



《Python程序设计基础教程（微课版）》

<http://dbl-lab.xmu.edu.cn/post/python>

第15章 常用的标准库和第三方库



林子雨 博士/副教授

厦门大学计算机科学与技术系

E-mail: ziyulin@xmu.edu.cn ▶▶

主页: <http://dbl-lab.xmu.edu.cn/linziyu>





主讲教师



2017年度厦门大学奖教金获得者

2020年度厦门大学奖教金获得者

主讲教师：厦门大学 林子雨 博士/副教授

中国高校首个“数字教师”提出者和建设者

2009年7月从事教师职业以来

累计**免费**网络发布超过**1500万**字高价值教学和科研资料

网络浏览量超过**1500万**次



提纲

- 15.1 turtle库
- 15.2 random库
- 15.3 time库
- 15.4 PyInstaller库
- 15.5 jieba库
- 15.6 wordcloud库
- 15.7 Matplotlib库

本PPT是如下教材的配套讲义：
《Python程序设计基础教程（微课版）》

厦门大学 林子雨,赵江声,陶继平 编著，人民邮电出版社
《Python程序设计基础教程（微课版）》教材官方网站：
<http://dblalab.xmu.edu.cn/post/python>

高等院校程序设计新形态精品系列

PYTHON

Python Programming Language

Python 程序设计基础教程

| 微课版 |

林子雨 赵江声 陶继平 编著



名师精品

多年计算机教学实践的厚积薄发



深入浅出

清晰呈现 Python 语言学习路径



实例丰富

有效提升编程语言的学习趣味



资源全面

构建全方位一站式在线服务体系



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS



15.1 turtle库

turtle库是Python语言中一个很流行的绘制图像的函数库。引用**turtle**库的方式有如下三种：

- (1) 方式1：使用**import turtle**，函数调用时使用的语句格式是**turtle.circle(10)**；
- (2) 方式2：使用**from turtle import ***，函数调用时使用的语句格式是**circle(10)**；
- (3) 方式3：使用**import turtle as t**，函数调用时使用的语句格式是**t.circle(10)**。



15.1 turtle库

15.1.1 turtle的常用函数

15.1.2 绘图实例

本PPT是如下教材的配套讲义：

《Python程序设计基础教程（微课版）》

厦门大学 林子雨,赵江声,陶继平 编著，人民邮电出版社

《Python程序设计基础教程（微课版）》教材官方网站：

<http://dbl原因.xmu.edu.cn/post/python>

高等院校程序设计新形态精品系列

PYTHON

Python Programming Language

Python

程序设计基础教程

| 微课版 |

林子雨 赵江声 陶继平 编著



名师精品

多年计算机教学实践的厚积薄发



深入浅出

清晰呈现 Python 语言学习路径



实例丰富

有效提升编程语言的学习趣味



资源全面

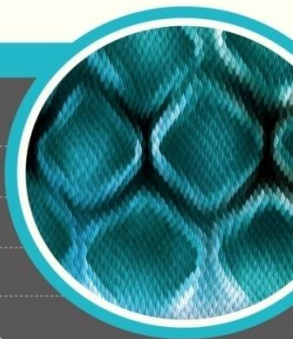
构建全方位一站式在线服务体系



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS





15.1.1 turtle的常用函数

1.设置画布

设置画布的函数如下：

```
turtle.screensize(canvwidth=None, canvheight=None, bg=None)
```

这个函数中的参数分别为画布的宽（单位是像素）、高、背景颜色，比如：

```
turtle.screensize(800,600, "green")
```

或者也可以使用如下函数：

```
turtle.setup(width=0.5, height=0.75, startx=None, starty=None)
```

在这个函数中，`width`、`height`表示宽和高，如果输入的为整数，则表示像素，如果输入的为小数，则表示占据电脑屏幕的比例。`(startx, starty)`这一坐标表示矩形窗口左上角顶点的位置，如果为空，则窗口位于屏幕中心。下面是两个实例：

```
turtle.setup(width=0.6,height=0.6)
```

```
turtle.setup(width=800,height=800, startx=100, starty=100)
```



15.1.1 turtle的常用函数

2.设置画笔

可以设置画笔颜色、画线的宽度、画笔的移动速度，具体如下：

(1) `turtle.pensize()`: 设置画笔的宽度；

(2) `turtle.pencolor()`: 如果没有参数传入，则返回当前画笔颜色，如果有传入参数，则设置画笔颜色，传入参数可以是字符串，比如“green”、“red”，也可以是RGB三元组。

(3) `turtle.speed(speed)`: 设置画笔移动速度，画笔绘制的速度范围是 $[0,10]$ 的整数，数字越大速度越快。



15.1.1 turtle的常用函数

3.绘图函数

表15-1、表15-2和表15-3分别给出了常用的画笔运动函数、画笔控制函数和其他函数。

表15-1 画笔运动函数

函数	说明
<code>turtle.forward(distance)</code>	向当前画笔方向移动distance像素长度
<code>turtle.backward(distance)</code>	向当前画笔相反方向移动distance像素长度
<code>turtle.right(degree)</code>	顺时针移动degree°
<code>turtle.left(degree)</code>	逆时针移动degree°
<code>turtle.pendown()</code>	移动时绘制图形
<code>turtle.goto(x,y)</code>	将画笔移动到坐标为x,y的位置
<code>turtle.penup()</code>	提起笔移动，不绘制图形，用于另起一个地方绘制
<code>turtle.circle()</code>	画圆，半径为正(负)，表示圆心在画笔的左边(右边)画圆
<code>turtle.setheading()</code>	设置画笔当前行进方向的角度(角度坐标体系中的绝对角度)



15.1.1 turtle的常用函数

表15-2 画笔控制函数

函数	说明
<code>turtle.fillcolor(colorstring)</code>	绘制图形的填充颜色
<code>turtle.color(color1, color2)</code>	同时设置 <code>pencolor=color1</code> , <code>fillcolor=color2</code>
<code>turtle.filling()</code>	返回当前是否在填充状态
<code>turtle.begin_fill()</code>	准备开始填充图形
<code>turtle.end_fill()</code>	填充完成
<code>turtle.hideturtle()</code>	隐藏画笔的turtle形状
<code>turtle.showturtle()</code>	显示画笔的turtle形状

表15-3 其他函数

函数	说明
<code>turtle.mainloop()</code> 或 <code>turtle.done()</code>	启动事件循环，调用Tkinter的 <code>mainloop</code> 函数
<code>turtle.delay(delay=None)</code>	设置或返回以毫秒为单位的绘图 延迟



15.1.2 绘图实例

1. 绘制五角星

下面的代码用于绘制一个五角星，绘制效果图如15-1所示。

```
01 # five-pointed-star.py
02 from turtle import Turtle
03 p = Turtle()
04 p.speed(3)
05 p.pensize(5)
06 p.color("black", "red")
07 p.begin_fill()
08 for i in range(5):
09     p.forward(200) #将箭头移到某一指定坐标
10     p.right(144) #当前方向上向右转动角度
11 p.end_fill()
```

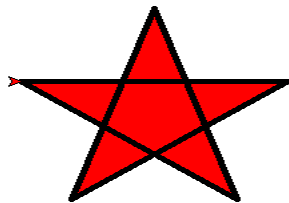


图15-1 绘制五角星



15.1.2 绘图实例

2. 绘制一条蛇

下面的代码用于绘制一条蛇，绘制效果图如15-2所示。

```
01 # snake.py
02 import turtle
03 turtle.setup(650,350,200,200)
04 turtle.penup()
05 turtle.forward(-250)
06 turtle.pendown()
07 turtle.pensize(25)
08 turtle.pencolor("purple")
09 turtle.setheading(-40)
10 for i in range(4):
11     turtle.circle(40,80)
12     turtle.circle(-40,80)
13     turtle.circle(40,80/2)
14     turtle.forward(40)
15     turtle.circle(16,180)
16     turtle.forward(40*2/3)
17 turtle.done()
```



图15-2 绘制一条蛇



15.1.2 绘图实例

3. 绘制太阳花

下面的代码用于绘制一朵太阳花，绘制效果图如15-3所示。

```
01 # sun-flower.py
02 import turtle
03 import time
04 turtle.color("red", "yellow")
05 turtle.begin_fill()
06 for i in range(50):
07     turtle.forward(200)
08     turtle.left(170)
09     turtle.end_fill()
10 turtle.mainloop()
```

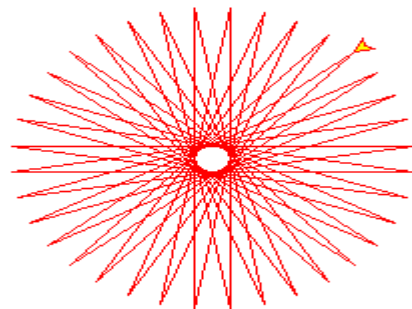


图15-3 绘制太阳花



15.2 random库

15.2.1 基本随机数函数

15.2.2 扩展随机数函数

本PPT是如下教材的配套讲义：

《Python程序设计基础教程（微课版）》

厦门大学 林子雨,赵江声,陶继平 编著，人民邮电出版社

《Python程序设计基础教程（微课版）》教材官方网站：

<http://dbl原因.xmu.edu.cn/post/python>

高等院校程序设计新形态精品系列

PYTHON

Python Programming Language

Python

程序设计基础教程

| 微课版 |

林子雨 赵江声 陶继平 编著



名师精品

多年计算机教学实践的厚积薄发



深入浅出

清晰呈现 Python 语言学习路径



实例丰富

有效提升编程语言的学习趣味



资源全面

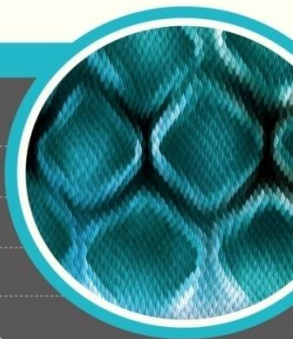
构建全方位一站式在线服务体系



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS





15.2.1 基本随机数函数

Python是通过使用随机数种子来产生随机数的，只要种子相同，产生的随机序列，无论是每一个数，还是数与数之间的关系都是确定的，所以随机数种子决定了随机序列的产生。`seed()`函数就是用来初始化给定的随机数种子，默认为当前系统时间。具体实例如下：

```
>>> import random
>>> random.seed(8) #随机数种子取值为8
>>> random.random()
0.2267058593810488
>>> random.random()
0.9622950358343828
>>> random.seed(8) #随机数种子取值为8
>>> random.random()
0.2267058593810488 #生成的随机数可以重现
```



15.2.1 基本随机数函数

```
>>> random.random()
0.9622950358343828 #生成的随机数可以重现
>>> random.seed() #随机数种子取值为当前系统时间
>>> random.random()
0.9159875847083423
>>> random.random()
0.9737887393351271
>>> random.seed() #随机数种子取值为当前系统时间
>>> random.random()
0.1830387102446276 #生成的随机数不可以重现
>>> random.random() #生成的随机数不可以重现
0.7392871445268515
```

从上面实例可以看出，当随机数种子一样时（比如，都取值为8），`random()`产生的随机数是可以重现的。但是，如果不使用随机数种子，`seed()`函数使用的是当前系统时间，后面产生的结果是完全不可重现的。



15.2.2 扩展随机数函数

1.randint(a,b)

生成一个[a,b]之间的整数，实例如下：

```
>>> import random
>>> random.randint(1,10)
7
```

2.getrandbits(k)

生成一个k比特长的随机整数，实例如下：

```
>>> import random
>>> random.getrandbits(8)
79
```

3.uniform(a,b)

生成一个[a,b]之间的随机小数，实例如下：

```
>>> import random
>>> random.uniform(1,10)
8.922182714902174
```



15.2.2 扩展随机数函数

4.random(m,n[,k])

用户生成一个[m,n]之间以k为步长的随机整数，当省略k时，默认步长为1，实例如下：

```
>>> import random
>>> random.randrange(1,10)
2
>>> random.randrange(1,10)
8
>>> random.randrange(1,10)
3
>>> random.randrange(1,10,2)
5
>>> random.randrange(1,10,2)
1
>>> random.randrange(1,10,2)
3
```



15.2.2 扩展随机数函数

5.choice(seq)

从序列seq中随机选择一个元素，实例如下：

```
>>> import random
>>> random.choice([1,2,3,4,5,6,7,8,9])
2
>>> random.choice([1,2,3,4,5,6,7,8,9])
3
>>> random.choice([1,2,3,4,5,6,7,8,9])
4
>>> random.choice([1,2,3,4,5,6,7,8,9])
6
```



15.2.2 扩展随机数函数

6.shuffle(seq)

将序列seq中元素随机排列，返回打乱以后的序列，实例如下：

```
>>> import random
>>> s=[1,2,3,4,5,6,7,8,9]
>>> random.shuffle(s)
>>> print(s)
[3, 6, 1, 4, 2, 7, 5, 8, 9]
```



15.3 time库

15.3.1 时间获取函数

15.3.2 时间格式化函数

15.3.3 程序计时函数

本PPT是如下教材的配套讲义：

《Python程序设计基础教程（微课版）》

厦门大学 林子雨,赵江声,陶继平 编著，人民邮电出版社

《Python程序设计基础教程（微课版）》教材官方网站：

<http://dbl原因.xmu.edu.cn/post/python>

高等院校程序设计新形态精品系列

PYTHON

Python Programming Language

Python

程序设计基础教程

| 微课版 |

林子雨 赵江声 陶继平 编著



名师精品

多年计算机教学实践的厚积薄发



深入浅出

清晰呈现 Python 语言学习路径



实例丰富

有效提升编程语言的学习趣味



资源全面

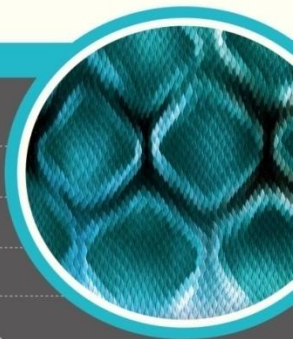
构建全方位一站式在线服务体系



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS





15.3.1 时间获取函数

1.time()

`time()`用于获取当前时间戳，即当前系统内表示时间的一个浮点数，实例如下：

```
>>> import time
>>> time.time()
1622081846.4055045
```

2. ctime()

`ctime()`用于获取当前时间，并返回一个以人类可读方式表示的字符串，实例如下：

```
>>> import time
>>> time.ctime()
'Thu May 27 10:19:18 2021'
```



15.3.1 时间获取函数

3. gmtime()

gmtime()用于获取当前时间，并返回计算机可处理的时间格式，实例如下：

```
>>> import time
>>> time.gmtime()
time.struct_time(tm_year=2021, tm_mon=5,
tm_mday=27, tm_hour=2, tm_min=20,
tm_sec=30, tm_wday=3, tm_yday=147,
tm_isdst=0)
```




15.3.2 时间格式化函数

时间格式化是将时间以合适方式展示出来的方法，类似于字符串的格式化，展示模板由特定格式化控制符组成。

1. `strftime(tpl,ts)`

`tpl`是格式化模板字符串（关于格式化字符串的详细说明请见表15-4），用来定义输出效果，`ts`是系统内部时间类型变量



15.3.2 时间格式化函数

表15-4 时间格式化字符串说明

格式化字符串	日期/时间说明	取值范围和实例
%Y	年份	0000-9999 例如: 1800
%m	月份	01-12 例如: 8
%B	月份名称	January-December 例如: July
%b	月份名称简写(三个字符)	Jan-Dec 例如: Apr
%d	日期	01-31 例如: 25
%A	星期	Monday-Sunday 例如: Tuesday
%a	星期缩写(三个字符)	Mon-Sun 例如: Wed
%H	小时(24小时)	00~23, 例如: 15
%h	小时(12小时)	01~12, 例如: 9
%P	上/下午	AM,PM, 例如:AM
%M	分钟	00~59, 例如: 34
%S	秒	00~59, 例如: 42



15.3.2 时间格式化函数

实例如下：

```
>>> import time
>>> t=time.gmtime()
>>> time.strftime("%Y-%m-%d %H:%M:%S",t)
'2021-05-27 02:48:30'
>>> time.strftime("%Y-%B-%d-%A-%H-%p-%S")
'2021-May-27-Thursday-10-AM-02'
>>> time.strftime("%A-%p")
'Thursday-AM'
>>> time.strftime("%M:%S")
'49:14'
```



15.3.2 时间格式化函数

2. `strptime(str,tpl)`

`str`是字符串形式的时间值，`tpl`是格式化模板字符串，用来定义输入效果，实例如下：

```
>>> import time
>>> timeStr='2021-05-27 10:26:21'
>>> time.strptime(timeStr,"%Y-%m-%d %H:%M:%S")
time.struct_time(tm_year=2021, tm_mon=5, tm_mday=27,
tm_hour=10, tm_min=26, tm_sec=21, tm_wday=3, tm_yday=147,
tm_isdst=-1)
```



15.3.3程序计时函数

- 程序计时是指测量程序从开始到结束所经历的时间，主要包括测量时间和产生时间两部分。
- time**库提供了一个非常精准的测量时间函数`perf_counter()`，该函数可以获取CPU以其频率运行的时钟，这个时间往往是以纳秒来计算的，所以这样获取的时间非常精准。
- Time**库提供的产生时间函数`sleep()`，可以让程序休眠或产生一段时间。



15.3.3程序计时函数

1. perf_counter()

perf_counter()会返回系统运行时间，由于返回值的基准点是未定义的，所以，只有连续调用的结果之间的差值才是有效的。实例如下：

```
>>> import time
>>> start=time.perf_counter()
>>> print(start)
1497.733367483
>>> end=time.perf_counter()
>>> print(end)
1522.141805637
>>> end-start
24.408438154000123
```



15.3.3程序计时函数

2. sleep(s)

在sleep(s)中，s为休眠时间，单位是秒，可以是浮点数，实例如下：

```
>>> import time
>>> def wait():
            time.sleep(4.5)
>>> wait()
```




15.3.3程序计时函数

3.应用实例

这里使用程序计时函数`perf_counter()`和`sleep(s)`实现一个实时显示程序执行进度的效果，具体代码如下：

```
01 #program_process.py
02 import time
03 scale = 40
04 print('执行开始')
05 start = time.perf_counter()
06 for i in range(scale+1):
07     a = '*' * i
08     b = '.' * (scale - i)
09     c = (i / scale) * 100
10     dur = time.perf_counter() - start
11     print("\r{:^3.0f}%[{}->{}]{:.2f}".format(c,a,b,dur))
12     time.sleep(0.1)
13 print('\n'+'执行结束')
```



15.3.3程序计时函数

```
执行开始
0 % [--->] 0.00
2 % [*-->] 0.10
5 % [***->] 0.21
8 % [****->] 0.32
10 % [*****->] 0.43
12 % [*****->] 0.54
15 % [*****->] 0.65
18 % [*****->] 0.76
20 % [*****->] 0.87
22 % [*****->] 0.97
25 % [*****->] 1.08
28 % [*****->] 1.18
30 % [*****->] 1.29
32 % [*****->] 1.39
35 % [*****->] 1.50
38 % [*****->] 1.61
40 % [*****->] 1.72
42 % [*****->] 1.83
45 % [*****->] 1.93
48 % [*****->] 2.04
50 % [*****->] 2.15
52 % [*****->] 2.26
55 % [*****->] 2.36
57 % [*****->] 2.49
60 % [*****->] 2.62
62 % [*****->] 2.74
65 % [*****->] 2.86
68 % [*****->] 2.98
70 % [*****->] 3.10
72 % [*****->] 3.22
75 % [*****->] 3.36
78 % [*****->] 3.49
80 % [*****->] 3.63
82 % [*****->] 3.74
85 % [*****->] 3.87
88 % [*****->] 4.00
90 % [*****->] 4.12
92 % [*****->] 4.24
95 % [*****->] 4.36
98 % [*****->] 4.50
100% [*****->] 4.61
```

执行结束



15.4 PyInstaller库

PyInstaller库用于将Python源代码文件转换成exe格式的可执行文件。在Windows系统中，打开一个cmd窗口，执行如下命令就可以在Python3环境中安装PyInstaller库：

```
> pip install pyinstaller
```

PyInstaller的常用参数如表15-5所示。

参数	作用
-h	查看帮助
--clean	清理打包过程临时文件
-D	默认值，生成dist文件夹
-F	只在dist文件夹中生成打包文件
-i<图标文件名.ico>	指定打包文件使用的图标文件



15.4 PyInstaller库

PyInstaller的最简单的使用方法是：

```
pyinstaller -F <文件名.py>
```

假设已经有一个代码文件“C:\mycode\hello.py”，里面只有一行代码“print("Hello World")”，可以使用如下命令生成可执行文件：

```
> cd C:\mycode
```

```
> pyinstaller -F hello.py
```

执行完上述命令以后，在“C:\mycode”目录下会生成三个新的目录，分别是__pycache__、build和dist，进入dist目录，里面就包含了一个可执行文件hello.exe，双击该文件就可以执行。



15.5 jieba库

15.5.1 jieba库简介

15.5.2 jieba的安装和使用

15.5.3 应用实例

本PPT是如下教材的配套讲义：

《Python程序设计基础教程（微课版）》

厦门大学 林子雨,赵江声,陶继平 编著，人民邮电出版社

《Python程序设计基础教程（微课版）》教材官方网站：

<http://dbl原因.xmu.edu.cn/post/python>

高等院校程序设计新形态精品系列

PYTHON

Python Programming Language

Python

程序设计基础教程

| 微课版 |

林子雨 赵江声 陶继平 编著



名师精品

多年计算机教学实践的厚积薄发



深入浅出

清晰呈现 Python 语言学习路径



实例丰富

有效提升编程语言的学习趣味



资源全面

构建全方位一站式在线服务体系



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS



15.5.1 jieba库简介

Jieba库是一款流行的Python第三方中文分词库。**jieba**分词采用的是基于统计的分词方法，首先给定大量已经分好词的文本，利用机器学习的方法学习分词规律，然后保存训练好的模型，从而实现对新的文本的分词。具体而言，包括如下步骤：

- (1) 先加载自带的字典，生成**trie**树；
- (2) 给定待分词的句子，使用正则表达式获取连续的中文字符和英文字符，切分成短语列表，对每个短语使用**DAG**（有向无环图）和动态规划，得到最大概率路径，对**DAG**中那些没有在字典中查到的字，组合成一个新的片段短语，使用**HMM**模型（隐马尔科夫模型）进行分词，也就是识别新词，即识别字典外的新词；
- (3) 使用Python的**yield**语法生成一个词语生成器，逐词语返回。



15.5.1 jieba库简介

jieba中文分词支持三种分词模式：

(1) 精确模式：试图将语句最精确地切分，不存在冗余数据，适合做文本分析；

(2) 全模式：将语句中所有可能是词的词语都切分出来，速度很快，但是存在冗余数据；

(3) 搜索引擎模式：在精确模式的基础上，对长词再次切分，提高召回率，适合用于搜索引擎分词。



15.5.1 jieba库简介

表15-6给出了jieba库的常用函数及其说明。

函数	说明
<code>jieba.cut(s)</code>	精确模式，返回一个可迭代的数据类型
<code>jieba.cut(s,cut_all= True)</code>	全模式，输出s中的所有可能单词
<code>jieba.cut_for_search(s)</code>	搜索引擎模式
<code>jieba.lcut(s)</code>	精确模式，返回一个列表类型
<code>jieba.lcut(s,cut_all= True)</code>	全模式，返回一个列表类型
<code>jieba.lcut_for_search(s)</code>	搜索引擎模式，返回一个列表类型
<code>jieba.add_word(w)</code>	向分词词典中增加新词w



15.5.2 jieba的安装和使用

在Windows系统中，打开一个cmd窗口，执行如下命令就可以在Python3环境中安装jieba库：

```
> pip install jieba
```



15.5.2 jieba的安装和使用

新建一个代码文件jieba_test.py，内容如下：

```
01 # -*- coding: utf-8 -*-
02 # jieba_test.py
03 import jieba
04 #全模式
05 text = "我来到厦门大学数据库实验室"
06 seg_list = jieba.cut(text, cut_all=True)
07 print(u"[全模式]: ", "/" .join(seg_list))
08
09 #精确模式
10 seg_list = jieba.cut(text, cut_all=False)
11 print(u"精确模式]: ", "/" .join(seg_list))
12
13 #默认是精确模式
14 seg_list = jieba.cut(text)
15 print(u"[默认模式]: ", "/" .join(seg_list))
16
17 #搜索引擎模式
18 seg_list = jieba.cut_for_search(text)
19 print(u"[搜索引擎模式]: ", "/" .join(seg_list))
```



15.5.2 jieba的安装和使用

代码的执行结果如下：

[全模式]: 我/ 来到/ 厦门/ 厦门大学/ 大学/ 数据/ 数据库/ 据库/ 实验/ 实验室

精确模式]: 我/ 来到/ 厦门大学/ 数据库/ 实验室

[默认模式]: 我/ 来到/ 厦门大学/ 数据库/ 实验室

[搜索引擎模式]: 我/ 来到/ 厦门/ 大学/ 厦门大学/ 数据/ 据库/ 数据库/ 实验/ 实验室



15.5.3 应用实例

这里给出一个具体应用实例，完成的功能是，给定一段语句，使用jieba分词对语句进行分词，并统计出出现次数排在前3位的词语，具体实现代码如下：

```
01 # -*- coding: utf-8 -*-
02 #wordcount.py
03 import jieba
04
05 text="厦门大学设有研究生院、6个学部以及30个学院和16个研究院，形成了包括人文科学、社会科学、自然科学、工程与技术科学、管理科学、艺术科学、医学科学等学科门类在内的完备学科体系。学校现有18个学科进入ESI全球前1%，拥有5个一级学科国家重点学科、9个二级学科国家重点学科。学校设有32个博士后流动站；36个博士学位授权一级学科，45个硕士学位授权一级学科；8个交叉学科；1个博士专业学位学科授权类别，28个硕士专业学位学科授权类别。”
```



15.5.3 应用实例

```
06 words = jieba.cut(text) # 使用精确模式对文本进行分词
07 counts = {} # 通过键值对的形式存储词语及其出现的次数
08
09 for word in words:
10     if len(word) == 1: # 不对单个字的词语进行统计
11         continue
12     else:
13         counts[word] = counts.get(word, 0) + 1 # 词语每出现一次，其对应的次数加 1
14
15 items = list(counts.items())
16 items.sort(key=lambda x: x[1], reverse=True) # 根据词语出现的次数进行从大到小排序
17
18 for i in range(3):
19     word, count = items[i]
20     print("{0:<4}{1:>4}".format(word, count))
```



15.6 wordcloud库

`wordcloud`是优秀的词云展示第三方库，它可以根据文本中词语出现的频率等参数绘制词云，而且词云的绘制形状、尺寸和颜色都可以设定。`Python`安装好以后，默认是没有安装`wordcloud`库的，需要单独安装。在`Windows`系统中打开一个`cmd`窗口，执行如下命令安装`wordcloud`库：

```
> pip install wordcloud
```



15.6 wordcloud库

在使用wordcloud制作词云时，首先要声明一个WordCloud对象，语法如下：

```
w=wordcloud.WordCloud(<参数>);
```

对于一个WordCloud对象w，它可以使用的基本函数如下：

- w.generate(): 向WordCloud对象中加载文本；
 - w.to_file(filename): 将词云输出为图像文件（png或jpg格式）。
- 对于一个WordCloud对象w，可以配置如表15-7所示的各种参数。



15.6 wordcloud库

表15-7 WordCloud对象的配置参数

参数	描述
width	指定词云对象生成图片的宽度，默认为400像素 实例： <code>w=wordcloud.WordCloud(width=500)</code>
height	指定词云对象生成图片的高度，默认为200像素 实例： <code>w=wordcloud.WordCloud(height=300)</code>
min_font_size	指定词云中字体的最小字号，默认为4号 实例： <code>w=wordcloud.WordCloud(min_font_size=10)</code>
max_font_size	指定词云中字体的最大字号，根据高度自动调节 实例： <code>w=wordcloud.WordCloud(max_font_size=20)</code>
font_step	指定词云中字体字号的步进间隔，默认为1 实例： <code>w=wordcloud.WordCloud(font_step=2)</code>
font_path	指定字体文件的路径，默认为None 实例： <code>w=wordcloud.WordCloud(font_path="msyh.ttc")</code>
max_words	指定词云显示的最大单词数量，默认为200 实例： <code>w=wordcloud.WordCloud(max_words=20)</code>
stop_words	指定词云的排除词列表，即不显示的单词列表 实例： <code>w=wordcloud.WordCloud(stop_words="Python")</code>
mask	指定词云形状，默认为长方形 实例： <pre>import imageio #需要事先安装imageio库 mk=imageio.imread("pic.png") w=wordcloud.WordCloud(mask=mk)</pre>
background_color	指定词云图片的背景颜色，默认为黑色 实例： <code>w=wordcloud.WordCloud(background_color="white")</code>



15.6 wordcloud库

绘制词云包含3个主要步骤：

- 配置对象参数；
- 加载词云文本；
- 输出文本。



15.6 wordcloud库

下面是制作词云的简单实例：

```
01 # wordcloud_university.py
02 import jieba
03 import wordcloud
04 txt="厦门大学设有研究生院、6个学部以及30个学院和16个研究院，形成了包括人文科学、社会科学、自然科学、工程与技术科学、管理科学、艺术科学、医学科学等学科门类在内的完备学科体系。学校现有18个学科进入ESI全球前1%，拥有5个一级学科国家重点学科、9个二级学科国家重点学科。学校设有32个博士后流动站；36个博士学位授权一级学科，45个硕士学位授权一级学科；8个交叉学科；1个博士专业学位学科授权类别，28个硕士专业学位学科授权类别。"
05
w=wordcloud.WordCloud(width=1000,font_path="C:\\Windows\\Fonts\\msyh.ttf",height=700)
06 w.generate(" ".join(jieba.lcut(txt)))
07 w.to_file("university.png")
```



15.6 wordcloud库

程序执行成功后会生成一个名称为“university.png”的词云图片（如图15-5所示）。



图15-5 一个词云的简单实例



15.7 Matplotlib库

Matplotlib是Python最著名的绘图库，它提供了一整套和Matlab相似的命令API，十分适合交互式地进行制图。而且也可以方便地将它作为绘图控件，嵌入到GUI应用程序中。Matplotlib能够创建多种类型的图表，如条形图、散点图、饼图、堆叠图、3D图和地图图表。

Python安装好以后，默认是没有安装Matplotlib库的，需要单独安装。在Windows系统中打开一个cmd窗口，执行如下命令安装Matplotlib库：

```
> pip install matplotlib
```



15.7 Matplotlib库

- 15.7.1 绘制折线图
- 15.7.2 绘制条形图
- 15.7.3 绘制直方图
- 15.7.4 绘制饼图

本PPT是如下教材的配套讲义：
《Python程序设计基础教程（微课版）》

厦门大学 林子雨,赵江声,陶继平 编著，人民邮电出版社
《Python程序设计基础教程（微课版）》教材官方网站：
<http://dbl原因.xmu.edu.cn/post/python>

高等院校程序设计新形态精品系列

PYTHON

Python Programming Language

Python 程序设计基础教程

| 微课版 |

林子雨 赵江声 陶继平 编著



名师精品

多年计算机教学实践的厚积薄发



深入浅出

清晰呈现 Python 语言学习路径



实例丰富

有效提升编程语言的学习趣味



资源全面

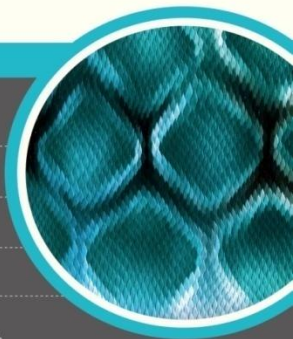
构建全方位一站式在线服务体系



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS





15.7.1 绘制折线图

首先要导入pyplot模块:

```
>>> import matplotlib.pyplot as plt
```

接下来, 调用plot方法绘制一些坐标:

```
>>> plt.plot([1,2,3],[4,8,5])
```

plot()方法需要很多参数, 但是最主要的是前2个参数, 分别表示x坐标和y坐标, 比如, 上面语句中放入了两个列表[1,2,3]和[4,8,5], 就表示生成了3个坐标(1,4)、(2,8)和(3,5)。

下面可以把图表显示到屏幕上 (如图15-6所示):

```
>>> plt.show()
```

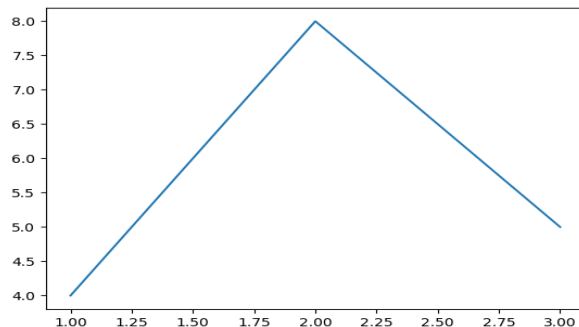


图15-6 三个坐标点生成的折线图



15.7.1 绘制折线图

下面画出两条折线，并且给每条折线一个名称：

```
>>> x = [1,2,3]   #第1条折线的横坐标
>>> y = [4,8,5]   #第1条折线的纵坐标
>>> x2 = [1,2,3]   #第2条折线的横坐标
>>> y2 = [11,15,13] #第2条折线的纵坐标
>>> plt.plot(x, y, label='First Line') #绘制第1条折线，给折线一个名称
'First Line'
>>> plt.plot(x2, y2, label='Second Line') #绘制第2条折线，给折线一个名
称'Second Line'
>>> plt.xlabel('Plot Number') #给横坐标轴添加名称
>>> plt.ylabel('Important var') #给纵坐标轴添加名称
>>> plt.title('Graph Example\nTwo lines') #添加标题
>>> plt.legend() #添加图例
>>> plt.show() #显示到屏幕上（如图15-7所示）
```



15.7.1 绘制折线图

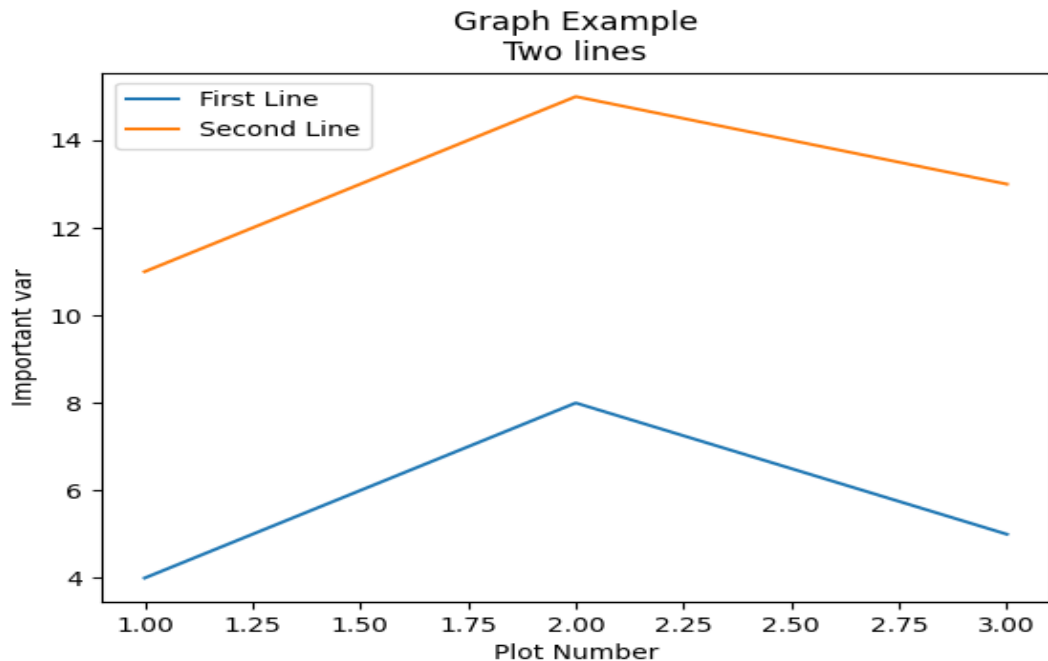


图15-7 绘制两条折线



15.7.2 绘制条形图

下面介绍条形图的绘制方法。

```
>>> plt.bar([1,3,5,7,9],[6,3,8,9,2], label="First Bar") #第1个数据系列
>>> #下面的color='g', 表示设置颜色为绿色
>>> plt.bar([2,4,6,8,10],[9,7,3,6,7], label="Second Bar", color='g') #第2
个数据系列
>>> plt.legend() #添加图例
>>> plt.xlabel('bar number') #给横坐标轴添加名称
>>> plt.ylabel('bar height') #给纵坐标轴添加名称
>>> plt.title('Bar Example\nTwo bars!') #添加标题
>>> plt.show() #显示到屏幕上（如图15-8所示）
```



15.7.2 绘制条形图

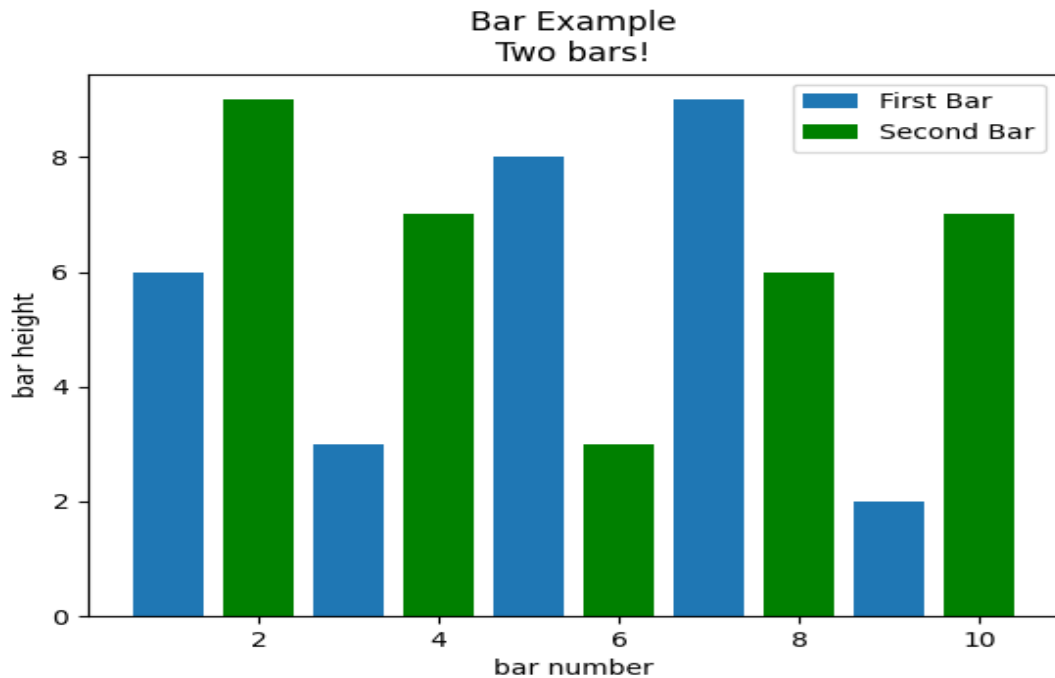


图15-8 条形图



15.7.3 绘制直方图

下面介绍直方图的绘制方法。

```
>>> population_ages = [21,57,61,47,25,21,33,41,41,5,96,103,108,  
                        121,122,123,131,112,114,113,82,77,67,56,46,44,45,47]  
>>> bins = [0,10,20,30,40,50,60,70,80,90,100,110,120,130]  
>>> plt.hist(population_ages, bins, histtype='bar', rwidth=0.8)  
>>> plt.xlabel('x')  
>>> plt.ylabel('y')  
>>> plt.title('Graph Example\n Histogram')  
>>> plt.show() #显示到屏幕上（如图15-9所示）
```



15.7.3 绘制直方图

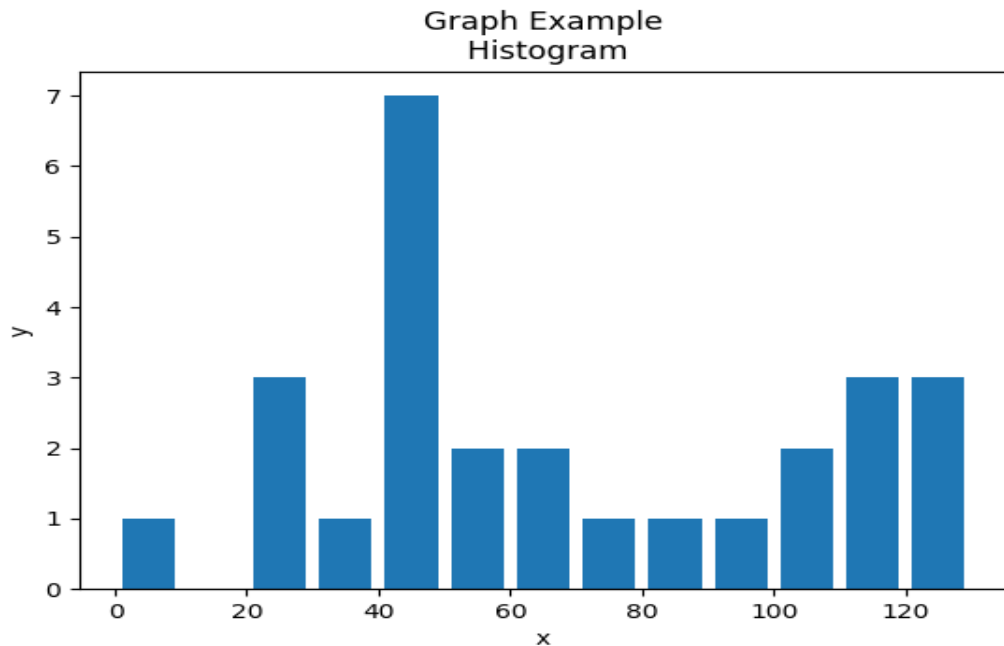


图15-9 直方图



15.7.4 绘制饼图

下面介绍饼图的绘制方法。

```
>>> slices = [7,2,2,13] #即activities分别占比7/24,2/24,2/24,13/24
>>> activities = ['sleeping','eating','working','playing']
>>> cols = ['c','m','r','b']
>>> plt.pie(slices,
            labels=activities,
            colors=cols,
            startangle=90,
            shadow= True,
            explode=(0,0.1,0,0),
            autopct='%1.1f%%')
>>> plt.title('Graph Example\n Pie chart')
>>> plt.show() #显示到屏幕上（如图15-10所示）
```



15.7.4 绘制饼图

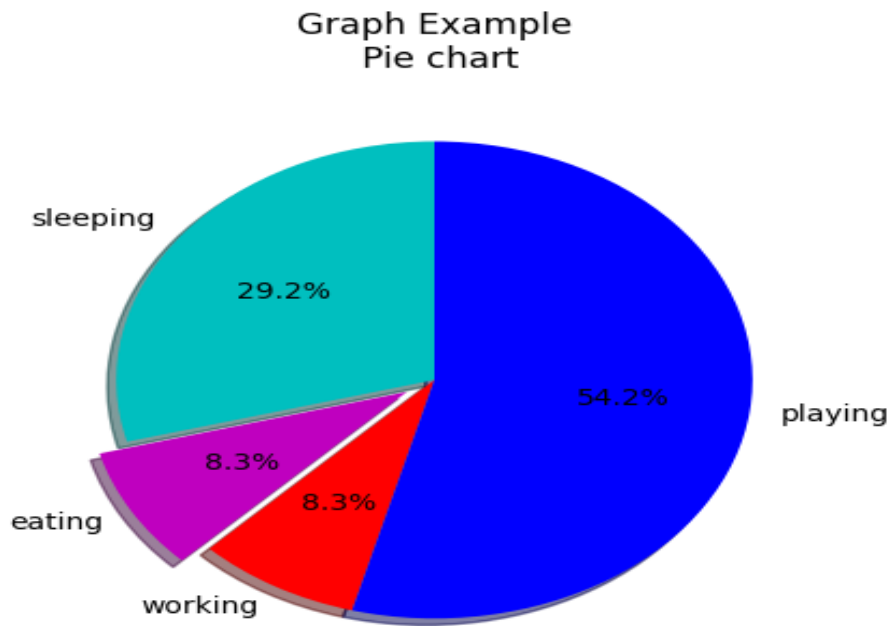


图15-10 饼图



附录A：主讲教师林子雨简介



主讲教师：林子雨

单位：厦门大学计算机科学与技术系

E-mail: ziyulin@xmu.edu.cn

个人网页: <http://dmlab.xmu.edu.cn/post/linziyu>

数据库实验室网站: <http://dmlab.xmu.edu.cn>

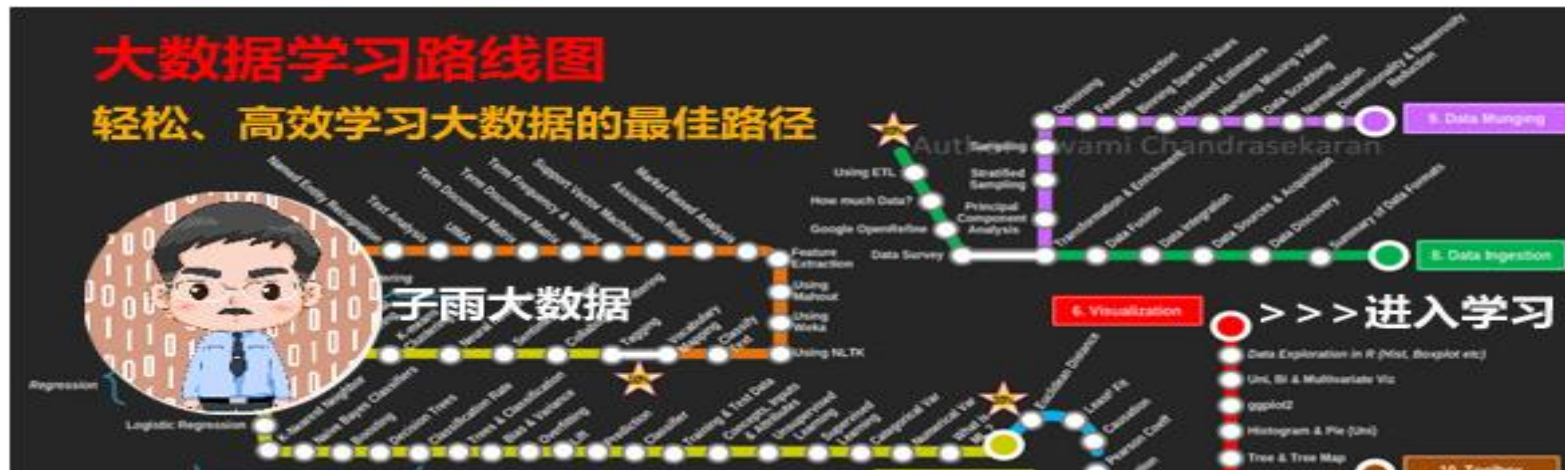


扫一扫访问个人主页

林子雨，男，1978年出生，博士（毕业于北京大学），全国高校知名大数据教师，现为厦门大学计算机科学系副教授，厦门大学信息学院实验教学中心主任，曾任厦门大学信息科学与技术学院院长助理、晋江市发展和改革局副局长。中国计算机学会数据库专业委员会委员，中国计算机学会信息系统专业委员会委员。国内高校首个“数字教师”提出者和建设者，厦门大学数据库实验室负责人，厦门大学云计算与大数据研究中心主要建设者和骨干成员，2013年度、2017年度和2020年度厦门大学教学类奖教金获得者，荣获2019年福建省精品在线开放课程、2018年厦门大学高等教育成果特等奖、2018年福建省高等教育教学成果二等奖、2018年国家精品在线开放课程。主要研究方向为数据库、数据仓库、数据挖掘、大数据、云计算和物联网，并以第一作者身份在《软件学报》《计算机学报》和《计算机研究与发展》等国家重点期刊以及国际学术会议上发表多篇学术论文。作为项目负责人主持的科研项目包括1项国家自然科学基金青年基金项目(No.61303004)、1项福建省自然科学基金青年基金项目(No.2013J05099)和1项中央高校基本科研业务费项目(No.2011121049)，主持的教改课题包括1项2016年福建省教改课题和1项2016年教育部产学协作育人项目，同时，作为课题负责人完成了国家发改委城市信息化重大课题、国家物联网重大应用示范工程区域试点泉州市工作方案、2015泉州市互联网经济调研等课题。中国高校首个“数字教师”提出者和建设者，2009年至今，“数字教师”大平台累计向网络免费发布超过1000万字高价值的研究和教学资料，累计网络访问量超过1000万次。打造了中国高校大数据教学知名品牌，编著出版了中国高校第一本系统介绍大数据知识的专业教材《大数据技术原理与应用》，并成为京东、当当网等网店畅销书籍；建设了国内高校首个大数据课程公共服务平台，为教师教学和学生学习大数据课程提供全方位、一站式服务，年访问量超过400万次，累计访问量超过1500万次。



附录B：大数据学习路线图



大数据学习路线图访问地址：<http://dblab.xmu.edu.cn/post/10164/>



附录C：林子雨大数据系列教材



林子雨大数据系列教材

用于导论课、专业课、实训课、公共课

了解全部教材信息：<http://dbllab.xmu.edu.cn/post/bigdatabook/>



附录D：《大数据导论（通识课版）》教材

开设全校公共选修课的优质教材



本课程旨在实现以下几个培养目标：

- 引导学生步入大数据时代，积极投身大数据的变革浪潮之中
- 了解大数据概念，培养大数据思维，养成数据安全意识
- 认识大数据伦理，努力使自己的行为符合大数据伦理规范要求
- 熟悉大数据应用，探寻大数据与自己专业的应用结合点
- 激发学生基于大数据的创新创业热情

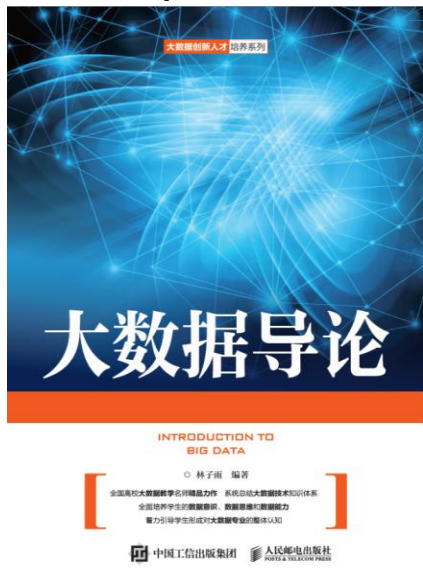
高等教育出版社 ISBN:978-7-04-053577-8 定价：32元 版次：2020年2月第1版
教材官网：<http://dbllab.xmu.edu.cn/post/bigdataintroduction/>



附录E：《大数据导论》教材

- 林子雨 编著《大数据导论》
- 人民邮电出版社，2020年9月第1版
- ISBN:978-7-115-54446-9 定价：49.80元

教材官网：<http://dblab.xmu.edu.cn/post/bigdata-introduction/>



开设大数据专业导论课的优质教材



扫一扫访问教材官网



附录F：《大数据技术原理与应用（第3版）》教材

《大数据技术原理与应用——概念、存储、处理、分析与应用（第3版）》，由厦门大学计算机科学系林子雨博士编著，是国内高校第一本系统介绍大数据知识的专业教材。人民邮电出版社 ISBN:978-7-115-54405-6 定价：59.80元

全书共有17章，系统地论述了大数据的基本概念、大数据处理架构Hadoop、分布式文件系统HDFS、分布式数据库HBase、NoSQL数据库、云数据库、分布式并行编程模型MapReduce、Spark、流计算、Flink、图计算、数据可视化以及大数据在互联网、生物医学和物流等各个领域的应用。在Hadoop、HDFS、HBase、MapReduce、Spark和Flink等重要章节，安排了入门级的实践操作，让读者更好地学习和掌握大数据关键技术。

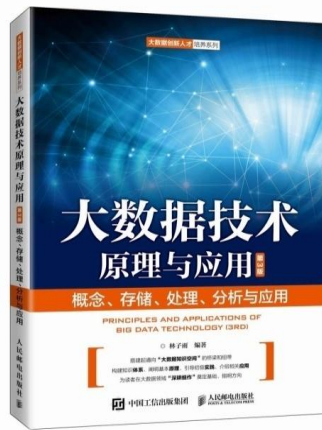
本书可以作为高等院校计算机专业、信息管理等相关专业的大数据课程教材，也可供相关技术人员参考、学习、培训之用。

欢迎访问《大数据技术原理与应用——概念、存储、处理、分析与应用》教材官方网站：

<http://dbllab.xmu.edu.cn/post/bigdata3>



扫一扫访问教材官网





附录G：《大数据基础编程、实验和案例教程（第2版）》

本书是与《大数据技术原理与应用（第3版）》教材配套的唯一指定实验指导书



大数据教材

配套实验指导书



1+1黄金组合
厦门大学林子雨编著

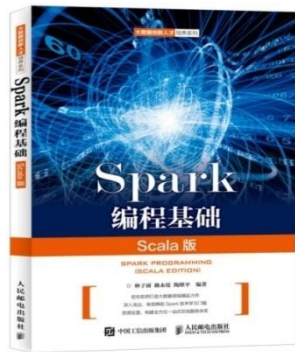
- 步步引导，循序渐进，详尽的安装指南为顺利搭建大数据实验环境铺平道路
- 深入浅出，去粗取精，丰富的代码实例帮助快速掌握大数据基础编程方法
- 精心设计，巧妙融合，八套大数据实验题目促进理论与编程知识的消化和吸收
- 结合理论，联系实际，大数据课程综合实验案例精彩呈现大数据分析全流程

林子雨编著《大数据基础编程、实验和案例教程（第2版）》

清华大学出版社 ISBN:978-7-302-55977-1 定价：69元 2020年10月第2版



附录H: 《Spark编程基础 (Scala版)》



《Spark编程基础 (Scala版)》

厦门大学 林子雨, 赖永炫, 陶继平 编著

披荆斩棘, 在大数据丛林中开辟学习捷径
填沟削坎, 为快速学习Spark技术铺平道路
深入浅出, 有效降低Spark技术学习门槛
资源全面, 构建全方位一站式在线服务体系

人民邮电出版社出版发行, ISBN:978-7-115-48816-9

教材官网: <http://dblab.xmu.edu.cn/post/spark/>

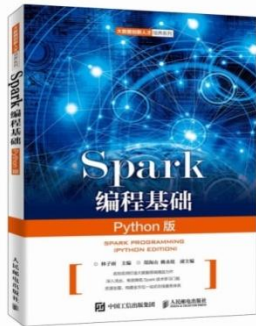


本书以Scala作为开发Spark应用程序的编程语言, 系统介绍了Spark编程的基础知识。全书共8章, 内容包括大数据技术概述、Scala语言基础、Spark的设计与运行原理、Spark环境搭建和使用方法、RDD编程、Spark SQL、Spark Streaming、Spark MLlib等。本书每个章节都安排了入门级的编程实践操作, 以便读者更好地学习和掌握Spark编程方法。本书官网免费提供了全套的在线教学资源, 包括讲义PPT、习题、源代码、软件、数据集、授课视频、上机实验指南等。



附录I: 《Spark编程基础 (Python版)》

《Spark编程基础 (Python版)》



厦门大学 林子雨, 郑海山, 赖永炫 编著

披荆斩棘, 在大数据丛林中开辟学习捷径
填沟削坎, 为快速学习Spark技术铺平道路
深入浅出, 有效降低Spark技术学习门槛
资源全面, 构建全方位一站式在线服务体系



人民邮电出版社出版发行, ISBN:978-7-115-52439-3

教材官网: <http://dbllab.xmu.edu.cn/post/spark-python/>

本书以Python作为开发Spark应用程序的编程语言, 系统介绍了Spark编程的基础知识。全书共8章, 内容包括大数据技术概述、Spark的设计与运行原理、Spark环境搭建和使用方法、RDD编程、Spark SQL、Spark Streaming、Structured Streaming、Spark MLlib等。本书每个章节都安排了入门级的编程实践操作, 以便读者更好地学习和掌握Spark编程方法。本书官网免费提供了全套的在线教学资源, 包括讲义PPT、习题、源代码、软件、数据集、上机实验指南等。



附录J：高校大数据课程公共服务平台



高校大数据课程

公 共 服 务 平 台

<http://dblab.xmu.edu.cn/post/bigdata-teaching-platform/>



扫一扫访问平台主页



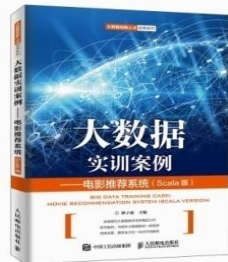
扫一扫观看3分钟FLASH动画宣传片



附录K：高校大数据实训课程系列案例教材

为了更好地满足高校开设大数据实训课程的教材需求，厦门大学数据库实验室林子雨老师团队联合企业共同开发了《高校大数据实训课程系列案例》，目前已经完成开发的系列案例包括：

- 《电影推荐系统》（已经于2019年5月出版）
- 《电信用户行为分析》（已经于2019年5月出版）
- 《实时日志流处理分析》
- 《微博用户情感分析》
- 《互联网广告预测分析》
- 《网站日志处理分析》



系列案例教材将于2019年陆续出版发行，教材相关信息，敬请关注网页后续更新！

<http://dblab.xmu.edu.cn/post/shixunkecheng/>



扫一扫访问大数据实训课程系列案例教材主页

The background is a solid blue color with faint, light blue silhouettes of people. At the top, there are two groups of people holding hands. On the right side, there is a silhouette of a person standing with their hand on their head. At the bottom left, there are silhouettes of people sitting at a table.

Thank You!

Department of Computer Science, Xiamen University, 2022