

Python 复习整理 (源自霸霸, 荣誉出品)

变量

变量的类型

1. 整数 (int)、浮点数 (float)、字符串 (str)

类型的转换函数: int (), str (), float

()

不同类型之间的运算规则 print 运行并换行 ②捕获屏幕输

2. 变量间的运算: + 入 * / % (余数)

** (次方) 4. 字符串变量 字符串变量的假发

3. 运行 ① eg: "ttt"+"111"+str(q)

ttt111q

5. python 中的格式: 冒号, 缩进

6. 循环的语法:

① 数组 range () 函数 eg: range (0,3)

[0,1,2]

(包括起点不包括终点)

② for 循环的格式: for i in range (0,3) :

③ 累加 累乘

7. 简单的条件 (逻辑运算)

① 逻辑运算符号: > 大于 < 小于 == 等

于 >= 不小于 <= 不大于 <> 不等于

② if.....than.....else

eg: a=6

if a<=3:

print "a"

else:

print "b"

8. 函数

格式: def ()return

函数变量间的作用域

eg: def sum1(x):

a=0

for i in range(0,x):

a=a+i

return a

print a

print i

运行的结果: 7

99

第一个复杂数据类型: 列表 (list)

range () 是最常用的列表生成函数[.....]

列表的索引

1. 列表的切片

① 单个选择

eg: a=[2,5,1,4]

a[2] (注意[]内为元素的位置, 即 2 是原来 a

里面的第 0 个元素) (元素的顺序是 0,1,2,3.....)

运行结果为 0

② 连续选择

eg: a[1:3]

运行结果为[5,1](即第 1 个元素的到第 2 个元素,

第 3 个元素不包括)

③ 跳跃选择

eg: a=[2, 5, 1, 4, 3, 7, 9, 0, 8]

a[1:6:3]

运行结果为[5, 3, 0] (即第 1 个元素到第 6

个元素, 跳跃间隔 3 个运行所有)

2. 分配赋值

eg: a=[2, 5, 1, 4, 3, 7]

a[3]=9

print a

运行结果为[2, 5, 1, 9, 3, 7] (即第三个元素要

是 9, 即插入 9 在第二个元素 "1" 后面)

3. 列表的加法与乘法

[2, 3]+[1, " e"]

运行结果为[2, 3, 1, ' e']

[2, 4]*3

运行结果为[2, 4, 2, 4, 2, 4]

4. 用于 list 的内建函数

len () 求列表长度

max () 求列表最大值

min () 求列表最小值

sum () 求列表总和

mean () 求 sum () /len () 即平均数

5. 复杂数据类型的方法, 特定的方法只能作

用到特定的类型上, 列表最简单的最有用的

方法是 append ()

<改变列表本身, 不产生结

的方法

eg: a=[]

a.append(3)

print a



运行结果为[3]（定义空列表，并用 append
() 生产一个你想象的列表。

①a.sort()

此外，列表的方法还有：
运行结果为[3, 5, 7, 1, 9, 18, ' 1']
假定 a=[5, 3, 18, 9, 7, 1, 1]

从小到大的排序

②a.reverse()

print a
运行结果为[' 1' , 7, 1, 9, 18, 3, 5]

原有顺序的倒序

③a.extend([2, 3])

print a
运行结果为[5, 3, 18, 9, 7, 1, ' 1' , 2, 3]

列表后加新列表

④a.insert(2, " 4")

print a
运行结果为[5, 3, ' 4' , 9, 7, 1, ' 1']

将第 2 个元素数改为字符串' 4'

即 **替换**

⑤a.pop()

print a
运行结果为[5, 3, 18, 9, 7, 1]

删除最后一个元素

⑥a.remove(3)

print a
运行结果为[5, 18, 9, 7, 1, ' 1']
移除某个数字（若有多多个元素为该数字，则
移除第一个出现的）

⑦b=[2, 3, 4, 1, 2, 2, 3]

b.count(2)

print b
运行结果为 3
计算某个元素（此处的 2）在列表中出现的
次数

6. 列表的逻辑运算：in, not in

eg: 4 in[2, 3, 1]

运行结果为 False

（True/False 成为**布尔型变量**）

7. for 循环与列表

eg: a=[3, 2, 0, 4]

for i in a:

……（各种循环条件内容）

8. 迭代：enumerate ()

eg: b=[2, 4, 3]

for i, j in enumerate(b):
print i, j

运行结果为 0 2

1 4

2 3

9. **元组**：不可改变的列表

eg: a=(2, 1, 4, 7, 11, 3)

print a, a[2], a[1:4]

运行行为 (2, 1, 4, 7, 11, 3) 4 (1, 4, 7)

与列表的区别：**不可改变**

不能插入减少任何元素

10. 列表的嵌套——构建更复杂的数据结构
切片

①a=[[1, 4], 1, [1, 6, 2]]

print a, a[0], a[1], a[2]

结果: [[1, 4], 1, [1, 6, 2]] [1, 4] 1 [1, 6, 2]

print a[2][1]

结果: 6 （即[1, 6, 2]的第 1 个元素 6）

②a=[[0, 2], [3, 2], [5, 1]]

for i, j in a:

print i, j

结果: 0 2

3 2

5 1

③c=[2, 3, 6]

for i, j in enumerate(c):

print i, j

结果: 0 2

1 3

2 6

即将每个数字与其元素位置对应起来

11. 作为列表的字符串集成列表方法与函数

字符串的方法：split (), join (),

replace (), find ()

<不改变列表本身，但是全
于列表的方法>

eg:a=" python"

print a, a[0], a[2:5],



得 python p th o n
①spli

```

t():
C= "I am a teacher"
print c.split(" ")
得 [' I' , ' am' , ' a' , ' teacher' ]
以空格为切分, 划分为 4 个元素
②join():
d=c.split(" ")
print "+".join(d)
得 ' I+am+a+teacher'
以+为连接, 将元素连起, join 起到添加的作用
③find():
b=" 323425"
print b.find(" 4")
得 3
找到列表中某一元素出现第一次时的位置,
如若没找到输出为-1
④replace():
a=" python"
a.replace(" n" , " N" )
得' pythoN'
将 n 与 N 替换

```

第二个复杂数据结构：集合 set

1. set() 将列表转换成集合, 并且能去重!

eg: a=[2, 3, 2, 5]

①print set(a) 结果为 set([2, 3, 5])

但 print set(a)[2] 不存在, 集合没有列表的特性

②for i in set(a):

print i

结果: 2

3

5

2. 集合函数:

b=set(a)

b.add(1) 集合中增加某元素

print b

结果 set([1, 2, 3, 5])

3. a-b a|b a&b a^b

已知 a=set([2, 3, 1, 11])

b=set([3, 4, 1, 6])

print a-b 得 set([2, 11])

集合 a 中 b 没有的

print b-a 得 set([4, 6])

集合 b 中 a 没有的

print a|b 得 set([1, 2, 3, 4, 6, 11])

a, b 的并集

print a&b 得 set([1, 3])

a, b 的交集

print a^b 得 set([2, 4, 6, 11])

a, b 交集关于并集的补集

4. 集合转换为列表 使用 list()

print list(set([2, 3]))

得 [2, 3]

5. in; not in



第三个复杂数据结构：字典

（字典的定义：数学中的映射，计算机中的哈希键值型存储）

1. 字典的创建，格式，调用 `{3:1, 4:1, "u":10}`

eg: 该字体的称为字典中的键 Key

格式: 该字体的称为字典中的值 Value

```
a={2:
print a.keys()
得['u', 2, 4]
print a.values()
得[10, [3, 1], 1]
print a.items()
得[('u', 10), (2, [3, 1]), (4, 1)]
```

```
a={2:[3, 1], 4:1, "u":10}
print a
得 {'u': 10, 2: [3, 1], 4:1}
键从小到大排序(ps-字符串比整数大)
print a[4]
得 1 （即键 4 对应的值是 1）
print a[2][0]
得 3 （即键 2 对应值第 0 个元素的数字为 3）
print a["u"]
得 10 （即键"u" 对应值为 10）
print len(a)
得 3 （即字典 a 的长度）
```

注意:

```
for i in a.keys():
    print i, a[i]
```

与

```
for i, j in a.items():
    print i, j
```

运行结果都是 u 10

2 [3, 1]

4 1

<不允许一个键对多个值，必须能映射>

2. 字典的更新 update

```
eg: a={'u':10, 2: [3, 1], 4:1}
update {[7:2]}
```

```
print a
得 {'u': 10, 2: [3, 1], 4:1, 7:2}
增加 7:2 之后字典重新更新
eg: c={}
for i in range (1,4):
    c.update({i:i**2})
print c
得 {1:1, 2:4, 3:9}
```

3. dict () 与列表、元组的替换

```
a={'u':10, 2: [3, 1], 4:1}
c=a.items()
print c
得[('u', 10), (2, [3, 1]), (4, 1), (7, 2)]
print dict (c)
得 {4: 1, 2: [3, 1], 'u': 10, 7:2}
```

```
a= {2:[3, 1], 4:1, "u":10}
print a
得 {'u': 10, 2: [3, 1], 4:1}
a.update ({7:3})
print a
得 {'u': 10, 2: [3, 1], 4:1, 7:3}
a[7]=2 （键 7 对应的值赋值为 2）
print a
得 {'u': 10, 2: [3, 1], 4:1, 7:2}

a={2:3, 1:9, 1:3}
a.pop(1) （删除键为 1 的键和值）
print a
得 {2:3}
```

4. 异常处理

一个统计套路/字典中的 get 函数

字典中的 get 方法:

```
d={1:3, 4:6}
print d.get(1)
得 3 （键 1 对应值=d[1]）
print d.get(4)
得 6 （键 4 对应值=d[4]）
print d.get(2)
得 None
print d.get(2, 7)
得 7 （相当于输出 7）=d
```



```

t", "a", "a", 3, "t", 4, 5]
d={}
for i in a:
    d[i]=d.get(i,0)+1 (代表 i 的值)
print d
得['a':2,3:1,4:1,'t':2,5:1]
(适合处理大数据)

```

文字处理

常有中文字符的字符串的定义需要加“u”，代表字符串是 utf-8 编码
把一个一般的字符转换为 utf-8 字符需要用 decode 方法

```

eg: a=u "可爱的 python"
print len(a), a[0:4]
得      9 可爱的 p
print a.replace("n",u"爱")
得      可爱的 pytho 爱

```

如果已经转换成 utf-8 后，对应的方法里的中文参数也记得加“u”
未转换的，简单的替代或 split 等，不用转换成 utf-8 也可，参数也不要带“u”

```

decode 方法:
eg:b=a.decode("utf-8")
print b[0:4]
得      可爱的 p

```

文件操作

数据的读入:

```

open-r-w-a close ()
注意加 r 或者//
打开文件
f=open(r" c:\text\somefile.txt", "r")
其中最后的 r 表示读模式，此外还有 r+ (读写模式)，w (写模式)，w+ (读写模式)，a (追加模式)，b (二进制模式)

```

```

read 读入连续的字符串
readline 读入第一行字符串
readlines 读入每行字符串，且分行输出

```

The End:

为了方便携带和随时随地记忆，各位 Ali 君们可以把此资料打印下来后裁剪成两列，订起来看~如若不清晰的可结合上课老师课件的例题与作业，如若不懂欢迎联系 Ali 君团哦~~~~

期望各位同学能掌握好这一门程序技术语言，当然，也期望大家在 python 的考试中发挥出色，取得理想成绩，变得出彩~

Ali 君团联系方式:
官方 QQ: 3282934855
微信账号: alij2015

