比较Firefox和谷歌的记住密码插件的实现区别

主要区别：

存储位置：Firefox使用基于本地浏览器的存储方式来保存密码，而谷歌浏览器使用Google账户来保存密码。

同步功能：Firefox采用本地存储，因此它不提供跨设备的密码同步功能。谷歌浏览器可以通过Google账户自动同步密码，使得用户可以在不同设备上访问已保存的密码。意味着在使用Firefox时，密码仅存储在本地计算机上，而在使用谷歌浏览器时，密码会与Google账户同步到云端。

安全性：Firefox和谷歌浏览器都使用了加密的方式来保存密码，以保证用户数据的安全性。然而，谷歌浏览器的密码同步功能涉及到云端存储和传输，在安全性方面可能存在一定风险。

插件支持：Firefox和谷歌浏览器都支持第三方开发的记住密码插件。这些插件可以增强密码管理功能，并提供额外的安全选项、自定义设置等

一，记住密码插件功能

### 1，存储用户密码

\* 记住密码插件通常在保护用户隐私安全前提下存储用户密码信息，具体而言，插件将在本地存储密码的hash值。

\* 加密存储能够提升安全性，即使插件中密码信息被泄露，也无法还原出原始的密码信息。当户使用密码时，插件计算hash值，与存储的值进行比对。

### 2，自动填充表单

\* 记住密码插件可以自动填充登录表单和其他网页表单中的用户名、密码以及其他个人信息。这样可以方便用户快速登录和填写表单。

### 3，密码同步与备份

\* 一些插件允许用户将保存的密码信息进行同步和备份，确保用户在不同设备上可以方便地访问并使用已保存的密码。

### 4，安全提示与警告

\* 一些插件会提供额外的安全功能，如检测弱密码、重复使用密码等，并给予用户相应的警报和提示，以增强密码的安全性。

二，Firefox记住密码插件

通过监听表单的提交事件，在用户登录时保存密码，并使用浏览器工具栏上的弹出窗口来展示已保存的密码。

### 1，Firefox插件实现简化

> ### 1,创建插件清单文件（manifest.json）：

>

> \* 在插件项目目录中创建一个名为 `manifest.json` 的文件。该清单文件描述了插件的名称、版本、权限等信息。

>

> \* 下面是一个示例的 `manifest.json` 文件内容：

>

> \* ```python

> - {

> -   "manifest\_version": 2,

> -   "name": "My Password Manager",

> -   "version": "1.0",

> -   "description": "A password manager extension for Firefox.",

> -   "permissions": ["storage", "<all\_urls>"],

> -   "browser\_action": {

> -     "default\_popup": "popup.html"

> -   },

> -   "icons": {

> -     "48": "icon.png"

> -   }

> - }

> ```

> ### 2,创建弹出窗口页面

>

> \* 在插件项目目录中创建一个名为 `popup.html` 的文件。该文件定义了插件弹出窗口的外观和交互。

>

> \* 下面是一个示例的 `popup.html` 文件内容：

> <!DOCTYPE html>

<html>

<head>

<title>Password Manager</title>

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

</head>

<body>

<h1>Password Manager</h1>

<form id="login-form">

<label for="username">Username:</label>

<input type="text" name="username" required><br>

<label for="password">Password:</label>

<input type="password" name="password" required><br>

<input type="submit" value="Login">

</form>

</body>

</html>

> ### 3,创建脚本文件

>

> \* 在插件项目目录中创建一个名为 `popup.js` 的文件。该文件包含与插件交互的 JavaScript 代码。

>

> \* 下面是一个示例的 `popup.js` 文件内容：

>

> *// 当popup.html加载完毕时，获取已保存的密码并填充到表单中*

document.addEventListener('DOMContentLoaded', function() {

browser.tabs.query({ active: true, currentWindow: true })

.then(tabs => {

const currentTab = tabs[0];

const origin = new URL(currentTab.url).origin;

browser.passwords.search({ origin: origin })

.then(credentials => {

if (credentials.length > 0) {

const { login, password } = credentials[0];

document.querySelector('input[name="username"]').value = login;

document.querySelector('input[name="password"]').value = password;

}

});

});

});> ```

> ### 4,打包插件

>

> \* 将插件项目目录打包成zip格式，确保清单文件和其他文件位于压缩包的根目录。

> ### 5,在Firefox中加载插件

>

> \* 打开 Firefox 浏览器，访问 `about:debugging` 页面，在左侧导航栏选择 "This Firefox"，然后点击 "Load Temporary Add-on" 按钮，选择刚才打包的压缩包。

三，Google chrome记住密码插件

### 1，chrome密码插件实现简化

> ### 1,创建插件文件夹

>

> \* 在计算机上创建一个新的文件夹，并为插件选择一个唯一的名称。

> ### 2,创建清单文件

>

> \* 在插件文件夹中创建一个名为 `manifest.json` 的文件，并添加以下基本配置信息：

>

> ```python

> - {

> -   "manifest\_version": 2,

> -   "name": "My Password Manager",

> -   "version": "1.0",

> -   "description": "A simple password manager plugin.",

> -   "permissions": ["tabs", "storage"],

> -   "content\_scripts": [

> -     {

> -       "matches": ["<all\_urls>"],

> -       "js": ["content.js"]

> -     }

> -   ]

> - }

> ```

> ### 3,创建内容脚本

>

> \* 在插件文件夹中创建一个名为 `content.js` 的 JavaScript 文件，其中包含与密码管理相关的逻辑。

>

> \* 以下是一个简单的示例来捕获和保存密码：

>

> ```python

> - // 监听登录表单的提交事件

>

> - document.addEventListener('submit', function(event) {

>

> -   var form = event.target;

>

> -   // 获取用户名和密码字段的值

>

> -   var username = form.querySelector('input[name="username"]').value;

>

> -   var password = form.querySelector('input[name="password"]').value;

>

> -   // 使用 Chrome 的存储 API 将用户名和密码保存起来

>

> -   chrome.storage.local.set({ 'username': username, 'password': password }, function() {

>

> -     console.log('保存密码成功！');

>

> -   });

>

> - });

> ```

> ### 4,加载插件

>

> \* 打开 Chrome 浏览器并进入扩展程序管理界面（chrome://extensions）。在顶部启用开发者模式，然后点击 "加载已解压的扩展程序" 按钮，选择创建的插件文件夹。