



# 第6届全国高校大数据与人工智能教学研讨会

2023.05.12-2023.05.13 中国·厦门

主办单位：教育部高等学校计算机类专业教学指导委员会

承办单位：



协办单位：



# Python语言贯穿的系列教材

## 教学方法与实践

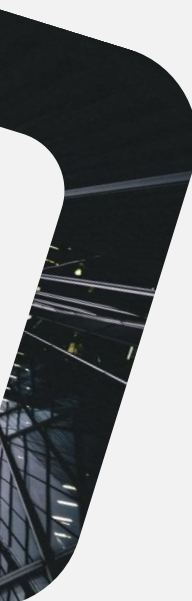
主讲人：周元哲

西安邮电大学

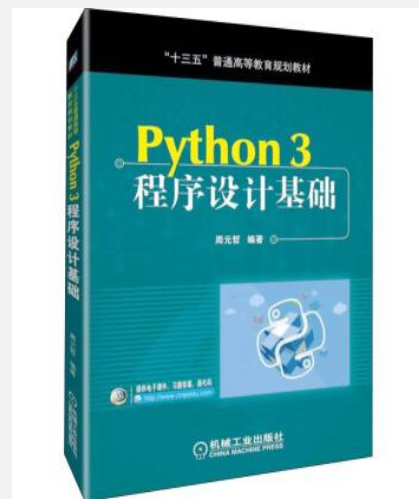
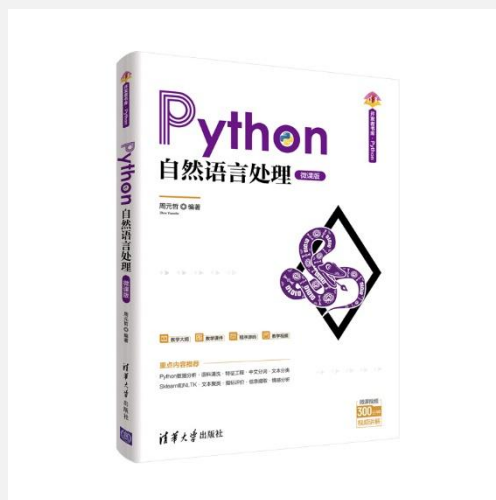
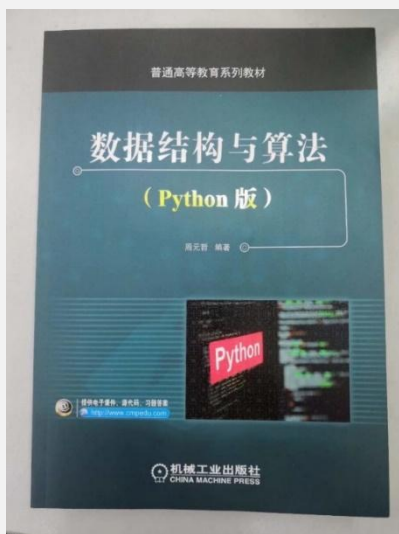
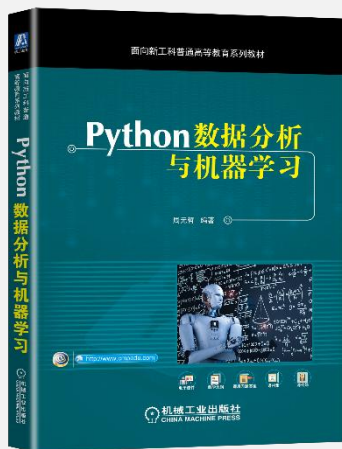
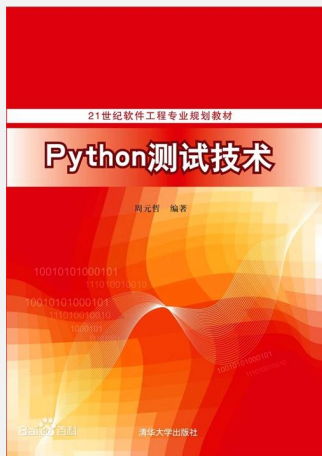
Email: [zhouyuanzhe@163.com](mailto:zhouyuanzhe@163.com)

2023年5月





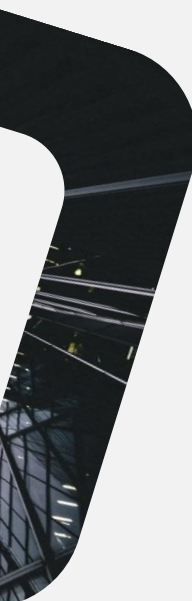
当今时代，人工智能和大数据的崛起带红了Python。Python由于自身语言的优势，如简洁的语法，众多的第三方开源库，在数据科学上广泛应用，备受学习者的青睐，成为目前最火的网红编程语言。各个高校纷纷开设Python课程

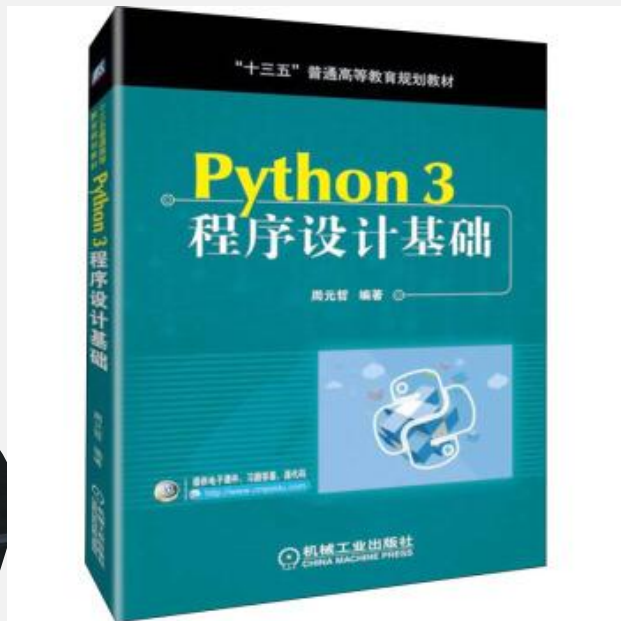


# 教学实践

——以《Python3程序设计基础》为例



- 
- 1、强调为什么，是什么，怎么办？
  - 2、强调语言的对比，如Python 和 C 的异同点
  - 3、强调知识点的应用场合
  - 4、强调知识点在本课程内以及课程相互之间的衔接关系



《Python3程序设计基础》，包括Python编程概述、Python编程基础、序列、流程控制、函数和模块。重点介绍Python语言序列、三大流程控制等，重在培养学生掌握编程的思想及其编程方法，意在提高学生的编程开发能力。

《Python二级考试》

<https://www.lanqiao.cn/courses/5670>

《新手一起学Python》

<https://www.educoder.net/paths/11077>

python使用缩进表示代码块。C语言使用大括号 `{}` 表示缩进，认为缩进是代码书写风格“有了更好”，而不是“没有不行”。但是，Python将缩进作为语法要求，必须缩进，否则会出现语法错误，这样使得代码不但具有层次性，而且可读性大为改善。

```
if(x > y) {  
    x = 1;  
    y = 2;  
}
```

```
if x > y:  
    x = 1  
    y = 2
```

python使用代码块的缩进来体现代码之间的逻辑关系，行首的空白称为缩进，缩进结束就表示一个代码块结束了。缩进的空格数是可变的，但是同一个代码块的语句必须包含相同的缩进空格数。

注意：要么都是空格，要是么都tab制表符，千万别混用。



算术运算符：如除法 (/) ， 整除 (//) ,求余数 (%) 应用场合

【解析】水仙花数是指一个三位数，若每位的数字的立方和等于该数字本身。  
一个三位正整数，例如， $153=1*1*1+5*5*5+3*3*3$ ，故153是水仙花数。

```
x=int(input("输入一个三位正整数："))    #从键盘上输入一个三位的正整数
```

```
bai=x//100          #百位
```

```
shi=(x-100*bai)//10    #十位
```

```
ge=x-100*bai-10*shi    #个位
```



## 列表添加元素的对比

方法一：使用“+”将一个新列表附加在原列表的尾部；

```
>>> a_list = [1]
>>> a_list = a_list + ['a', 2.0]
>>> a_list
[1, 'a', 2.0]
```

方法二：使用append()方法向列表尾部添加一个新元素；

```
>>> a_list = [1, 'a', 2.0]
>>> a_list.append(True)
>>> a_list
[1, 'a', 2.0, True]
```

方法三：使用extend()方法将一个列表添加在原列表的尾部；

```
>>> a_list = [1, 'a', 2.0, True]
>>> a_list.extend(['x', 4])
>>> a_list
[1, 'a', 2.0, True, 'x', 4]
```

方法四：使用insert()方法将一个元素插入到列表的任意位置。

```
>>> a_list = [1, 'a', 2.0, True, 'x', 4]
>>> a_list.insert(0, 'x')
>>> a_list
['x', 1, 'a', 2.0, True, 'x', 4]
```

列表与元组的区别：不同点在于列表是可变序列

与列表相比，元组具有以下优点。

(1) 元组的处理速度和访问速度比列表快。如果定义了一系列常量值，主要对其进行遍历或者类似用途，而不需要对其元素进行修改，这种情况一般使用元组。可以认为元组对不需要修改的数据进行了“写保护”，可以使代码更安全。

(2) 作为不可变序列，元组（包含数值、字符串和其他元组的不可变数据）可用作字典的键，而列表不可以充当字典的键，因为列表是可变的。



## 字符串

### 子串查找

```
str.find(substr, [start, [, end]])
```

```
>>> s1="beijing xi'an tianjin beijing chongqing"
```

```
>>> s1.find("beijing")
```

```
0
```

```
>>> s1.find("beijing",3)
```

```
22
```

```
>>> s1.find("beijing",3,20)
```

```
-1
```

思考：【1】 正则表达式，python的re模块

【2】 引申到《数据结构》的KMP算法



字典（**dict**）在其他程序设计语言中称为映射（**map**），通过键值对（**key-value**）存储数据，键和值之间用冒号间隔，元素项之间用逗号间隔，整体用一对大括号“{”和“}”括起来。字典语法结构如下所示：

```
dict_name={key:value, key:value}
```

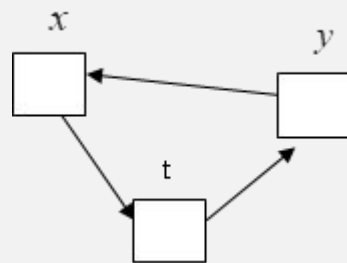
字典有如下特性：

- （1）字典的值可以是任意数据类型，包括字符串、整数、对象，甚至字典。
- （2）键/值对用冒号分割，而键值对用逗号分隔，所有这些都包括在花括号中
- （3）键/值没有顺序。
- （4）键必须是唯一，不允许同一个键重复出现

思考：【1】 应用场合，网页数据结构JSON

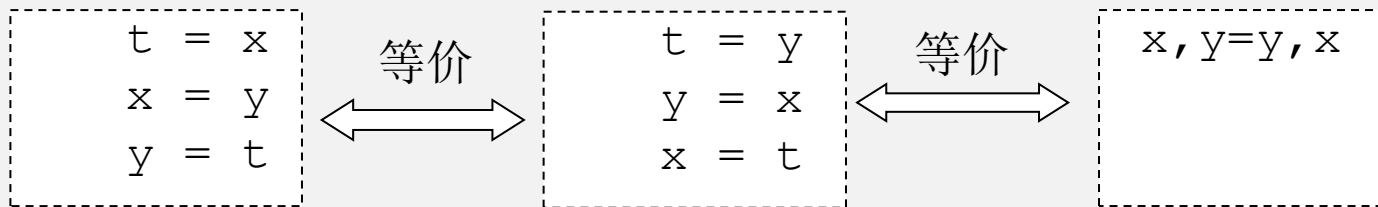
【2】 引申到《数据结构》的哈希算法等

## 三大流程控制：选择结构



【例】从键盘上输入两个正整数 $x$ 和 $y$ ，升序输出。

if  $x > y$ : #如果 $x$ 大于 $y$ 条件成立，则引入 $t$ 交换 $x$ 和 $y$



思考：【1】python的语法与C语言的语法区别

【2】引申题意，如果是三个数，需要多少条if语句

方法一	方法二
<b>if的单分支语句</b>	<b>if语句的多分支</b>
<pre>hour=int(input("hour")) if hour&lt;=12:     print("Good morning") if (hour&gt;12) and (hour&lt;18):     print("Good afternoon") if hour&gt;=18:     print("Good Evening")</pre>	<pre>hour=int(input("hour")) if hour &lt;= 12 :     print("Good morning") elif hour &lt; 18:     print("Good afternoon") else:     print("Good Evening")</pre>
<p>程序执行按照三个if语句的顺序依次执行。例如，hour小于12，则第1个if语句的判断条件hour&lt;=12为真，执行“Good morning”；之后还要对第2个和第3个if语句的判断条件进行执行。而在这种情况下，第2个和第3个if语句已经没有必要执行了。</p>	<p>程序执行按照if语句的多分支执行。例如，hour小于12，则第1个if语句的判断条件hour&lt;=12为真，执行“Good morning”；之后不再对第2个和第3个if语句的判断条件进行执行，实现功能。</p>
<b>三条if的单分支语句的并列使用虽然实现功能，但效率较差</b>	<b>采用if语句的多分支语句执行效率较快</b>

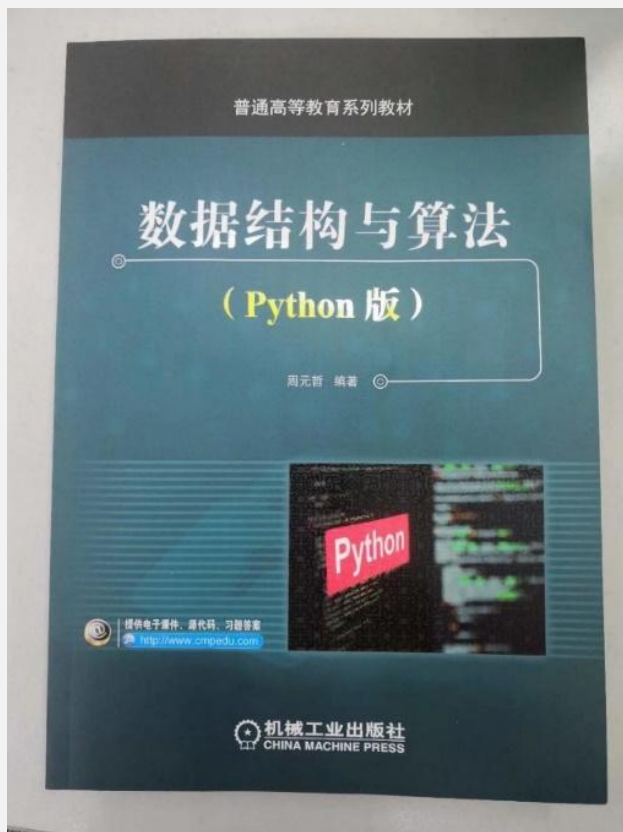


方法一	方法二	方法三
<pre>mark = int(input("输入x值")) if mark &gt;= 90:     print("优秀") elif mark &gt;= 80:     print("良好") elif mark &gt;= 70:     print("中等") elif mark &gt;= 60:     print("及格") else:     print("不及格")</pre>	<pre>mark = int(input("输入x值")) if mark &lt;= 60:     print("不及格") elif mark &lt;= 70:     print("及格") elif mark &lt;= 80:     print("中等") elif mark &lt;= 90:     print("良好") else:     print("优秀")</pre>	<pre>mark = int(input("输入x值")) if mark &gt;= 60:     print("及格") elif mark &gt;= 70:     print("中等") elif mark &gt;= 80:     print("良好") elif mark &gt;= 90:     print("优秀") else:     print("不及格")</pre>



# 数据结构与算法 (Python)

《数据结构与算法 (Python)》 (周元哲) 机械工业出版社 2020.9  
(蓝桥云课) 每章视频讲解网址: <https://www.lanqiao.cn/courses/5046>



本书具有如下特点：

- 1、代码详解：传统的数据结构教材“重理论轻代码”，往往只是给出伪代码。而本书的代码都是采用Python实现。
- 2、图文并茂：本书通过Python语言的特性，例如使用Python的*deque*讲解栈、*networkX*讲解图论，使得数据结构算法可视化，从而便于学生更快地掌握数据结构的思想，提高学生的编程应用开发能力。
- 3、突出实用性。本书每章都有一个或几个Python实现该章内容的案例。

# 串——BF匹配算法

第一趟匹配

a b a b c a b c a c b a b  
a b c a c

第二趟匹配

a b a b c a b c a c b a b  
a b c a c

第三趟匹配

a b a b c a b c a c b a b  
a b c a c

第四趟匹配

a b a b c a b c a c b a b  
a b c a c

第五趟匹配

a b a b c a b c a c b a b  
a b c a c

第六趟匹配

a b a b c a b c a c b a b  
a b c a c

根据BF算法思想，需要将搜索词整个后移一位，再从 头 逐个比较。搜索词需要把“搜索位置”移到已经比较过的位置，重新再比一遍。这就是BF 算法速度较慢的原因所在——进行了没有必要的回溯行为。

## KMP——改进后的匹配情况：

第一趟匹配

	1	2	3	4	5	6	7	8	9
			↓						
	a	b	a	b	c	a	b	c	a
	a	b	c	a	c				

第二趟匹配

	1	2	3	4	5	6	7	8	9
			↓	→	→	→	↓		
	a	b	a	b	c	a	b	c	a
			a	b	c	a	c		

第三趟匹配

	1	2	3	4	5	6	7	8	9
						↓	→	→	↓
	a	b	a	b	c	a	b	c	a
						(a)	b	c	a

## 串——KMP匹配算法

KMP算法不把"搜索位置"移回已经比较过的位置，而是继续向后移。KMP算法需要用到《部分匹配表》（Partial Match Table），"部分匹配值"就是"前缀"和"后缀"的最长的共有元素的长度。

搜索词	A	B	C	D	A	B	D
部分匹配值	0	0	0	0	1	2	0

移动位数

= 已匹配的字符数 - 对应的部分匹配值

- "ABCD"的前缀为[A, AB, ABC], 后缀为[BCD, CD, D], 共有元素的长度为0;
- "ABCDA"的前缀为[A, AB, ABC, ABCD], 后缀为[BCDA, CDA, DA, A], 共有元素为"A", 长度为1;
- "ABCDAB"的前缀为[A, AB, ABC, ABCD, ABCDA], 后缀为[BCDAB, CDAB, DAB, AB, B], 共有元素为"AB", 长度为2;

## NetworkX库

- NetworkX在2002年5月产生，是一个用Python语言开发的图论与复杂网络建模工具，networkx支持创建简单无向图、有向图和多重图；内置许多标准的图论算法，节点可为任意数据；支持任意的边值维度，内置了常用的图与复杂网络分析算法，可以方便的进行复杂网络数据分析、仿真建模等工作，功能丰富，简单易用。
- Networkx的制图功能，需要安装matplotlib和numpy

# Python在图论中的应用，将逻辑抽象进行可视化

graph editor是一个在线画图软件

输入网址：[https://csacademy.com/app/graph\\_editor/](https://csacademy.com/app/graph_editor/)

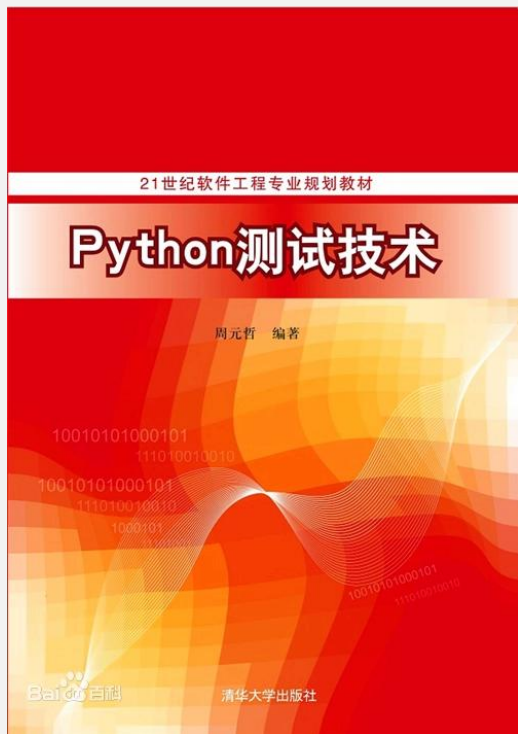
The screenshot displays the CS Academy Graph Editor interface. At the top, there are navigation tabs for 'Undirected' and 'Directed' graphs, and options for '0-index', '1-index', and 'Custom Labels'. A toolbar includes buttons for 'Force', 'Draw', 'Edit', 'Delete', and 'Config'. On the left, a sidebar shows 'Node Count' (6) and 'Graph Data' (a list of node degrees: 0, 1, 2, 3, 4, 5, 0, 2, 0, 4, 0, 5, 1, 4, 1, 5, 2, 3, 2, 4, 4, 5). The central canvas shows a graph with 6 nodes (0-5) and edges. Node 3 is connected to node 2. Node 2 is connected to nodes 4 and 0. Node 4 is connected to nodes 0, 5, and 1. Node 0 is connected to nodes 2, 4, and 1. Node 5 is connected to nodes 4 and 1. Node 1 is connected to nodes 0, 4, and 5. On the right, a 'Force mode' box explains that nodes are pulled towards the center and exert forces on each other. Below this box are buttons for 'Download as PNG' and 'Generate Markup'.

## **collections**模块

namedtuple、deque、Counter、OrderedDict和defaultdict等



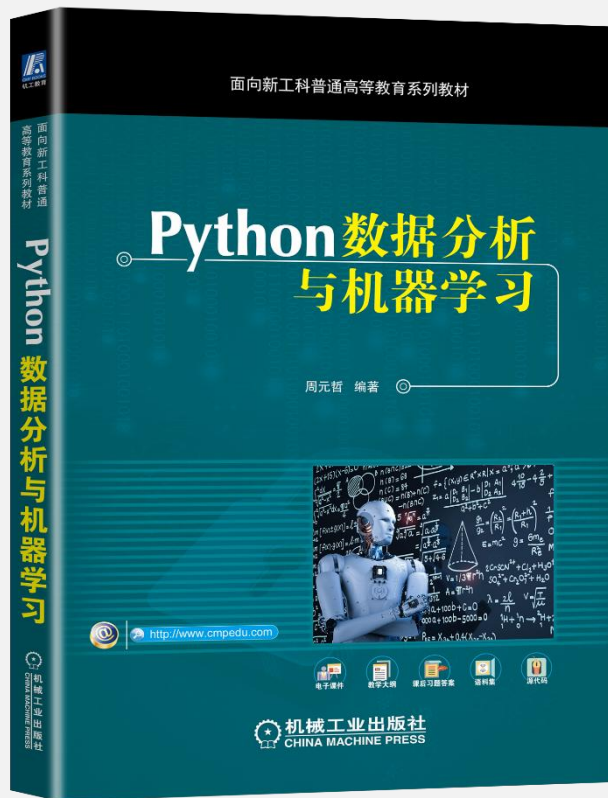
《Python测试技术》,讲述了与Python语言相关的测试,包括单元测试unittest,web自动化测试selenium和移动app自动化测试。具体章节包括软件测试基础、自动测试技术、python测试框架、Python与unittest单元测试、python与DDT数据驱动测试、python与selenium网络测试、Android UI测试、Python与Appium移动测试等相关内容。





《Python数据分析与机器学习》

《Python数据分析与机器学习》（周元哲）机械工业出版社



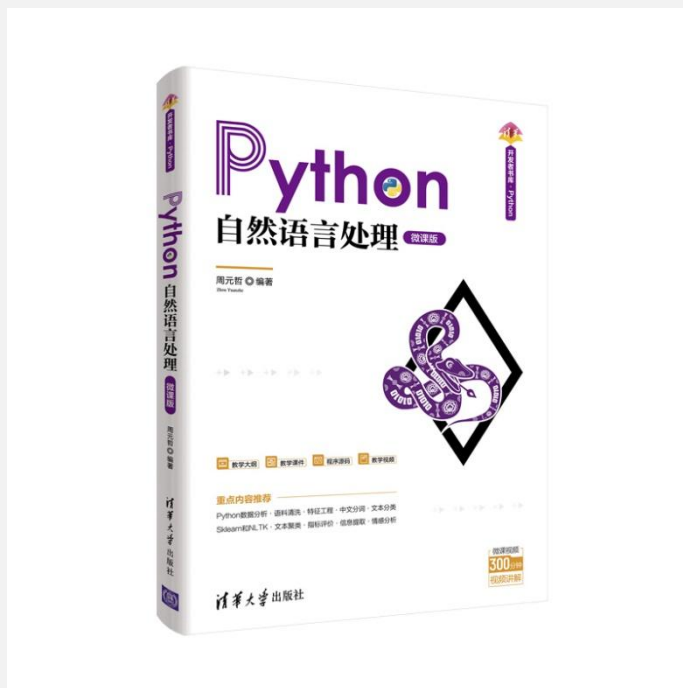
## 机器学习入门——基于SKlearn

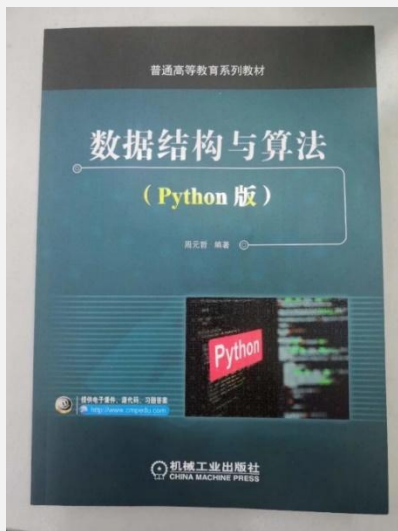
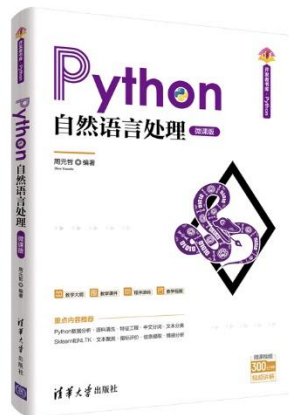
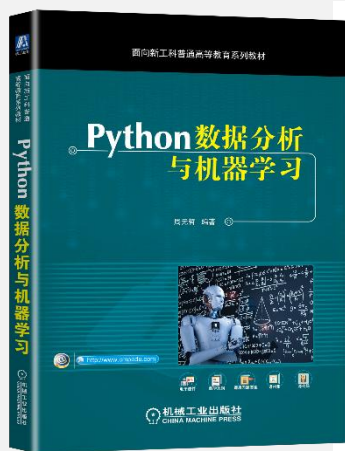
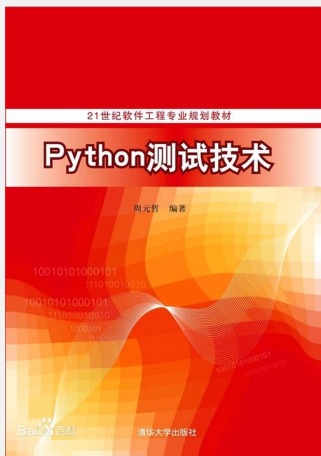
《机器学习入门——基于SKlearn》,与Python语言相关的机器学习内容,包括数据预处理、特征工程、指标评价、K近邻算法、决策树、线性模型、朴素贝叶斯、支持向量机、Kmeans算法。



《Python自然语言处理》（周元哲）清华大学出版社 2021年11月

本教材包括自然语言处理概述、Sklearn和NLTK、语料清洗、特征工程、中文分词、文本分类、文本聚类、指标评价、信息提取和情感分析。





# 谢 谢

主讲人：周元哲

西安邮电大学

Email: [zhouyuanzhe@163.com](mailto:zhouyuanzhe@163.com)

