

数据类型/建立复杂的数据结构

陈斌 北京大学 gischen@pku.edu.cn

# 建立复杂的数据结构

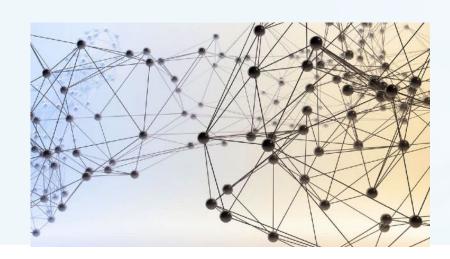
- > 比较几种数据结构
- > 建立大型数据结构

## 比较几种数据结构

- 〉使用方括号[]创建列表
- 〉使用圆括号()创建元组
- 〉使用花括号{}创建字典
- 每种类型中,都可以通过方括号[]对单个元素进行访问
- 对于列表和元组,方括号里是整型的偏移量
- 对于字典, 方括号里的是键
- 都返回元素的值

# 建立大型数据结构

- 》 将这些内置的数据结构自由地组合成 更大、更复杂的结构
- )创建自定义数据结构的过程中,唯一 的限制来自于这些内置数据类型本身



## 建立大型数据结构:列表/元组

#### 〉建立3个不同的列表

```
alist = [1,2,3]
blist = ['Hello', 'Python']
clist = [True, False]
```

#### 〉嵌套列表/元组

```
list_of_lists = [[1,2,3],
['Hello','Python'], [True,False]]
tuple_of_lists = ([1,2,3],
['Hello','Python'], [True,False])
```

## 建立大型数据结构:字典

#### 〉嵌套字典

```
dict_of_lists = {'num':[1,2,3],
  'word':['Hello', 'Python'],
  'bool':[True,False]}
```

〉字典的元素可以是任意类型,甚至也 可以是字典

## 建立大型数据结构:字典

#### 〉字典的键值可以是任意不可变类型

```
例如: 用元组来作为坐标,索引元素

poi={(100,100):'bus stop'}

>>> alist=[[23, 34, 45], [True, 'ab']]
>>> alist[0][2]
45
>>> bands={'Marxes':['Moe','Curly'], 'KK':[True, 'moon']}
>>> bands['KK'][0]
True
>>> poi={(100,100):'Zhongguancun', (123,23):'Pizza'}
>>> poi[(100,100)]
'Zhongguancun'
```