

# python

The Python logo, consisting of two interlocking snakes, one blue and one yellow, is positioned below the word "python".

```
import turtle
turtle.setup(650,350,200,200)
turtle.penup()
turtle.fd(-250)
turtle.pendown()
turtle.pensize(25)
turtle.pencolor("purple")

for i in range(4):
    turtle.circle(40, 80)
    turtle.circle(-40, 80)
    turtle.circle(40, 80/2)
    turtle.fd(40)
    turtle.circle(16, 180)
    turtle.fd(40 * 2/3)
```

Python语言程序设计

# 一维数据的格式化和处理

---



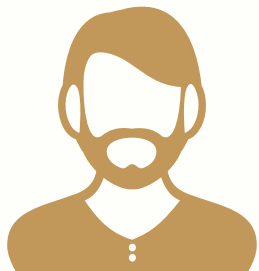
嵩 天  
北京理工大学



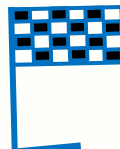


# 单元开篇

# 一维数据的格式化和处理



- 数据组织的维度
- 一维数据的表示
- 一维数据的存储
- 一维数据的处理





# 数据组织的维度

# 从一个数据到一组数据

3.14



3.1413

3.1398

3.1404

3.1401

3.1349

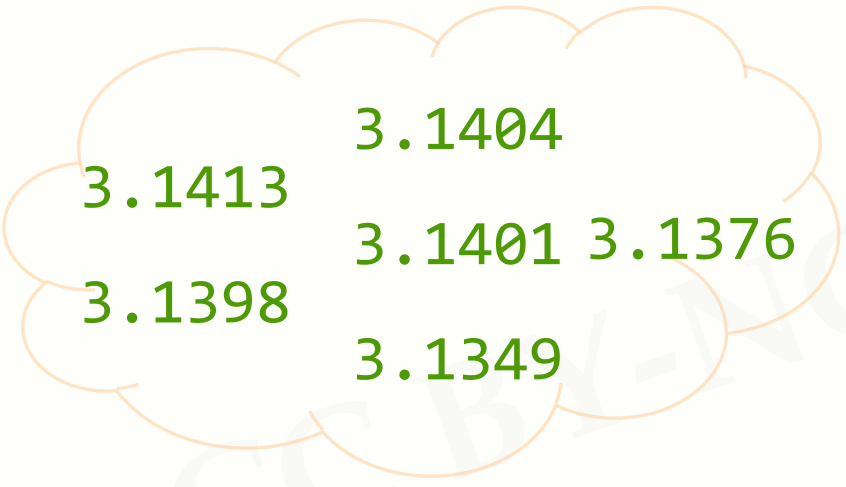
**一个数据**

**表达一个含义**

**一组数据**

**表达一个或多个含义**

# 维度：一组数据的组织形式



3.1413  
3.1398  
3.1404  
3.1401  
3.1376  
3.1349

一组数据



3.1413, 3.1398, 3.1404, 3.1401, 3.1349, 3.1376

或



3.1398, 3.1349, 3.1376  
3.1413, 3.1404, 3.1401

数据的组织形式

# 一维数据

由对等关系的有序或无序数据构成，采用线性方式组织

3.1413, 3.1398, 3.1404, 3.1401, 3.1349, 3.1376

- 对应列表、数组和集合等概念



# 二维数据

由多个一维数据构成，是一维数据的组合形式

排名	学校名称	省市	总分	指标得分
				生源质量（新生高考成绩得分） ▾
1	清华大学	北京	94.0	100.0
2	北京大学	北京	81.2	96.1
3	浙江大学	浙江	77.8	87.2
4	上海交通大学	上海	77.5	89.4
5	复旦大学	上海	71.1	91.8
6	中国科学技术大学	安徽	65.9	91.9
7	南京大学	江苏	65.3	87.1
8	华中科技大学	湖北	63.0	80.6
9	中山大学	广东	62.7	81.1
10	哈尔滨工业大学	黑龙江	61.6	76.4

表格是典型的二维数据

其中，表头是二维数据的一部分

# 多维数据

由一维或二维数据在新维度上扩展形成

排名	学校名称	省市	总分	指标得分
				生源质量（新生高考成绩得分） ▾
1	清华大学	北京市	95.9	100.0
2	北京大学	北京市	82.6	98.9
3	浙江大学	浙江省	80	88.8
4	上海交通大学	上海市	78.7	90.6
5	复旦大学	上海市	70.9	90.4
6	南京大学	江苏省	66.1	90.7
7	中国科学技术大学	安徽省	65.5	90.1
8	哈尔滨工业大学	黑龙江省	63.5	80.9
9	华中科技大学	湖北省	62.9	83.5
10	中山大学	广东省	62.1	81.8

时间维度



2016

排名	学校名称	省市	总分	指标得分
				生源质量（新生高考成绩得分） ▾
1	清华大学	北京	94.0	100.0
2	北京大学	北京	81.2	96.1
3	浙江大学	浙江	77.8	87.2
4	上海交通大学	上海	77.5	89.4
5	复旦大学	上海	71.1	91.8
6	中国科学技术大学	安徽	65.9	91.9
7	南京大学	江苏	65.3	87.1
8	华中科技大学	湖北	63.0	80.6
9	中山大学	广东	62.7	81.1
10	哈尔滨工业大学	黑龙江	61.6	76.4

2017

# 高维数据

仅利用最基本的二元关系展示数据间的复杂结构

```
{  
  "firstName" : "Tian" ,  
  "lastName"  : "Song" ,  
  "address"   : {  
    "streetAddr" : "中关村南大街5号" ,  
    "city"       : "北京市" ,  
    "zipcode"    : "100081"  
  } ,  
  "professional" : ["Computer Networking" , "Security"]  
}
```

键值对

# 数据的操作周期

存储 <-> 表示 <-> 操作





# 一维数据的表示

# 一维数据的表示

如果数据间有序：使用列表类型

```
ls = [3.1398, 3.1349, 3.1376]
```

- 列表类型可以表达一维有序数据
- for循环可以遍历数据，进而对每个数据进行处理

# 一维数据的表示

如果数据间无序：使用集合类型

```
st = {3.1398, 3.1349, 3.1376}
```

- 集合类型可以表达一维无序数据
- for循环可以遍历数据，进而对每个数据进行处理



# 一维数据的存储



# 一维数据的存储

## 存储方式一：空格分隔

中国 美国 日本 德国 法国 英国 意大利

- 使用一个或多个空格分隔进行存储，不换行
- 缺点：数据中不能存在空格

# 一维数据的存储

## 存储方式二：逗号分隔

中国,美国,日本,德国,法国,英国,意大利

- 使用英文半角逗号分隔数据进行存储，不换行
- 缺点：数据中不能有英文逗号

# 一维数据的存储

## 存储方式三：其他方式

中国\$美国\$日本\$德国\$法国\$英国\$意大利

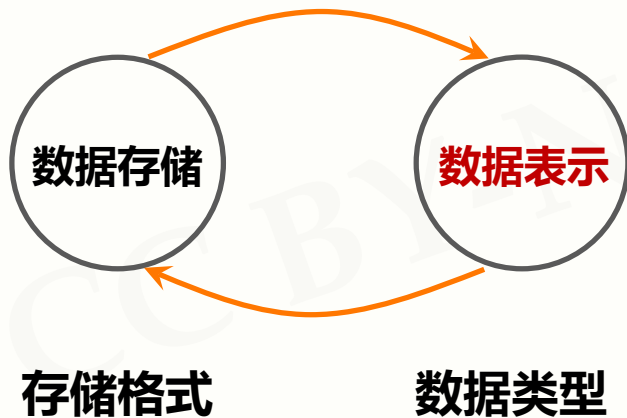
- 使用其他符号或符号组合分隔，建议采用特殊符号
- 缺点：需要根据数据特点定义，通用性较差



# 一维数据的处理

# 数据的处理

存储  $\leftrightarrow$  表示



- 将存储的数据读入程序
- 将程序表示的数据写入文件

# 一维数据的读入处理

## 从空格分隔的文件中读入数据

中国 美国 日本 德国 法国 英国 意大利

```
txt = open(fname).read()
```

```
ls = txt.split()
```

```
f.close()
```

```
>>> ls
```

```
['中国', '美国', '日本', '德国',  
, '法国', '英国', '意大利']
```

# 一维数据的读入处理

## 从特殊符号分隔的文件中读入数据

中国\$美国\$日本\$德国\$法国\$英国\$意大利

```
txt = open(fname).read()
```

```
ls = txt.split("$")
```

```
f.close()
```

```
>>> ls
```

```
['中国', '美国', '日本', '德国',  
, '法国', '英国', '意大利']
```

# 一维数据的写入处理

采用空格分隔方式将数据写入文件

```
ls = ['中国', '美国', '日本']
```

```
f = open(fname, 'w')
```

```
f.write(' '.join(ls))
```

```
f.close()
```



# 一维数据的写入处理

采用特殊分隔方式将数据写入文件

```
ls = ['中国', '美国', '日本']
```

```
f = open(fname, 'w')
```

```
f.write('$'.join(ls))
```

```
f.close()
```



# 单元小结

# 一维数据的格式化和处理

- 数据的维度：一维、二维、多维、高维
- 一维数据的表示：列表类型(有序)和集合类型(无序)
- 一维数据的存储：空格分隔、逗号分隔、特殊符号分隔
- 一维数据的处理：字符串方法 `.split()` 和 `.join()`





# 小花絮

# Python是最受欢迎的编程语言吗？

嵩老师从来不评论各种编程语言排名

因为

在现在或很近的未来

Python必然排名第一！又何必评论？



