GOTOP



Python

基础案例教程

（基于计算思维）

李启龙 著
周春元 李 宁 译

- 知识点系统完整，讲解深入浅出
- 案例设计精巧，以易教、易学、典型、新鲜为导向
- 知识点与案例无缝衔接，轻松上手，加深理解
- 教学配套完善，实验环境、**全书源代码、授课PPT、习题答案轻松下载**

中国水利水电出版社
www.waterpub.com.cn

需要全本请访问 www.ertongbook.com

Python 基础案例教程
（基于计算思维）

李启龙 著

周春元 李 宁 译

中国水利水电出版社
www.waterpub.com.cn

·北京·

内 容 提 要

编写一本“难”的Python教材很容易，但编写一本“易”的Python教材却尤为不易。本书希望解决的问题，是让Python“教”起来得心应手，“学”起来轻松简单。

本书具有两个特点：一是案例力求选择“最新”的应用；二是力求把这些案例设计为“最简”模式，即案例中一切与知识点无关的内容全部去除，保证案例与知识点对应的精准性。这样，老师教学时容易教，同学们学习时无障碍。本书的每个知识点都配备了精简示例或案例，每一段示例或案例都配有完整代码，主要代码都配有详细的代码说明。同时为了便于教师的教学，还配备了精彩的电子教案。

本书适合作为各高校Python教材，同时也适合Python爱好者自学或参考。

北京市版权局著作权合同登记号：图字01-2019-0701

本书为碁峰信息股份有限公司独家授权出版发行的中文简体字版本。本书中文简体字版在中国大陆之专有出版权属中国水利水电出版社所有。在没有得到本书原版出版者和本书出版者书面许可时，任何单位和个人不得擅自摘抄、复制本书的全部或一部份以任何方式包括（数据和出版物）进行传播。本书原版版权属碁峰信息股份有限公司。版权所有，侵权必究。

图书在版编目（CIP）数据

Python基础案例教程 : 基于计算思维 / 李启龙著 ; 周春元, 李宁译. -- 北京 : 中国水利水电出版社, 2019.5

Python学程式设计运算思维 : 收录MTA Python微软国际认证模拟试题

ISBN 978-7-5170-7647-6

Ⅰ. ①P… Ⅱ. ①李… ②周… ③李… Ⅲ. ①软件工具－程序设计－教材 Ⅳ. ①TP311.561

中国版本图书馆CIP数据核字(2019)第079663号

责任编辑：周春元　　加工编辑：王开云　　封面设计：杨玉兰

书　　名	Python 基础案例教程（基于计算思维） Python JICHU ANLI JIAOCHENG (JIYU JISUAN SIWEI)
作　　者	李启龙　著　周春元　李　宁　译
出版发行	中国水利水电出版社 （北京市海淀区玉渊潭南路1号D座　100038） 网址：www.waterpub.com.cn E-mail：mchannel@263.net（万水） sales@waterpub.com.cn 电话：（010）68367658（营销中心）、82562819（万水）
经　　售	全国各地新华书店和相关出版物销售网点
排　　版	北京万水电子信息有限公司
印　　刷	三河市铭浩彩色印装有限公司
规　　格	184mm×240mm　16开本　17.5印张　397千字
版　　次	2019年5月第1版　2019年5月第1次印刷
印　　数	0001—3000册
定　　价	48.00元

凡购买我社图书，如有缺页、倒页、脱页的，本社营销中心负责调换

版权所有·侵权必究

学 Python，从“娃娃”抓起

—— 推荐序

以其开源、易学、易用的特点，再加上众多第三方开发包的加持，Python 编程语言在码界展现出旺盛的生命力和越来越强大的统治力。

关于 Python，我们所听说过的最传奇的一句话莫过于“学 Python，从娃娃抓起”。尽管调侃的意思更多一些，但毋庸置疑，学习 Python，越早开始越好。

关于 Python 学习的书有太多太多，涵盖了从入门到各专业领域深入应用的方方面面。其中，《毫无障碍学 Python》就是本人最喜爱的一本。

有经验的 Python 开发人员，可以用 Python“轻松”实现科学计算、绘图、爬虫、图像识别、人工智能等各种当下炙手可热的应用。但所谓的“轻松”，需要在熟练掌握 Python 的前提下才可以有。比如，一本讲爬虫的 Python 图书，无论对于专业人员多么有帮助，对于初学者来说，都像是在读“天书”一般。

相对于上述的“轻松”而言，虽然 Python 的最大特点号称是“易学”，但事实是，作为初学者，当你真正想要一本“易学”书来学习 Python 时，却是相当的不容易。

如果您是一名大学的 Python 老师，您更想找一本“易学”又“易教”的 Python 教材，那么，这个困难就更大一些。

基于此，我们策划了本书，希望一并解决 Python 的“教”“学”之苦。本书具有两个特点：一是案例力求选择“最新”应用；二是把这些案例设计为“最简”模式，我们把案例中一切与知识点无关的内容全部去除，保证案例与知识点对应的精准性。这样，老师教学时容易教，学生在学习时无障碍。

本书的每个知识点都配备了精简示例或案例，每一段示例或案例代码，都配有详细的代码说明。同时为了便于教师的教学，还配备了精彩的电子教案。扫描下面的二维码，可以下载本书的案例代码、电子教案、习题答案以及开发环境安装软件。

扫一扫：案例代码+电子教案+
习题答案+开发环境安装软件

目　录

学 Python，从“娃娃”抓起——推荐序

0 计算思维与计算机求解 …… 1

0-1 计算思维 …… 1
0-2 垂直与水平的逻辑思维 …… 2
0-2-1 垂直式思维 …… 2
0-2-2 水平式思维 …… 3
0-3 计算机求解的特点 …… 4
0-4 计算机求解的应用 …… 4
0-5 计算机求解步骤 …… 6
0-5-1 问题分析 …… 6
0-5-2 解题方法设计 …… 7
0-5-3 解题实现 …… 7
0-5-4 测试与改正 …… 7
0-6 计算思维体验 …… 7

1 Python 简介与开发环境安装 …… 9

1-1 编程语言简介 …… 9
1-2 Python 的起源及特性 …… 10
1-3 官方版 Python 开发环境 …… 11
1-3-1 官方版 Python 的下载与安装 …… 11
1-3-2 官方版 Python 开发环境的编辑与测试 …… 14
1-3-3 官方版 IDLE 集成开发环境的编辑与测试 …… 14
1-4 建议使用 Anaconda 套件开发 …… 17
1-4-1 Anaconda 套件的下载与安装 …… 17
1-4-2 Spyder 编辑器的编辑与测试 …… 21
1-4-3 Jupyter Notebook 编辑器的编辑与测试 …… 25
习题 …… 28

需要全本请访问 www.ertongbook.com

2 变量、数据类型与输入输出 ……29

2-1 变量的使用 ……29

2-1-1 变量的命名规则 ……30

2-1-2 变量的赋值 ……30

2-2 基本数据类型 ……32

2-2-1 数值型 ……32

2-2-2 布尔型 ……32

2-2-3 字符串 ……32

2-2-4 数据类型转换 ……33

2-3 print()输出函数 ……34

2-3-1 格式化输出功能 ……35

2-3-2 format()方法 ……36

2-4 input()输入函数 ……37

2-5 程序练习 ……38

习题 ……40

3 运算符与表达式 ……43

3-1 赋值运算符 ……43

3-2 算术运算符 ……44

3-3 关系运算符 ……48

3-4 逻辑运算符 ……50

3-5 复合赋值运算符 ……52

3-6 程序练习 ……53

习题 ……58

4 流程图与判断结构 ……60

4-1 流程图的表示符号 ……60

4-2 算法的基本结构 ……62

4-3 if 语句 ……63

4-4 if…else…语句 ……66

4-5 if…elif…else…语句 ……68

4-6 嵌套 if 语句 ……70

4-7 程序练习 …… 74
习题 …… 82

5 循环 …… 84

5-1 for 循环 …… 84
5-2 while 循环 …… 88
5-3 break …… 91
5-4 continue …… 92
5-5 循环嵌套 …… 93
5-6 程序练习 …… 96
习题 …… 101

6 数据类型 …… 103

6-1 字符串型的函数 …… 103
6-1-1 字符串的索引 …… 103
6-1-2 字符串函数 …… 104
6-2 列表 List …… 105
6-2-1 列表结构 …… 105
6-2-2 列表函数 …… 106
6-3 元组 Tuple …… 111
6-4 字典 Dict …… 111
6-4-1 字典数据的访问 …… 112
6-4-2 字典数据的操作 …… 112
6-4-3 字典操作相关函数 …… 113
6-5 集合 Set …… 116
6-5-1 集合元素的增删 …… 117
6-5-2 集合运算 …… 118
6-5-3 复合数据类型综述 …… 120
6-6 程序练习 …… 120
习题 …… 124

7 函数 …… 125

7-1 函数的定义与调用 …… 125

7-2 多个参数的函数的调用 …… 127
7-3 函数的返回值 …… 128
7-4 参数的传递 …… 130
7-5 模块与包 …… 132
7-5-1 导入一个包 …… 132
7-5-2 导入多个包 …… 134
7-5-3 安装第三方的包 …… 134
7-5-4 常用的内置函数 …… 136
7-6 递归函数 …… 136
7-7 程序练习 …… 140
习题 …… 144

8 文件处理 …… 145
8-1 文件路径基本概念 …… 145
8-1-1 绝对路径 …… 145
8-1-2 相对路径 …… 146
8-2 文件操作 …… 146
8-2-1 文件创建与关闭 …… 146
8-2-2 文件处理函数 …… 147
8-2-3 写文件操作 …… 148
8-2-4 读文件操作 …… 149
8-3 文件的目录操作 …… 151
8-3-1 os.path 包 …… 151
8-3-2 文件和目录的创建与删除 …… 155
8-3-3 检查文件是否存在 …… 160
8-4 程序练习 …… 161
习题 …… 166

9 网络服务与数据抓取及分析 …… 167
9-1 网络服务与 HTML …… 167
9-1-1 万维网 …… 167
9-1-2 域名服务器 …… 168
9-1-3 HTML 语法 …… 168
9-2 用 urllib 包解析网址及抓取数据 …… 170

9-2-1 网址解析函数 urlparse() ····· 170
9-2-2 网页数据抓取函数 urlopen() ····· 173
9-3 用 requests 包抓取网页数据 ····· 175
9-4 用 BeautifulSoup 包对网页进行解析 ····· 178
9-5 异常处理 ····· 184
9-6 程序练习 ····· 187
习题 ····· 192

10 图形用户界面 ····· 193

10-1 tkinter 包 ····· 193
10-2 tkinter 对象的基本方法 ····· 195
10-2-1 标签（Label） ····· 195
10-2-2 按钮（Button） ····· 199
10-2-3 用 Entry()方法创建输入框 ····· 201
10-2-4 用文本控件 Text()输入文本 ····· 202
10-2-5 滚动条控件（Scrollbar） ····· 205
10-3 tkinter 的高级控件 ····· 208
10-3-1 对话框控件（messagebox） ····· 208
10-3-2 复选按钮控件（Checkbutton） ····· 211
10-3-3 单选按钮控件（Radiobutton） ····· 213
10-3-4 图片（Photoimage） ····· 216
10-3-5 菜单控件（Menu） ····· 218
习题 ····· 222

11 绘制图表 ····· 223

11-1 Matplotlib 官方网站 ····· 223
11-2 绘制线条图 ····· 224
11-3 绘制柱状图 ····· 227
11-4 绘制饼图 ····· 229
11-5 与 numpy 包的综合运用 ····· 233
11-5-1 创建矩阵 ····· 233
11-5-2 矩阵运算 ····· 234
11-5-3 综合运算 matplotlib 与 numpy 来绘制曲线 ····· 236
11-6 绘制多图 ····· 237

习题 …… 241

12 图片处理与生成可执行文件 …… 242

12-1 pillow 包的安装 …… 242

12-2 pillow 包的功能 …… 243

12-2-1 图片属性 …… 244

12-2-2 改变图片色彩模式 …… 245

12-2-3 图片旋转 …… 247

12-2-4 图片滤镜 …… 248

12-2-5 图片的缩放 …… 250

12-2-6 向图片指定位置添加文字 …… 251

12-2-7 新建空白图片 …… 253

12-3 使用 ImageDraw 绘图 …… 254

12-3-1 线段绘制方法 line() …… 255

12-3-2 矩形绘制方法 rectangle() …… 256

12-3-3 绘制椭圆的方法 ellipse() …… 257

12-3-4 绘制弧线的方法 arc() …… 259

12-3-5 绘制扇形的方法 pieslice() …… 261

12-4 生成可执行文件 …… 263

习题 …… 265

习题答案 …… 267



计算思维与计算机求解

计算机是人类解决问题的好帮手，由于计算机具有速度快、容量大、计算精确、可以处理大量数据、重复作业等特性，非常适合帮助人类来解决各种问题。我们只要设计出正确的解题方法，就可以通过计算机来帮助解决问题。

在日常生活中，处处可见用计算机来解决实际问题的例子，小到通讯录或数据处理，大到国家实验室或企业的仪器设备，都需要使用计算机来完成各种操作。

计算机不像人类会自主思考解决问题，但如果我们能以计算机处理问题的方式，给予计算机正确的指令，那计算机就能按照我们的指示来处理问题。计算思维（Computational Thinking）就是指能够按计算机处理问题的方式，构思出各种计算方法来解决问题的思维能力。

0-1 计算思维

计算思维能力是每个人除了听、说、读、写等基本素养外，也应具备的基本能力，此能力并非专属于计算机科学家；计算思维是利用归纳、嵌套、变换或模拟等方法，将复杂问题转化为我们所熟悉的模式，以利问题的解决。

因此，计算思维具有以下特性：

- 是一种基本的素养，并非死记硬背的技能。
- 不是指编写计算机代码。
- 是人类解决问题的方法或策略。
- 结合了数学及工程的思维。
- 是一种概念或构思，并非指相关作品。
- 适用于每个人与每个地方，是人人都需具备的能力。

美国的计算机科学教师协会（Computer Science Teacher Association，CSTA）将计算思维定义

需要全本请访问 www.ertongbook.com

为计算机可执行的解决问题的策略，包含数据搜索、数据分析、数据表示、问题分解、抽象化、算法与程序、自动化、仿真及并行计算等概念。它所提出的计算机科学核心能力指标中，将计算思维视为贯穿整个课程的重要理念。通过计算思维，来培养学生解决问题、系统设计、知识创新的能力并了解信息科技的能力与限制，该协会的网站网址为：https://www.csteachers.org/，如图 0-1 所示。

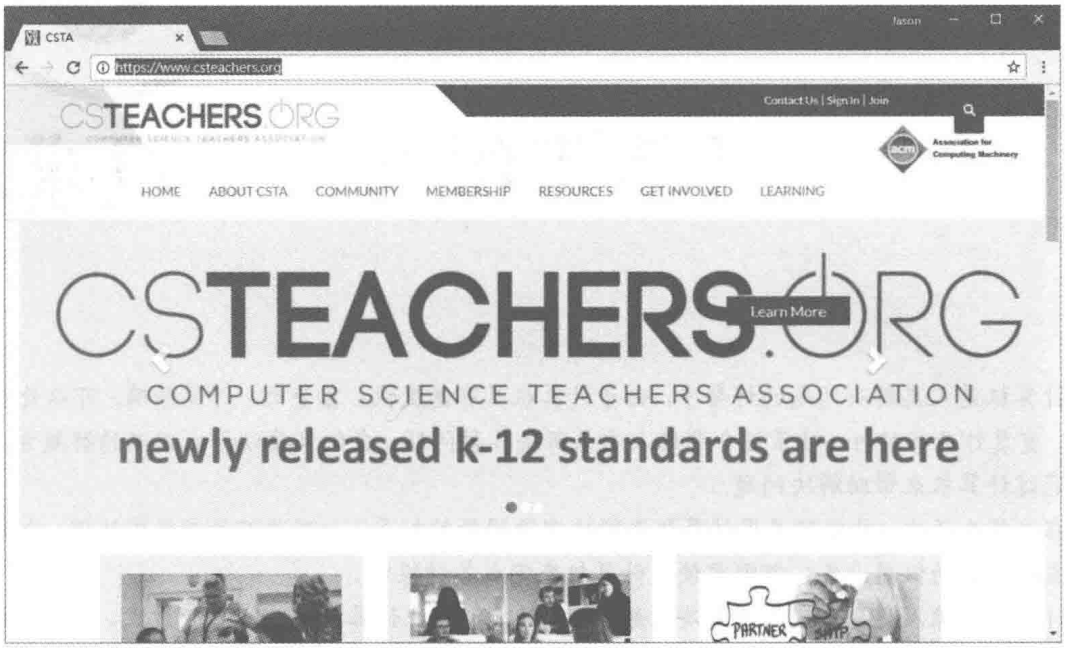

图 0-1

0-2 垂直与水平的逻辑思维

计算机要解决问题，必须由人来设计其处理问题的步骤，然后再由计算机根据我们输入的数据，进行各种计算、搜寻、比较等运算来得到答案。人类在思考问题的解决方法时，大致可以分为“垂直式思维”与“水平式思维”两大类。

0-2-1 垂直式思维

垂直式思维（Vertical Thinking），又称收敛性思维，是指循序渐进地从问题本身开始思考，讲究步骤，每一步骤及阶段都必须是确定的，思路会朝着问题的答案集中收敛，通过垂直式思维所获得的结论比较具有正确性、系统性及普遍性。

垂直式思维比较符合计算机求解的特点。其答案比较具有确定性，并尝试将问题的解决分解为可实施的步骤。垂直式思维具有以下优点：

（1）所获得的逻辑推理结论较具有正确性、系统性及普遍性，较适合计算机进行问题解决。

（2）因其步骤的明确性，每一次进行推论，都可以得到一致的答案，具有一致性。

（3）较具有实用价值，只要掌握了推理的原则，就可以相互检验。

垂直式思维也有一些缺点，包括：结论会受前提所影响、思考时比较容易形成框架造成画地为牢的现象、不利于创新等。

在日常生活中，我们经常用垂直式思维的方式思考问题。比如，我们要规划一趟旅行，就要从决定日期开始，依次选定旅游目的地、规划交通工具与路线、确认参与人数、安排住宿与餐饮、计算各项费用等，这就是一个垂直式思维的例子，如图 0-2 所示。

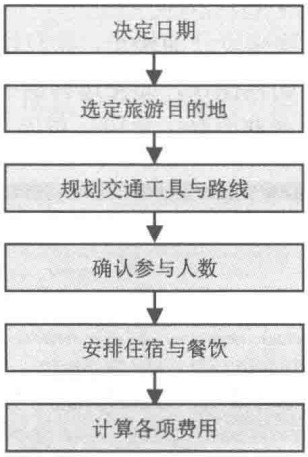

图 0-2

0-2-2　水平式思维

水平式思维（Lateral Thinking）又称发散性思维，是从问题本身向各个方向去思考发散，可能会找出各个不同的答案，常应用于创造性思维训练，常见的方法有自由联想法、脑力激荡法等。水平式思维较不具有计算机求解所需的绝对性、有限性、唯一性等特点，但是在创新创意方面有很重要的作用，有助于我们跳出框架去思考。

有一则水平式思考的故事提到，某位农夫不小心把一块怀表遗失在稻草堆里，他找了一群孩子，请他们帮忙寻找，这群孩子相当认真地用眼睛到处寻找怀表，但是都找不到，后来又来了一位小孩，他趴在地上，不用眼睛却用耳朵去听怀表滴答滴答的声音，找出怀表掉落的方向后，很轻松地就拨开草丛找到了遗失的怀表。这个孩子，就是跳脱了用眼睛找东西的垂直式思维方式。

人们在面对问题时，习惯于遵循已有的认识、理论或知识，然后再沿着既定的方向系统地深入思考，而计算机解决问题的方式也趋近于 Step by Step 的模式，上一步的工作完成后，才会接着进行下一步，因此对计算机而言，垂直式的思维方式比较容易转化成实际的解题程序；不过水平思维更具有突破现状的特性，较易于找到新的解题方向，因此在使用计算机求解时，也需善用水平式思维，向着不同方向找寻解题方法。垂直思考与水平思考两者间并不矛盾，有时还能相辅相成，水平式思维有利于产生新想法，而垂直式思维则有利于对水平思考衍生出的想法进行发展。

0-3 计算机求解的特点

计算机求解的特点就是会根据我们设计的步骤顺序地执行，每次执行都会获得一致的结果。由于垂直式思维的逻辑推理结论较具有正确性、系统性及普遍性，大多都能转换成可以执行的步骤来解决各种问题。

当我们要解决的问题比较复杂或庞大时，可以采取顺序式的解决问题方式。先将大问题分成几个较小的问题，再设计较小问题的顺序执行方案。

日常生活中，有许多应用顺序流程来设计的例子，我们炒菜时就是按照一定的顺序来处理；网购时，也是按照操作顺序来完成各项购物操作，如先选择商品、填写信息后完成付款，如图 0-3 所示；使用自动提款机进行交易时，也需要先输入密码、再选择交易方式及输入相关金额等。



图 0-3

顺序式的流程就是会按照一定的次序，逐步完成各个步骤，最后获得预期的结果。这样所设计出的解题步骤过程明确、顺序清楚，所以非常适合用计算机来处理。

0-4 计算机求解的应用

计算机求解的应用领域相当广泛，只要是计算机所提供的服务，其背后都可以观察到计算机求解的过程。常见的计算机在各领域的应用实例包括：网络购物系统、电子商务系统、搜索引擎系统、医学工程系统、气象预测系统、校务行政系统、电子地图应用、各种数学问题的解决等。下面以生活中的电子地图规划路线作为例子，来说明隐藏在系统背后的计算机求解的应用。

电子地图的规划路线功能，就是计算机求解的一个应用。计算机根据用户输入的起点与终点位置，来规划可行的路线，并且还可让用户选择交通方式，如自驾、公交车、步行等。此处是以天安门为起

点，故宫博物院为终点，并且选择自行开车的方式，来测试电子地图规划路线的功能，如图 0-4 所示。

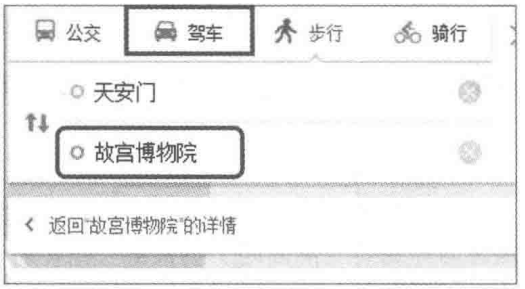

图 0-4

百度地图规划出来的路线如图 0-5 所示。它会在地图上标示出路线，我们在使用时可以放大或缩小地图的显示比例，以便我们能看清楚交通路线。



图 0-5

除了显示路线之外，知道路名与路线距离也是非常重要的信息。Google 地图会把规划出来的路线，详细地显示出相关的信息，如会经过哪些路线、每段路线的距离等，如图 0-6 所示。

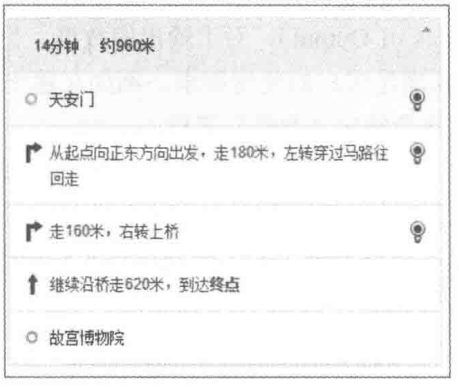

图 0-6

Google 地图根据用户的输入数据，帮助用户规划路线，解决了从起点到终点的路线问题，是一个用计算机来解决生活中问题的实例。

0-5 计算机求解步骤

使用计算机来解题，其步骤大致可以分为：问题分析、解题方法设计、解题实现、测试与修正这几步，如图 0-7 所示。

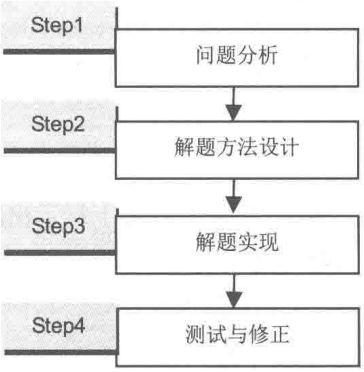

图 0-7

0-5-1 问题分析

用计算机求解之前，需要先对问题进行分析，问题分析是一种思考过程，我们对于要解决的问题，需要从不同的角度来进行思考，确定出问题的定义及范畴。在问题分析的过程中，需要先理清楚“输入规范”“输出规范”及“输入与输出对应关系”等要素。

- 输入规范（Specifications of Input）：对于用户输入的数据类型与范围，需要有一些规范，包括输入的数据内容、数据格式、数据范围等。例如在输入身高的程序中，就要限制用户不能输入负值，因为没有人的身高是负值。
- 输出规范（Specifications of Output）：对于输出的数据，也要做一些规范，因为计算机处理后的输出结果，需要满足用户的使用需求。例如在输出成绩的程序中，我们会希望当成绩超过 60 分时，程序会输出“及格”字样。
- 输入与输出对应关系：在问题分析阶段，应该理清输入与输出数据之间的对应关系，也就是要清楚地了解用户需要输入何种数据，而计算机会输出何种结果。我们经常使用输入处理输出图（Input Process Output, IPO）来描述输入与输出的关系，如图 0-8 为华氏温度转为摄氏温度的 IPO 示意图，华氏温度为 Input，转换公式为 Process，摄氏温度为 Output。

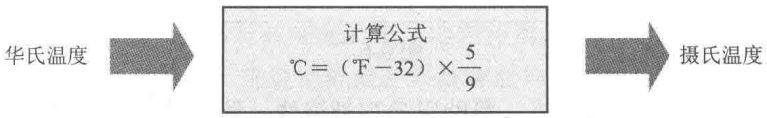

图 0-8

0-5-2 解题方法设计

完成了问题分析之后，接下来就要进行解题方法设计了，此步骤就是根据问题的需求，详细思考后得出的解决问题的步骤。在此阶段，可以试着多想几种方法来解决问题，试着从多种方法中，找出最好的解题策略。

解题方法的设计基本上有两种常用的思考策略，说明如下：

- 由上而下法（Top-Down）

  由上而下的解题设计方法，是将较大的问题，分解成许多能被处理的小问题，并通过处理这些小问题逐步解决整个问题，也就是先从整个问题的主要功能开始规划，然后再往下设计每个子问题的解题方法，直到最底层的小问题都解决为止。此处以学校行政系统设计为例，通常我们会先去思考整个学校的运作需求，然后考虑各个单位的工作需求，如教务处、学务处、总务处等处室的需求。

- 由下而上法（Bottom-Up）

  由下而上的解题设计方法，与由上而下法刚好相反，是先从小问题的功能开始规划，然后再往上设计比较大的功能，最后完成整个问题的解决。也就是说，由下而上的解题方法，会先从细部功能开始处理，然后根据问题的类别与特性，将相同属性的解题方法归纳在一起，由下而上，逐级完成整个问题的解决。在我们的日常生活中，也有许多由下而上的例子，最常见的就是比赛的赛程架构，由下而上，依次决出胜利的队伍，逐级比赛，得到最后的总冠军。

0-5-3 解题实现

在完成了解题方法设计之后，接下来我们需要选择编程语言来设计程序，把解题的方法变为解题的实现。

许多软件都可以用来实现计算机求解，甚至连 Excel 都可以。其实，不管是使用哪种编程语言或工具来进行计算机求解，其解题的设计方法是相同的，只是使用的编程语言或工具有所不同而已。

0-5-4 测试与改正

计算机求解的最后一个阶段，就是测试与修正阶段，也就是将所设计的解决问题的步骤，在完成解题之后，使用工具加以测试与修正，最后得到正确的解题策略与结果。测试与修正阶段是检查解题方法的最重要过程，主要目的是要确定解题方法是否可以真正解决问题。一般在测试时，会将各种情况下的数据都输入，尤其是一些边界条件值一定要加以测试，如极大值、极小值、负值、零值等。

0-6 计算思维体验

我们现在通过 Blockly Games 网站来体验一下计算思维与程序设计的乐趣，其网址为：

需要全本请访问 www.ertongbook.com