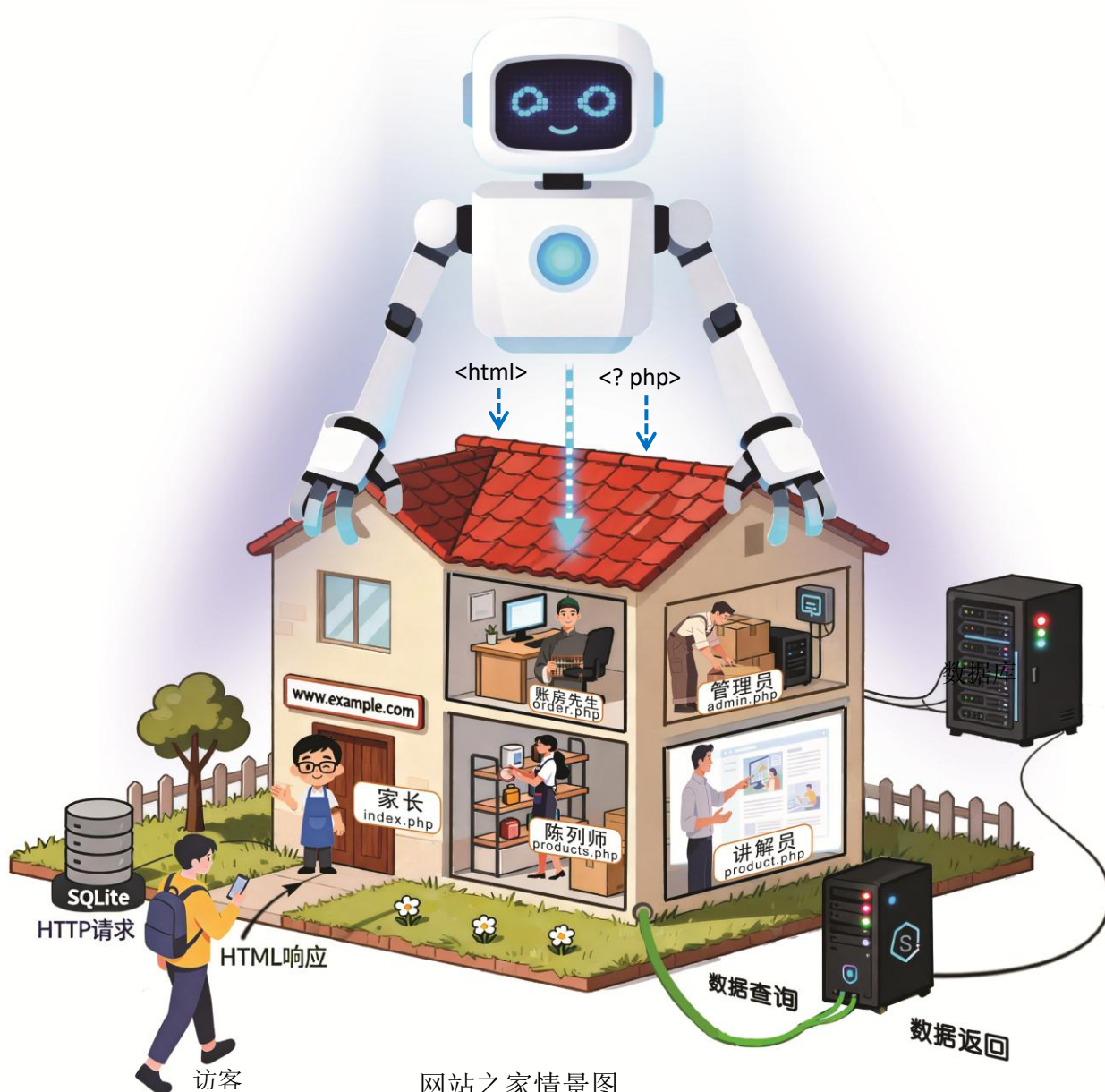


# 第 1 章 网站基础概念：从零理解网站是怎么工作的

## 本章学习目标

- 理解什么是网站（从建站角度）
- 理解网页的本质是什么
- 能区分静态网页和动态网页
- 理解客户端和服务器的什么、它们怎么“对话”
- 理解浏览器在整个过程中扮演的角色
- 能画出网站工作的完整流程图
- 了解网站制作技术发展的几个关键阶段
- 建立“网站是代码组成的，而 AI 可以帮助我们写代码”的初步认知

## 网站建设总设计师



# 1.1 什么是网站？一个“家”的比喻

在正式学习建站之前，我们先理解一个最基本的问题：网站到底是什么？

从网站建设的角度，我们可以这样定义：

网站，就是由网页及其网页元素，按不同组织方式，存放在不同目录里的一个系统。

这个定义听起来可能有点抽象，但没关系。我们用生活中的一个比喻，就能轻松理解。

💡 **技术说明：**严格来说，网站还需要服务器、域名等才能正常运行，但这些概念我们会在后面章节逐步介绍。这里我们先从“建站”的角度理解网站的内部组成。

## 把网站想象成一个“家”

- 网站 = 一个家
- 网页 = 家里的人（每个成员有不同的角色）
- 网页元素（文字、图片、多媒体等）= 家里的各种物品（家具、电器、装饰）
- 目录（文件夹）= 房间、衣柜（用来分类存放物品）

## 人与文件名的对应关系

在站点里，每个网页都是一个文件，有自己的名字。就像每个人都有自己的名字一样。

| 角色        | 网页文件名                  | 比喻解释                                |
|-----------|------------------------|-------------------------------------|
| 家长（爸爸/妈妈） | index.php 或 index.html | 家里的主心骨，访客最先见到的人。浏览器访问网站时，默认就会找这个文件。 |
| 陈列师       | products.php           | 负责展示所有商品。                           |
| 讲解员       | product.php            | 详细介绍某一件商品。                          |
| 账房先生      | order.php              | 记录交易明细。                             |
| 后台管理员     | admin.php              | 后台管理，整理商品、查看订单。                     |

表 1-1

## 为什么首页文件叫 **index**?

**index** 的意思是“索引”、“首页”。浏览器访问一个网站时（比如 `www.example.com`），会自动去寻找名为 `index.html` 或 `index.php` 的文件，把它显示出来。这就像你走进一个家，第一个见到的人就是家长。

## 🏠 理解“路径”：家里的地址

在我们的“网站之家”里，不仅要有成员，还要有地址。你需要知道每个成员住在哪个房间，才能找到他们。

- **根目录 (/):** 就是家里的“一楼大厅”。`index.php`（家长）通常就住在这里，因为他是门面。你走进这个家，首先见到的就是他。
- **子目录 (如 `/products/`):** 就是“一楼的商品展示区”。`products.php`（陈列师）住在这里。所有商品展示相关的文件都放在这个区域。
- **子目录 (如 `/product/`):** 就是“一楼的详情讲解区”。`product.php`（讲解员）在这里工作，为你详细介绍每一件商品。
- **子目录 (如 `/order/`):** 就是“二楼的订单处理室”。`order.php`（账房先生）在这里处理你提交的订单。
- **子目录 (如 `/admin/`):** 就是“二楼的仓库管理区”。`admin.php`（仓库管理员）在这里管理后台数据和商品库存。
- **URL 路径:** 当你访问 `www.mystore.com/products/` 时，就相当于你去朋友家做客：先在一楼大厅见到家长（`index.php`），然后家长指引你：“请到一楼的商品展示区，找陈列师（`products.php`）浏览商品。”如果你决定购买，订单会被送到二楼，由账房先生（`order.php`）处理，仓库管理员（`admin.php`）负责发货。

💡 **技术小贴士：**在站点里，文件也是放在不同文件夹中的。根目录（`/`）就是最外层的主文件夹，子目录就是里面的子文件夹。URL 中的路径（如 `/products/`）告诉浏览器要去哪个文件夹找文件。这个比喻我们后面还会用到，帮助你理解文件组织方式。

# 用这个比喻理解网站的定义

现在，我们来拆解刚才那个定义：

- “由网页及其网页元素” → 家里有人（网页），人手里拿着各种物品（网页元素）。
- “按不同组织方式” → 家庭成员有不同的分工，有的负责展示，有的负责导购，有的负责结账。
- “存放在不同目录里” → 家里的物品放在不同房间、不同衣柜里（比如图片放 images 文件夹，样式放 css 文件夹）。
- “一个系统” → 所有成员和物品协同工作，形成一个完整的家（网站）。

## 举个例子：一个咖啡电商网站的“家庭成员”

| 网页文件         | 角色     | 职责                |
|--------------|--------|-------------------|
| index.php    | 家长     | 介绍咖啡店品牌、主打产品、优惠活动 |
| products.php | 陈列师    | 列出所有咖啡豆（图片、名称、价格） |
| product.php  | 讲解员    | 详细介绍某款咖啡豆的产地、风味   |
| cart.php     | 购物车管理员 | 暂存用户选中的商品         |
| order.php    | 账房先生   | 记录已下单的订单号、金额、收货信息 |
| admin.php    | 仓库管理员  | 上架新商品、查看订单、发货     |

表 1-2

## 为什么这个比喻对建站有帮助？

- 当你以后向 AI 描述需求时，可以先想：我要组建一个什么样的家庭？需要哪些成员？每个成员负责什么？他们之间怎么沟通？
- 这个思维方式，就是五步建站法第一步“需求分析”的基础。

**课后思考：**如果你想做一个个人博客网站，按照这个比喻，你需要设计哪些“家庭成员”？每个成员对应什么文件名？

---

## 1.2 网页的本质：HTML 是“骨架”

在 1.1 节中，我们把网页比作“家里的人”。那么，一个人之所以能站立、行走、活动，靠的是什么呢？是**骨骼**。

网页也一样。一个网页（一个人）要想在浏览器中“站”起来，首先要有**骨架**。这个骨架就是 **HTML**。

如果你在浏览器里打开任何一个网页，右键点击空白处，选择“查看网页源代码”（或者按 Ctrl+U），你会看到一大堆由尖括号<>组成的文本。这就是网页的“真面目”——HTML 代码。

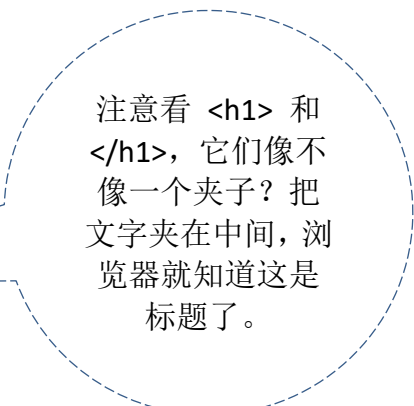
### HTML 是什么？

HTML 全称是 HyperText Markup Language，翻译过来叫“超文本标记语言”。你可以把它理解成一种“用标签描述内容”的语言。

打个比方：盖房子需要先搭骨架——哪里是柱子、哪里是梁、哪里是墙。网页也一样，HTML 就是网页的“骨架”。它告诉浏览器：这里放标题、那里放段落、这里放图片、那里放按钮。

### 看一个最简单的例子

```
html
<!DOCTYPE html>
<html>
<head>
  <title>我的第一个网页</title>
</head>
<body>
  <h1>欢迎来到我的网站</h1>
  <p>这是一段文字介绍。</p>
  
  <button>点击我</button>
```



注意看 `<h1>` 和 `</h1>`，它们像不像一个夹子？把文字夹在中间，浏览器就知道这是标题了。

```
</body>
</html>
```

这里面每一对尖括号就是一个“标签”，它们各有各的职责：

- `<h1>` —— 一级标题，最大最醒目
- `<p>` —— 段落，放正文文字
- `<img>` —— 图片
- `<button>` —— 按钮

浏览器读懂了这些标签，就知道页面上该显示什么、按什么顺序显示。

**但是，HTML 只有骨架**

光有骨架的人不能动，光有骨架的房子也不能住。所以除了 HTML，网页通常还需要：

- **CSS**（层叠样式表）——给骨架“穿衣打扮”，负责颜色、字体、布局、动画等。这就像给一个人穿上衣服、化妆打扮。
- **JavaScript**——给网页“注入灵魂”，负责交互和动态效果。这就像给一个人赋予肌肉和神经，让他能动起来、能响应你的操作。

**一个形象的比喻**

如果把网页比作一个人：

- **HTML 是骨骼**（支撑身体结构）
- **CSS 是皮肤和衣服**（外观颜值）
- **JavaScript 是肌肉和神经**（动作和反应）

这本书里，CSS 和 JavaScript 我们也会用到，但不用你写——AI 会帮我们生成。你只需要知道它们各自负责什么就行。

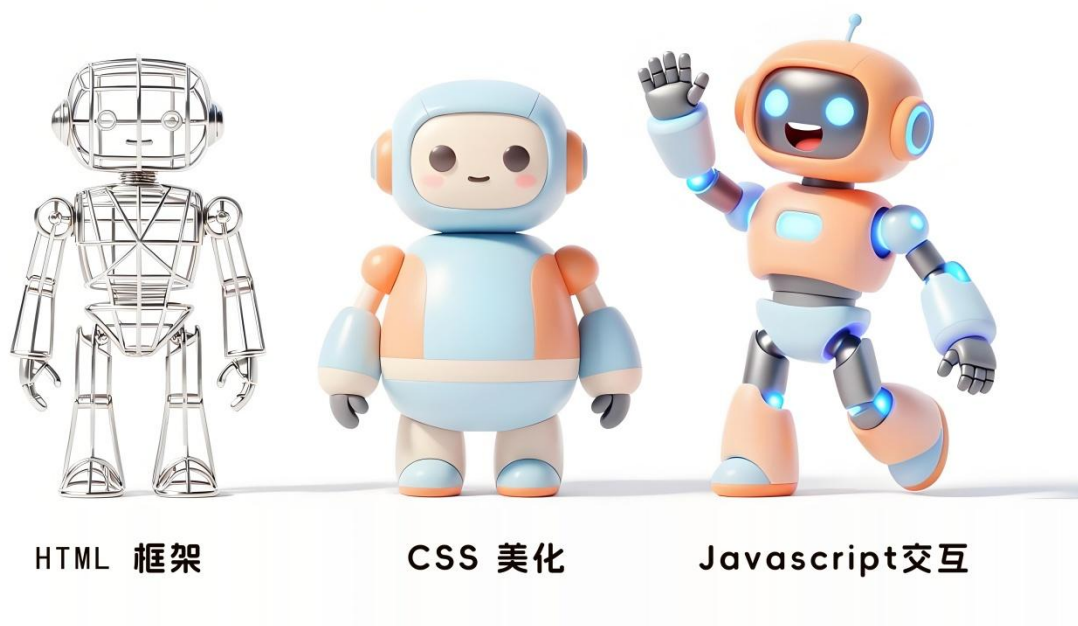


图 1-1

## 1.3 静态网页 vs 动态网页

打开一个网页，你怎么判断它是“静态”的还是“动态”的？一个简单的办法是：看不同人、不同时间访问时，内容会不会变。

### 静态网页——像公告板

公告栏上贴着一张通知：“本周五下午一点半起闭馆维护”。谁路过看到的内容都一样。静态网页就是这样——内容写死在 HTML 文件里。要修改内容必须手动改 HTML 文件。

- **识别特征：**通常以 .html 或 .htm 结尾。
- **特点：**每个人看到的内容一模一样；制作简单，访问速度快。
- **典型场景：**公司简介页、个人简历页、帮助文档。

### 动态网页——像寄存柜

寄存柜前，不同的人输入自己的取件码，打开不同的格口，取出属于自己的东西。动态网页的内容不是写死的，而是根据用户的操作、时间、数据库里的数据实时生成的。

比如你在淘宝搜索“蓝牙耳机”，页面会瞬间列出相关商品——这些商品是程序去数据库里查出来、当场拼成的页面。

- **识别特征：**传统上以 .php、.asp、.jsp 等结尾（但现在为了美观，常伪装成 .html 或无后缀）。
- **特点：**不同人、不同时间看到的内容可能不同；可以实现搜索、登录、留言、下单等交互功能。
- **核心机制：**服务器先执行程序代码（如 PHP），再生成 HTML 发给浏览器。

### 🔍 技术小贴士：如何一眼识破“伪装”？

现在的网站为了好看和 SEO（搜索引擎优化），常把动态网页伪装成静态网页（技术上叫“伪静态”）。

所以，最准确的判断方法不是看后缀，而是**看内容**：

- **看交互：**能不能登录？能不能搜索？能不能下单？
- **看个性化：**不同人看到的内容一样吗？（比如“欢迎回来，张三”）

只要答案是“能”，那就是动态网页！

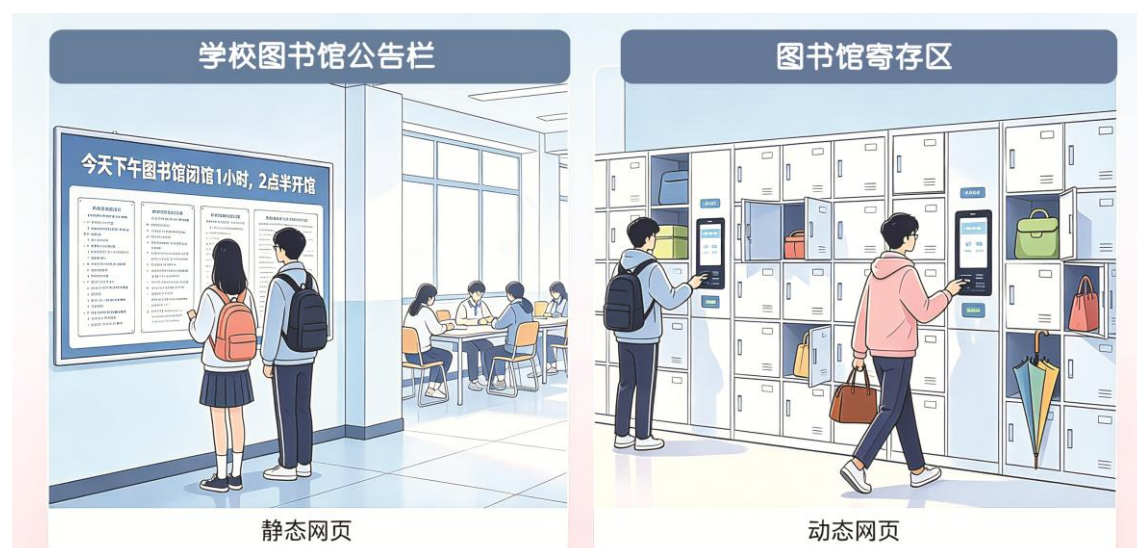


图 1-2

## 1.4 客户端与服务器

理解客户端和服务端是理解网站工作原理的关键。它们俩就像“顾客和厨师”。

### 客户端——你手中的设备

客户端就是你用来上网的设备（电脑、手机、平板）和里面运行的浏览器。它的任务是：发送请求、接收内容、渲染页面。

### 服务器——远端的“大本营”

服务器是一台 24 小时开机的远程电脑。它的任务是：接收请求、处理（可能查数据库）、返回结果。

例子：你在淘宝搜“蓝牙耳机”——你的手机（客户端）把关键词发给淘宝服务器，服务器去数据库搜索，把结果拼成网页返回，你的手机渲染出来。客户端负责“让你看得爽”，服务器负责“把事办妥”。



图 1-2

## 1.5 浏览器的作用

浏览器是客户端上最重要的软件。它的主要工作：

- **发送请求**：把你想看的网址告诉服务器
- **解析代码**：把 HTML/CSS/JS 翻译成可视化的页面
- **运行 JS**：让页面产生动态效果
- **管理缓存和 Cookie**：记住你的登录状态等

常见浏览器：Chrome、Edge、Safari、Firefox。

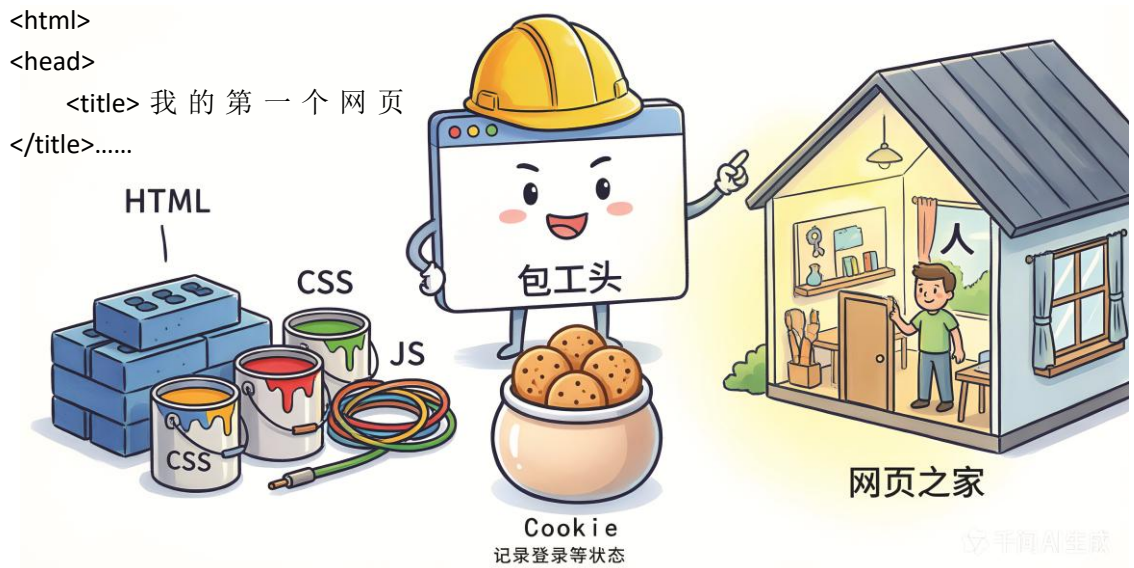


图 1-4

## 1.6 网站工作原理：核心流程

假设你在浏览器输入 `www.example.com` 并按回车：

1. **域名解析**：浏览器把域名转成 IP 地址（DNS 解析）
2. **建立连接**：浏览器与服务器建立 TCP 连接
3. **发送 HTTP 请求**：浏览器请求某个文件

4. 服务器处理：静态直接读文件，动态执行 PHP 等程序（可能查数据库）
5. 返回响应：服务器把 HTML 等内容返回给浏览器
6. 浏览器渲染：解析 HTML、下载 CSS/JS、执行，画出页面
7. 页面呈现：你看到了完整网页



。图 1-5

## 1.7 网站制作技术的几个关键阶段

| 技术阶段     | 代表工具                   | 交通工具比喻  | 核心特征             |
|----------|------------------------|---------|------------------|
| 手写代码     | 记事本                    | 🚶 两条腿走路 | 完全手工，每行代码都要自己敲   |
| 可视化工具    | Dreamweaver            | 🚲 电动车   | 拖拽生成，有辅助动力       |
| CMS      | WordPress              | 🚗 汽车    | 标准化，装主题插件就能用     |
| AI 辅助编程  | GitHub Copilot、Cursor  | 🚗 辅助驾驶  | AI 帮你写代码，但你需要懂编程 |
| AI 智能体建站 | <b>QClaw、WorkBuddy</b> | 🚗 全自动驾驶 | 你说需求，AI 自主完成建站   |

表 1-3

💡 为什么本书不教 **Cursor**，而要学 **QClaw**？

Cursor 是目前很火的 AI 编程工具，但它是“辅助驾驶”——你必须自己会写代码、会调试才能用好它，适合已经会编程的人。

而本书教的 QClaw 是“全自动驾驶”——你不需要懂代码，只要用自然语言描述需求，AI 智能体自己规划、写代码、生成网站。你只需要验收。

一句话：**Cursor** 让程序员写得更快；**qclaw** 让零基础学生自动写代码。

### 💡 延伸阅读：**Cursor** 也在向智能体进化

有意思的是，Cursor 正在全力向“智能体”方向转型。其内部已有超过 35%的代码提交由云端智能体自动完成，还推出了多智能体并行协作、云端开发环境等功能。这说明智能体主导开发是未来的大方向。无论用 QClaw 还是 Cursor，殊途同归。你现在用 QClaw 入门，就是为未来铺路。

**关键认知：**前四个阶段都需要你“懂技术”。第五阶段——AI 智能体建站——你只需要“会说话、会验收”。这就是当前网站开发的前沿方向——**Vibe Coding（氛围编程）**。你不用亲手写代码，而是用自然语言描述需求，AI 负责生成和修改，你通过不断对话“调氛围”，直到网站完全符合你的预期。

本书后面所有章节，你不需要手写任何代码，只需要学会描述需求。



---

## 1.8 关键认知：用“家”来结尾

回顾本章所有的比喻，它们其实可以串成一张完整的图：

**网站是一个家。** 家里每个人（网页）有自己的分工，人有骨骼（HTML）支撑身体，衣服（CSS）决定颜值，肌肉和神经（JavaScript）让人能动起来。不同的房间（目录）存放不同的物品。家需要一个地址（域名）让别人找到，需要一栋房子（服务器）安放。

建站技术前四个建站阶段，就像你自己设计图纸、搬砖、装修——你得懂建筑、懂水电、懂木工。而 AI 智能体建站，就像请了一个智能管家。你只需要说“我想要一个什么样的家”，管家帮你盖好、装修好、布置好。

本书要教你的，就是如何向这位管家清晰地描述你理想中的家。

---

## 课后练习

### 必做题

1. （推荐体验，但不强制）如果你想感受“手写代码”是什么感觉，可以用记事本输入下面的 HTML 代码，保存为 mypage.html，用浏览器打开看看效果。

如果你只想专注 AI 建站，可以直接跳过，或者用 AI 生成类似页面：“请生成一个简单的 HTML 页面，标题为‘我的第一个网页’，内容包含一级标题和两个段落。”

```
<html>
<head>
  <title>我的第一个网页</title>
</head>
<body>
  <h1>你好，世界！</h1>
  <p>这是我手写的第一个网页。HTML 是骨架，CSS 是衣服，JavaScript 是肌肉。</p>
  <p>我学的是电子商务网站建设，用 AI 智能体来建站。</p>
```

```
</body>
</html>
```

2. 画一张“网站工作原理流程图”。至少包含 6 个步骤，每步一句话说明。

### 思考题

1. 你平时访问的网站中，哪些是静态、哪些是动态？各举两例并说明理由。
  2. 如果向不懂技术的朋友解释建站技术五阶段，请完成填空：
    - 手写代码 = 自己一砖一瓦盖房子
    - 可视化工具 = 用乐高积木拼房子
    - CMS = 买精装房，自己配家具
    - AI 辅助编程 = 请一个智能助手帮你砌墙
    - AI 智能体 = \_\_\_\_\_
- 

## 本章彩蛋

你刚刚手写的那个 `mypage.html`，用的是“两条腿走路”的方式。而接下来，我们要直接跳到“全自动驾驶”——让 AI 帮你生成专业级别的网站代码。从手写第一行 HTML 到 AI 智能体建站，五阶段的演进，你将在这一门课里全部体验一遍。

如果把它放在服务器上，绑定一个域名，全世界的任何人都可以通过网址访问到它——这就是一个“网站”。虽然它很简陋，但它的工作原理和淘宝、京东、微博是一样的。从本质上讲，你已经迈出了建站的第一步。接下来，我们会让 AI 帮你写出更复杂、更专业的代码。

---

## 下一章预告

第 1 章我们把网站比作一个“家”，认识了家里的成员（网页）、骨骼（HTML）、衣服（CSS）、肌肉（JavaScript），还了解了家里的房间布局（目录）和地址（路径）。

但一个家光有这些还不够——它需要一个环境来安这个家，需要**水电煤气**才能运转。对应到网站，就是**服务器和运行环境**。

**下一章：本地开发环境搭建——为你的“网站之家”创建环境。**