hw1：前端实验初探

1. 调研不同方式请求

我们来调研南开大学计算机学院网站的主页。使用Google Chorme

我们打开F12的网络模式，刷新主页页面，发现跳出60多个请求，我们从中选取几条展示

1.1 Get

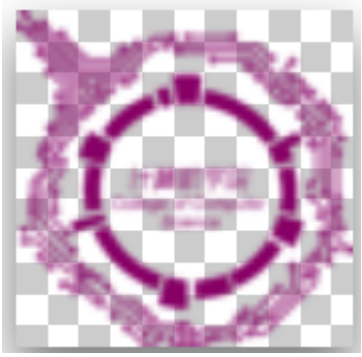
- 如下，这是logo图标的请求信息

▼ 常规

请求网址	https://cc.nankai.edu.cn/_upload/tpl/01/bf/447/template447/logo.ico
请求方法	GET
状态代码	200 OK （来自磁盘缓存）
远程地址	127.0.0.1:7897
引荐来源网址政策	strict-origin-when-cross-origin

▼ 响应标头

Accept-Ranges	bytes
Content-Length	16958
Content-Type	image/x-icon
Date	Fri, 31 Oct 2025 12:01:31 GMT
Etag	"423e-58c73f7f9b53e"
Frame-Options	SAMEORIGIN
Last-Modified	Sat, 29 Jun 2019 10:36:33 GMT
Vary	Accept-Encoding
X-Content-Type-Options	nosniff
X-Frame-Options	SAMEORIGIN
X-Protected-By	NKSOC
X-Xss-Protection	1;mode=block



我们关注以下重点：

1. 请求方法： GET

2. 请求 URL： `https://cc.nankai.edu.cn/_upload/tpl/01/bf/447/temlate447/logo.ico`

3. 请求目的：

用于加载网页的图标文件，属于静态资源请求。

4. 请求状态码：

`200 OK`，表示成功响应

5. 响应头（关键字段）：

- `Content-Type: image/x-icon` —— 返回的是图标文件
- `Content-Length: 16958` —— 图标文件大小约 16 KB
- `Last-Modified: Sat, 29 Jun 2019 10:36:33 GMT` —— 文件的最后修改时间，用于缓存

1.2 Post

- 如下，这是身份登录时候的信息

请求网址	https://iam.nankai.edu.cn/api/quick-login?os=web
请求方法	POST
状态代码	200 OK
远程地址	127.0.0.1:7897
引荐来源网址政策	strict-origin-when-cross-origin

▼响应标头

Content-Encoding	gzip
Content-Type	application/json; charset=utf-8
Date	Thu, 06 Nov 2025 11:11:14 GMT
Server	nginx
Set-Cookie	csrf-token=yzeDXFvjAcYaBaLgikIkdwqNYtSFFhukmJWLszeu; Path=/; Expires=Thu, 06 Nov 2025 11:41:14 GMT; Max-Age=1800
Set-Cookie	session=MTc2MjQyNzQ3NHxOd3dBTkZoWIZUSXISbFpKUWtOT05sZEJORVZKV2paRlVUSkpWalpFU1VjMlVGbEpXVWxEVVZoSVVGbzFUVVUzUWxsWU5FWkNOVUU9fPWzYIrc4BOvIJKfBSw6V2LZTU1GqKadeELm6oMv6ASR; Path=/; Expires=Thu, 06 Nov 2025 11:41:14 GMT; Max-Age=1800; HttpOnly; Secure

▼**请求载荷** 查看源代码

▼ {token: "gUQcxuBghBftmbvqVmGfCQKkNQYrSFcmyKGHzixx", login_scene: "feilian"}
login_scene: "feilian"
token: "gUQcxuBghBftmbvqVmGfCQKkNQYrSFcmyKGHzixx"

对于Post提交操作，真正重要的是Payload载荷：

```
{
  "token": "gUQc...zixx",
  "login_scene": "feilian"
}
```

这是一个 **JSON 格式** 的请求体，服务器接收的数据只有两项：

- `token`：用于身份认证

- `login_scene`：用于服务器判断用户来源场景

2. Jquery

- Jquery是用来简化JavaScript查询语句的

2.1 检查网页是否启用JQuery

我们先在Console中检查网页是否有JQuery：

```
> window.jQuery
< ƒ (a,b){return new n.fn.init(a,b)}
```

可以发现网页已启用

2.2 三条语句

- 我们打算在网页中增加一个按钮，绑定点击操作，最后再用控制台模拟点击

创建按钮

我们在主页append一个button标签。

这里Z-index设置很大，是为了显示在最上层

```
$('body').append(`
  <button id="testBtn" style="
    position:fixed;bottom:30px;right:30px;
    padding:10px 20px;background:#ccc;
    border:none;border-radius:8px;cursor:pointer;
    font-size:16px;z-index:999999">
    点我（点击前）
  </button>
`);
```

- 结果如下图，显示了一个灰色按钮

绑定点击事件

- 我们设置一个点击事件，修改按钮的文字与css样式，区分是否发生点击事件

```
$('#testBtn').on('click', function () {
    $(this).text("已点击！");
    $(this).css('background', '#90ee90');
    $(this).css('color', '#000');
});
```

此时因为还没有发生点击，因此没有图的变化。如果鼠标点击，是会和控制台模拟鼠标点击的效果相同。

控制台模拟点击

```
$('#testBtn').trigger('click');
```

运行代码后如图所示，样式发生改变，说明完成一次点击操作。

3. 浏览器插件

- 我设计了一个小雅答题提醒的插件，方便在小雅有题目的时候及时通知，防止错过答题

```
// ==UserScript==
// @name         Auto Refresh & Detect Rows with GM_notification
// @namespace    https://example.com
// @version      1.1
// @description  每 15 秒刷新页面，若找到 1 个题目元素，提醒并发送桌面通知。
// @author       我
// @match        https://nankai.ai-augmented.com/*
// @grant        GM_notification
// @run-at       document-idle
// ==/UserScript==

(function () {
    "use strict";

    let intervalId = setInterval(() => {
        let rows = document.querySelectorAll(
            ".ant-table-row.ant-table-row-level-0"
        );

        if (rows.length === 1) {
            // 清空网页内容
            document.documentElement.innerHTML = "";

            // 创建提示 div
            let messageDiv = document.createElement("div");
```

```
            messageDiv.textContent = "有作业赶紧做";
            messageDiv.style.position = "absolute";
            messageDiv.style.top = "50%";
            messageDiv.style.left = "50%";
            messageDiv.style.transform = "translate(-50%, -50%)";
            messageDiv.style.fontSize = "50px";
            messageDiv.style.color = "red";
            messageDiv.style.fontWeight = "bold";
            messageDiv.style.zIndex = "9999";
            document.body.appendChild(messageDiv);

            // 发送桌面通知
            GM_notification({
                title: "检测到作业",
                text: "有作业赶紧做！",
                image: "https://example.com/icon.png", // 可自定义图标显示
                timeout: 50000, // 通知显示时间（毫秒）
                onclick: () => {
                    window.focus(); // 点击通知时聚焦窗口
                },
            });

            clearInterval(intervalId); // 停止定时器
        } else {
            location.reload(); // 刷新网页
        }
    }, 15000);
})();
```

3.1 元数据块

元数据块是以下部分： // ==UserScript== ... ==/UserScript==

- **@name / @description / @version / @author**

  纯说明信息；在管理器里展示用。

- **@match**

  只在 `https://nankai.ai-augmented.com/*` 这个域名/路径下生效。

- **@grant GM_notification**

  脚本声明会使用 Tampermonkey 的 `GM_notification` API，发送系统桌面通知。

- **@run-at document-idle**

  脚本注入时机为页面空闲阶段（文档和子资源基本加载完后）；比 `document-start` 晚，比 `document-end` 再晚一些。

3.2 主体 IIFE 与严格模式

```
(function () {
    "use strict";
    ...
})();
```

- IIFE是**立即执行函数表达式**，也就是说定义完函数后立即执行。

  IIFE可以创建独立的作用域，外部页面脚本访问不到，不会冲突。因为Tampermonkey 脚本运行在页面里，页面本身也有大量 JS，如果变量名冲突，可能造成奇怪错误

- use strict是 JavaScript 的**严格模式**，避免 JS 的各种隐性错误。

  严格模式能避免很多 bug，比如：

  - 意外创建全局变量
  - 删除不可删除属性
  - 重名参数

3.3 定时器

```
let intervalId = setInterval(() => {
    let rows = document.querySelectorAll(
        ".ant-table-row.ant-table-row-level-0"
    );

    if (rows.length === 1) {
        ...
        clearInterval(intervalId); // 停止定时器
    } else {
        location.reload(); // 刷新网页
    }
}, 15000);
```

这段代码是在：每隔 15 秒检测一次表格行数，如果不是 1，就刷新页面；如果是 1，就执行提醒并停止轮询。

以下为每次执行Fn的过程：

1. 在整个页面中查找所有 class 为 `.ant-table-row.ant-table-row-level-0` 的元素，即查找表格行
2. 判断行数是否 === 1，只有当行数“恰好等于 1”时才触发提醒，这时说明恰好发布了新题目。
3. 当行数 === 1 时，代码将会：清空网页；在页面中间显示提醒文字；发桌面通知（GM_notification）；停止定时器。
4. 行数不是 1 → 刷新网页，这是因为平台不会在有题目的时候自动推送题目，需要手动刷新。

3.4 rows.length === 1时动作

首先，我们清空网页内容，然后创建提示文字

```
document.documentElement.innerHTML = "";
let messageDiv = document.createElement("div");
messageDiv.textContent = "有作业赶紧做";
...
```

代码后面就是给提示 DIV 设置样式，使其居中显示；设置字体样式，强调提示

3.5 桌面通知模块

```
// 发送桌面通知
GM_notification({
    title: "检测到作业",
    text: "有作业赶紧做！",
    image: "https://example.com/icon.png", // 可自定义图标显示
    timeout: 50000, // 通知显示时间（毫秒）
    onclick: () => {
        window.focus(); // 点击通知时聚焦窗口
    },
});
```

- `GM_notification` 是 **Tampermonkey 提供的 API**，属于油猴脚本的扩展功能。

  它的作用是：在电脑系统层级发送通知

  GM_notification 能在浏览器后台时也弹窗，有图标、标题、文本。

在浏览器位于后台时，需要通过桌面通知来提醒同学来进行答题。