任务的意义有助于我们把握研究方向，确保研究成果的实用性和创新性。

毕设任务的目标应该具体、明确、可衡量。在明确目标时，我们需要考虑以下几个方面：

**解决实际问题：**毕设任务应该针对某一实际问题展开研究，通过提出解决方案来解决实际问题。这要求我们对问题进行深入的分析和研究，了解问题的根源和关键要素。**提高技术水平：**毕设任务应该能够提高学生的技术水平和实践能力。在任务过程中，学生需要运用所学的专业知识进行设计和开发，这将有助于提升他们的技术能力。**培养创新思维：**毕设任务应该能够培养学生的创新思维和创新能力。在解决问题时，学生需要尝试新的思路和方法，这将有助于培养他们的创新思维和创新能力。在明确任务目标后，我们需要对任务的具体要求进行详细的分析。这包括以下几个方面：**技术路线要求：**毕设任务需要明确技术路线，包括所使用的技术框架、算法、数据处理方法等。这要求我们在设计技术路线时要充分考虑问题的需求和技术的可行性。**系统实现要求：**毕设任务需要实现一个完整的系统或模块。在系统实现过程中，我们需要遵循软件工程的原则和规范，确保系统的稳定性和可靠性。**实验与测试要求：**毕设任务需要进行实验和测试来验证系统的性能和效果。我们需要设计合理的实验方案和数据收集方法，并对实验结果进行科学的分析和评估。**论文撰写要求：**毕设任务需要撰写一篇毕业论文来展示研究成果。论文需要遵循学术论文的规范和要求，包括引言、正文、结论等部分。在撰写论文时，我们需要注重论文的逻辑性和条理性，确保论文的清晰易懂。

在完成毕设任务的过程中，我们可能会遇到一些难点和挑战。这包括技术难题、时间压力、团队协作等方面。我们需要认真分析这些难点和挑战，并制定相应的解决方案和应对策略。例如，我们可以加强技术学习和交流，提高技术水平；制定合理的时间计划，确保任务按时完成；加强团队协作和沟通，提高团队效率等。

通过对毕设任务的分析，我们可以更清晰地认识到任务的背景、意义、目标、要求以及难点和挑战。这将有助于我们更好地把握研究方向和方法，确保毕设任务的顺利完成。同时，我们也应该看到毕设任务是一个学习和成长的过程，通过完成任务我们可以提高自己的技术水平和实践能力，培养创新思维和创新能力。展望未来，我们将继续努力学习和探索新的技术和方法，为解决更多实际问题做出贡献。

## 1.2 研学计划

为了顺利完成毕设任务，我制定了以下学习计划：

**基础知识学习：**在项目开始之前，我将系统地学习所需的基础理论知识，包括编程语言、算法原理、数据处理方法等。这些基础知识将为后续的研究提供坚实的支撑。**文献调研与资料收集：**在明确了问题之后，我将进行广泛的文献调研和资料收集工作。通过查阅相关领域的文献和资料，了解当前的研究动态和前沿技术，为技术路线的设计和系统的实现提供重要的参考。**技术路线设计与实践：**在掌握了足够的基础知识和资料之后，我将开始设计技术路线并进行实践。这个过程需要综合考虑问题的需求、技术的可行性和创新性等因素，通过实践不断调整和优化技术路线。**论文撰写与修改：**在系统实现和测试完成后，我将开始撰写毕业设计论文。我将按照学术论文的规范和要求，系统地展示研究成果和实验数据。在论文撰写完成后，我将进行反复的修改和完善工作，确保论文的质量和水平。

## 1.3 研究方法

在进行毕业设计项目和撰写毕业论文的过程中，研究方法的选择和应用至关

重要。它决定了我们如何系统地收集、整理、分析和解释数据，从而得出有效的结论。以下是我对研究方法的理解和阐述：

文献综述法是我研究工作的起点和基础。通过系统地查阅和整理相关领域的文献资料，我可以了解问题的研究现状、理论基础、技术方法以及前人的研究成果。这有助于我明确研究方向，避免重复工作，并为后续的研究提供理论支持。在文献综述的过程中，我将采用关键词搜索、引文追踪等方法，广泛收集相关文献，并对其进行分类、整理和分析。通过对比不同文献中的观点和方法，我可以形成自己的研究思路，为后续的实验设计和数据分析提供指导。实验研究法是我进行毕业设计项目时的主要研究方法。通过设计实验方案、构建实验环境、收集实验数据等步骤，我可以验证和评估所提出的解决方案的有效性和可行性。在实验设计过程中，我将充分考虑问题的需求和技术的可行性，选择合适的实验对象、实验条件和实验方法。同时，我也将注重实验的可重复性和数据的可靠性，确保实验结果的准确性和可信度。在实验过程中，我将采用适当的实验技术和数据处理方法，对实验数据进行收集、整理和分析。通过对比不同实验条件下的实验结果，我可以得出有意义的结论，并为后续的应用和推广提供实验依据。

案例研究法是我了解实际问题、评估解决方案效果的重要方法。通过选择典型的案例进行深入剖析和研究，我可以了解问题的实际情况、解决方案的应用效果以及存在的问题和挑战。

在案例选择过程中，我将充分考虑案例的代表性、典型性和实用性，选择能够反映问题实际情况和解决方案效果的案例。同时，我也将注重案例的多样性和可比性，以便进行更全面的分析和评估。在案例研究过程中，我将采用访谈、观察、文档分析等方法收集案例数据，并运用定性和定量分析方法对数据进行处理和分析。通过案例研究，我可以更深入地了解问题的实际情况和解决方案的效果，为后续的改进和优化提供有益的参考。

归纳总结法是我对整个研究过程进行梳理和总结的重要方法。通过回顾研究过程、分析研究结果、总结经验教训等步骤，我可以得出有意义的结论和启示，为未来的学习和工作提供借鉴和参考。

在归纳总结过程中，我将注重研究的系统性和逻辑性，确保结论的准确性和可信度。同时，我也将注重研究的创新性和实用性，提出具有创新性和实用性的

观点和建议。

总之，研究方法的选择和应用是毕业设计项目和毕业论文成功的关键。通过综合运用文献综述法、实验研究法、案例研究法和归纳总结法等多种研究方法，我可以更系统地收集、整理、分析和解释数据，从而得出有效的结论和启示。

## 2. 技术总结和文献综述

### 2.1 Web 平台和客户端技术概述

Web 之父 Tim Berners-Lee 在发明 Web 的基本技术架构以后，就成立了 W3C 组织，该组织在 2010 年后推出的 HTML5 国际标准，结合欧洲 ECMA 组织维护的 ECMAScript 国际标准，几乎完美缔造了全球开发者实现开发平台统一的理想，直到今天，科学家与 Web 行业也还一直在致力于完善这个伟大而光荣的理想<sup>[1]</sup>。学习 Web 标准和 Web 技术，学习编写 Web 程序和应用有关工具，最终架构一套高质量代码的跨平台运行的应用，是我的毕设项目应用的技术路线。

#### 2.2.1 历史

在 1989 年，蒂姆·伯纳斯-李爵士（Tim Berners-Lee）发明了万维网（参见原始提案）。他于 1990 年 10 月创造了“万维网”这个术语，编写了第一个万维网服务器“httpd”，以及第一个客户端程序（一个浏览器和编辑器）“WorldWideWeb”。

他编写了“超文本标记语言”（HyperText Markup Language, HTML）的第一个版本，这是一种具有超文本链接能力的文档格式化语言，后来成为网络的主要发布格式。随着网络技术的普及，他最初的关于统一资源标识符（URIs）、超文本传输协议（HTTP）和 HTML 的规范得到了进一步的完善，并在更广泛的范围内进行了讨论。

#### 2.2.2 万维网联盟

1994 年，在众多公司不断增加对互联网投资的压力下，人们决定成立万维网联盟。蒂姆·伯纳斯-李爵士开始领导万维网联盟团队的重要工作，以培育一种一致的架构，以适应网站、浏览器、设备构建过程中网络标准快速发展的步伐，以体验网络所能提供的一切。

在创建万维网联盟时，蒂姆·伯纳斯-李爵士创建了一个同行社区。当时网络技术发展得如此之快，以至于组建一个单一的组织来协调网络标准变得至关重要。蒂姆接受了麻省理工学院（MIT）的提议，该学院有联合体（consortia）方面的经验，以作为 W3C 的东道主。他一开始就要求 W3C 具有全球影响力。

### 2.2.3 Web 平台与网络编程

让我们首先简要描述一下 Web，它是 World Wide Web 的缩写。大多数人会说“Web”而不是“World Wide Web”，我们将遵循这一惯例。Web 是由文档组成的集合，这些文档被称为网页，主要由全世界的计算机用户共享。不同类型的网页有不同的功能，但最基本的是，它们都在计算机屏幕上显示内容。这里的“内容”指的是文本、图片以及用户输入机制，如文本框和按钮。[2]

网络编程是一个广泛的领域，由不同类型的网络编程工具实现。所有工具都使用核心语言 HTML，因此几乎所有的网络编程书籍都在一定程度上描述了 HTML。这本教科书深入讲解了 HTML5、CSS 和 JavaScript。这三种技术被称为客户端网络编程的支柱。在客户端网络编程中，所有网页计算都在最终用户的计算机（客户端计算机）上执行。[3]

Web 应用的程序设计体系由三大语言有机组成：HTML，CSS， JavaScript。这三大语言的组合也体现了人类社会化大生产分工的智慧，可以看作用三套相对独立体系实现了对一个信息系统的描述和控制，可以总结为：HTML 用来描述结构（Structure）、CSS 用来描述外表（presentation）、Javascript 用来描述行为（Behavior）<sup>[3]</sup>；这也可以用经典的 MVC 设计模式来理解 Web 平台架构的三大基石，Model 可以理解为 HTML 标记语言建模，View 可以理解为用 CSS 语言来实现外观，Controller 则可理解为用 JavaScript 结合前面二个层次，实现了在微观和功能层面的代码控制。

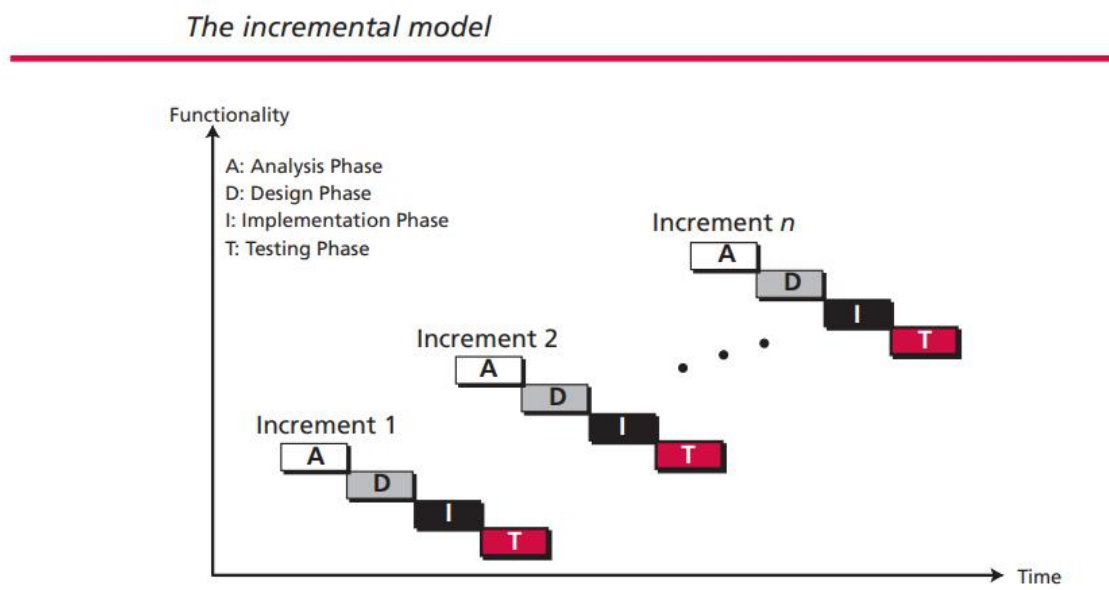


## 2.2 项目的增量式迭代开发模式

本项目作为一个本科专业学生毕业设计的软件作品，与单一用途的程序相比较为复杂，本项目所涉及的手写代码量远超过简单一二个数量级以上，从分析问题的到初步尝试写代码也不是能在几天内能落实的，可以说本项目是一个系统工程，因此需要从软件工程的管理视角来看待和规范项目的编写过程。

而本项目考虑选择的软件工程开发过程管理模式有两种经典模型：瀑布模型（The waterfall model）和增量式迭代模型（The incremental model）。而任何开发模式则都必须同样经历四个阶段：分析（Analysis）、设计（Design）、实施（Implementation）、测试（test）。

瀑布模型需要专业团队完美的配合，从分析、设计到实施，最后到测试，任何阶段的开始必须基于上一阶段的完美结束。而这对于我们大多数普通开发者是不太现实的，作为小微开发者由于身兼数职，其实无法 1 次就能完美完成任何阶段的工作，比如在实施过程中，开发者会发现前面的设计存在问题，则必须在下一次迭代项目时改良设计。在当今开源的软件开发环境中，开发者在软件的开发中总是在不断地优化设计、重构代码，持续改进程序的功能和代码质量。因此在本项目的开发中，也采用了增量模型的开发模式<sup>[5]</sup>。本项目中我一共做了六次项目的开发迭代，如下图 2-1 所示：



增量模型

在增量模型中，软件是通过一系列步骤开发的。开发人员首先完成整个系统的简化版本。这个版本代表了整个系统，但不包括细节。图 X 展示了增量模型的概念（注：由于原文中未提供具体的图号，这里用“图 X”代替）。

在第二个版本中，会添加更多的细节，同时有些部分会保持未完成状态，并再次对系统进行测试。如果出现问题，开发人员知道问题出在新功能上。他们不会在现有系统工作正常之前添加更多功能。这个过程一直持续到所有需要的功能都被添加为止。[5]

### 3. 内容设计概要

#### 3.1 分析和设计

这一步是项目的初次开发，本项目最初使用人们习惯的“三段论”式简洁方式开展内容设计，首先用一个标题性信息展示 logo 或文字标题，吸引用户的注意力，迅速表达主题；然后展现主要区域，也就是内容区，“内容为王”是项目必须坚守的理念，也是整个 UI 应用的重点；最后则是足部的附加信息，用来显示一些用户可能关心的细节变化。如图 4-1 用例图所示：

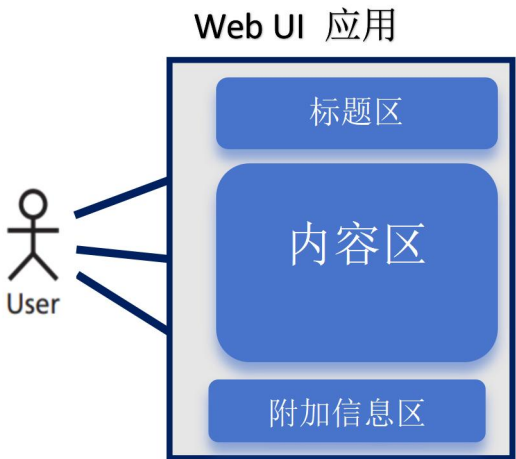


图 4-1 用例图

#### 3.2 项目的实现和编程

### 一、HTML 代码编写如下：

```
<header>
    《 我的毕设题目 》
</header>
<main>
    我的主题内容： ‘读好书、练思维、勤编程’ @masterLijh 计算思维系列课程
</main>
<footer>
    Copyright 李添鹏 江西科技师范大学 2024-2025
</footer>
```

### 二、CSS 代码编写如下：

```
*{
    margin: 10px;
    text-align: center;
    font-size: 30px ;
}
header{
    border: 2px solid blue;
    height: 200px;
}

main{
    border: 2px solid blue;
    height: 400px;
}
footer{
    border: 2px solid blue;
    height: 100px;
}
a{
    display: inline-block ;
    padding: 10px ;
    color: white;
    background-color: blue;
    text-decoration: none ;
}
```

## 3.3 项目的运行和测试

项目的运行和测试至少要通过二类终端，本文此处仅给出 PC 端用 Chrome 浏览器打开项目的结果，如下图 4-2 所示。由于本项目的阶段性文件已经上传 [github](#) 网站，移动端用户可以通过扫描图 4-3 的二维码，运行测试本项目的第一次开发的阶段性效果。



图 4-2 PC 端运行效果图



图 4-3 移动端二维码

### 3.4 项目的代码提交和版本管理

本项目的文件通过 `gitBash` 工具管理，作为项目的第一次迭代，在代码提交和版本管理环节，我们的目标是建立项目的基本文件结构，还有设置好代码仓库的基本信息：如开发者的名字和电子邮件。

进入 `gitBash` 命令行后，按次序输入以下命令：

```
$ cd /  
$ mkdir webUI  
$ cd webUI  
$ git init  
$ git config user.name 江科师大李添鹏  
$ git config user.email 3578206180@qq.com  
$ touch index.html myCss.css
```

编写好 `index.html` 和 `myCss.css` 的代码，测试运行成功后，执行下面命令提交代码：

```
$ git add index.html myCss.css  
$ git commit -m 项目第一版：“三段论”式的内容设计概要开发
```

成功提交代码后，`gitbash` 的反馈如下所示：

```
[main 6a3d273] 三段论“式的内容设计概要  
2 files changed, 46 insertions(+)  
create mode 100644 2index.html  
create mode 100644 2myCSS.css
```

项目代码仓库自此也开启了严肃的历史记录，我们可以输入日志命令查看，

```
$ git log
```

`gitbash` 反馈代码的仓库日志如下所示：

```
tcp@LAPTOP-N8081J3K MINGW64 ~/abc (main)
$ git log
commit 6a3d27306a2aa3a9ff27fc2e6892fb33c5957265 (HEAD -> main)
Author: 李添鹏 <3578206180@qq.com>
Date: Sun Jun 16 14:52:35 2024 +0800

    三段论“式”的内容设计概要

git commit -m 三段论“式”的内容设计概要
```

## 4. 移动互联时代的 UI 开发初步——窄屏终端的响应式设计

### 响应式设计——适应显示硬件

与计算机一同使用的显示硬件千差万别，显示器的尺寸和分辨率取决于成本。与其为每个类型的显示器都设计一个网页版本，设计师们选择让网页给出一般的布局指导，并让浏览器自行选择在特定计算机上如何显示页面。因此，网页不会给出太多细节。例如，网页的作者可以指定一组句子构成一个段落，但作者不能指定诸如每行文字的精确长度或是否要缩进段落开头等细节。[1]

允许浏览器选择显示细节有一个有趣的后果：当通过两个不同的浏览器查看或在两个具有不同硬件的计算机上查看时，一个网页的外观可能会有所不同。如果一个屏幕的宽度大于另一个屏幕，那么一行文本的长度或可以显示的图像大小就会有所不同。重点在于：网页给出了关于所需呈现方式的总体指导；浏览器在显示页面时选择细节。因此，当在不同的计算机上或由不同的浏览器显示时，相同的网页可能会略有不同。[1]

分析移动互联时代的多样化屏幕的需求。

用 JavaScript 开动态读取显示设备的信息，然后按设计，使用 js+css 来部署适配当前设备的显示的代码。

实现代码

用汉语言来描述我们是如何实现的，与上一阶段比较，本阶段初次引入了 em 和 %，这是 CSS 语言中比较高阶的语法，可以有效地实现我们的响应式设计。如代码块 4-1 所示：

```
<style>
*{
```

```

margin: 10px;
text-align: center;
}

header{
  border: 2px solid blue;
  height: 15%;
  font-size: 1.66em;
}

main{
  border: 2px solid blue;
  height: 70%;
  font-size: 1.2em;
}

nav{
  border: 2px solid blue;
  height: 10%;
}

nav button{
  font-size: 1.1em;
}

footer{
  border: 2px solid blue;
  height: 5%;
}
</style>

```

代码块 4-1

用汉语言来描述我们是如何实现的：与上一阶段比较，本阶段首次使用了 JavaScript，首先创建了一个 UI 对象，然后把系统的宽度和高度记录在 UI 对象中，又计算了默认字体的大小，最后再利用动态 CSS，实现了软件界面的全屏设置。如代码块 4-2 所示：

```

<script>
  var UI = {};
  UI.appWidth = window.innerWidth > 600 ? 600 : window.innerWidth ;
  UI.appHeight = window.innerHeight;
  const LETTERS = 22 ;
  const baseFont = UI.appWidth / LETTERS;

  //通过更改 body 对象的字体大小，这个属性能够遗传其子子孙孙

```

```
document.body.style.fontSize = baseFont + "px";
//通过把 body 对象的宽度和高度设置为设备/屏幕的宽度和高度，实现全屏。
//通过 CSS 对子对象百分比（纵向）的配合，从而实现响应式设计的目标。
document.body.style.width = UI.appWidth - 2*baseFont + "px" ;
document.body.style.height = UI.appHeight - 4*baseFont + "px";
</script>
```

代码块 4-2

本文通过 PC 平台，用 chrome 浏览器打开本阶段上传至 http 服务器的代码，网址（<https://litianpeng123.github.io/1.2.html>），效果图如图 4-3 所示。还可以通过手机平台，用 safari 浏览器打开本阶段上传至 http 服务器的代码。网址（<https://litianpeng123.github.io/1.2.html>），效果图如图 4-4 所示。表明目前项目运行正常，达到了项目初步的目标。移动端可以使用图 4-5 的二维码，查看代码运行效果。

代码的运行效果图：



图 4.3

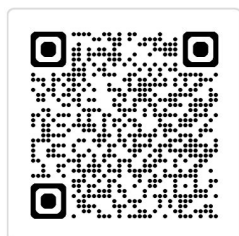


图 4.5

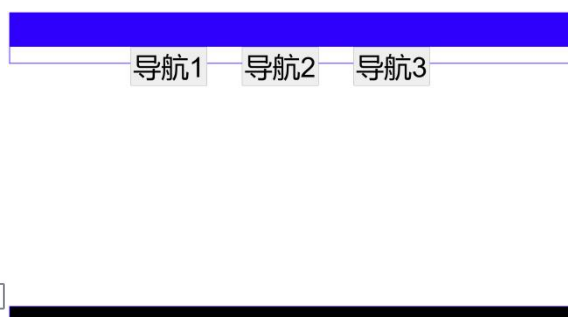


图 4.4

## 4.4 代码提交

本阶段完成的代码存放在 3index.html 和 3myCSS.css 两个文件中，通过 gitbash 工具作正式的代码提交。结果如图 4-6 所示：

```

ltp@LAPTOP-N808IJ3K MINGW64 ~/abc (main)
$ git add 3index.html 3myCSS.css

ltp@LAPTOP-N808IJ3K MINGW64 ~/abc (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   3index.html
        new file:   3myCSS.css

```

图 4-6

命令行如下：

```
git add 1index.html 1myCSS.css
```

```
git commit -m 移动互联网时代的响应式设计和窄屏代码实现
```

本次代码提交成功，结果如图 4-7 所示：

```

ltp@LAPTOP-N808IJ3K MINGW64 ~/abc (main)
$ git commit -m 移动互联网时代的UI开发初步—窄屏终端的响应式设计
[main 2416974] 移动互联网时代的UI开发初步—窄屏终端的响应式设计
2 files changed, 61 insertions(+)
create mode 100644 3index.html
create mode 100644 3myCSS.css

```

图 4-7

gitbash 反馈代码的仓库日志如下所示：

```

ltp@LAPTOP-N808IJ3K MINGW64 ~/abc (main)
$ git log
commit 24169748517bb4101cadb460594a88330d9b6b6d (HEAD -> main)
Author: 李添鹏 <3578206180@qq.com>
Date: Sun Jun 16 15:09:19 2024 +0800

    移动互联网时代的UI开发初步—窄屏终端的响应式设计

```

## 5. 应用响应式设计技术开发可适配窄屏和宽屏的 UI

在移动互联网时代，用户终端的多样性是一个显著的特点。这些终端不仅包括传统的智能手机和平板电脑，还涵盖了智能手表、智能电视以及其他物联网设备。每种设备都有其独特的屏幕尺寸、分辨率、触摸交互方式以及处理能力。为了满足这些不同设备上的良好用户体验，设计师和开发者采用了响应式设计（Responsive Design）的方法。

响应式设计的目标是确保网站或应用在各种设备和屏幕尺寸上都能良好地运行和显示。它不仅仅是调整布局和字体大小，还包括考虑交互方式和性能优化。

使用 CSS 语言实现响应式设计：媒体查询（Media Queries）：CSS3 引入了媒体查询，允许你根据设备特性（如宽度、高度、分辨率等）应用不同的样式规则。例如，你可以为一个较窄的屏幕设置一个单列的布局，而为较宽的屏幕设置



一个多列的布局。视口单位（Viewport Units）：如 vw（视口宽度的 1%）、vh（视口高度的 1%）等，可以帮助你创建与屏幕尺寸成比例的布局和元素。Flexbox 和 Grid：这两个 CSS 布局模块提供了更强大和灵活的布局工具，可以轻松创建响应式布局。使用 JavaScript 语言实现响应式设计：虽然 CSS 是响应式设计的主要工具，但 JavaScript 也可以在某些情况下发挥作用。动态样式修改：JavaScript 可以动态地修改 HTML 元素的样式，以适应不同的屏幕尺寸或设备特性。例如，你可以使用 JavaScript 检测设备的屏幕宽度，并根据该宽度应用不同的样式。交互式响应：JavaScript 可以用于创建交互式响应式元素，如折叠菜单、滑动图片等。这些元素可以根据用户的触摸或点击事件进行动态调整。性能优化：对于资源受限的设备（如某些移动设备），JavaScript 可以用于动态加载资源（如图片、脚本等），以提高页面加载速度和性能。在移动互联时代，用户终端的多样性要求设计师和开发者采用响应式设计的方法。通过结合使用 CSS 和 JavaScript，可以创建在各种设备和屏幕尺寸上都能良好运行的网站和应用。同时，还需要考虑交互方式和性能优化，以确保用户获得最佳体验。

## 5.2 项目的运行和测试

本文通过 PC 平台，用 chrome 浏览器打开本阶段上传至 http 服务器的代码，网址（<https://litianpeng123.github.io/1.3.html>），效果图如图 5-3 所示。还可以通过手机平台，用 safari 浏览器打开本阶段上传至 http 服务器的代码。网址（<https://litianpeng123.github.io/1.3.html>），效果图如图 5-4 所示。表明目前项目运行正常，达到了项目初步的目标。移动端可以使用图 5-5 的二维码，查看代码运行效果。代码的运行效果图：



图 5-3 PC 端运行效果图



图 5-4 手机端运行效果图

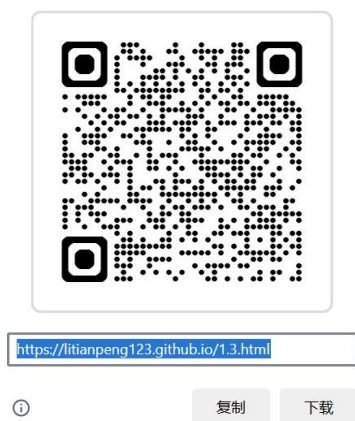


图 5-5 移动端二维码入口

## 5.2 代码提交

本阶段完成的代码存放在 4index.html 和 4myCSS.css 两个文件中，通过 gitbash 工具作正式的代码提交。结果如图 5-6 所示：

```
ltp@LAPTOP-N808I13K MINGW64 ~/abc (main)
$ git add 4index.html 4myCSS.css

ltp@LAPTOP-N808I13K MINGW64 ~/abc (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
(use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   4index.html
        new file:   4myCSS.css
```

图 5-6 本次代码提交的文件列表

命令行如下：

```
git add 4index.html 4myCSS.css
git commit -m 适用移动互联网时代的响应式设计
```

本次代码提交成功，结果如图 5-7 所示：

```
ltp@LAPTOP-N808I13K MINGW64 ~/abc (main)
$ git commit -m 5.应用响应式设计技术开发可适配窄屏和宽屏的UI
[main eb7bf36] 5.应用响应式设计技术开发可适配窄屏和宽屏的UI
2 files changed, 85 insertions(+)
create mode 100644 4index.html
create mode 100644 4myCSS.css
```

图 5-7 本次代码提交反馈图

gitbash 反馈代码的仓库日志如下所示：

```
commit eb7bf36ebeca4e01385f7a491a9655596810dcb6 (HEAD -> main)
author: 李添鹏 <3578206180@qq.com>
date: Sun Jun 16 15:41:53 2024 +0800

    5.应用响应式设计技术开发可适配窄屏和宽屏的UI
```

## 6. 个性化 UI 设计中对鼠标交互的设计开发

### 6.1 个性化 UI 设计

在开发一个同时支持触屏和鼠标交互的网页或应用时，我们可以使用一套代码逻辑来建立对象模型，以便能够处理来自不同输入设备（如触屏和鼠标）的事件。以下是如何实现这一点的概述：

首先，我们需要能够识别用户当前使用的输入设备是触屏还是鼠标。虽然这通常不是直接识别的（因为浏览器并不直接提供这样的 API），但我们可以根据触发的事件类型来间接判断。例如，触屏设备通常会产生 `touchstart`、`touchmove` 和 `touchend` 事件，而鼠标则会产生 `mousedown`、`mousemove` 和 `mouseup` 事件。接下来，我们可以创建一个统一的交互对象模型，该模型可以处理来自不同输入设备的事件。这个模型可以是一个类或者一组函数，它们封装了处理不同交互事件的逻辑。然后，我们需要在需要响应交互的元素上添加事件监听器。这些监听器将调用 `InteractionHandler` 类中的相应方法。在 `InteractionHandler` 类中，我们可以根据具体需求实现事件处理逻辑。这些逻辑可以包括更新 UI、发送网络请求、改变应用状态等。为了使代码更加通用和可维护，我们可以进一步抽象输入数据。例如，我们可以创建一个 `InputEvent` 类，该类封装了来自不同输入设备的事件数据（如位置、压力等）。然后，在 `InteractionHandler` 的方法中，我们可以使用 `InputEvent` 对象而不是原始事件对象来处理交互。性能优化：在处理大量事件时，需要注意性能问题。避免不必要的 DOM 操作，使用事件委托等技术来减少事件监听器的数量。

跨浏览器兼容性：不同的浏览器可能对触摸事件和鼠标事件的支持有所不同。确保你的代码在各种浏览器上都能正常工作。触摸和鼠标的混合使用：在某些情况下，用户可能会同时使用触摸和鼠标（例如，在触摸屏笔记本电脑上）。确保你

的代码能够妥善处理这种情况。

## 6.2 项目的运行和测试

本文通过 PC 平台,用 chrome 浏览器打开本阶段上传至 http 服务器的代码,网址 (<https://litianpeng123.github.io/1.4.html>),效果图如图 6-3 所示。还可以通过手机平台,用 safari 浏览器打开本阶段上传至 http 服务器的代码。网址 (<https://litianpeng123.github.io/1.4.html>),效果图如图 6-4 所示。表明目前项目运行正常,达到了项目初步的目标。移动端可以使用图 6-5 的二维码,查看代码运行效果。

代码的运行效果图:

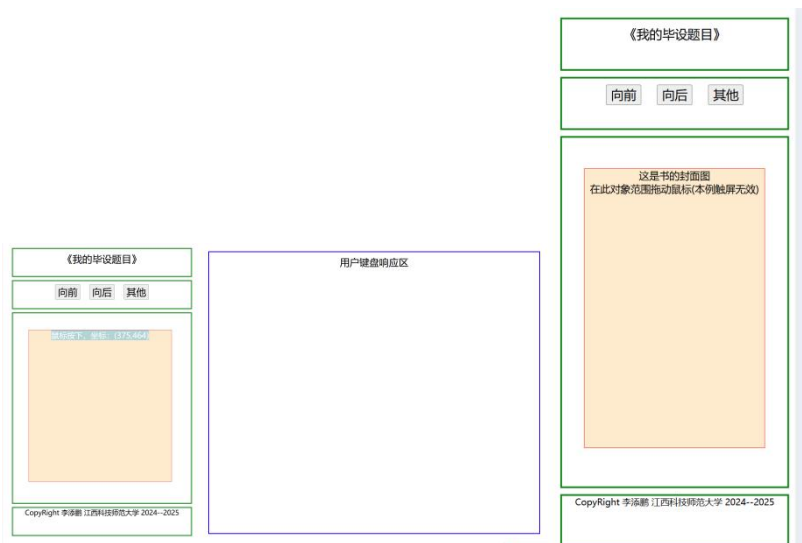


图 6-3 PC 端运行效果图

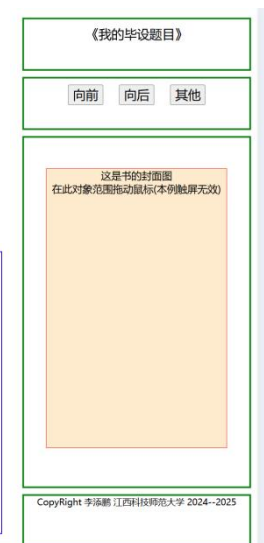


图 6-4 手机端运行效果图



图 6-5 移动端二维码入口

## 6.3 代码提交

本阶段完成的代码存放在 5index.html 和 5myCSS.css 两个文件中，通过 gitbash 工具作正式的代码提交。结果如图 6-6 所示：

```
ltp@LAPTOP-N808IJ3K MINGW64 ~/abc (main)
$ git add 5index.html 5myCSS.css

ltp@LAPTOP-N808IJ3K MINGW64 ~/abc (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 3 commits.
(use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   5index.html
        new file:   5myCSS.css
```

图 6-6 本次代码提交的文件列表

命令行如下：

```
git add 5index.html 5myCSS.css
git commit -m 个性化 UI 设计中鼠标模型
```

本次代码提交成功，结果如图 6-7 所示：

```
ltp@LAPTOP-N808IJ3K MINGW64 ~/abc (main)
$ git commit -m 6.个性化UI设计中对鼠标交互的设计开发
[main 1fd609a] 6.个性化UI设计中对鼠标交互的设计开发
2 files changed, 86 insertions(+)
create mode 100644 5index.html
create mode 100644 5myCSS.css
```

图 6-7 本次代码提交反馈图

gitbash 反馈代码的仓库日志如下所示：

```
ltp@LAPTOP-N808IJ3K MINGW64 ~/abc (main)
$ git log
commit 1fd609a1c723360b7e6f165e4be7b04c77f76500 (HEAD -> main)
Author: 李添鹏 <3578206180@qq.com>
Date: Sun Jun 16 15:55:44 2024 +0800

    6.个性化UI设计中对鼠标交互的设计开发
```

## 7. 对触屏和鼠标的通用交互操作的设计开发

当我们要为触屏和鼠标建立对象模型，并使用一套代码逻辑来处理这两种不同的输入方式时，我们需要考虑的是如何抽象出这两种输入方式的共同点，并编写能

够处理这些共同点的代码。以下是如何实现这一点的详细阐述：虽然我们不能直接识别输入设备是触屏还是鼠标，但我们可以根据触发的事件类型来推断输入源。触屏设备通常触发 `touchstart`、`touchmove`、`touchend` 等事件，而鼠标则触发 `mousedown`、`mousemove`、`mouseup` 等事件。我们可以创建一个抽象的事件对象或类，用于封装不同输入设备的事件数据。这个抽象事件对象应该包含一些共同的属性，如位置（`x`、`y` 坐标）、时间戳等。然后，我们可以为触屏和鼠标事件创建不同的实现，它们都将数据封装成这个抽象事件对象的形式。接下来，我们可以创建一个交互管理器（`Interaction Manager`）对象或类，它负责处理来自不同输入设备的事件。这个交互管理器应该包含一些方法，如 `startInteraction`、`moveInteraction`、`endInteraction` 等，这些方法将接收抽象事件对象作为参数，并执行相应的逻辑。在需要响应交互的元素上，我们添加事件监听器来捕获触屏和鼠标事件。当事件被触发时，我们创建相应的抽象事件对象，并将其传递给交互管理器进行处理。这样，交互管理器就能够根据输入源的不同来执行相应的逻辑。在交互管理器中，我们实现具体的交互逻辑。这些逻辑可能包括更新 UI、发送网络请求、改变应用状态等。由于我们使用了抽象事件对象，因此这些逻辑可以同时适用于触屏和鼠标输入。性能优化：确保事件监听器的添加是高效的，并避免不必要的 DOM 操作。跨浏览器兼容性：测试代码在不同浏览器上的表现，确保兼容性。错误处理：对可能出现的错误进行妥善处理，如事件数据提取失败等。扩展性：设计良好的抽象和接口，以便将来能够轻松地添加对其他输入设备的支持。

## 7.2 项目的运行和测试

本文通过 PC 平台，用 chrome 浏览器打开本阶段上传至 http 服务器的代码，网址（<https://litianpeng123.github.io/1.5.html>），效果图如图 7-2 所示。还可以通过手机平台，用 safari 浏览器打开本阶段上传至 http 服务器的代码。网址（<https://litianpeng123.github.io/1.5.html>），效果图如图 7-3 所示。表明目前项目运行正常，达到了项目初步的目标。移动端可以使用图 7-4 的二维码，查看代码运行效果。

代码的运行效果图：





图 7-2 PC 端运行效果图

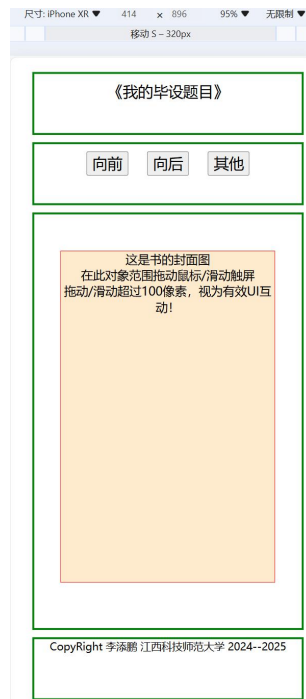


图 7-3 手机端运行效果图



图 7-4 移动端二维码入口

## 7.3 代码提交

本阶段完成的代码存放在 6index.html 和 6myCSS.css 两个文件中，通过 gitbash 工具作正式的代码提交。结果如图 7-5 所示：

```

ltp@LAPTOP-N808IJ3K MINGW64 ~/abc (main)
$ git add 6index.html 6myCSS.css

ltp@LAPTOP-N808IJ3K MINGW64 ~/abc (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 4 commits.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   6index.html
        new file:   6myCSS.css

Untracked files:

```

图 7-5 本次代码提交的文件列表

命令行如下：

```

git add 6index.html 6myCSS.css
git commit -m UI 的个性化键盘交互控制的设计开发

```

本次代码提交成功，结果如图 7-6 所示：

```

ltp@LAPTOP-N808IJ3K MINGW64 ~/abc (main)
$ git commit -m 7.对触屏和鼠标的通用交互操作的设计开发
[main aff7308] 7.对触屏和鼠标的通用交互操作的设计开发
2 files changed, 95 insertions(+)
create mode 100644 6index.html
create mode 100644 6myCSS.css

```

图 7-6 本次代码提交反馈图

gitbash 反馈代码的仓库日志如下所示：

```

commit aff730814f5c4f9005f2a024f53db530ab88b5c5 (HEAD -> main)
Author: 李添鹏 <3578206180@qq.com>
Date: Sun Jun 16 16:08:17 2024 +0800

    7.对触屏和鼠标的通用交互操作的设计开发

```

## 8. UI 的个性化键盘交互控制的设计开发

探索和利用 keydown 和 keyup 这两个键盘底层事件对于为未来的 UI 界面构建强大的键盘功能具有巨大的潜力。这两个事件提供了捕获用户键盘输入的基本机制，使得开发者能够创建出响应键盘操作的丰富交互体验。keydown 事件：当用户按下键盘上的某个键时触发。该事件发生在字符显示在屏幕上之前。在这个



事件中，你可以访问被按下键的键码（keyCode）或者键（key），以及其他相关信息，如 Shift 键或 Alt 键是否同时被按下。keyup 事件：当用户释放键盘上的某个键时触发。这通常用于在用户完成按键操作后执行某些操作，比如确认输入或触发某个动作。

通过监听 keydown 事件，你可以创建自定义的快捷键系统。例如，在一个文本编辑器中，你可以设置 Ctrl+S 为保存文档的快捷键，或者 Ctrl+C 为复制文本的快捷键。当用户按下这些组合键时，你可以执行相应的操作。对于游戏开发者来说，keydown 和 keyup 事件是不可或缺的。通过监听这些事件，你可以捕获用户的键盘输入，并将其映射到游戏角色的动作上。例如，你可以将 W、A、S、D 键映射为角色的移动，将空格键映射为跳跃等。对于需要频繁使用键盘的用户，如程序员或数据录入员，你可以创建一些辅助工具来提高他们的工作效率。例如，你可以创建一个自动补全工具，当用户输入某个字符时，通过监听 keydown 事件并查询数据库或 API 来提供可能的补全选项。对于视觉障碍用户，键盘是他们与 UI 交互的主要方式。通过精心设计的 keydown 和 keyup 事件处理逻辑，你可以创建出更易于键盘访问的 UI 界面。例如，确保所有可交互元素都支持键盘焦点，并提供清晰的键盘导航路径。利用 keydown 和 keyup 事件，你可以为用户提供实时的键盘输入反馈。例如，在一个搜索框中，当用户输入字符时，你可以实时显示匹配的结果或建议。这种实时反馈可以提高用户体验，并帮助用户更快地找到他们需要的信息。

**事件冒泡与捕获：**了解事件如何在 DOM 中传播（冒泡和捕获）对于正确处理 keydown 和 keyup 事件至关重要。你可能需要设置适当的事件监听器来确保在正确的位置捕获到这些事件。

**跨浏览器兼容性：**不同的浏览器可能对 keydown 和 keyup 事件的处理方式有所不同。确保你的代码在所有目标浏览器上都能正常工作。

**性能优化：**虽然 keydown 和 keyup 事件处理通常不会造成严重的性能问题，但如果你需要在大量元素上添加这些事件监听器，或者需要在每次事件触发时执行复杂的操作，那么性能可能会成为一个问题。考虑使用事件委托等技术来优化性能。

**用户体验：**在设计键盘交互时，始终要考虑用户体验。确保你的快捷键和键盘导航逻辑直观易用，并为用户提供清晰的反馈和错误处理机制。因为系统中只有一个键盘，所以我们在部署代码时，把键盘事件的监听设置在 DOM 文档最大的可视对象——body 上，通过测试，不宜把键盘事件注册在 body

内部的子对象中。代码如下所示：

```
$("#body").addEventListener("keydown",function(ev){
    ev.preventDefault() ; //增加“阻止事件对象的默认事件后”，不仅 keypress 事件将不再响应，而且系统的热键，如“F5 刷新页面/Ctrl+R”、“F12 打开开发者面板”等也不再被响应
```

```
    let k = ev.key;
    let c = ev.keyCode;
    $("#keyStatus").textContent = "按下键 : " + k + " , "+ "编码 : " + c;
});
```

```
$("#body").addEventListener("keyup",function(ev){
    ev.preventDefault() ;
    let key = ev.key;
    $("#keyStatus").textContent = key + " 键已弹起" ;
    if (printLetter(key)){
        $("#typeText").textContent += key ;
    }
}
```

```
function printLetter(k){
    if (k.length > 1){ //学生须研究这个逻辑的作用
        return false ;
    }
    let puncs =
['~','`','!','@','#','$','%','^','&','*','(',')','-','_','+','=',' ','.',
',',';','<','>','?','/',' ','\\','\'''] ;
    if ( (k >= 'a' && k <= 'z') || (k >= 'A' && k <= 'Z')
|| (k >= '0' && k <= '9')) {
        console.log("letters") ;
        return true ;
    }
    for (let p of puncs ){
        if (p === k) {
            console.log("puncs") ;
            return true ;
        }
    }
    return false ;
    //提出更高阶的问题，如何处理连续空格和制表键 tab?
} //function printLetter(k)
});
```

## 8.2 项目的运行和测试

本文通过 PC 平台，用 chrome 浏览器打开本阶段上传至 http 服务器的代码，网址（<https://litianpeng123.github.io/1.6.html>），效果图如图 8-2 所示。还可以通过手机平台，用 safari 浏览器打开本阶段上传至 http 服务器的代码。网址（<https://litianpeng123.github.io/1.6.html>），效果图如图 8-3 所示。表明目前项目运行正常，达到了项目初步的目标。移动端可以使用图 8-4 的二维码，查看代码运行效果。代码的运行效果图：



图 8-2 PC 端运行效果图



图 8-3 手机端效果运行图

图 8-4 移动端二维码入口

## 8.3 代码提交

本阶段完成的代码存放在 7index.html 和 7myCSS.css 两个文件中，通过 gitbash 工具作正式的代码提交。结果如图 8-5 所示：

```

ltp@LAPTOP-N808IJ3K MINGW64 ~/abc (main)
$ git add 7index.html 7myCSS.css

ltp@LAPTOP-N808IJ3K MINGW64 ~/abc (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 5 commits.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   7index.html
        new file:   7myCSS.css

```

图 8-5 本次代码提交的文件列表

本次代码提交成功，结果如图 8-6 所示：

```

ltp@LAPTOP-N808IJ3K MINGW64 ~/abc (main)
$ git commit -m 8.UI的个性化键盘交互控制的设计开发
[main f70a4be] 8.UI的个性化键盘交互控制的设计开发
2 files changed, 105 insertions(+)
create mode 100644 7index.html
create mode 100644 7myCSS.css

```

图 8-6 本次代码提交反馈图

gitbash 反馈代码的仓库日志如下所示：

```

commit f70a4be5ed3cf554e3710261c07516afbf330dfb (HEAD -> main)
Author: 李添鹏 <3578206180@qq.com>
Date:   Sun Jun 16 16:17:27 2024 +0800

    8.UI的个性化键盘交互控制的设计开发

```

## 9. 谈谈本项目中的高质量代码

这是一本关于计算机教学的书。今天，电脑就像螺丝刀一样常见，但它们要复杂得多，让它们做你想让它们做的事情并不总是那么容易。如果你给你的电脑的任务是一个常见的，很容易理解的任务，比如显示你的电子邮件或像计算器一样工作，你可以打开适当的应用程序并开始工作。但对于独特的或开放式的任务，可能没有应用程序。

这就是编程可能发挥作用的地方。编程是构建程序的行为——一组告诉计算机该做什么的精确指令。因为计算机是愚蠢的、迂腐的野兽，编程从根本上来说

是乏味和令人沮丧的。幸运的是，如果你能克服这个事实，甚至享受用愚蠢的机器可以处理的方式思考的严谨性，编程是有回报的。它可以让你在几秒钟内完成手工永远做不完的事情。它是一种让你的计算机工具做它以前不能做的事情的方法。它还提供了一个很好的抽象思维练习。[6]

创建一个 **Pointer** 对象，践行 MVC 设计模式，设计一套代码同时对鼠标和触屏实现控制。面向对象思想，封装，抽象，局部变量，函数式编程，逻辑。  
(围绕着抽象定义函数、代码块、模型设计以及降低全局变量的使用来写)

## 10. 用 gitBash 工具管理项目的代码仓库和 http 服务器

### 10.1 经典 Bash 工具介绍

当我们谈到命令行时，我们实际上指的是 shell。shell 是一个接受键盘命令并将其传递给操作系统执行的程序。几乎所有的 Linux 发行版都提供了一个来自 GNU 项目的 shell 程序，名为 bash。这个名字是 Bourne -again shell 的首字母缩略词，指的是 bash 是 sh 的增强替代品，sh 是 Steve Bourne 编写的原始 Unix shell 程序。[7]

像 Windows 一样，像 Linux 这样的类 unix 操作系统用所谓的分层目录结构来组织文件。这意味着它们被组织成树状的目录模式(在其他系统中有时称为文件夹)，其中可能包含文件和其他目录。文件系统中的第一个目录称为根目录。根目录包含文件和子目录，子目录包含更多的文件和子目录，以此类推。[7]

### 10.2 通过 gitHub 平台实现本项目的全球域名

### 10.3 创建一个空的远程代码仓库

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (\*).

Owner \*  / Repository name \*

✔ litianpeng123456.github.io is available.

Great repository names are short and memorable. Need inspiration? How about [sturdy-octo-happiness](#) ?

Create repository

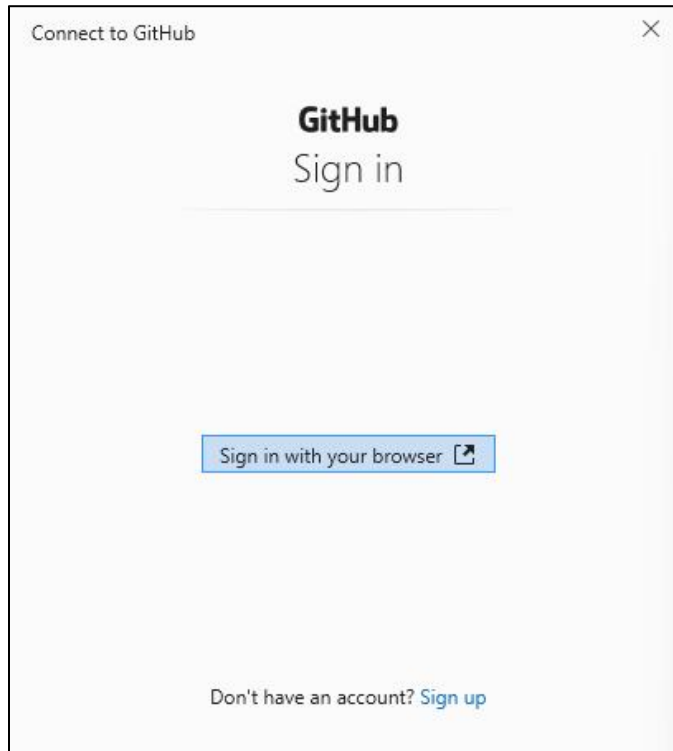
点击窗口右下角的绿色“Create repository”，则可创建一个空的远程代码仓库。

## 10.4 设置本地仓库和远程代码仓库的连接

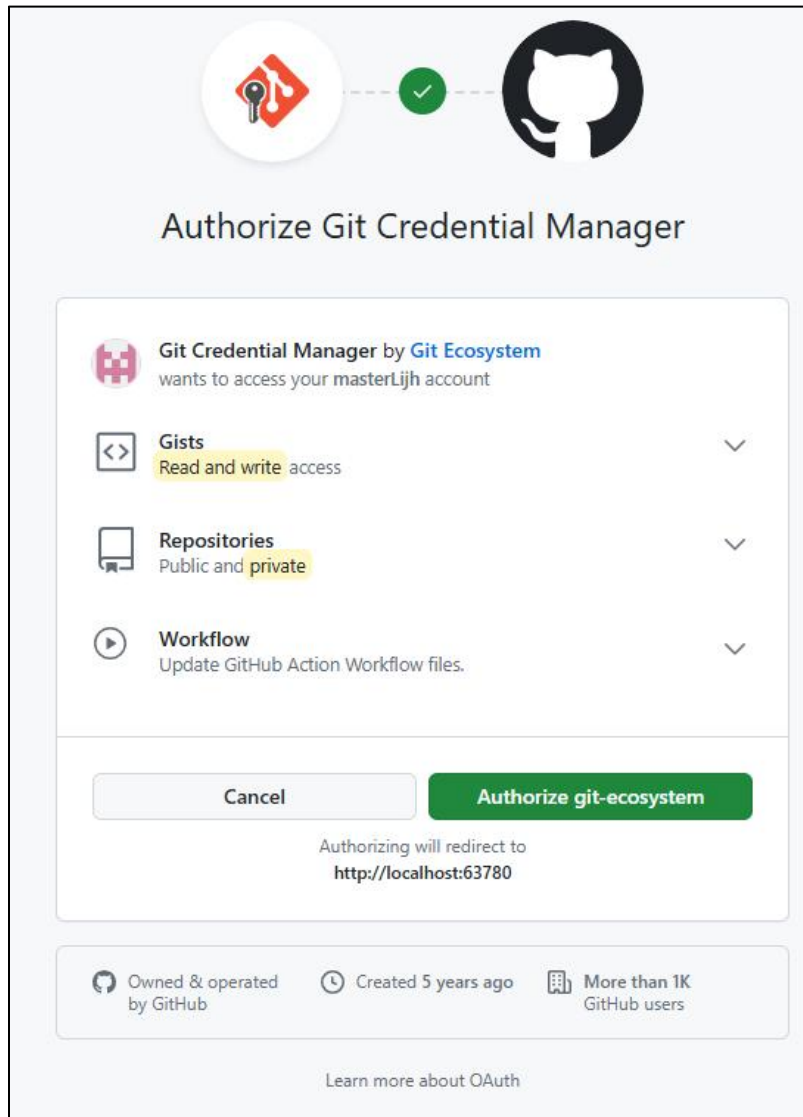
进入本地 webUI 项目的文件夹后，通过下面的命令把本地代码仓库与远程建立密钥链接

```
$ echo "WebUI 应用的远程 http 服务器设置" >> README.md
$ git init
$ git add README.md
$ git commit -m "这是我第一次把代码仓库上传至 gitHub 平台"
$ git branch -M main
$ git remote add origin
    https://github.com/litianpeng123/litianpeng123456.github.io.git
$ git push -u origin main
```

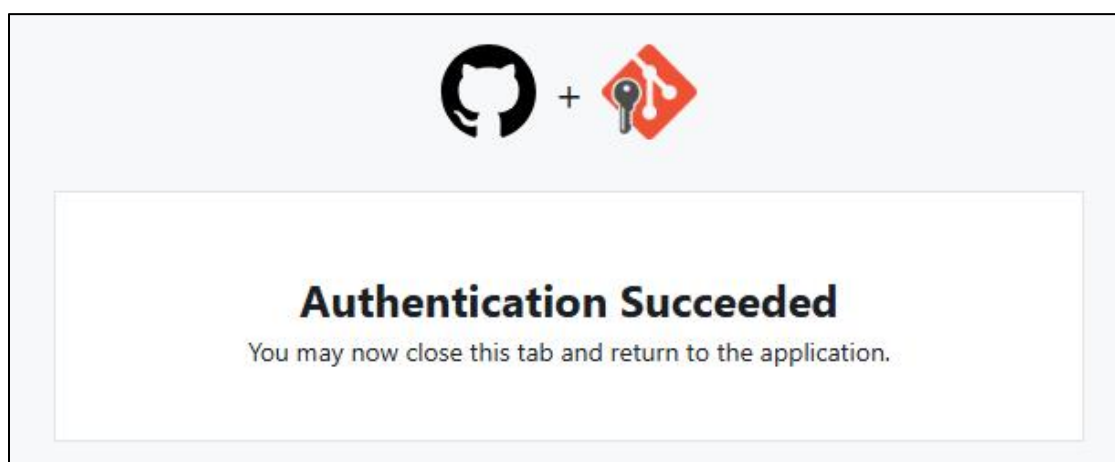
本项目使用 window 平台，gitbash 通过默认浏览器实现密钥生成和记录，第一次链接会要求开发者授权，如下图所示：



再次确认授权 `gitBash` 拥有访问改动远程代码的权限，如下图所示：



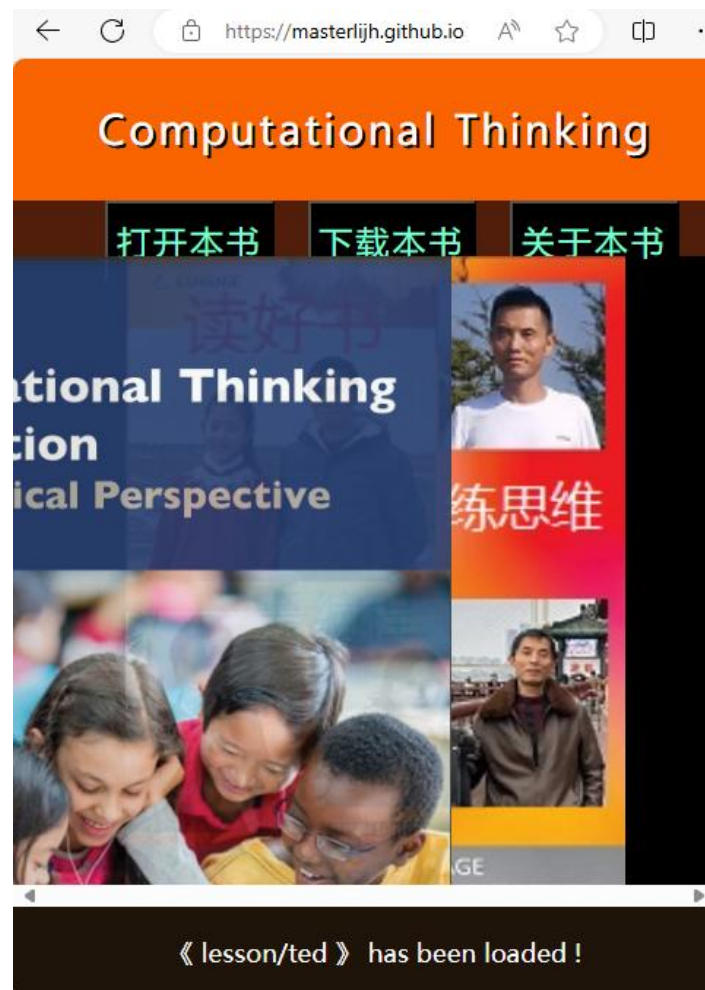
最后，GitHub 平台反馈：gitBash 和 gitHub 平台成功实现远程链接。



从此，我们无论在本地做了任何多次代码修改，也无论提交了多少次，上传远程时都会把这些代码和修改的历史记录全部上传 github 平台，而远程上传命令则可简化为一条：git push，极大地方便了本 Web 应用的互联网发布。



远程代码上传后，项目可以说免费便捷地实现了在互联网的部署，用户可以通过域名或二维码打开，本次使用 PC 的微软 Edge 浏览器打开，本文截取操作中间的效果图，如下所示：



全文完成，谢谢！

## 参考文献:

- [1] W3C. W3C's history. W3C Community. [EB/OL]. <https://www.w3.org/about/>.  
<https://www.w3.org/about/history/>. 2023.12.20
- [2] Douglas E. Comer. The Internet Book [M] (Fifth Edition). CRC Press Taylor & Francis Group, 2019: 217-218
- [3] John Dean, PhD. Web programming with HTML5, CSS, and JavaScript[M]. Jones & Bartlett Learning, LLC. 2019: 2
- [4] John Dean, PhD. Web programming with HTML5, CSS, and JavaScript[M]. Jones & Bartlett Learning, LLC. 2019: xi
- [5] Behrouz Forouzan. Foundations of Computer Science[M](4th Edition). Cengage Learning EMEA, 2018: 274--275
- [6] Marijn Haverbeke. Eloquent JavaScript 3rd edition. No Starch Press, Inc, 2019.
- [7] William Shotts. The Linux Command Line, 2nd Edition [M]. No Starch Press, Inc, 245 8th Street, San Francisco, CA 94103, 2019: 3-7

## 写作指导:

实质上任何章节都可以按三段论模式写，第一段是写研究的背景和目标，第二段是写你所开展的工作步骤和工作量，第三段是用了哪些方法和工作的结果或意义。

**【摘要案例 1】**近十年来，html5 为核心的 web 标准的软件开发技术以其跨平台、开源的优势广泛地运用在各个领域的应用软件开发中。通过分析本次毕设任务，本项目选择 html5 的 web 客户端技术为技术路线，展开对程序设计和软件开发的研究和实践。通过广泛查阅相关技术书籍、开发者论坛和文献，设计开发了一个个性化的用户界面（UI）的应用程序。在开发中综合应用了 html 语言进行内容建模、css 语言展开 UI 的外观设计、javascript 语言编程实现 UI 的交互功能，除直接使用了 web 客户端最底层的 API 外，本项目的每条代码都是手工逐条编写，没有导入他人的任何的代码（框架和库）。本项目也采用了响应式设计编程，可以智能地适应移动互联网时代用户屏幕多样化的需要；另外大量地运用了面向对象的程序设计思想，比如用代码构建了一个通用的 pointer 模型，该模型仅用一套代码就实现了对鼠标和触屏的控制，实现了高质量的代码，这也是本项目的亮点。从工程管理的角度看，

本项目采用的增量式开发模式，以逐步求精的方式展开了六次代码的增量式重构（A:Analysis, D:Design, I: Implementation, T:Testing），比较愉快地实现项目的设计开发和测试。从代码的开源和分享的角度看，本项目采用了 git 工具进行版本管理，在漫长的开发过程中重构代码六次并正式做了代码提交，另外在测试中修改提交了代码两次，最后利用 gitbash 工具 把本项目的代码仓库上传到著名的 github 上，再利用 github 提供的 http 服务器，本项目实现了 UI 应用在全球互联网的部署，我们可以通过地址和二维码便捷地跨平台高效访问这个程序。

**【摘要案例 2】：**Web 技术以其跨操作系统平台的优势成为了广泛流行的软件开发手段，为了适应移动互联网时代软件项目的前端需求，本项目以 Web 客户端技术为研究学习内容，广泛查阅了技术资料与相关文献，尤其是 mozilla 组织的 MDN 社区的技术实践文章，探索了 HTML 内容建模、CSS 样式设计和 JavaScript 功能编程的基本技术和技巧。通过集成上述技术，再应用本科的相关课程的知识，实现了一个个性化的用户界面（UI: uer interface）的项目，该用户界面以响应式技术为支撑做到了最佳适配用户屏幕，程序可以动态适用于当前 PC 端和移动设备；在功能上以 DOM 技术和事件驱动模式的程序为支撑实现了对鼠标、触屏、键盘的底层事件响应和流畅支持，为鼠标和触屏设计了一个对象模型，用代码实现了对这类指向性设备的模拟（这是本项目模型研究法的一次创新实践，也是本项目的亮点。）。为了处理好设计和开发的关系，项目用了工程思想管理，使用了软件工程的增量式开发模式，共做了 6 次项目迭代开发，每次迭代都经历了开发 4 个经典开发阶段（A:Analysis, D:Design, I: Implementation, T:Testing），以逐步求精的方式编写了本 UI 的应用程序。为了分享和共享本代码，与网上的开发者共同合作，本项目还使用了 git 工具进行代码和开发过程日志记录，一共做了 12 次提交代码的操作，详细记录和展现了开发思路和代码优化的过程，最后通过 gitbash 把项目上传到 github 上，建立了自己的代码仓库，并将该代码仓库设置成为了 http 服务器，实现了本 UI 应用的全球便捷访问。