

Omarchy 新手指南

装完之后的第一个 60 分钟，该做的事

火箭君CC

费曼式写法 · 由多个 AI 代理撰写与互相审校

导读

装完之后的第一个 60 分钟，该做的事

先别找鼠标。如果你刚装完 Omarchy，这句话可能正是你此刻的处境：屏幕干干净净，壁纸漂亮，顶上一条细栏，然后——没有了。没有图标，没有开始菜单，鼠标滑来滑去，像在对一面墙打招呼。

这不是系统坏了。这是它递给你的第一句话：**从今天起，键盘是方向盘，鼠标是副驾。**

这本书回答一个问题，也只回答一个：**装完 Omarchy 后的第一个 60 分钟，按什么顺序做什么，能让这块「什么都没有」的屏幕变成一台真正能干活、而且不怕折腾的机器？**

答案不是一张功能清单——网上多的是。答案是一条**顺序**：先学会三个「保命键」再谈别的，因为那是你在黑暗里摸到的手电筒；先理解窗口为什么自己排队，再装软件，不然你会和系统互相较劲；先更新、先确认「后悔药」快照在位，再开始折腾——顺序对了，这套以「固执」出名的系统会出奇地温顺。

Omarchy 是 Ruby on Rails 之父 DHH 给 Arch Linux 做的一次「厨师发办」

(omakase)：所有选择题他都替你答了，从窗口怎么排到终端用什么配色。你要做的不是从零搭建，而是入住——这本书就是你的入住指南：钥匙在哪、水电怎么开、卡住了找谁。

60 分钟后你成不了 Linux 高手，但你会拥有一台更新到最新、软件装齐、随时能读档重来的机器，以及一张「接下来一周练什么」的路线图。剩下的，交给肌肉记忆。

好，从第一章开始——先认识一下这位固执的厨师。

第一章 · 这是什么固执的系统

你刚装好的这个系统叫 Omarchy。名字是拼出来的：**Omakase** + **Arch**。Omakase（お任せ）是日语「交给你决定」——去吃日料时跟师傅说一句 omakase，就是「我不点菜，您看着上」。Arch 是它脚下的那个 Linux 发行版。这个名字就是整本说明书：**你别操心选择，让厨师来。**

一个固执的人，替你吵完了所有的架

Omarchy 的作者叫 **DHH**——David Heinemeier Hansson，Ruby on Rails 之父，37signals 的老板。这位老兄在程序员圈子里以「固执」出名：他坚信工具应该有明确的「正确用法」，而不是给你一千个开关让你自己纠结。他管这叫 opinionated——有主见的。软件替你选好路，你只管走。他先在 Ubuntu 上做过一版叫 Omakub，后来把地基换成了 Arch，就成了 Omarchy。

大白话

大白话：普通 Linux 像宜家，给你一堆板材和图纸，说「随便搭」；Omarchy 像精装房，DHH 已经替你吵完了「壁纸用什么颜色」「沙发摆哪」的所有架，你拎包入住。不喜欢他的品味可以改，但你第一天就能住。

这套「精装房」的地基是两块。第一块是 Arch Linux，一个以「软件永远是最新版」闻名的发行版——别人一年发一次大版本，它像自来水一样天天流新版本，这叫滚动更新。第二块是 Hyprland，一个平铺式窗口管理器：打开一个窗口，它自动占满屏幕；再开一个，屏幕自动切成两半，一人一边；再开，再切。**窗口自己会排队，永远不用拖标题栏找位置。**

为什么这件事值得你做

过去想用上 Hyprland 这种环境，你得先学会 Arch 那套出名的「手工安装仪式」，再对着 wiki 抄几百行配置，折腾一个周末，大概率中途放弃。Omarchy 把这个周末压缩成了一张安装盘：半小时后你得到的是 DHH 团队每天在用的那一整套——终端、编辑器、浏览器、主题、快捷键，全部调好、互相配得上。

更重要的是，37signals 这家公司在 2025 年宣布全员开发机器转向 Omarchy。意思是：有一群靠它吃饭的人每天替你踩坑、修坑。你搭的是一辆有人定期保养的顺风车。

这 60 分钟，你会得到什么

这本书只解决一个问题：**装完系统、坐在屏幕前发愣的第一个小时，按什么顺序做什么事，才能让它变成一台能干活的机器？** 走完这本书，你会：

- 学会三个「保命键」，从此不怕在黑屏幕前手足无措；
- 理解平铺窗口的思维方式——为什么它比鼠标拖窗口快；
- 用 Omarchy 自带的菜单装好你需要的软件，而不是一头栽进命令行文档；
- 给系统上好「后悔药」，以后怎么折腾都不怕；
- 知道卡住时去哪四条路求救。

它不会教你成为 Arch 专家。一小时后是「能用了」，一周后是「顺手了」，这就够了。

动手前唯一的警告

Omarchy 的安装器会**清空整块硬盘**——不是商量，是设计。它默认独占整台机器，并且顺手帮你做**全盘加密**——硬盘上所有数据都上了锁，开机要输密码才能解开（这本来是保护你的好事，前提是你知道它干了什么、记住了那个密码）。所以：

- 用一台**旧笔记本或备用机**练手，别拿存着全家照片的主力机开刀；
- 实在只有一台机器，先把重要文件拷走；双系统可以，但第一小时先别碰这个高级话题；
- 把这台机器当成新玩具，而不是旧生活的搬家现场。

大白话

大白话：装 Omarchy 之前问自己一句——「这块硬盘全空了，我心疼吗？」心疼就换块硬盘或换台机器。这是整本书里唯一可能让你真的哭出来的坑，我们已经替你插好旗了。

好，警告插完。下一章，我们去把系统装进机器里——比你想象的简单。

第二章 · 十分钟把系统请进门

上一章说好的警告插完了，现在真动手。安装 Omarchy 全程大约半小时，其中二十分钟是机器自己拷贝文件，你要做的决策只有四五个。这一章把每一步「它在干什么」讲清楚，你就不怕它了。

第一步：把系统塞进 U 盘

你需要三样东西：一台能上网的电脑（什么系统都行）、一个 **8GB 以上的 U 盘**（里面原来的东西会被清空）、二十分钟。

先去 Omarchy 官网（omarchy.org）下载 ISO——一个几 GB 的大文件，它是整个系统的「压缩镜像」，可以理解成一张光盘的照片。光有照片没用，得把它「洗」到 U 盘上，让电脑能把 U 盘当成光盘来开机。这个动作叫刷盘，推荐用 **balenaEtcher**（免费、三大系统都有、界面上只有三个按钮：选 ISO、选 U 盘、开始）。

大白话

大白话：ISO 是图纸，U 盘是地皮，balenaEtcher 是施工队。点完三个按钮去倒杯水，回来地就平了。

第二步：进 BIOS，关一个门卫

把 U 盘插进要装系统的机器，开机，在厂商标志出现的瞬间狂按某个键（常见是 **F2**、**F12**、**Del** 或 **Esc**，开机画面上一闪而过的小字会告诉你）进入 BIOS/UEFI 设置——主板的控制面板，系统在开机前最早醒来的那一小段程序。

在里面做两件事：

- **关掉 Secure Boot**。这是一个「门卫」功能，只放行微软签过名的系统启动。Omarchy 没在它的名单上，门卫会把你的安装盘拦在门外。找到 **Secure Boot** 选项，设成 **Disabled**。
- **把 U 盘设为第一启动项**（Boot Order / Boot Priority），让机器开机先读 U 盘。

保存退出（通常 **F10**），机器重启，这次它会从 U 盘醒来。装完系统想恢复 Secure Boot 可以以后再研究，第一小时先别管它。

第三步：回答安装器的几个问题

从 U 盘启动后你会看到 Omarchy 的安装界面。它问的问题不多，但有两个要想清楚：

- **装到哪块盘**：还记得第一章的警告吗？被选中的这块盘会被**完整清空**。插了多块硬盘的话，睁大眼睛选对。
- **加密密码**：安装器默认给整块硬盘上锁（全盘加密），这个密码**每次开机都要输**。选一个你肯定不会忘、别人猜不到的——忘掉它的后果约等于硬盘报废。

剩下的时区、键盘布局、用户名照实填。然后安装器开始干活：分区、加密、拷贝系统、装上桌面环境和那一整套 Omarchy 配置。全程不用你管。

大白话

大白话：这二十分钟机器在替你干 Arch Linux 历史上最劝退的那部分活——手动分区、配引导、装桌面。以前这一步要写一下午命令，现在你去吃个苹果。

第四步：拔 U 盘，进新家

装完提示重启，**记得拔掉 U 盘**（不然它又从 U 盘开机，你会以为没装上）。重启后先看到加密盘的密码输入框——这就是你刚才设的那个密码——然后用户名密码登录。

登进去的一瞬间，很多人会愣住：屏幕干干净净，顶上一根细条（时间、电量、几个小图标），中间一张壁纸，**没有图标，没有开始菜单，鼠标移来移去什么也没发生**。

别慌，没装坏。这正是下一章要解决的：在这个「什么按钮都没有」的屏幕上，怎么用三个键活下来。

第三章 · 第一次开机：别找鼠标

屏幕上什么都没有——这不是装坏了，这是 Hyprrland 的见面礼：它假设你的手长在键盘上。这一章只教三个键，学会就饿不死、困不住、丢不了。

先认识 Super 键

Omarchy 里几乎所有快捷键都围着 Super 键转——就是键盘上画着 Windows 旗子（Mac 键盘上是 ⌘ Command）的那个键。从现在开始，它是你的「系统键」：按住它再按别的键，就是在跟系统说话。这本书后面写 Super+空格，意思就是「按住 Super 不放，敲一下空格」。

保命键一：Super + 空格（要什么，喊一声）

按下 Super+空格，屏幕上方会弹出一个输入框——这叫启动器（Omarchy 用的是 Walker），功能类似 Mac 的 Spotlight 或 Windows 的开始菜单搜索。直接打字：想开浏览器就打浏览器名字的前几个字母，回车，它开了。这就是你在这套系统里打开一切东西的主要方式——没有图标可点，也不需要。

大白话

大白话：传统桌面是「东西摆在桌面上，你自己找」；启动器是「前台有个服务员，你报菜名」。一旦习惯报菜名，你会再也回不去翻图标的日子。

保命键二：Super + 回车（开终端）

Super+回车 打开终端——那个黑底白字、可以直接给系统下命令的窗口。在 Linux 世界里它是万能钥匙：装软件、看状态、改配置，终极手段都在这。现在先记住两件事：终端可以用 exit 回车关掉；以及你在网上看到的任何「在终端里输入.....」的教程，说的都是这个窗口。

保命键三：Super + K（忘了键位？问系统）

这是最温柔的一个设计。按 Super+K，屏幕会弹出一张**快捷键速查表**——Omarchy 把常用的键位全列在里面，还能搜索。也就是说：**你不需要背任何快捷键，只需要记住「忘了就按 Super+K」这一个。**

顺手再送两个马上用得上的：

- Super+W：关掉当前窗口（相当于别家系统的 Alt+F4）；
- Super+Esc：系统菜单——锁屏、重启、关机都在这里。

实战：连上 WiFi

光说不练假把式，来做第一件事：联网。插网线的可以跳过；用 WiFi 的，按 Super+Alt+空格——这是 **Omarchy 总菜单**（第五章会专门讲它），找到网络相关的选项，会打开一个叫 Impala 的 WiFi 管理界面：方向键选你的网络，回车，输密码，回车。顶栏的小图标亮起，通了。

如果连不上或者找不到选项，别硬猜——打开终端（还记得按什么吗），输入 `impala` 回车，是同一个东西。

鼠标呢？

鼠标没坏，能用——点击、滚动都行。但你会慢慢发现它越来越像个摆设：打开东西用启动器更快，关窗口 Super+W 更快，切窗口也有键。**这套系统不禁止鼠标，它只是让鼠标显得多余。**第一周别强迫自己纯键盘，哪个快用哪个，肌肉会自己投票。

现在你会开程序、会关窗口、会联网、忘了键位会问系统。下一个该理解的问题是：为什么开的窗口一个挤一个、还拖不动？那是下一章——平铺窗口的思维方式。

第四章 · 窗口自己会排队

你开了三个窗口，发现它们像排队打饭一样自动把屏幕切成了三份，没有重叠，没有缝隙，也找不到标题栏可以拖。这一章解释它在想什么——理解了，你会从「它怎么不听我话」变成「原来我以前的窗口管理才是体力活」。

两种窗口哲学：桌子 vs 相册

Windows 和 macOS 的窗口系统是「书桌派」：窗口是纸，随便叠、随便摊，乱了自己收拾，代价是你一辈子在干三件事——找窗口、拖窗口、把窗口拉大到合适尺寸。Hyprland 是「相册派」：屏幕是一本摊开的相册，**每开一个窗口，系统自动分给它一格，所有格子拼满整页，永不重叠、永不浪费**。开第四个？每个人让一让，重新平分。

大白话

大白话：桌子派把「窗口摆哪」这个决定每天推给你几百次；相册派替你做了，而且做得比你好——因为它从不手抖，永远严丝合缝。

所以第一个要扔掉的直觉是「拖窗口」。在平铺世界里没有拖动，只有**交换位置**和**调整布局**：

- **Super+T**：让当前窗口脱离队列、变成可以随便拖的浮动窗口（对话框、计算器这类小东西适合），再按一次归队；
- **Super+F**：当前窗口全屏，再按恢复；
- **Super+J**：切换分屏方向——下一个窗口开在右边还是下边；
- **Super+Shift+方向键**：把当前窗口和相邻的格子换个位置。

记不全没关系，**Super+K** 随时查。现在只要体会一件事：**布局是系统的事，你的事是「开」和「关」**。

工作区：你有十张桌面，不是一张

真正让平铺好用的另一半是工作区（workspace）——可以理解成十张独立的桌面。默认你坐在 1 号桌。按 **Super+2**，唰，换到一张全新的空桌子；再按 **Super+1**，回到原来那张，窗口们原地没动。**Super+3**、**Super+4** 最多到 9。

这是新手最容易「哇」出来的功能，因为它治一个老毛病：**窗口堆一屏、Alt+Tab 翻半天**。在工作区思维下，你给每张桌子分工：

- 1 号：浏览器（查资料）
- 2 号：编辑器 + 终端（干活）
- 3 号：聊天软件（摸鱼和被打扰专用）

想看聊天？**Super+3**。看完 **Super+2** 回去，干活的两个窗口连位置都没变过。**切换的是整张桌子，不是单个窗口**——这是它比 Alt+Tab 快的本质：你找的不是「窗口」，是「那件事」。

把窗口扔到别的桌子

开错桌子了？不用搬。**Super+Shift+数字** 把当前窗口直接送去那个工作区（人不过去）；想连人带窗口一起走，查一下 **Super+K** 里的对应键位。

为什么一小时后你就会觉得值

平铺 + 工作区加起来，消灭了三样隐形的日常损耗：鼠标找窗口的眼力、拖对齐的手力、以及「我刚才那个窗口呢」的心力。第一天你会觉得被管束，第三天你的手会先于脑子按下 **Super+2** ——那就是回本的时刻。

窗口会排队了，桌子会换了。但你很快会想要新东西：装个别的浏览器？换个主题？系统更新在哪？这些全归一个总开关管——下一章的 Omarchy 菜单。

第五章 · Omarchy 菜单：系统的遥控器

到现在你已经会开窗口、换桌面了。接下来的问题必然是：「我的微信/Spotify/VS Code 在哪装？」答案不是百度、不是官网下载页，而是一个你其实已经摸过的键：`Super+Alt+空格`——**Omarchy 菜单**。上一章用它连过 WiFi，这一章把它整个拆开看。

一个菜单，管住整个系统

按下 `Super+Alt+空格`，弹出一个全键盘操作的菜单。它是 Omarchy 的「控制面板 + 应用商店 + 设置中心」三合一，第一小时你会用到其中三类：

- **Install (安装)**：里面是一份精选软件清单——主流浏览器、编辑器、聊天工具、开发环境，方向键选中、回车，它替你下载、安装、配置好。不要去官网下 `.deb` 或 `.exe` 那样的安装包，Arch 世界不这么玩。
- **Theme (主题)**：一键换整套配色——壁纸、窗口边框、顶栏、终端，全部一起换。第九章细说。
- **Update (更新)**：系统升级入口。下一章专门讲，因为滚动发行版里这是头等大事。

大白话

大白话：Windows 装软件像「自己去各个厂家门市部提货」，Mac 是「App Store + 官网混搭」，Omarchy 菜单是「楼下有个代办处，报个名它全给你办好，还顺手登记在册」。

菜单背后：pacman 和 AUR

菜单只是前台，真正干活的叫 `pacman`——Arch Linux 的包管理器。你可以把它理解成系统的「总管家」：所有软件从哪里来（官方仓库）、装到哪、依赖什么别的软件、怎么干净地卸载，全由它记账。好处是装得干净、卸得也干净，不会像某些系统用久了满地残留。

官方仓库没有的软件（比如一些国产应用、新出的小工具），大概率在 AUR（Arch User Repository，Arch 用户仓库）——全世界 Arch 用户自发上传的「民间货架」，品类极全。

Omarchy 的菜单装软件时会自动兼顾两边，所以新手期你几乎不需要知道自己在用哪一个。

哪天你想自己动手，终端里搜一下软件名（`pacman -Ss 名字` 搜官方仓库），装一个用 `sudo pacman -S 名字`，卸载用 `sudo pacman -Rns 名字`——`sudo` 的意思是「以管理员身份」，系统会要你输密码确认。但第一个月，优先走菜单：它选的版本和配置都是 Omarchy 替你验过的。

第一小时推荐装的

别贪多，先把「今天就离不开」的装上：

- 你惯用的**浏览器**（如果自带的合手）；
- 你惯用的**编辑器**（VS Code、Zed 之类——第七章会讲为什么可以先试试自带的 Neovim）；
- 日常通讯和音乐软件。

装完按 `Super+空格` 打名字，能喊出来，就算落户成功。

一个重要的节奏提醒

Arch 是滚动更新的，装软件和更新系统是同一条河的两岸：**装新软件之前，最好先更新系统**（原因下一章讲透）。所以现在，别急着继续装——先跟我去上一章说过的「头等大事」：更新，以及给系统上好后悔药。

第六章 · 先更新，再上保险

第五章结尾留了个悬念：为什么装新软件之前最好先更新系统？这一章把它讲透，然后教你做第一小时里**最划算的一笔投资**——确认你的「后悔药」在。

滚动更新：一条不停流的河

回忆第一章：Arch 是滚动更新的——没有「Windows 11 升 12」那种几年一次的大版本，而是所有软件像河水一样天天流新版本。好处是你永远在用最新功能；代价是**河里的部件得一起往前走**。

关键在这：Linux 的软件互相依赖，A 软件可能要求「B 库至少 2.0 版」。如果你半年不更新系统（B 库还停在 1.0），今天却单独装了一个刚上架的新软件 A——A 要求 B 是 2.0，一跑就崩。这叫部分更新，是 Arch 官方唯一明令禁止的玩法。**所以在 Arch 世界，「装新东西」和「更新系统」必须结伴而行，这就是第五章那句提醒的原因。**

大白话

大白话：滚动更新像一列行进中的火车，所有车厢用同一批挂钩。你可以随时跳上车（装新软件），但不能只给某一节车厢换新型挂钩——会脱轨。要么整车一起升。

怎么更新

两条路，走哪条都行：

- 菜单派：`Super+Alt+空格` → **Update**，回车，输密码，等它跑完；
- 终端派：`Super+回车` 开终端，输入 `omarchy-update` 回车。

它会把 Omarchy 自身和整个 Arch 系统一起拉到最新。看到一堆软件名滚过去是正常的；偶尔它问你「要不要继续 (Y/n)」，回车就是同意。**建议频率：每周一次，或者每次装新软件前顺手一次。**别怕更新，它是这套系统的日常呼吸，不是大手术。

后悔药：快照

好，更新万一真把什么弄坏了呢？这就是 Omarchy 3 之后自带的一张安全网：快照（snapshot）。

原理一句话：Omarchy 用的硬盘格式（btrfs）能像拍照一样，**在每次大动作之前给整个系统存一个「那一瞬间的完整样子」**，几乎不占额外空间。每次系统更新前，它都会自动拍一张。

真出了事——比如更新完开机黑屏、桌面起不来——怎么吃药？重启电脑，在开机的启动菜单（Limine，就是那个列出系统选项的极简界面）里，能看到**以往快照的列表**。选更新前的那个，回车，系统就回到了「拍照那一刻」的样子，仿佛那场坏更新从未发生。进去之后再从容研究哪里出了问题。

大白话

大白话：快照就是游戏里的「存档点」。每次更新前系统自动存一档；玩砸了，读档重来。知道存档点在，你才敢放心折腾——这就是第一小时必须确认它存在的原因。

第一小时的两个确认动作

1. **跑一次更新**：Super+Alt+空格 → Update，亲眼看着它跑完。既是习惯的第一课，也顺便验证网络、权限一切正常。
2. **知道存档点在哪**：下次重启时留意开机菜单里的快照入口，眼熟一下位置。不用现在试，知道它存在就行。

系统更新到最新、后悔药确认在位——从现在起，你怎么折腾都翻不了车。下一章终于可以放心装点真东西了：你的编辑器、浏览器和开发环境。

第七章 · 把你的工具装齐

系统新鲜、保险上好，现在把这台机器变成你的机器。这一章按「先用什么装、再装什么、最后试试那个自带的家伙」的顺序来。

装机顺序：一律先走菜单

原则还是第五章那条： `Super+Alt+空格` → **Install**，在里面翻清单。第一小时的购物单：

- **浏览器**：自带的 Chromium 其实够用（就是 Chrome 的开源内核）。习惯 Firefox、Brave 之类，清单里都有，回车即装。
- **编辑器**：VS Code、Zed 都在清单里。但先别急着装——往下看第三节，Omarchy 自带了一个值得你给它十分钟的编辑器。
- **日常软件**：聊天、音乐、笔记，清单里有就装清单里的；没有的去 AUR 找（菜单装的时候会自动处理，你感觉不出来区别）。

装完一律用 `Super+空格` 喊名字验证。喊得出来 = 落户成功。

开发者加餐：mise 管版本

写代码的读者注意这个：Omarchy 自带 mise——一个「语言版本管家」。它解决的老大难问题是：项目 A 要 Node 18、项目 B 要 Node 22，以前你得装一堆互相打架的版本工具；现在 mise 一个全管，Node、Ruby、Python、Go 通吃。

用法直白：终端里 `mise use node@22`，当前目录就用上 Node 22；换个项目目录写个 `mise use ruby@3.3`，Ruby 3.3 就位。装语言、切版本、按项目隔离，三件事一个工具。Omarchy 菜单的 Install 里也有现成的开发环境套餐（Rails、Laravel、Node 等），本质就是替你调好 mise 加一堆周边——按你的技术栈选一个即可。

大白话：mise 像一个工具墙管理员，你要什么型号的扳手他递什么型号，还按项目分格子放，绝不拿错。以前这面墙要自己钉一星期。

给自带的 Neovim 十分钟

现在说那个「自带的家伙」。按 `Super+Shift+N` ——会打开 Neovim，一个纯键盘的终端编辑器，Omarchy 已经替你配好了 LazyVim（一套现成的流行配置，补全、搜索、文件树开箱即用）。

我知道，「纯键盘编辑器」听起来像上个世纪的遗物。但既然这本书叫「第一个 60 分钟」，就值得给它十分钟，因为它和你在第四章学的平铺思维是同一个哲学：**手不离键盘，眼睛不离内容**。三分钟生存指南：

- 进去是普通模式，按键是「命令」不是打字——按 `i` 进入插入模式才能写字，`Esc` 回普通模式；
- 普通模式下按 `空格`，会弹出命令面板（几乎所有功能都在里面，还能搜索）——这就是 Neovim 版的 `Super+K`；
- 存盘退出：普通模式输入 `:wq` 回车；不存盘强退：`:q!` 回车。

十分钟后的决定完全自由：觉得妙，它会成为你最快的编辑器；觉得累，回去菜单装 VS Code，毫无损失。但很多人就是在这十分钟里被「空格键弹出一切」种草的。

收尾检查

到这儿，你的机器应该已经有：惯用的浏览器、装好的日常软件、（可选）开发环境。工作区 1 放浏览器、工作区 2 放编辑器和终端——第四章的分工还记得吗？摆好之后，这套系统第一次有点「主力机」的样子了。

下一章我们掀开地板看一眼：这些主题、键位、布局，到底存在哪几个文件里？知道以后，你就敢改任何东西了。

第八章 · 配置文件的门道

到这一章，机器已经能干活了。现在掀开地板看一眼——不是折磨你，是因为 Omarchy 的配置设计得比你想的友好，看懂以后，「改键位」「换字体」这种事就从「求人」变成「三分钟」。

一切设置都是文本文件

先建立一个心理模型：Linux 桌面几乎所有设置都存在纯文本文件里，没有注册表、没有隐藏数据库。你的个人配置集中在一个目录：`~/.config/`（`~` 是你家目录的缩写，`.` 开头的文件夹默认隐藏，所以叫「点文件」dotfiles）。桌面环境 Hyprland 的那部分，就在 `~/.config/hypr/` 里面。

键位在哪个文件、输入法在哪个文件、显示器在哪个文件，分门别类，打开就是人话。比如想改键，找 `bindings.conf` ——里面每行长这样：

```
bind = SUPER SHIFT, F, exec, nautilus
```

不用教你也猜得到：「Super+Shift+F，执行，文件管理器」。这就是全部魔法——配置文件的阅读门槛约等于零。

两层楼：默认层和你的层

现在说 Omarchy 最重要的一个设计，也是新手最常踩的坑。配置分两层：

- **默认层：** `~/.local/share/omarchy/default/` ——Omarchy 出厂自带的全套配置。这一层不归你管：每次 `omarchy-update`，系统会用新版默认配置覆盖它。
- **你的层：** `~/.config/hypr/` ——你的个人修改。更新永远不动这一层。

推论只有一条，但是铁律：**永远不要改默认层，所有修改都写进你的层**。在默认层里改得再爽，一次更新就打回原形——社区里「我的修改怎么没了」的帖子，九成是这个原因。

大白话：默认层是房东的精装，你的层是你的私人物品。往墙上钉钉子（改你的层）随便；把承重墙砸了（改默认层），房东下次翻新直接给你砌回去，你的东西还埋在里面。

改一个键试试

实战：把「关闭窗口」从 `Super+W` 加成 `Super+Q` 也生效（很多人肌肉记忆在这）。打开终端，用第七章学的 Neovim（或任何编辑器）打开你的层：

```
nvim ~/.config/hypr/bindings.conf
```

加一行：

```
bind = SUPER, Q, killactive
```

存盘退出（`:wq`）。Hyprland 会热加载配置，通常**不用重启，立刻生效**——按 `Super+Q` 试试，当前窗口关了。改坏了也不怕：删掉那行，存盘，世界恢复。

为什么这套设计值得你感激

传统 Arch 用户的配置是「全靠自己攒」，升级全靠手动合并，几年后没人敢动。Omarchy 的两层结构让「系统进步」和「我的习惯」各走各的路：每次更新你白拿 DHH 团队的改进，你的私货一行不丢。这是 **omakase 精神的另一面——他不仅替你选了，还替你铺好了将来反悔的路。**

会看配置了，下一章干件开心的事：给它换张脸——主题、壁纸、顶栏，十分钟改头换面。

第九章 · 给它换张脸

干活的骨架都搭好了，这一章纯开心：十分钟把系统换成你的审美。别担心「美化 Linux 要折腾一星期」的老黄历——在 Omarchy 里，换脸是**一次按键**的事。

主题：一换全换

按 `Super+Alt+空格` → **Theme**。里面躺着十几套现成的主题——默认的 Tokyo Night（深夜霓虹）、Catppuccin（奶油色）、Gruvbox（复古暖棕）、Nord（极地蓝灰）……方向键上下翻的时候**整个桌面实时预览**：壁纸、窗口边框、顶栏、终端配色、甚至 Neovim 的配色，**全部一起换**。翻到顺眼的，回车，落户。

「全部一起换」这四个字，是 Omarchy 主题最值钱的地方。传统 Linux 美化的痛苦在于每个部件各管各的配色——终端一套、编辑器一套、顶栏一套，你像个装修工头一样挨个谈。Omarchy 的主题是一整套设计语言，**作者已经把所有部件调到互相配得上**，你只负责选风格。

大白话

大白话：别的系统换主题是「自己搭配一身衣服」，经常是格子衫配条纹裤；Omarchy 的主题是「成衣套装」，设计师配好了，你只管挑穿哪套出门。

壁纸：主题已经替你选了一批

每套主题自带一组配套的壁纸（都是按那套配色挑的）。还是在菜单里找背景/壁纸相关的选项，能在这套主题的壁纸库里轮换。想用自己的照片也可以——把图片丢进壁纸目录（菜单里会有入口），它同样走预览那一套。先用自带的不会丢人，那些壁纸都是精选。

顶栏：屏幕上唯一常驻的东西

屏幕顶上那条细栏叫 Waybar——状态栏。左边是当前工作区号码（第四章的「桌子」，哪张有窗口一目了然），右边是网络、音量、电量、时间。**它是这套「无图标」系统里唯一**

常驻的图形界面，作用类似飞机驾驶舱的仪表：不多，但关键的都在。

第一小时只要会读它就够：连没连上网看右边图标，自己在几号桌看左边数字。哪天想往上面加东西（天气、内存占用、自定义脚本），它是用配置文件驱动的——你现在知道该去 `~/.config/` 里找了，也知道别碰默认层。

字体和缩放（高分屏用户看这条）

如果你用 4K 屏，字可能小得像蚂蚁。这是高分屏 Linux 的经典第一课：调缩放（scaling）——告诉系统「把界面整体放大 1.5 倍」。设置在 Hyprland 的显示器配置里（`~/.config/hypr/` 下的显示器配置文件，改完即生效）。具体数值看眼睛舒服：1.5 和 2 是两个常见档位。拿不准就先 1.5。

换脸的边界

给个品味建议：第一个星期**只换主题，别自制**。Omarchy 这十几套主题的作者们在配色上花的心思，超过你周末两天的即兴发挥。等用熟了、知道自己真想要什么，再基于某套主题做微调——那时你已经是改配置的老手了。

系统现在是你的样子了。下一章换个严肃话题：总有卡住的一天，到时候去哪求救？

第十章 · 卡住怎么办

用 Linux，卡住不是「会不会」的问题，是「什么时候」的问题。好消息：卡住不可怕，**不知道怎么求救才可怕**。这一章给你四条退路，按从近到远排序。记住它们，你就永远不会困在黑屏前。

退路一：Super + K——八成的问题在这解决

新手期八成的「卡住」其实是「忘了键位」：想关窗口、想切桌面、想调音量，手悬在半空。别去搜，按 **Super+K**，速查表就在屏幕上，还能打字过滤。**遇到「系统能开、功能找不到」的问题，第一反应应该是它。**

退路二：官方手册——被低估的宝藏

Omarchy 有一本写得相当好的在线手册（在 omarchy.org 上能找到入口，叫 The Omarchy Manual）。它不像典型开源文档那样又臭又长，而是按「你想干什么」组织的：快捷键大全、怎么装应用、怎么配双显示器、怎么设输入法……**遇到「系统正常、但某个事不知道怎么做」的问题，查它。**手册里还有一页完整的 Hotkeys 列表，值得收藏。

退路三：社区——你踩的坑，前面有人插过旗

系统行为奇怪、疑似 bug？轮到社区出场：

- **GitHub 仓库的 Discussions** (github.com/basecamp/omarchy)：问答主阵地，先搜再发——你的问题大概率有人问过了；
- **Discord 和 Reddit (r/omarchy)**：实时闲聊和疑难杂症的分享地，社区气氛出奇地友好。

提问有个让回答率翻倍的姿势：**说清楚三件事——你想干什么、实际发生了什么、你用的版本号**（终端里 `omarchy-version` 可查）。「我的电脑坏了求救」没人接得住；「更新到 3.x 后 Super+Shift+F 没反应，版本 x.y.z」十分钟就有人回。

退路四：快照——天塌下来时的读档键

最严重的情形：更新或折腾之后，**开机进不了桌面了**。别慌，第六章的后悔药就是给这一天准备的：

1. 重启，在开机的 Limine 启动菜单里选**更新前的快照**，系统回到出事前的样子；
2. 进去以后，先去社区或 GitHub 搜症状，看看是不是已知问题、有没有修复；
3. 确认修复了再更新一次，或者干脆等几天——滚动更新的好处是，修复通常来得很快。

再补一个进阶常识，万一连登录界面都进不去：开机后按 `Ctrl+Alt+F3` 可以切到一个纯文字的登录终端（TTY，系统留的后门），用账号密码登进去就能敲命令排查；`Ctrl+Alt+F2` 或重启回图形界面。它像汽车的手动挡，平时用不上，关键时刻能救命。

一张求救决策表

- 忘了键位 → `Super+K`
- 不知道怎么做某事 → 官方手册
- 疑似 bug / 行为异常 → GitHub Discussions 先搜后问（带上版本号）
- 进不了桌面 → Limine 选快照读档，再排查

四条退路在手，这套系统对你来说已经没有「绝境」了。最后一章，把散落的珠子串起来：六十分钟之后，接下来一周怎么练，才算真正「上手」。

第十一章 · 六十分钟之后

六十分钟到了。回头看看你做了什么：装好系统、学会保命键、理解平铺和工作区、用菜单装了软件、更新了系统、确认了快照、给机器换了张脸、还背熟了四条退路。这台机器此刻的状态是——**能用、安全、是你的**。

剩下的差距只有一个：熟练。这一章是接下来一周的训练计划。

先认清一个规律：难受期大约三天

所有从鼠标世界转过来的人都会经历同一条曲线：**第一天新奇，第二天烦躁（手总比脑子慢），第三天开始顺，一周左右回本**。第二天的烦躁最容易让人放弃——「明明鼠标一下就行，我为什么要记这些键？」答案是：你正在把「想想再动」换成「肌肉自动」，每种值得的技能都要过这道坎，骑车、盲打、开车都一样。知道烦躁是过程的一部分而不是「我不适合」的证据，是熬过它的关键。

第一周训练表

不用背清单，每天只给自己加一个习惯，旧的保留：

- **第 1 天**：一切打开都走 `Super+空格`，一切关闭都走 `Super+W`。鼠标可以干别的，这两样不许碰。
- **第 2 天**：加 `Super+回车`（终端）和 `Super+F`（全屏）。
- **第 3 天**：强制工作区分工——浏览器固定 1 号桌、干活固定 2 号桌、聊天固定 3 号桌，切换只用 `Super+数字`。今天会有点烦，正常。
- **第 4 天**：加窗口调度——`Super+T` 浮动、`Super+Shift+方向键` 换位。
- **第 5 天**：给 Neovim 第二次机会（第七章那十分钟不算数，这次开一个真实的文本文件改点东西）。不习惯就换回你的编辑器，无压力。
- **第 6 天**：跑一次 `omarchy-update`，顺手翻翻 `Super+K` 里还没用过的键，挑一个顺眼的收编。
- **第 7 天**：改一个属于你自己的键位（第八章的方法）。能主动定制，就是毕业考试通过。

大白话：别试图一天背完快捷键表——那不叫学习，叫自虐。一天一个键，七天七个键，足够覆盖你 95% 的日常操作，而且每一个都是肌肉记住的，不是脑子记住的。

「上手」的三个标志

不用掐表，出现这三个信号就是上手了：

1. 你想开一个程序时，**手已经按出去了，脑子才想起来键名**；
2. 你不再注意到「没有图标」这件事——桌面空着反而让你安心；
3. 你第一次对别人说：「等等，我按个键」——然后事情就办完了。

再往后

上手之后的世界很大：工作区可以绑定到双显示器、配置文件里藏着动画和手势、dotfiles 可以放进 git 跟着你走南闯北、社区里天天有人分享新主题和新玩法。但那些都是想玩才去的游乐场，不是必修课。这本书承诺的只是第一个小时和第一周——到这里，承诺兑现了。

最后说句题外话。Omarchy 的作者把这套系统免费送出来的理由，和他当年写 Rails 一样：好工具不该只属于愿意折腾一星期的人。你现在拥有的，是一个顶尖工程师团队每天的日常武器，而你只花了六十分钟。

去用吧。键盘在你手里。