

成绩:

# 江西科技师范大学

## 毕业论文（设计）

题目（中文）：基于 web 客户端技术的个性化 UI 设计与实现

（外文）：Personalized UI design and implementation

院（系）：元宇宙产业学院

专 业：计算机科学与技术

学生姓名：华勋

学 号：20213607

指导教师：李建宏

2024 年 6 月 19 日

# Web 技术的个性化 UI 设计

**摘要:** Web 技术凭借其跨平台优势, 成为流行的软件开发手段。为适应移动互联网时代前端需求, 本项目研究了 Web 客户端技术, 广泛查阅了资料和文献, 特别是 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。通过集成这些技术, 并应用本科课程知识, 项目实现了个性化用户界面(UI)。该界面采用响应式设计, 通过媒体查询、灵活网格布局和 CSS 框架(如 Bootstrap), 动态适配 PC 端和移动端设备, 自动调整布局, 提供一致优质的用户体验。在功能方面, 项目以 DOM 技术和事件驱动模式为基础, 实现了对鼠标、触屏、键盘等事件的流畅响应。通过应用 DOM, 项目能动态操作网页内容, 实时更新界面, 增强交互性。同时, 利用 JavaScript 的事件监听机制, 程序灵敏响应用户输入, 设计了一个模拟指向性设备操作的对象模型。项目不仅处理简单的点击、滚动事件, 还支持复杂手势操作, 提供丰富的交互体验。

**关键词:** Web 技术; JavaScript; 个性化设计; 双客户端

## Personalized UI design for Web technology

**Abstract:** Web technology, with its cross-platform advantage, has become a popular means of software development. To meet the front-end demands of software projects in the mobile internet era, this project focuses on Web client technology. We extensively reviewed technical materials and related literature, especially the technical practice articles from Mozilla's MDN community, exploring the basic techniques and skills of HTML content modeling, CSS style design, and JavaScript functional programming. By integrating these technologies and applying knowledge from undergraduate courses, the project achieved a personalized user interface (UI). This interface uses responsive design, leveraging media queries, flexible grid layouts, and CSS frameworks (such as Bootstrap) to dynamically adapt to both PC and mobile devices, automatically adjusting the layout to provide a consistent and high-quality user experience. Functionally, the project is based on DOM technology and event-driven programming, enabling smooth support for mouse, touch, and keyboard events. By applying DOM, the project can dynamically manipulate web content, updating the interface in real-time and enhancing interactivity. Additionally, using JavaScript's event listener mechanism, the program responds sensitively to user inputs, designing an object model to simulate the operation of pointing devices. The project not only handles simple click and scroll events but also supports complex gesture operations, providing a rich interactive experience.

**Key words:** Web technology; JavaScript; Personalization design; Different clients

# 引言

## 项目概述

在现代 Web 开发中，用户界面（UI）设计是至关重要的部分。随着用户需求的多样化和技术的迅猛发展，个性化 UI 设计已成为提升用户体验的关键因素。个性化 UI 能够根据用户的偏好和行为进行动态调整，确保在不同设备和屏幕尺寸上提供一致且流畅的交互体验。这涉及实现响应式设计原则、使用自适应布局，并利用高级 CSS 框架和 JavaScript 库创建动态的、以用户为中心的界面。通过专注于个性化 UI 设计，开发人员能够显著提高用户的可用性、参与度和整体满意度，以适应日益多样化的数字环境。

本项目旨在基于 Web 客户端技术开发一个个性化的用户界面。通过利用现代 Web 技术，如 HTML5、CSS3、JavaScript 以及响应式设计原则，项目将实现一个能够自适应不同用户端的操作与用户进行交互方式的 UI 系统。该系统不仅支持传统的鼠标和键盘输入，还能够兼容触屏操作，为用户提供适配当前使用设备的良好使用体验。

本项目的主要目标包括：

- 响应式设计：**设计和实现能够自适应不同屏幕尺寸（如手机、平板和桌面设备）的用户界面，确保在各种设备上的一致性和可用性。利用媒体查询、灵活网格布局和 CSS 框架（如 Bootstrap），确保界面在不同分辨率下自动调整布局。
- 个性化交互：**根据用户的操作习惯和偏好，动态调整 UI 布局和交互方式，提供个性化的用户体验。通过分析用户行为数据，使用 JavaScript 和相关库实现实时界面更新和动态内容加载。
- 多种输入支持：**开发支持鼠标、键盘和触屏的通用交互模式，确保用户能够在各种设备上进行操作。实现对 DOM 事件的全面处理，包括点击、触摸、滚动和键盘输入，提升交互的灵敏度和响应速度。

4. **高质量代码和版本管理：**采用良好的编码规范和版本控制工具（如 Git），确保代码的可维护性和可扩展性。使用模块化设计和代码复用策略，结合持续集成和自动化测试，保障项目的稳定性和可靠性。

通过这个项目，我们不仅希望展示现代 Web 技术在个性化 UI 设计中的应用，还希望探索 and 解决在多设备、多交互模式下的设计和开发挑战。最终，项目将为开发人员提供一个参考案例，展示如何利用先进的 Web 客户端技术实现高质量、个性化的用户界面。

## 研学计划

为确保项目的顺利进行并实现预期目标，收集和阅读相关文献，了解 Web 客户端技术和个性化 UI 设计的最新发展。分析已有案例，明确项目的功能需求和用户体验目标。制定系统架构设计，确定整体框架和模块划分。选择合适的前端框架（如 React 或 Vue）、CSS 预处理器（如 Sass 或 Less）及版本控制工具（如 Git）。开发系统原型，初步实现主要界面和基本功能。收集初步用户反馈，调整和优化原型设计。实现项目的核心功能模块，如用户管理、数据处理、个性化推荐等。进行模块集成测试，确保功能的正确性和稳定性。设计和实现响应式 UI，适配不同屏幕尺寸。优化系统性能，提高响应速度和用户体验。

## 研究方法

为了确保项目的科学性和系统性，本研究采用了多种研究方法，包括文献研究、需求分析、原型开发、增量迭代开发以及用户测试。具体方法如下：获取最新的 Web 客户端技术和个性化 UI 设计的理论基础和实践经验。查阅学术论文、技术报告、专业书籍和在线资源，梳理现有研究成果和案例分析。文献综述报告，涵盖技术背景、发展趋势和关键技术点。明确项目的功能需求和用户体验目标。采用访谈、问卷调查和用户故事等方法，收集潜在用户的需求和期望。需求分析文档，详细描述用户需求、功能要求和设计目标。制定系统的整体架构和详细设计。使用 UML（统一建模语言）进行系统建模，设计系统架构图、模块划分和接口定义。系统设计文档，包括架构设计、模块设计和数据库设计。通过原型展

示项目的主要功能和界面设计。采用前端开发框架和设计工具进行快速原型开发。系统原型和初步用户反馈报告。

# 技术总结

## Web 平台和客户端技术概述

Web 之父 Tim Berners-Lee 在发明 Web 的基本技术架构之后，成立了 W3C 组织。该组织在 2010 年后推出的 HTML5 国际标准，结合由欧洲 ECMA 组织维护的 ECMAScript 国际标准，几乎完美地实现了全球开发者统一开发平台的理想。时至今日，科学家与 Web 行业仍在致力于完善这一伟大且光荣的愿景。学习 Web 标准和技术，掌握编写 Web 程序和应用的相关工具，最终构建一套高质量代码的跨平台运行应用，是我的毕业设计项目的技术路线。

Web 应用的程序设计体系由三大语言有机组成：HTML、CSS 和 JavaScript。这三大语言的组合体现了人类社会化大生产分工的智慧，分别实现了对信息系统的结构（Structure）、外观（Presentation）和行为（Behavior）的描述与控制。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 可理解为 HTML 标记语言建模，View 可理解为用 CSS 语言实现外观，而 Controller 则可理解为用 JavaScript 结合前两个层次，进行微观和功能层面的代码控制。

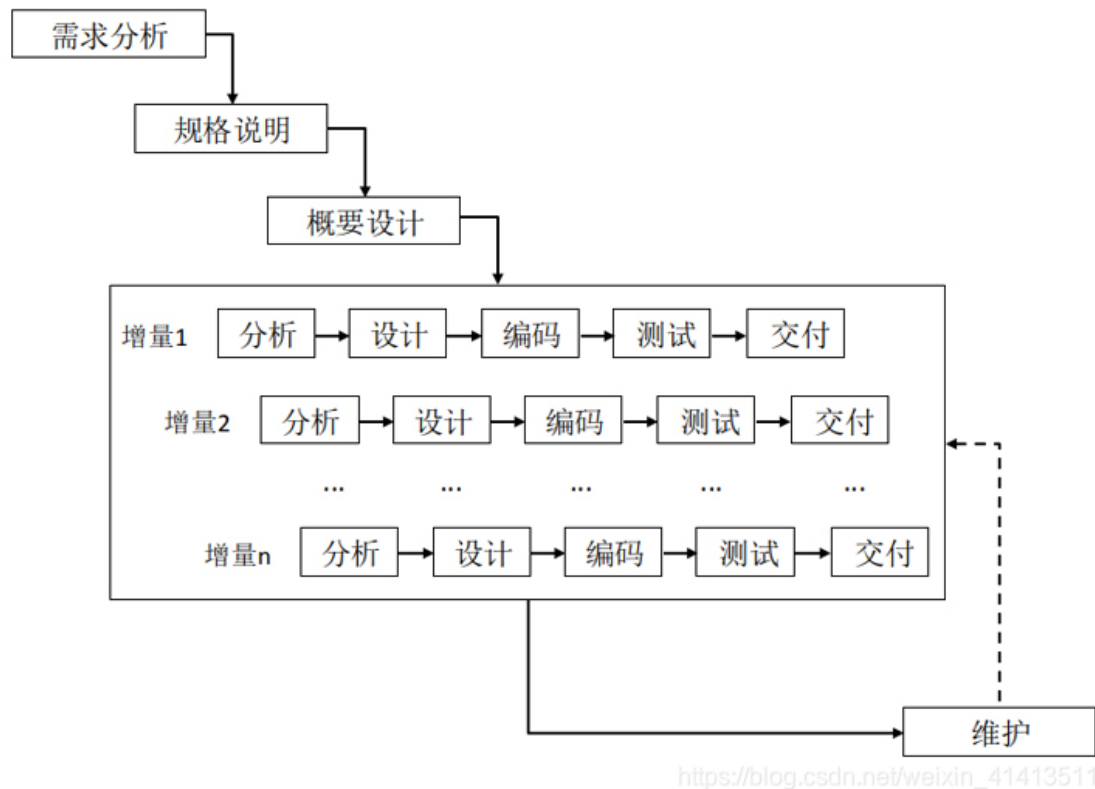
在现代 Web 开发中，基于 Web 客户端技术的个性化 UI 设计已成为热门话题。诸如 React、Vue.js 和 Angular 等前端框架提供了强大的 UI 组件和状态管理功能，极大地简化了前端开发的复杂性，并促进了组件化和模块化的开发方式。响应式设计使网站能够根据不同设备（如电脑、平板电脑和手机）和屏幕尺寸自动调整其布局和样式，通过媒体查询和流式布局等技术，网站在不同设备上均能实现良好的显示效果。增量式迭代开发是一种将项目分解为多个增量，逐步开发、测试和集成的开发模式。每个增量都包含完整的开发周期，有助于提高项目的灵活性、可控性和可维护性。采用 Git 等版本控制工具进行代码管理，能够有效协调多人开发，追踪代码修改历史，管理分支和解决冲突，为团队协作提供了良好的基础。通过深入理解和应用这些技术，可以显著提升 Web 应用的开发效率和质量。

## 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[5]。本项目中我一共做了六次项目的开发迭代，如下所示。

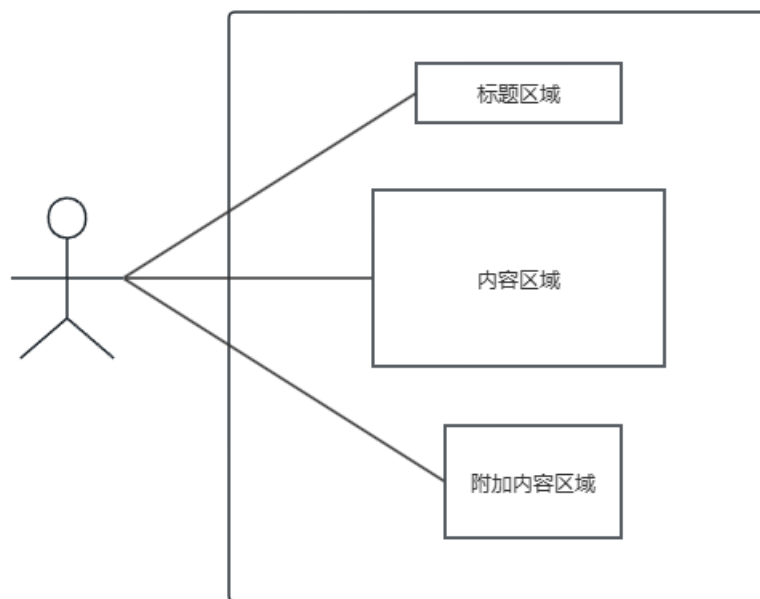


## 内容设计概要

### 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。具体用例如下图所示。





## 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 XXX 江西科技师范大学 2024-2025
</footer>
```

### 二、CSS 代码编写如下：

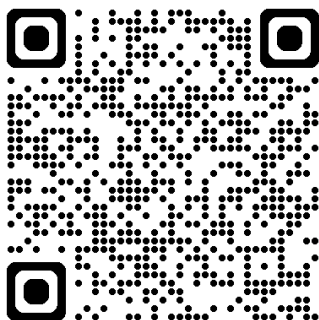
```
*{
    margin: 10px;
    text-align: center;
}
```

```
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 ;  
}
```

## 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的二维码，运行测试本项目的第一次开发的阶段性效果。

|   |
|---|
| 《我的毕设题目》  |
| 我的主题内容: ‘读好书、练思维、勤编程’ @masterLijh 计算思维系列课程  |
| CopyRight XXX 江西科技师范大学 2024-2025  |
| 其他章节: <a href="#">1.2</a> <a href="#">1.3</a> <a href="#">1.4</a> <a href="#">1.5</a> <a href="#">1.6</a> |



## 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 科师大华勋  
$ git config user.email 490216112@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
```

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
```

## 移动互联时代的 UI 开发初步——窄屏终端的响应式设计

### 分析和设计

随着移动设备的普及和网络技术的飞速发展，用户通过不同尺寸和类型的终端访问 Web 应用的需求日益增加。窄屏终端（如智能手机和平板电脑）在移动互联时代扮演着至关重要的角色。响应式设计成为适应这一趋势的重要技术手段，通过适配不同屏幕尺寸的 UI 设计，确保在各种设备上的一致性和优质用户体验。

### 项目的实现和编程

#### 一、HTML 代码编写如下：

```
<header>
  <p id="book">
    我的毕设题目
  </p>
</header>

<nav>
  <button>导航一</button>
  <button>导航二</button>
  <button>导航三</button>
</nav>
```

```
<main id = 'main'>
    软件内容区域
</main>
<footer>
    <p id="statusInfo">
        XXX @ 江西科技师范大学 2025
    </p>
</footer>
```

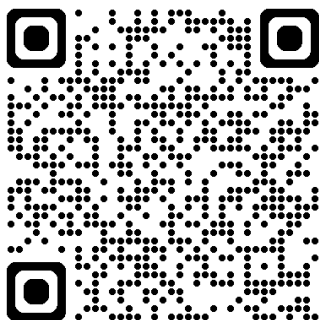
## 二、CSS 代码编写如下：

```
*{
    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%;  
}
```

## 项目的运行和测试

确保项目在不同设备和浏览器上的兼容性和性能是响应式设计的关键。在不同的浏览器（如 Chrome、Firefox、Safari、Edge）上测试 UI，确保一致的显示效果。在实际的移动设备上测试，确保 UI 在各种屏幕尺寸和分辨率上的表现。使用开发者工具（如 Chrome DevTools）中的设备模拟功能进行快速验证。使用工具（如 Lighthouse 或 PageSpeed Insights）分析页面的性能指标，如加载时间、交互延迟和渲染速度。实施代码分割和异步加载，提升应用的响应速度。通过用户测试收集实际用户的反馈，了解 UI 在不同设备上的使用体验。根据反馈进行调整和改进，提升用户满意度。



## 项目的代码提交和版本管理

编写好的代码，测试运行成功后，执行下面命令提交代码：

```
$ git add 1.2.html
```

## 个性化 UI 设计——鼠标模型

### 分析和设计

在 Web 应用中，鼠标作为主要的输入设备之一，对于 PC 端用户来说至关重要。设计和实现针对鼠标的个性化交互，能够提升用户体验，增强应用的易用性和吸引力。本节将详细介绍如何进行基于鼠标的个性化 UI 设计，包括分析和设计、实现和编程、运行和测试，以及代码提交和版本管理。

### 项目的实现和编程

#### 一、HTML 代码编写如下：

```
<header>
  <p id="book">
    《我的毕设题目》
  </p>
</header>

<nav>
  <button>导航 1</button>
  <button>导航 2</button>
  <button>导航 3</button>
</nav>

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

```
</div>

</main>

<footer>

Copyright XXX 江西科技师范大学 2024--2025

</footer>

<div id="aid">

    <p>用户键盘响应区</p>

    <p id="keyboard"></p>

</div>
```

## 二、CSS 代码编写如下：

```
*{

    text-align: center;

    box-sizing: border-box ;

}

header,main,div#bookface,nav,footer{

    margin: 1em;

}

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%;
}
body{
position:relative ;
}
#aid{
    position: absolute;
    border: 3px solid blue;
    top: 0.5em;
    left: 600px;
}
#bookface{
    width: 80%;
    height: 80%;
    border: 1px solid red;
    background-color: blanchedalmond;
    margin:auto;
}
```

## 项目的运行和测试

为了确保基于鼠标的个性化 UI 功能正常运行并提供优质的用户体验，编写单元测试，确保每个鼠标事件处理函数和交互逻辑都能正常工作。进行集成测试，验证各个模块之间的交互和整体功能的正确性。在不同的浏览器（如 Chrome、Firefox、Safari、Edge）上测试鼠标交互，确保一致的行为。在各种设备上测试，确保在不同屏幕尺寸和分辨率下，鼠标操作的响应和显示效果正常。



## 项目的代码提交和版本管理

编写好的代码，测试运行成功后，执行下面命令提交代码：

```
$ git add 1.3.html
```

# 对触屏和鼠标的通用交互操作的设计开发

## 分析和设计

在现代 Web 应用中，用户可能通过各种设备访问，包括带有触摸屏的移动设备和使用鼠标的桌面电脑。为了提供一致的用户体验，设计和开发能够同时适应触屏和鼠标操作的通用交互界面是至关重要的。以下内容将详细探讨如何分析、设计和实现这一类交互操作。

## 项目的实现和编程

### 一、HTML 代码编写如下：

```
<header>
  <p id="book">
    《我的毕设题目》
  </p>
</header>

<nav>
  <button>向前</button>
  <button>向后</button>
  <button>其他</button>
</nav>

<main id="main">
  <div id="bookface">
    这是书的封面图<br>
    在此对象范围拖动鼠标(本例触屏无效)
  </div>
</main>

<footer>

  Copyright XXX 江西科技师范大学 2024--2025
```

```
</footer>

<div id="aid">

  <p>用户键盘响应区</p>

  <p id="keyboard"></p>

</div>
```

## 二、CSS 代码编写如下：

```
*{
    margin: 10px;
    text-align: center;
}

header{
    border: 3px solid green;
    height: 10%;
    font-size: 1em;
}

nav{
    border: 3px solid green;
    height: 10%;
}

main{
    border: 3px solid green;
    height: 70%;
    font-size: 0.8em;
    position: relative;
}

#box{
    position: absolute;
    right: 0;
```

```
    width: 100px;
}

footer{
    border: 3px solid green;
    height:10%;
    font-size: 0.7em;
}

body{
    position: relative;
}

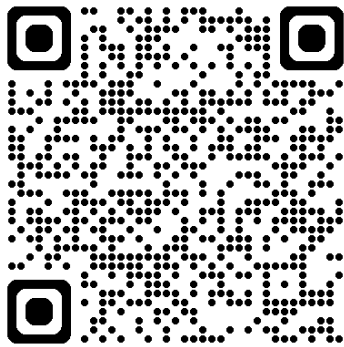
button{
    font-size: 1em;
}

#aid{
    position: absolute;
    border: 3px solid blue;
    top:0px;
    left:600px;
}

#bookface{
    position: absolute;
    width: 80%;
    height: 80%;
    border: 1px solid red;
    background-color: blanchedalmond;
    left: 7% ;
    top: 7% ;
}
```

## 项目的运行和测试

在确保触屏和鼠标的通用交互操作正常运行时，应该进行全面的测试和优化：编写测试用例，确保各个交互操作在不同的输入设备上都能正常工作。进行集成测试，验证多个交互操作和模块之间的协调工作。在主流的桌面浏览器和移动浏览器中进行测试，确保一致的行为和体验。在各种设备上测试，包括不同品牌和型号的智能手机、平板电脑和 PC。邀请用户在实际设备上使用应用，观察他们的操作行为和反馈。通过问卷调查或用户访谈，收集关于交互操作的使用体验和改进建议。分析和优化事件处理的响应速度，确保在触屏和鼠标操作下都能提供流畅的体验。



## 项目的代码提交和版本管理

编写好的代码，测试运行成功后，执行下面命令提交代码：

```
$ git add 1.4.html
```

# UI 的个性化键盘交互控制的设计与开发

## 分析和设计

键盘交互在 Web 应用的用户体验中扮演着重要的角色，尤其是在桌面环境中。个性化的键盘交互设计不仅能提升用户操作的效率，还能为特定的应用场景提供便利。本节将详细介绍如何设计、实现和管理针对键盘的个性化 UI 交互控制。

## 项目的实现和编程

### 一、HTML 代码编写如下：

```
<header>
  <p id="book">
    《我的毕设题目》
  </p>
</header>

<nav>
  <button>向前</button>
  <button>向后</button>
  <button>其他</button>
</nav>

<main id="main">
  <div id="bookface">
    这是书的封面图<br>
    在此对象范围拖动鼠标/滑动触屏<br>
    拖动/滑动超过 100 像素，视为有效 UI 互动！
  </div>
</main>
<footer>
```

Copyright XXX 江西科技师范大学 2024--2025

</footer>

<div id="aid">

<p>用户键盘响应区</p>

</div>

## 二、CSS 代码编写如下：

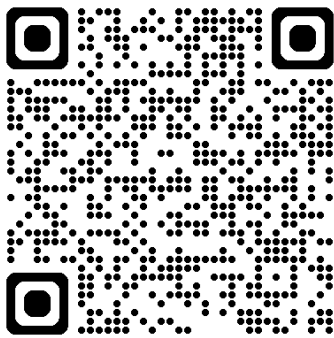
```
* {  
    margin: 10px;  
    text-align: center;  
}  
  
header {  
    border: 3px solid green;  
    height: 10%;  
    font-size: 1em;  
}  
  
nav {  
    border: 3px solid green;  
    height: 10%;  
}  
  
main {  
    border: 3px solid green;  
    height: 70%;  
    font-size: 0.8em;  
    position: relative;  
}
```



```
#box {  
    position: absolute;  
    right: 0;  
    width: 100px;  
}  
  
footer {  
    border: 3px solid green;  
    height: 10%;  
    font-size: 0.7em;  
}  
  
body {  
    position: relative;  
}  
  
button {  
    font-size: 1em;  
}  
  
#aid {  
    position: absolute;  
    border: 3px solid blue;  
    top: 0px;  
    left: 600px;  
}  
  
#bookface {  
    position: absolute;  
    width: 80%;
```

```
height: 80%;  
border: 1px solid red;  
background-color: blanchedalmond;  
left: 7%;  
top: 7%;  
}
```

## 项目的运行和测试



## 项目的代码提交和版本管理

编写好的代码，测试运行成功后，执行下面命令提交代码：

```
$ git add 1.5.html
```

# 用 gitBash 工具管理项目的代码仓库和 http 服务器

## 跨世纪的经典 Bash 工具

Bash (Bourne Again SHell) 是由 GNU 项目开发的一个广泛应用于 Unix 和类 Unix 操作系统的命令行解释器和脚本语言。自 1989 年发布以来, Bash 已成为系统管理员、开发者和高级用户进行系统管理和脚本编程的首选工具。它融合了早期 Unix shell 的功能, 并进行了大量增强, 使得 Bash 成为一个跨世纪的经典工具。

Bash 是一款既可以作为命令行解释器, 也可以作为脚本编程语言的工具。在 Unix 系统中, shell 是用户与操作系统之间的接口, 它允许用户输入命令并得到系统的响应。Bash 作为一个 shell, 它不仅支持基本的命令执行, 还提供了强大的脚本编写功能。这使得用户可以编写脚本来自动化日常任务, 如文件管理、系统维护和软件安装等。

Bash 的名字来源于 Bourne shell 的增强版本 ("Bourne Again SHell")。Bourne shell (sh) 是由斯蒂芬·布尔恩在 1970 年代开发的, 它是 Unix 操作系统的标准 shell。Bash 继承了 Bourne shell 的许多特性, 并在此基础上进行了扩展。它的发布不仅标志着 Unix shell 的一个重大进步, 也成为了 GNU 操作系统的重要组成部分。在过去的几十年里, Bash 不断发展和演变。它的兼容性和功能性使其成为 Linux 和 macOS 等系统的默认 shell。随着版本的升级, Bash 不断引入新的功能和改进, 如数组支持、改进的错误处理、增强的字符串操作等。

在现代计算环境中, Bash 仍然发挥着重要作用。虽然图形用户界面 (GUI) 变得越来越流行, 但命令行界面 (CLI) 在许多任务中仍然具有无可替代的效率和灵活性。Bash 提供的强大命令行功能和脚本能力, 使其成为自动化、系统管理和开发工作的重要工具。特别是在 DevOps 和云计算领域, Bash 脚本被广泛用于配置管理、部署自动化和系统监控等任务。无论是编写启动脚本, 还是配置服务器环境, Bash 都是开发者和系统管理员的得力助手。对于 Windows 用户来说, GitBash 是一个将 Bash 的强大功能带到 Windows 平台的重要工具。它集成了 Git 和 Bash, 提供了一个仿 Linux 的环境, 使得 Windows 用户能够享

受到 Bash 的强大功能。GitBash 使得开发者可以在 Windows 系统上方便地使用 Bash 命令和脚本，同时也可以管理 Git 仓库，进行版本控制。

## 利用 gitBash 工具对本项目进行管理

### 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，包括开发者的名字和电子邮箱等。具体操作步骤如下，通过 gitBash 命令行执行以下命令（如所示）。

```
$ cd /d/git——远程管理文件/github
$ mkdir WebUI
$ cd WebUI
$ git init
$ git config user.name 科师大华勋
$ git config user.email 490216112@qq.com
$ touch index.html myCss.css myJs.js
```

### 项目代码的提交与上传

在编写并成功运行 index.html、myCss.css 和 myJs.js 的代码后，执行以下操作（）。其中，成功提交文件后，gitBash 的反馈如图所示。

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

```
Administrator@WIN-SND7LHJPSHV MINGW64 /d/git——远程管理文件/github/WebUI (master)
$ git add index.html myCSS.css myJs.js

Administrator@WIN-SND7LHJPSHV MINGW64 /d/git——远程管理文件/github/WebUI (master)
$ git commit -m '项目第一版：三段论式的内容设计概要'
[master 5da63d8] 项目第一版：三段论式的内容设计概要
1 file changed, 351 insertions(+)
create mode 100644 myJs.js
```

## 项目日志的反馈查询

在项目开发过程中，使用 `git log` 命令查看和管理 Git 的提交日志是很重要的。

```
Administrator@WIN-SND7LHJPSHV MINGW64 /d/git—远程管理文件/github/WebUI (master)
$ git log
commit 3d564816fe618b837025fb0cd65474427e15c62b (HEAD -> master)
Author: youzelian <1431850082@qq.com>
Date:   Wed Jun 12 10:26:29 2024 +0800

    第五次提交：UI的个性化键盘交互控制的设计与开发

commit 722cd5e5c009cc0612f71daf2438e2005e16dbf2
Author: youzelian <1431850082@qq.com>
Date:   Wed Jun 12 10:20:57 2024 +0800

    第四次提交：对触屏和鼠标的通用交互操作的设计开发

commit 1439c4d8c9d697968fcf2186cabd849991f50c93
Author: youzelian <1431850082@qq.com>
Date:   Wed Jun 12 10:13:05 2024 +0800

    第三次提交：个性化UI设计——鼠标模型

commit e042f534996fdf254d69dbf679139b5e018fd9f9
Author: youzelian <1431850082@qq.com>
Date:   Wed Jun 12 09:53:59 2024 +0800

    第二次提交：移动互联时代的UI开发初步——窄屏终端的响应式设计
... skipping ...
commit 3d564816fe618b837025fb0cd65474427e15c62b (HEAD -> master)
Author: youzelian <1431850082@qq.com>
Date:   Wed Jun 12 10:26:29 2024 +0800

    第五次提交：UI的个性化键盘交互控制的设计与开发
```

## 通过 GitHub 平台实现本项目的全球域名

通过 GitHub 平台实现本项目的全球域名 GitHub 是全球最受欢迎的代码托管平台之一，不仅提供强大的版本控制和协作工具，还为开发者提供便捷的方式来托管静态网站——这就是 GitHub Pages。通过 GitHub Pages，您可以将项目托管在 GitHub 上，并且获得一个免费且全球可访问的域名，使项目能够被他人访问和分享。

## 创建空的远程代码仓库

点击主页右上角的+号的下拉菜单的 New repository 进行创建空的远程代码仓库。

## 设置本地仓库与远程代码仓库的连接并推送

创建好仓库后，我们需要将本地与 GitHub 创建的仓库进行连接。具体命令如下所示。

```
$ git init
```

```
$ git remote add origin https://github.com/huaxun/WebUI.git
```

