

## 第 16 章 附录



### 本章内容

- 16.1 编程环境安装
- 16.2 一起参与创作
- 16.3 术语表

## 16.1 编程环境安装

### 16.1.1 安装 IDE

推荐使用开源、轻量的 VS Code 作为本地集成开发环境（IDE）。访问 [VS Code 官网](https://code.visualstudio.com)，根据操作系统选择相应版本的 VS Code 进行下载和安装。

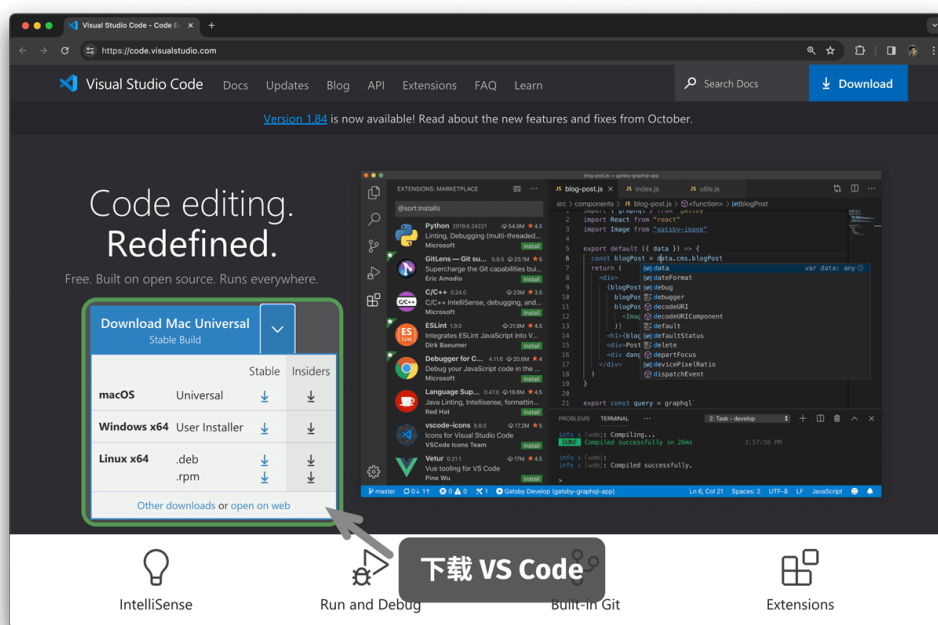


图 16-1 从官网下载 VS Code

VS Code 拥有强大的扩展包生态系统，支持大多数编程语言的运行和调试。以 Python 为例，安装“Python Extension Pack”扩展包之后，即可进行 Python 代码调试。安装步骤如图 16-2 所示。

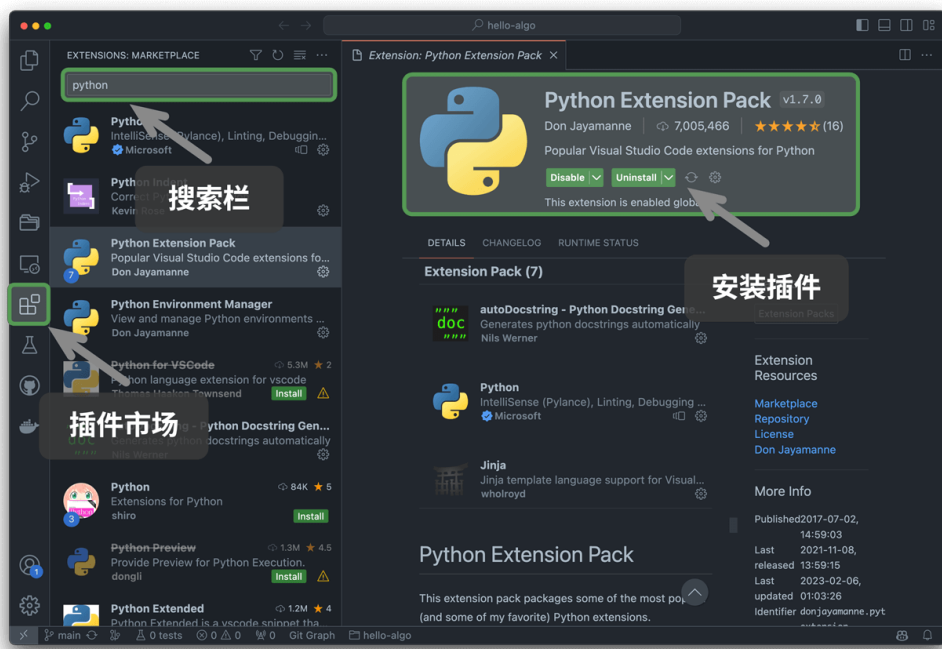


图 16-2 安装 VS Code 扩展包

## 16.1.2 安装语言环境

### 1. Python 环境

1. 下载并安装 [Miniconda3](#)，需要 Python 3.10 或更新版本。
2. 在 VS Code 的插件市场中搜索 `python`，安装 Python Extension Pack。
3. (可选) 在命令行输入 `pip install black`，安装代码格式化工具。

### 2. C/C++ 环境

1. Windows 系统需要安装 [MinGW](#) ([配置教程](#))；MacOS 自带 Clang，无须安装。
2. 在 VS Code 的插件市场中搜索 `c++`，安装 C/C++ Extension Pack。
3. (可选) 打开 Settings 页面，搜索 `Clang_format_fallback Style` 代码格式化选项，设置为 `{ BasedOnStyle: Microsoft, BreakBeforeBraces: Attach }`。

### 3. Java 环境

1. 下载并安装 [OpenJDK](#) (版本需满足 > JDK 9)。
2. 在 VS Code 的插件市场中搜索 `java`，安装 Extension Pack for Java。

#### 4. C# 环境

1. 下载并安装 [.Net 8.0](#)。
2. 在 VS Code 的插件市场中搜索 `C# Dev Kit`，安装 C# Dev Kit（[配置教程](#)）。
3. 也可使用 Visual Studio（[安装教程](#)）。

#### 5. Go 环境

1. 下载并安装 [go](#)。
2. 在 VS Code 的插件市场中搜索 `go`，安装 Go。
3. 按快捷键 `Ctrl + Shift + P` 呼出命令栏，输入 `go`，选择 `Go: Install/Update Tools`，全部勾选并安装即可。

#### 6. Swift 环境

1. 下载并安装 [Swift](#)。
2. 在 VS Code 的插件市场中搜索 `swift`，安装 [Swift for Visual Studio Code](#)。

#### 7. JavaScript 环境

1. 下载并安装 [Node.js](#)。
2. (可选) 在 VS Code 的插件市场中搜索 `Prettier`，安装代码格式化工具。

#### 8. TypeScript 环境

1. 同 JavaScript 环境安装步骤。
2. 安装 [TypeScript Execute \(tsx\)](#)。
3. 在 VS Code 的插件市场中搜索 `typescript`，安装 [Pretty TypeScript Errors](#)。

#### 9. Dart 环境

1. 下载并安装 [Dart](#)。
2. 在 VS Code 的插件市场中搜索 `dart`，安装 [Dart](#)。

#### 10. Rust 环境

1. 下载并安装 [Rust](#)。
2. 在 VS Code 的插件市场中搜索 `rust`，安装 [rust-analyzer](#)。

## 16.2 一起参与创作

由于笔者能力有限，书中难免存在一些遗漏和错误，请您谅解。如果您发现了笔误、链接失效、内容缺失、文字歧义、解释不清晰或行文结构不合理等问题，请协助我们进行修正，以给读者提供更优质的学习资源。

所有撰稿人的 GitHub ID 将在本书仓库、网页版和 PDF 版的主页上进行展示，以感谢他们对开源社区的无私奉献。

### 开源的魅力

纸质图书的两次印刷的间隔时间往往较久，内容更新非常不方便。

而在本开源书中，内容更迭的时间被缩短至数日甚至几个小时。

### 1. 内容微调

如图 16-3 所示，每个页面的右上角都有“编辑图标”。您可以按照以下步骤修改文本或代码。

1. 点击“编辑图标”，如果遇到“需要 Fork 此仓库”的提示，请同意该操作。
2. 修改 Markdown 源文件内容，检查内容的正确性，并尽量保持排版格式的统一。
3. 在页面底部填写修改说明，然后点击“Propose file change”按钮。页面跳转后，点击“Create pull request”按钮即可发起拉取请求。



图 16-3 页面编辑按钮

图片无法直接修改，需要通过新建 Issue 或评论留言来描述问题，我们会尽快重新绘制并替换图片。



2. 内容创作

如果您有兴趣参与此开源项目，包括将代码翻译成其他编程语言、扩展文章内容等，那么需要实施以下 Pull Request 工作流程。

- 1. 登录 GitHub，将本书的代码仓库 Fork 到个人账号下。
- 2. 进入您的 Fork 仓库网页，使用 `git clone` 命令将仓库克隆至本地。
- 3. 在本地进行内容创作，并进行完整测试，验证代码的正确性。
- 4. 将本地所做更改 Commit，然后 Push 至远程仓库。
- 5. 刷新仓库网页，点击“Create pull request”按钮即可发起拉取请求。

3. Docker 部署

在 `hello-algo` 根目录下，执行以下 Docker 脚本，即可在 `http://localhost:8000` 访问本项目：

```
docker-compose up -d
```

使用以下命令即可删除部署：

```
docker-compose down
```

16.3 术语表

表 16-1 列出了书中出现的重要术语，值得注意以下几点。

- 建议记住名词的英文叫法，以便阅读英文文献。
- 部分名词在简体中文和繁体中文下的叫法不同。

表 16-1 数据结构与算法的重要名词

English	简体中文	繁体中文
algorithm	算法	演算法
data structure	数据结构	資料結構
code	代码	程式碼
file	文件	檔案
function	函数	函式
method	方法	方法
variable	变量	變數
asymptotic complexity analysis	渐近复杂度分析	漸近複雜度分析

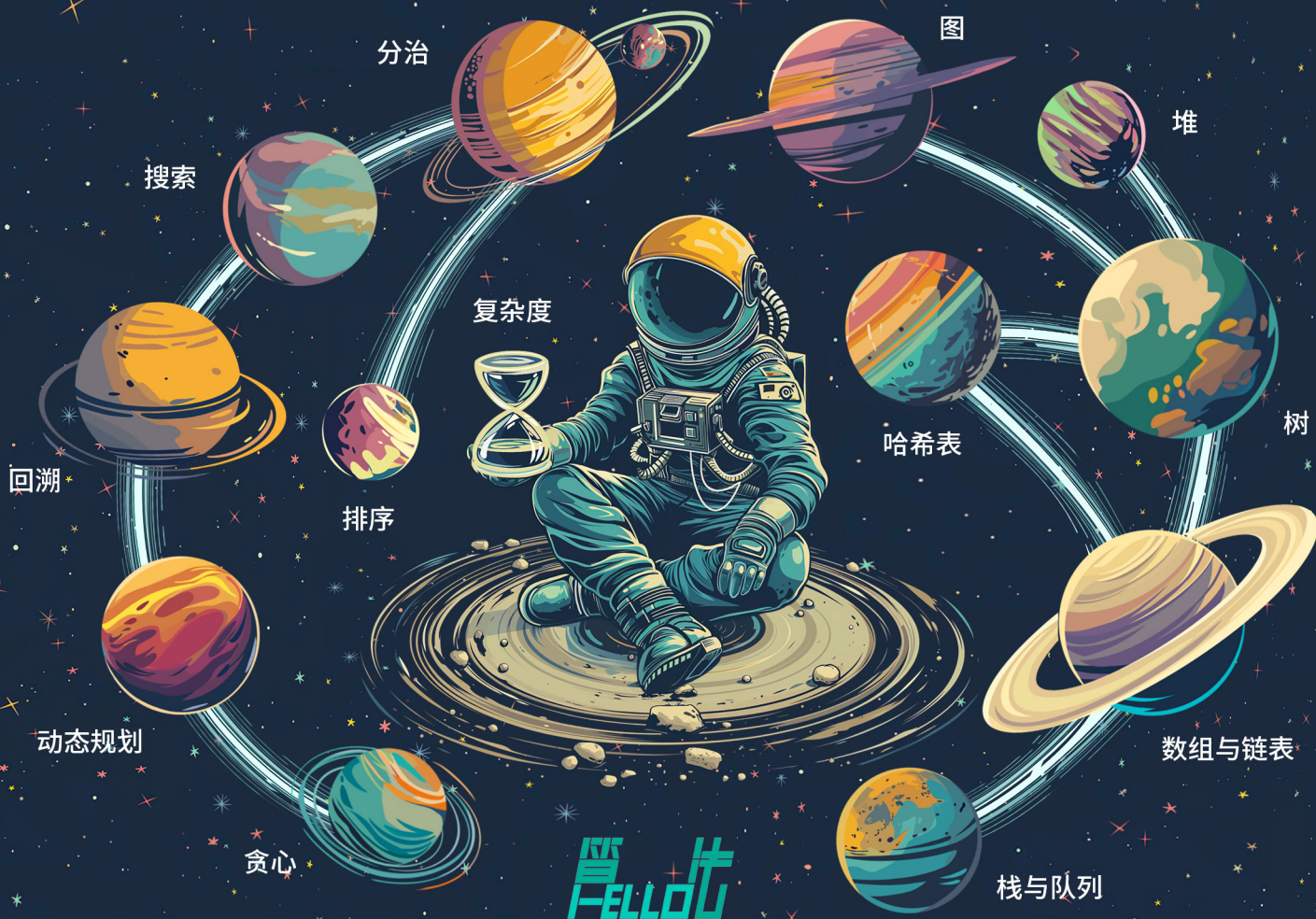
English	简体中文	繁体中文
time complexity	时间复杂度	時間複雜度
space complexity	空间复杂度	空間複雜度
loop	循环	迴圈
iteration	迭代	迭代
recursion	递归	遞迴
tail recursion	尾递归	尾遞迴
recursion tree	递归树	遞迴樹
big- $O$ notation	大 $O$ 记号	大 $O$ 記號
asymptotic upper bound	渐近上界	漸近上界
sign-magnitude	原码	原碼
1' s complement	反码	一補數
2' s complement	补码	二補數
array	数组	陣列
index	索引	索引
linked list	链表	鏈結串列
linked list node, list node	链表节点	鏈結串列節點
head node	头节点	頭節點
tail node	尾节点	尾節點
list	列表	串列
dynamic array	动态数组	動態陣列
hard disk	硬盘	硬碟
random-access memory (RAM)	内存	記憶體
cache memory	缓存	快取
cache miss	缓存未命中	快取未命中
cache hit rate	缓存命中率	快取命中率
stack	栈	堆疊
top of the stack	栈顶	堆疊頂
bottom of the stack	栈底	堆疊底
queue	队列	佇列
double-ended queue	双向队列	雙向佇列
front of the queue	队首	佇列首
rear of the queue	队尾	佇列尾
hash table	哈希表	雜湊表
hash set	哈希集合	雜湊集合
bucket	桶	桶

English	简体中文	繁体中文
hash function	哈希函数	雜湊函式
hash collision	哈希冲突	雜湊衝突
load factor	负载因子	負載因子
separate chaining	链式地址	鏈結位址
open addressing	开放寻址	開放定址
linear probing	线性探测	線性探查
lazy deletion	懒删除	懶刪除
binary tree	二叉树	二元樹
tree node	树节点	樹節點
left-child node	左子节点	左子節點
right-child node	右子节点	右子節點
parent node	父节点	父節點
left subtree	左子树	左子樹
right subtree	右子树	右子樹
root node	根节点	根節點
leaf node	叶节点	葉節點
edge	边	邊
level	层	層
degree	度	度
height	高度	高度
depth	深度	深度
perfect binary tree	完美二叉树	完美二元樹
complete binary tree	完全二叉树	完全二元樹
full binary tree	完满二叉树	完滿二元樹
balanced binary tree	平衡二叉树	平衡二元樹
binary search tree	二叉搜索树	二元搜尋樹
AVL tree	AVL 树	AVL 樹
red-black tree	红黑树	紅黑樹
level-order traversal	层序遍历	層序走訪
breadth-first traversal	广度优先遍历	廣度優先走訪
depth-first traversal	深度优先遍历	深度優先走訪
binary search tree	二叉搜索树	二元搜尋樹
balanced binary search tree	平衡二叉搜索树	平衡二元搜尋樹
balance factor	平衡因子	平衡因子
heap	堆	堆積



English	简体中文	繁体中文
max heap	大顶堆	大頂堆積
min heap	小顶堆	小頂堆積
priority queue	优先队列	優先佇列
heapify	堆化	堆積化
top- $k$ problem	Top- $k$ 问题	Top- $k$ 問題
graph	图	圖
vertex	顶点	頂點
undirected graph	无向图	無向圖
directed graph	有向图	有向圖
connected graph	连通图	連通圖
disconnected graph	非连通图	非連通圖
weighted graph	有权图	有權圖
adjacency	邻接	鄰接
path	路径	路徑
in-degree	入度	入度
out-degree	出度	出度
adjacency matrix	邻接矩阵	鄰接矩陣
adjacency list	邻接表	鄰接表
breadth-first search	广度优先搜索	廣度優先搜尋
depth-first search	深度优先搜索	深度優先搜尋
binary search	二分查找	二分搜尋
searching algorithm	搜索算法	搜尋演算法
sorting algorithm	排序算法	排序演算法
selection sort	选择排序	選擇排序
bubble sort	冒泡排序	泡沫排序
insertion sort	插入排序	插入排序
quick sort	快速排序	快速排序
merge sort	归并排序	合併排序
heap sort	堆排序	堆積排序
bucket sort	桶排序	桶排序
counting sort	计数排序	計數排序
radix sort	基数排序	基數排序
divide and conquer	分治	分治
hanota problem	汉诺塔问题	河內塔問題
backtracking algorithm	回溯算法	回溯演算法

English	简体中文	繁体中文
constraint	约束	約束
solution	解	解
state	状态	狀態
pruning	剪枝	剪枝
permutations problem	全排列问题	全排列問題
subset-sum problem	子集和问题	子集合問題
$n$ -queens problem	$n$ 皇后问题	$n$ 皇后問題
dynamic programming	动态规划	動態規劃
initial state	初始状态	初始狀態
state-transition equation	状态转移方程	狀態轉移方程
knapsack problem	背包问题	背包問題
edit distance problem	编辑距离问题	編輯距離問題
greedy algorithm	贪心算法	貪婪演算法



“Among the universe's 200 billion galaxies, encountering you, a shining star, is this book's great fortune.”

“在宇宙的 2000 亿个星系中，遇到你这颗闪耀的星星，是这本书的巨大幸运”