

青少年学Python

全彩印刷，讲解细致，寓教于乐，附赠教学辅导视频

250 多道判断题、100 道选择题和 100 道复习题，边学边练，学以致用

[希] 阿里斯提德·波拉斯 (Aristides S. Bouras)

著

[希] 卢卡雅·阿伊纳罗斯托 (Loukia V. Ainarozidou)

荣耀 程晶 译



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS



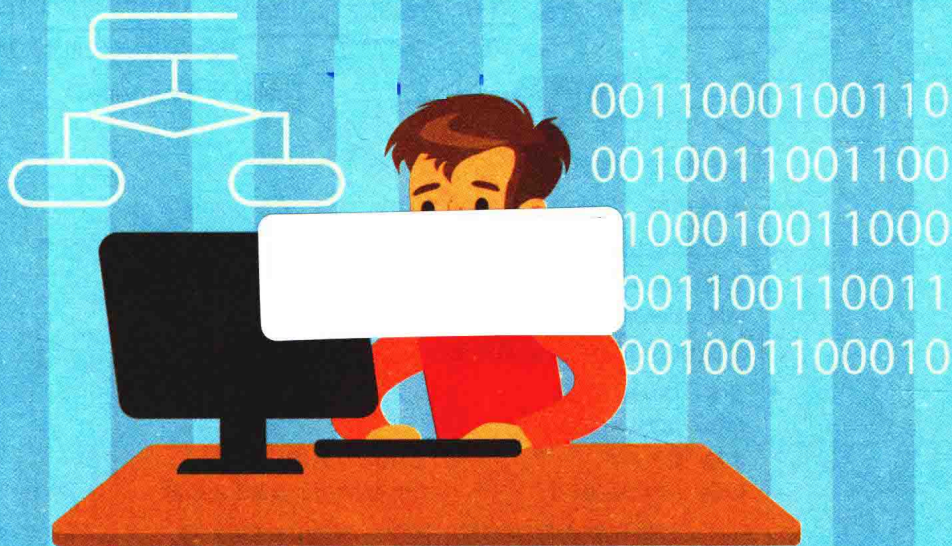
青少年学Python

[希] 阿里斯提德·波拉斯 (Aristides S. Bouras)

著

[希] 卢卡雅·阿伊纳罗斯托 (Loukia V. Ainarozidou)

荣耀 程晶 译



人民邮电出版社

北京

图书在版编目 (C I P) 数据

青少年学Python / (希) 阿里斯提德·波拉斯, (希) 卢卡雅·阿伊纳罗斯托著; 荣耀, 程晶 译. — 北京: 人民邮电出版社, 2018. 7

(少儿学编程)

ISBN 978-7-115-48357-7

I. ①青… II. ①阿… ②卢… ③荣… ④程… III. ①软件工具—程序设计—青少年读物 IV. ①TP311.561-49

中国版本图书馆CIP数据核字(2018)第086219号

版 权 声 明

Simplified Chinese translation copyright ©2017 by Posts and Telecommunications Press
ALL RIGHTS RESERVED

Python for Tweens and Teens, by Aristides S. Bouras Loukia V. Ainarozidou

Copyright © 2016 by Aristides S. Bouras Loukia V. Ainarozidou

本书中文简体版由作者授权人民邮电出版社出版。未经出版者书面许可, 对本书的任何部分不得以任何方式或任何手段复制和传播。

版权所有, 侵权必究。

-
- ◆ 著 [希]阿里斯提德·波拉斯 (Aristides S. Bouras)
[希]卢卡雅·阿伊纳罗斯托 (Loukia V. Ainarozidou)
- 译 荣 耀 程 晶
- 责任编辑 陈冀康
- 责任印制 焦志炜
- ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京缤索印刷有限公司印刷
- ◆ 开本: 720×960 1/16
印张: 18.75
字数: 334 千字 2018 年 7 月第 1 版
印数: 1—3 000 册 2018 年 7 月北京第 1 次印刷
- 著作权合同登记号 图字: 01-2018-1398 号
-

定价: 79.00 元

读者服务热线: (010)81055410 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字 20170147 号

内容提要

Python 是一门非常流行的编程语言，不仅有着非常广泛的应用，而且由于学习门槛较低，适合中小学生和青少年学习。

本书帮助读者利用 Python 语言进入编程世界。本书强调以计算和算法思维训练为导向，从计算机的工作原理和算法基础开始，详细介绍了 Python 编程的基础知识，进而引入了数据结构、过程式编程和面向对象编程等较为高级的话题。全书图文并茂，讲解细致，包含 100 多道已解答和 200 道未解答的练习，250 多道判断题，100 道多选题和 100 道复习题，可以帮助读者牢固地掌握所学的知识。

本书适合任何希望使用正确的习惯和技术开始学习或教授计算机编程的人，尤其适合 10 岁以上的孩子或者他们的父母和老师教孩子学习编程。

前言

作者简介

Aristides S. Bouras

Aristides S. Bouras 出生于 1973 年。早在孩提时期，他就发现自己对计算机编程充满热爱。他在 12 岁时得到自己的第一台计算机——一台 Commodore 64。这台计算机拥有基于 ROM 的 BASIC 编程语言和 64KB RAM（内存）！

他获得比雷埃夫斯技术教育学院计算机工程学位、色雷斯德漠克利特大学电子和计算机工程学位。

他曾在一家专门从事工业数据流和产品标签化的公司担任软件开发人员。他的主要工作是开发数据终端的软件应用程序，以及用于在数据库服务器上收集和存储数据的 PC 软件应用程序。

他开发了许多应用程序，如仓库管理系统、公司网站以及其他组织的网站。目前他是一名高中教师，主要教授计算机网络课程、Internet/Intranet 编程工具和数据库课程。

Loukia V. Ainarozidou

Loukia V. Ainarozidou 出生于 1975 年，她 13 岁时得到了自己的第一台计算机——拥有 128KB 的 RAM（内存）和 3 英寸软盘驱动器的 Amstrad CPC 6128！

她获得比雷埃夫斯技术教育学院计算机工程学位、色雷斯德漠克利特大学电子和计算机工程学位。

她曾在一家主营业务为水果和蔬菜包装的公司担任数据后勤部门主管。如今，她是一名高中教师。她主要教授计算机网络、计算机编程和数字设计课程。

致谢

特别感谢我们的朋友和资深编辑 Victoria (Vicki) Austin 不厌其烦地回答我们所有的问题——甚至是愚蠢的问题，以及在书稿编辑中所给予的慷慨帮助。没有她，这本书就难以释放全部的潜能。她的耐心指导和宝贵的建设性意见帮助我们在这本书提升到更高的水准！

谁应该买这本书

算法思维涉及的不仅仅是学习编写代码，而是解决问题的过程，只不过其中涉及学习如何编程！本书在教授计算和算法思维时假定读者对计算机编程一无所知！

毫无疑问，Python 是一门非常流行的编程语言。本书可以帮助读者利用 Python 语言进入编程世界。本书包括很多图解，100 多道已解决和 200 道未解决的练习，250 多道判断题，100 道多选题和 100 道复习题（可以在异步社区网站上找到解答）。本书适合任何希望使用正确的习惯和技术开始学习或教授计算机编程的人，尤其适合 10 岁以上的孩子或者他们的父母和老师教孩子学习编程。

本书使用的约定

以下是对本书中使用的约定的一些说明。“约定”是指显示特定文本的标准方式。

Python 语句

本书使用了大量以 Python 语言编写的示例。Python 语句以如下字体显示：

```
This is a Python statement
```

文本段落中的关键字、变量、函数和参数

关键字、变量、函数和参数有时显示在文本段落中。如果是这样，那么这些特殊的文字会以不同于该段落其余部分的文字显示出来。例如，`first_name = 5` 是一个段落文本中的 Python 语句例子。

以斜体显示的特殊文字

您可能会注意到，某些特殊字词（关键字、变量、函数和参数）也以斜体显示。当看到这些以斜体显示的特殊字词时，意味着它们是一般的类型，必须用适合您的数据的具体名称加以替换。例如，一条 Python 语句可能会显示为：

```
def name (arg1, arg2):
```

这条语句是以一般形式写的，这意味着它不完整。这种一般形式只是向您展示真实的语句大致的样子。为了完成该语句，关键字 `name`、`arg1` 和 `arg2` 必须替换为有意义的内容。在程序中使用该语句时，可以使用如下形式显示它：

```
def display_rectangle (width, height):
```

三个小圆点 (...) ——省略号

在语句的一般形式中，您可能还会注意到三个小圆点 (...) (即省略号)，跟在一个示例参数列表之后。这些小圆点并不是语句的一部分。省略号意味着您可以在列表中声明很多参数。例如，以下一般形式语句中的省略号：

```
display_messages (arg1, arg2, ...)
```

显示该列表可能包含两个以上的参数。当在程序中具体使用这条语句时，您的语句可能像这样：

```
display_messages (message_A, message_B, message_C, message_D)
```

方括号

一些语句或函数的一般形式可能包含方括号 []，这意味着括起来的部分是可选的。例如，以下一般形式的语句：

```
subject.sort([reverse = True])
```

说明 [reverse = True] 部分可以省略。

以下两个语句会产生不同的结果，但它们在语法上都是正确的（即它们都使用了正确的语法）：

```
x.sort()
x.sort(reverse = True)
```

深色标题

这本书的大部分例子都是采用如下显示方式：

```
file_29_2_3
```

```
a = 2
b = 3

c = a + b

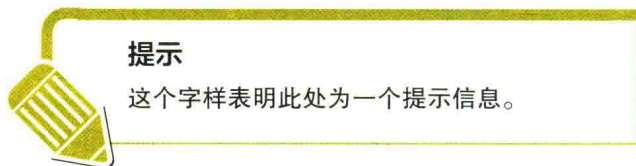
print(c)
```

顶部的深色标题 file_29_2_3 显示您必须打开进行试验的文件名。所有包含这种标题的示例均已免费开放在因特网上。您可以从以下地址下载它们：

<http://www.epubit.com>

提示

这本书经常使用提示框帮助您更好地理解一个概念的含义。提示框的样式如下：



已经了解或需要记住的内容

这本书时常帮助您回忆一些已经学过的知识（可能在前一章刚学过）。其他时候，它会指出一些您应该记住的内容。样式如下所示：

请记住！ 这个样式表示回忆某些知识或您应该记住的知识。



资源与支持

本书由异步社区出品，社区（<https://www.epubit.com/>）为您提供相关资源和后续服务。

配套资源

本书提供如下资源：

- 本书源代码；
- 书中彩图文件；
- 配套视频。

要获得以上配套资源，请在异步社区本书页面中点击 **配套资源**，跳转到下载界面，按提示进行操作即可。注意：为保证购书读者的权益，该操作会给出相关提示，要求输入提取码进行验证。

提交勘误

作者和编辑尽最大努力来确保书中内容的准确性，但难免会存在疏漏。欢迎您将发现的问题反馈给我们，帮助我们提升图书的质量。

当您发现错误时，请登录异步社区，按书名搜索，进入本书页面，点击“提交勘误”，输入勘误信息，点击“提交”按钮即可。本书的作者和编辑会对您提交的勘误进行审核，确认并接受后，您将获赠异步社区的 100 积分。积分可用于在异步社区兑换优惠券、样书或奖品。

The screenshot shows a web form titled "提交勘误" (Submit勘误) with three tabs: "详细信息" (Detailed Information), "写书评" (Write a Review), and "提交勘误" (Submit勘误). The form contains three input fields: "页码:" (Page Number), "页内位置 (行数):" (Position within page (Line Number)), and "勘误内容:" (勘误内容). Below these fields is a rich text editor with icons for bold (B), italic (I), underline (U), strikethrough (ABC), bulleted list (•), numbered list (1), link (🔗), and unlink (🔗). A red "提交" (Submit) button is located at the bottom right of the form.

扫码关注本书

扫描下方二维码，您将会在异步社区微信服务号中看到本书信息及相关的服务提示。



与我们联系

我们的联系邮箱是 contact@epubit.com.cn。

如果您对本书有任何疑问或建议，请您发邮件给我们，并请在邮件标题中注明本书书名，以便我们更高效地做出反馈。

如果您有兴趣出版图书、录制教学视频，或者参与图书翻译、技术审校等工作，可以发邮件给我们；有意出版图书的作者也可以到异步社区在线提交投稿（直接访问 www.epubit.com/selfpublish/submission 即可）。

如果您是学校、培训机构或企业，想批量购买本书或异步社区出版的其他图书，也可以发邮件给我们。

如果您在网上发现有针对异步社区出品图书的各种形式的盗版行为，包括对图书全部或部分内容的非授权传播，请您将怀疑有侵权行为的链接发邮件给我们。您的这一举动是对作者权益的保护，也是我们持续为您提供有价值的内容的动力之源。

关于异步社区和异步图书

“异步社区”是人民邮电出版社旗下 IT 专业图书社区，致力于出版精品 IT 技术图书和相关学习产品，为译者提供优质出版服务。异步社区创办于 2015 年 8 月，提供大量精品 IT 技术图书和电子书，以及高品质技术文章和视频课程。更多详情请访问异步社区官网 <https://www.epubit.com>。

“异步图书”是由异步社区编辑团队策划出版的精品 IT 专业图书的品牌，依托于人民邮电出版社近 30 年的计算机图书出版积累和专业编辑团队，相关图书在封面上印有异步图书的 LOGO。异步图书的出版领域包括软件开发、大数据、AI、测试、前端、网络技术等。



异步社区



微信服务号

目 录



1 第1章 计算机是如何工作的

- 1.1 引言 / 1
- 1.2 什么是硬件 / 1
- 1.3 什么是软件 / 2
- 1.4 计算机如何执行（运行）程序 / 2
- 1.5 编译器和解释器 / 3
- 1.6 什么是源代码 / 3
- 1.7 复习题：判断对错 / 3
- 1.8 复习题：选择题 / 4
- 1.9 复习题 / 6

2 第2章 Python和集成开发环境（IDE）

- 2.1 什么是Python / 7
- 2.2 Python是如何工作的 / 7
- 2.3 如何配置Python / 7
- 2.4 集成开发环境 / 8
- 2.5 IDLE / 8
- 2.6 如何设置IDLE / 8
- 2.7 Eclipse / 8
- 2.8 如何设置Eclipse / 9
- 2.9 复习题 / 11

3 第3章 基础算法概念

- 3.1 什么是算法 / 12
- 3.2 制作一杯奶茶的算法 / 12
- 3.3 什么是计算机程序 / 13
- 3.4 3个参与者 / 13
- 3.5 创建算法的3个主要阶段 / 13
- 3.6 什么是“保留字” / 14
- 3.7 您的第一个Python程序 / 15
- 3.8 语法错误、逻辑错误和运行时错误有什么区别 / 15
- 3.9 “调试”是什么意思 / 16
- 3.10 为您的代码添加注释 / 16
- 3.11 复习题：判断对错 / 17
- 3.12 复习题：选择题 / 18
- 3.13 复习题 / 19

4 第4章 变量与常量

- 4.1 什么是变量 / 20
- 4.2 变量类型 / 21
- 4.3 Python中的变量命名规则 / 22
- 4.4 “声明一个变量”是什么意思 / 22
- 4.5 复习题：判断对错 / 23
- 4.6 复习题：选择题 / 23
- 4.7 巩固练习 / 24
- 4.8 复习题 / 24

5

第5章 处理输入和输出

- 5.1 使用什么语句将消息和结果输出到用户的屏幕 / 25
- 5.2 如何修改print语句的默认行为 / 26
- 5.3 用户输入及提示 / 27
- 5.4 复习题：判断对错 / 29
- 5.5 复习题：选择题 / 29
- 5.6 复习题 / 30

7

第7章 使用IDLE

- 7.1 引言 / 40
- 7.2 创建一个新的Python模块 / 40
- 7.3 编写和执行Python程序 / 41
- 7.4 发现运行时错误和语法错误 / 42

9

第9章 编写第一个真正的程序

- 9.1 介绍 / 50
- 9.2 巩固练习 / 52
- 9.3 复习题 / 53

6

第6章 运算符

- 6.1 赋值运算符 / 31
- 6.2 算术运算符 / 32
- 6.3 什么是算术运算符的优先级 / 33
- 6.4 复合赋值运算符 / 34
- 6.5 字符串运算符 / 35
- 6.6 复习题：判断对错 / 36
- 6.7 复习题：选择题 / 37
- 6.8 巩固练习 / 38
- 6.9 复习题 / 39

8

第8章 使用Eclipse

- 8.1 引言 / 44
- 8.2 创建一个新的Python项目 / 44
- 8.3 编写和执行Python程序 / 47
- 8.4 发现运行时错误和语法错误 / 48

10

第10章 操作数字

- 10.1 引言 / 54
- 10.2 有用的函数和方法 / 54
- 10.3 复习题：判断对错 / 58
- 10.4 复习题 / 59

11

第11章 操作字符串

- 11.1 引言 / 61
- 11.2 从字符串中检索单个字符 / 61
- 11.3 提取子串 / 63
- 11.4 有用的函数和方法 / 65
- 11.5 复习题：判断对错 / 70
- 11.6 复习题：选择题 / 71
- 11.7 巩固练习 / 72
- 11.8 复习题 / 72

12

第12章 提出问题

- 12.1 引言 / 74
- 12.2 如何编写简单的问题 / 74
- 12.3 逻辑运算符和复杂的问题 / 76
- 12.4 Python成员关系运算符 / 78
- 12.5 逻辑运算符的优先顺序 / 78
- 12.6 算术、比较和逻辑运算符的
优先顺序 / 79
- 12.7 将自然语句转换为布尔表达式 / 80
- 12.8 复习题：判断对错 / 82
- 12.9 复习题：选择题 / 83
- 12.10 巩固练习 / 83
- 12.11 复习题 / 85

13

第13章 提出问题—if结构

- 13.1 if结构 / 86
- 13.2 复习题：判断对错 / 91
- 13.3 复习题：选择题 / 91
- 13.4 巩固练习 / 92
- 13.5 复习题 / 93

14

第14章 提出问题—if-else结构

- 14.1 if-else结构 / 94
- 14.2 复习题：判断对错 / 97
- 14.3 复习题：选择题 / 97
- 14.4 巩固练习 / 98

15

第15章 提出问题—if-elif结构

- 15.1 if-elif结构 / 100
- 15.2 复习题：判断对错 / 104
- 15.3 巩固练习 / 105

16

第16章 提出问题—嵌套结构

- 16.1 嵌套决策结构 / 109
- 16.2 复习题：判断对错 / 112
- 16.3 巩固练习 / 112
- 16.4 复习题 / 113

17

第17章 循环

- 17.1 循环结构 / 114
- 17.2 从顺序结构到循环结构 / 114
- 17.3 复习题：判断对错 / 116

18

第18章 循环—while结构

- 18.1 while结构 / 117
- 18.2 复习题：判断对错 / 122
- 18.3 复习题：选择题 / 123
- 18.4 巩固练习 / 124

19

第19章 循环——for结构

- 19.1 for结构 / 126
- 19.2 复习题：判断对错 / 130
- 19.3 复习题：选择题 / 130
- 19.4 巩固练习 / 131

20

第20章 循环——嵌套结构

- 20.1 嵌套循环结构 / 133
- 20.2 复习题：判断对错 / 135
- 20.3 复习题：选择题 / 135
- 20.4 巩固练习 / 137

21

第21章 循环结构使用技巧和窍门

- 21.1 引言 / 139
- 21.2 选择循环结构 / 139
- 21.3 “终极”规则 / 139
- 21.4 跳出循环 / 142
- 21.5 无限循环及如何避免 / 143
- 21.6 “由内而外”法 / 144
- 21.7 复习题：判断对错 / 145
- 21.8 复习题：选择题 / 145
- 21.9 巩固练习 / 146

22

第22章 循环结构专项练习

- 22.1 循环结构的一般性质的练习 / 148
- 22.2 巩固练习 / 157

23

第23章 海龟绘图

- 23.1 引言 / 159
- 23.2 x-y平面 / 159
- 23.3 海龟在哪儿 / 160
- 23.4 前后移动 / 161
- 23.5 左转和右转 / 162
- 23.6 将方向设置为指定的角度 / 165
- 23.7 设置延迟时间 / 166
- 23.8 改变笔的颜色和大小 / 167
- 23.9 把海龟的笔抬起或落下 / 168
- 23.10 将海龟直接移动到指定的位置 / 170
- 23.11 利用决策和循环结构控制海龟 / 171
- 23.12 巩固练习 / 181

24

第24章 Python中的数据结构

- 24.1 数据结构引介 / 184
- 24.2 什么是列表 / 185
- 24.3 在Python中创建列表 / 186
- 24.4 什么是元组 / 188
- 24.5 在Python中创建元组 / 188
- 24.6 如何从列表或元组中获取值 / 189
- 24.7 如何更改列表元素的值 / 192
- 24.8 如何遍历列表或元组 / 192
- 24.9 如何将用户输入的值添加到列表中 / 195
- 24.10 什么是字典 / 200
- 24.11 在Python中创建字典 / 200
- 24.12 如何从字典中获取值 / 201
- 24.13 如何更改字典元素的值 / 202
- 24.14 如何遍历字典 / 203
- 24.15 有用的语句、函数和方法 / 204
- 24.16 复习题：判断对错 / 207
- 24.17 复习题：选择题 / 211
- 24.18 巩固练习 / 214
- 24.19 复习题 / 217

25

第25章 数据结构专项练习

- 25.1 数据结构简单练习 / 218
- 25.2 如何在程序中使用多个数据结构 / 221
- 25.3 查找列表中的最大值和最小值 / 223
- 25.4 在数据结构中查找元素 / 227
- 25.5 复习题：判断对错 / 230
- 25.6 巩固练习 / 230

26

第26章 子程序简介

- 26.1 什么是子程序 / 233
- 26.2 什么是过程式编程 / 233
- 26.3 什么是模块化编程 / 234
- 26.4 复习题：判断对错 / 235
- 26.5 复习题 / 235

27

第27章 用户自定义的子程序

- 27.1 有返回值的子程序 / 236
- 27.2 如何调用有返回值的函数 / 237
- 27.3 无返回值的子程序 / 239
- 27.4 如何调用无返回值的函数 / 240
- 27.5 形参和实参 / 241
- 27.6 子程序是如何执行的 / 242
- 27.7 两个子程序可以使用相同名字的变量吗 / 243
- 27.8 一个子程序可以调用另一个子程序吗 / 244
- 27.9 默认参数值和关键字参数 / 244
- 27.10 变量的作用域 / 246
- 27.11 复习题：判断对错 / 248
- 27.12 巩固练习 / 249
- 27.13 复习题 / 250

28

第28章 子程序专项练习

- 28.1 一些额外的练习 / 251
- 28.2 巩固练习 / 256

29

第29章 面向对象编程

- 29.1 什么是面向对象编程 / 258
- 29.2 Python中的类和对象 / 259
- 29.3 构造方法和关键字self / 261
- 29.4 将初始值传递给构造方法 / 263
- 29.5 类变量和实例变量 / 264
- 29.6 Getter、Setter方法与属性 / 267
- 29.7 一个方法可以调用同一个类内的另一个方法吗 / 273
- 29.8 类继承 / 274
- 29.9 复习题：判断对错 / 277
- 29.10 巩固练习 / 278
- 29.11 复习题 / 282

第1章 计算机是如何工作的

1.1 引言

现如今，几乎所有工作都需要使用计算机。在学校，学生使用计算机上网搜索、发送邮件；在工作中，人们使用计算机制作报告、分析数据并与客户通信；在家里，人们使用计算机玩游戏，连接社交网络，与世界各地的人聊天。当然，不要忘记智能手机，比如 iPhone。智能手机也属于计算机！

计算机能执行如此之多不同的任务是因为它们具有编程能力。换句话说，计算机可以执行任何程序要它执行的任务。程序是计算机执行特定任务所遵循的一组语句（通常称为指令或命令）。

程序对计算机来说是必不可少的。如果没有程序，计算机就变成了傻瓜，无法做任何事情。实际上，是程序告诉计算机做什么以及何时做。另一方面，是程序员设计、创建和测试了计算机程序。

本书使用 Python 语言介绍计算机编程的基本概念。



1.2 什么是硬件

术语“硬件”是指组成计算机的所有设备或组件。如果您曾打开过计算机或笔记本电脑外壳，可能已经发现里面有许多组件，如微处理器（CPU）、内存和硬盘。计算机不是一个简单的设备，而是由多个设备一起组成的系统。典型计算机系统的基本组件包括以下几个部件。

- **中央处理器（CPU）**：这是计算机的一个重要组成部分，负责实际执行程序中定义的所有任务。
- **主存（RAM，随机存取存储器）**：这是计算机中保存程序（正被执行或运行）和程序正在处理的数据的区域。当您关闭计算机或将其从壁装电源插座拔下时，所有存储在 RAM 中的程序和数据都将丢失。
- **主存（ROM，只读存储器）**：ROM 是一种特殊类型的存储器，只能由计算机读取（但不能更改）。计算机关闭时，所有存储在 ROM 中的程序和数据都不会丢失。ROM 通常包含制造商的指令以及称为引导程序的程序，后者负责在电源接通后启动计算机系统。
- **辅助存储设备**：通常是指硬盘，有时（但很少时候）是指 CD / DVD 驱动器。与主存（RAM）相反，即使计算机断电了，这种类型的存储器也可以长时间保存数据。但是，