

成绩：

江西科技师范大学

毕业设计（论文）

题目（中文）：基于 Web 客户端技术的个性化 UI 设计和实现

（外文）：Web client based customized UI design and Programming

院（系）：元宇宙产业学院

专 业：计算机科学与技术

学生姓名：孔卓

学 号：20213630

指导教师：李健宏

2024 年 6 月 27 日

基于 Web 客户端技术的个性化 UI 的设计和实现.....	4
（Customized UI design and Programming based on Web client technology）	4
Abstract.....	5
1. 前言.....	6
1.1 毕设任务分析.....	6
1.2 研学计划.....	6
1.3 研究方法.....	7
2. 技术总结和文献综述.....	8
2.1 Web 平台和客户端技术概述.....	8
2.2 项目的增量式迭代开发模式.....	9
3. 内容设计概要.....	11
3.1 分析和设计.....	11
3.2 项目的实现和编程.....	12
3.3 项目的运行和测试.....	13
3.4 项目的代码提交和版本管理.....	13
4. 移动互联时代的 UI 开发初步——窄屏终端的响应式设计	14
4.1 分析和设计.....	14
4.2 项目的实现和编程.....	15
4.3 项目的运行和测试.....	16
4.4 项目的代码提交和版本管理.....	17
5. 应用响应式设计技术开发可适配窄屏和宽屏的 UI.....	18
5.1 分析和设计.....	18
5.2 项目的实现和编程.....	19
5.3 项目的运行和测试.....	20
5.4 项目的代码提交和版本管理.....	21
6. 个性化 UI 设计中对鼠标交互的设计开发	22
6.1 分析和设计.....	22
6.2 项目的实现和编程.....	22
6.3 项目的运行和测试.....	24
6.4 项目的代码提交和版本管理.....	24
7. 对触屏和鼠标的通用交互操作的设计开发.....	25
7.1 分析和设计.....	25
7.2 项目的实现和编程.....	25
7.3 项目的运行和测试.....	27
7.4 项目的代码提交和版本管理.....	28
8. UI 的个性化键盘交互控制的设计开发	29
8.1 分析和设计.....	29
8.2 项目的实现和编程.....	29
8.3 项目的运行和测试.....	31
8.4 项目的代码提交和版本管理.....	31
9. 谈谈本项目中的高质量代码.....	32
10. 用 gitBash 工具管理项目代码仓库和 http 服务器	34

10.1 经典 Bash 工具介绍.....	34
10.2 通过 gitHub 平台实现本项目的全球域名	35
10.3 创建一个空的远程代码仓库.....	36
10.4 设置本地仓库和远程代码仓库的链接.....	36
参考文献:	40

基于 Web 客户端技术的个性化 UI 的设计和实现

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

科师大元宇宙产业学院 2021 级 孔卓

摘要: 近十年来, 以 HTML5 为核心的 Web 标准软件开发技术凭借其跨平台、开源的优势, 广泛应用于各个领域的应用软件开发中。本项目选择 HTML5 的 Web 客户端技术为技术路线, 开展程序设计和软件开发的研究与实践。通过广泛查阅相关技术书籍、开发者论坛和文献, 特别是 Mozilla 组织的 MDN 社区的技术实践文章, 项目设计并开发了一个个性化的用户界面(UI) 应用程序。在开发过程中, 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 语言实现 UI 的交互功能。项目采用响应式设计编程, 能够智能适应移动互联网时代用户屏幕多样化的需求。为确保设计和开发的有序进行, 项目采用了工程管理的增量式开发模式 (A: Analysis, D: Design, I: Implementation, T: Testing), 通过逐步求精的方式进行了六次代码增量式重构。每次迭代都包含了分析、设计、实现和测试四个经典开发阶段, 愉快地实现了项目的设计开发和测试目标。在代码管理和分享方面, 项目使用了 Git 工具进行版本管理, 在漫长的开发过程中共进行了 12 次代码提交, 详细记录了开发思路 and 代码优化过程。最终, 通过 Gitbash 工具将项目的代码仓库上传至 GitHub, 建立了自己的代码仓库, 并利用 GitHub 提供的 HTTP 服务器, 实现了 UI 应用在全球互联网的部署, 使我们可以通过地址和二维码便捷地跨平台访问该程序。通过上述研究和实践, 项目探索了 HTML5、CSS 和 JavaScript 的基本技术和技巧, 结合课程知识, 实现了一个动态适应 PC 端和移动端设备的个性化用户界面。这不仅展示了现代 Web 技术的优势, 还体现了软件工程的最佳实践和代码分享的理念, 增强了我对软件工程的理

关键词: HTML5; 增量式开发模式; Git

Abstract

In the past ten years, the Web standard software development technology with HTML5 as the core has been widely used in application software development in various fields by virtue of its cross-platform and open source advantages. This project chooses HTML5 Web client technology as the technical route to carry out the research and practice of program design and software development .. Through extensive access to technical books, developer forums, and literature, especially technical practice articles from the MDN community organized by Mozilla, the project designed and developed a personalized user interface (UI) application. In the development process, the comprehensive application of HTML language for content modeling, CSS language to expand the appearance of UI design, JavaScript language to achieve the interactive function of UI. The project adopts responsive design and programming, which can intelligently adapt to the diversified needs of users' screens in the era of mobile Internet. (A: Analysis, D: Design, I: Implementation, T: Testing) In order to ensure the orderly progress of Design and development, the project adopts the incremental development mode of engineering management (A: Analysis, D: Design, I: Implementation, T: Testing), and carries out six incremental code reconfiguration through the way of gradual refinement. Each iteration consists of the four classic development phases of analysis, design, implementation, and testing, happily achieving the design development and testing goals of the project.. In terms of code management and sharing, Git tool was used for version management, and a total of 12 code submissions were made during the long development process, which recorded the development ideas and code optimization process in detail. Finally, we uploaded the project's code warehouse to GitHub through Gitbash tool, established our own code warehouse, and utilized the HTTP server provided by GitHub to realize the deployment of UI application on the global Internet, so that we could easily access the program across platforms through address and two-dimensional code. Through the above research and practice, the project explores the basic technologies and techniques of HTML5, CSS and JavaScript, and combines the course knowledge to achieve a personalized user interface that dynamically ADAPTS to both PC and mobile devices. This not only shows the advantages of modern Web technology, but also reflects the best practices of software engineering and the idea of code sharing, which enhances my understanding of software engineering.

Key words: HTML5;Incremental development model; Git

1. 前言

1.1 毕设任务分析

作为计算机科学技术专业的本科生，毕业设计不仅是对我们学术生涯的总结，也是展示我们掌握核心课程知识和技能的机会。在这个项目中，我们将应用面向对象的程序设计语言、数据结构和算法、操作系统以及软件工程等核心课程的理论知识。这些知识虽然在课程学习中显得抽象，但通过具体的项目实践，可以帮助我们深刻理解和灵活应用这些理论。

我选择了以 **HTML5** 为核心的 **Web** 技术来进行毕业设计，这是因为其跨平台、开源的特性，能够在现代软件开发中得到广泛应用。项目的目标是开发一个个性化的用户界面（**UI**）应用程序，该应用程序不仅需要实现基本的交互功能，还需要能够适应不同的设备和屏幕尺寸，展示我对响应式设计的理解和应用。此外，通过面向对象的设计思想，我将实现一个通用的 **Pointer** 模型，用一套代码同时控制鼠标和触屏，体现高质量代码的编写能力。

1.2 研学计划

1.2.1 调研与学习

广泛查阅相关技术书籍、开发者论坛和文献，特别是 **Mozilla** 组织的 **MDN** 社区的技术实践文章，学习 **HTML5**、**CSS** 和 **JavaScript** 的最新技术和最佳实践；回顾核心课程的理论知识，特别是面向对象的程序设计语言、数据结构和算法、操作系统、软件工程等，确保能够将这些理论应用于实际项目中。

1.2.2 项目设计与规划

在明确项目的功能需求和技术要求后，我们将进行系统设计，包括系统架构的规划和模块划分。设计阶段的重点是确保系统的高内聚和低耦合，采用面向对象的设计思想进行模块化设计，定义清晰的类和接口规范。

1.2.3 项目开发

项目开发采用增量式开发模式（**A: Analysis, D: Design, I: Implementation, T: Testing**），分阶段进行：

需求分析与设计：每个迭代周期开始时进行详细的需求分析和设计，确保对每个功能模块的清晰理解。

功能实现与测试：在每个迭代周期中实现具体功能，并进行充分的单元测试

和集成测试，确保代码的正确性和稳定性。

代码优化与重构：在每个迭代结束时进行代码的优化和重构，提高系统性能和代码质量。

在功能实现过程中，综合应用 **HTML** 进行内容建模，**CSS** 进行 **UI** 设计，**JavaScript** 实现交互功能。特别是构建通用的 **Pointer** 模型，用一套代码同时控制鼠标和触屏，展示面向对象编程的应用。

1.2.4 版本控制与协作

使用 **Git** 工具进行版本管理，详细记录每次代码提交和修改历史，确保代码的可追溯性和协作开发的效率。通过 **GitHub** 平台实现代码的开源和共享，并利用 **GitHub** 提供的 **HTTP** 服务器进行项目的在线部署和测试。

通过以上研学计划，我将理论学习与实践操作相结合，全面提升对核心课程知识的理解和应用能力，同时掌握最新的 **Web** 技术，展现我作为计算机科学与技术专业本科生的专业素养和能力。

1.3 研究方法

理论与实践相结合：在项目的设计和开发过程中，我们将理论学习和实际操作相结合。通过回顾核心课程的理论知识，并在项目中实际应用，增强对这些理论的理解和掌握。

面向对象的设计思想：项目开发过程中，我们将运用面向对象的设计思想，进行模块化设计，确保代码的高内聚和低耦合。通过定义类和对象，实现代码的封装、继承和多态性，提升代码的可维护性和扩展性。

增量式开发模式：采用增量式开发模式（**A: Analysis, D: Design, I: Implementation, T: Testing**），每次迭代都包括需求分析、系统设计、功能实现和测试优化四个阶段。通过逐步求精的方式，不断完善和优化项目，确保项目的高质量。

版本控制：使用 **Git** 工具进行版本管理，详细记录开发过程中的每次代码提交和修改历史，确保代码的可追溯性和协作开发的效率。最终通过 **GitHub** 平台，实现代码的开源和共享。

响应式设计：应用响应式设计技术，确保应用程序能够适应不同设备和屏幕尺寸。通过 **CSS** 媒体查询和灵活的布局设计，实现用户界面的自适应，提升用户体验。

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^[1]。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 History

In 1989, Sir Tim Berners-Lee invented the World Wide Web (see the original proposal). He coined the term "World Wide Web," wrote the first World Wide Web server, "httpd," and the first client program (a browser and editor), "WorldWideWeb," in October 1990.

He wrote the first version of the "HyperText Markup Language" (HTML), the document formatting language with the capability for hypertext links that became the primary publishing format for the Web. His initial specifications for URIs, HTTP, and HTML were refined and discussed in larger circles as Web technology spread.

2.2.2 A Consortium for the World Wide Web

In 1994, the decision to form the World Wide Web Consortium came at the urging of many companies investing increasing resources into the web. Sir Tim Berners-Lee started leading the essential work of the Web Consortium team to foster a consistent architecture accommodating the rapid pace of progress in web standards for building websites, browsers, devices to experience all that the web has to offer.

In founding the World Wide Web Consortium, Sir Tim Berners-Lee created a community of peers. Web technologies were already moving so quickly that it was critical to assemble a single organization to coordinate web standards. Tim accepted the offer from MIT, who had experience with consortia, to host W3C. He required from the start that W3C have a global footprint.

2.2.3 Web platform and Web Programming

Let's start with a brief description of the Web, which is short for World Wide Web. Most people say "Web" instead of "World Wide Web," and we'll follow that convention. The Web is a collection of documents, called web pages, that are shared (for the most part) by computer users throughout the world. Different types of web pages

do different things, but at a minimum, they all display content on computer screens. By “content,” we mean text, pictures, and user input mechanisms like text boxes and buttons.^[2]

Web programming is a large field, with different types of web programming implemented by different tools. All the tools work with the core language, HTML, so almost all the web programming books describe HTML to some extent. This textbook covers HTML5, CSS, and JavaScript, all in depth. Those three technologies are known to be the pillars of client-side web programming. With client side web programming, all web page calculations are performed on end users’ computers (the client computers).^[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）^[3]；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。

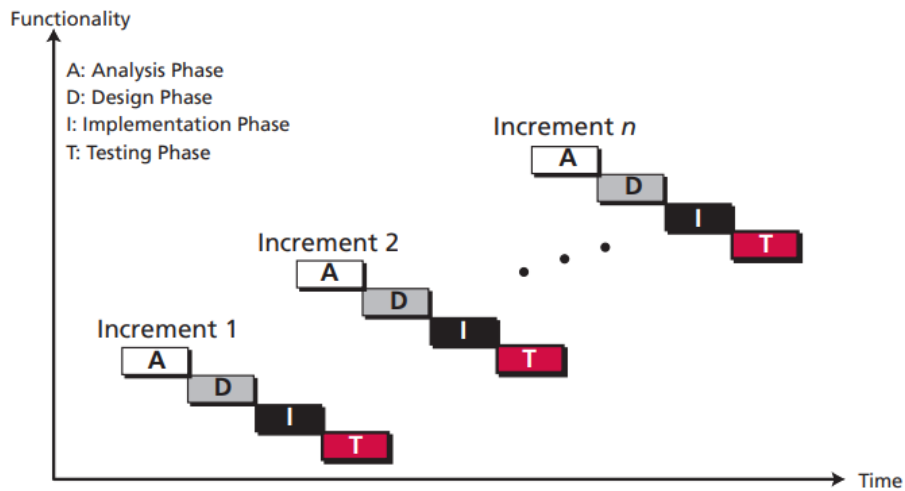
2.2 项目的增量式迭代开发模式

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^[5]。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:

The incremental model



The incremental model

In the incremental model, software is developed in a series of steps. The developers first complete a simplified version of the whole system. This version represents the entire system but does not include the details. Figure shows the incremental model concept. In the second version, more details are added, while some are left unfinished, and the system is tested again. If there is a problem, the developers know that the problem is with the new functionality. They do not add more functionality until the existing system works properly. This process continues until all required functionality has been added. [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 3-1 用例图所示：

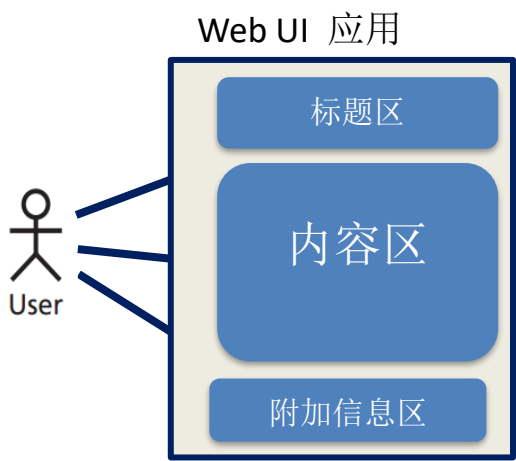
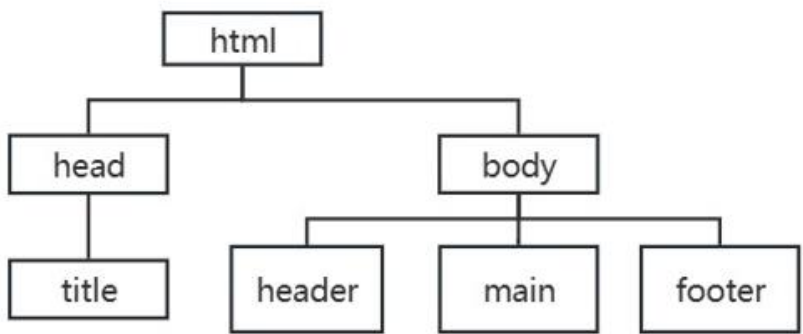


图 3-1 用例图



3.2 项目的实现和编程

一、HTML 代码编写如下：

```
<header>
    《 基于 Web 客户端技术的个性化 UI 设计和实现 》
</header>
<main>
    我的主题内容： ‘读好书、练思维、勤编程’ @kongzhuoz
</main>
<footer>
    Copyright 孔卓 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 3-2 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 3-3 的二维码，运行测试本项目的第一次开发的阶段性效果。

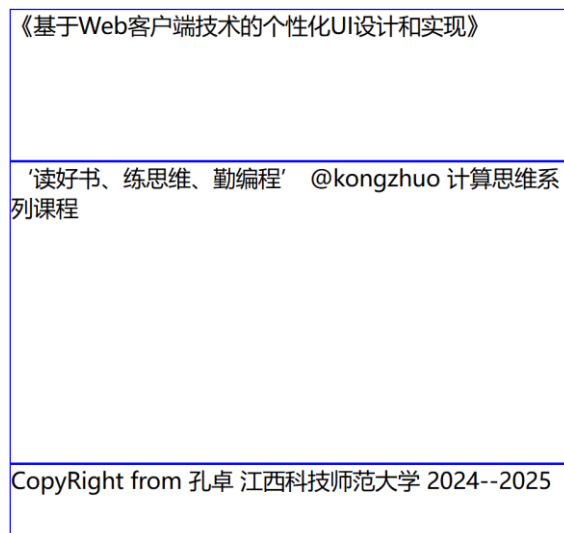


图 3-2 PC 端运行效果图



图 3-3 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 孔卓
```

```
$ git config user.email 751590281@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：

```
$ git commit -m 项目第一版：“三段论”式的内容设计概要开发
[master (root-commit) 32de024] 项目第一版：“三段论”式的内容设计概要开发
2 files changed, 46 insertions(+)
create mode 100644 index.html
create mode 100644 myCss.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
```

git bash 反馈代码的仓库日志如下所示：

```
commit 26e83d558289999ef9faa3c50f8cd982f0669fca
Author: 孔卓 <751590281@qq.com>
Date: Sun Jun 2 21:58:43 2024 +0800

    项目第一版：“三段论”式的内容设计概要开发
```

4. 移动互联时代的 UI 开发初步——窄屏终端的响应式设计

4.1 分析和设计

移动互联时代的 UI 开发必须考虑到用户在各种设备上的使用体验，尤其是针对窄屏终端的响应式设计。窄屏终端通常指的是智能手机和平板电脑等小尺寸屏幕设备，设计师需要特别关注这些设备上的布局 and 交互方式，以确保用户能够方便地访问和操作应用程序或网站用例图和 dom 图如下图 4.1。

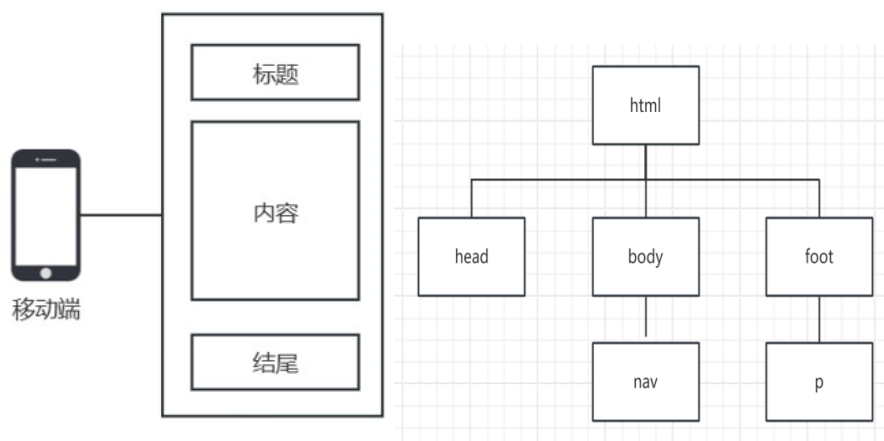


图 4.1

4.2 项目的实现和编程

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

4.3 项目的运行和测试

本文此处给出 PC 端用 Edge 浏览器打开项目的结果，如下图 4-3 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二维码，运行测试本项目的第一次开发的阶段性效果。



图 4.3

4.4 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，作为项目的第二次迭代，在代码提交和版本管理环节，我们的目标是提交本次版本代码并增加注释。

进入 gitBash 命令行后，按次序输入以下命令：

```
cd /e/Code/webUI
Git add 1.2.html
Git commit -m "第四章版本"
```

截图图下所示：

```
Minglizzz@LAPTOP-LS17EVI4 MINGW64 /e/Code/Git/webUI (master)
$ Git add 1.2.html

Minglizzz@LAPTOP-LS17EVI4 MINGW64 /e/Code/Git/webUI (master)
$ git commit -m "第四章版本"
[master 4307a95] 第四章版本
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 1.2.html

Minglizzz@LAPTOP-LS17EVI4 MINGW64 /e/Code/Git/webUI (master)
$ git log
commit 4307a95ab572f821bba463253d11d773cc900223 (HEAD -> master)
Author: 孔卓 <751590281@qq.com>
Date: Wed Jun 12 20:26:54 2024 +0800
```

第四章版本

5.应用响应式设计技术开发可适配窄屏和宽屏的 UI

5.1 分析和设计

经过了上一章对响应式技术开发适配移动端设备的网页的设计经验后，这一章我们通过更高级的代码来实现更多的功能，阐述移动互联时代的用户终端的多样性，以及如何用 `css` 语言和 `JavaScript` 语言实现响应式设计，用例图和 `dom` 图如下图 5.1。

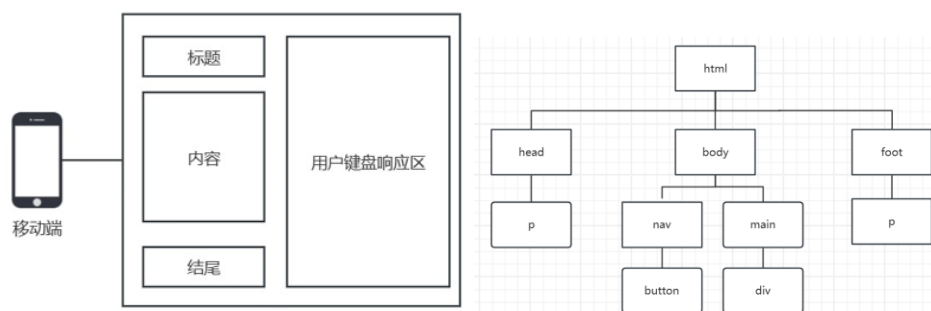


图 5.1

5.2 项目的实现和编程

页面的整体布局如代码块 5.2 所示

```
<header>
  <p id="book">
    《现代电影赏析》
  </p>
</header>
<nav>
  <button>向前</button>
  <button>向后</button>
  <button>暂停</button>
</nav>

  <main id="main">
    <div id="bookface">
      书的封面图
    </div>
  </main>

<footer>

  Copyright from 孔卓 江西科技师范大学 2022--2025

</footer>
```

代码块 5.2

其次，在本次代码中，首次尝试了对鼠标的设计 UI 控制，如下代码块 5.3。

```

//尝试对鼠标设计 UI 控制
var mouse={};
mouse.isDown= false;
mouse.x= 0;
mouse.deltaX=0;
$("#bookface").addEventListener("mousedown",function(ev){
    let x= ev.pageX;
    let y= ev.pageY;

    console.log("鼠标按下了，坐标为: "+"("+x+", "+y+")");
    $("#bookface").textContent= "鼠标按下了，坐标为: "+"("+x+", "+y+")";
});
$("#bookface").addEventListener("mousemove",function(ev){
    let x= ev.pageX;
    let y= ev.pageY;

    console.log("鼠标正在移动，坐标为: "+"("+x+", "+y+")");
    $("#bookface").textContent= "鼠标正在移动，坐标为: "+"("+x+", "+y+")";
});
$("#bookface").addEventListener("mouseout",function(ev){
    //console.log(ev);
    $("#bookface").textContent="鼠标已经离开";
});
$("#body").addEventListener("keypress",function(ev){
    let k=ev.key;
    let c=ev.keyCode;
    let s1="按键是: ";
    let s2="编码是: ";
    $("#keyboard").textContent=s1+k+" "+s2+c;
});

```

代码块 5.3

5.3 项目的运行和测试

本文此处给出 PC 端用 Edge 浏览器打开项目的结果，如下图 5-4 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二维码，运行测试本项目的本次开发的阶段性效果。



图 5.4

5.4 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，作为项目的第三次迭代，在代码提交和版本管理环节，我们的目标是将本次代码上传到 github 仓库，并进行本次提交的信息备注。

进入 gitBash 命令行后，按次序输入以下命令：

```
cd /e/Code/webUI  
Git add 1.3.html  
Git commit -m "增加全部文件"
```

截图图下所示：

```
Minglizzz@LAPTOP-LS17EVI4 MINGW64 /e/Code/Git/webUI (master)
$ Git add 1.3.html

Minglizzz@LAPTOP-LS17EVI4 MINGW64 /e/Code/Git/webUI (master)
$ git commit -m "增加全部文件"
[master 5b2414f] 增加全部文件
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 1.3.html
```

6. 个性化 UI 设计中对鼠标交互的设计开发

6.1 分析和设计

经过了上一章对响应式技术开发适配移动端设备的网页的设计经验后，这一章我们通过更高级的代码来实现对鼠标交互功能的初步开发，阐述移动互联时代的用户终端的多样，用例图和 dom 图如下图 6.1。

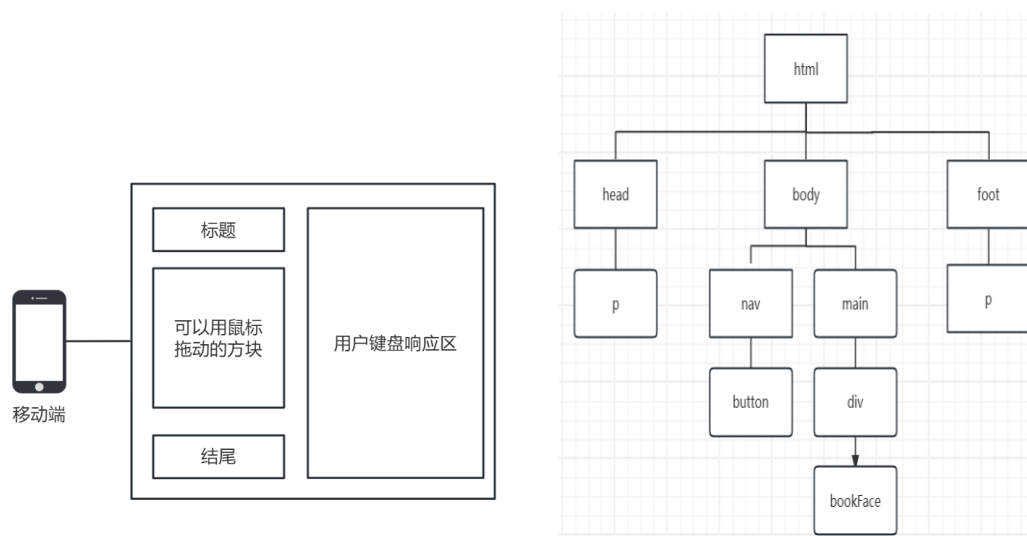


图 6.1

6.2 项目的实现和编程

主要实现功能的代码块如下代码块 6.2.

```
var mouse={};
mouse.isDown= false;
mouse.x= 0;
```

```

mouse.y= 0;
mouse.deltaX=0;
$("#bookface").addEventListener("mousedown",function(ev){
    mouse.isDown=true;
    mouse.x= ev.pageX;
    mouse.y= ev.pageY;
    console.log("mouseDown at x: "+"("+mouse.x +"," +mouse.y +")" );
    $("#bookface").textContent= "鼠标按下，坐标： "+"("+mouse.x+","+mouse.y+")";
});
$("#bookface").addEventListener("mouseup",function(ev){
    mouse.isDown=false;

    $("#bookface").textContent= "鼠标松开!";
    if(Math.abs(mouse.deltaX) > 100){
        $("#bookface").textContent += "，这是有效拖动！ " ;
    }else{
        $("#bookface").textContent += " 本次算无效拖动！ " ;
        $("#bookface").style.left = '7%';
    }
});
$("#bookface").addEventListener("mouseout",function(ev){
    ev.preventDefault();
    mouse.isDown=false;

    $("#bookface").textContent= "鼠标松开!";
    if(Math.abs(mouse.deltaX) > 100){
        $("#bookface").textContent += " 这次是有效拖动！ " ;
    }else{
        $("#bookface").textContent += " 本次算无效拖动！ " ;
        $("#bookface").style.left = '7%';
    }
});
$("#bookface").addEventListener("mousemove",function(ev){
    ev.preventDefault();
    if (mouse.isDown){
        console.log("mouse isDown and moving");
        mouse.deltaX = parseInt( ev.pageX - mouse.x );
        $("#bookface").textContent= "正在拖动鼠标，距离： " + mouse.deltaX +"px 。 ";
        $('#bookface').style.left = mouse.deltaX + 'px' ;
    }
}

```

6.3 项目的运行和测试

本文此处给出 PC 端用 Edge 浏览器打开项目的结果，如下图 6-3 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二维码，运行测试本项目的本次开发的阶段性效果。



图 6.3

6.4 项目的代码提交和版本管理

```
cd /e/Code/webUI
Git add 1.4.html
Git commit -m "提交关于鼠标交互的设计开发版本"
```


截图图下所示:

```
Minglizzz@LAPTOP-LS17EVI4 MINGW64 /e/Code/Git/webUI (master)
$ Git add 1.4.html

Minglizzz@LAPTOP-LS17EVI4 MINGW64 /e/Code/Git/webUI (master)
$ git commit -m "提交关于鼠标交互的设计开发版本"
[master 3d648c1] 提交关于鼠标交互的设计开发版本
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 1.4.html
```

7.对触屏和鼠标的通用交互操作的设计开发

7.1 分析和设计

经过了上一章对鼠标交互设计网页的设计经验后,这一章我们通过更高级的代码来实现对触屏和鼠标功能的通用交互操作初步开发,阐述移动互联时代的用户终端的多样,用例图和 dom 图如下图 7.1。

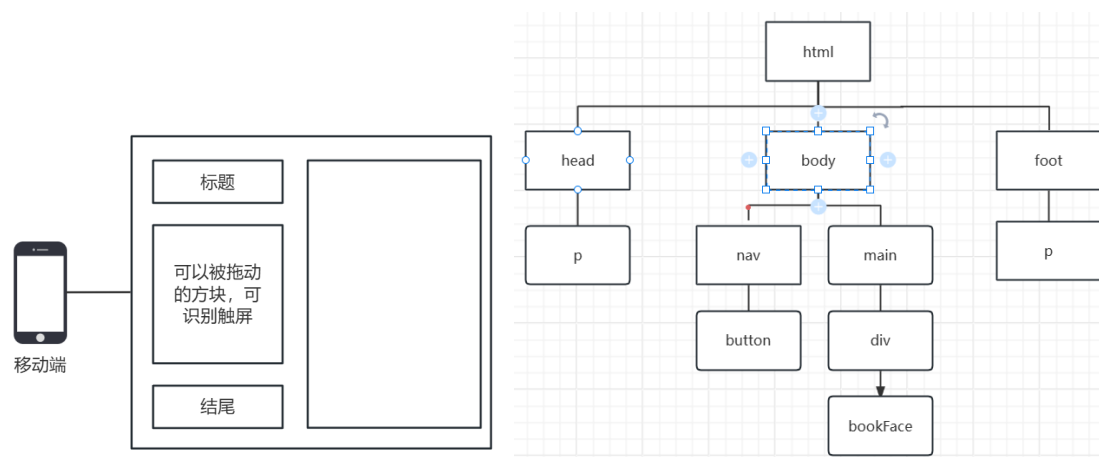


图 7.1

7.2 项目的实现和编程

主要实现功能的代码块如下代码块 7.2.

```
//尝试对鼠标和触屏设计一套代码实现 UI 控制
var Pointer = {};
  Pointer.isDown= false;
  Pointer.x = 0;
  Pointer.deltaX =0;
  { //Code Block begin
```

```

let handleBegin = function(ev){
  Pointer.isDown=true;

  if(ev.touches){console.log("touches1"+ev.touches);
    Pointer.x = ev.touches[0].pageX ;
    Pointer.y = ev.touches[0].pageY ;
    console.log("Touch begin : "+"("+Pointer.x +"," +Pointer.y +")" ) ;
    $("bookface").textContent= "触屏事件开始， 坐标: "+"("+Pointer.x+","+Pointer.y+")";
  }else{
    Pointer.x= ev.pageX;
    Pointer.y= ev.pageY;
    console.log("PointerDown at x: "+"("+Pointer.x +"," +Pointer.y +")" ) ;
    $("bookface").textContent= "鼠标按下， 坐标: "+"("+Pointer.x+","+Pointer.y+")";
  }
};

let handleEnd = function(ev){
  Pointer.isDown=false;
  ev.preventDefault()
  //console.log(ev.touches)
  if(ev.touches){
    $("bookface").textContent= "触屏事件结束!";
    if(Math.abs(Pointer.deltaX) > 100){
      $("bookface").textContent += "， 这是有效触屏滑动！ " ;
    }else{
      $("bookface").textContent += " 本次算无效触屏滑动！ " ;
      $("bookface").style.left = '7%';
    }
  }
  else{

    $("bookface").textContent= "鼠标松开!";
    if(Math.abs(Pointer.deltaX) > 100){
      $("bookface").textContent += "， 这是有效拖动！ " ;
    }else{
      $("bookface").textContent += " 本次算无效拖动！ " ;
      $("bookface").style.left = '7%';
    }
  }
};

let handleMoving = function(ev){
  ev.preventDefault();
  if (ev.touches){
    if (Pointer.isDown){
      console.log("Touch is moving");
      Pointer.deltaX = parseInt( ev.touches[0].pageX - Pointer.x );
    }
  }
}

```

```

        $("bookface").textContent= "正在滑动触屏，滑动距离： " + Pointer.deltaX +"px 。 ";
        $('bookface').style.left =  Pointer.deltaX + 'px' ;
    }
}else{
    if (Pointer.isDown){
        console.log("Pointer isDown and moving");
        Pointer.deltaX = parseInt( ev.pageX - Pointer.x );
        $("bookface").textContent= "正在拖动鼠标，距离： " + Pointer.deltaX +"px 。 ";
        $('bookface').style.left =  Pointer.deltaX + 'px' ;
    }
}
};

$("bookface").addEventListener("mousedown",handleBegin );
$("bookface").addEventListener("touchstart",handleBegin );
$("bookface").addEventListener("mouseup", handleEnd );
$("bookface").addEventListener("touchend",handleEnd );
$("bookface").addEventListener("mouseout", handleEnd );
$("bookface").addEventListener("mousemove", handleMoving);
$("bookface").addEventListener("touchmove", handleMoving);
$("body").addEventListener("keypress", function(ev){
    $("aid").textContent += ev.key ;
});

```

代码块 7.2

7.3 项目的运行和测试

本文此处给出 PC 端用 Edge 浏览器打开项目的结果，如下图 6-3 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二维码，运行测试本项目的本次开发的阶段性效果。



图 7.3

7.4 项目的代码提交和版本管理

```
cd /e/Code/webUI
Git add 1.5.html
Git commit -m "提交对触屏鼠标的通用交互操作版本"
```

截图如下所示：

```
Minglizzz@LAPTOP-LS17EVI4 MINGW64 /e/Code/Git/webUI (master)
$ Git add 1.5.html

Minglizzz@LAPTOP-LS17EVI4 MINGW64 /e/Code/Git/webUI (master)
$ git commit -m "提交对触屏鼠标的通用交互操作版本"
[master 8585749] 提交对触屏鼠标的通用交互操作版本
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 1.5.html
```

8.UI 的个性化键盘交互控制的设计开发

8.1 分析和设计

经过了上一章对触屏技术设计网页的设计经验后，这一章我们通过更高级的代码来实现对个性化键盘交互功能的通用交互操作初步开发，阐述移动互联时代的用户终端的多样，用例图和 dom 图如下图 8.1。

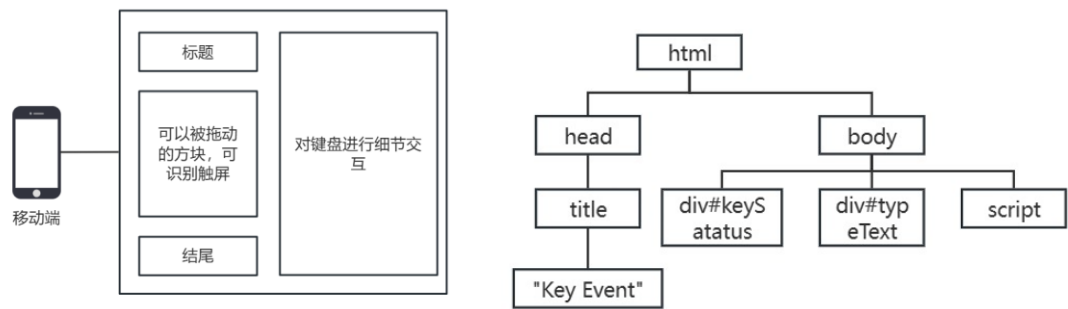


图 8.1

8.2 项目的实现和编程

主要实现功能的代码块如下代码块 8.2.

```
$("#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)
});
} //Code Block End
function $(ele){
    if (typeof ele !== 'string'){
        throw("自定义的$函数参数的数据类型错误，实参必须是字符串！");
        return
    }
    let dom = document.getElementById(ele) ;
    if(dom){
        return dom ;
    }else{
        dom = document.querySelector(ele) ;
        if (dom) {
            return dom ;
        }else{
            throw("执行$函数未能在页面上获取任何元素，请自查问题！");
            return ;
        }
    }
}
} //end of $

```

代码块 8.2

8.3 项目的运行和测试

本文此处给出 PC 端用 Chrome 浏览器打开项目的结果，如下图 6-3 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二维码，运行测试本项目的本次开发的阶段性效果。



图 8.3

8.4 项目的代码提交和版本管理

```
cd /e/Code/webUI
Git add 1.6.html
Git commit -m "此版本跟新了对键盘交互的详细设计"
```

截图如下所示：

```

minglizzz@LAPTOP-LS17EVI4 MINGW64 /e/Code/Git/webUI (master)
$ git add 1.6.html

minglizzz@LAPTOP-LS17EVI4 MINGW64 /e/Code/Git/webUI (master)
$ git commit -m "此版本更新了对键盘交互的详细设计"
[master 42faa36] 此版本更新了对键盘交互的详细设计
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 1.6.html

```

9. 谈谈本项目中的高质量代码

This is a book about instructing computers. Computers are about as common as screwdrivers today, but they are quite a bit more complex, and making them do what you want them to do isn't always easy.

If the task you have for your computer is a common, well-understood one, such as showing you your email or acting like a calculator, you can open the appropriate application and get to work. But for unique or open-ended tasks, there probably is no application. That is where programming may come in. Programming is the act of constructing a program--a set of precise instructions telling a computer what to do. Because computers are dumb, pedantic beasts, programming is fundamentally tedious and frustrating. Fortunately, if you can get over that fact, and maybe even enjoy the rigor of thinking in terms that dumb machines can deal with, programming can be rewarding. It allows you to do things in seconds that would take forever by hand. It is a way to make your computer tool do things that it couldn't do before. And it provides a wonderful exercise in abstract thinking.^[6]

上述文字强调了计算机与螺丝刀等常见工具不同，它们更加复杂。并且由于计算机是“愚蠢而迂腐的生物”，所以编程过程基本上是乏味且令人沮丧的。编程虽然很复杂且有时令人沮丧，但它能让你在计算机上执行独特的任务，并用几秒钟完成手动需要很长时间才能完成的工作，这也是它吸引我的地方。

在 1.6.html 中的这段代码让我印象最为深刻如下图代码块 9.1.

```

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 ;
```

代码块 9.1

这段代码的功能是检查一个给定的字符 `k` 是否是字母、数字或特定的标点符号。如果 `k` 是字母、数字或标点符号中的一个，函数将打印相应的信息（"letters" 或 "puncs"）并返回 `true`。如果 `k` 既不是字母也不是数字或标点符号，则函数返回 `false`。。让我们详细讨论一下这段代码的优点：

1. 代码清晰易懂

这段代码使用了适当的变量名和注释，使得代码逻辑清晰，易于理解。通过注释提醒学生需要研究逻辑的作用，有助于促进对代码的深入理解。

2. 使用了临时数组存储标点符号

使用 `puncs` 数组存储了常见的标点符号，这样可以方便地扩展或修改标点符号集合。遍历 `puncs` 数组来检查输入字符是否为标点符号，代码结构清晰。

3. 使用了直观的条件判断

使用了直观的条件判断来检测输入字符是字母、数字还是标点符号，使得代码易于阅读和理解。代码通过逻辑运算符 `&&` 和 `||` 来组合条件判断，减少了代码的复杂度。

4. 考虑了多种情况

函数考虑了输入字符既可以是字母或数字，也可以是标点符号的情况，对不同类型的字符作出了不同的处理。当输入是多个字符时，返回 `false`，这种情况下函数会提前结束并返回结果，避免了不必要的遍历操作。

如果 `k` 是 'A'，会输出 "letters" 并返回 `true`。

如果 `k` 是 '!', 会输出 "puncs" 并返回 `true`。

如果 `k` 是 '#', 会输出 "puncs" 并返回 `true`。

如果 `k` 是一个空白字符 ' ', 会输出 "puncs" 并返回 `true`。

如果 `k` 是其他非字母、非数字、非标点符号的字符（如某些特殊符号），不会输出任何内容，返回 `false`。

5. 使用了循环来检查标点符号

使用了 `for` 循环遍历标点符号数组，以检查输入字符是否为标点符号，这种方法简洁高效。总体来说，这段代码的优点在于其清晰的逻辑结构，易于理解和扩展。通过对不同类型字符的分类判断，这段代码能够有效地识别输入字符的类型，并返回相应的结果。

10. 用 **gitBash** 工具管理项目代码仓库和 **http** 服务器

10.1 经典 Bash 工具介绍

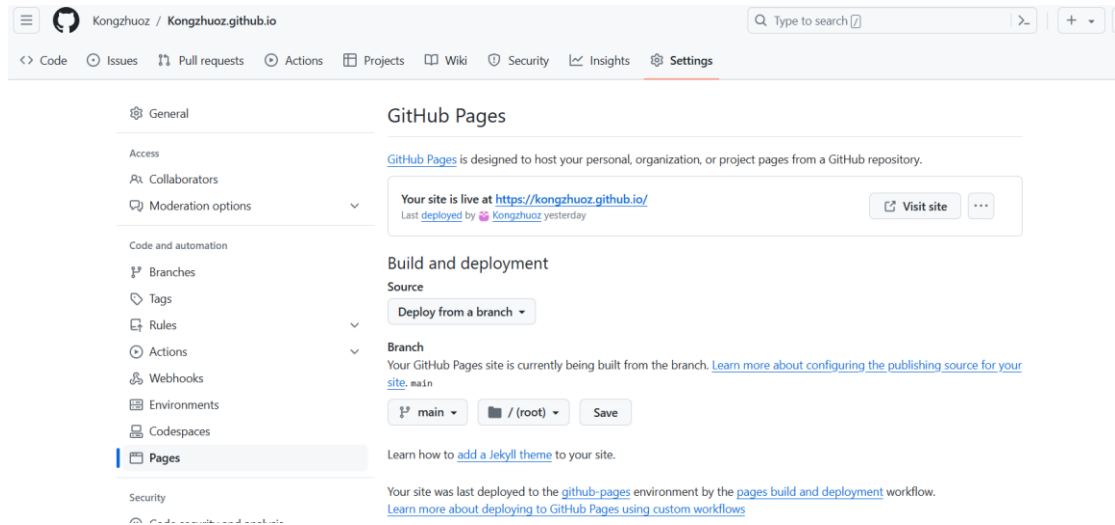
When we speak of the command line, we are really referring to the shell. The shell is a program that takes keyboard commands and passes them to the operating system to carry out. Almost all Linux distributions supply a shell program from the GNU Project called `bash`. The name is an acronym for bourne-again shell, a reference to the fact that `bash` is an enhanced replacement for `sh`, the original Unix shell program written by Steve Bourne.^[7]

Like Windows, a Unix-like operating system such as Linux organizes its files in what is called a hierarchical directory structure. This means they are organized in a tree-like pattern of directories (sometimes called folders in other systems), which may contain files and other directories. The first directory in the file system is called the root directory. The root directory contains files and sub directories, which contain more files and sub directories, and so on.^[7]

也就是说，命令行实际指的是 `shell`，它是一个接收键盘命令并将其传递给操作系统执行的程序。而且在大多数 Linux 发行版中，其中常用的 `shell` 是 `bash`，它是 `sh`（由 Steve Bourne 编写的原始 Unix shell 程序）的增强替代品。Linux 等类 Unix 操作系统采用层次目录结构来组织文件，即以树状模式排列的目录和文件。文件系统的顶级目录称为根目录，包含文件和子目录，而子目录中又可以包含更多的文件和子目录，是一层一层包含的关系。

10.2 通过 gitHub 平台实现本项目的全球域名

首先，你需要在 **GitHub** 上有一个属于自己的账户，然后按着我的步骤操作就可以



1. 打开自己的任意一个 **GitHub** 仓库，例如我自己的是 *Kongzhuoz*
2. 点击右侧的 **settings**
3. 点击侧边栏的 **Pages**
4. 选择存放个人网站代码的分支和文件夹，建议：**gh-pages** 分支的 **docs** 文件夹（便于区分源代码和打包的代码），但是我这个用的还是 **main** 分支的 **root** 文件夹

需要注意的是，每次更改完项目的内容时，还不能够第一时间看到页面的变化，这是因为 **GitHub Pages** 需要时间来构建和部署更新的内容，这可能需要几分钟，可以在 **pages** 页面中观察，看到其中的一个红点变为绿点时，即代表项目部署完成。

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 *

 Kongzhuoz /

Great repository names are short and memorable. Need inspiration? How about [miniature-goggles](#) ?

Description (optional)

- ☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.
- ☐  **Private**
You choose who can see and commit to this repository.

Create repository

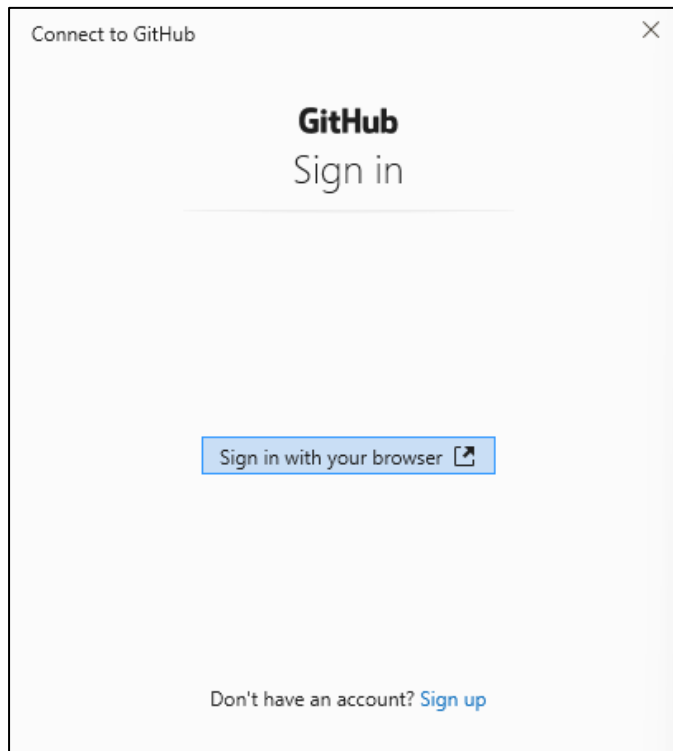
点击窗口右下角的绿色“Create repository”，则可创建一个空的远程代码仓库。

10.4 设置本地仓库和远程代码仓库的链接

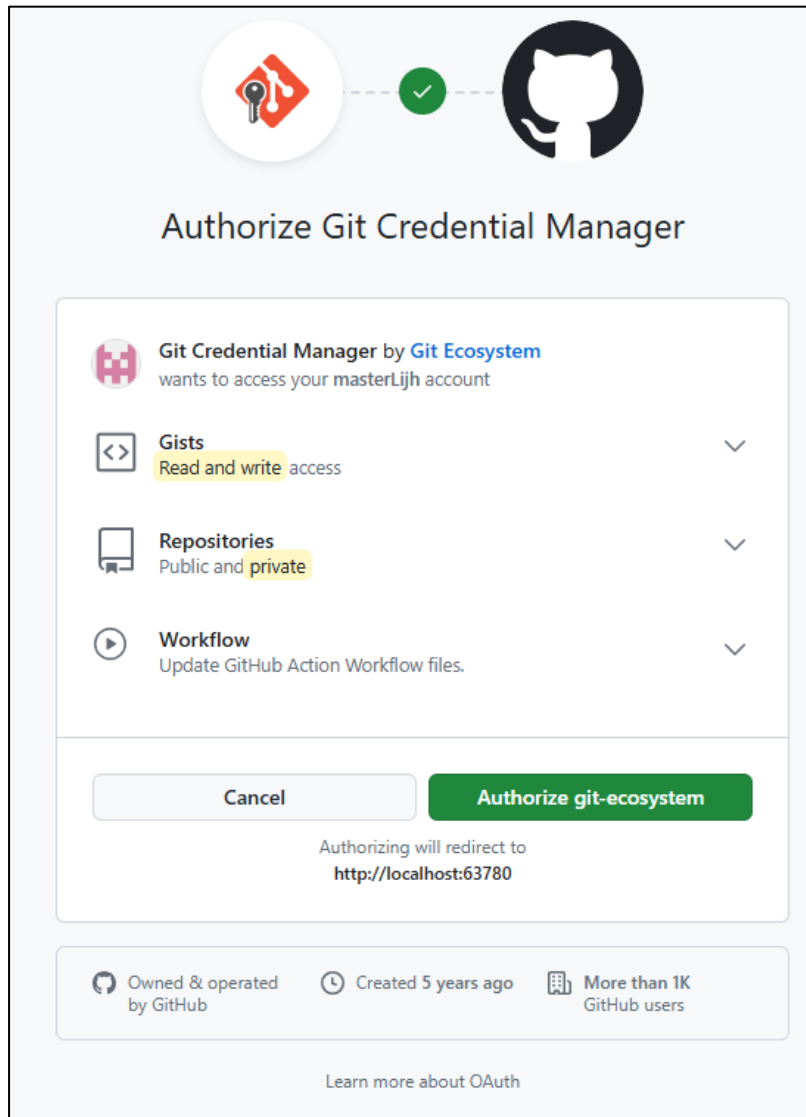
进入本地 webUI 项目的文件夹后，通过下面的命令把本地代码仓库与远程建立密钥链接

```
$ 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/kongzhuoz/kongzhuoz.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：



过程中出现错误

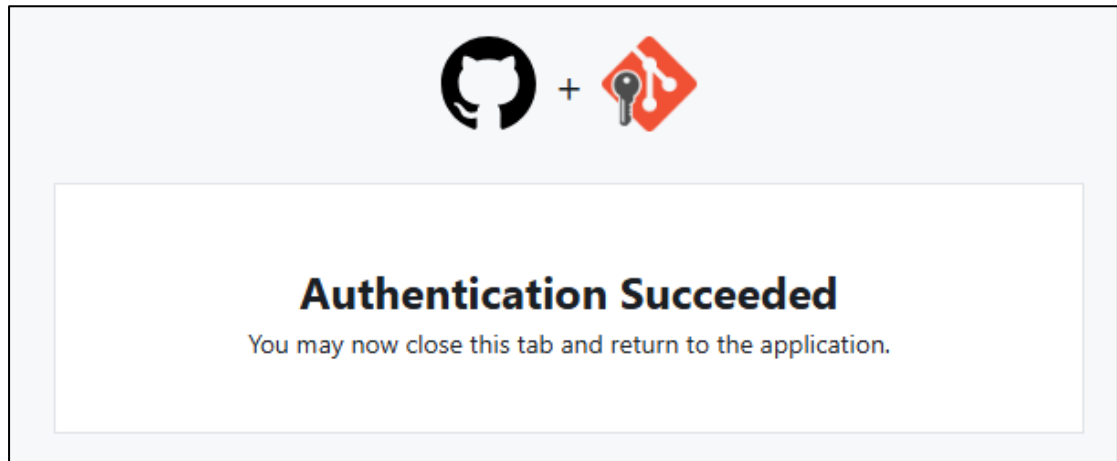
在命令行中和 gitBash 使用命令

`git config --global http.sslVerify "false"`

得已解决

```
$ git clone https://github.com/Kongzhuoz.github.io/kxxz.git
Cloning into 'kxxz'...
warning: SECURITY WARNING
warning: TLS certificate verification has been disabled!
warning: HTTPS connections may not be secure. See https://aka.ms/gcmcore-tlsverifi
warning: for more information.
warning: SECURITY WARNING
warning: TLS certificate verification has been disabled!
warning: HTTPS connections may not be secure. See https://aka.ms/gcmcore-tlsverifi
warning: for more information.
```

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