Tedu.cn
达内教育

Python开发入门

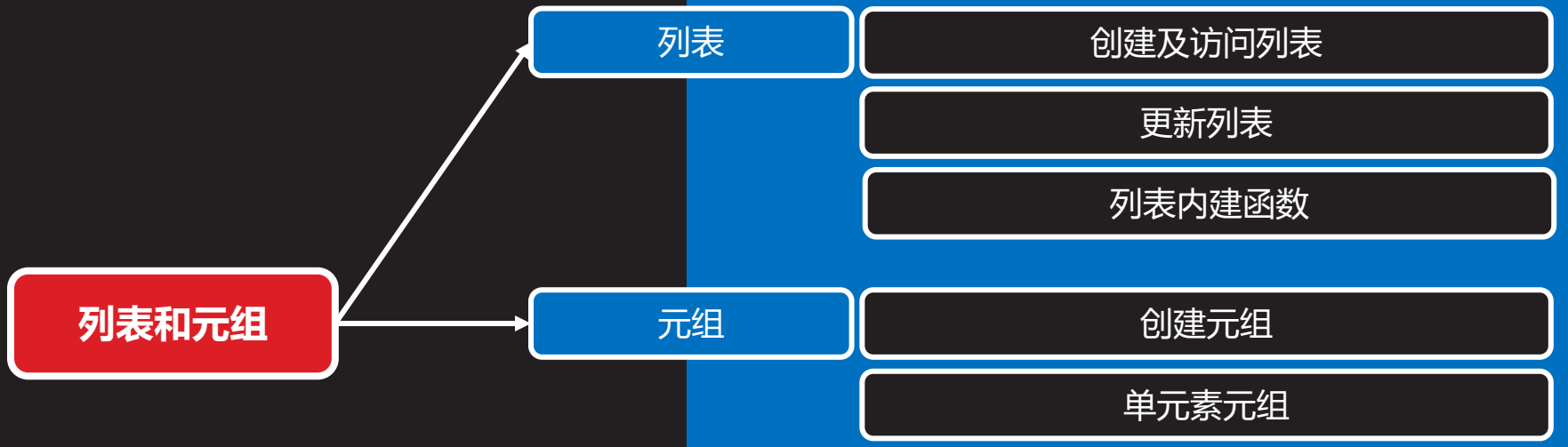
NSD PYTHON1

DAY04

内容

上午	09:00 ~ 09:30	作业讲解和回顾
	09:30 ~ 10:20	列表和元组
	10:30 ~ 11:20	
	11:30 ~ 12:00	字典
下午	14:00 ~ 14:50	
	15:00 ~ 15:50	集合
	16:10 ~ 17:00	
	17:10 ~ 18:00	总结和答疑

列表和元组



Tedu.cn
达内教育

列表

创建及访问列表

- 列表是有序、可变的数据类型
- 列表中可以包含不同类型的对象
- 列表可以由[]或工厂函数创建
- 支持下标及切片操作

知识讲解

更新列表

- 通过下标只能更新值，不能使用标添加新值

```
>>> alist = [10, 35, 20, 80]
>>> alist[-1] = 100
>>> alist[1:3] = [30, 50, 80]
```

知识讲解

列表内建函数

知识讲解

列表方法	操　作
list.append(obj)	向列表中添加一个对象obj
list.count(obj)	返回一个对象obj 在列表中出现的次数
list.extend(seq)	把序列seq的内容添加到列表中
list.index(obj)	返回obj对象的下标
list.insert(index, obj)	在索引量为index 的位置插入对象obj
list.reverse()	原地翻转列表
list.sort()	排序

Tedu.cn
达内教育

元组

创建元组

- 通过()或工厂函数tuple()创建元组
- 元组是有序的、不可变类型
- 与列表类似，作用于列表的操作，绝大数也可以作用于元组

知识讲解

单元素元组

- 如果一个元组中只有一个元素，那么创建该元组的时候，需要加上一个逗号

```
>>> atuple = ('hello')
>>> print(atuple)
hello
>>> type(atuple)
<class 'str'>
>>> atuple = ('hello',)
>>> print(atuple)
('hello',)
>>> type(atuple)
<class 'tuple'>
```

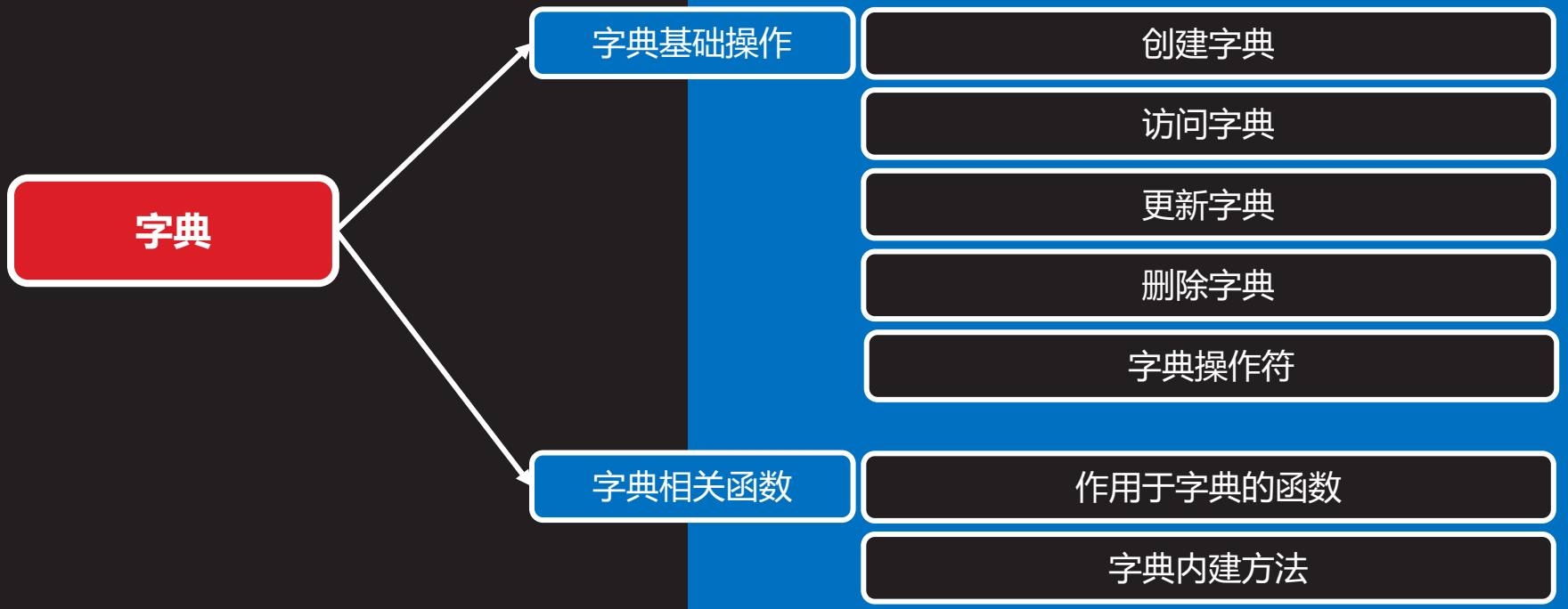
知识讲解

案例1：用列表构建栈结构

1. 栈是一个后进先出的结构
2. 编写一个程序，用列表实现栈结构
3. 需要支持压栈、出栈、查询功能

课堂练习

字典



Tedu.cn
达内教育

字典基础操作

创建字典

- 通过{ }操作符创建字典
- 通过dict()工厂方法创建字典
- 通过fromkeys()创建具有相同值的默认字典

```
>>> adict = {'name':'bob', 'age':23}
>>> bdict = dict((['name', 'bob'], ['age', 23]))
>>> print(bdict)
{'age': 23, 'name': 'bob’}
>>> cdict = {}.fromkeys(('bob', 'alice'), 23)
>>> print(cdict)
{'bob': 23, 'alice': 23}
```

知识讲解

访问字典

- 字典是映射类型，意味着它没有下标，访问字典中的值需要使用相应的键

知识讲解

```
>>> for each_key in adict:
...         print 'key=%s, value=%s' % (each_key, adict[each_key])
key=age, value=23
key=name, value=bob

>>> print('%(name)s' % adict)
bob
```

更新字典

- 通过键更新字典
  - 如果字典中有该键，则更新相关值
  - 如果字典中没有该键，则向字典中添加新值

```
>>> print adict
{'age': 23, 'name': 'bob'}

>>> adict['age'] = 22
>>> print(adict)
{'age': 22, 'name': 'bob'}

>>> adict['email'] = 'bob@tarena.com.cn'
>>> print adict
{'age': 22, 'name': 'bob', 'email': 'bob@tarena.com.cn'}
```

知识讲解

删除字典

- 通过del可以删除字典中的元素或整个字典
- 使用内部方法clear()可以清空字典
- 使用pop()方法可以“弹出”字典中的元素

```
>>> del adict['email’]
>>> print(adict)
{'age': 22, 'name': 'bob’}
>>> adict.pop('age’)
22
>>> print(adict)
{'name': 'bob’}
>>> adict.clear()
>>> print(aDict)
{}
```

知识讲解

字典操作符

- 使用字典键查找操作符[ ]，查找键所对应的值
- 使用in和not in判断键是否存在于字典中

```
>>> adict = {'age': 23, 'name': 'bob'}
>>> print(adict['name’])
Bob

>>> 'bob' in adict
False

>>> 'name' in adict
True
```

知识讲解

Tedu.cn
达内教育

字典相关函数

作用于字典的函数

- len()：返回字典中元素的数目
- hash()：本身不是为字典设计的，但是可以判断某个对象是否可以作为字典的键

```
>>> print(adict)
{'age': 23, 'name': 'bob’}
>>> print(len(adict))
2
>>> hash(3)
3
>>> hash([])
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unhashable type: 'list’
```

知识讲解

字典内建方法

- dict.copy()：返回字典（深复制）的一个副本

```
>>> print(adict)
{'age': 23, 'name': 'bob'}

>>> bdict = adict.copy()
>>> bdict['name'] = 'alice’
>>> print(adict)
{'age': 23, 'name': 'bob'}

>>> print(bdict)
{'age': 23, 'name': 'alice’}
```

知识讲解

字典内建方法（续1）

- dict.get(key, default=None)：对字典dict中的键key，返回它对应的值value，如果字典中不存在此键，则返回default的值

dict.get(key, default=None)：对字典dict中的键key，返回它对应的值value，如果字典中不存在此键，则返回default的值

知识讲解

字典内建方法（续2）

- dict.setdefault(key, default=None)：如果字典中不存在key键，由dict[key]=default为它赋值

```
>>> print(adict)
{'age': 23, 'name': 'bob’}
>>> adict.setdefault('age', 20)
23
>>> print(adict)
{'age': 23, 'name': 'bob’}
>>> adict.setdefault('phone', '15033448899’)
'15033448899’
>>> print(adict)
{'phone': '15033448899', 'age': 23, 'name': 'bob’}
```

知识讲解

字典内建方法（续3）

- dict.items()：返回一个包含字典中(键，值)对元组的列表
- dict.keys()：返回一个包含字典中键的列表
- dict.values()：返回一个包含字典中所有值的列表
- dict.update(dict2)：将字典dict2的键-值对添加到字典dict

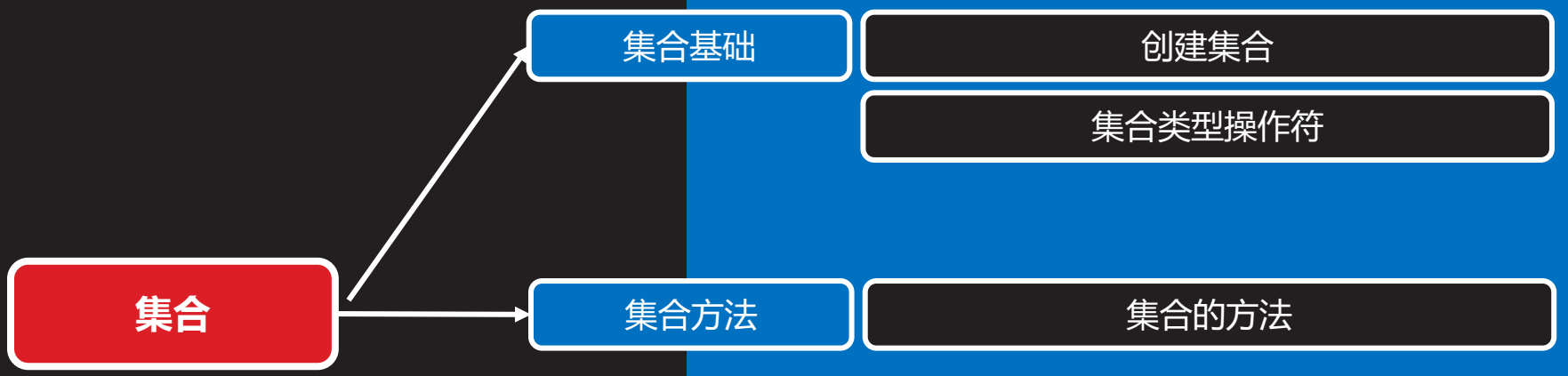
知识讲解

案例2：模拟用户登陆信息系统

1. 支持新用户注册，新用户名和密码注册到字典中
2. 支持老用户登陆，用户名和密码正确提示登陆成功
3. 主程序通过循环询问进行何种操作，根据用户的选择，执行注册或是登陆操作

课堂练习

集合



Tedu.cn
达内教育

集合基础

创建集合

知识讲解

- 数学上，把set称做由不同的元素组成的集合，集合（set）的成员通常被称做集合元素
- 集合对象是一组无序排列的可哈希的值
- 集合有两种类型
  - 可变集合set
  - 不可变集合frozenset

```
>>> s1 = set('hello')
>>> s2 = frozenset('hello')
>>> s1
{'l', 'e', 'o', 'h'}
>>> s2
frozenset({'l', 'e', 'o', 'h'})
```

集合类型操作符

- 集合支持用in和not in操作符检查成员
- 能够通过len()检查集合大小
- 能够使用for迭代集合成员
- 不能取切片，没有键

```
>>> len(s1)
4
>>> for ch in s1:
...   print(ch)
l
e
o
h
```

知识讲解

集合类型操作符（续1）

- |：联合，取并集
- &：交集
- -：差补

```
>>> s1 = set('abc')
>>> s2 = set('cde')
>>> s1 | s2
{'e', 'd', 'b', 'a', 'c'}
>>> s1 & s2
{'c'}
>>> s1 - s2
{'b', 'a'}
```

知识讲解

Tedu.cn
达内教育

集合方法

集合的内建方法

- set.add()：添加成员
- set.update()：批量添加成员
- set.remove()：移除成员

```
>>> s1 = set('hello')
>>> s1.add('new')
>>> s1
{'h', 'o', 'l', 'e', 'new'}
>>> s1.update('new')
>>> s1
{'h', 'o', 'l', 'w', 'e', 'new', 'n'}
>>> s1.remove('n')
>>> s1
{'h', 'o', 'l', 'w', 'e', 'new'}
```

知识讲解

集合的内建方法（续1）

- s.issubset(t)：如果s是t的子集，则返回True,否则返回False
- s.issuperset(t)：如果t是s的超集，则返回True,否则返回False
- s.union(t)：返回一个新集合，该集合是s和t的并集
- s.intersection(t)：返回一个新集合，该集合是s和t的交集
- s.difference(t)：返回一个新集合，该集合是s的成员，但不是t的成员

知识讲解

案例3：比较文件内容

1. 有两个文件：a.log和b.log
2. 两个文件中有大量重复内容
3. 取出只有在b.log中存在的行

课堂练习

Tedu.cn
达内教育

总结和答疑