用Python做些事

02-变量这小东西

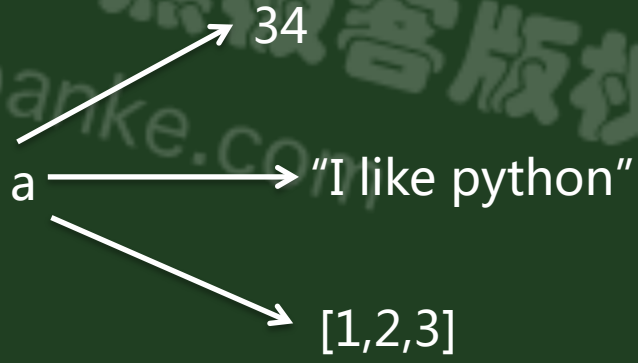
变量这小东西

- 和C++,JAVA的区别
- 数字
- 字符串
- 日期和时间
- 列表
- 元组
- 字典
- 文件

和C++, JAVA的区别

动态类型

```
a=34, 3.4, 2**1000
a=  “I like python”
a=[1,2,3]
type(a)
```



和C++，JAVA的区别

一切皆对象

对象是类的实例化

封装，继承，多态

汽车 → 红旗汽车 → 红旗轿车 → 红旗H7轿车 → 小李家买的红旗H7轿车

红旗汽车 → 红旗SUV → 红旗L5轿车 → 习总的红旗L5轿车

红旗L5轿车 → 小王家买的红旗L5轿车

汽车：
长,宽,高
品牌，生产国
轮子, 发动机
启动()，停()，拐弯()

红旗L5轿车：
5555,2018,1578
红旗，中国
玲珑轮胎，一汽发动机
启动()，停()，拐弯()

变量这小东西

- ☑ ~~和C++, JAVA的区别~~
- ☐ 数字
- ☐ 字符串
- ☐ 日期和时间
- ☐ 列表
- ☐ 元组
- ☐ 字典
- ☐ 文件

数字

自动转换类型

```
a=34
a=3.14
a=2**1000
//
```

```
0.3*3
0.3/3
1/ 2**10000
```

decimal

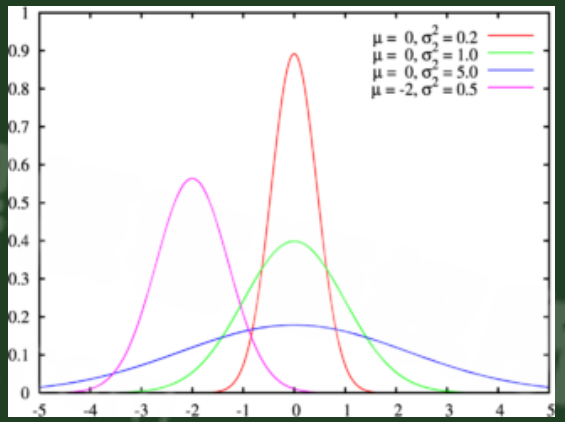
```
from decimal import Decimal as D
D(‘0.3’)/D(3)
D(1)/D(2**10000)
```

```
python -mtimeit -s "from decimal import Decimal as D"
"D('1.2')+D('3.4')"
```

数字

常用库

```
import math
math.pi
math.sqrt(80)
math.log10(2**1000)
math.pow(x,y)
math.factorial(x)
```




```
import random
random.random()
random.choice([1,2,3,4])
random.randint(a,b)
random.uniform(a,b)
random.gauss(mu,lamda)
```

数字

numpy

产生数组或矩阵，正态分布的随机数
矩阵运算
矩阵求逆，转置等操作

scipy

拟合，线性插值, 样条插值
积分，微分
解非线性方程
滤波器设计

变量这小东西

- ☑ ~~和C++, JAVA的区别~~
- ☑ ~~数字~~
- ☐ 字符串
- ☐ 日期和时间
- ☐ 列表
- ☐ 元组
- ☐ 字典
- ☐ 文件

字符串

切片，索引

```
s= “use python do something”
s[1], s[-1], s[1:3], s[1:6:2], s[1:], s[:-1], s[:]
split, join, [start : stop : step]
```

常用方法集合

```
 “let us ” +s, s*2
 s.upper()
 s.find('pa')
 s.replace('python','java')

 print "%s like %s" %('we','python')
```

字符串

转义r’’

s="C:\newpython"，print s, len(s)
r’\n’前缀字符串，不考虑转义
s=r"C:\newpython”

“”
“let’s have a fun”
“”“

Unicode u’’

ASCII, 128
ISO Latin, 256
Unicode, 2 bytes
utf8
gb2312
gbk

字符串-re模块

import re

Regular expression

re.match(p, text)

re.search(p, text)

re.findall(p, text)

re.split(p, text)

re.sub(p,s, text)

pattern = re.compile(p)

results = pattern.match(text)

字符串-正则表达式

整体介绍

11个元字符，\，^，$，.，|，?，*，+，()，[]，{}，

特殊含义，\，.，

可选，|，[]

重复，*，+，?，{}，（贪婪模式）

6个字符类，\d，\D，\s，\S，\w，\W

4个位置类，\b，\B，\A，\Z，（^，$）

分组，()

编译选项，I，L，M，S，U，X

字符串-正则表达式

re.compile(p, re.VERBOSE)

编译选项

I (IGNORECASE)	忽略大小写
L (LOCALE)	使预定字符类 \w \W \b \B \s \S 取决于当前区域设定
M (MULTILINE)	多行模式，改变'^'和'$'的行为
S (DOTALL)	点任意匹配模式，改变'.'的行为
U (UNICODE)	使预定字符类 \w \W \b \B \s \S \d \D 取决于unicode定义的字符属性
X (VERBOSE)	详细模式。这个模式下正则表达式可以是多行，忽略空白字符，并可以加入注释

变量这小东西

- ☑ ~~和C++, JAVA的区别~~
- ☑ ~~数字~~
- ☑ ~~字符串~~
- ☐ 日期和时间
- ☐ 列表
- ☐ 元组
- ☐ 字典
- ☐ 文件

日期和时间

datetime

日期：Datetime.date.today()
日期和时间：datetime.datetime.now()
1000天之后是哪一天：datetime.timedelta(days=1)
打印格式的问题：Isoformat(), strftime(),
字符串转换：strptime()

time

Datetime.time(12,11,30)

Time.time() # 实际时间
Time.clock() #cpu时间
Time.sleep() #以s为单位

变量这小东西

- ☑ ~~和C++, JAVA的区别~~
- ☑ ~~数字~~
- ☑ ~~字符串~~
- ☑ ~~日期和时间~~
- ☐ 列表
- ☐ 元组
- ☐ 字典
- ☐ 文件

列表

切片，索引，引用

```
a=[1,2,3]
a_ref=a
a[2]=100
```

常用操作

```
a_copy
a.append(300)
a.insert(1,50)
a.pop()
a.sort()
a.reverse()
del a[1]
```

嵌套，多种类型并存

```
b=[a,a_ref, a_copy]
c=[1,2, '123' , 'abc' ]
```

+, *

Count(val)，对某个元素计数

变量这小东西

- ☑ ~~和C++, JAVA的区别~~
- ☑ ~~数字~~
- ☑ ~~字符串~~
- ☑ ~~日期和时间~~
- ☑ ~~列表~~
- ☐ 元组
- ☐ 字典
- ☐ 文件

元组

不可变的列表

(a,b,c)

不能原处修改

常用操作

index
count，对某个元素计数
+，*
嵌套：可嵌套可变的list
转换：tuple()

变量这小东西

- [x] ~~和C++, JAVA的区别~~
- [x] ~~数字~~
- [x] ~~字符串~~
- [x] ~~日期和时间~~
- [x] ~~列表~~
- [x] ~~元组~~
- [ ] 字典
- [ ] 文件

字典

Key-Value

dict = {'xiaoming':90,'xiaohong':80,'xiaomao':60,'xiaoli':54}

Dict, zip

常用操作

Keys, values
Get
Update
Del
Clear
嵌套

散列表，没有顺序，适合插入，查询操作

Key不一定是字符串，但一定是不可变对象

排序
[(k,dict[k]) for k in sorted(dict.keys())]

sorted(dict.iteritems(),key=lambda d:d[1], reverse=True)

字典

再谈引用和拷贝

引用
L=[4,5,6]
X=L*4, Y=[L]*4
L[1]=0
print X, Y

浅拷贝
字典D.copy(), copy.copy(D)
列表L[:]

深拷贝
copy.deepcopy(D)

变量这小东西

- ☑ ~~和C++, JAVA的区别~~
- ☑ ~~数字~~
- ☑ ~~字符串~~
- ☑ ~~日期和时间~~
- ☑ ~~列表~~
- ☑ ~~元组~~
- ☑ ~~字典~~
- ☐ 文件

文件

常用操作

F=open(path, ‘r’)返回对象为file-like object
还可以是内存，网络等，r，w，a

F.read()
F.readline()
F.write()
F.close()

中文支持
import codecs
f=codecs.open(filename, mode, encoding)

文件操作
import os
os.path.exists(filename)
os.rename(old, new)

文件

Shelve库

```
Import shelve

D = Shelve.open(file)
D[ ‘name’ ]= ‘heibanke’
D.close()
```

Pickle/cPickle库

```
Import cPickle
f= open(file, mode)
cPickle.dump(obj, f)
Obj = cPickle.load(f)
```

THANKS

?

作业

- 2.1 编写验证email的正则表达式。邮箱名可以是英文字母或数字或 -, _符号，邮箱后缀网址名可以是英文字母或数字，域名可以是com, org, edu
chu-tian-shu_1981@heibanke2015.com

- 2.2 利用随机函数产生一个用户的用户名，密码，并利用文件将用户名和密码保存下来。

- 2.3 上面的文件中密码没有加密，不安全，请将文件内容读出后将密码字段通过md5的库处理后，再保存至另一个文件。

md5加密处理库
```
import hashlib
hashlib.md5(password).hexdigest()
```

作业

- 2.4 公交车数据读取，并存入字典

读取文件（中文），处理中文字符串，字典和列表操作

步骤：

读取Linenum，station两个字段，处理后，最终保存结果：

{

“1(马官营-四惠站)”：[马官营,六里桥北里,。。。,四惠站],

”1(四惠站-马官营)”：[四惠站,八王坟,。。。,马官营],

…}