比较Firefox和谷歌的记住密码插件的实现区别

一．主要区别：

安全性能与扩展

>

> \* Firefox：Firefox 可以使用各种安全插件和扩展来增强其安全性，例如密码生成器、多因素身份验证等。Mozilla 组织还专注于隐私保护，通过阻止跟踪器和提供增强的隐私功能来提高用户的在线安全性。

>

> \* Chrome：Chrome 也支持各种安全扩展和功能，例如密码生成器和广告拦截器。Google 在 Chrome 中实施了一些安全措施，如安全浏览器警示，帮助用户避免访问可能存在恶意软件或欺诈网站。

### 存储位置

\* Firefox使用称为"密码管理器"的内部工具来存储和管理保存的密码。这些密码以加密形式存储在Firefox的配置文件中。而Chrome使用Google账户进行同步，保存的密码将与Google账户关联，并且可以在不同设备间同步。

### 主密码功能

\* Firefox提供了一个主密码功能，可以为保存在浏览器中的所有密码设置额外的层级保护。这样，用户必须输入主密码才能访问保存的密码列表。Chrome没有类似的内置主密码功能。

### ，安全性控制

\* Firefox提供了一些设置选项，例如启用或禁用自动填充密码，启用或禁用密码同步等。而Chrome在安全性控制方面提供了更多的选项，例如可选的账户验证、指纹识别等。

###二Firefox插件实现：

> ### 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。