

成 绩:

# 江西科技师范大学

## 课程设计（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）：Customized UI design and Programming based on Web client technology

院（系）： 元宇宙产业学院

专 业： 计算机科学与技术

学生姓名： 聂雨欣

学 号： 20213634

指导教师： 李健宏

2024 年 5 月 22 日

# 目录

1. 引言 .....	1
1.1 项目概述 .....	1
1.2 研学计划 .....	2
1.3 研究方法 .....	3
2. Web 平台和客户端技术概述 .....	4
2.1 Web 平台 .....	4
2.2 Web 编程 .....	4
2.2.1 MVC 设计模式 .....	4
2.3 软件开发的过程管理——增量式开发模式 .....	5
2.3.1 瀑布模型 .....	5
2.3.2 增量模型 .....	5
3. 内容设计概要 .....	7
3.1 分析与设计 .....	7
3.2 编程与实现 .....	7
3.3 测试与运行 .....	8
3.4 代码提交和版本管理 .....	8
4. 移动互联时代的响应式设计 .....	10
4.1 分析与设计 .....	10
4.2 编程与实现 .....	11
4.3 测试与运行 .....	13
4.4 代码提交和版本管理 .....	14
5. 个性化交互 UI 初步——鼠标模型的设计 .....	15
5.1 分析与实现 .....	15
5.2 编程与实现 .....	15
5.3 测试与运行 .....	16
5.4 代码提交和版本管理 .....	16
6. 探索 UI 对鼠标拖动（模拟触屏）的操作模式 .....	17
6.1 分析与设计 .....	17

6.2 编程和实现 .....	17
6.3 测试与运行 .....	19
6.4 代码提交和版本管理 .....	20
7. 个性化 UI 对键盘控制 .....	21
7.1 分析与设计 .....	21
7.2 编程与实现 .....	22
7.3 测试与运行 .....	24
7.4 代码提交和版本管理 .....	24
8. 鼠标滚轮事件的应用-翻页 .....	27
8.1 分析与设计 .....	27
8.2 编程与实现 .....	27
8.3 测试与运行 .....	29
8.4 代码提交和版本管理 .....	29
9. 本项目的高质量代码的编写 .....	31
10. 用 gitBash 工具管理本项目的 http 服务器 .....	32
10.1 跨世纪的经典 Bash 工具 .....	32
10.2 利用 gitBash 工具对本项目进行管理 .....	32
10.2.1 项目文件的基本搭建 .....	32
10.2.2 项目代码的提交与上传 .....	33
10.2.3 项目日志的反馈查询 .....	33
10.3 通过 GitHub 平台实现本项目的全球域名 .....	34
10.3.1 创建空的远程代码仓库 .....	35
10.3.2 设置本地仓库与远程代码仓库的连接以及推送 .....	35
参考文献 .....	38

# 基于 Web 客户端技术的个性化 UI 设计和实现

**摘要:** 近十年来, 基于 HTML5 核心的 Web 标准软件开发技术因其跨平台和开源的优势, 已广泛应用于各类领域的应用软件开发中。通过对本次毕业设计任务的分析, 本项目选择了 HTML5 的 Web 客户端技术作为技术路线, 进行了深入的程序设计和软件开发研究与实践。经过广泛查阅相关技术书籍、开发者论坛和文献, 我们设计并开发了一款个性化用户界面 (UI) 的应用程序。在开发过程中, 我们综合运用了 HTML 语言进行内容建模, 使用 CSS 语言设计 UI 的外观, 并通过 JavaScript 编程实现 UI 的交互功能。项目中的每一行代码均为手工逐条编写, 未引入任何第三方代码 (包括框架和库), 除直接使用 Web 客户端最底层的 API 外。项目采用了响应式设计, 能够智能适应移动互联网时代用户屏幕的多样化需求。此外, 我们大量运用了面向对象的程序设计思想, 例如构建了一个通用的 pointer 模型, 通过一套代码实现了对鼠标和触屏的控制, 展现了项目的高代码质量, 这是本项目的一大亮点。从工程管理的角度看, 本项目采用了增量式开发模式, 以逐步求精的方式进行了七次代码的增量式重构 (A:分析 D:设计 I:实现 T:测试), 顺利实现了项目的设计、开发和测试。在代码开源和分享方面, 本项目使用 Git 工具进行版本管理, 在漫长的开发过程中进行了七次重构并正式提交了代码, 同时在测试阶段修改并提交了两次代码。最终, 我们利用 Gitbash 工具将项目的代码仓库上传至 GitHub 平台, 并借助 GitHub 提供的 HTTP 服务器, 实现了 UI 应用在全球互联网的部署, 用户可以通过地址或二维码便捷地跨平台高效访问该程序。总而言之, 本项目通过采用 HTML5 技术进行全面开发和优化, 展示了跨平台和响应式设计的优势, 同时通过严格的工程管理和版本控制, 实现了高质量的代码和广泛的应用部署。

**关键词:** Web 客户端技术; html5; css; javascript; github

## 1. 引言

在信息化快速发展的今天, Web 应用已成为日常生活和工作的核心。随着移动互联网的普及和用户需求的多样化, Web 技术不断演进。HTML5 作为新一代 Web 标准, 以其强大功能和跨平台优势广泛应用。它不仅提供丰富的标签和 API, 还通过 CSS3 和 JavaScript 实现灵活和动态的用户界面设计, 使开发者能够创建功能强大、美观且兼容性高的 Web 应用。

### 1.1 项目概述

本次毕业设计的目标是基于 HTML5 技术, 设计和实现一个个性化的用户界面 (UI) 应用程序。通过分析项目需求, 我们选择 HTML5 为核心技术路线, 结合 CSS 进行 UI 设计, 利用 JavaScript 实现交互功能。为保证代码质量和可维护性, 项目采用面向对象设计思想和响应式设计, 以适应不同设备屏幕的需求。

整个开发过程中, 我们手工编写每一行代码, 深入理解 Web 客户端技术的核心原理。项目采用增量式开发模式, 经过多次迭代和重构, 不断优化代码, 实现了设计、开发和测试的有机结合。为了更好地进行版本控制和代码管理, 我们使用 Git 工具, 将项目托管在 GitHub 平台上, 并通过 GitHub 的 HTTP 服务器实现全球互联网部署, 使用户

能够便捷地访问和体验该程序。本论文将详细介绍本次毕业设计的技术路线、开发过程、功能实现及代码管理方法。

## 1.2 研学计划

表 1.1 研学计划表

时间安排	目标
第一周	进行“三段论”式内容设计概要开发。通过分析需求和功能，初步构建应用程序的结构和逻辑。
第二周	实现网页窄屏代码，并进行个性化 UI 设计中的鼠标模型 1.0 的探究。利用鼠标监听事件，显示鼠标的实时坐标。
第三周	实现个性化 UI 设计中的鼠标模型 2.0，支持触屏功能。利用鼠标监听事件，实现页面元素的拖动，并显示拖动距离。
第四周	实现个性化 UI 设计中的键盘控制 1.0，利用 keydown 和 keyup 事件，显示按下键的字符以及 key 和 keyCode 值。
第五周	实现个性化 UI 设计中的键盘控制 2.0，利用 for 循环和条件判断进行筛选，只显示单个字符。
第六周	实现个性化 UI 设计中的键盘控制 3.0，处理特殊键 Back 和 Enter，通过监听事件实现文本删除与换行功能。
第七周	综合前六周的内容，设计并实现一个个性化网站。主要功能包括： 1. 显示我喜欢的设计师的部分作品信息(包括翻页功能); 2. 附有键盘响应区，可以进行文本编辑。
第八周	逐步实现其他功能模块，采用增量式开发模式，每完成一个功能模块即进行测试和优化。
第九周	对整个应用程序进行全面的测试，发现并修复 BUG，优化代码性能，确保程序的稳定性和高效性。
第十周	使用 Git 进行版本控制和代码管理，将项目托管在 GitHub 平台上，并通过 GitHub 的 HTTP 服务器进行部署，确保用户能够便捷地访问和体验应用程序。
第十一周	总结整个开发过程，撰写毕业论文，准备答辩材料。通过论文和答辩展示项目的设计思想、实现过程及创新点。

### 1.3 研究方法

为了确保本项目的科学性和实用性，本论文采取了六种研究方法。首先，我们采用文献综述法，查阅了大量关于 HTML5、CSS、JavaScript 以及相关技术的文献资料，了解了当前领域的发展动态和研究成果，为项目的技术使用和开发提供理论支持；使用原型设计法，使用工具绘制了界面原型，确保设计方案的可行性。其次，我们利用增量开发法，采用迭代增量式开发模式，将整个项目划分为若干小模块，逐步开发和测试，且每次都对已有的功能进行优化，确保项目的质量的进度。最后，我们采用实验测试法和版本控制法，在浏览器的开发者面板中进行调试修改，并使用 Git 进行版本控制和代码管理，通过 GitHub 平台进行项目的托管和全球互联网部署。

本项目基于 Web 客户端技术开发，以适应移动互联网时代 Web application 的前端需求。本文通过广泛查阅技术资料和相关文献，特别是 [mozilla](#) 组织的 MDN 社区的技术实践文章，通过了 HTML 内容建模、CSS 样式设计和 JavaScript 功能编程的基本技术和技巧。结合本学科的核心课程知识，实现了一个个性化的基于 web 客户端技术的响应式用户界面（UI），该 UI 能够较好的适配各种屏幕。在功能上，通过 DOM 技术和事件监听实现了对鼠标、触屏和键盘的底层事件的支持并实现了流畅地响应。基于面向对象的思想，为鼠标和触屏设计了对象模型，并通过代码实现了对这类指向性设备的模拟。为了能更好地应用工程思想、设计和开发管理项目，本项目采用了软件工程的增量式开发模式。共进行了 6 次项目迭代开发，每次迭代都经历了 ADIT（分析（Analysis）、设计（Design）、实施（Implementation）、测试（test））四个经典开发阶段。通过逐步求精的方式编写了 UI 应用程序。最后，为了分享和共享代码，与其他开发者合作，使用了 git 工具进行代码和开发过程日志记录。总共进行了 7 次代码提交操作，详细记录和展现了开发思路和代码优化的过程，然后通过 [gitbash](#) 将项目上传到 GitHub 并建立了自己的代码仓库，将该代码仓库设置成为了 HTTP 服务器，实现了全球便捷访问。

## 2. Web 平台和客户端技术概述

### 2.1 Web 平台

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，这也是我的毕设项目应用的技术路线。

Web 是万维网的缩写。但是，我们都遵从这样一个惯例—Web 并等于 World Wide Web。网络是文档的集合，称之为网页，绝大部分由世界各地的计算机用户所共享。不同类型的网页做不同的事情，其中它们有一件共同事情就是在电脑屏幕上显示内容。这里我们提到的内容指的是文本，图片和用户输入机制（如文本和按钮等）<sup>[3]</sup>。

### 2.2 Web 编程

Web 编程是一个很大的领域，不同类型的 Web 编程由不同的工具实现。但是，基本上所有的工具都使用核心语言 HTML。其中，HTML5，CSS 和 JavaScript 这三种技术被认为是 Web 编程的主要支柱。与此同时，当我们使用客户端进行 Web 编程时，所有网页计算最终都将在用户的计算机（客户端计算机）上执行<sup>[4]</sup>。

#### 2.2.1 MVC 设计模式

MVC 全名为 Model View Controller，即模型—视图—控制三层体系架构。它是一种设计模式，同时也反映了人类社会生产中的伟大分工智慧。它使用一种业务逻辑，数据与界面显示分离的方法来组织代码。在 Web 中，我们可以这样理解：M 对应的是 Model(模型层)，可采用 HTML 标记语言进行建模；V 对应的是 View(视图层)，单独用 CSS 语言来渲染外观；JavaScript 对应的是 Controller(控制层)，它结合前二者，实现功能层面和微观代码的所有控制。这样将对象、显示、控制分离有助于提高代码的灵活性与复用性。

## 2.3 软件开发的过程管理——增量式开发模式

从软件工程视角的项目开发和管理来看，软件生命周期中的开发过程包括四个阶段：分析、设计、实现和测试。开发过程中有几个模型，我们在这里讨论最常见的模型：瀑布模型和增量模型。

### 2.3.1 瀑布模型

瀑布模型，亦称瀑布模式，是于 1970 年由温斯顿·W·罗伊斯等人所发展之系统发展生命周期的模型，如图 2.1 瀑布模型所示。该模型将系统发展的过程，大致区分为四个阶段：分析、设计、实现、测试，其并且明确的定义每一阶段中的工作。当完成一个阶段的工作以后，才会进入下一个阶段的工作。

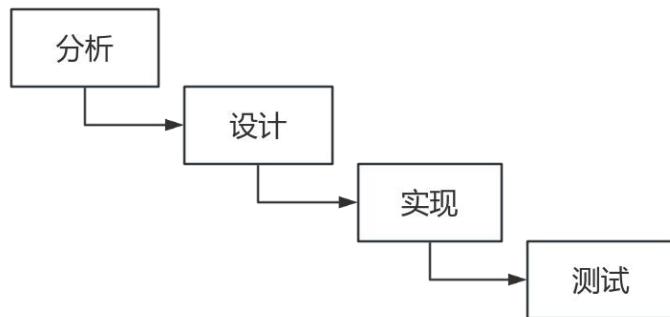


图 2.1 瀑布

瀑布模型最大的优点就是，每个阶段都在下一个阶段开始之前完成。这也就意味着，我们有了分析阶段的完整结果，在设计阶段就可以明确知道该做什么。同时，我们也可以测试整个系统，因为在整个开发中的系统已经准备好了。当然，瀑布模型也存在一定的局限性，它难以定位问题。如果在部分流程中存在问题，则必须要检测整个系统[5]。

### 2.3.2 增量模型

增量模型融合了瀑布模型的基本成分（重复应用）和原型实现的迭代特征。在增量模型中，软件是按一系列步骤开发的。开发人员首先完成整个系统的简化版本。这个版本代表整个系统，但不包括细节，如图 2.2 增量模型显示了增量模型的概念。

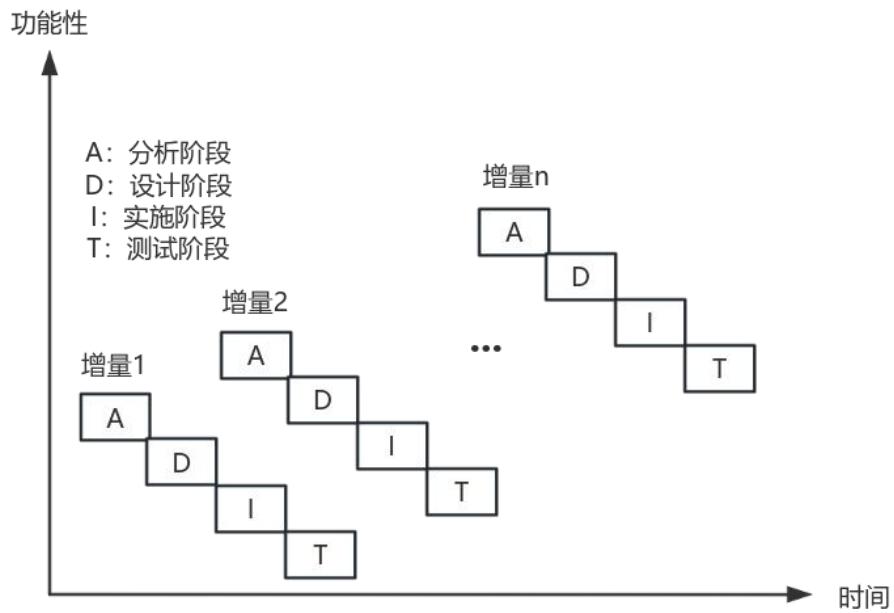


图 2.2 增量模型

在第二个版本中，添加了更多的细节，而一些未完成，并再次测试系统。如果有问题，开发人员就会知道问题出在了新功能上。在现有系统正常工作之前，它们不会添加更多的功能。这个过程会一直持续到添加完所有需要的功能<sup>[5]</sup>。

### 3. 内容设计概要

#### 3.1 分析与设计

为了在互联网上更加方便且直观地展示我们所收集到的一些艺术作品的相关信息（包括图片以及作者，创作日期等内容），我计划搭建一个开源的个性化展示网站。同时，为了增加网站的可读性和美观性，我们首先给这些信息进行了分类，我将页面划分为三个主要部分包括标题区、内容区、附加信息区等，如图 3.1 所示。再根据以上分析，我们可以设计出对应的 DOM 结构树如图 3.2 所示。

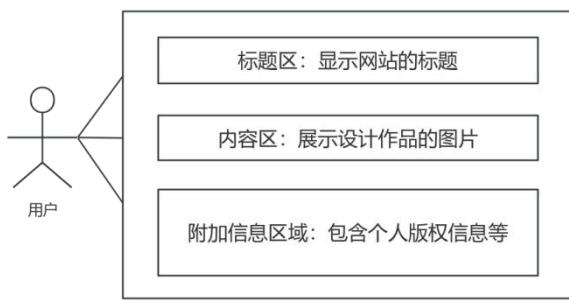


图 3.1 内容划分用例图

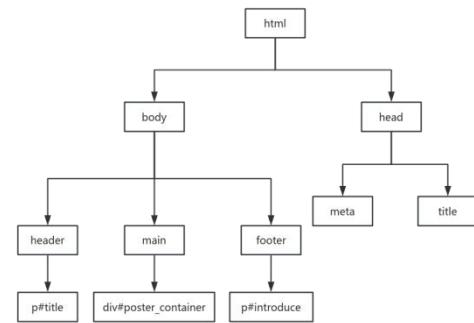


图 3.2 内容划分 DOM 树

#### 3.2 编程与实现

首先，我利用 HTML5 创建了页面的结构模型，然后再分别创建子元素 header、main、footer 用于分别存储标题，作品内容以及个人版权等信息。接着，我使用 CSS 对页面进行渲染，以便呈现出如图 3.3 所示的页面效果。其中，实现方式如代码块 3.1 所示。

```
<header>
    Design Works Display
</header>
<main>
    Poster Content
</main>
<footer>
    &copy; 2024 聂雨欣 25897274922qq.com
</footer>
```

代码块 3.1 划分页面结构

### 3.3 测试与运行

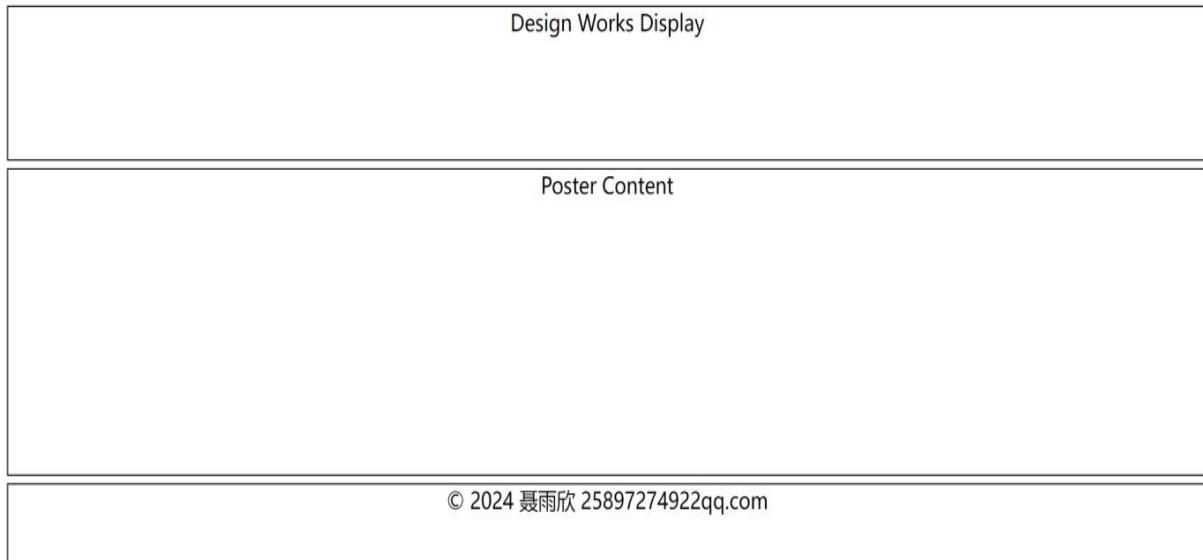


图 3.3 运行结果

### 3.4 代码提交和版本管理

在第一次的代码编写以及运行结束之后。为了使得我们网页代码更加具有组织性，我便选择了 gitBash 工具进行管理，下面我将简述一下我们的操作步骤：

#### (1) 项目代码的上传

在编写并成功运行了 index.html 和 myCss.css 以及 myJs.js 的代码后，执行以下操作（如命令 3.1 所示）对第一次的项目代码进行提交。项目代码提交成功之后，gitBash 的反馈信息如图 3.4 所示。

```
1. 添加 `index.html` 和 `myCss.css` 以及 `myJs.js` 文件到暂存区:  
$ git add index.html myCss.css myJs.js  
2. 提交代码并添加提交信息：“项目第一版：“三段论”式的内容设计概要”  
$ git commit -m “项目第一版：“三段论”式的内容设计概要”
```

命令 3.1 项目代码的提交

```
25897@LAPTOP-UHCF3JU0 MINGW64 /d/webUI (master)  
$ git commit -m 项目第一版：“三段论”式的内容设计概要  
[master (root-commit) 65deb61] 项目第一版：“三段论”式的内容设计概要  
 3 files changed, 0 insertions(+), 0 deletions(-)  
  create mode 100644 index.html  
  create mode 100644 myCss.css  
  create mode 100644 myJs.js
```

图 3.4 代码提交反馈

#### (2) 项目日志的反馈查询

自此，项目代码仓库也就开启了严谨的历史记录，我们可以使用日志命令(git log)

进行查看。gitBash 反馈的代码仓库日志如图 3.5 代码仓库日志反馈所示。

```
25897@LAPTOP-UHCF3JU0 MINGW64 /d/webUI (master)
$ git log
commit 65deb61a490e15f8ed798a4558e85435198eb1fc (HEAD -> master)
Author: 江科师大聂雨欣 <alianwhite1928@gmail.com>
Date:   Thu May 30 09:20:48 2024 +0800
```

项目第一版：“三段论”式的内容设计概要

图 3.5 代码仓库日志反馈

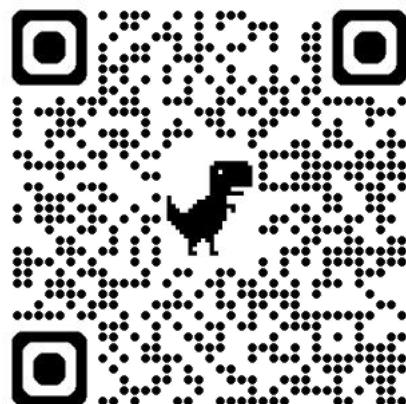


图 3.6 WebUI\_v1.0 移动端二维码

通过上述操作，我们成功完成了项目代码的第一次上传与提交。随着学习的深入，我们的代码将也不断更新和迭代。

## 4. 移动互联时代的响应式设计

由于成本和技术等因素，不同的设备（如电脑，手机等）拥有各种不同的屏幕尺寸和分辨率。为了应对这种差异，设计师们选择制定一般的布局准则，让浏览器自行解决如何在给定的计算机上显示页面，而不是为了每个设备类型提供专门的网页版本<sup>[1]</sup>。

这种方式有一个有趣的结果，那就是同一个网页在不同浏览器或硬件上可能会有略微不同的显示效果。例如，如果一个屏幕比另一个屏幕宽，则可以显示的文本行长度或图像大小会有所不同。这是因为，网页提供了关于期待呈现的一般指导方针，具体的显示细节由浏览器根据设备特性自行决定。

这种响应式设计方法，使得网页能够以适应各种不同的设备和屏幕尺寸，为用户体用最佳的浏览体验。尽管同以网页不同设备上可能略有差异，但整体布局和内容仍能保持一致，满足了现代 Web 设计的需要。

### 4.1 分析与设计

人们获取信息的途径是多样且不断变化的，在一些固定场所比如家中或教室，我们更倾向于使用电脑来进行学习和工作。但是，在一些变化的场景如出行时，我们却更倾向于使用手机和平板等设备来进行信息的快速查看。那现在我们的网站仅仅在电脑端实现了较为舒适的布局，如果在手机端呢？它可能就错位了！因此，我们需要一个同时适用于电脑端和手机端的网站，以确保无论在哪一种设备上，用户都能获得良好的使用体验，如图 4.1 所示。再根据以上分析，我们可以设计出对应的 DOM 结构树和图 4.2 所示。

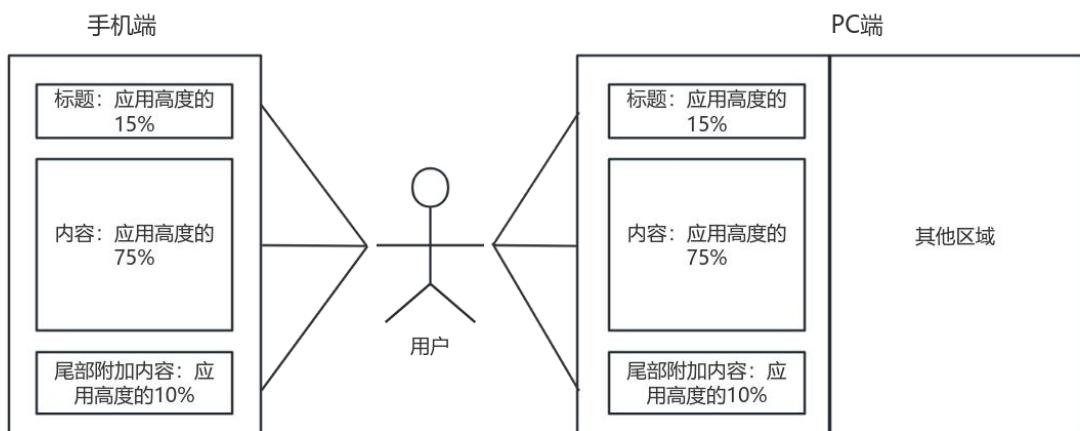


图 4.1 窄屏幕设计用例图

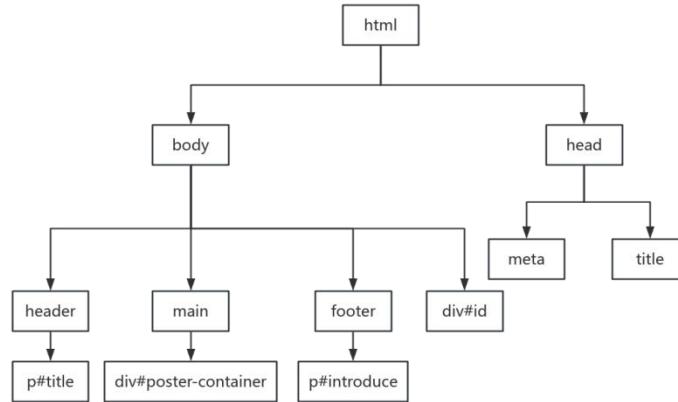


图 4.2 窄屏幕设计 DOM 树

## 4.2 编程与实现

为了让页面能够自动适应不同设备（例如 iPhone 12 Pro 和电脑端）的屏幕尺寸，使其呈现出良好的自适应效果，我采取了 JavaScript 和 CSS 来动态调整页面布局和样式。在宏观上，我利用 JavaScript 动态获取设备窗口的宽度和高度信息，以便于根据不同设备的特性，动态调整页面元素的大小。其次，我使用 CSS 对页面内容进行整体性的划分和样式的定义，来实现页面布局的自适应。

本阶段初次引入了 CSS 中比较高阶的语法，如 em 和%单位，这些单位在响应式设计中非常有效。通过利用%单位，我给 header，body，footer 分别赋予了相对于父容器的比例，如 15%，75%，10%。而通过 em 单位，继承 body 的属性，灵活地设置了字体的大小。如代码块 4.1 所示。

```

<style>
  * {
    margin: 0;
    color: grey;
    text-align: center;
    align-content: center;
    justify-content: center;
    box-sizing: border-box;
    background-color: ghostwhite;
    font-family: Karla, sans-serif;
  }
  header {
    height: 15%;
    display: flex;
    border: 5px solid black;
    border-bottom: none;
  }

```

```



```

#### 代码块 4.1CSS 整体样式设计

此外，我引入了 JavaScript 来实现页面的响应式设计。首先，我创建了一个名为 UI 的对象，用于存储显示设备窗口的宽度和高度信息。然后，我再通过获取到的设备窗口宽度计算了默认字体的大小。最后，通过动态 CSS 的设置，实现了软件界面的全屏显示，以及对应字体大小的动态调节，呈现结果如图 4.5 笔记本电脑页面显示图所示。具体实现如代码块 4.2 所示。

```

// 创建一个名为 UI 的对象，用于存储系统的宽度和高度信息
var UI = {};
// 记录系统窗口的宽度和高度，并限制宽度不超过 600px
UI.deviceWidth = window.innerWidth > 600 ? 600 : window.innerWidth;
UI.deviceHeight = window.innerHeight;
// 计算默认字体大小
const Letters = 33; // 字母数量
const baseFont = UI.deviceWidth / Letters; // 基准字体大小
// 设置页面的字体大小为默认字体大小
document.body.style.fontSize = baseFont + 'px';
// 通过动态 CSS 设置页面全屏显示
document.body.style.width = UI.deviceWidth + 'px';
document.body.style.height = UI.deviceHeight + 'px';

```

#### 代码块 4.2JS 部分页面响应式设计

### 4.3 测试与运行



图 4.3 iPhone SE 屏幕显示图



图 4.4 Samsung Galaxy S8+屏幕显示图

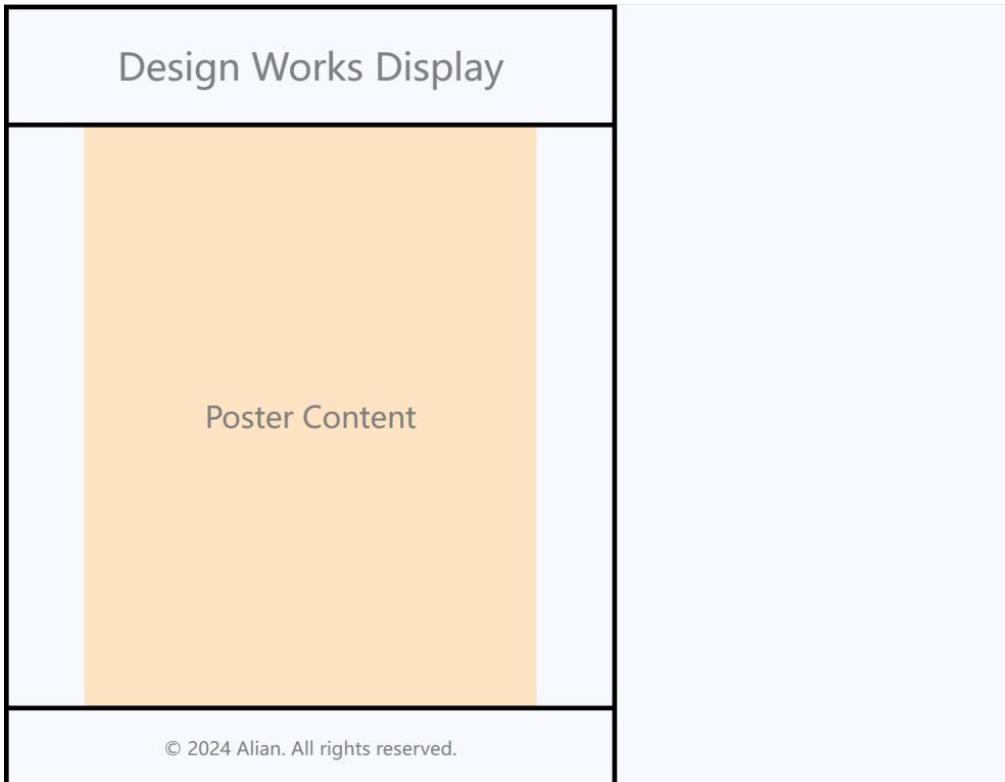


图 4.5 笔记本电脑页面显示图

## 4.4 代码提交和版本管理

```
commit 0564c8015e8c6ff8eb7a46ba978b82997e1c1100
Author: 江科师大聂雨欣 <alianwhite1928@gmail.com>
Date:   Thu May 30 10:35:25 2024 +0800
```

项目第二版：网页窄屏代码实现以及个性化UI设计中的鼠标模型1.0的探究，利用鼠标监听事件，显示鼠标坐标

图 4.6 第二次代码提交

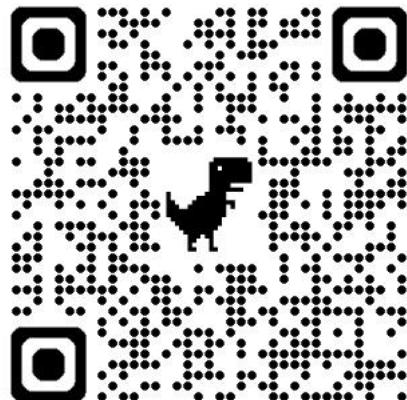


图 4.7 WebUI\_v2.0 移动端二维码

## 5. 个性化交互 UI 初步——鼠标模型的设计

### 5.1 分析与实现

在人机交互的过程中，鼠标是不可或缺的关键一环，所以我们在设计个性化用户界面时，探究了一下 UI 设计中的鼠标模型，特别是关注了鼠标在计算机中所呈现的坐标信息。我们希望当对电脑固定区域进行鼠标点击事件时，页面会显示响应的鼠标的坐标信息，如图 5.1 所示。

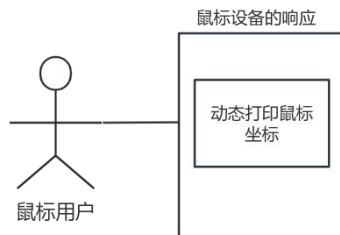


图 5.1 鼠标模型

### 5.2 编程与实现

考虑到信息量的多少，我决定将鼠标的坐标信息显示在 main 元素的子元素.poster 中。我首先创建了一个 mouser 对象，用于存储鼠标的部分信息如 mouseisDown, mouse.x 和 mouse.deltaX 等。然后，我添加三个监听事件，mousedown、mousemove、mouseout 用于获取鼠标按下以及移动时的坐标。最终，通过 p 元素的 textContent 属性，将获取到的信息呈现在页面上，如图 5.2 所示。其中，代码实现方式如代码块 5.1 所示。

```
// 监听鼠标在 .poster 区域的按下事件
$(.poster).addEventListener(mousedown, function (ev) {
    // 获取鼠标按下时的坐标，并在 .poster 区域显示鼠标按下的信息及坐标
    let x = ev.pageX;
    let y = ev.pageY;
    $(.poster).textContent = 鼠标按下了，坐标为: + ( + x + , + y + );
});

// 监听鼠标在 .poster 区域的移动事件
$(.poster).addEventListener(mousemove, function (ev) {
    // 获取鼠标当前坐标，并在 .poster 区域显示鼠标移动的信息及坐标
    let x = ev.pageX;
    let y = ev.pageY;
    $(.poster).textContent = 鼠标正在移动，坐标为: + ( + x + , + y + );
});
```

代码块 5.1 鼠标模型实现

### 5.3 测试与运行

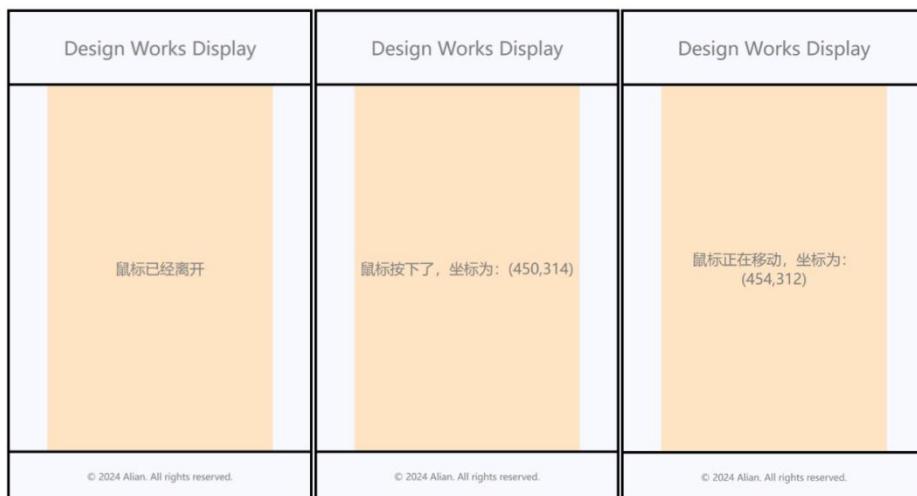


图 5.2 鼠标模型实现

### 5.4 代码提交和版本管理

```
commit 0564c8015e8c6ff8eb7a46ba978b82997e1c1100
Author: 江科师大聂雨欣 <alianwhite1928@gmail.com>
Date:   Thu May 30 10:35:25 2024 +0800
```

项目第二版：网页窄屏代码实现以及个性化UI设计中的鼠标模型1.0的探究，利用鼠标监听事件，显示鼠标坐标

图 5.3 第三次代码提交



图 5.4 WebUI\_v3.0 移动端二维码

## 6. 探索 UI 对鼠标拖动（模拟触屏）的操作模式

### 6.1 分析与设计

在上一节中，我们研究了鼠标模型，并通过监听鼠标的点击事件来获取鼠标相应的位置。但遗憾的是，这种方式只适用于电脑端。当我们使用手机进行操作时，是无法实现该效果的，所以我们本节探究了一下鼠标的横向滑动以此来模拟手机触屏的拖动功能，如图 6.1 所示。

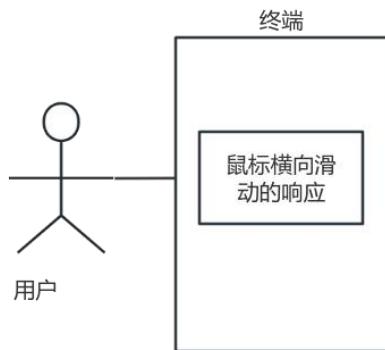


图 6.1 触屏与鼠标统一操作模式

### 6.2 编程和实现

我们的目标是实现简单的拖动功能，通过检测触摸开始、移动和结束事件，并计算移动的距离，来实现页面元素的拖动操作。

首先，我定义了一个名为 `mouse` 的对象，用于存储与鼠标相关的状态和信息，例如鼠标是否按下、鼠标的位置等。然后，我为 `.poster` 元素添加了四个事件监听器。当鼠标按下时，记录下鼠标的位置，并将 `mouse.isDown` 设置为 `true`；当鼠标松开时，将 `mouse.isDown` 设置为 `false`，并根据水平移动距离判断拖动是否有效；当鼠标移出元素时，同样将 `mouse.isDown` 设置为 `false`，并根据移动距离判断拖动是否有效；最后，当鼠标移动时，若鼠标处于按下状态，则计算移动距离并更新页面内容，实现拖动效果，如图 6.2 所示。其中，具体实现如代码块 6.1 所示。

```
//尝试对鼠标设计 UI 控制
var mouse={};
mouse.isDown= false;
mouse.x= 0;
mouse.y= 0;
mouse.deltaX=0;
$('.poster').addEventListener('mousedown',function(ev){
```

```

mouse.isDown=true;
mouse.x= ev.pageX;
mouse.y= ev.pageY;
console.log(mouseDown at x: +(mouse.x +, +mouse.y +) );
$('.poster').textContent= 鼠标按下，坐标: +(mouse.x+,+mouse.y+);
});

$('.poster').addEventListener('mouseup',function(ev){
mouse.isDown=false;

$('.poster').textContent= 鼠标松开!;
if(Math.abs(mouse.deltaX) > 100){
    $('.poster').textContent += , 这是有效拖动! ;
}else{
    $('.poster').textContent += 本次算无效拖动! ;
}
});

$('.poster').addEventListener('mouseout',function(ev){
ev.preventDefault();
mouse.isDown=false;
$('.poster').textContent= 鼠标松开!;
if(Math.abs(mouse.deltaX) > 100){
    $('.poster').textContent += 这次是有效拖动! ;
}else{
    $('.poster').textContent += 本次算无效拖动! ;
}
});

$('.poster').addEventListener('mousemove',function(ev){
ev.preventDefault();
if (mouse.isDown){
    console.log(mouse isDown and moving);
    mouse.deltaX = parseInt( ev.pageX - mouse.x );
    $('.poster').textContent= 正在拖动鼠标，距离: + mouse.deltaX +px .;
    $('.poster').style.left = mouse.deltaX + 'px' ;
}
});
})

```

#### 代码块 6.1 鼠标横向滑动的实现

首先，我定义了一个 touch 对象来存储触屏开始时的坐标和滑动距离。接着，分别在用户触屏开始时（touchstart 事件），记录触点的起始位置并显示坐标信息；在触屏滑动时（touchmove 事件），计算滑动距离并实时更新.poster 元素的文本内容和位置；在触屏结束时（touchend 事件），根据滑动距离判断是否为有效滑动，并更新显示信息，代码实现如代码块 6.2 所示。

```

// 触屏事件处理
var touch = {
  startX: 0,
  startY: 0,
  deltaX: 0
};

$(".poster").addEventListener("touchstart", function (ev) {
  var touchEvent = ev.touches[0];
  touch.startX = touchEvent.pageX;
  touch.startY = touchEvent.pageY;
  $(".poster").textContent = "触屏开始，坐标: " + "(" + touch.startX + ", " + touch.startY + ")";
});

$(".poster").addEventListener("touchmove", function (ev) {
  ev.preventDefault();
  var touchEvent = ev.touches[0];
  touch.deltaX = parseInt(touchEvent.pageX - touch.startX);
  $(".poster").textContent = "正在触屏滑动，距离: " + touch.deltaX + "px。";
  $(".poster").style.left = touch.deltaX + 'px';
});

$(".poster").addEventListener("touchend", function (ev) {
  $(".poster").textContent = "触屏结束!";
  if (Math.abs(touch.deltaX) > 100) {
    $(".poster").textContent += ", 这是有效滑动!";
  } else {
    $(".poster").textContent += " 本次算无效滑动!";
  }
});

```

代码块 6.2 手机屏幕横向滑动的实现

### 6.3 测试与运行

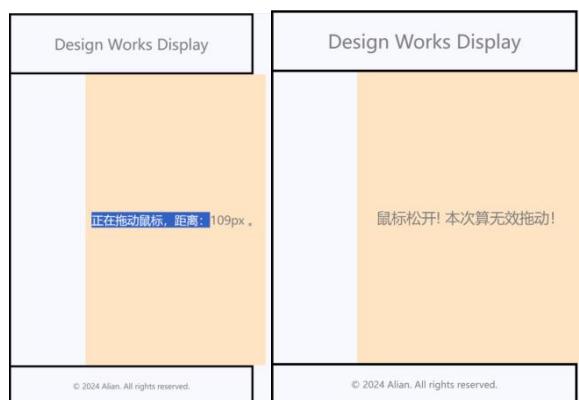


图 6.2 电脑端触屏事件



图 6.3 iPhone SE 触屏横向滑动页面显示



图 6.4 Samsung Galaxy 鼠标横向滑动页面显示

## 6.4 代码提交和版本管理

```
commit d070b07f2bf7f726cc19d80803581a7c1fcc816b
Author: 江科师大聂雨欣 <alianwhite1928@gmail.com>
Date:   Thu May 30 10:36:57 2024 +0800
```

项目第三版：个性化UI设计中的鼠标模型2.0-触屏的实现，利用鼠标监听事件，对页面元素进行拖动，同时显示拖动距离

图 6.5 第四次代码提交



图 6.6 WebUI\_v4.0 移动端二维码

## 7. 个性化 UI 对键盘控制-单个字符输入

### 7.1 分析与设计

在日常的人机交互中，键盘作为输入设备发挥着重要作用。键盘与计算机进行交互的底层逻辑涉及到输入事件的处理和文本传输。当用户按下键盘上的键时，键盘会发送相应的信号给计算机。计算机通过键盘控制器接收这些信号，并将它们转换为数字代码或字符编码，然后发送给操作系统。操作系统接收到键盘输入后，会根据键盘映射表将接收到的代码转换为相应的字符或功能。然后，操作系统将这些字符或功能发送到当前活动的应用程序中。应用程序根据接收到的字符或功能进行相应的处理，比如在文本输入框中显示字符，或者执行与功能键相关的操作。

为了进一步理解键盘的底层原理，以及实现通过键盘输入从而呈现页面信息，我们搭建了一个用户键盘相应区域进行探索，如图 7.1 所示。再根据以上分析，我们可以设计出对应的 DOM 结构树如图 7.3 所示。

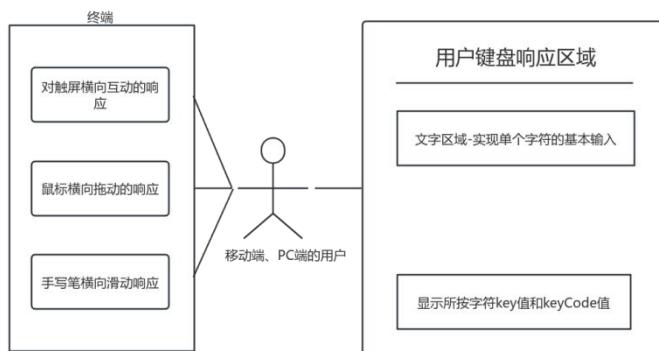


图 7.1 增加个性化键盘响应的用例图

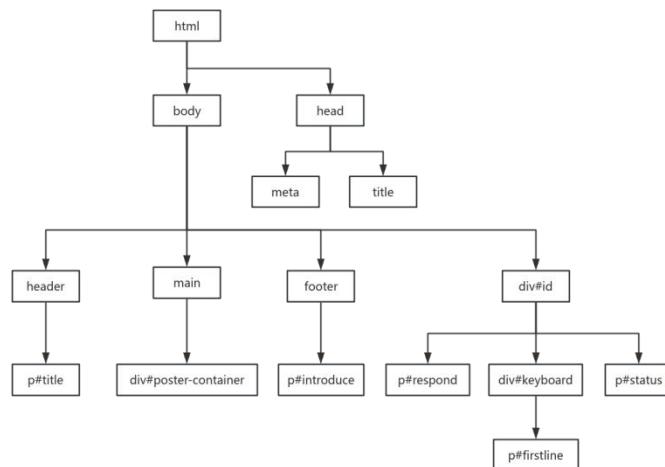


图 7.2 增加个性化键盘响应的完整 DOM 树

## 7.2 编程与实现

为了页面的直观度，我将用户键盘响应区主要分为了两个部分，中间部分用于显示输入字符的内容，尾部用于显示输入字符的 key 值和 keyCode 值。首先，我增加了一个文本输入区域，以实现用户键盘的个性化响应。这个任务分为两个部分：首先，添加一个文本输入区域，让用户可以在网页中输入文字；其次，通过监听键盘的 keydown 和 keyup 事件，实现输入的即时显示，用户按下键盘时在输入区域中显示相应字符，释放键盘时更新显示内容。这样，用户可以在输入区域中自由地输入文本，并即时看到输入的内容变化，步骤如下所示。

### (1) 添加文本输入区域

首先，我创建了一个

元素，并给它分配了一个特定的 ID，命名为 aid，方便我在后续代码中引用。接下来，便是在这个大的 div 中填充内容。首先，我添加了一个标题区域，用一个

元素来显示，对标题进行了简单的说明。然后，是文本输入内容的部分。我创建了另一个

元素，其 ID 设置为 keyboard，用于容纳后续实现换行功能的内容。在这个

中，我添加了一个

元素，用于存储用户输入字符的第一行内容。这个

元素将在后续的交互中不断更新以显示用户的输入。最后，我添加了一个

元素，用于显示与输入字符相关的信息，作为尾部。通过这样的设置，就实现了一个具有清晰结构的键盘响应区域，为后续的功能扩展做好了准备，它的具体实现如代码块 7.1 所示。

```
<div id=aid>
    <p id=respond>用户键盘响应区域</p>
    <div id=keyboard>
        <p id=firstLine>
            &nbsp;
        </p>
    </div>
    <p id=status>
        &nbsp;
    </p>
</div>
```

代码块 7.1 添加文本编辑区

### (2) 字符的即时显示

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 boy 内部的子对象

22

中。具体实现如代码块 7.2 所示：

```
$(body).addEventListener(keypress, function(ev){  
    $(firstLine).textContent += ev.key ;  
});  
$(body).addEventListener(keydown, function (ev) {  
    let k = ev.key;  
    let c = ev.keyCode;  
    $(status).textContent = 您已按下键 :  + k + , + 字符编码 : + c;  
});
```

代码块 7.2 字符的即时显示

为了保证我在后续处理中只处理单个字符，避免多个字符的情况，保证程序逻辑的正确性。所以，我编写了一个 printLetter(k)，用于判断输入字符是否为单个字符。这个函数首先会检查输入字符串的长度，如果大于 1，则说明不是单个字符，返回 false。然后，函数定义了一个包含标点符号的数组用于存储所有可能的标点符号。再通过一系列的条件判断，检查字符是否为字母、数字或者是再标点符号数组中。如果满足任何一种情况，则返回 true，表示输入的是单个字符，并将其显示在屏幕上。具体实现如代码块 7.3 所示。

```
/*判断是否是单个字符*/  
function printLetter(k) {  
    //判断字符串长度是否大于 1  
    if (k.length > 1) {  
        return false;  
    }  
    let puncs = ['~', '`', '!', '@', '#', '$', '%', '^', '&', '*', '(', ')', '+', '=', ',', '.', '<', '>', '?', '/', ' ', ' '];  
    /*字母输出*/  
    if ((k >= 'a' && k <= 'z') || (k >= 'A' && k <= 'Z') || (k >= '0' && k <= '9')) {  
        return true;  
    }  
    /*符号输出*/  
    for (let p of puncs) {  
        if (p === k) {  
            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?  
}  
});
```

代码块 7.3 判读是否为单个字符

### 7.3 测试与运行



图 7.3 键盘模型实现效果图

### 7.4 代码提交和版本管理

```
commit cc27dd81a46ce5f7b1ab7c24d1cd68ca7fdfbb59
Author: 江科师大聂雨欣 <alianwhite1928@gmail.com>
Date:   Thu May 30 10:42:32 2024 +0800
```

项目第六版：个性化UI设计中的键盘控制3.0-利用监听事件，对特殊键Back和Enter进行处理，实现文本的删除与换行

图 7.4 第五次代码提交



图 7.5 WebUI\_v5.0 移动端二维码

## 8. 个性化 UI 对键盘对键盘的控制-特殊键的处理

### 8.1 分析与设计

仅仅是单个字符的显示是远远达不到文本编辑的效果，人在编辑文本的时候不免会出现错误的情况。此外，由于我用于显示文本的 p 元素只能存储一行文字，当输入文本过多时，会导致文本溢出。因此，我对回车键和退格键进行了处理用于换行和删除操作，如图 8.1 所示。

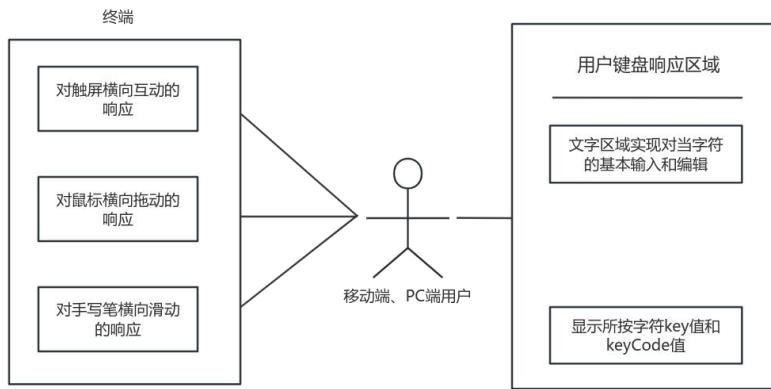


图 8.1 键盘响应区用例图

### 8.2 编程与实现

对于回车键的处理，我首先是判断键盘按下的是否为 Enter 键。如果是的话，则通过 `document.createElement(p)` 来创建子节点。通过添加 p 元素，来实现换行。对于删除键的处理，第一步同样是对输入字符进行判断。如果输入字符是 Backspace，则通过 `slice(0, -1)` 删除最后一个节点的最后一个字符。如果已经没有字符可以删除了，则利用 `$(keyboard).removeChild($(keyboard).lastElementChild)` 删除最后一个子节点，具体实现如代码块 8.1 所示。

```
$(body).addEventListener('keyup', function (ev) {
    let k = ev.key;
    let c = ev.keyCode;
    $status.textContent = '松开按键：' + k + ', 字符编码：' + c;
    /*判断键盘按下的是否为 Enter 键，如果是添 p 元素实现换行*/
    if (k === 'Enter') {
        //有且只能在 document 中创建子节点
        let p = document.createElement('p');
        //通过创建 p 元素，添加子节点来实现换行。
        $(keyboard).appendChild(p);
    } else if (k === 'Backspace') {
```

```

/*没有字符可以删除了，则将该子节点删除*/
if ($keyboard.lastElementChild.textContent === '') {
    /*删除前保证 keyboard 中至少有一个字节点*/
    if ($keyboard.childElementCount > 1) {
        $keyboard.removeChild($keyboard.lastElementChild);
    }
} else {
    $keyboard.lastElementChild.textContent =
$keyboard.lastElementChild.textContent.slice(0, -1);
}
} else if (printLetter(k)) {
    $keyboard.lastElementChild.textContent += k;
}

```

代码块 8.1 删除与换行的实现

### 8.3 测试与运行

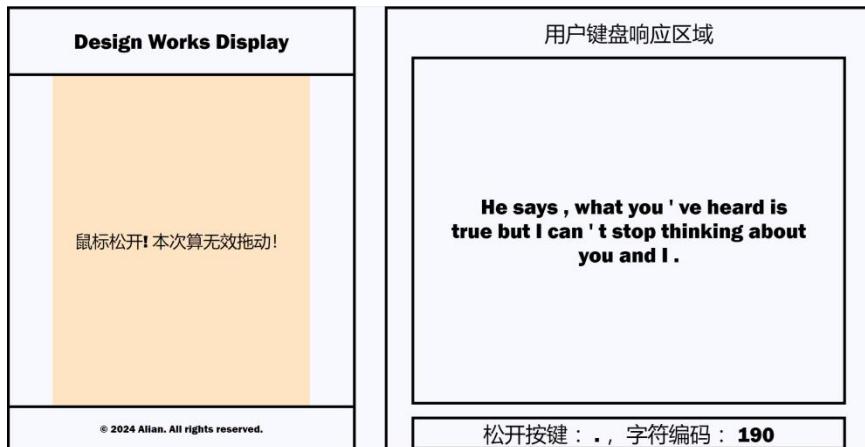


图 8.2 特殊键实现效果图

### 8.4 代码提交和版本管理

```

commit cc27dd81a46ce5f7b1ab7c24d1cd68ca7fdfbb59
Author: 江科师大聂雨欣 <alianwhite1928@gmail.com>
Date:   Thu May 30 10:42:32 2024 +0800

```

项目第六版：个性化UI设计中的键盘控制3.0-利用监听事件，对特殊键Back和Enter进行处理，实现文本的删除与换行

图 8.3 项目代码第六次提交

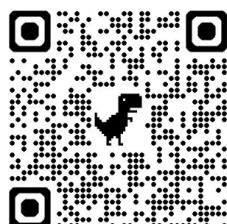


图 8.4 WebUI\_v6.0 移动端二维码

## 9. 鼠标滚轮事件的应用-翻页

### 9.1 分析与设计

在上一节，我们已经实现了通过个性化 UI 对键盘的控制，在本节我们要讨论一下如何通过对鼠标控制来实现页面信息的切换。目前，我们的网页已经可以呈现完整的页面信息了，只不过它的交互性不是很强，而且只能呈现当前屏幕所能显示的信息，显得过于单调。为了丰富网页的信息以及网站本身的互动性，添加翻页功能是十分必要的。我将我们所要处理的信息以数据结构图的形式列出，如图 9.1 所示。

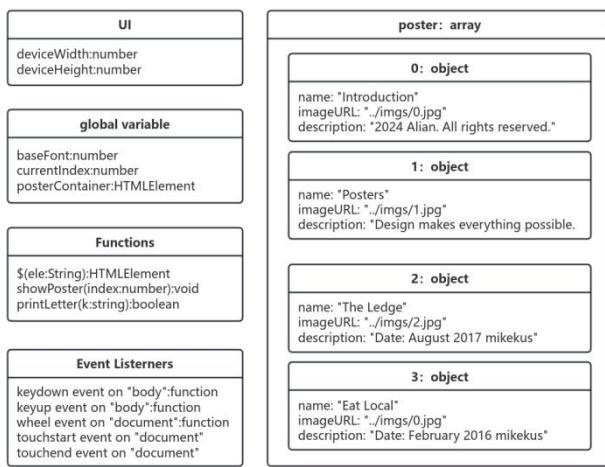


图 9.1 数据结构图

### 9.2 编程与实现

翻页功能的实现，本质就是页面信息的替换。只要将当前页面的图片，以及标题等附加信息随着翻页行为（在这里我们使用的是鼠标滚轮事件）的变化进行切换，我们便可以实现翻页了。

首先，我定义了一个存储海报信息的数组 `poster`，其中包含每个海报的名称、图片 URL 和描述信息，如代码块 9.2 所示。接着，通过变量 `currentIndex` 来追踪当前显示的海报索引，以及变量 `posterContainer` 来获取海报容器元素。然后定义了一个函数 `showPoster(index)`，用于根据传入的索引值显示对应的海报信息。在页面加载时，调用 `showPoster(currentIndex)` 显示第一张海报。接下来，通过监听鼠标滚轮事件，在鼠标向上滑动时切换到前一张海报，在鼠标向下滑动时切换到后一张海报，并调用 `showPoster(currentIndex)` 更新海报容器中的内容，以实现海报的翻页功能，呈现效果如

图 9.2 和图 9.3 所示。具体实现如代码块 9.2 所示。

```
//用于存储数据信息
var poster = [
    {
        "name": "Introduction", "imageURL": "../imgs/0.jpg", "description": "2024 Alian. All rights reserved."
    },
    {
        "name": "Posters", "imageURL": "../imgs/1.jpg", "description": "Design makes everything possible. "
    },
    {
        "name": "The Ledge", "imageURL": "../imgs/2.jpg", "description": "Date: August 2017 mikekus"
    },
    {
        "name": "Eat Local", "imageURL": "../imgs/3.jpg", "description": "Date: February 2016 mikekus"
    }
]
```

代码块 9.1 海报信息数组

```
/*实现翻页功能*/
var currentIndex = 0;
var posterContainer = $("poster-container");
function showPoster(index) {
    var currentPoster = poster[index];
    var posterHTML = `
        <div class="poster" style="width: 100%;height: 100%">
            
        </div>
    `;
    posterContainer.innerHTML = posterHTML;
    $("title").textContent = currentPoster.name;
    $("introduce").textContent = currentPoster.description;
}
// 初始化显示第一张海报
showPoster(currentIndex);
// 监听鼠标滑动事件
document.addEventListener('wheel', function (event) {
    // 鼠标向上滑动，显示前一张海报
    if (event.deltaY < 0) {
        currentIndex = (currentIndex === 0) ? poster.length - 1 : currentIndex - 1;
    }
    // 鼠标向下滑动，显示后一张海报
    else {
        currentIndex = (currentIndex === poster.length - 1) ? 0 : currentIndex + 1;
    }
    showPoster(currentIndex);
});
```

代码块 9.2 翻页功能实现

### 9.3 测试与运行



图 9.2 翻页效果



图 9.3 最终项目实现

### 9.4 代码提交和版本管理

```
commit a29b15ec6d76e4de10ee4dff414239cc1de11265
Author: 江科师大聂雨欣 <alianwhite1928@gmail.com>
Date: Thu May 30 10:45:37 2024 +0800
```

项目第七版：综合前6版的内容，设计个性化网站。具体功能有两点：1. 显示我喜欢的设计师的部分作品信息（为了使得内容更加丰富，已添加翻页功能）2. 附有键盘响应区，可进行文本编辑

图 9.4 第七次代码提交



图 9.5 WebUI\_v7.0 移动端二维码

## 10. 本项目的高质量代码的编写

今天, 电脑就像螺丝刀一样普遍存在, 但它们要复杂得多, 让它们按照你的意愿工作并不总是那么容易。如果你要求电脑完成一个常见且容易理解的任务, 比如显示你的电子邮件或像计算器一样工作, 你可以打开相应的应用程序并开始使用。但对于独特或开放式的任务, 可能没有现成的应用程序可用。

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。由于计算机是愚蠢而固执的野兽, 编程从根本上来说是乏味和令人沮丧的<sup>[5]</sup>。不过, 如果你能克服这一事实, 甚至享受用这种愚蠢的机器可以处理的方式进行严谨思考, 编程就会是有回报的。它可以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。此外, 它还提供了一个很好的抽象思维练习。

# 11. 用 gitBash 工具管理本项目的 http 服务器

## 11.1 跨世纪的经典 Bash 工具

当我们谈到命令行时，我们实际上指的是 shell。Shell 是一个程序，它接受键盘输入的命令，并将其传递给操作系统执行。几乎所有 Linux 发行版都提供一个来自 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 Bashsh Shell 的增强替代品，Shell 是 Steve Bourne 编写的原始 Unix Shell 程序<sup>[6]</sup>。

与 Windows 类似，像 Linux 这样的类 Unix 操作系统使用所谓的分层目录结构来组织文件。这意味着文件被组织为树状的目录模式（在其他系统有时称为文件夹），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。

这种分层目录结构使得 Linux 操作系统的文件管理更加灵活和有条理。用户可以在命令行中使用 Bash 等 Shell 程序来浏览和操作这些目录和文件，从而实现更加精细的文件管理和系统控制。

## 11.2 利用 gitBash 工具对本项目进行管理

下面，我将利用 gitBash 工具对本项目的代码进行系统性的提交和管理。在这个环节，我将按照以下三个步骤进行叙述：项目文件的基本搭建、项目代码的提交与上传、项目日志的反馈查询。

### 11.2.1 项目文件的基本搭建

第一步，建立项目的基本文件结构，并设置好代码仓库的基本信息，包括开发者的姓名和电子邮箱等。具体操作步骤如下，通过 gitBash 命令行执行以下命令（如命令 11.3 所示）。

```
1. 进入根目录下的 d 盘:  
$ cd /d/  
2. 创建名为 `webUI` 的目录:  
$ mkdir webUI  
3. 进入 `webUI` 目录:  
$ cd webUI  
4. 初始化 Git 仓库:  
$ git init  
5. 配置用户名为“江科师大聂雨欣”:  
$ git config user.name 江科师大聂雨欣  
6. 配置用户邮箱为“alianwhite1928@gmail.com”:  
$ git config user.email alianwhite1928@gmail.com
```

```
7. 创建 `index.html` 和 `myCss.css` 以及 `myJs.js` 文件:
```

```
$ touch index.html myCss.css myJs.js
```

### 命令 11.1 搭建项目结构

#### 11.2.2 项目代码的提交与上传

第二步，在编写并成功运行了 index.html 和 myCss.css 以及 myJs.js 的代码后，执行以下操作（如命令 11.2 所示）。其中，成功提交代码后，gitBash 的反馈信息如错误！未定义书签。所示。

```
1. 添加 `index.html` 和 `myCss.css` 以及 `myJs.js` 文件到暂存区:
```

```
$ git add index.html myCss.css myJs.js
```

```
2. 提交代码并添加提交信息：“项目第一版：“三段论”式的内容设计概要”
```

```
$ git commit -m “项目第一版：“三段论”式的内容设计概要”
```

### 命令 11.2 项目代码的提交

```
25897@LAPTOP-UHCF3JU0 MINGW64 /d/webUI (master)
```

```
$ git commit -m 项目第一版：“三段论”式的内容设计概要  
[master (root-commit) 65deb61] 项目第一版：“三段论”式的内容设计概要  
 3 files changed, 0 insertions(+), 0 deletions(-)  
  create mode 100644 index.html  
  create mode 100644 myCss.css  
  create mode 100644 myJs.js
```

图 11.1 代码提交反馈

#### 11.2.3 项目日志的反馈查询

自此，项目代码仓库也就开启了严谨的历史记录，我们可以使用日志命令(git log)进行查看。gitBash 反馈的代码仓库日志图 11.2 所示。

```
25897@LAPTOP-UHCF3JU0 MINGW64 /d/webUI (master)
```

```
$ git log  
commit 65deb61a490e15f8ed798a4558e85435198eb1fc (HEAD -> master)  
Author: 江科师大聂雨欣 <alianwhite1928@gmail.com>  
Date:   Thu May 30 09:20:48 2024 +0800
```

```
项目第一版：“三段论”式的内容设计概要
```

图 11.2 代码仓库日志反馈

通过上述操作，我们成功完成了项目代码的第一次上传与提交。随着学习的深入，我们的代码将也不断更新和迭代。在上面的章节，我们其实已经将代码跟新迭代了六次。下面是本项目几次迭代的日志，如图 11.3 所示。

```
commit a29b15ec6d76e4de10ee4dff414239cc1de11265 (HEAD -> master)
Author: 江科师大聂雨欣 <alianwhite1928@gmail.com>
Date: Thu May 30 10:45:37 2024 +0800
```

项目第七版：综合前6版的内容，设计个性化网站。具体功能有两点：1.显示我喜欢的设计师的部分作品信息（为了使得内容更加丰富，已添加翻页功能）2.附有键盘响应区，可进行文本编辑

```
commit cc27dd81a46ce5f7b1ab7c24d1cd68ca7fdfbb59
Author: 江科师大聂雨欣 <alianwhite1928@gmail.com>
Date: Thu May 30 10:42:32 2024 +0800
```

项目第六版：个性化UI设计中的键盘控制3.0-利用监听事件，对特殊键Back和Enter进行处理，实现文本的删除与换行

```
commit e8758eddfa396a133d5147254bac78ef5823c79a
Author: 江科师大聂雨欣 <alianwhite1928@gmail.com>
Date: Thu May 30 10:40:50 2024 +0800
```

项目第五版：个性化UI设计中的键盘控制2.0-利用for循环和条件判断，进行筛选，只显示单个字符

```
commit cdaf069efe3388461896affd214c86217914fb9
Author: 江科师大聂雨欣 <alianwhite1928@gmail.com>
Date: Thu May 30 10:39:00 2024 +0800
```

项目第四版：个性化UI设计中的键盘控制1.0-利用keydown和keyup键盘底层事件显示所按下键的字符，以及key和keyCode值

```
commit d070b07f2bf7f726cc19d80803581a7c1fcc816b
Author: 江科师大聂雨欣 <alianwhite1928@gmail.com>
Date: Thu May 30 10:36:57 2024 +0800
```

项目第三版：个性化UI设计中的鼠标模型2.0-触屏的实现，利用鼠标监听事件，对页面元素进行拖动，同时显示拖动距离

```
commit 0564c8015e8c6ff8eb7a46ba978b82997e1c1100
Author: 江科师大聂雨欣 <alianwhite1928@gmail.com>
Date: Thu May 30 10:35:25 2024 +0800
```

项目第二版：网页窄屏代码实现以及个性化UI设计中的鼠标模型1.0的探究，利用鼠标监听事件，显示鼠标坐标

```
commit 96f742b56aaaf1da23a68d80f18c8034350abb68d
Author: 江科师大聂雨欣 <alianwhite1928@gmail.com>
Date: Thu May 30 10:11:39 2024 +0800
```

项目第一版：“三段论”式的内容设计概要开发

图 11.3gitBash 代码仓库日志最终反馈

### 11.3 通过 GitHub 平台实现本项目的全球域名

在上一节，我们已经利用 gitBash 实现了在本地对我们的项目代码进行系统性的管理，但是仅仅只是局限于本地是远远不够的，我们希望我们的代码进行开源，让所有人可以进行公开的访问，从而提出更对的参考性意见。所以，本节我们要引入全球知名的免费且开源的代码托管平台 GitHub。

GitHub 提供了便捷的版本控制和协作功能，助力我们高效地管理代码和文档。利用 GitHub Pages 功能，我们能够轻松托管静态网站，并通过全球域名进行访问，使全球用户能够便捷获取所需信息。在遵循开源协议的前提下，开发者可以在 GitHub 上 fork 他

人的项目，进行二次开发或提交修改请求，实现代码的共同协作。同时，GitHub 为每一位用户提供了一个自己的专属域名，可用于公开展示个人项目及作品，进一步促进交流与合作。下面，我们就来实现一下如何将本项目进行 GitHub 的远程托管，我将分以下三步进行阐述。

### 11.3.1 创建空的远程代码仓库

进入个人的代码仓库后，点击页面右上角的加号键，即可创建新的代码仓库。然后，在这里要注意一点是，我们所创建的仓库名一定要与我们的用户名一致（如图 11.4 所示）。因为 GitHub 所提供的给我们的域名，就是由我们的用户名所组成的。只有严格遵循了这个原则，我在对域名进行访问时，才能呈现我们想要的页面信息。

#### 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 *
 NieYuXin19170109270	/ NieYuXin19170109270.github.io
<small>✓ NieYuXin19170109270.github.io is available.</small>	

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

图 11.4 GitHub 创建空仓库

### 11.3.2 设置本地仓库与远程代码仓库的连接以及推送

创建好远程仓库之后，我们下一步就是将我们本地的 Git 仓库与原曾的 GitHub 仓库进行初步的连接并进行推送了，具体操作步骤如命令 11.3 所示。若出现如图 11.5 所示情况，则表明推送成功了。

```
1. 向 README.md 文件中追加内容“WebUI 应用的远程 http 服务设置”
$ echo “WebUI 应用的远程 http 服务设置”>> README.md

2. 将 README.md 文件添加到 Git 的暂存区
$ git add README.md

3. 将暂存区的文件提交到本地仓库，并添加一条提交信息“这是我第一次把代码仓库上传到 GitHub 平台”
$ git commit -m “这是我第一次把代码仓库上传到 GitHub 平台”

4. 将当前所在分支重命名为 main
$ git branch -M main

5. 向 Git 仓库中添加远程仓库，初步建立连接
$ git remote add origin
https://github.com/NieYuXin19170109270/NieYuXin19170109270.github.io.git

6. 将代码推送至远程仓库
$ git push -u origin main
```

命令 11.3

```

25897@LAPTOP-UHCF3JUO MINGW64 /d/webUI (main)
$ git push -u origin main
Enumerating objects: 41, done.
Counting objects: 100% (41/41), done.
Delta compression using up to 12 threads
Compressing objects: 100% (39/39), done.
Writing objects: 100% (41/41), 1.86 MiB | 762.00 KiB/s, done.
Total 41 (delta 18), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (18/18), done.
To https://github.com/NieYuxin19170109270/NieYuxin19170109270.github.io.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

```

图 11.5 代码推送至远程仓库

为了增加本论文的真实性和直观性，我已经将本论文中所涉及的每个阶段的代码都推送到了我所使用的 GitHub 仓库中，以供各位读者通过网站的域名或扫描二维码（如表 11.1 所示）进行查阅。

表 11.1 项目网址以及二维码

项目版本	项目功能	项目网址	项目二维码
第一版	“三段论”式内容设计概要开发	<a href="https://nietuxin19170109270.github.io/exp/WebUI_v1.0.html">https://nietuxin19170109270.github.io/exp/WebUI_v1.0.html</a>	
第二版	网页窄屏代码实现以及个性化 UI 设计中的鼠标模型 1.0 的探究，利用鼠标监听事件，显示鼠标坐标	<a href="https://nietuxin19170109270.github.io/exp/WebUI_v2.0.html">https://nietuxin19170109270.github.io/exp/WebUI_v2.0.html</a>	
第三版	个性化 UI 设计中的鼠标模型 2.0-触屏的实现，利用鼠标监听事件，对页面元素进行拖动，同时显示拖动距离	<a href="https://nietuxin19170109270.github.io/exp/WebUI_v3.0.html">https://nietuxin19170109270.github.io/exp/WebUI_v3.0.html</a>	
第四版	个性化 UI 设计中的键盘控制 1.0- 利用 keydown 和 keyup 键盘底层事件显示所按下键的字符，以及 key 和 keyCode 值	<a href="https://nietuxin19170109270.github.io/exp/WebUI_v4.0.html">https://nietuxin19170109270.github.io/exp/WebUI_v4.0.html</a>	

第五版	个性化 UI 设计中的键盘控制 2.0-利用 for 循环和条件判断，进行筛选，只显示单个字符	<a href="https://nietuxin19170109270.github.io/exp/WebUI_v5.0.html">https://nietuxin19170109270.github.io/exp/WebUI_v5.0.html</a>	
第六版	个性化 UI 设计中的键盘控制 3.0-利用监听事件，对特殊键 Back 和 Enter 进行处理，实现文本的删除与换行	<a href="https://nietuxin19170109270.github.io/exp/WebUI_v6.0.html">https://nietuxin19170109270.github.io/exp/WebUI_v6.0.html</a>	
最终版	综合前 6 版的内容，设计个性化网站。具体功能有两点：1.显示我喜欢的设计师的部分作品信息（为了使得内容更加丰富，已添加翻页功能）2.附有键盘响应区，可进行文本编辑	<a href="https://nietuxin19170109270.github.io">https://nietuxin19170109270.github.io</a>	

## 参考文献

- [1] W3C. W3C's history. W3C Community. [EB/OL]. <https://www.w3.org/about/>. 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: 6
- [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
- [8] Martina Seidl, Marion Scholz, et al. UML @ Classroom An Introduction to Object-Oriented Modeling [M]. Springer International Publishing Switzerland,2015
- [9] Matti Tedre, Peter J.Denning. Computational Thinking, A Professional and Historical Perspective. Computational Thinking in Education A Pedagogical Perspective[C]. Routledge Taylor & Francis Group, 2022:1-17

## Web client based customizd UI design and Programming

**Abstract:** Over the past decade, software development technologies centered around HTML5 as the core web standard have been widely utilized in various fields of application software development due to its cross-platform and open-source advantages. Through analyzing the task of this thesis, this project selects HTML5 web client technology as its technical route, undertaking research and practice in program design and software development. By extensively consulting relevant technical books, developer forums, and literature, a personalized user interface (UI) application is designed and developed. In the development process, HTML language is used for content modeling, CSS language for UI appearance design, and JavaScript language for implementing UI interactive functions. Apart from directly using the lowest-level APIs of web clients, every line of code in this project is manually written without importing any code from others (frameworks and libraries). The project also adopts responsive design programming, intelligently adapting to the diversified needs of users' screens in the mobile internet era. Additionally, a considerable amount of object-oriented programming thinking is applied, such as building a universal pointer model with code, which implements control over both mouse and touchscreen with just one set of code, achieving high-quality code, which is also a highlight of this project. From the perspective of project management, the project adopts an incremental development model, gradually refining through six iterations of incremental refactoring (A: Analysis, D: Design, I: Implementation, T: Testing), accomplishing the design, development, and testing of the project pleasantly. From the perspective of code open-sourcing and sharing, the project utilizes Git tool for version management, refactors the code six times during the lengthy development process, and formally commits the code. Additionally, the code is modified and committed twice during testing. Finally, using Git Bash tool, the project's code repository is uploaded to the famous GitHub, and then utilizing GitHub's provided HTTP server, the project achieves deployment of UI applications on the global Internet. We can conveniently access this program across platforms efficiently through the address and QR code provided.

**Keywords:** Web client technology; HTML5; CSS; JavaScript; GitHub