

数学：编程与作图
Python、PostScript 和 SVG 的浅入

龙 冰

目录

第一部分 数学：表达美和创造美	5
第一章 数学：表达美和创造美	6
I.1 常用进位制	10
第二部分 Python 程序设计	13
第二章 选择 Python 开发环境，安装软件	14
第三部分 PostScript 编程制图	15
III.1 Some graphs	16
III.2 A List	18

Review Copy

前言

上世纪笔者考入四川大学数学系七七级，1958 年设立的计算机专业还是数学系的一部分，分出去成立计算机系并壮大成计算机（及软件）学院是以后的事了。计算机程序设计是数学系的必修课，但由于各种条件的限制，课程的开设成了摆设，猜想大多数同学认为是浪费时间。我们学习的语言是 ALGOL 60，编程只能写在作业本上。学校有一台巨大的当时比较高级的计算机放在一大房间里精心维护着。工作人员穿着白大褂，地板打着蜡。可是我们平时没有机会接触。只是到了期末考试，才有机会进机房。开卷考试题是解二次方程 $ax^2 + bx + c = 0$ 。由专门操作人员在纸带上打孔，我们也不懂打孔，只听机器卡嚓卡嚓响，我们就都稀里糊涂地通过了。由于计算机技术的巨大进步，任何一台个人电脑的都会拥有比那台宝贝大不是知多少倍的计算能力。在个人电脑上编程以及当时不可能的作图是件再容易不过的事儿了，普通中学数学教育足以

本书所有的程序代码可从 <https://github.com/bingmath> 下载

龙 冰

2021 年 10 月

第一部分

数学：表达美和创造美

第一章 数学：表达美和创造美

<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。

现代的二进制记数系统由戈特弗里德·莱布尼茨于 1679 年设计，在他 1703 年发表的文章《论只使用符号 0 和 1 的二进制算术，兼论其用途及它赋予伏羲所使用的古老图形的意义》[1] 出现。与二进制数相关的系统在一些更早的文化中也有出现，包括古埃及、古代中国和古印度。中国的《易经》尤其引起了莱布尼茨的联想。

坤：黑黑黑，卦符阴阴阴艮：黑黑白，卦符阴阴阳坎：黑白黑，卦符阴阳阴巽：黑白白，卦符阴阳阳震：白黑黑，卦符阳阴阴离：白黑白，卦符阳阴阳兑：白白黑，卦符阳阳阴乾：白白白，卦符阳阳阳

$$210 \times 297 \quad 8.27 \times 11.69$$

早在 1786 年，一位德国的科学家格奥尔格·克里斯托夫·利希滕贝格 (Georg Christoph Lichtenberg, 1742 年 7 月 1 日 - 1799 年 2 月 24 日) 就发现宽高比为 2 的矩形具有许多特点 [1]。在 20 世纪初期 Walter Porstmann 将此概念应用在一系列纸张尺寸的制定，并且在 1922 年被制定为德国标准 (DIN 476)。随着各国逐渐采用，后来此标准被定为国际标准 (ISO 216)。国际标准的纸张尺寸可分为 A、B、和 C 三种系列。此标准的特色是纸张尺寸的宽高比均为 2 (约为 1.4142)。同系列但不同尺寸的纸张，其几何比例相同，因此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。A 系列 [编辑] A 系列的制定基础首先是求取一张宽高比为 2 且面积为 1 平方米 (m²) 的纸张。因此这张纸的宽长分别为 841 毫米和 1189 毫米 (宽高比为 2:1)，并且编号为 A0。若将 A0 纸张的长边对切为二，则得到两张 A1 的纸张，其宽高均为 594 毫米和 841 毫米。依此方式继续将 A1 纸张对切，则可以依序得到 A2、A3、A4 等等纸张尺寸。在制定标准时，尺寸均以整数为准，因此对切的纸张尺寸若带有小数 (小于 1 毫米) 则会舍入计算。

Format A 系列 B 系列 C 系列尺寸毫米 × 毫米 英寸 × 英寸 毫米 × 毫米 英寸 × 英寸 毫米 × 毫米 英寸 × 英寸

0	841 × 1189	33.11 × 46.81	1000 × 1414	39.37 × 55.67	917 × 1297	36.10 × 51.06
1	594 × 841	23.39 × 33.11	707 × 1000	27.83 × 39.37	648 × 917	25.51 × 36.10
2	420 × 594	16.54 × 23.39	500 × 707	19.69 × 27.83	458 × 648	18.03 × 25.51
3	297 × 420	11.69 × 16.54	353 × 500	13.90 × 19.69	324 × 458	12.76 × 18.03
4	210 × 297	8.27 × 11.69	250 × 353	9.84 × 13.90	229 × 324	9.02 × 12.76
5	148 × 210	5.83 × 8.27	176 × 250	6.93 × 9.84	162 × 229	6.38 × 9.02
6	105 × 148	4.13 × 5.83	125 × 176	4.92 × 6.93	114 × 162	4.49 × 6.38
7	74 × 105	2.91 × 4.13	88 × 125	3.46 × 4.92	81 × 114	3.19 × 4.49
8	52 × 74	2.05 × 2.91	62 × 88	2.44 × 3.46	57 × 81	2.24 × 3.19
9	37 × 52	1.46 × 2.05	44 × 62	1.73 × 2.44	40 × 57	1.57 × 2.24
10	26 × 37	1.02 × 1.46	31 × 44	1.22 × 1.73	28 × 40	1.10 × 1.57

$$186\text{mm} \times 240\text{mm} \quad 787 \times 1092 \quad 1/32$$



图 I.0.1: 简单例子

Color

I.1 常用进位制

处理字的符号的 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下：

```
import numpy as np

def incmatrix(genl1, genl2):
    m = len(genl1)
    n = len(genl2)
    M = None #to become the incidence matrix
    VT = np.zeros((n*m, 1), int) #dummy variable

    #compute the bitwise xor matrix
    M1 = bitxormatrix(genl1)
    M2 = np.triu(bitxormatrix(genl2), 1)

    for i in range(m-1):
    for j in range(i+1, m):
        [r, c] = np.where(M2 == M1[i, j])
        for k in range(len(r)):
            VT[(i)*n + r[k]] = 1;
            VT[(i)*n + c[k]] = 1;
            VT[(j)*n + r[k]] = 1;
            VT[(j)*n + c[k]] = 1;

        if M is None:
            M = np.copy(VT)
        else:
            M = np.concatenate((M, VT), 1)

    VT = np.zeros((n*m, 1), int)

    return M
```

字符串处理

```

1  ##### 生成一组 Unicode 值在 m 和 n (包括) 之间的字符列表。
2  ## 输入: Unicode 起始值 m 和终止值 (包括) n。
3  ## 输出: 所定范围内相关应字符的列表。
4  ##### Unicode 的各种字符表可见 https://www.unicode.org/charts/
5  """"
6  ## 用遍历方法生成的, 不推荐。
7  def charactersFromTo(m, n):
8      result = []
9      for x in range(m, n+1):
10         result.append(chr(x))
11     return result
12 """"
13 def charactersFromTo(m, n):
14     return list(map(lambda x: chr(x), list(range(m, n+1))))
15
16
17 ##### 汉字表。
18 def chineseCharacters():
19     return charactersFromTo(0x4E00, 0x9FFF)
20
21
22 ##### 大写希腊字母表: 0x0391 ~ 0x03A9。
23 def upperCaseGreekAlphabet():
24     return charactersFromTo(0x0391, 0x03A9)
25
26
27 ##### 小写希腊字母表: 0x03B1 ~ 0x03C9。
28 def lowerCaseGreekAlphabet():
29     return charactersFromTo(0x03B1, 0x03C9)
30
31
32 ##### 大写俄语字母表: 0x0410 ~ 0x042F。
33 def upperCaseCyrillicAlphabet():
34     return charactersFromTo(0x0410, 0x042F)
35
36
37 ##### 小写俄语字母表: 0x0430 ~ 0x044F。
38 def lowerCaseCyrillicAlphabet():
39     return charactersFromTo(0x0430, 0x044F)
40
41
42 ##### 数学运费算符号表: 0x2200 ~ 0x22FF。
43 def mathOperators():
44     return charactersFromTo(0x2200, 0x22FF)
45
46
47 #print(chineseCharacters())
48 print(" ".join(upperCaseGreekAlphabet()))
49 print(" ".join(lowerCaseGreekAlphabet()))
50
51 print(upperCaseCyrillicAlphabet())
52 print(lowerCaseCyrillicAlphabet())
53 print(" ".join(mathOperators()))
54 print(mathOperators())

```


第二部分

Python 程序设计

第二章 选择 Python 开发环境，安装软件

在 Python software installation:

Ananconda: Spyder, Jupiter 集成开发环境 (IDE - Integrated Development Environment)

Visual Studio Code

PyCharm

第三部分

PostScript 编程制图

III.1 Some graphs

Postscript Language Reference, 3rd Edition^[1] <https://www.adobe.com/content/dam/acom/en/devnet>

III.1.1 一些数学公式的几何解释

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

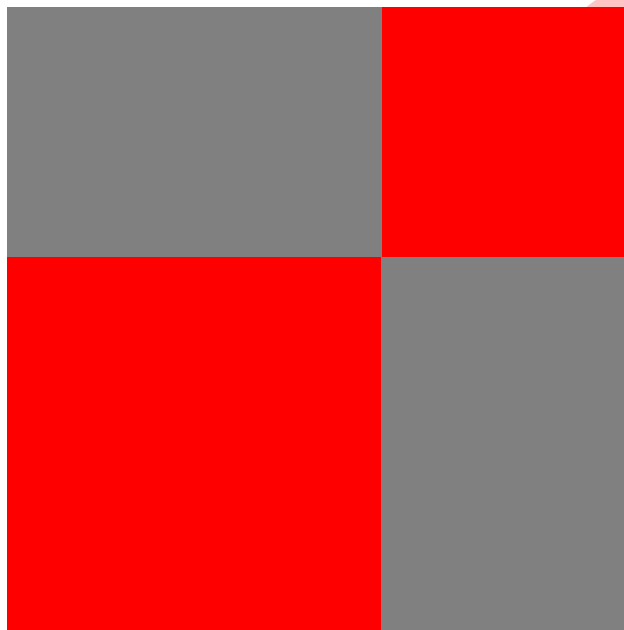


图 III.1.1: $(a + b)^2 = a^2 + 2ab + b^2$

例 III.1.1. 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。

```
line spread
line spread dfdsa;fkadf
%!PS
/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 {
```

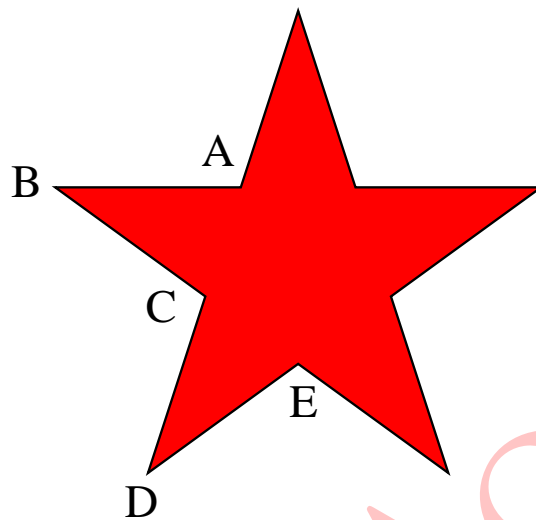



图 III.1.2: 红五星

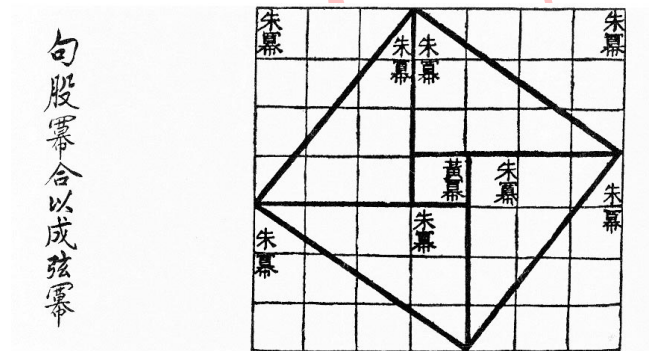


图 III.1.3: 《周髀算经》勾股定理图

```

newpath
moveto
angle %%先定义一个尖角
4 { currentpoint translate
    108 rotate
    angle } repeat %%然后靠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 %%填充红色
fill %%绘出红五星，然后轮廓路径就消失
grestore

```

```

300 500 star %%重要新闻描出五星的轮廓路径
0 setgray %%加黑边
stroke

```

showpage

Something

```

%!PS
/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 %%然后靠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 %%填充红色
fill %%绘出红五星,然后轮廓路径丢失
grestore

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

showpage

```

图 III.1.4: $(a + b)^2 = a^2 + 2ab + b^2$

III.2 A List

华罗庚数论导引^{[2]p. 1}

钟尔杰数学实验方法^[3]

钟尔杰数值分析讲义^[4]

Is God a Mathematician?^[5]

最后的数学问题^[6]

菲赫金哥尔茨《微积分学教程》^{[7][8], [9]}

参考文献

-
- [1] Adobe. Postscript Language Reference, 3rd Edition[M]. Addison-Wesley, 1999.
 - [2] 华罗庚. 数论导引[M]. 科学出版社, 1957.
 - [3] 钟尔杰. 数值分析讲义[M]. 电子科技大学出版社, 2009.
 - [4] 钟尔杰. 数学实验方法[M]. 电子科技大学出版社, 2009.
 - [5] LIVIO M. Is God a Mathematician?[M]. Simon & Schuster, 2009.
 - [6] LIVIO M. 最后的数学问题 (Is God a Mathematician? 中译本) [M]. 人民邮电出版社出版, 2020.
 - [7] 菲赫金哥尔茨. 微积分学教程, 第一卷, 第 8 版[M]. 高等教育出版社, 2006.
 - [8] 菲赫金哥尔茨. 微积分学教程, 第二卷, 第 8 版[M]. 高等教育出版社, 2006.
 - [9] 菲赫金哥尔茨. 微积分学教程, 第三卷, 第 8 版[M]. 高等教育出版社, 2006.