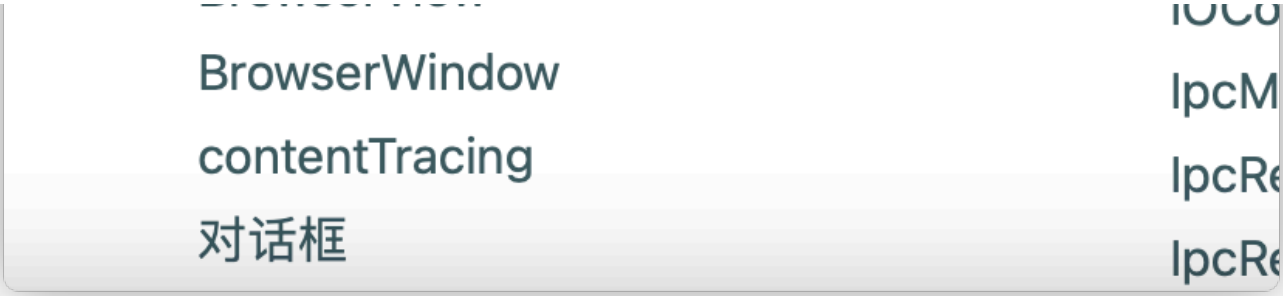
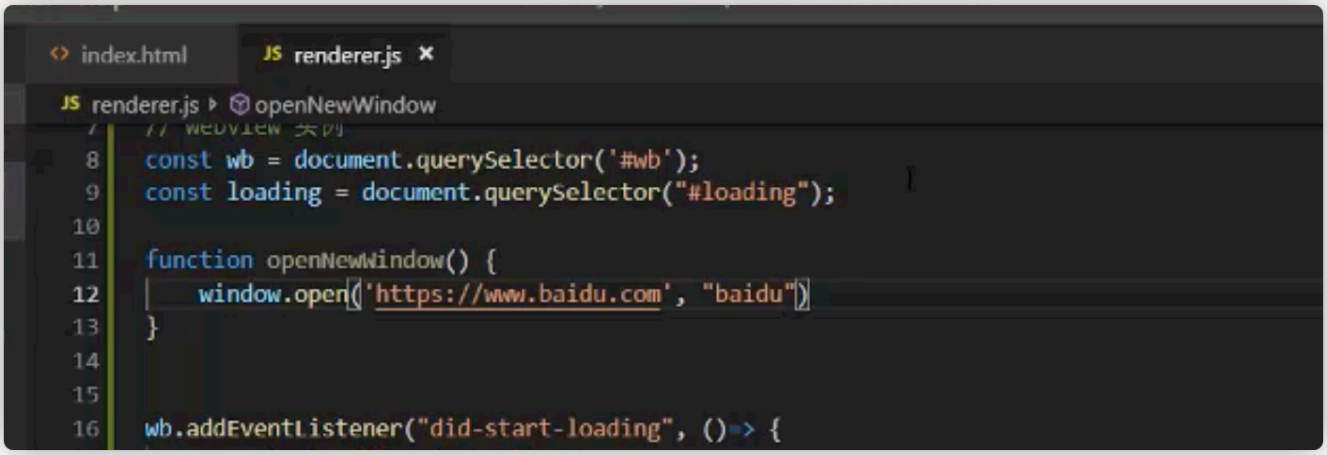
# windowOpen字符窗口的通信

https://www.electronjs.org/docs/api/window-open







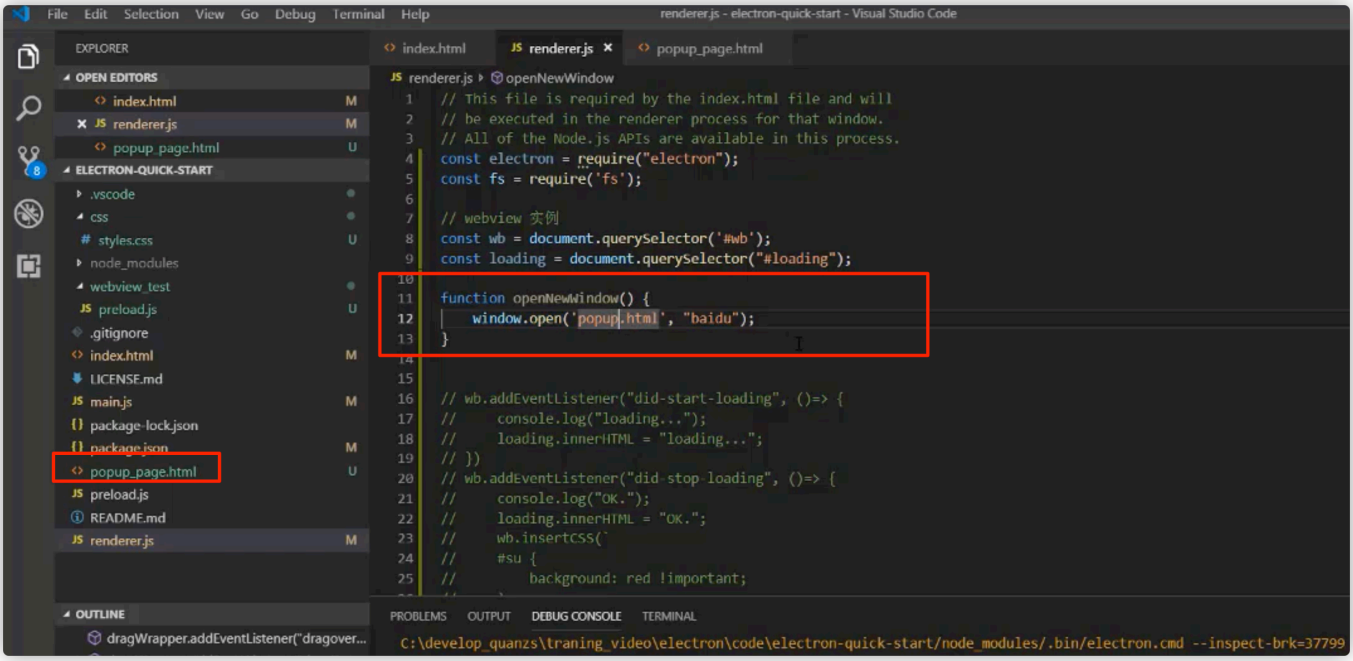

子窗口向父窗口传递消息

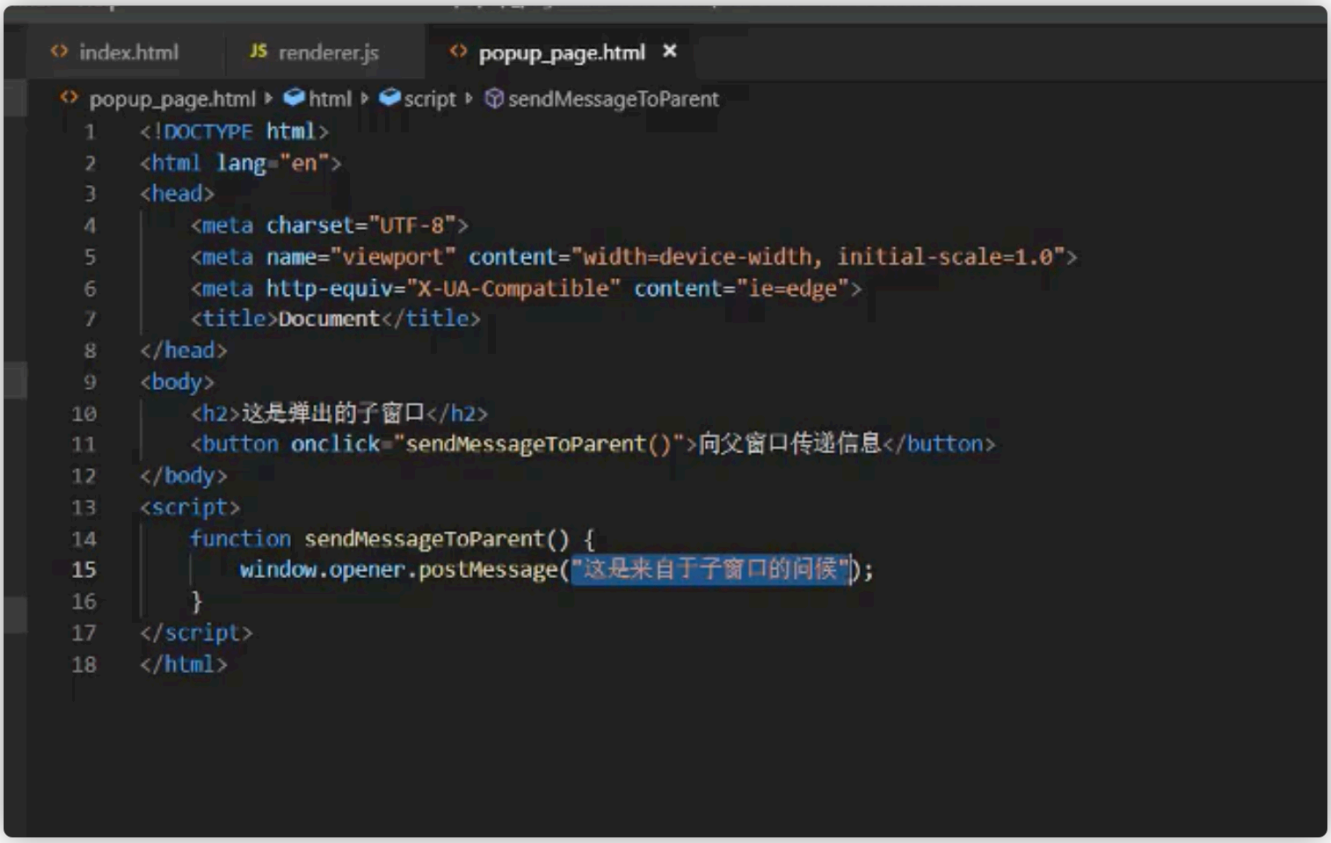
```
window.opener.postMessage(message, targetOrigin)
```

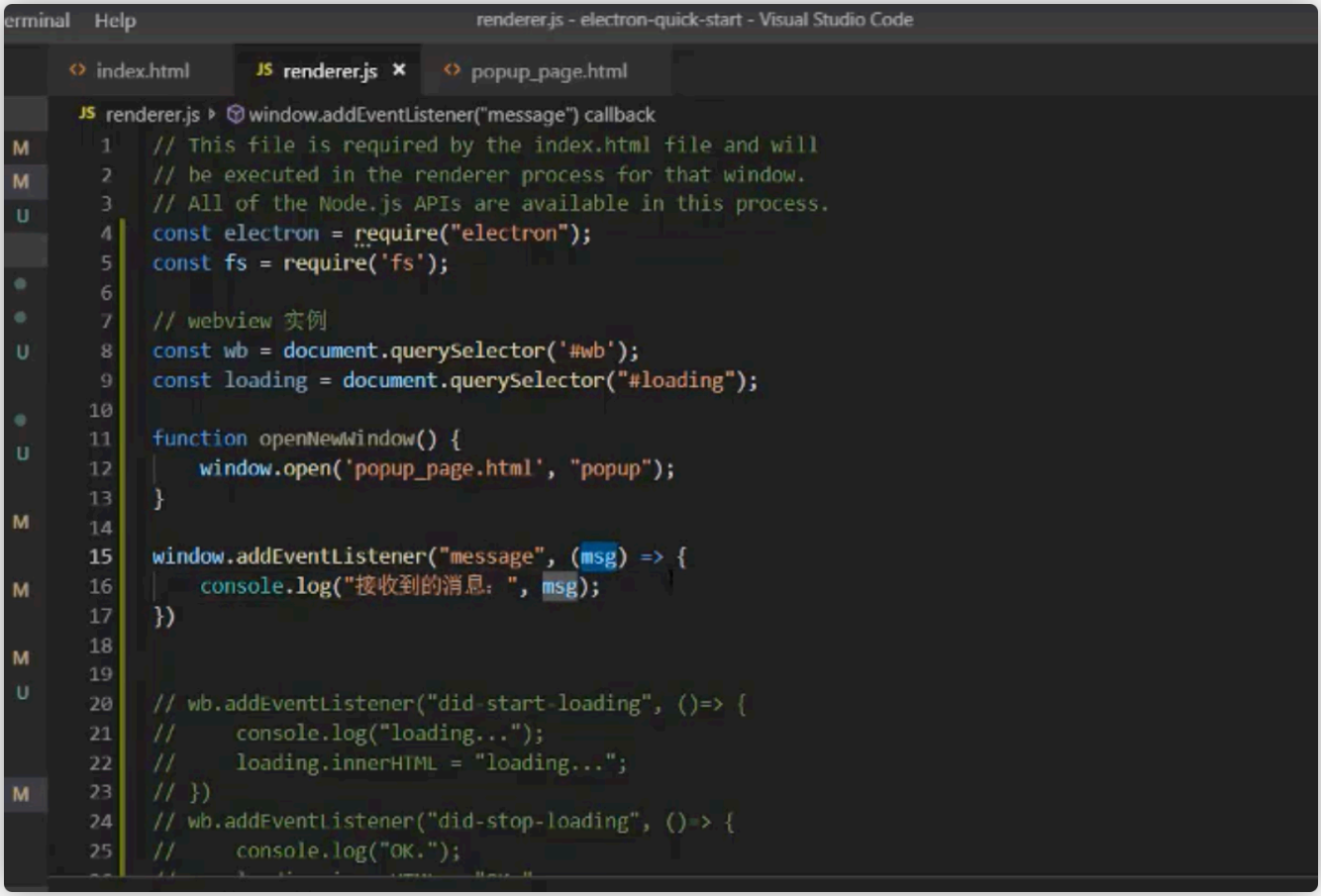
- `message` String
- `targetOrigin` String

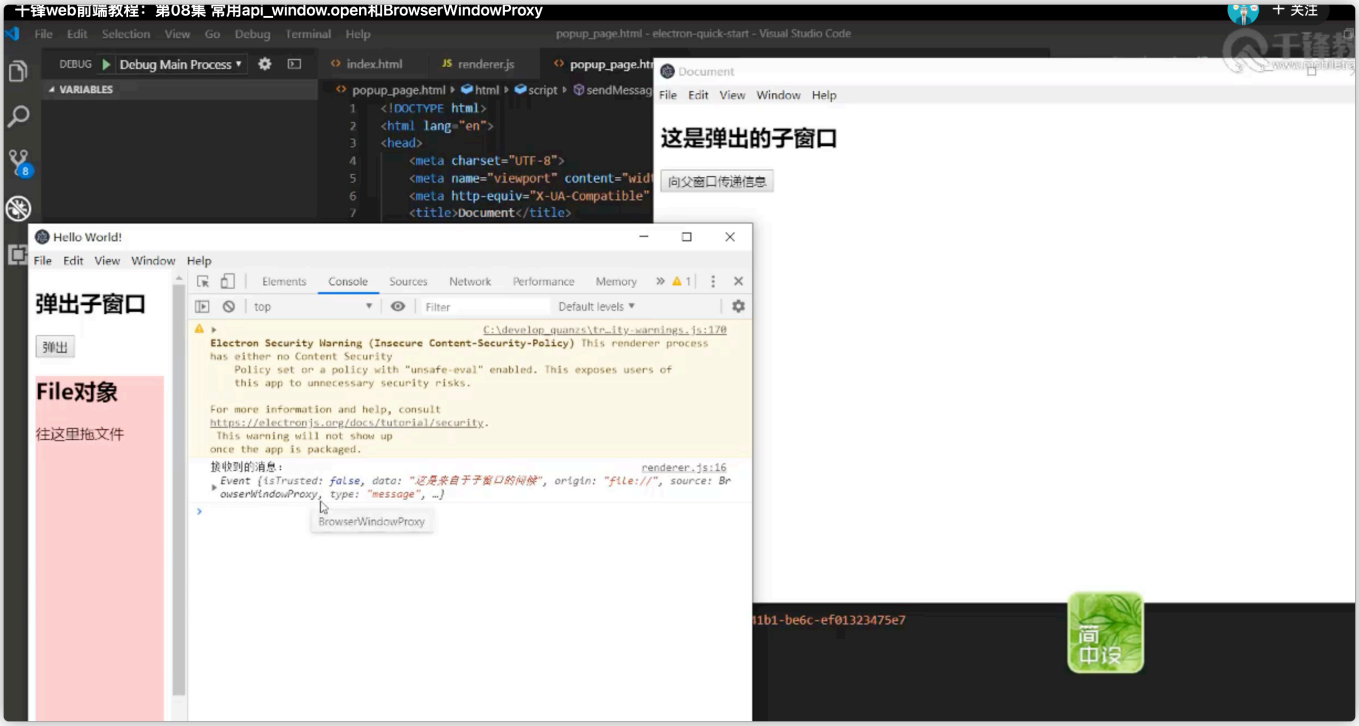
将消息发送给指定来源的父窗口，如果未指定来源则发送给 `*` ，即所有窗口。

新建一个子窗口




renderer.js - electron-quick-start - Visual Studio Code
index.html
renderer.js
popup_page.html
renderer.js ▸ window.addEventListener("message") callback
// This file is required by the index.html file and will
// be executed in the renderer process for that window.
// All of the Node.js APIs are available in this process.
const electron = require("electron");
const fs = require('fs');
// webview 实例
const wb = document.querySelector('#wb');
const loading = document.querySelector("#loading");
function openNewWindow() {
window.open('popup_page.html', "popup");
}
window.addEventListener("message", (msg) => {
console.log("接收到的消息: ", msg);
})
// wb.addEventListener("did-start-loading", ()=> {
// console.log("loading...");
// loading.innerHTML = "loading...";
// })
// wb.addEventListener("did-stop-loading", ()=> {
// console.log("OK.");


十锋web前端教程：第08集 常用api_window.open和BrowserWindowProxy
这是弹出的子窗口
向父窗口传递信息
Hello World!
弹出子窗口
弹出
File对象
往这里拖文件
接收到的消息：
renderer.js:16
BrowserWindowProxy

也可以穿json

index.html　renderer.js　popup_page.html ×

popup_page.html ▸ html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
</head>
<body>
    <h2>这是弹出的子窗口</h2>
    <button onclick="sendMessageToParent()">向父窗口传递信息</button>
</body>
<script>
    function sendMessageToParent() {
        window.opener.postMessage({
            type: 1,
            message: "这是来自于子窗口的问候"
        });
    }
</script>
</html>
```

# 关闭子窗口

index.html ×　renderer.js　popup_page.html

index.html ▸ html ▸ body ▸ div ▸ button

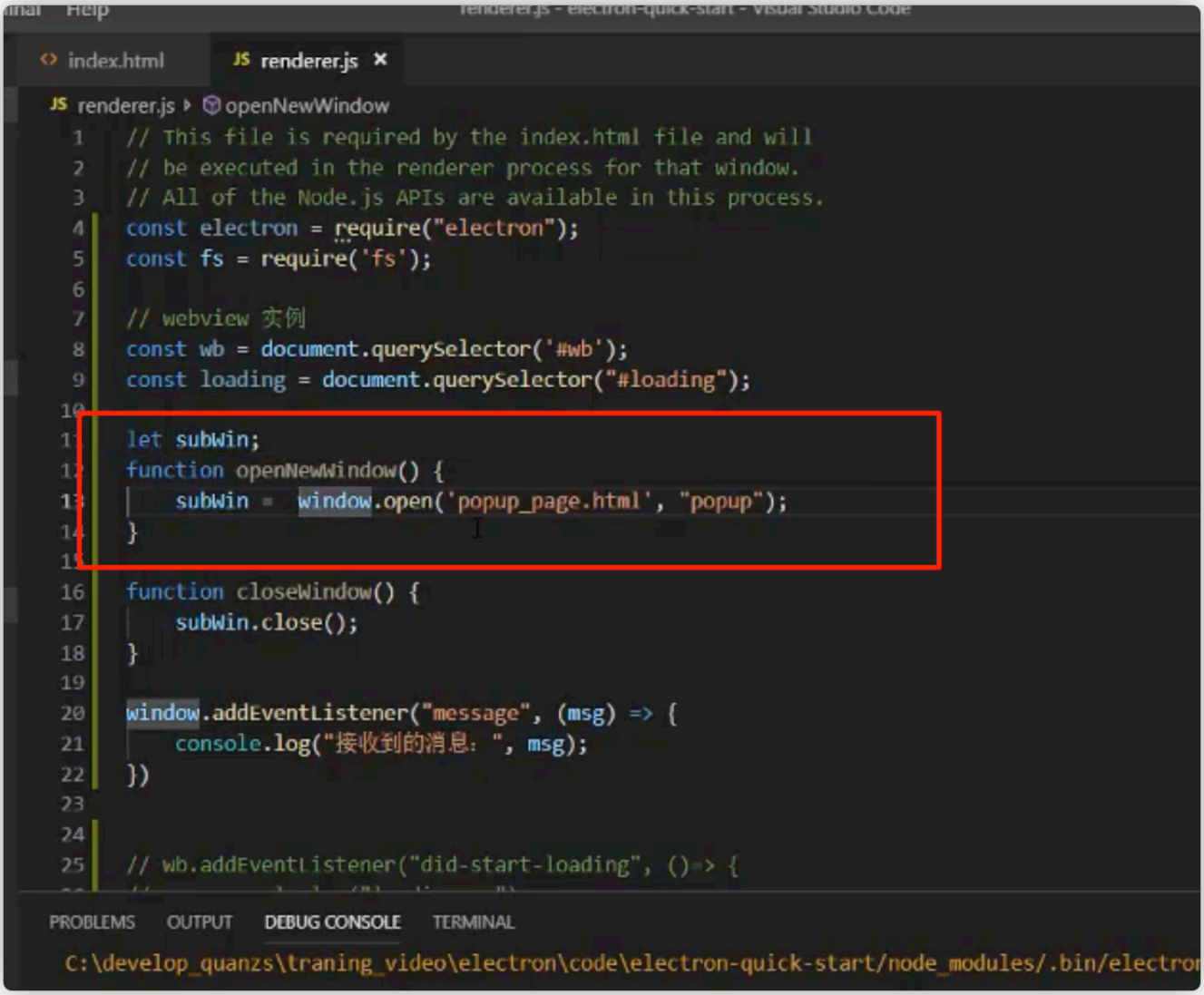
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Hello World!</title>
    <link rel="stylesheet" href="css/styles.css" />
  </head>
  <body>
    <div>
      <h2>弹出子窗口</h2>
      <button onclick="openNewWindow()">弹出</button>
      <button onclick="closeWindow()">关闭子窗口</button>
    </div>

    <!-- <div>
      <h2>Webview</h2>
      <span id="loading"></span>
      <webview id="wb" src="https://www.baidu.com/" style="height: 400px;" preload="./webview_test/pr
    </div> -->

    <div class="for_file_drag" id="drag_test">
      <h2>File对象</h2>
      <span>往这里拖文件</span>
    </div>
```

# BrowserWindowProxy

https://www.electronjs.org/docs/api/browser-window-proxy


index.html
renderer.js
renderer.js ▸ openNewWindow
// This file is required by the index.html file and will
// be executed in the renderer process for that window.
// All of the Node.js APIs are available in this process.
const electron = require("electron");
const fs = require('fs');
// webview 实例
const wb = document.querySelector('#wb');
const loading = document.querySelector("#loading");
let subWin;
function openNewWindow() {
subWin = window.open('popup_page.html', "popup");
}
function closeWindow() {
subWin.close();
}
window.addEventListener("message", (msg) => {
console.log("接收到的消息: ", msg);
})
// wb.addEventListener("did-start-loading", ()=> {
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
C:\develop_quanzs\traning_video\electron\code\electron-quick-start/node_modules/.bin/electron