Python for Senior

Lesson 3 v1.0.0

2016.10 by David.Yi

复习

- 函数使用进阶：位置参数，默认参数，元组参数，字典参数
- 匿名函数，lambda 的用法
- 计算一个正整数的因数

```
def add(x, y):
    return x + y

lambda x, y: x + y
```

```
# Find the factors of a number

import time

def print_factors(x):
   print("The factors of",x,"are:")
   for i in range(1, x + 1):
       if x % i == 0:
           print(i)

# take input from the user
num = int(input("Enter a number: "))

t1 = time.time()
print_factors(num)
t2 = time.time()
print(t2-t1)
```

本次内容要点

- python 常用内置函数
- List 列表的10个方法
- List 列表的复制
- 写一个简单的图形程序

Python 内置函数介绍 - max() min()

Python 将很多最常用的功能作为内置函数提供给开发者使用

- max() min() 取最大值、最小值

```
print(max(1,3,4,5))
```

```
5
```

```
print(max((1,2),(2,3),(2,4),(1,5)))
```

```
(2, 4)
```

```
print(max((1,2),(2,3),(2,4),(2,4,1)))
```

```
(2, 4, 1)
```

```
print(min((1,2),(2,3),(2,4),(2,4,1)))
```

```
(1, 2)
```

Python 内置函数介绍 - range()

- range() xrange() 生成有序列表，后者适用于大范围数值
- 在 python 3 开始，range() 增强为 xrange()，xrange()被取消

```
print(range(10))
```

```
range(0, 10)
```

```
print(range(1,10,3))
```

```
range(1, 10, 3)
```

```
print(type(range(10)))
```

```
<class 'range'>
```

Python 内置函数介绍 – int() float()...

- int() float() 数字转换
- tuple() list() dict() 数据结构转换

```
print(int('10'))
print(int(9))
```

```
10
9
```

```
print(float('123.45')+20)
```

```
143.45
```

```
l = [('tom', '100'), ('jerry', '90'), ('mary', '80')]
d = dict(l)
print(d)
```

```
{'tom': '100', 'mary': '80', 'jerry': '90'}
```

Python 内置函数介绍 – all() any()

- all() any() 判断符合条件情况的函数

```
l = [True,True,True,True]
print(all(l))
```

```
True
```

```
l = [True,True,False,True]
print(all(l))
```

```
False
```

```
l = [True,True,False,True]
print(any(l))
```

```
True
```

Python 内置函数介绍 – enumerate()

- enumerate() 生成迭代对象的序列

```
l = ['amy', 'tom', 'jerry ']
for item in l:
    print(item)
```

```
amy
tom
jerry
```

```
l = ['amy', 'tom', 'jerry ']
for i, item in enumerate(l):
    print(i, item)
```

```
0 amy
1 tom
2 jerry
```

List 列表方法 – append, extend

- append：向列表的尾部添加一个新的元素
- extend：用于在列表末尾一次性追加另一个列表中的多个值（用新列表扩展原来的列表）

```
# list append
l = ['tom', 'jerry', 'steven']
l.append('mary')
print(l)
```

```
['tom', 'jerry', 'steven', 'mary']
```

```
# list extend
l.extend(['may', 'david'])
print(l)
```

```
['tom', 'jerry', 'steven', 'mary', 'may', 'david']
```

List 列表方法 – insert, remove, pop, clear

- Insert：在列表指定位置插入元素
- remove： 删除列表中第一个匹配的值
- pop： 删除列表中指定位置的值
- clear：清空整个列表

```
# list insert
l = ['tom', 'jerry', 'steven']
l.insert(2,'helen')
print(l)
```

```
['tom', 'jerry', 'helen', 'steven']
```

```
# list remove
l = ['tom', 'jerry', 'steven']
l.remove('tom')
print(l)
```

```
['jerry', 'steven']
```

```
# list pop
l = ['tom', 'jerry', 'steven']
l.pop(1)
print(l)
```

```
['tom', 'steven']
```

List 列表方法 – index，count

- index：返回查找内容的索引
- count：返回查找内容的数量

```
# list index
l = ['tom', 'jerry', 'steven']
print(l.index('jerry'))
```

```
1
```

```
# list count
l = ['tom', 'jerry', 'steven', 'steven']
print(l.count('tom'))
print(l.count('steven'))
```

```
1
2
```

List 列表方法 – sort, reverse

- sort：对列表中的元素进行排序
  - 列表的 sort 方法和 sorted() 函数
- reverse：反转列表中的元素

```
# sort method of list
l = ['tom', 'jerry', 'steven']
l.sort()
print(l)
```

```
['jerry', 'steven', 'tom']
```

```
# sorted function
l = ['tom', 'jerry', 'steven']
l1 = sorted(l)
print(l1)
```

```
['jerry', 'steven', 'tom']
```

```
# reverse method of list
l = ['tom', 'jerry', 'steven']
l.reverse()
print(l)
```

```
['steven', 'jerry', 'tom']
```

List 列表内容的复制

- 简单复制
- 真正的列表内容复制

```
# copy list #1
l = ['tom', 'jerry', 'steven']
l1 = l
print(l)
print(l1)

l.append('may')
print(l)
print(l1)
```

```
['tom', 'jerry', 'steven']
['tom', 'jerry', 'steven']
['tom', 'jerry', 'steven', 'may']
['tom', 'jerry', 'steven', 'may']
```

```
# copy list, real copy #2
l = ['tom', 'jerry', 'steven']
l1 = l[:]
print(l)
print(l1)

l.append('may')
print(l)
print(l1)
```

```
['tom', 'jerry', 'steven']
['tom', 'jerry', 'steven']
['tom', 'jerry', 'steven', 'may']
['tom', 'jerry', 'steven']
```

写一个简单的图形界面程序

- 程序按照界面可以区分为：没有界面、web 界面、windows 界面、手机界面等
- python 完全可以写 windows 界面程序，不过不适合界面过于复杂的程序

```
import tkinter

top = tkinter.Tk()
top.mainloop()
```

```
import tkinter

label = tkinter.Label(text='Hello World!')
label.pack()
tkinter.mainloop()
```

举例

- 打印乘法表

```
1x1=1
2x1=2   2x2=4
3x1=3   3x2=6   3x3=9
4x1=4   4x2=8   4x3=12  4x4=16
5x1=5   5x2=10  5x3=15  5x4=20  5x5=25
6x1=6   6x2=12  6x3=18  6x4=24  6x5=30  6x6=36
7x1=7   7x2=14  7x3=21  7x4=28  7x5=35  7x6=42  7x7=49
8x1=8   8x2=16  8x3=24  8x4=32  8x5=40  8x6=48  8x7=56  8x8=64
9x1=9   9x2=18  9x3=27  9x4=36  9x5=45  9x6=54  9x7=63  9x8=72  9x9=81
```