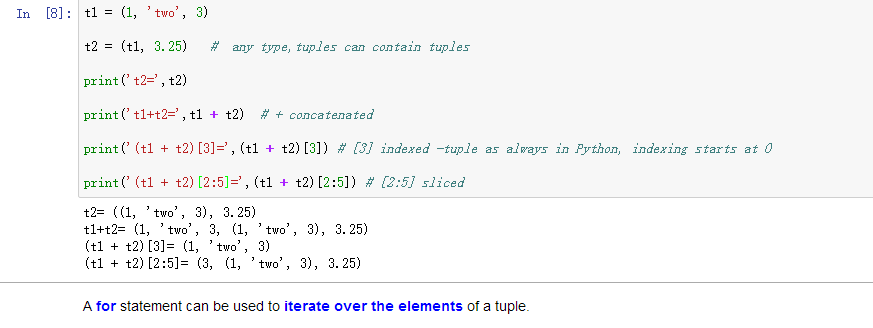
软件重点

5.1 Tuples

有序的

表示 单个 /一个元素的元组后面加逗号

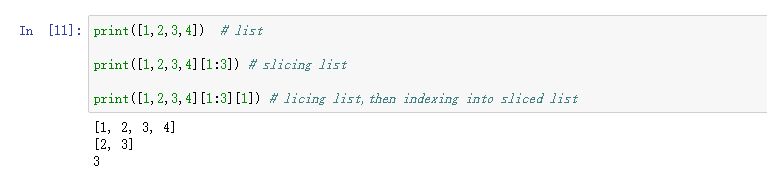


2：5表示从第二个 开始 有5-2个

考试考sliced 分片必考

列表（必考）（5.2 LIST）

没有孤立的语法题 都是放在程序里考



记住每个都是从0开始





不考：

## **5.3 Functions as Objects**

In Python, functions are first-class objects

In [3]:



type(abs)

Out[3]:

builtin\_function\_or\_method

In [5]:



type(removeDups)

---------------------------------------------------------------------------NameError Traceback (most recent call last)<ipython-input-5-b71a1bc5e531> in <module>()----> 1 type(removeDups)

NameError: name 'removeDups' is not defined

Using **functions as arguments** can be particularly convenient in conjunction with lists.

It allows a style of coding called **higher-order programming**

In [6]:



*#Page 64, Figure 5.5*

​

**from** functionsFromChapter4 **import** **\***

​

**def** applyToEach(L, func):

"""Assumes L is a list, func a function

Mutates L by replacing each element, e, of L by f(e)"""

**for** i **in** range(len(L)):

L[i] = func(L[i])

L = [1, **-**2, 3.33]

print('L =', L)

print('\nApply abs to each element of L.')

​

applyToEach(L, abs)

​

print('L =', L)

​

print('\nApply int to each element of', L)

​

applyToEach(L, int)

​

print('L =', L)

​

print('\nApply factorial to each element of', L)

​

applyToEach(L, factR)

​

print('L =', L)

​

print('\nApply Fibonnaci to each element of', L)

​

applyToEach(L, fib)

​

print('L =', L)

L = [1, -2, 3.33]

Apply abs to each element of L.

L = [1, 2, 3.33]

Apply int to each element of [1, 2, 3.33]

L = [1, 2, 3]

Apply factorial to each element of [1, 2, 3]

L = [1, 2, 6]

Apply Fibonnaci to each element of [1, 2, 6]

L = [1, 2, 13]



​

#### **Python has a built-in higher-order function**

**map**

1 Its **simplest form** ：

the first argument to map is **a unary function**

a function that has only **one parameter**

the second argument is any ordered collection of values suitable as arguments to the first argument.

In [7]:



list(map(factR, [1, 2, 3]))

Out[7]:

[1, 2, 6]

In [8]:



l=[]

**for** i **in** [1,2,3]:

l.append(factR(i))

l

Out[8]:

[1, 2, 6]

2 More generally,

the first argument to map can be of function of **n arguments**,

in which case it must be followed by **n subsequent ordered collections**

In [9]:



help(min)

Help on built-in function min in module builtins:

min(...)

min(iterable, \*[, default=obj, key=func]) -> value

min(arg1, arg2, \*args, \*[, key=func]) -> value

With a single iterable argument, return its smallest item. The

default keyword-only argument specifies an object to return if

the provided iterable is empty.

With two or more arguments, return the smallest argument.

In [10]:



*#Page 64*

L1 = [1, 28, 36]

L2 = [2, 57, 9]

​

print(list(map(min, L1, L2))) *# min*

[1, 28, 9]

In [11]:



L1 = [1, 28, 36]

L2 = [2, 57, 9]

lmin=[]

**for** i **in** range(3):

lmin.append(min(L1[i],L2[i]))

print(lmin)

[1, 28, 9]

Python Library **CHAPTER TWO: BUILT-IN FUNCTIONS**

**Page13:**

**map(function, iterable, ...)**

Return an iterator that applies function to every item of iterable, yielding the results. If additional iterable arguments are passed, function must take that many arguments and is applied to the items from all iterables in parallel. With multiple iterables, the iterator stops when the shortest iterable is exhausted.

In [12]:



help(map)

Help on class map in module builtins:

class map(object)

| map(func, \*iterables) --> map object

|

| Make an iterator that computes the function using arguments from

| each of the iterables. Stops when the shortest iterable is exhausted.

|

| Methods defined here:

|

| \_\_getattribute\_\_(self, name, /)

| Return getattr(self, name).

|

| \_\_iter\_\_(self, /)

| Implement iter(self).

|

| \_\_new\_\_(\*args, \*\*kwargs) from builtins.type

| Create and return a new object. See help(type) for accurate signature.

|

| \_\_next\_\_(self, /)

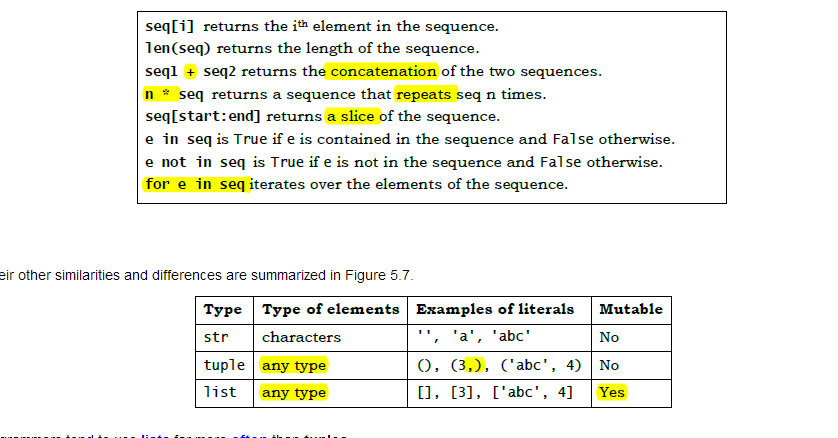
| Implement next(self).

|

| \_\_reduce\_\_(...)

| Return state information for pickling.

重点：



S.slid必考

字典 必考（5.5）

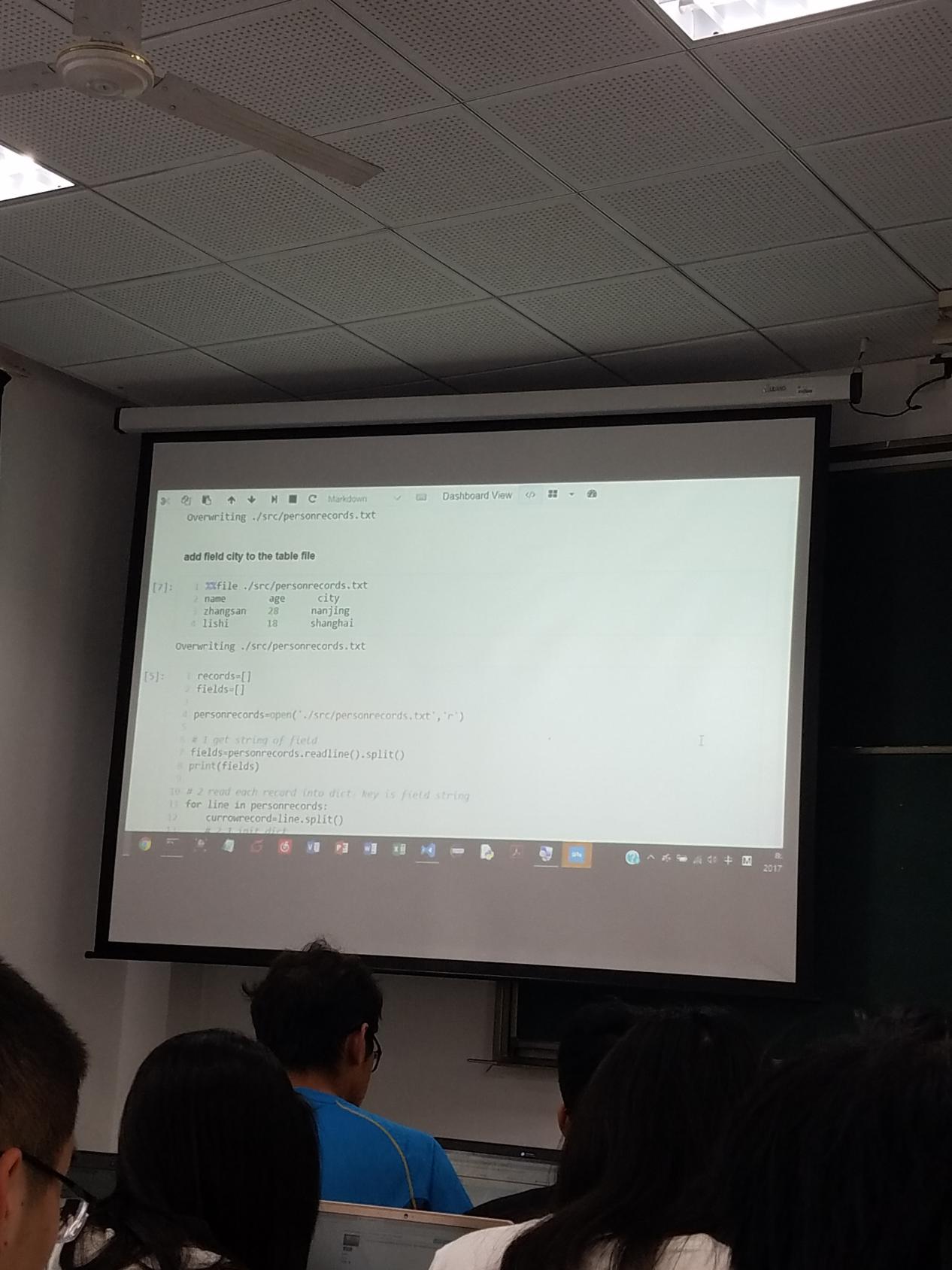


KEY不能是可变类型

字典的三个（？）：

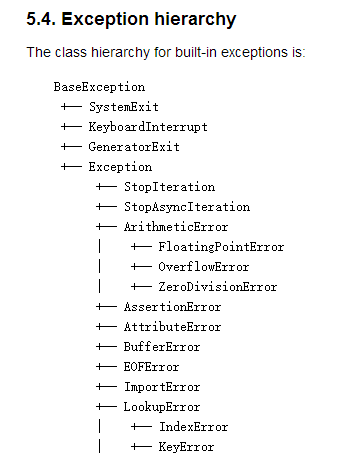
1. { }
2. Leg：value
3. ,

必考：列表和字典的组合



1. ：

给出错误要知道错误在哪里



必考：异常处理

必考：devtools-unittest

第六章

DEBUG