覆盖性质的特征理论
一般拓扑学论题之一

蒋继光

序　言

本书介绍基本覆盖性质的特征理论, 重心是仿紧空间和次亚紧空间的刻画. 所谓基本覆盖性质还包括亚紧性, 次仿紧性, 正规覆盖, 可缩性和強仿紧性等. 某类覆盖性质的特征或刻画 (characterizations) 是指与该性质的原始定义等价的命题. 这些特征命题一般比原始定义更弱. 自从 A.H.Stone [1948] 建立 T_2 仿紧性的重合定理以来，寻找仿紧空间及广义仿紧空间的各种各样的特征的努力一直延续至今. 发现不少美妙定理与精巧技术，形成系统理论. 本书试图叙述这一领域的主要成果，全部给予证明. 仿紧空间的刻画分别在节 6 与节 10 介绍，节 6 介绍不附加分离公理的仿紧性的刻画. 节 10 则介绍包含 T_2 仿紧空间为子类的 λ 完满正规空间的刻画. 作为它的另一子类, 正规 λ 强仿紧空间, 在节 11 中介绍. 节 4 介绍次亚紧空间的刻画. 这三节是特征理论的重心. 节 7 介绍正规覆盖的刻画，包含了点集拓扑学发展早期的一些好结果. 节 8 介绍集体正规空间与可缩空间的刻画. 这两节的内容是基本的, 不仅有其自身的意义，也是节 10 中定理证明需要引用的. 节 9 与节 5 分别介绍次仿紧与亚仿紧空间的刻画. 每种覆盖性质有一类型刻画，我们称之为分解定理, 即这种刻画表现为比它较弱的两种拓扑性质之和 (或两类较弱空间类的交). 其中一类是可扩型空间，另一类我们称之为弱覆盖性质. 我们在节 2 与节 3 中分别介绍这些空间需要的知识. 其中也有值得学习和欣赏的好结果。

有关覆盖性质的特征理论的已有文献，我愿推荐: [Bur84], [Jun80], [Gao2008] 和 [Yas89].

本书假定读书了解点集拓扑学的基础知识, 如 [Eng77] 的前 5 章, 或 [Gao2008] 的前 6 章. 集合论只需了解基数与序数的一般性质.

本书的参考文献只限于本书所引用者. 我在此向每一位作者谨致敬意和谢忱. 最后, 我对部分作者被引用的工作 (他们引入的概念和建立的定理的) 做了一个索引. 按照每一位作者被引用论文的最早发表年份为序排列他们的姓名. 这个索引有助于了解特征理论的发展历史.

蒋继光

于成都四川大学竹林村

乙未暮春

目　录

序言

第一章　预备

本章有两个目的. 一是为编程作些数学铺垫, 是漫谈和热身, 算不上是在严格意义上的数学准备. 二是通过一些实际例子说明, 在与时俱进的今天, 编程即使在纯数学学习中也不是可有可无. 在这些实际例子中, 或者数学扮演着的不可缺少的角色, 或者数学需要不同的优美表达. 但所涉及的计算, 手工进行的话太繁重, 但用编程则是轻而易举的. 本章的中的一些数据的生成和图表的绘制是以后各章的编程例子. 但愿达成一个共识, 编程与作图作为一个工具, 会成为数学想象的翅膀.

读者可把本章先当着趣味数学阅读, 在以后有关章节中实际编程时, 再回到本章有关问题. 教师在实际讲课时, 也可先从第??部分 Python 编程开始, 在编程和作图中再结合例子介绍本章内容.

.1　万物皆数?

.1.1　一切从 0 开始

老子说: “道生一，一生二，二生三，三生万物”[†]. “万物皆数”[‡] 则是古希腊数学家毕达哥拉斯的宇宙观. 他们的观点反映了数在现实世界中的重要性. 数在数学中的地位更是不言而喻. 不过要把数说清楚也不是一件容易的事情. 考虑本书着眼的是数学编程和作图, 我们在这里仅适可而止地讨论.

公理集合论是数理逻辑不可缺少的分支之一, 它奠定了整个现代数学的基础. 在公理集合论的宇宙里, 可以认为万物皆数. 不过这个数可不是毕达哥拉斯认为的有理数, 而是公理集合论里讲的基数. 基数是自然数的无穷推广, 就像每个有限集合都可用一个月自然数表示其多少一样, 每个集合都有一个人基数衡量其大小.

那么, 究竟什么是自然数呢? 过去, 从小学到大学的教科书和专著都说, 自然数的集合 $\mathbb{N} = \{1, 2, 3, \dots\}$. 华罗庚的《数论导引》[HuaL] 和菲赫金哥尔茨的《微积分学教程》[FeiH1] 等如是说. 范德瓦尔登《代数学 I》[derWaerden] 里关于皮亚诺公设的原来形式的叙述也是这样的. 但是现在的说法是, 自然数从 0 开始,

$$\mathbb{N} = \{0, 1, 2, 3, \dots\}.$$

[†]《道德经》第四十二章: 道生一，一生二，二生三，三生万物. 万物负阴而抱阳，冲气以为和. 人之所恶，唯孤、寡、不穀，而王公以为称。故物或损之而益，或益之而损。人之所教，我亦教之。强梁者不得其死，吾将以为教父.

[‡]万物皆数的英文原意为 Everything is arranged according to number. 法国数学家 Mickaël Launay 的《Le grand roman des maths: De la préhistoire à nos jours》一书的中译本书名《万物皆数》则是意译.

国际标准化组织 (ISO) 的标准化文件 ISO 80000-2* 规定 0 是自然数. 较早版本 ISO 31-11 已经这样规定了. ISO 凭什么来掺和纯数学的概念? 其中一个原因是绝大部分计算机编程语言†里, 列表 (`list`)或者组元 (`array`)之类的数据的索引是从 0 开始, 专业术语是"基于 0 的" (0-based)索引. 这在我们以后的 Python 编程中会看到. 其实更重要的原因还是纯数学界已经改口了. 所有关于现代公理集合论的著述都如是说[HalmosP][KelleyJ][JiangJi]. 各种级别的数学教材也纷纷把 0 纳入了自然数. 我们接下来会讨论整个纯数学是怎样无中生有, 从空集 ϕ, 也就 0 开始, 构造自然数. "道生一，一生二，二生三，三生万物"成了"0 生一，一生二，二生三，三生..."

人类对 0 的认识比其它对正整数的认识晚很多. 0 的出现起初不那么自然, 但现在为什么就很自然了呢? 除了为什么, 更重的问题是: 0、1、2、3 ... 究竟是什么东西? 读者可能会疑惑, 中国人从小学生起就熟背加法口诀和乘法九九表, 还问这样低级的问题? 事实上, 问题并不低级, 也不简单. 自然数 0、1、2、3 ... 在代数或集合论出现之前并不被视为具体东西, 而是一种抽象的比较关系. 数学上我们通常把俗称的东西叫做元素 (Element). 满足一些条件的元素又组成集合 (Set). 自然数本来是抽象概念, 展示事物共有的一些性质, 不是元素. 你的我的他/她的每只手和每只脚都有 5 个指头. 但是鸡不同, 它的每只爪子只有 4 个脚趾. 一只手/脚的 5 个指头不能和鸡爪的 4 个趾 1-1 对应, 但可以和五角星的 5 个角 1-1 对应. 像 5 一样, 一个个单独的自然数, 依靠这种 1-1 对应关系而被定义. 它常常被不严格地视为一个元素, 但是不能说它是一个集合. 它是一类有相同个数的有限集合的共同性质. 在朴素集合论里, 这种定义被推广到无穷集合上, 叫做势.

Ordinal number, Cardinal, Cardinal number 和 Cardinality 在朴素集合论‡ 和公理集合论里的含义是不同的. 在朴素集合论里, 它们是比较和等价关系, 是类型. 而在公理集合论里, 它们还是明确定义的集合. 为了不引起混淆, 在公理集合论里, 我们的说序数 (Ordinal number)和基数 (Cardinal number, 或 Cardinal), 而在朴素集合论里, 我们说势 (Cardinality).

康托 (Georg Cantor) 创立朴素集合论时就有等势的概念. 两个集合 A 和 B, 如果其元素有一个 1-1 对应关系, 就说 A 和 B 等势, 记为 $|A| = |B|$. 例如全体正整数的集合和全体正偶数的集合有相同的势, 朴素集合论里把这个势记为 $\aleph_0$. 虽然前者远多于后者, 但是 1-1 对应是很容易看出的. 这是因为, 乘以 2 可从前者映射到后者, 而除以 2 则可以从后者逆映射回到前者. 可以找到无穷多个不同集合, 具有势 $\aleph_0$. 比如平面上坐标为整数的所有点的集合, 也有势 $\aleph_0$. 任何势, 不管有穷还是无穷, 它是一类集合所共享的一种等价比较关系. 无穷势被视为一种无穷的数. 有限集合和无穷集合的区别在于, 任何一个无穷集合可以与它的真子集等势, 而有限集合则不能. 有无穷多个房间的 Hilbert 旅馆, 客满以后还可以继续接待无穷多的客人: 让已经住下的房客一个一个地依次挪房间就行了. 数学家们发现了无穷集合的许多跟直观一样感觉不一样的性质. 朴素集合论让阿猫阿狗进入集合通常也不是问题. 问题是, 谈一类集合的类 (class) 时限制太少, 集合和真类区分不清楚. 再就是早就有的自指 (self-reference) 逻辑困境以不同形式出现. 在"所有集合的集合"的罗素悖论中, 两种情形都有. 之前的关于势的康托悖论

*较早版本 ISO 31-11, 见 https://en.wikipedia.org/wiki/ISO_31-11

†统计学编程的 R 语言的索引不是基于 0 的.

‡有趣的是, Paul Halmos 的《朴素集合论》[HalmosP] 讲的是 ZFC 公理集合论.

和关于朴素集合论序数的 Burali-Forti 悖论则是起因于集合和真类区分不清楚.

在公理集合论里, 数学家们小心翼翼, 完全形式化、公理化地推理, 尽量杜绝一切逻辑漏洞. 别说让阿猫阿狗进入集合, 连什么是集合, 什么是元素, 什么是 $\in$(属于), 都坚称没有定义. 跟大家在日常谈论中的理解, 甚至跟朴素集合论的语境完全不一样, 集合的存在在严格的逻辑意义上是不可证明的. 如果可以证明的话, 何需集合的存在性公理?

每个集合都有一个基数*. 基数是特殊的序数 (Ordinal). † 最简单的序数和基数是自然数.

$0 = \phi, 1 = \{\phi\}, 2 = \{\phi, \{\phi\}\} \dots$ 如果把集合比为一个无形的盒子的话, 0 (空集 ϕ) 就是一个空盒子, 1 是套着一个空盒子的盒子, 2 是一个装着一个空盒子和一个套着空盒子的盒子的盒子, ...与在通常数学里用 $\mathbb{N}$ 表示自然数的集合不同, 公理集合论里一般用记号

$$\omega = \{0, 1, 2, 3, \dots\}.$$

.1.2 有理数与无理数都不能局限于有限

$365 = 10^2 + 11^2 + 12^2$. $365 = 1^3 + 2^3 + 3^3 + 4^3$.

1 inch = 96px https://zh.wikipedia.org/zh-cn/像素

像素，为影像显示的基本单位，译自英文“pixel”，pix 是英语单词 picture 的常用简写，加上英语单词“元素”element，就得到 pixel，故“像素”表示“画像元素”之意，有时亦被称为 pel（picture element）。每个这样的消息元素不是一个点或者一个方块，而是一个抽象的取样。仔细处理的话，一幅影像中的像素可以在任何尺度上看起来都不像分离的点或者方块；但是在很多情况下，它们采用点或者方块显示。每个像素可有各自的颜色值，可采三原色显示，因而又分成红、绿、蓝三种子像素（RGB 色域），或者青、品红、黄和黑（CMYK 色域，印刷行业以及打印机中常见）。照片是一个个取样点的集合，在影像没有经过不正确的/有损的压缩或相机镜头合适的前提下，单位面积内的像素越多代表分辨率越高，所显示的影像就会接近于真实物体。

《庄子 · 杂篇 · 天下》一尺之棰，日取其半，万世不竭。

一尺长的木棍，每天截掉一半，永远也截不完。这是庄子的好朋友、名家人物惠施的命题之一。惠施本人没有留下著作，《庄子 天下》保存了他的“历物十事”和二十一个命题。这个命题的意思是：一尺长的木棍，每天截去它的一半，千秋万代也截不完。

.2 数的进位制

《易经》有二进制的萌芽, 但不是真正的二进制计数和计算. 八卦卦符

- 坤：阴阴阴
- 艮：阴阴阳
- 坎：阴阳阴

*公理集合论里没有真正的元素, 因为每个元素本身就是一个集合, (除空集外) 它又含元素.

†根据选择公理, 每个集合有一个良序, 从而可找到一个与之有 1-1

- 巽：阴阳阳
- 震：阳阴阴
- 离：阳阴阳
- 兑：阳阳阴
- 乾：阳阳阳

例 .2.1. *1/4 属于 Cantor 集.*

我们来计算一下 1/4 的三进位表示.

$$\begin{aligned}
\frac{1}{4} &= \frac{1}{1+3} = \frac{1}{3(1+\frac{1}{3})} = \frac{1}{3}\cdot\frac{1}{1-(-\frac{1}{3})} \\
&= \frac{1}{3}\cdot\left[1+\left(-\frac{1}{3}\right)+\left(-\frac{1}{3}\right)^2+\left(-\frac{1}{3}\right)^3+\left(-\frac{1}{3}\right)^4+\left(-\frac{1}{3}\right)^5+\cdots\right] \\
&= \frac{1}{3}\cdot\left\{\left[1-\left(\frac{1}{3}\right)\right]+\left[\left(\frac{1}{3}\right)^2-\left(\frac{1}{3}\right)^3\right]+\left[\left(\frac{1}{3}\right)^4-\left(\frac{1}{3}\right)^5\right]+\cdots\right\} \\
&= \frac{1}{3}\cdot\left[\frac{2}{3}+\frac{2}{3^3}+\frac{2}{3^5}+\cdots\right] \\
&= \frac{2}{3^2}+\frac{2}{3^4}+\frac{2}{3^6}+\cdots
\end{aligned}$$

它的三进位表示里没有任何位数是 1, 由此可见 1/4 属于 Cantor 集.

习 题

1. 1/5 属于 Cantor 集.（提示：$1/5=\sum_{n=0}^{\infty}\frac{2^{2n}}{3^{2n+2}}$.）

.3 偏爱有理数

在纯数学中, 我们对有理数的分数形式有特别的偏爱. 毕竟, 有理数是自然数.

.4 尺寸

我们要作图，或为写文章写书排版，了解纸张的尺寸自然是有用的。国际标准 ISO 216 的纸张尺寸可分为 A、B、和 C 三种系列，其制定都跟数学有关。在中国和欧洲，打印文件纸张的比较常用标准尺寸是 A 系列中的 A4，即，210 毫米 x 297 毫米 (≈ 8.27 英寸 x 11.69 英寸)。德国科学家利希滕贝格 (Georg Christoph Lichtenberg) 发现高宽比为 $\sqrt{2}$ 的矩形具有有趣的特点。设一张矩形的纸宽为 w, 高为 h，竖着看我们假设 $w<h$。横着在高度一半处切裁为两个全等的更小矩形。小矩形不管是顺时针还是逆时针旋转 90°，竖起来后设新的宽为 w',

高为 h'。则 $w' = h/2$ 且 $h' = w$。若使原矩形与新的小矩形相似，则两个高宽比须相等:

$$\frac{h}{w} = \frac{w}{h/2}.$$

于是，$h^2 = 2w^2$, 或

$$h = \sqrt{2}w.$$

这里我们又遇到了一个无理数, $\sqrt{2}$。在工程上，我们较真的程度有赖于技术。

习题. 执行如下 Python 代码，确认记忆中的 $\sqrt{2}$ 值 $1.4142135623730951\cdots$。

```
print("2␣的平方根␣=␣", 2**.5)
```

A 系列不同开本的纸张从一平方米 (1,000,000 平方毫米) 的 A0 开始。设宽为 w_0 毫米，则高为 $\sqrt{2}w_0$ 毫米。于是 $\sqrt{2}w_0^2 = 1000000$。由此，$w_0 = \sqrt{1000000/\sqrt{2}} = 1000/\sqrt[4]{2}, h_0 = \sqrt{2}w_0 = 1000 \cdot \sqrt[4]{2}$。几行 Python 代码可算出 A 系列不同开本纸张的尺寸。注意裁开一次页数加倍，所以 An 是 2^n 开，A4 是 16 开 ($2^4 = 16$)。

```
#%% A系列纸张的理论尺寸
w = 1_000 / 2**.25 # 下划线"_"分隔仅为阅读方便，无实际编程作用
h = 1_000 * 2**.25 # 2**.25=2^.25是2的4次方根即2的0.25次幂
print('A系列纸张的理论尺寸(mm␣x␣mm):')
for i in range(0, 11):
  # {:>3} 表示3个字符向右对齐。宽高四舍五入到整数。
  # {:3.0f} 浮点数只显示3位整数，自动向右对齐。
  # {:<4.0f} 浮点数只显示4位整数，向左对齐。
  print("{:>3}:␣{:3.0f}␣x␣{:<4.0f}".format(
    'A'+str(i), w, h), end='␣␣␣')
  # {:6.2f}: 包括小数点，浮点数展示6个字符向右对齐，小数点后面保留2位数。
  # {:<7.2f}: 包括小数点，浮点数展示7个字符向左对齐，小数点后面保留2位数。
  print("␣{:6.2f}␣x␣{:7.2f}".format(w, h), end='␣␣')
  # 用 int() 舍弃小数部分，宽高仅保留整数部分。
  print("␣␣␣{:3.0f}␣x␣{:4.0f}".format(int(w), int(h)))
  tmp = w  #暂存原宽度备作下一个高度。
  w = h/2   #新宽度为原高度的一半。
  h = tmp   #新高度为原宽度。
```

输出结果如下:

```
A 系列纸张的理论尺寸 (mm x mm):
 A0:  841 x 1189   840.90 x 1189.21   840 x 1189
 A1:  595 x 841    594.60 x 840.90    594 x 840
 A2:  420 x 595    420.45 x 594.60    420 x 594
 A3:  297 x 420    297.30 x 420.45    297 x 420
 A4:  210 x 297    210.22 x 297.30    210 x 297
 ...
```

国际标准 ISO 216 中毫米仅保留到整数。第一列结果四舍五入显然不符合切裁纸张的实际。第三列用 `int()` 丢弃小数部分，宽高仅保留整数部分，所以应该最符合实际。事实上，除了 ISO 216 A1 的尺寸是 594 毫米 `x` 841 毫米而不同外，第三列的其它都符合国际标准。

练习. 在上面的 Python 代码中的，把 `for` 循环句改为 `while` 循环句，并且使输出加一列说明开数。

B 系列纸张仍然遵循高宽比约为 $\sqrt{2}$ 的原则，但定义 B0 的宽为 1000 毫米 (1 米)，高为 1414 毫米.

练习. 写出 Python 程序，计算 B0 到 B10 的开本和理论尺寸。

采用国际标准中不同尺寸的纸张，由于高宽比例固定为 $\sqrt{2}$，除了切裁加开在工程意义上不改变高宽比例外, 不同系列和开本的文件可以直接缩放影印而不会造成纸面图案有边缘裁切的问题。同系列纸张的缩放倍数更是 2 的幂.

中国书籍，查看菲页背后有关印刷信息的一页，一般会见到纸张尺寸的字样. 比如：开本 787x1092 1/32, 850x1168 1/32, ... 787x1092 是 787 毫米 x 1092 毫米 (约 31 英寸 x 43 英寸) 的正度纸, 高宽比例约为 1.37. 850x1168 是 850 毫米 x 1168 毫米的大度纸, 高宽比例约为 1.39. 1/32 则是整幅纸裁为 32 张. 它们跟国际标准不一致, 大概会被逐步淘汰.

美国纸张尺寸标准也不按 $\sqrt{2}$ 的高宽比例。常用标准信纸的尺寸是 8.5 英寸 x 11 英寸 (≈ 215.9 毫米 x 279.4 毫米) 。法律用纸尺寸为 8.5 英寸 x 14 英寸 (≈ 215.9 毫米 x 355.6 毫米)。美国是为数不多的不采用公制而采用英制的国家。尤为麻烦的是，计算机网络和软件方面，仍然时不时要同英寸打交道：1 英寸 = 2.54 厘米，或者 $1'' = 2.54$ cm.

习 题

1. 在仍然常见的书籍印刷 787x1092 毫米和 850x1168 毫米纸张系列中, 16 开本每页的理论尺寸是多少? 32 开本每页的理论尺寸是多少? 找本所说规格的书测量一下, 验证你的答案. 如果答案不符, 为什么?

.5 汉字的数字化

.5.1 BG 码与统一码 Unicode

.5.2 utf-8

π 的

赵钱孙李 周吴郑王 冯陈褚 (ch) 卫 蒋沈韩杨朱秦尤许 何吕施张 孔曹严华 金魏陶姜

《百家姓》各个姓氏排列次序不是以人口数量多少, 而是以政治地位为准则. 根据南宋学者王明清考证,"赵钱孙李"之所以是《百家姓》前四姓, 是因为百家姓在北宋初年的吴越钱塘地区形成, 所以就用当时最重要的家庭姓氏：赵氏：宋朝皇帝的姓氏. 钱氏：吴越国王的姓氏. 孙氏：吴越国王钱俶正妃的姓氏. 李氏：南唐国王的姓氏.

pi = 0x3c0 11 1100 0000 b xcf x80

.6 颜色

.6.1 彩虹、日光和调色板

世界是多姿多彩的, 而姿需图, 彩即色. 先说色. “五颜六色”是一个汉语成语, 形容色彩复杂或花样繁多, 但数字 5 和 6 并无确切的含义.“五彩缤纷、万紫千红”里面的 1,000 和 10,000 也如此, 五彩则说法各异, 有联系上方位的, 也有联系上五行的. 但色彩需要数学来准确地描述. 古人以最突出的七种颜色的合称“七彩”, 则是确有所指, 来自彩虹. 采用传统中国文化说法, 毛泽东把它们写进了诗: “赤橙黄绿青蓝紫, 谁持彩练当空舞?”*赤就是红, 而青色 (Cyan) 就是偏蓝的绿色. 在不同的场合见到的彩虹色彩会不同，而且色谱是连续变化的. 色谱段的一种取舍可以使主要颜色符合这个顺序. 现在更被采纳的彩虹主要颜色顺序是红橙黄绿蓝靛紫, 牛顿是把这个当作日光光谱主要颜色的顺序. 他先把日光光谱的主要颜色分为五种: 红黄绿蓝紫, 后因相信古希腊人对数字 7 的推崇，把橙色和靛色加了进去. 日光通过棱镜折出来的光谱与彩虹光谱有所不同, 通过对日光的现代光谱分析, 把赤橙黄绿青蓝紫作为日光光谱主要颜色顺序更为恰当. 经编程计算, 列出这两种顺序如 .6.1.

图 .6.1: “红橙黄绿蓝靛紫”与“赤橙黄绿青蓝紫”

根据物理学, 光谱中的颜色是和光波的波长 (1 THz = 1 太赫兹 = 10^{12} 赫兹) 和频率 (1 nm = 1 纳米 = 10^{-9} 米) 相对应的.

颜色	频率	波长
紫色	668–789 THz	380–450 nm
蓝色	631–668 THz	450–475 nm
青色	606–630 THz	476–495 nm
绿色	526–606 THz	495–570 nm
黄色	508–526 THz	570–590 nm
橙色	484–508 THz	590–620 nm
红色	400–484 THz	620–750 nm

HTML 的标签 <input type="color"> 可为网页搭建一个调色板. 不同的浏览器实施各不相同. 苹果浏览器 Safari 展示的调色板如 .6.2, 其首选正是“赤橙黄绿青蓝紫”, 紫前加上洋红 (magenta), 紫后加上棕白灰黑.

SVG 1.1 中定义了 148 种颜色的英文名†, 后来也包括在了 CSS ‡ 4 中 §. 所有互联网浏览

* 《菩萨蛮 · 大柏地》

† `https://www.w3.org/TR/SVG11/types.html`

‡ Cascade Style Sheet: 在 HTML 中用来添加样式的语言.

§ `https://www.w3.org/TR/css-color-4/`

图 .6.2: 苹果浏览器 Safari 的 HTML 调色板

器都支持 HTML/CSS 使用其中 141 种颜色的英文名. 用 Python 编程 SVG 生成的这些颜色如图.6.4. 但对比实际生成的颜色发现, 个别用英文名生成的颜色有偏差, 如 FireBrick (耐火红砖) 和 VioletRed (紫红). 另外也应注意 Green (绿色 #008000) 和 Lime (鲜绿色 #00FF00) 的确切 RGB 定义. RGB 是我们接着要讨论的.

最常用的则是 HTML 4.01 最初支持的 16 种基本颜色, 如图 **??**.

http://www.w3.org/TR/html5/

.7 美离不开图

感人的歌声留给人的记忆是长远的。

- 摘自中学语文课文

《周髀算经》勾股定理图

勾股定理 - 北京 2002 年国际数学家大会会标中的图案

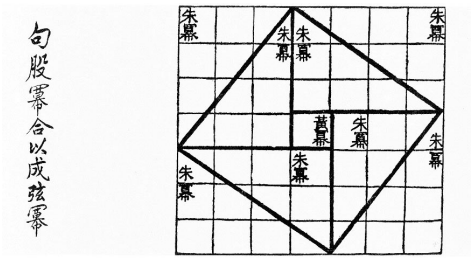
对每个正整数 n, 在抛物线 $y = x^2$ 的杯子中间作半径为 n 的圆与之相切. 则根据中学的解析几何进行计算, 切点为 $(\sqrt{n^2 - 1/4}, \pm(n^2 - 1/4))$, 圆心为 $(0\ n^2 + 1/4)$. 各圆也正好相切. 于是, 在旋转抛物面 $z = x^2 + y^2$ 中从公元元年开始, 每个新的公元 n 年可放入一个半径为 n 的球体, 使它们完美地与曲面互切并与上年的球体刚好相切. 冷饭可以年年热炒恭贺新年.

AliceBlue 爱丽丝蓝 #F0F8FF	AntiqueWhite 古董白 #FAEBD7	Aqua 水色 #AFDFE4
Aquamarine 碧蓝色 #7FFFD4	Azure 湛蓝色 #F0FFFF	Beige 米黄色 #F5F5DC
Bisque 陶坯黄 #FFE4C4	Black 黑色 #000000	BlanchedAlmond 杏仁白 #FFEBCD
Blue 蓝色 #0000FF	BlueViolet 蓝紫 #8A2BE2	Brown 褐色 #A52A2A
BurlyWood 硬木色 #DEB887	CadetBlue 军服蓝 #5F9EA0	Chartreuse 查特酒绿 #7FFF00
Chocolate 巧克力色 #D2691E	Coral 珊瑚红 #FF7F50	CornflowerBlue 矢车菊蓝 #6495ED
Cornsilk 玉米丝色 #FFF8DC	Crimson 绯红 #DC143C	Cyan 青色 #00FFFF
DarkBlue 暗蓝 #00008B	DarkCyan 暗青 #008B8B	DarkGoldenRod 暗金菊色 #B8860B
DarkGray 暗灰 #A9A9A9	DarkGreen 暗绿 #006400	DarkKhaki 暗卡其色 #BDB76B
DarkMagenta 暗洋红 #8B008B	DarkOliveGreen 暗橄榄绿 #556B2F	Darkorange 暗橙 #FF8C00
DarkOrchid 暗兰紫 #9932CC	DarkRed 暗红 #8B0000	DarkSalmon 暗鲑红 #E9967A
DarkSeaGreen 暗海绿 #8FBC8F	DarkSlateBlue 暗岩蓝 #483D8B	DarkSlateGray 暗岩灰 #2F4F4F
DarkTurquoise 暗绿松石色 #00CED1	DarkViolet 暗紫 #9400D3	DeepPink 深粉红 #FF1493
DeepSkyBlue 深天蓝 #00BFFF	DimGray 昏灰 #696969	DodgerBlue 道奇蓝 #1E90FF
FireBrick 耐火砖红 #B22222	FloralWhite 花卉白 #FFFAF0	ForestGreen 森林绿 #228B22
Fuchsia 品红 #F400A1	Gainsboro 庚斯博罗灰 #DCDCDC	GhostWhite 幽灵白 #F8F8FF
Golden 金色 #FFD700	GoldenRod 金菊色 #DAA520	Gray 灰色 #808080
Green 绿色 #008000	GreenYellow 绿黄 #ADFF2F	HoneyDew 蜜瓜绿 #F0FFF0
HotPink 暖粉红 #FF69B4	IndianRed 印度红 #CD5C5C	Indigo 靛色 #4B0082
Ivory 象牙色 #FFFFF0	Khaki 卡其色 #F0E68C	Lavender 薰衣草紫 #E6E6FA
LavenderBlush 薰衣草紫红 #FFF0F5	LawnGreen 草坪绿 #7CFC00	LemonChiffon 柠檬绸色 #FFFACD
LightBlue 亮蓝 #ADD8E6	LightCoral 亮珊瑚色 #F08080	LightCyan 亮青色 #E0FFFF
LightGoldenRodYellow 亮秋黄 #FAFAD2	LightGrey 亮灰色 #D3D3D3	LightGreen 亮绿 #90EE90
LightPink 浅粉红 #FFB6C1	LightSalmon 浅鲑红 #FFA07A	LightSeaGreen 亮海绿 #20B2AA
LightSkyBlue 亮天蓝 #87CEFA	LightSlateGray 淡岩灰 #778899	LightSteelBlue 亮钢蓝 #B0C4DE
LightYellow 亮黄 #FFFFE0	Lime 鲜绿色 #00FF00	LimeGreen 酸橙绿 #32CD32
Linen 亚麻色 #FAF0E6	Magenta 洋红 #FF00FF	Maroon 栗色 #800000
MediumAquaMarine 中碧 #66CDAA	MediumBlue 中蓝色 #0000CD	MediumOrchid 中兰紫 #BA55D3
MediumPurple 中紫红 #9370D8	MediumSeaGreen 中海绿 #3CB371	MediumSlateBlue 中岩蓝 #7B68EE
MediumSpringGreen 中春绿 #00FA9A	MediumTurquoise 中绿松 #48D1CC	MediumVioletRed 中紫罗兰色 #C71585
MidnightBlue 午夜蓝 #191970	MintCream 薄荷奶油色 #F5FFFA	MistyRose 雾玫瑰色 #FFE4E1
Moccasin 鹿皮鞋色 #FFE4B5	NavajoWhite 那瓦霍白 #FFDEAD	Navy 藏青 #000080
OldLace 旧蕾丝色 #FDF5E6	Olive 橄榄色 #808000	OliveDrab 橄榄军服绿 #6B8E23
Orange 橙色 #FFA500	OrangeRed 橙红色 #FF4500	Orchid 兰紫 #DA70D6
PaleGoldenRod 灰金菊色 #EEE8AA	PaleGreen 灰绿 #98FB98	PaleTurquoise 灰绿松石色 #AFEEEE
PaleVioletRed 灰紫红 #D87093	PapayaWhip 蕃木瓜色 #FFEFD5	PeachPuff 桃肉色 #FFDAB9
Peru 秘鲁色 #CD853F	Pink 粉红 #FFC0CB	Plum 梅红色 #DDA0DD
PowderBlue 粉蓝色 #B0E0E6	Purple 紫色 #800080	Red 红色 #FF0000
RosyBrown 玫瑰褐 #BC8F8F	RoyalBlue 品蓝 #4169E1	SaddleBrown 鞍褐 #8B4513
Salmon 鲑红 #FA8072	SandyBrown 沙褐色 #F4A460	SeaGreen 海藻绿 #2E8B57
SeaShell 海贝色 #FFF5EE	Sienna 赭黄 #A0522D	Silver 银色 #C0C0C0
SkyBlue 天蓝 #87CEEB	SlateBlue 岩蓝 #6A5ACD	SlateGray 岩灰 #708090
Snow 雪色 #FFFAFA	SpringGreen 春绿 #00FF7F	SteelBlue 钢蓝色 #4682B4
Tan 日晒色 #D2B48C	Teal 凫绿 #008080	Thistle 蓟紫 #D8BFD8
Tomato 蕃茄红 #FF6347	Turquoise 绿松石色 #40E0D0	Violet 紫罗兰色 #EE82EE
VioletRed 紫红 #D02090	Wheat 小麦色 #F5DEB3	White 白色 #FFFFFF
WhiteSmoke 白烟色 #F5F5F5	Yellow 黄色 #FFFF00	YellowGreen 黄绿色 #9ACD32

图 .6.3: HTML/SVG 支持的 141 种颜色

Aqua 水色 #AFDFE4	Black 黑色 #000000	Blue 蓝色 #0000FF	Fuchsia 品红 #F400A1
Gray 灰色 #808080	Green 绿色 #008000	Lime 鲜绿色 #00FF00	Maroon 栗色 #800000
Navy 藏青 #000080	Olive 橄榄色 #808000	Purple 紫色 #800080	Red 红色 #FF0000
Silver 银色 #C0C0C0	Teal 鸢绿 #008080	White 白色 #FFFFFF	Yellow 黄色 #FFFF00

图 .6.4: HTML 的 16 种基本颜色



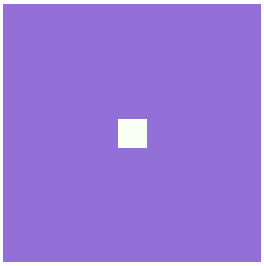


图 .7.1: 勾股定理

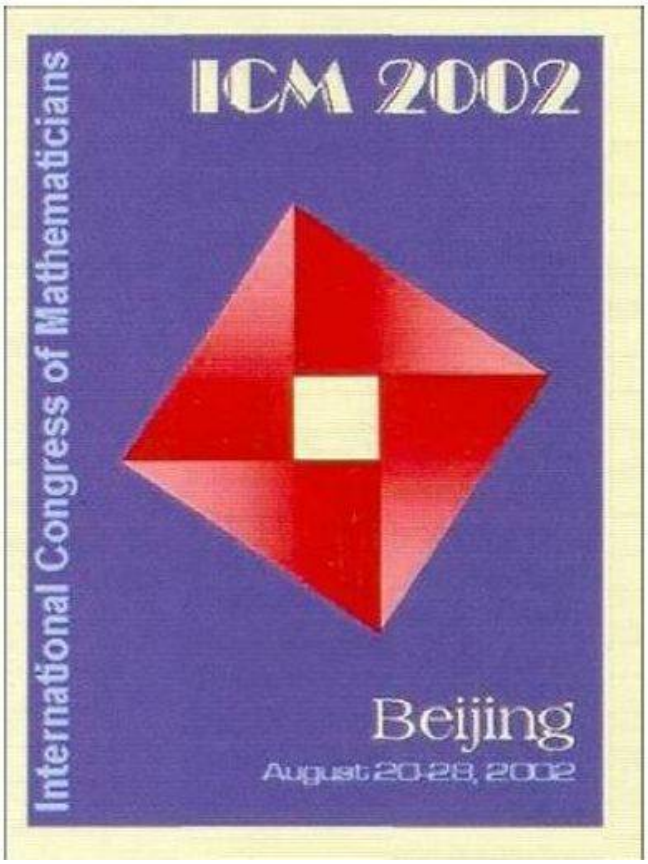


图 .7.2: 北京 2002 年国际数学家大会会徽

第二章　Python 入门

.1　关于 Python

不犯错者一事无成.
Не ошибается тот, кто ничего не лелает.
俄国彦语

Python 语言的最初设计者和主要架构师吉多 · 范罗苏姆 (Guido van Rossum)，荷兰人，1982 年在阿姆斯特丹大学获得数学和计算机科学硕士学位。据说 1989 年的圣诞节期间，为了打发时间而开发一个新的脚本解释器(Script Interpreter)，以替代使用 Unix shell 和 C 语言进行系统管理。后来 Python 成了一种广泛使用的解释型、高级和通用的编程语言，支持多种编程范型，包括函数式、指令式、结构化、面向对象和反射式编程。它拥有动态类型系统和垃圾回收功能，能够自动管理内存使用，并且其本身拥有一个巨大而广泛的标准库。本节提到的名词术语, 如果初次接触的话, 可能有些不太好理解. 这里作些解释, 但是就本书的使用来说, 完全可以忽略, 而不影响我们做要做的事情.

解释型 (interpreting) 意味着，写好的、人可读代码直接一行一行地被执行. 而在 C/C++、Java 等语言里, 编译器 (Compiler)先把相关的代码文件编译 (compile) 成可执行的、机器可读而人不可读的二进制指令文件, 然后被执行或解释。

.2　安装 Python 开发环境软件集 Anaconda

Python 在 MacOS 和 Unix/Linux 操作系统是默认安装好的，但版本一般是 `2.x`，且没有图形用户界面，只适合在终端命令行上进行交互运行或脚本运行。微软视窗没有默认安装 Python。

本书采用 Python `3.x`。虽然可访问 `https://www.python.org/downloads/`安装 IDLE (Integrated Development and Learning Environment, 集成开发学习环境)，但是这个安装可省去. 这个 IDLE 是终端命令行式的, 无图形用户界面. 而本书将在有图形用户界面的 IDE (Integrated Development Environment, 集成开发环境) 里操作。

Python 有好几种免费的，且功能卓越的 IDE 软件，其中 Spyder 和 Jupyter 都包括在内 Anaconda. Spyder 提供一个优秀的开发平台和用户界面，进行 Python 的代码编辑、交互运行和图形绘制。Jupyter 也提供这样的开发平台，但它以互联网浏览器为用户界面，背后

由一个本地网络服务器引擎在支持。其它很受欢迎并且为软件开发人员广泛使用的免费 IDE 有 Microsoft Visual Studio Code (https://code.visualstudio.com) 和 IntelliJ (Community 版本) (https://www.jetbrains.com). 这两种 IDE 支持除了 Python 以外的其它语言，如 Java，JavaScript，Node.js 和 C++ 等，但 Python 以及它的一些常用包和库需要另行安装和配置. Jetbrain 开发的 Intellij 和 Pycharm 都非常好. 但 Intellij Ultimate 版和 PyCharm Professional 版都需要付费购买许可证, 而免费的 Intellij Community 版和 PyCharm Community 版不支持本书要使用的一些 Python 库. 本书主要使用 Spyder, 为开拓眼见我们也使用 Jupyter.

若只安装 Spyder，可访问 `https://www.spyder-ide.org` 。

若只安装 Jupyter，可访问 `https://www.Jupyter.org` 。

我们建议安装 Anaconda Navigator，以便可同时使用 Spyder 和 Jupyter 及其它。Anaconda 的网址是:

`https://www.anaconda.com` 。

需要注意的是，Anaconda 个人版本（Individual Edition）是免费的，而其它版本则不是。个人版本从

`https://www.anaconda.com/products/individual` 。

下载安装。安装好后，打开 Anaconda Navigator 所见如图 .2.1。

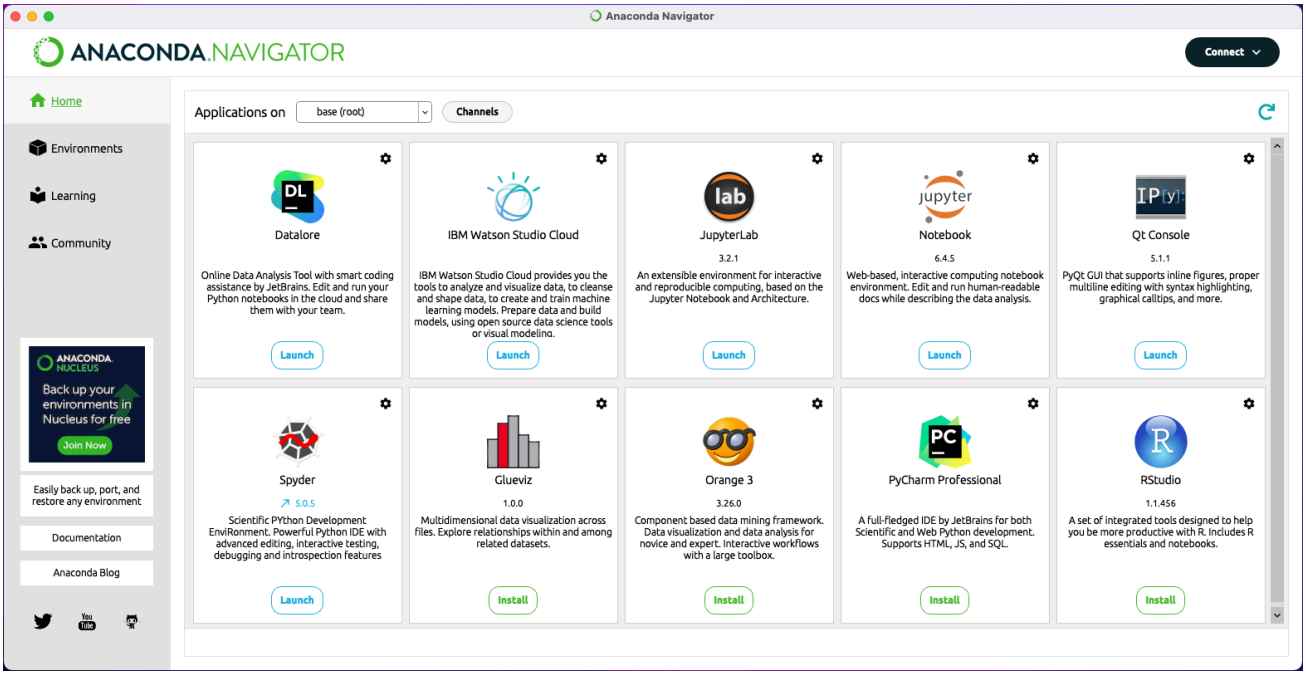


图 .2.1: Anaconda Navigator

```
conda install -c conda-forge imagemagick
conda install -c conda-forge jieba
conda install -c conda-forge wordcloud
```

.3 初识 Spyder

然后单击 Spyder 下面的 Launch 就会打开 Spyder。单击 Jupyter 下面的 Launch 就会打开 Spyder 打开 Spyder

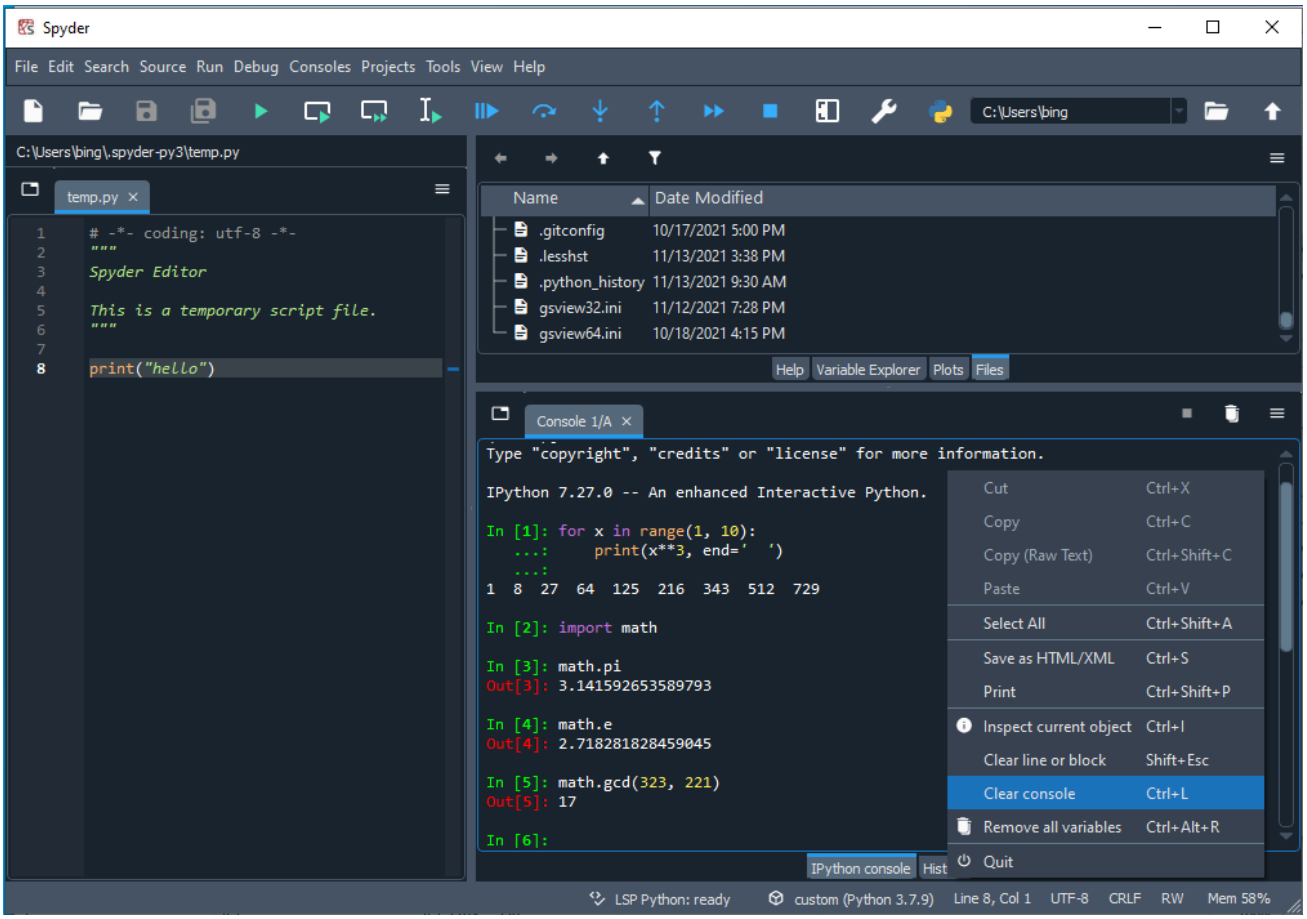


图 .3.1: Spyder IDE

the Specifying Colors tutorial; the matplotlib.colors API; the Color Demo.

```
Text enclosed 'inside' \texttt{verbatim} environment
is printed directly
and all \LaTeX{} commands are ignored.下面的
```

```
import re

x = "100以内有25个质数。"
x += "不实际计算的话,51和91很容易被误判为是素数。"
y = re.sub("质数", "素数", x)
z = x.replace("素数", "质数")
print(y)
print(z)
print(x)

#%%
import math
```

```
13
14 print("hello")
15 print(re.sub("This", "That", "This is not"))
16 print(math.pi)
```

.4 初识 Jupyter

然后单击 Spyder 下面的 Launch 就会打开 Spyder。单击 Jupyter 下面的 Launch 就会打开 Spyder 打开 Spyder

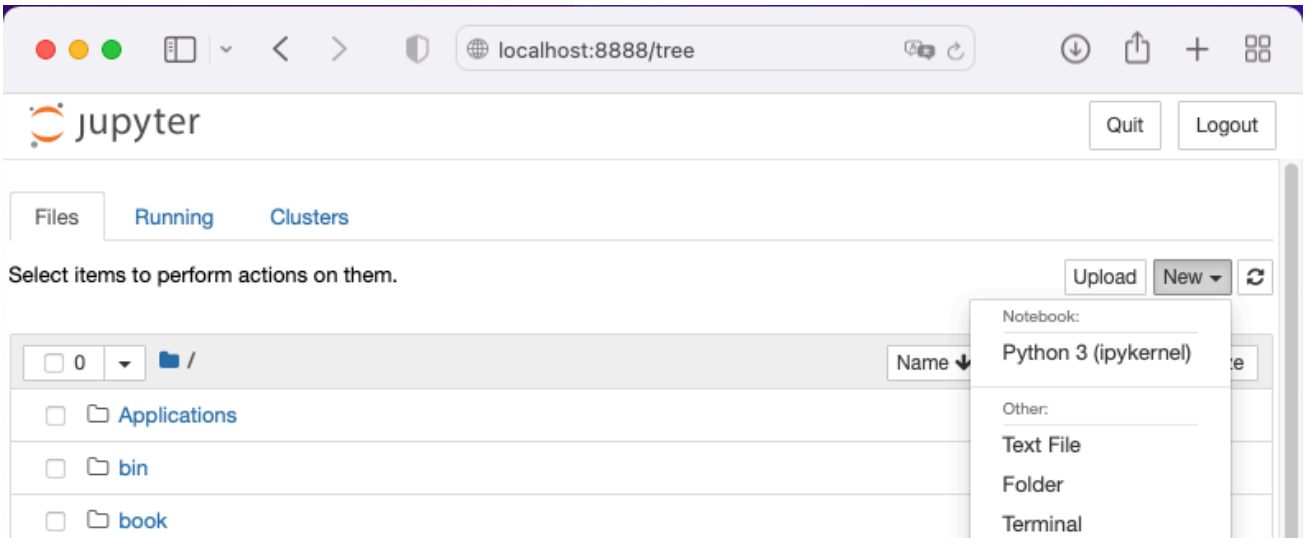


图 .4.1: Jupyter 网页界面

the Specifying Colors tutorial; the matplotlib.colors API; the Color Demo.

.5 常用进位制

处理字的符的 Python 程序

Python 程序 xxx 生成希腊的字母如下:

从 Unicode 表中 Python 程序 xxx 生成部分常用数学运算速度符号如果下:

字符串处理

习 题

1. 把下列值代入函数 `int()` 后的结果是什么?

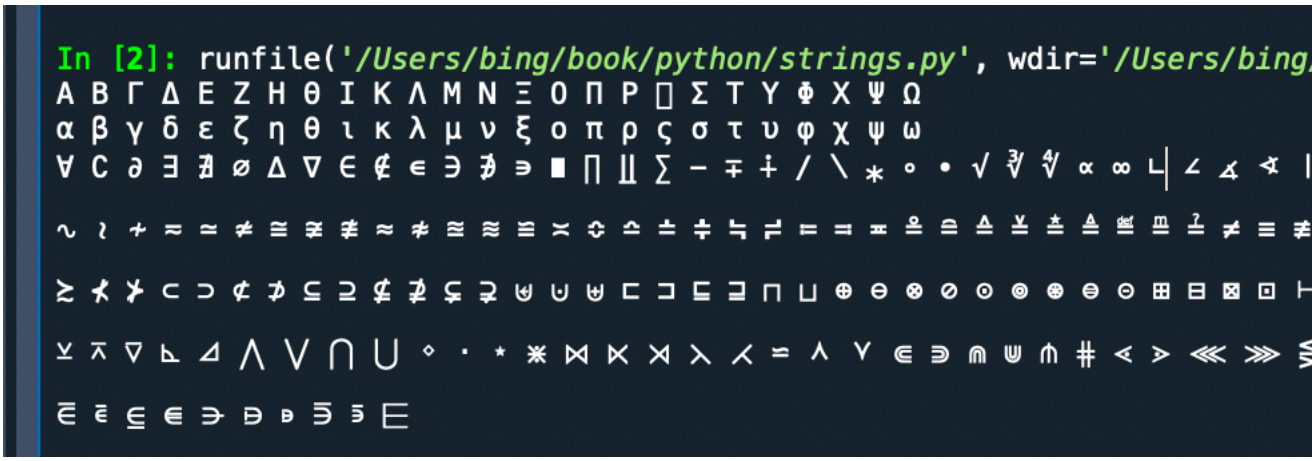
(a) 4.5; (b) 4.4; (c) -4.5; (d) -4.4;

(e) `True`; (f) `False`; (g) `true`; (h) `false`;

(i) 5 == 1+4; (j) 6 < 5; (k) 6 > -1.2; (l) '3.14159';

2. 把下列各对值代入函数 `int()` 后的结果是什么?

```
#### 生成一组 Unicode 值在 m 和 n（包括）之间的字符列表。
##   输入：Unicode 起始值 m 和终止值（包括）n。
##   输出：所定范围内相关应字符的列表。
#### Unicode 的各种字符表可见 https://www.unicode.org/charts/
"""
## 用遍历方法生成的，不推荐。
def charactersFromTo(m, n):
    result = []
    for x in range(m, n+1):
        result.append(chr(x))
    return result
"""
def charactersFromTo(m, n):
    return list(map(lambda x: chr(x), list(range(m, n+1))))


#### 汉字表。
def chineseCharacters():
    return charactersFromTo(0x4E00, 0x9FFF)


#### 大写希腊字母表：0x0391 ~ 0x03A9。
def upperCaseGreekAlphabet():
    return charactersFromTo(0x0391, 0x03A9)


#### 小写希腊字母表：0x03B1 ~ 0x03C9。
def lowerCaseGreekAlphabet():
    return charactersFromTo(0x03B1, 0x03C9)


#### 大写俄语字母表：0x0410 ~ 0x042F。
def upperCaseCyrillicAlphabet():
    return charactersFromTo(0x0410, 0x042F)


#### 小写俄语字母表：0x0430 ~ 0x044F。
def lowerCaseCyrillicAlphabet():
    return charactersFromTo(0x0430, 0x044F)


#### 数学运费算符号表：0x2200 ~ 0x22FF。
def mathOperators():
    return charactersFromTo(0x2200, 0x22FF)


#print(chineseCharacters())
print(" ".join(upperCaseGreekAlphabet()))
print(" ".join(lowerCaseGreekAlphabet()))

print(upperCaseCyrillicAlphabet())
print(lowerCaseCyrillicAlphabet())
print(" ".join(mathOperators()))
print(mathOperators())
```



(a) 4.5; (b) 4.4; (c) -4.5; (d) -4.4;
(e) `True`; (f) `False`; (g) `true`; (h) `false`;
(i) 5 == 1+4; (j) 6 < 5; (k) 6 > -1.2; (l) '3.14159';

第三章　Python 应用

.1　Jieba

gb2312

import urllib.request import urllib.parse

url = 'http://xh.5156edu.com' f = urllib.request.urlopen(url) txt = f.read().decode('gb2312') print(txt)

GB 18030，全称《信息技术中文编码字符集》，是中华人民共和国国家标准所规定的变长多字节字符集。

1、jieba 库基本介绍 (1)、jieba 库概述 jieba 是优秀的中文分词第三方库 - 中文文本需要通过分词获得单个的词语 - jieba 是优秀的中文分词第三方库，需要额外安装 - jieba 库提供三种分词模式，最简单只需掌握一个函数 (2)、jieba 分词的原理 Jieba 分词依靠中文词库 - 利用一个中文词库，确定汉字之间的关联概率 - 汉字间概率大的组成词组，形成分词结果 - 除了分词，用户还可以添加自定义的词组 jieba 库使用说明 (1)、jieba 分词的三种模式精确模式、全模式、搜索引擎模式

精确模式：把文本精确的切分开，不存在冗余单词 - 全模式：把文本中所有可能的词语都扫描出来，有冗余 - 搜索引擎模式：在精确模式基础上，对长词再次切分 (2)、jieba 库常用函数

cut cutforsearch lcut addword

.2　Some graphs

Postscript Language Reference, 3rd Edition[PSLF3]

https://www.adobe.com/content/dam/acom/en/devnet/actionscript/articles/PLRM.pdf

.2.1　一些数学公式的几何解释

$(a+b)^2 = a^2 + 2ab + b^2$

例 .2.1. *n 边形的内角和是 $180(n-2)$ 度，所以正五边形的内角和是 $180 \cdot 3 = 540$ 度，而单个内角则是 $540/5 = 108$ 度。*

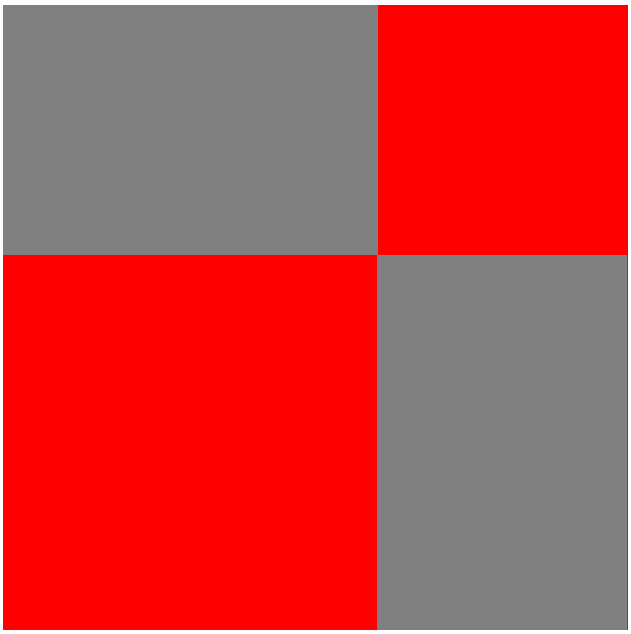


图 .2.1: $(a+b)^2 = a^2 + 2ab + b^2$

line spread

line spread dfdsa;fkadf

```
%!PS-Adobe-3.0
/side 80 def %%定义边长

/angle { %%定义五角星的一个尖角
 side neg 0 rlineto
 currentpoint translate
 -36 rotate
  side 0 rlineto } def

/star {
 newpath
 moveto
 angle  %%先定义一个尖角
 4 { currentpoint translate
  108 rotate angle
 } repeat %%然后旋转坐标系后作出其它四个角色
 closepath } def

gsave  %%保存原始坐标系统的信息
300 500 star  %%描出五星的轮廓路径
1 0 0 setrgbcolor  %%设置红色
fill %%填充绘出红五星，然后轮廓路径就消失
grestore

300 500 star %%重新描出五星的轮廓路径
0 setgray %%加黑边
stroke

showpage
```

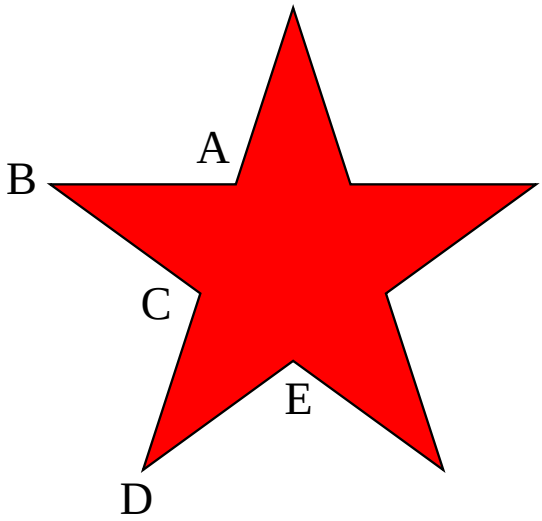


图 .2.2: 红五星

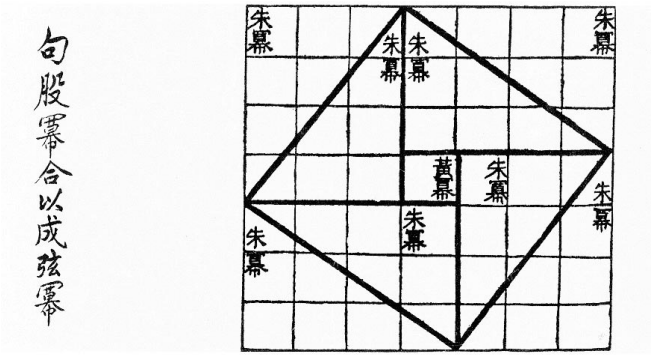


图 .2.3: 《周髀算经》勾股定理图

图 .2.4: $(a+b)^2 = a^2 + 2ab + b^2$

.3 A List

华罗庚数论导引[HuaL]
钟尔杰数学实验方法[ZhongEr1]
钟尔杰数值分析讲义[ZhongEr2]
Is God a Mathematician?[LivioM]
最后的数学问题[LivioM2]
菲赫金哥尔茨《微积分学教程》[FeiH1][FeiH2],[FeiH3]
Atiyah[AM69]
Naive Set Theory[HalmosP]
KelleyJ, General Topology[KelleyJ]

蒋继光, 一般拓扑学专题选讲[JiangJiguangYibantuopuxuezhuantixuanjiang]
[derWaerden]

名词索引