

计算机与信息类专业创新实践报告

**题 目**  网页计算器

**姓 名**  陈嘉乐

**学 号**  2021218152

**专 业**  计算机科学与技术

**班 级**  21-3班

**完成日期**  2023/7/4

|  |  |  |  |
| --- | --- | --- | --- |
| **计算机与信息类专业创新实践项目验收细则（占比40%）** | | | |
| **成绩等级** | **具体表现** | **教师评分** | |
| 优秀（100-85] | 1）能够在规定时间内完成项目，项目具有完整性，交互设计优秀；2）工作量充足；3）能正确、完整回答教师提问； | □ |  |
| 良好（85-75] | 1）能够在规定时间内完成项目，且项目具有较好的完整性，交互设计较为优秀；2）工作量较为充足；3）能较为正确、完整回答教师提问； | □ |  |
| 中等（75-65] | 1）能够在规定时间内完成项目，项目完整性较好，交互设计一般；2）工作量一般； | □ |  |
| 及格（65-60] | 1）虽完成项目编码但未能在规定时间内完成的； | □ |  |
| 不及格（<60） | 1）未进行验收的； | □ |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **计算机与信息类专业创新实践报告评分细则（占比60%）** | | | |
| **成绩等级** | **具体表现** | **教师评分** | |
| 优秀（100-85] | 1）报告格式完美，能按照规定格式撰写报告，能充分采用图表说明问题，章节设计优秀，工作量饱满； | □ |  |
| 良好（85-75] | 1）报告格式良好，格式正确，采用了图表来说明问题，章节设计良好，工作量较为饱满； | □ |  |
| 中等（75-65] | 1）报告格式较好，章节设计较好，工作量一般； | □ |  |
| 及格（65-60] | 1）虽提交报告但未能在规定时间内完成的； | □ |  |
| 不及格（<60） | 1）未提交报告的； | □ |  |

教师签名：

#### 1 题目

网页计算器

#### 2 需求分析与设计

##### 2.1 需求分析

在全球数学运算迅速发展的背景下，为了满足用户对十进制和二进制计算的需求和便捷，提高生产力和效率，设计并实现了一个网页十进制和二进制计算器。该系统能够根据用户选择的计算器模式，根据用户输入的计算数字和符号，在线实时进行相应的数学计算。

##### 2.2 系统设计

系统为纯前端的设计思路，使用HTML、CSS和JavaScript实现十进制和二进制两种模式的计算功能，通过JavaScript对用户输入的被操作数据的获取和计算。系统的设计主要包括前端界面设计和计算功能设计两个部分。

#### 3 系统实现与使用方法

##### 3.1 系统开发环境

前端开发：HTML5、CSS3、JavaScript

开发工具：Sublime Text

##### 3.2 系统界面简介

网页计算器的界面采用简洁直观的设计，支持两种计算模式的切换：十进制和二进制。在十进制模式下，根据用户通过按钮输入相应的操作数和操作符号进行相应的计算，并将结果打印在计算器的文本框里。在二进制模式下，用户可以在输入框中输入需要进行计算的两个二进制数，系统将根据用户选择的运算类型实时计算出结果，并在页面上显示相应的计算结果。

##### 3.3 系统功能模块简介

模式选择模块：可以在十进制和二进制计算之间切换

计算机模块：分为文本框和按钮输入模块

计算机按钮模块：可以接受用户在计算器上的输入和符号操作

计算器文本框模块：可以显示当前的的运算式和结果

##### 3.4 系统实现

1. 计算器界面实现

通过div标签给页面分界布局，定好每一块的内容。在每一块容器里通过html标签添加要实现的功能，例如在一个div里嵌套两个div，一个用来做文本输入框和显示框，一个用来做计算器的键盘。在实现键盘的div里又可以嵌套数个div，每一个里面放入一个<button>。通过类选择器和id选择器分别对他们设置相应属性以美化界面。例如background属性有color、image、size等属性，padding、margin分别设置内边距和外边距，通过调整属性值和颜色等，让界面更美观舒适，让用户也能更清晰分辨和理解。

2. 计算器模式切换实现

通过<a>标签的属性和功能，给它提供点击后的跳转地址路径即可实现

<div class="ribbon">

<a href="./十进制.html"><span>十进制计算器</span></a>

<a href="./二进制.html"><span>二进制计算器</span></a>

</div>

3. 计算器计算功能的实现

通过JavaScript的事件以实现按钮点击后做出反应。给每个<button>里添加onclick 事件，即会在对象被点击时发生该事件。在js里设计好运算的功能实现函数，通过onclick与相应的函数操作连接。例如：appendToDisplay(value)实现得到用户输入的操作数并保存，与数字按钮连接，即可实现当用户按下数字后，运算器接收到相应的值。calculate()与“=”按钮连接，即可实现运算处理和结果显示。在calculate()里设计好运算操作逻辑，处理各种功能和类型的运算形式。

##### 3.5使用手册

1.打开网页计算器

2.默认模式为十进制计算器，可以通过上方标签进行模式切换，有十进制和二进制两种选择。点击相应标签即可切换到相应的模式。

3.十进制模式下，用户根据要计算的式子依次按下相应的按钮，例如算式：“（2+3）\*3.56”，只需在计算器上用鼠标依次按下相应按钮，屏幕上会依次显示出当前输入的全部式子，如果输入错误，可以通过DEL按键删除一位。最后按下“=”，屏幕上即可显示结果。

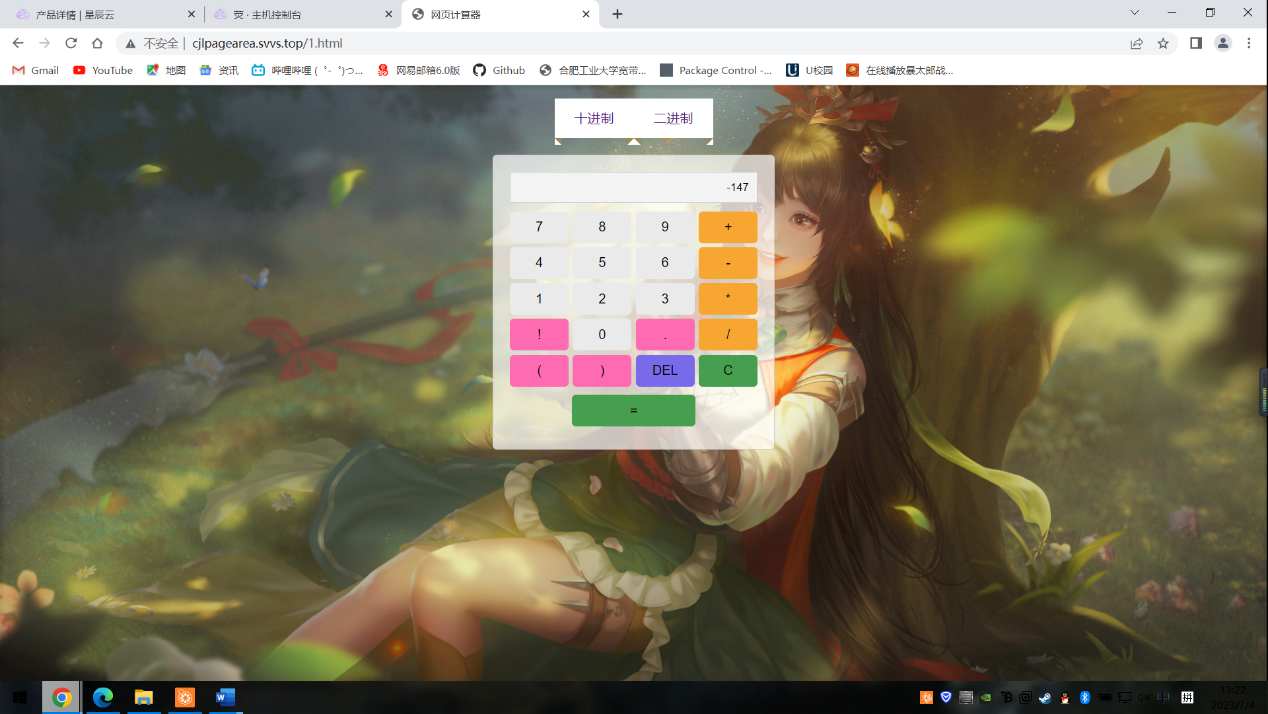
4.二进制模式下，用户根据要计算的计算数字，通过键盘输入到依次的两个文本框里，然后根据计算类型按下下方相应的运算按钮，即可在结果框里得到结果。

5.按下“C”即可清零。

#### 4运行实例与系统功能测试

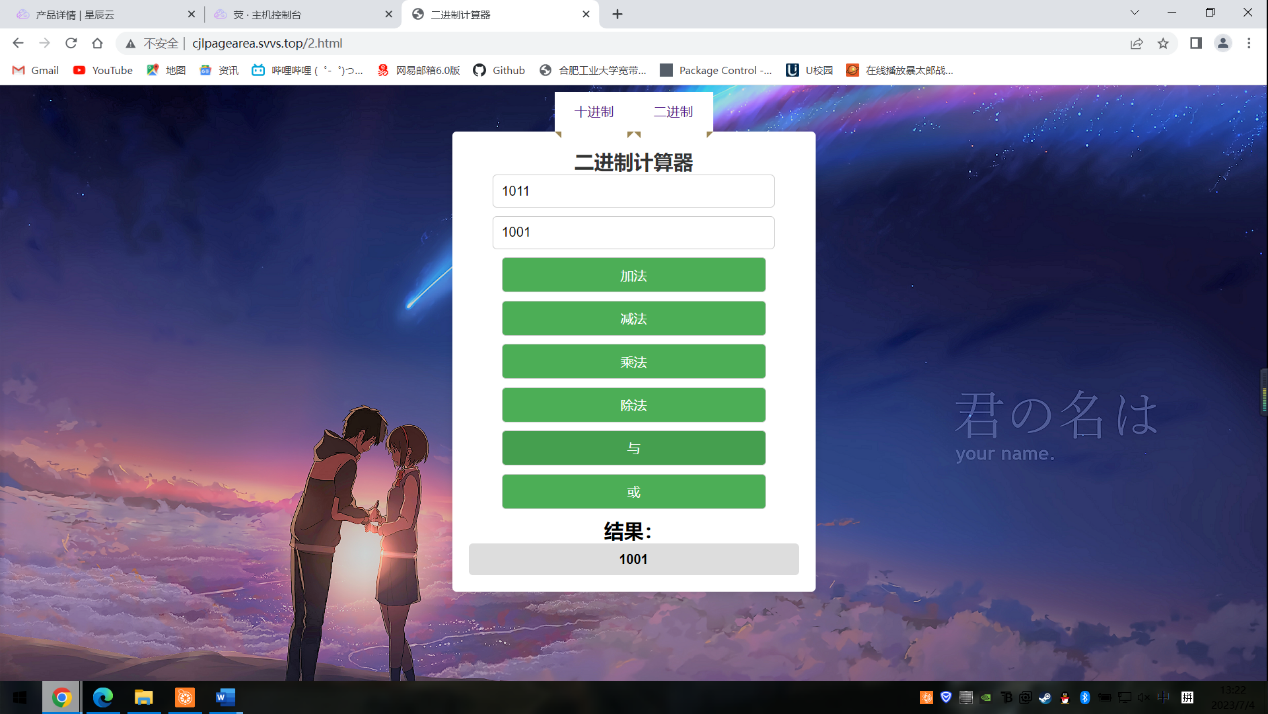
##### 4.1 系统功能测试

计算9+（6-3）\*4-84\*（6+4）/5

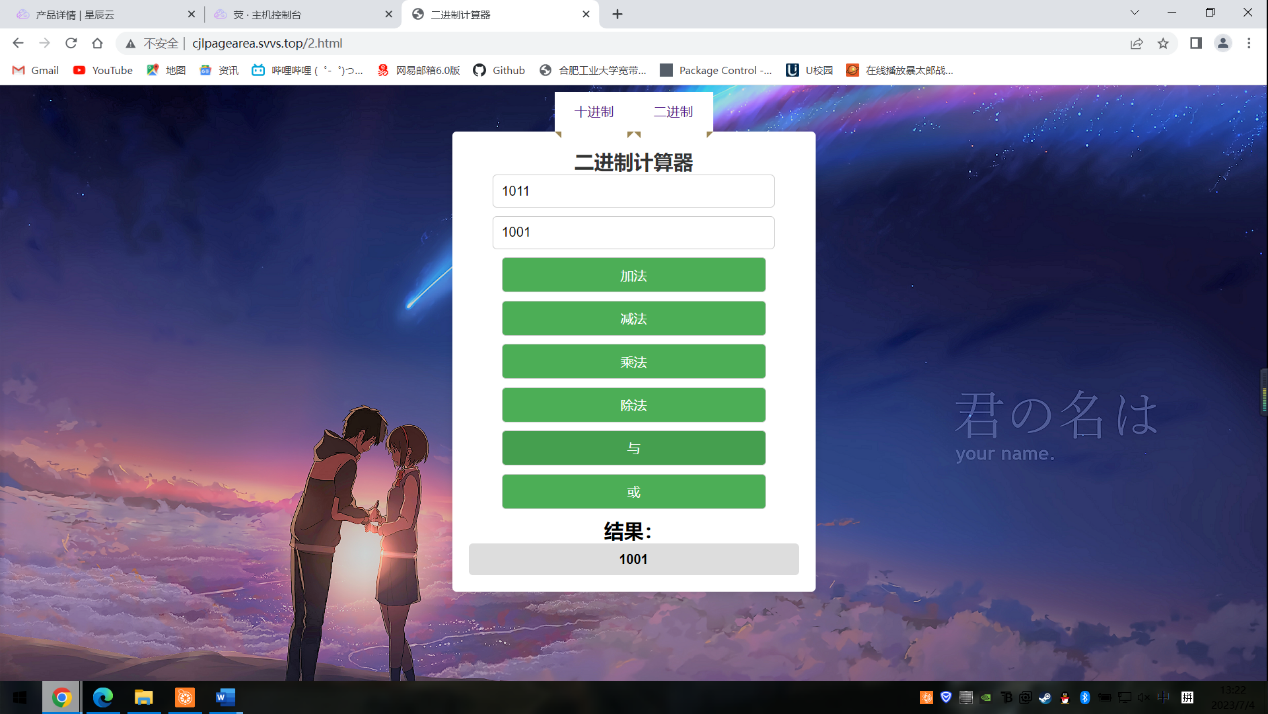
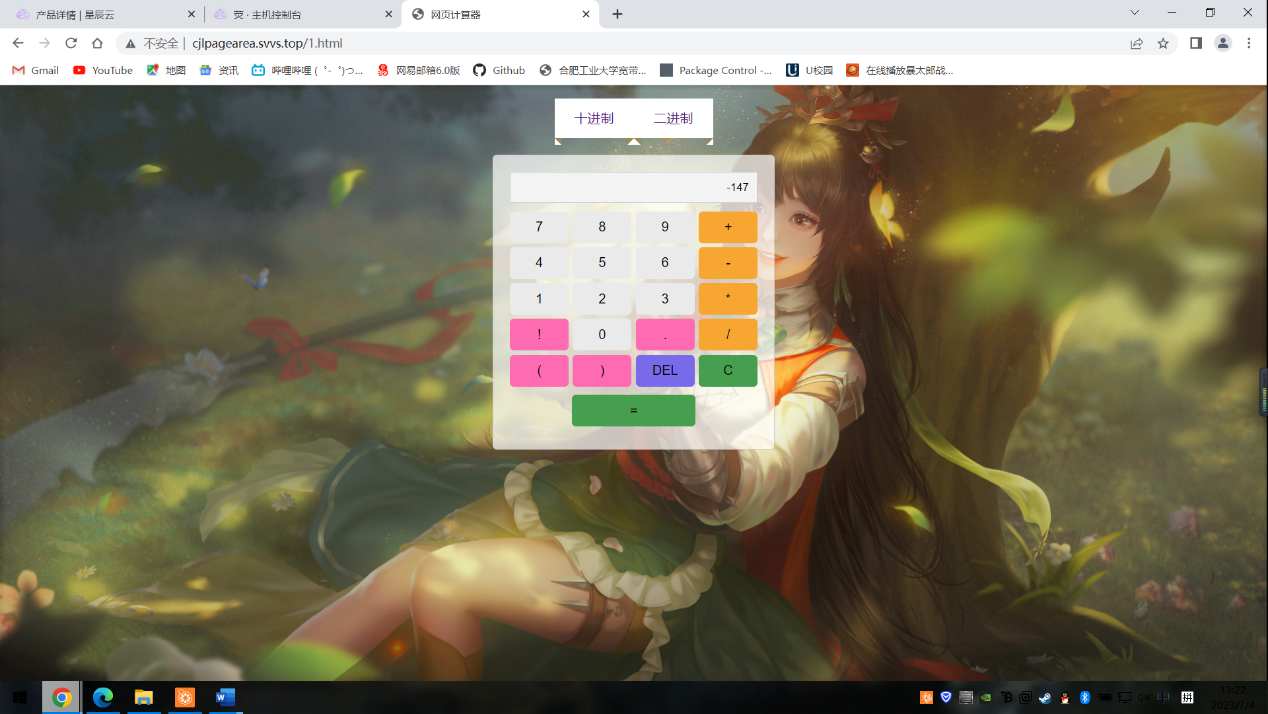


计算结果正确

二进制数1011&1001，结果为1001，计算正确



##### 4.2 运行实例



#### 5 总结与收获

##### 5.1 总结（心得体会）

在实现网页计算器过程中，我有一些心得体会：

1.HTML 结构：使用合适的 HTML 元素和布局，对于构建一个清晰、有组织的界面非常重要。在这个例子中，我使用了 div 元素和 CSS 网格布局，来创建一个简洁、易于使用的计算器界面。

2.JavaScript 逻辑：合理的 JavaScript 逻辑是实现计算器功能的关键。在这个例子中，我使用 JavaScript 处理按钮点击事件，并根据按钮的值执行相应操作，例如追加数字、运算符，清除结果，计算结果等。

3.输入验证与错误处理：在实际生产环境中，我们需要对用户的输入进行验证和处理错误情况。在实际开发中，应该对用户输入进行检查，确保输入的表达式是有效的，并通过适当的错误处理机制，如异常处理，向用户报告错误。

4.用户体验：为了提高用户体验，我们可以通过一些视觉效果和动画来增加交互性。进一步改进计算器的外观，并添加更多的功能，例如键盘支持、取反操作、取余等，来满足用户的需求。

5.注重安全性：在实际应用中，需要谨慎处理用户的输入，特别是在执行 eval() 等计算操作时。为了防止潜在的安全漏洞，应该对输入进行严格的过滤和验证，以避免执行恶意的代码。

6.可扩展性：在设计过程中，考虑到可能的功能扩展。通过添加函数，使得可以进行基本的四则运算和逻辑运算，且通过函数的模块化设计，方便在后续扩展时进行维护和添加新功能。

7.界面美化：通过选择合适的颜色、字体和布局，使网页更加美观和吸引人。使用圆角和阴影效果，使计算器区域更具深度感，并突出显示结果区域。

8.响应式设计：为了适应不同设备上的显示屏，采用了自适应的设计方法，使得网页能够在不同尺寸的屏幕上提供良好的显示效果。

通过考虑上述因素，设计了一个符合美观、易用和功能完善的二进制计算器网页。设计这个网页的过程也提醒我，在开发过程中，始终将用户体验放在首位，注重细节，力求提供更好的用户体验。

##### 5.2 收获

学习HTML、CSS和JavaScript是建立网页和网页交互性的基础，以下我的的学习收获：

HTML：

1.理解网页的结构和语义化：学习HTML可以让我更好地理解网页的结构，包括头部、导航栏、内容区域和底部等部分。此外，学习语义化的HTML标签可以提高网页的可读性和可访问性。

2.掌握HTML元素和属性：学习HTML可以让我熟悉各种HTML元素和属性的使用方式，如标题标签、段落标签、链接标签等，这些是构建网页内容的基础。容器标签<div>是我们用来让页面排版美观的利器，用它我可以更方便的操作各个标签元素。

CSS：

1.布局和样式控制：学习CSS可以让我掌握布局和样式控制技巧，如盒模型、定位、选择器和样式属性等。通过CSS，我可以为网页添加各种样式，包括颜色、字体、边框和背景等，让网页更具吸引力和专业性。

2.响应式设计：学习CSS的响应式设计技术能够让我创建适应不同屏幕尺寸（如电脑、平板和手机）的网页布局，提供优秀的用户体验。

JavaScript：

1.实现交互和动态效果：学习JavaScript可以使我具备为网页添加交互性和动态效果的能力。例如，我可以创建表单验证、幻灯片、下拉菜单、响应用户点击等功能，提升用户与网页的互动体验。

2.提升开发技能和职业竞争力：JavaScript是一种广泛使用的脚本语言，学习它可以为我提供开发Web应用程序的基础知识，提升我在前端开发领域的竞争力。

总之，学习HTML、CSS和JavaScript可以让我掌握构建网页和实现网页交互的基本技能。这些技能不仅能够让我创建出令人印象深刻的网页体验，还能为我在Web开发领域寻求职业发展提供有力支持。

#### 6 对本课程的建议

1.鼓励学生多角度思考和提出问题：引导学生思考问题的不同层面和解决方案的多样性，培养他们的创造力和批判性思维能力。

2.提供开放性的项目和挑战：给学生提供探索自己创新潜力的机会，鼓励他们尝试新领域、挑战自己的舒适区。

3.强调实践和实际问题解决：将理论知识与实际应用结合起来，鼓励学生在实践中学习，并解决现实生活中的问题。

4.鼓励团队合作和协作：设计项目和任务，鼓励学生在小组中互相合作、分享想法、互补长短，促进团队合作的能力。

5.提供专业导师和支持：招募有经验和专业知识的导师，指导学生进行创新实验，提供专业意见和指导。

6.提供必要的资源：确保学生能够获得他们所需的资源和设备，以实现他们的创新想法。

7.不断改进课程内容和教学方法：根据学生反馈和课程效果，不断调整和改进创新实验课程，使其更具挑战性和实用性。

#### 7附录⎯⎯源程序

十进制.html

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>网页计算器</title>

<link rel="stylesheet" type="text/css" href="shijinzhi.css">

</head>

<body>

<div class="ribbon">

<a href="./十进制.html"><span>十进制计算器</span></a>

<a href="./二进制.html"><span>二进制计算器</span></a>

</div>

<div class="calculator">

<input id="display" class="display" type="text" readonly>

<div class="keypad">

<button onclick="appendToDisplay('7')">7</button>

<button onclick="appendToDisplay('8')">8</button>

<button onclick="appendToDisplay('9')">9</button>

<button style="background-color: #f5a623;" onclick="appendOperatorToDisplay('+')">+</button>

<button onclick="appendToDisplay('4')">4</button>

<button onclick="appendToDisplay('5')">5</button>

<button onclick="appendToDisplay('6')">6</button>

<button style="background-color: #f5a623;" onclick="appendOperatorToDisplay('-')">-</button>

<button onclick="appendToDisplay('1')">1</button>

<button onclick="appendToDisplay('2')">2</button>

<button onclick="appendToDisplay('3')">3</button>

<button style="background-color: #f5a623;" onclick="appendOperatorToDisplay('\*')">\*</button>

<button style="background-color: #ff69b4;" onclick="appendOperatorToDisplay('!')">!</button>

<button onclick="appendToDisplay('0')">0</button>

<button style="background-color: #ff69b4;" onclick="appendOperatorToDisplay('.')">.</button>

<button style="background-color: #f5a623;" onclick="appendOperatorToDisplay('/')">/</button>

<button style="background-color: #ff69b4;" onclick="appendParenthesis('(')">(</button>

<button style="background-color: #ff69b4;" onclick="appendParenthesis(')')">)</button>

<button style="background-color: #7b68ee;" onclick="deleteLastChar()">DEL</button>

<button style="background-color: #45a049;" onclick="clearDisplay()">C</button>

</div>

<div class="but1">

<button class="button1" onclick="calculate()" >=</button>

</div>

</div>

<script src="shijinzhi.js"></script>

</body>

</html>

Shijinzhi.css

body {

font-family: Arial, sans-serif;

background-image: url('./beijing/1.png');

background-size: cover;

background-repeat: no-repeat;

}

.calculator {

margin: 20px auto;

width: 300px;

background-color: rgba(255, 255, 255, 0.7);

border: 1px solid #ccc;

border-radius: 5px;

padding: 20px;

}

.display {

background-color: #f3f3f3;

border: 1px solid #ccc;

padding: 10px;

text-align: right;

margin-bottom: 10px;

resize: none;

width: 278.4px;

}

.keypad {

display: grid;

grid-template-columns: repeat(4, 1fr);

grid-gap: 5px;

/\*text-align: center;\*/

}

.keypad button {

padding: 10px;

font-size: 16px;

background-color: #eaeaea/\*#4caf50\*/;

color: #000000;

border: none;

border-radius: 5px;

}

.keypad button:hover {

background-color: #45a049;

cursor: pointer;

}

.keypad button:active {

background-color: #3e8942;

}

.but1{

text-align: center;

padding-top: 10px;

padding-bottom: 7px;

padding-left: 75px;

padding-right: 75px;

display: inline-block;

}

.button1{

width: 150px;

height: 38.4px;

font-size: 16px;

background-color: #45a049;

border: none;

border-radius: 5px;

}

.button1:hover{

cursor: pointer;

}

.ribbon {

display: flex;

justify-content: center;

/\*position: absolute;\*/

right: 300px;

top: 50px;

margin: auto;

}

.ribbon span {

background: #fff;

display: inline-block;

line-height: 3em;

padding: 0 1.5em;

margin-top: 0.5em;

position: relative;

-webkit-transition: background-color 0.2s, margin-top 0.2s;

/\* Saf3.2+, Chrome \*/

-moz-transition: background-color 0.2s, margin-top 0.2s;

/\* FF4+ \*/

-ms-transition: background-color 0.2s, margin-top 0.2s;

/\* IE10 \*/

-o-transition: background-color 0.2s, margin-top 0.2s;

/\* Opera 10.5+ \*/

transition: background-color 0.2s, margin-top 0.2s;

}

.ribbon a:hover span {

background: #FFD204;

margin-top: 0;

}

.ribbon span:before {

content: "";

position: absolute;

top: 3em;

left: 0;

border-right: 0.5em solid #9B8651;

border-bottom: 0.5em solid #fff;

}

.ribbon span:after {

content: "";

position: absolute;

top: 3em;

right: 0;

border-left: 0.5em solid #9B8651;

border-bottom: 0.5em solid #fff;

}

Shijinzhi.js

function appendToDisplay(value) {

var display = document.getElementById("display");

display.value += value;

}

function appendOperatorToDisplay(operator) {

var display = document.getElementById("display");

var lastChar = display.value.slice(-1);

if (lastChar !== '+' && lastChar !== '-' && lastChar !== '\*' && lastChar !== '/' && lastChar !== '!' && lastChar !== '.') {

display.value += operator;

}

}

function appendParenthesis(parenthesis) {

var display = document.getElementById("display");

display.value += parenthesis;

}

function calculate() {

var display = document.getElementById("display");

var expression = display.value;

// 处理阶乘运算

expression = expression.replace(/(\d+)!/g, function(match, num) {

var result = 1;

for (var i = 1; i <= num; i++) {

result \*= i;

}

return result;

});

// 使用eval()函数进行基本计算

try {

var result = eval(expression);

if (isNaN(result) || !isFinite(result)) {

throw "计算结果非法";

}

display.value = result;

} catch (err) {

display.value = "错误";

}

}

function deleteLastChar() {

var display = document.getElementById("display");

display.value = display.value.slice(0, -1);

}

function clearDisplay() {

document.getElementById("display").value = "";

}

二进制.html

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>二进制计算器</title>

<link rel="stylesheet" type="text/css" href="erjinzhi.css">

</head>

<body>

<div class="ribbon">

<a href="./十进制.html"><span>十进制计算器</span></a>

<a href="./二进制.html"><span>二进制计算器</span></a>

</div>

<div class="container">

<h1>二进制计算器</h1>

<input type="text" id="number1" placeholder="请输入第一个二进制数">

<input type="text" id="number2" placeholder="请输入第二个二进制数">

<input type="button" value="加法" onclick="add()">

<input type="button" value="减法" onclick="subtract()">

<input type="button" value="乘法" onclick="multiply()">

<input type="button" value="除法" onclick="divide()">

<input type="button" value="与" onclick="and()">

<input type="button" value="或" onclick="or()">

<h2>结果：</h2>

<div id="result"></div>

</div>

<script src="erjinzhi.js"></script>

</body>

</html>

Erjinzhi.css

\* {

margin: 0;

padding: 0

}

body {

font-family: Arial, sans-serif;

background-color: #f1f1f1;

background-image: url('./beijing/5.png');

background-size: cover;

background-repeat: no-repeat;

}

.container {

width: 400px;

margin: 0 auto;

padding: 20px;

background-color: #fff;

border-radius: 5px;

box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);

text-align: center;

}

h1 {

font-size: 24px;

color: #333;

margin-top: 0;

}

input[type="text"], input[type="button"] {

width: 80%;

padding: 10px;

margin-bottom: 10px;

font-size: 16px;

border-radius: 5px;

border: 1px solid #ccc;

}

input[type="button"] {

background-color: #4caf50;

color: #fff;

cursor: pointer;

}

input[type="button"]:hover {

background-color: #45a049;

}

#result {

font-weight: bold;

padding: 10px;

background-color: #ddd;

border-radius: 5px;

width: 95%;

}

.ribbon {

display: flex;

justify-content: center;

/\*position: absolute;\*/

right: 300px;

top: 50px;

margin: auto;

}

.ribbon span {

background: #fff;

display: inline-block;

line-height: 3em;

padding: 0 1.5em;

margin-top: 0.5em;

position: relative;

-webkit-transition: background-color 0.2s, margin-top 0.2s;

-moz-transition: background-color 0.2s, margin-top 0.2s;

-ms-transition: background-color 0.2s, margin-top 0.2s;

-o-transition: background-color 0.2s, margin-top 0.2s;

transition: background-color 0.2s, margin-top 0.2s;

}

.ribbon a:hover span {

background: #FFD204;

margin-top: 0;

}

.ribbon span:before {

content: "";

position: absolute;

top: 3em;

left: 0;

border-right: 0.5em solid #9B8651;

border-bottom: 0.5em solid #fff;

}

.ribbon span:after {

content: "";

position: absolute;

top: 3em;

right: 0;

border-left: 0.5em solid #9B8651;

border-bottom: 0.5em solid #fff;

}

Erjinzhi.js

function add() {

var number1 = document.getElementById("number1").value;

var number2 = document.getElementById("number2").value;

var decimal1 = parseInt(number1, 2);

var decimal2 = parseInt(number2, 2);

var result = (decimal1 + decimal2).toString(2);

document.getElementById("result").innerHTML = result;

}

function subtract() {

var number1 = document.getElementById("number1").value;

var number2 = document.getElementById("number2").value;

var decimal1 = parseInt(number1, 2);

var decimal2 = parseInt(number2, 2);

var result = (decimal1 - decimal2).toString(2);

document.getElementById("result").innerHTML = result;

}

function multiply() {

var number1 = document.getElementById("number1").value;

var number2 = document.getElementById("number2").value;

var decimal1 = parseInt(number1, 2);

var decimal2 = parseInt(number2, 2);

var result = (decimal1 \* decimal2).toString(2);

document.getElementById("result").innerHTML = result;

}

function divide() {

var number1 = document.getElementById("number1").value;

var number2 = document.getElementById("number2").value;

var decimal1 = parseInt(number1, 2);

var decimal2 = parseInt(number2, 2);

var result = (decimal1 / decimal2).toString(2);

document.getElementById("result").innerHTML = result;

}

function and() {

var number1 = document.getElementById("number1").value;

var number2 = document.getElementById("number2").value;

var decimal1 = parseInt(number1, 2);

var decimal2 = parseInt(number2, 2);

var result = (decimal1 & decimal2).toString(2);

document.getElementById("result").innerHTML = result;

}

function or() {

var number1 = document.getElementById("number1").value;

var number2 = document.getElementById("number2").value;

var decimal1 = parseInt(number1, 2);

var decimal2 = parseInt(number2, 2);

var result = (decimal1 | decimal2).toString(2);

document.getElementById("result").innerHTML = result;

}