Python的数据类型

本节课内容

- 数据类型由来
- 为什么会有多种数据类型
- 初次见面-Python中的数据类型

数据类型的由来

0，1

控制电脑

为什么会有多种数据类型

◆ 为了可以适应更多的使用场景，将数据划分为多种类型，每种类型都有着各自的特点和使用场景，帮助计算机高效的处理与展示数据

初次见面-Python中的数据类型

数字类型

字符串类型

布尔类型

空类型

列表类型

元组类型

字典类型

初识数字类型

本节课内容

◆整型 int

◆浮点型 float

◆内置函数--type

整型

- 整型就是我们说的整数，0 也是整数，但是特殊的整数
- int 既是 整型的代表，又是定义整型的内置函数
- 定义一个整型，并不一定非要使用int


```
count_100_01 = int(100)
count_100_02 = 100
```

```
100 100
```

Python2 中曾经有 long 整型，在python3中已经弃用了

浮点型

◆ 浮点型 就是我们生活中的小数，凡是带有小数点的类型，都可以认为是浮点类型

慕课网春节放假时间为：1.23-2.02

◆在python中， float 既是 浮点型的代表， 又是浮点类型定义的内置函数

```
pi_01 = float(3.14)
pi_02 = 3.14
```

◆ 定义float类型的时候，并不需要一定使用float来声明

内置函数--type

- 返回变量的类型
- type(已经被赋值的变量名或变量)

```
count = 1050
print(type(count))
print(type(3.1415926))
```

```
<class 'int'>
<class 'float'>
```

数字类型的简单应用

本节课内容

- 对int 与 float的简单应用练习---初中生春游，主人公小慕

起因

- 小慕学校组织春游，他的班级学生共51人，男生28人，女生23人
- 每人缴费35.5元，并且根据最后使用的情况多退少补

经过

- 他们早上8点出发，每个大巴可以坐30人，所以需要2辆大巴
- 上午10点33分到达公园开始游玩
- 中午12点开始吃饭，伙食费是25.5元
- 到下午3点05分时集体离开公园，坐大巴回去，来回大巴费用是5元

结果

- 到晚上5点回到学校，每人退回5元钱

我们的任务

- 把这一天有关整型，浮点型的信息 按照先后顺序打印出来
- 定义整型与浮点型，并熟练赋值语句与print，type的使用

初识字符串

本节课内容

- 什么是字符串
- 字符串的内置函数与定义方法
- 字符串的重要思想
- Python的内置函数 id
- Python的内置函数 len

什么是字符串

全站结果（16）　课程内容（157）

全部　课程　专栏　猿问　手记　WIKI　找到相关内容 17 个

从零开始 一站搞定Python Django开发

路线专为零Django基础学员定制，层层深入，助你熟练掌握Django应用技能。

4步骤 · 4门课　★2433人收藏

Django入门到进阶-更适合Python小白的系统课程

掌握Django的基础知识，学习Web的相关扩展知识，学会开发c/s服务与apiserver服务；学习多方面非

讲师： deweizhang　初级　234　试看

热搜关键词

Vue　Python　Java　flutter　springboot

docker　React　小程序

从Java后端到全栈

浓缩7七月从工程师到CTO的10年成长历程

金职位

Web前端攻城狮

踏入IT行业从此开始

Java攻城狮

什么是字符串

- 用 ‘’ 或 “” 包裹的信息 就是字符串

- 字符串中可以包含任意字符：如字母，数字，符号，且没有先后顺序

字符串的定义方法

- 在python中，使用 str 来代表字符串类型，并且通过该函数可以定义字符串

```
safe = str('健康的体温是36.5左右')
name = '小慕'
message = 'dewei是1986年出生的!'
info = '祝愿大家身体健康'
```

```
健康的体温是36.5左右
小慕
dewei是1986年出生的!
祝愿大家身体健康
```

```
<class 'str'>
```

字符串的重要思想

- 字符串不可改变!
- name = ‘dewei’

```
data_01 = '我对他说:"你好呀"'
data_02 = "我对他说:'你好呀'"
```

内置函数id

- 返回变量的内存地址
- 数字地址 = id(变量)

```
In [1]: name = 'dewei'

In [2]: id(name)
Out[2]: 4358118128

In [3]: name = 'xiaomu'

In [4]: id(name)
Out[4]: 4358761968
```

内置函数len

- 返回 字符串的长度
- 无法返回数字类型的长度，因为数字类型没有长度
- 返回值 = len(字符串)

```
length = len('python是一门很好的语言')
print(length)  14
```

count = len(3.14) ✖

字符串的简单操作

本节课内容

- 内置成员运算符 in 的使用
- 内置函数 max
- 内置函数 min
- 字符串的叠加

内置成员运算符 in 的使用

- 成员运算符是用来判断你的数据中是否存在你想要的成员

‘开发’ in ‘从零开始 一战搞定 python django开发’ -> True(真) or False(假)

in 左右是一个空格

not in

内置函数max

- max函数返回数据中最大的成员
- max（数据）-> 成员值
- 中文符号 > 字母 > 数字 > 英文符号
- 中文按照拼音的首字母来计算

print(max('今天是1月3日!　　月'))？

内置函数min

- min函数返回数据中最小的成员
- min（数据）-> 成员值
- 中文符号 > 字母 > 数字 > 英文符号
- 中文按照拼音的首字母来计算

print(min('今天是1月3日!'))？　　！

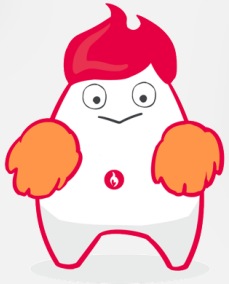
字符串的累加

- 字符串不是数字不能做减法，乘除法
- 字符串的拼接，用 “+”这个符号

```
a = ’123’  id(a) ->
107898032
b = ’456’
c = a + b
print(c) -> ’123456’
```

```
a = a + b
print(a) -> ’123456’  id(a)
80202416
```

布尔类型与空类型



本节课内容

- 什么是布尔类型，布尔类型的固定值
- 布尔类型的使用场景
- 布尔函数的使用
- 数字，字符串在布尔类型上的应用（内置函数 bool）
- 空类型 None

布尔类型

- 定义：真假的判断 即 布尔类型
- 固定值： True -> 真； False -> 假；
- 布尔值

布尔函数的使用

- bool 代表布尔类型 也可以对于结果进行真假的判断

```
In [1]: res = bool('name' in 'my name is xiaomu')

In [2]: print(res)
True
```

使用场景

- 常被用来判断一件事儿的真假
- dewei是个男生 √
- dewei不喜欢Python X

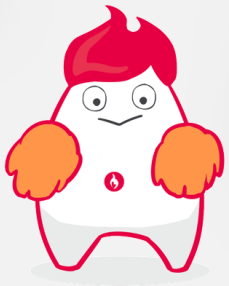
数字与字符串的布尔应用

- int 0 -> False, 非0 -> True
- float 0.0 -> False, 非0.0 -> True
- str ‘’ -> False （即 空字符串）， 非空字符串 -> True
- 在计算机中 0 1是计算机的最原始形态，单个占空间也最小，故而经常会将 0 1 用来替代 True 与 False

Python中的空类型

- 不属于任何数据类型 就是 空类型
- 固定值：None
- 空类型 属于 False 的范畴
- 如果不确定类型的时候 可以使用空类型

初识列表类型



本节课内容

- 什么是列表
- 列表的定义
- 列表中的类型
- in, max, min 在列表中的使用

什么是列表

- 列表就是队列
- 他是各种数据类型的集合，也是一种数据结构
- 列表是一种有序，且内容可重复的集合类型

列表的定义

- 在Python中， list 代表着 列表这种类型，也可以用它定义一个列表
- 在Python中，列表中的元素存在于一个 [] 中

```
In [1]: names_01 = list(['dewei', '小慕', 'dewei'])

In [2]: names_02 = ['dewei', '小慕', 'dewei']

In [3]: print(type(names_01))
<class 'list'>
```

- 在Python中，列表是一个无限制长度的数据结构

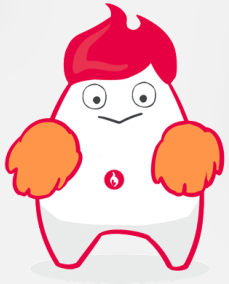
列表中的类型

- str_array = [’ dewei’ , ’ haha’ , ’ ’ , ’ ’]
- int_array = [1, 2, 3, 0, 10, 110]
- float_array = [1.1, 10.3, 0.1, 0.0, 3.1415926]
- bool_array = [True, False, False, True]
- none_array = [None, None, None]
- list_array = [[1,2,3], [1.2, 3.1]]
- mix_array = [’ dewei’ , 1, 3.14, None, True]

in, max, min 在列表中的使用

- 1 in [1，2，3，4] -> True; 10 in [1,2,3,4] -> False
- max([1,2,3,4]) -> 4
- min([1,2,3,4]) -> 1
- max 和 min 在 列表中使用的时候，列表中的元素不能是多种类型，如果类型不统一，则会报错

初识元组类型



本节课内容

- 什么是元组
- 元组的创建
- 元组与列表关系
- in, max, min 在元组中的使用

什么是元组

- 元组与列表一样，都是一种可以存储多种数据结构的队列
- 元组也是一个有序的，且元素可以重复的集合

元组的定义

◆在python中， tuple 代表着元组这种类型，也可以用它定义一个元组

◆在python中，元组中的元素存在于一个 () 小括号中

```
names_01 = tuple(('dewei', '小慕'))
names_02 = ('dewei', '小慕')
names_03 = ('dewei',)
```

◆ 在python中，元组是一个无限制长度的数据结构

列表与元组的区别

- 元组比列表占用资源更小
- 列表是可变的，元组是不可变的

```
a = (1,2,3)    id-> 116294080
b = (5,6,7)
a = a + b
print(a) -> (1,2,3,5,6,7)  id -> 110060064
```

元组中的类型

- str_tuple = （’ dewei’ , ’ haha’ , ’ ’ , ’ ’ ）
- int_tuple = （1, 2, 3, 0, 10, 110）
- float_tuple = （1.1, 10.3, 0.1, 0.0, 3.1415926）
- bool_tuple = （True, False, False, True）

元组中的类型

- none_tuple = （None, None, None）
- tuple_tuple = （（1,2,3），（1.2, 3.1））
- list_typle = ([123,456], [6789,1234])
- mix_tuple = （’dewei’, 1, 3.14, None, True）

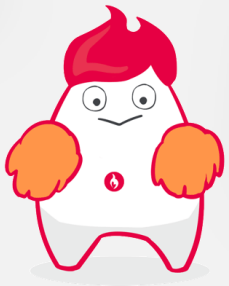
tuple_array = [(’a’, ’b’), (’c’, ’d’), (’e’,)]

in, max, min 在元组中的使用

- 1 in (1，2，3，4) -> True; 10 in (1,2,3,4) -> False
- max((1,2,3,4)) -> 4
- min((1,2,3,4)) -> 1
- max 和 min 在 元组中使用的时候，元组中的元素不能是多种类型

  如果类型不统一，则会报错

初识字典类型

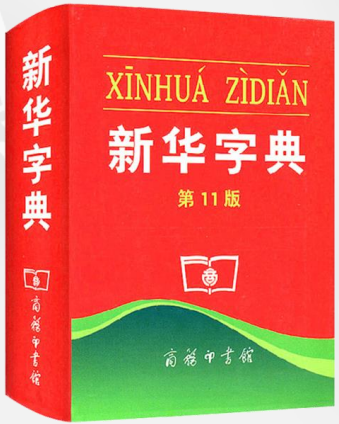


本节课内容

- 什么是字典
- 字典的结构与创建方法
- 字典支持的数据类型
- 列表与元组中的字典
- Python3.7与之前版本 字典的区别

什么是字典

- 字典是由多个键（key）及其对应的值（value）所组成的一种数据类型



字典的结构与创建方法

- 在python中，dict 用来代表字典，并且可以创建一个字典
- 在python中，通过{}将一个个key 与 value 存入字典中

```
a = dict()
a = {}

person = {’ name’ : ’ dewei’ , ’ age’ : 33}
```

字典支持的数据类型

- key 支持 字符串，数字和元组类型，但列表是不支持的
- value 支持所有python的数据类型

```
a = {’name’: ’dewei’, ’age’: 30}
b = {1: ’one’, 2: ’two’}
c = {(1,2,3): [1,2,3], (4,5,6): [4,5,6]}
```

列表与元组中的字典

- dict_array = [{1:1, 2:2}, { ‘one’ : 1, ‘two’ : 2}]
- dict_tuple = ({1:1, 2:2}, { ‘one’ : 1, ‘two’ : 2})
- 元组一旦创建，就不可改变

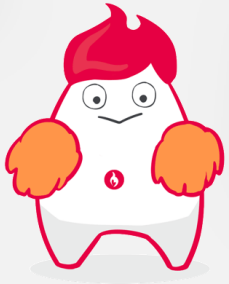
Python3.7与之前版本字典的区别

- 无序
- 有序

字典的重要特性

- 字典中每一个key一定是唯一的

数字的运算



本节课内容

- 赋值运算符有哪些
- 小练习 b kb mb gb 的转换
- 字符串与数字的乘法

赋值运算符有哪些

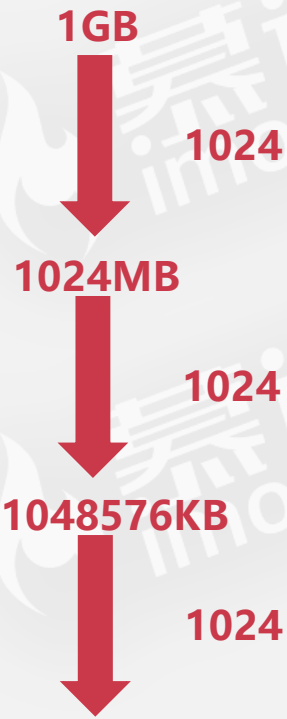
运算符	描述	举例
=	等于运算符	c = a + b
+=	加法运算符	c += a -> c = c + a
-=	减法运算符	c -= a -> c = c – a
*=	乘法运算符	c *= a -> c = c * a
/=	除法运算符	c /= a -> c = c / a
%=	取模运算符	c %=a -> c = c % a
**=	幂运算符	c **= a -> c = c ** a
//=	整除运算符	c //= a -> c = c // a

b kb mb gb 的转换

◆b kb mb gb 是计算机的计量单位

◆1024相差量

gb = 1 -> b ?



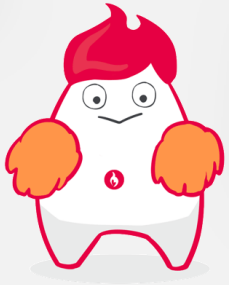
字符串与数字的乘法

- 字符串无法与字符串做乘法
- 字符串只可以和数字作乘法

```
name = 'xiaomu'
print(name * 3)
>> 'xiaomuxiaomuxiaomu'
```

列表和元组和字典可以做乘法吗?

数字的比较运算



本节课内容

◆比较与身份运算符有哪些

比较与身份运算符有哪些

运算符	描述	举例
==	判断是否等于	a == b
! =	判断是否不等于	a != b
>	判断是否大于	a > b
<	判断是否小于	a < b
>=	判断是否大于等于	a >= b
<=	判断是否小于等于	a <= b
<>	判断是否不等于	a <> b
is	判断两个对象存储单元是否相同	a is b
is not	判断两个对象存储单元是否不同	a is not b

<>在python3里已经废弃，仅python2里可用

单元存储 就是我们提过的内存块