

个人作业1：Web前端初探

一、作业要求：

- 针对任意网页，**调研其不同方式请求**。至少包括get、post请求，写出或截图其请求及相应数据包。
- 针对任意网页，使用JQuery，能够触发某一事件。写出至少二条语句，截图响应前后不同的状态。
- 完成一个**浏览器插件**，功能不限，文档中写明功能及代码。

二、作业内容

本项目包含三个任务，旨在练习网页请求调试、使用jQuery触发事件、以及创建一个Chrome浏览器插件。

任务 1：研究不同方式请求

1.要求

对任意网页，研究其不同方式请求，包括 GET、POST 请求，写出或截图其请求及相应数据包。

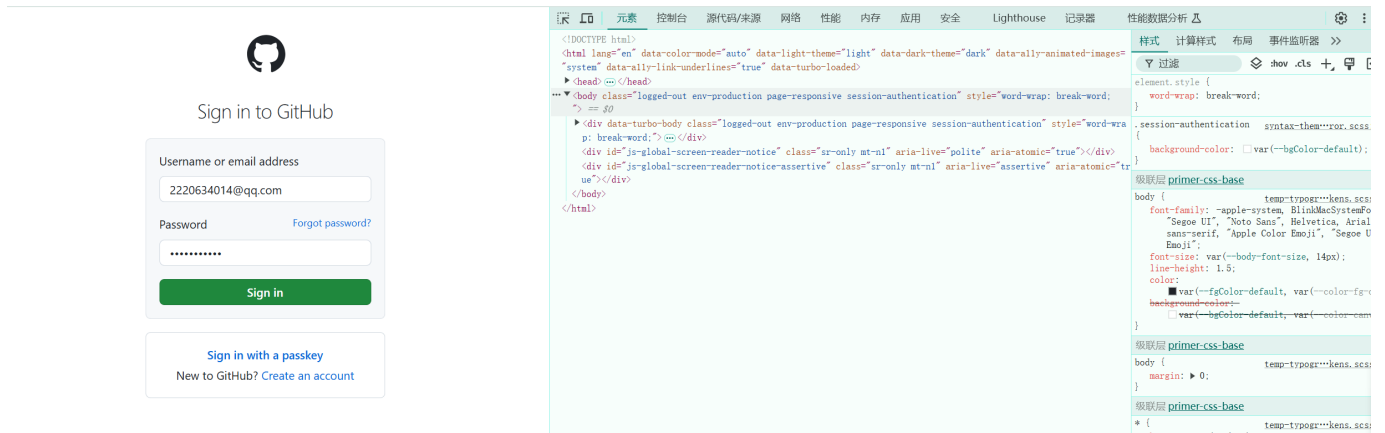
2.基本步骤

1. 打开开发者工具（按 F12 或 Ctrl+Shift+I）。
2. 进入“Network”（网络）面板。
3. 访问网页并筛选出 GET 和 POST 请求。
4. 选择任意请求，查看 Headers 和 Payload 标签，截图或记录请求头、响应头及数据包内容。

3.POST请求

我们选择github进行调研，因为在github中运用到了用户登录的操作，且POST和GET请求可能出现在该操作中。以我自己的账号为例我们进行测试。

首先打开github和对应的开发者工具，如下图所示：



点击登录之后，开始网页的交互，并且我们观察网络中的内容：



我们可以观察到网页捕获到了一个名为session的包，并且其采用的是POST的请求方式：

- **POST请求**是一种HTTP请求方法，用于将数据发送到服务器。它主要用于向服务器提交数据，通常会导致服务器上的资源发生改变（例如添加数据、更新数据库等）。
- **Session（会话）** 是一种用于在用户和服务器之间维护交互状态的机制。它通常用于在短时间内保存用户的状态和数据，以便在多次请求之间识别同一用户。

在本次交互中，我们向服务端发送了账号与密码请求登录，服务端反馈我们一个session用于记录交互的状态和数据。

我们观察状态代码可以发现302 Found的代码，其介绍如下：

- **302 Found** 是一种HTTP状态码，表示请求的资源临时性地被移动到另一个位置。服务器在响应中包含一个新的URL，告诉客户端应该重新请求这个新URL。这个状态码通常用于临时重定向，用户的浏览器会自动跳转到指定的新URL。

然后我们将目光看向Cookie上，我们可以发现如下所示的数据：

X

标头

载荷

预览

响应

启动器

时间

Cookie

请求 Cookie

☐ 显示滤除的请求 Cookie

名称	值	Do...	Path	Expi...	大小	Htt...	Sec...	Sa...	Part...	Cro...	Prio...
_device_id	d5efb6b78b100d0fe343208d8ec2...	gith...	/	202...	42	✓	✓	Lax			Me...
_gh_sess	QeqJUHcWRRoxAR5V8BYFGknm...	gith...	/	会话	578	✓	✓	Lax			Me...
_octo	GH1.1.1849460932.1725189831	.git...	/	202...	32		✓	Lax			Me...
color_mode	%7B%22color_mode%22%3A%22...	.git...	/	会话	214		✓	Lax			Me...
logged_in	no	.git...	/	202...	11	✓	✓	Lax			Me...
preferred_color_mode	light	.git...	/	会话	25		✓	Lax			Me...
tz	Asia%2FShanghai	.git...	/	会话	17		✓	Lax			Me...

响应 Cookie

名称	值	Do...	Path	Expi...	大小	Htt...	Sec...	Sa...	Part...	Cro...	Prio...
_Host-user_session_sa...	St554WghgLSMw6p1Mo2ghDW...	gith...	/	202...	161	✓	✓	Strict			Me...
_gh_sess	w9IAIsgYpD6%2F5hMWxgSYNIL...	gith...	/	会话	1095	✓	✓	Lax			Me...
dotcom_user	wallgreataa	gith...	/	202...	122	✓	✓	Lax			Me...
logged_in	yes	gith...	/	202...	112	✓	✓	Lax			Me...
saved_user_sessions	103624906%3ASt554WghgLSMw...	gith...	/	202...	160	✓	✓	Lax			Me...
tz	Asia%2FShanghai	gith...	/	会话	59	✓	✓	Lax			Me...1
user_session	St554WghgLSMw6p1Mo2ghDW...	gith...	/	202...	141	✓	✓	Lax			Me...

通过调研我们得到：

请求 Cookie

- `_device_id`：这是一个设备标识符 Cookie，可能用于识别用户的设备或跟踪设备使用情况。
- `_gh_sess`：表示一个会话 Cookie，通常用于保持用户的登录状态。此 Cookie 存在于请求和响应中，表明它用于会话管理。
- `_octo`：这是一个带有 GitHub 的标识符 Cookie，可能用于特定用户或设备的唯一标识。
- `color_mode`、`preferred_color_mode`：存储用户的颜色模式偏好，比如深色模式或浅色模式。
- `logged_in`：用于标记用户是否已登录，值为 `no` 表示用户未登录。
- `tz`：存储用户的时区信息，如 "Asia/Shanghai"，便于服务器调整时间显示格式。

响应 Cookie

- `_Host-user_session_same_site` 和 `user_session`：这两个 Cookie 用于管理用户的会话和身份验证。它们通常存储用户的加密会话信息，用于安全验证。
- `dotcom_user`：这个 Cookie 存储用户名（如 `wallgreataa`），用于标识当前登录的用户。
- `logged_in`：与请求 Cookie 中的 `logged_in` 类似，这里显示的值为 `yes`，表明用户在会话中已登录。
- `saved_user_sessions`：通常用来保存用户的会话数据，可能包含多个会话的加密标识符。
- `tz`：同样表示用户的时区信息，和请求 Cookie 中的 `tz` 保持一致。

这些Cookie表示用户具体的信息。

之后我们再观察 **载荷 (Payload)** 信息。

× 标头 载荷 预览 响应 启动器 时间 Cookie

▼ 表单数据 查看源代码 查看网址编码格式的数据

commit: Sign in

authenticity_token: ZYzxUABikWHxuhXhorav6aqD1daqR0K2Iui42Tgnw3KJPwEplrN4A2QH7UdvQmiBjXZcQ+MhalIqkTZRH1hA==

add_account:

login: 2220634014@qq.com

password: [REDACTED]

webauthn-conditional: undefined

javascript-support: true

webauthn-support: supported

webauthn-iuvpaa-support: supported

return_to: https://github.com/login

allow_signup:

client_id:

integration:

required_field_16b5:

timestamp: 1730359191597

timestamp_secret: 254c2ccda9c88f074a88ef27c4b6370873aa40bc74cb0483531e846b58999813

我们可以观察到该载荷中存放着我们的登录信息（账号密码登）、返回的网址（return to）等信息。

3.GET请求

在登录成功之后，网页还通过GET请求获取github.com的网页，并且200 OK的状态码代表了请求的成功。

- 200 OK 代表GET请求成功，并且返回了相应的请求数据。

🔍 元素 控制台 源代码/来源 网络 性能 内存 应用 安全 Lighthouse 记录器 性能数据分析 1 ⚙️

🔍 过滤 反转 更多过滤条件 全部 Fetch/XHR 文档 CSS JS 字体 图片 媒体 清单 WS Wasm 其他

200000 毫秒 400000 毫秒 600000 毫秒 800000 毫秒 1000000 毫秒 1200000 毫秒 1400000 毫秒 1600000 毫秒 1800000 毫秒 2000000 毫秒

名称 session github.com vendors-node_modules... vendors-node_modules... vendors-node_modules... ui_packages_failbot_failb... environment-b7c8196d95... vendors-node_modules... vendors-node_modules... vendors-node_modules... vendors-node_modules... vendors-node_modules... vendors-node_modules...

× 标头 预览 响应 启动器 时间 Cookie

▼ 常规

请求网址: https://github.com/

请求方法: GET

状态码: 200 OK

远程地址: 127.0.0.1:7890

引荐来源网址政策: strict-origin-when-cross-origin

▼ 响应标头

Cache-Control: max-age=0, private, must-revalidate

Content-Encoding: gzip

Content-Security-Policy: default-src 'none'; base-uri 'self'; child-src github.com/assets-cdn/worker/ github.com/webpack/ github.com/assets/ gist.github.com/assets-cdn/worker/; connect-src 'self' uploads.github.com www.githubstatus.com collector.github.com raw.githubusercontent.com api.github.com github-cloud.s3.amazonaws.com github-production-repository-file-5c1aeb.s3.amazonaws.com github-production-upload-

因为请求成功，所以反悔了该网页的详细信息，其存放在**响应（response）** 中。

- **响应主体（Body）**：请求的是 HTML 页面，主体会包含 HTML 代码，如下所示。

× 标头 预览 响应 启动器 时间 Cookie

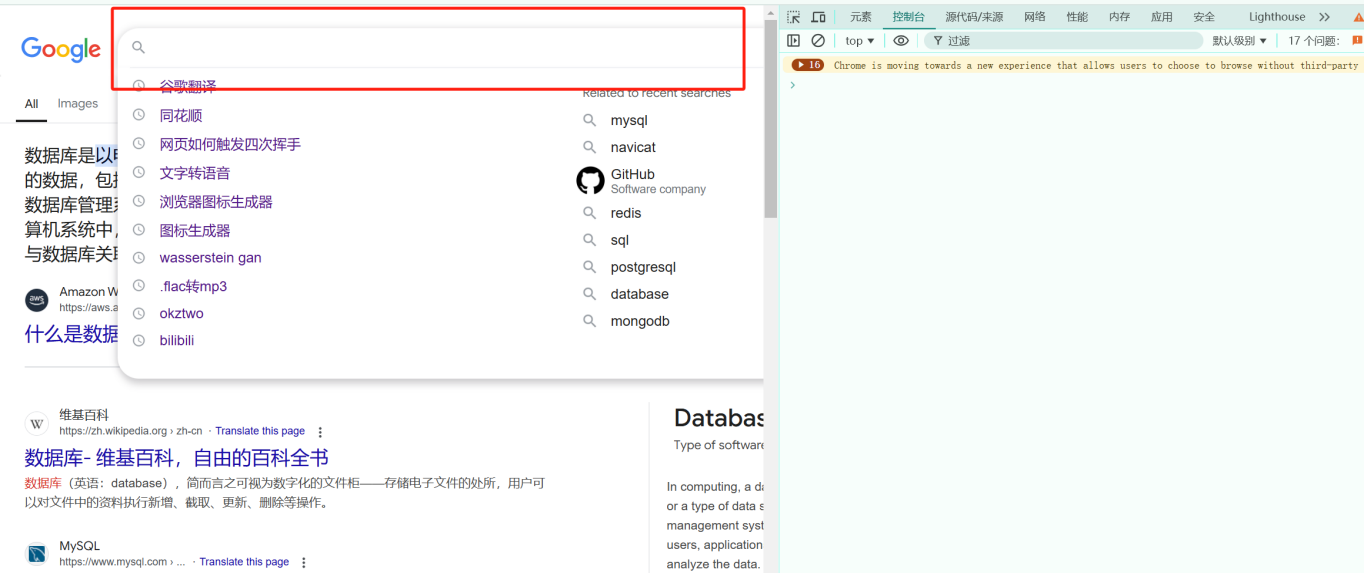
```
1 <!DOCTYPE html>
5 <html lang="en" data-color-mode="auto" data-light-theme="light" data-dark-theme="dark" data-ally-animated-images="system" data-ally-link-underlines="true">
12 <head> ... </head>
212 <body class="logged-in env-production page-responsive full-width" style="word-wrap: break-word;"> ... </body></html>
2107
```

任务 2：使用 jQuery 触发事件

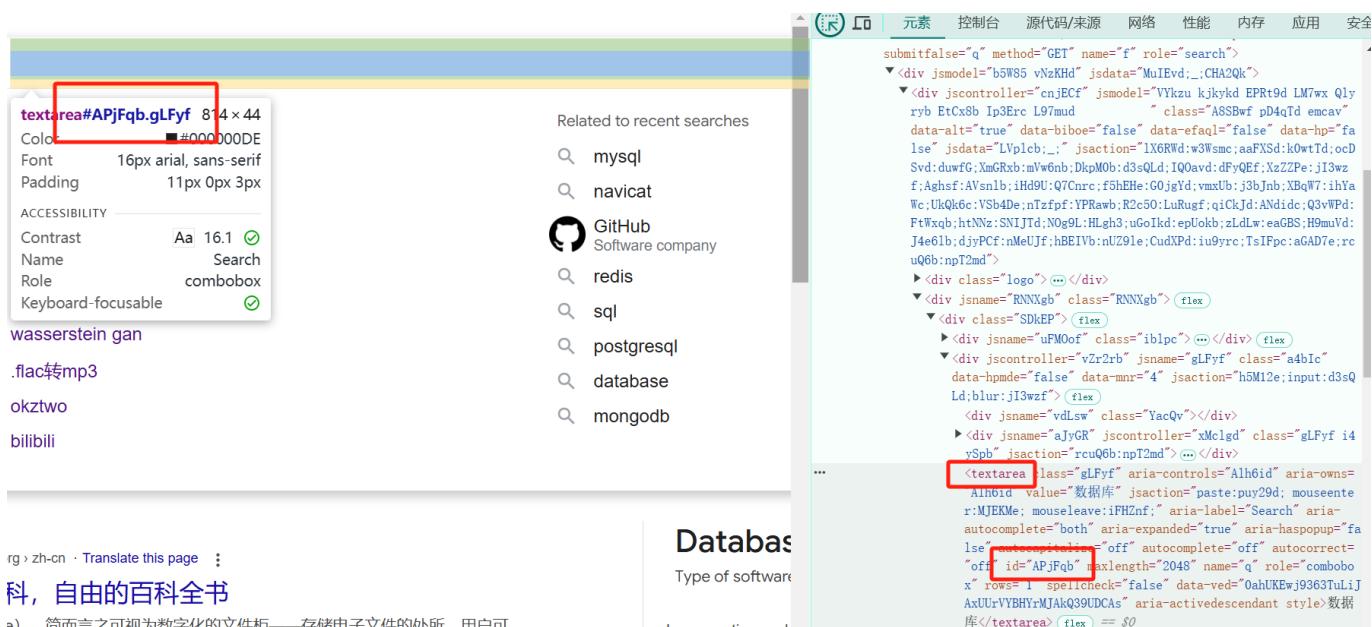
1.要求

针对任意网页，使用 jQuery 能够出发某一事件，并写出至少三条语句，同时截图响应前后的不同状态。

2.更改Google对话框内容

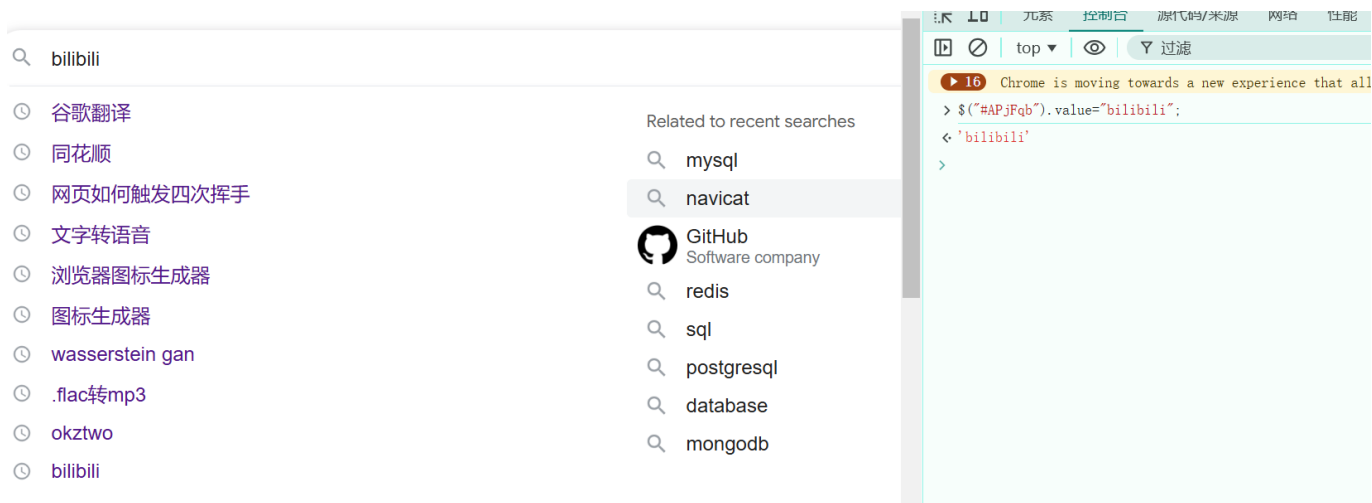


使用jquery更改Google搜索引擎的输入框。通过F12的开发者模式左上角的小箭头头定位页面正下方的对话框，可以获取该item的信息，通过提示发现其**对话框是一个textArea类型**。id为"APjFqb"。为了修改其内容，我们需要编写对应改变其值的JQuery代码：



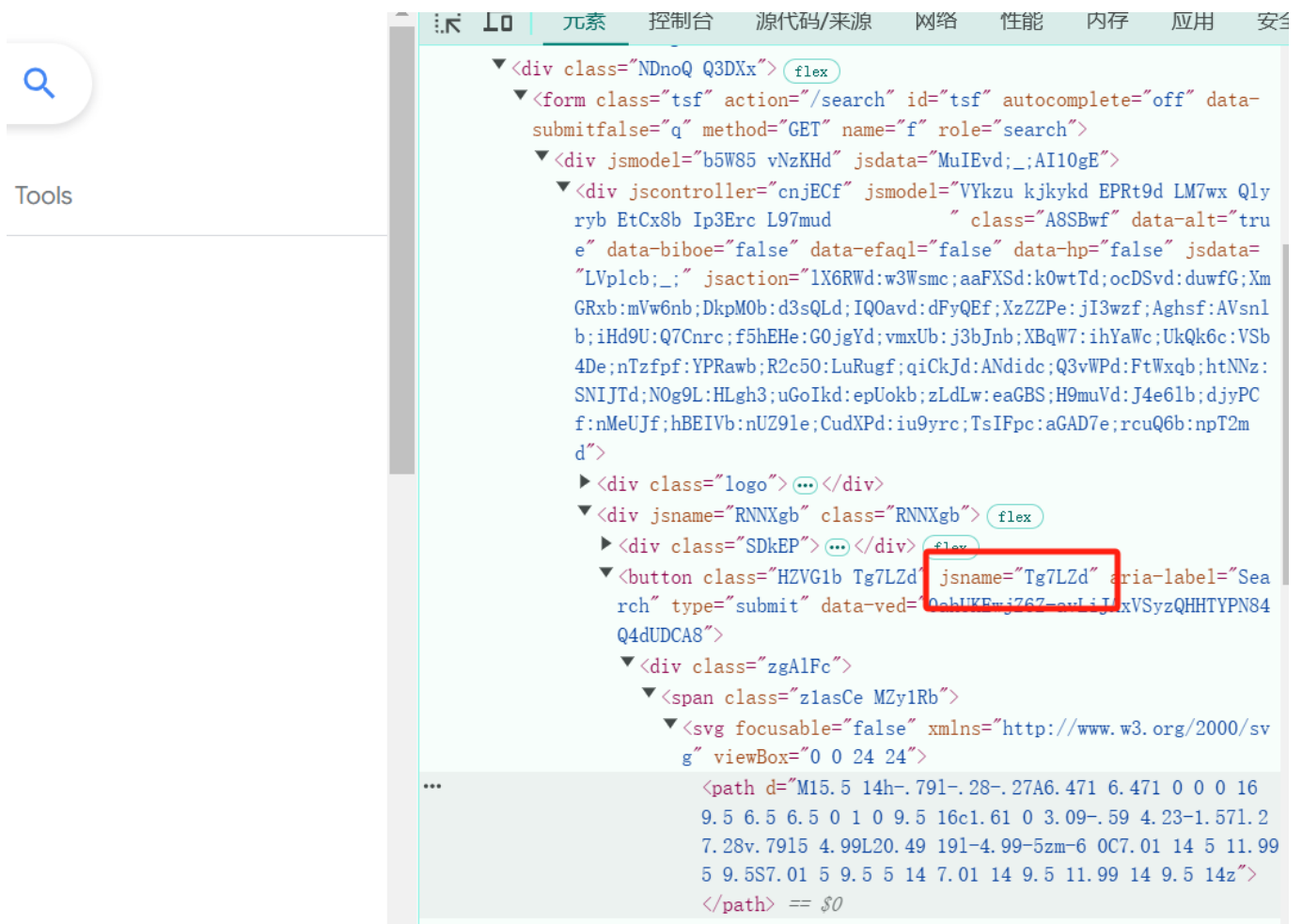
打开控制台，输入如下代码并回车。

```
$("#APjFqb").value="bilibili";
```



结果如上图所示，最开始查看其内容value是为空，然后通过控制台输入上面对应修改搜索内容的代码后，可以看到在左面对话框的内容被改变了。

然后我们再输入一个命令进行提交搜索



同样的，通过F12的开发者模式左上角的小箭头定位页面的搜索按钮，获取该item的信息，得到button的 `jsname="Tg7LZd"`。为了修改其内容，我们需要编写对应改变其值的jQuery代码：

```
$(".Tg7LZd").click();
```

执行以上命令，我们的网页就发生了跳转。

任务 3：创建一个 Chrome 插件

要求

完成一个 Chrome 浏览器插件，功能不限，并在文档中写明功能及代码。

插件功能

- 插件的弹出页面（popup）会在用户点击 Chrome 浏览器工具栏中的插件图标时打开；
- 页面上方显示用户的基本信息，包括：专业、学号、姓名；

- 页面设计中包含一个用于展示个人 LOGO 的图片区域。LOGO 以图片形式展示，通过 `` 标签加载；
- 页面提供一个音频控件，用于播放自我介绍音频文件。音频文件路径为 `audio/myself.mp3`，在 `popup.html` 中通过 `<audio>` 标签引用。
- 页面底部有一个带有 `id="myButton"` 的按钮，称为“魔法链接按钮”，用于引导用户访问特定页面（GitHub 主页）。

实现步骤

1. **创建项目文件夹**，在本项目中我设置为 `chrome_extension`。
2. **创建 `manifest.json`**，配置插件基本信息：

```
{
  "manifest_version": 3,
  "name": "mu_mian's Chrome Extension",
  "version": "1.0",
  "description": "Halooooooo!.",
  "author": "2220634014@qq.com",
  "permissions": [
    "scripting",
    "activeTab",
    "tabs",
    "notifications",
    "storage"
  ],
  "background": {
    "service_worker": "background.js"
  },
  "action": {
    "default_popup": "popup.html",
    "default_icon": {
      "16": "imgs/icon16.png",
      "32": "imgs/icon32.png",
      "48": "imgs/icon48.png",
      "64": "imgs/icon64.png",
      "128": "imgs/icon128.png",
      "256": "imgs/icon256.png"
    }
  }
}
```

3. **background.js**：实现插件安装时候，**检测当前页面**，向控制台打印 `Extension installed`。


```
chrome.runtime.onInstalled.addListener(() => {  
  console.log('Extension installed');  
});
```

4. **创建 popup.html 和 popup.css 文件**，设计插件的界面。从页面设计的角度来看，popup.html 和 popup.css 文件构成了此 Chrome 扩展的用户界面部分。以下是它们的设计细节和功能说明。

- **创建项目文件夹**，在本项目中我设置为 `chrome_extension`。
- **创建 manifest.json**，配置插件基本信息：

```
{  
  "manifest_version": 3,  
  "name": "mu_mian's Chrome Extension",  
  "version": "1.0",  
  "description": "Halooooooo!.",  
  "author": "2220634014@qq.com",  
  "permissions": ["scripting",  
    "activeTab",  
    "tabs",  
    "notifications",  
    "storage"],  
  "background": {  
    "service_worker": "background.js"  
  },  
  "action": {  
    "default_popup": "popup.html",  
    "default_icon": {  
      "16": "imgs/icon16.png",  
      "32": "imgs/icon32.png",  
      "48": "imgs/icon48.png",  
      "64": "imgs/icon64.png",  
      "128": "imgs/icon128.png",  
      "256": "imgs/icon256.png"  
    }  
  }  
}
```

- **background.js**：实现插件安装时候，**检测当前页面**，向控制台打印 `Extension installed`。

```
chrome.runtime.onInstalled.addListener(() => {  
  console.log('Extension installed');  
});
```

- **创建 popup.html 和 popup.css 文件**，设计插件的界面。从页面设计的角度来看，popup.html 和 popup.css 文件构成了此 Chrome 扩展的用户界面部分。以下是它们的设计细节和功能说明。

```
<!-- popup.html -->
<!DOCTYPE html>
<html lang="zh-CN">
<head>
  <meta charset="UTF-8">
  <title>个人主页</title>
  <!-- 引入外部CSS文件 -->
  <link rel="stylesheet" href="popup.css">
</head>
<body>
  <div class="info-box">
    <p>-----</p>
    <h1>个人主页</h1>
    <p>专业：密码科学与技术</p>
    <p>学号：2213912</p>
    <p>姓名：罗劲</p>
    <p>-----</p>
    <h2>LOGO</h2>
    
    <p>-----</p>
    <h2>自我介绍</h2>
    <audio controls>
      <source src="audio/myself.mp3" type="audio/mpeg">
      您的浏览器不支持音频播放。
    </audio>
    <p>-----</p>
    <h2>魔法链接按钮</h2>
    <button id="myButton">Click me</button>
    <script src="popup.js"></script>
    <p>-----</p>
  </div>
</body>
</html>
```

```
/* popup.css */
/* 设置页面背景 */
body {
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    margin: 0;
    background-image: url('imgs/background.png');
    background-size: cover;
    background-position: center;
    font-family: Arial, sans-serif;
}

/* 方框样式 */
.info-box {
    padding: 70px;
    width: 700px;
    border-radius: 10px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
    text-align: center;
    color: #333;
    background-color: rgba(255, 255, 255, 0.5); /* 添加半透明背景颜色提高可读性 */
}

/* 标题和文本样式 */
h1 {
    font-size: 24px;
    margin: 10px 0;
}

p {
    margin: 5px 0;
    font-size: 16px;
}

/* 音频控制样式 */
audio {
    margin-top: 12px;
    width: 80%;
}

/* LOGO图片样式（减小大小） */
.info-box img {
    margin-top: 5px;
    width: 20%;
    height: auto;
}
```

```
#popup.js
document.getElementById('myButton').addEventListener('click', () => {
  alert('欢迎来到我的github!');
  chrome.tabs.create({ url: 'https://github.com/wallgreateaa' });
});
```

- `popup.html` 定义了页面结构，展示个人信息、LOGO、音频文件和导航按钮。
- `popup.css` 提供了弹出页面的样式，使页面内容居中，并使用半透明背景和阴影效果提升用户体验。
- 页面通过 `popup.js` 实现了按钮点击后的导航功能，完成了 Chrome 扩展的基本交互功能。

这个设计通过 HTML、CSS 和 JavaScript 的结合，实现了简洁、功能明确的 Chrome 扩展弹出页面。

插件展示

该插件在浏览器工具栏上显示一个按钮，点击后弹出页面，提供跳转到指定网站的功能。

☆

🎓

👤

🖼️

|

👤

⋮

📁 所有书签

🖼️

📁

👤

⋮


个人主页

专业：密码科学与技术

学号：2213912

姓名：罗劲

LOGO



人生啊

自我介绍

☆

🎓

👤

🖼️

|

👤

⋮


📁 所有书签

🖼️

📁

👤

⋮



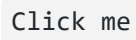


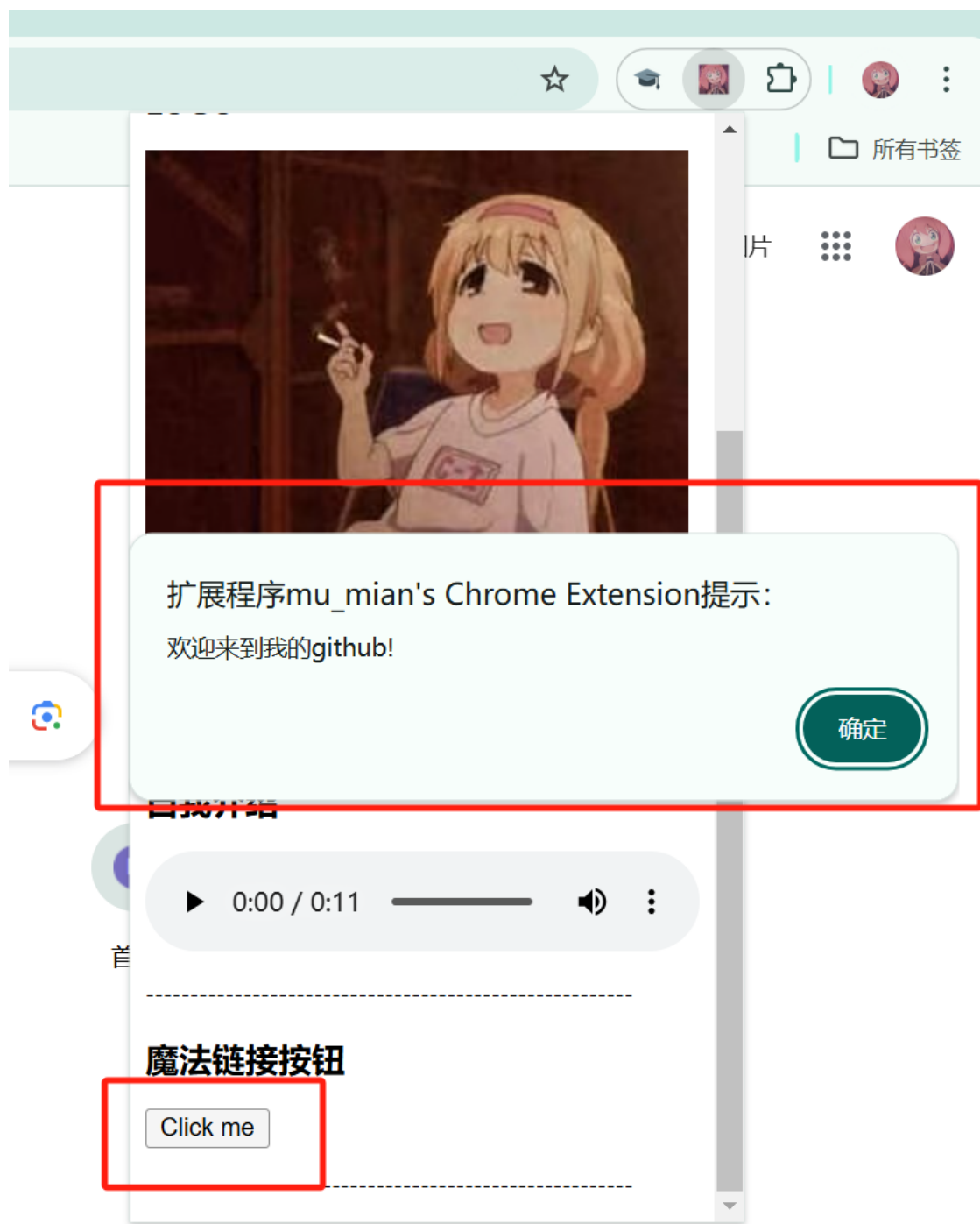
自我介绍

▶ 0:00 / 0:11 ———— 🔊 ⋮

魔法链接按钮

Click me

- 个人信息展示：包括专业、学号、姓名
- 个人LOGO展示；
- 个人的一段语音自我介绍
- 一个点击按钮即可抵达我的github主页的按钮，如下图所示，当点击**魔法按钮**：时，就会跳转到我的github主页！



跳转到我自己的github页面!



以上是项目的详细说明和代码示例。根据这些内容可以进一步调整和扩展插件的功能。