Life is Short, you need python

—— An introduction to python

←请扫码
签到

图书馆学科咨询部 蔺梅芳

2017.12

WHY ?

为什么要学Python？

这是趋势

目前最流程最火的编程语言,没有之一

简单易学

学习简单,容易上手,使用灵活,可扩展方向广

能拿高薪

工资远超其他编程语言,学会即可获得高薪

Windows应用

网络编程

嵌入和扩展

数据库

多媒体处理

游戏编程

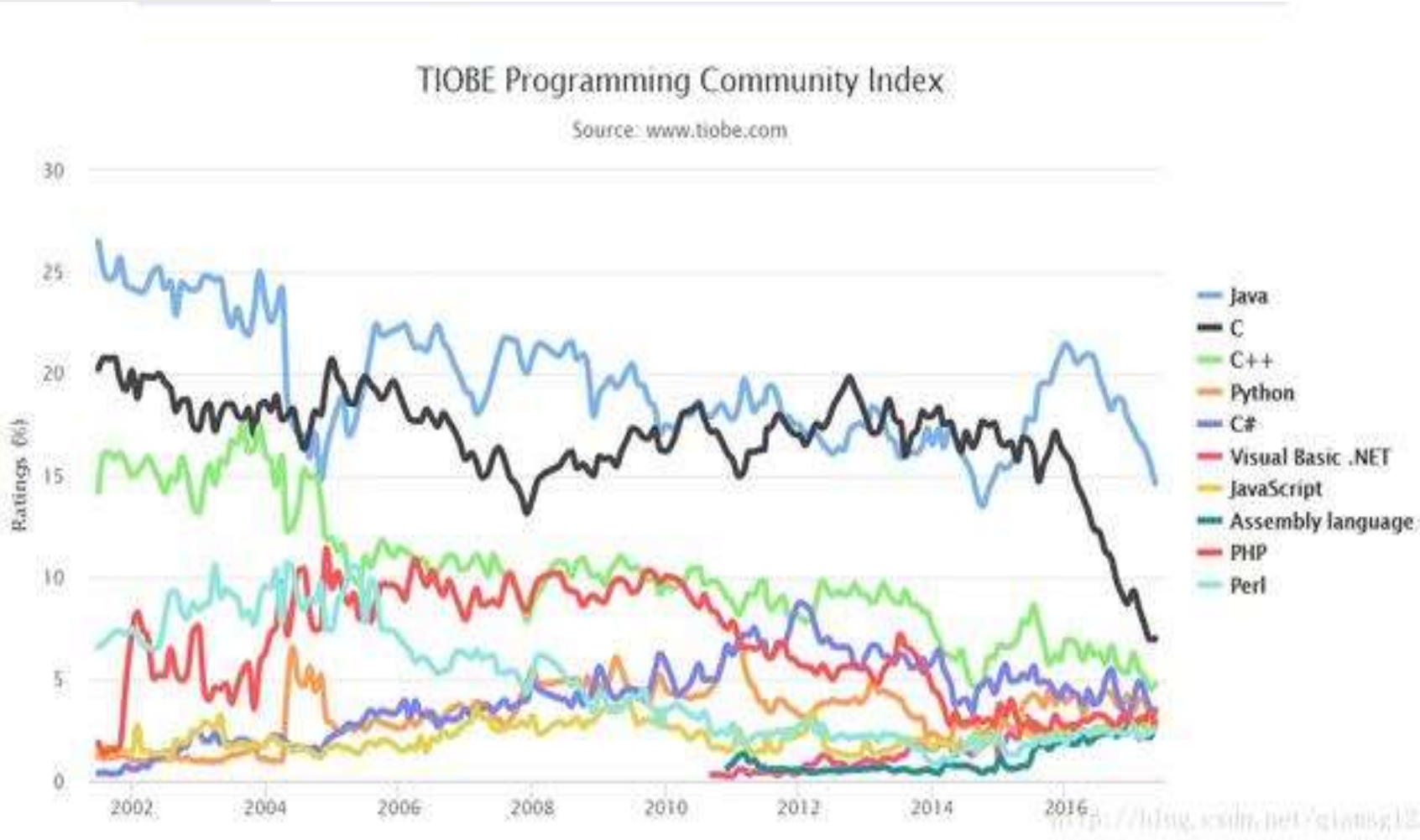
科学计算

GUI编程

企业与政务应用

TIOBE：2017年5月全球编程语言排行榜 Python升至第四名

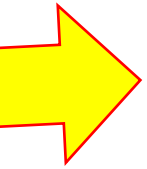
May 2017	May 2016	Change	Programming Language	Ratings
1	1		Java	
2	2		C	
3	3		C++	
4	5	↑	Python	
5	4	↓	C#	
6	10	⇈	Visual Basic .NET	
7	7		JavaScript	
8	12	⇈	Assembly language	
9	6	↓	PHP	
10	9	↓	Perl	
11	8	↓	Ruby	
12	13	↑	Visual Basic	
13	15	↑	Swift	
14	16	↑	R	
15	14	↓	Objective-C	
16	42	⇈	Go	
17	18	↑	MATLAB	
18	11	⇊	Delphi/Object Pascal	
19	19		PL/SQL	
20	22	↑	Scratch	



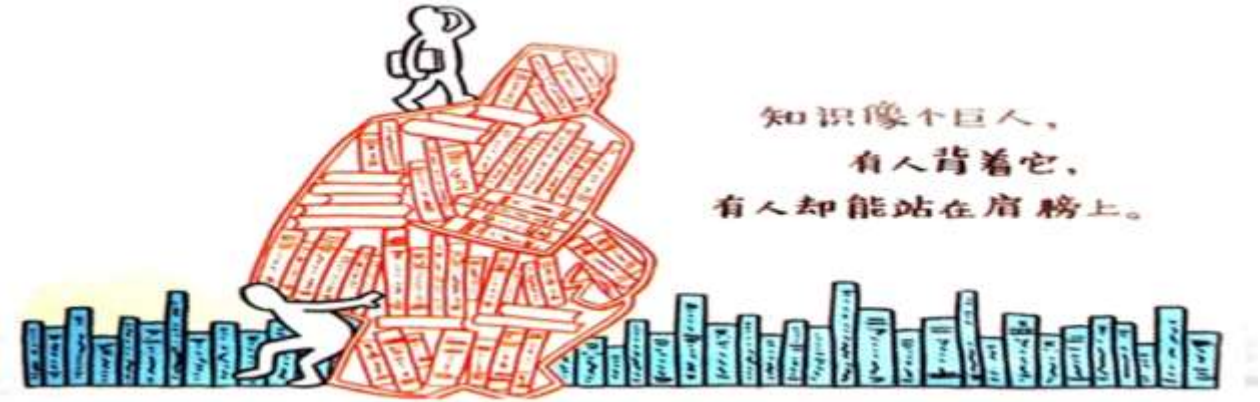

简单
明确
优雅
拥有简单脚本语言和解释
型程序语言的易用性
拥有传统编译型程序语言所
有强大通用的功能
Python是一种解释型的、面向对象的、
带有动态语义的高级程序设计语言


python

Open
&
Free



Simple
&
Strong



python

Python 格言

The Zen of Python

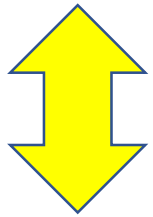
Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!

by Tim Peters

Why am I here ?
Who am I ?

抛砖引玉，
启迪思维，
总有一款是
你的菜。

Need



Interest

Thanks to :

Python 2.7.9 documentation

Welcome! This is the documentation for Python 2.7.9, last updated Dec 10, 2014.

Parts of the documentation:

What's new in Python 2.7?
or all "What's new" documents since 2.0

Tutorial
start here

Library Reference
keep this under your pillow

Language Reference
describes syntax and language elements

Python Setup and Usage
how to use Python on different platforms

Python HOWTOs
in-depth documents on specific topics

Extending and Embedding
tutorial for C/C++ programmers

Python/C API
reference for C/C++ programmers

Installing Python Modules
information for installers & sys-admins

Distributing Python Modules
sharing modules with others

FAQs
frequently asked questions (with answers!)

Thanks to:

- Mooc-中国大学在线

Python语言程序设计

北京理工大学

用Python玩转数据

南京大学

Python科学计算三维可视化

北京理工大学

Python网络爬虫与信息提取

北京理工大学

2017年度全新上线的Python语言系列专题课，带给

>>Python 网络爬虫与信息提取

http://www.icourse163.org/course/BIT-1001870001

>>Python 数据分析与展示

http://www.icourse163.org/course/BIT-1001870002

>>Python 机器学习应用

http://www.icourse163.org/course/BIT-1001872001

>>Python 科学计算三维可视化

http://www.icourse163.org/course/BIT-1001871001

>>Python 游戏开发入门

http://www.icourse163.org/course/BIT-1001873001

>>Python 云端系统开发入门

http://www.icourse163.org/course/BIT-1001871002

Thanks to:

廖雪峰的官方网站

- Python教程
  - Python简介
  - 安装Python
  - 第一个Python程序
  - Python基础
  - 函数
  - 高级特性
  - 函数式编程
  - 模块
  - 面向对象编程
  - 面向对象高级编程
  - 错误、调试和测试
  - IO编程
  - 进程和线程
  - 正则表达式
  - 常用内建模块
  - 常用第三方模块
  - virtualenv
  - 图形界面
  - 网络编程
  - 电子邮件
  - 访问数据库
  - Web开发
  - 异步IO

- 语法基础

  —— Start from “Hello World!”

- 综合示例

  —— When 苏轼 met Python

PART 1

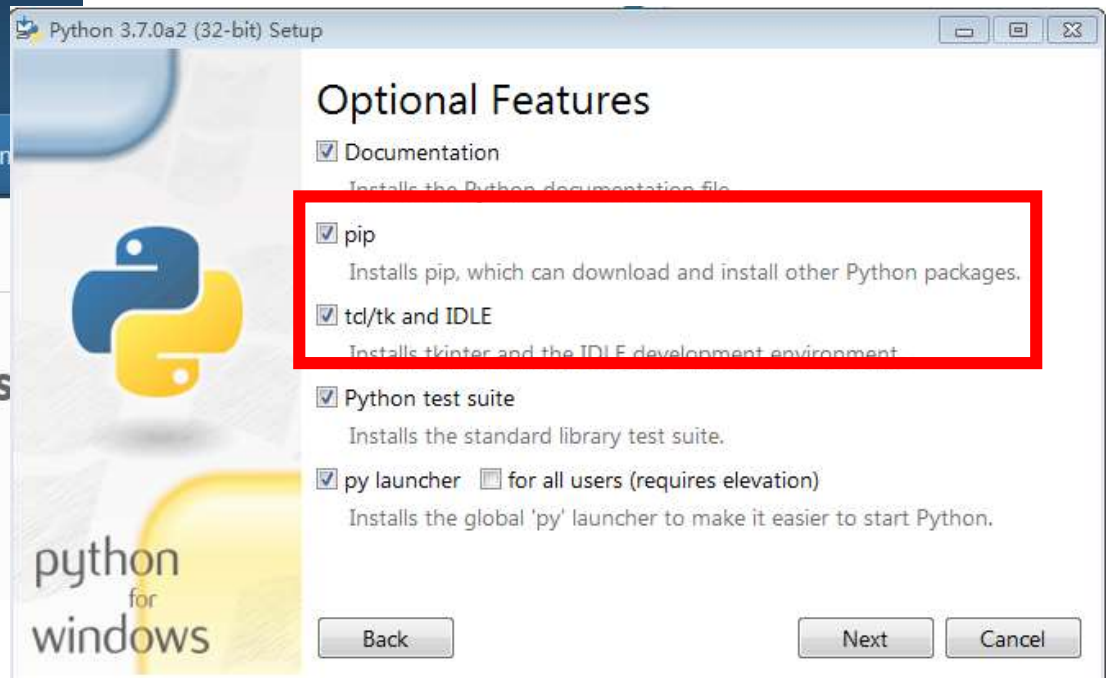
语法基础 — Start from “Hello World!”

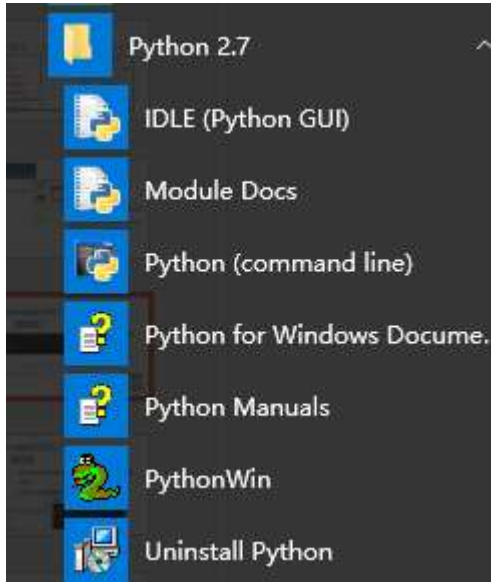
1. 基础：	2. 示例	3. 从示例再出发：
① 安装 ② 运行方式 ③ 书写与变量 ④ 数据类型 ⑤ 表达式与运算符 ⑥ 流控制语句	——猜数字	① More on Lists ② Functions ③ Exceptions

1. 基础：

① 安装


python
About
Downloads
Documen
Python ››› Downloads ››› Windows
Python Releases for Windows
Latest Python 3 Release - Python 3.6.3
Latest Python 2 Release - Python 2.7.14
Python 3.7.0a2 - 2017-10-17


Python 3.7.0a2 (32-bit) Setup
Optional Features
Documentation
pip
Installs pip, which can download and install other Python packages.
tcl/tk and IDLE
Python test suite
Installs the standard library test suite.
py launcher
for all users (requires elevation)
Installs the global 'py' launcher to make it easier to start Python.
python
for
windows
Back
Next
Cancel


Python 2.7
IDLE (Python GUI)
Module Docs
Python (command line)
Python for Windows Docume.
Python Manuals
PythonWin
Uninstall Python


python

1. 基础：

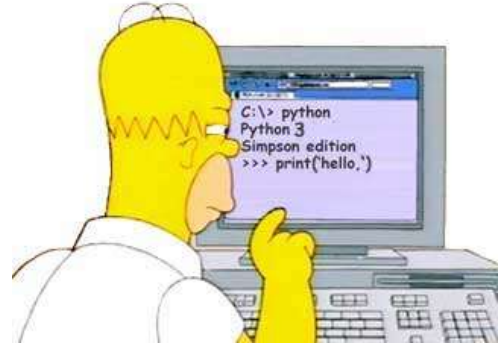
② 运行方式

- Shell方式

```
D:\Python27\python.exe
Python 2.7.9 (default, Dec 10 2014, 12:24:55) [MSC v.1500 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 100+200
300
>>> print('Hello, World!')
Hello, World!
>>> print "Hello, World!"
Hello, World!
>>>
```

```
Python 2.7.9 Shell
File  Edit  Shell  Debug  Options  Windows  Help
Python 2.7.9 (default, Dec 10 2014, 12:24:55) [MSC v.1500 32 bit (
32
Type "copyright", "credits" or "license()" for more information.
>>> print('Hello, World!')
Hello, World!
>>>
```

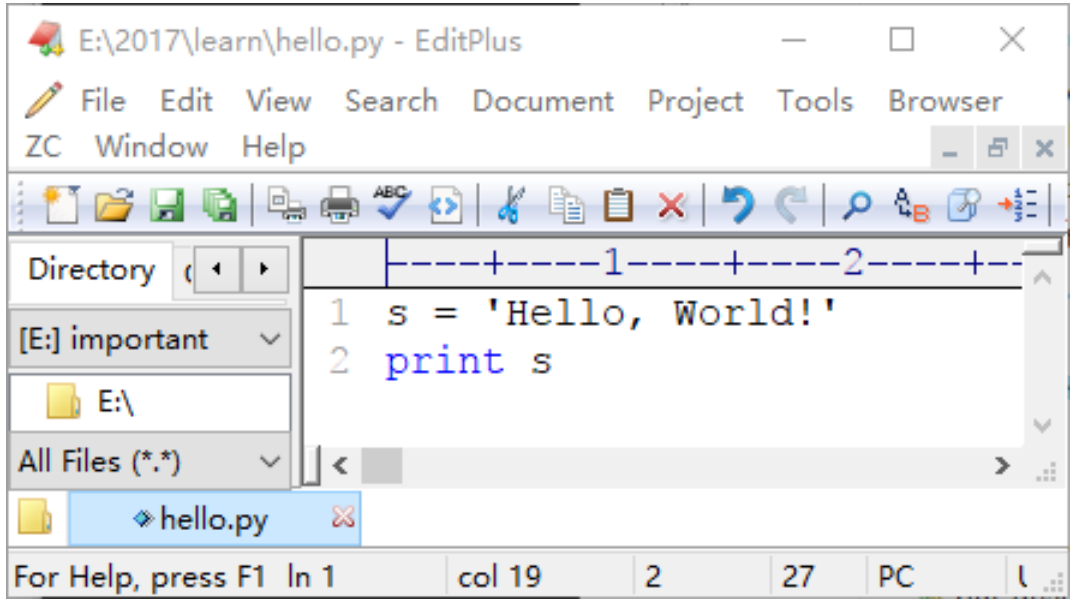
- Shell是交互式的解释器
- 输入一行命令，解释器就解释运行出相应结果

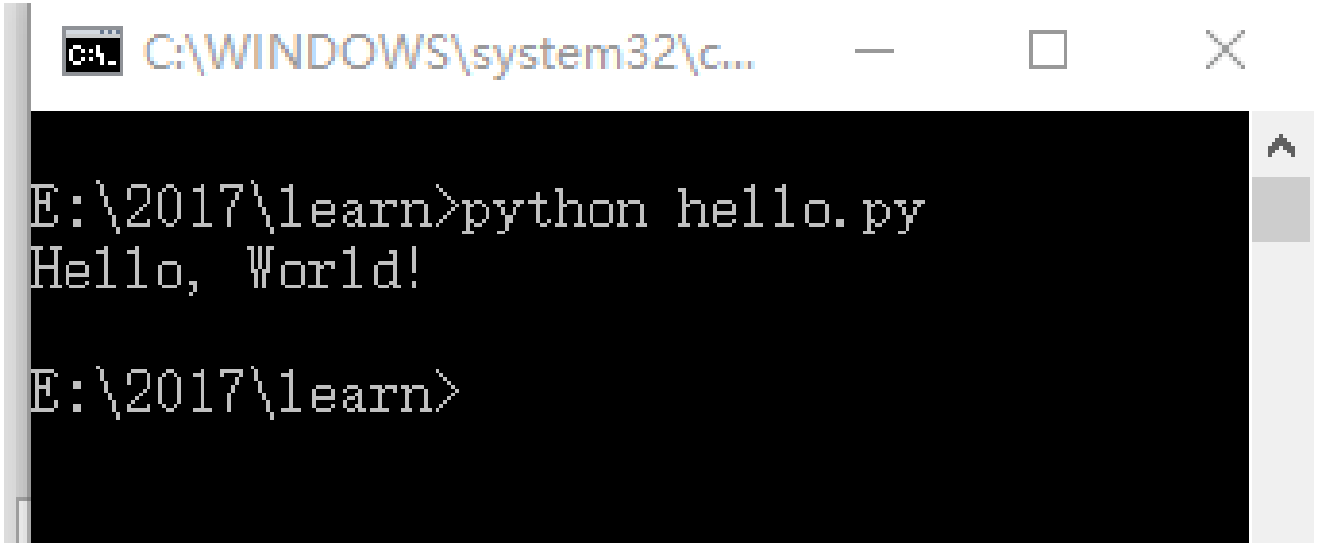


1. 基础：

② 运行方式

- 文件方式





- 在python的IDE环境中，创建一个以py为扩展名的文件
- 用python解释器在Shell中运行出结果

1. 基础：

② 运行方式

经典 Hello World

Python 3.x中print语句用print()函数替代

Source

```
>>> myString = 'Hello, World!'
>>> print myString
Hello World!
>>> myString
'Hello World!'
```

File

```
# Filename: helloworld.py
mystring = 'Hello, World!'
print mystring
```

1. 基础：

③ 书写与变量

注释

换行

一行多语句

缩进

01 增加缩进表示语句块的开始

02 Python用相同的缩进表示同级别语句块

03 减少缩进表示语句块的退出

Source

```
>>> # Indentation
>>> if (signal == 'red') and (car == 'moving'):
        car = 'stop'
        singal = 'yellow'
    elif (signal == 'green') and (car == 'stop'):
        car = 'moving'
        singal = 'yellow'
```

1. 基础：

③ 书写与变量

- 变量第一次赋值，同时获得类型和“值”
  - Python是动态的强类型语言
  - 不需要显式声明，根据“值”确定类型
  - 以“引用”的方式实现赋值

大小写敏感

1. 基础：

④ 数据类型

- 整型
- 布尔型
- 浮点型
- 复数型

- 字符串 string
- 列表 list
- 元组 tuple

} sequence data types

- 集合 set
- 字典 dict

1. 基础：

④ 数据类型


序列类型
01
字符串
单引号、双引号、三引号内的都是字符串，不可变类型
02
列表
强大的类型，用方括号 [] 界别，可变类型
03
元组
与列表相似，用小括号 () 界别，不可变类型

1. 基础：

④ 数据类型

01 字符串

单引号、双引号、三引号内的都是字符串
不可变类型

```
 +---+---+---+---+---+---+
 | P | y | t | h | o | n |
 +---+---+---+---+---+---+
 0   1   2   3   4   5   6
-6  -5  -4  -3  -2  -1
```

Strings can be indexed and sliced:

```
>>> word = 'Python'
>>> word[0]  # chara
'P'
>>> word[5]  # chara
'n'
```

```
>>> word[0:2]
'Py'
>>> word[2:5]
'tho'
```

Question:

请给出word的

最后1个字符？

最后3个字符？

```
>>> word = 'Python'
>>> word[-1]
'n'
>>> word[-3:]
'hon'
```

02 列表

强大的类型，用方括号[]界别，可变类型

```
squares = [1,4,9,16,25,36,49,64,80,100]
```

Lists

非常重要，

非常好用，

非常非常！

Question:请给出squares的第1个元素，最后1个元素，以及最后3个元素？

```
>>> squares = [1,4,9,16,25,36,49,64,80,100]
>>> squares[-2]=81
>>> squares
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

02 列表

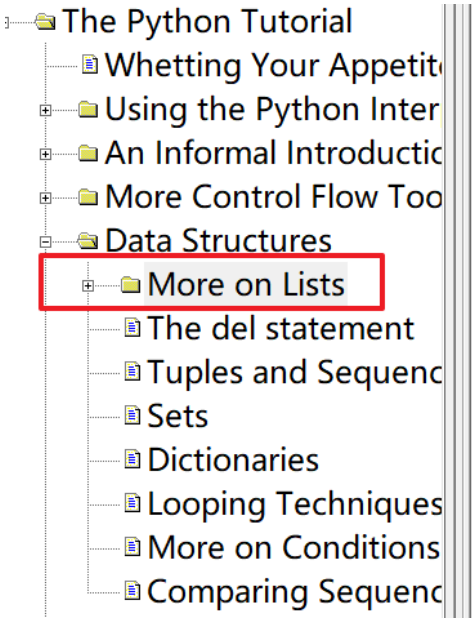
强大的类型，用方括号 [] 界别，可变类型

It is possible to nest lists (create lists containing other lists), for example:

```
>>> a = ['a', 'b', 'c']
>>> n = [1, 2, 3]
>>> x = [a, n]
>>> x
[['a', 'b', 'c'], [1, 2, 3]]
>>> x[0]
['a', 'b', 'c']
>>> x[0][1]
'b'
```

02 列表

强大的类型，用方括号[]界别，可变类型



list.insert(*i*, *x*)

Insert an item at a g
list, and a.insert(le

list.remove(*x*)

Remove the first ite

list.pop([*i*])

Remove the item at

```
>>> a = [66.25, 333, 333, 1, 1234.5]
>>> print a.count(333), a.count(66.25), a.count('x')
2 1 0
>>> a.insert(2, -1)
>>> a.append(333)
>>> a
[66.25, 333, -1, 333, 1, 1234.5, 333]
>>> a.index(333)
1
>>> a.remove(333)
>>> a
[66.25, -1, 333, 1, 1234.5, 333]
>>> a.reverse()
>>> a
[333, 1234.5, 1, 333, -1, 66.25]
>>> a.sort()
>>> a
[-1, 1, 66.25, 333, 333, 1234.5]
>>> a.pop()
1234.5
>>> a
[-1, 1, 66.25, 333, 333]
```

02 列表

强大的类型，用方括号 [] 界别，可变类型

```
>>> squares = []
>>> for x in range(10):
...     squares.append(x**2)
...
>>> squares
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

Question:

生成包含小于100的所有平方数

List Comprehensions

```
squares = [x**2 for x in range(10)]
```

```
>>> a = 1
>>> b = a
>>> a,b
(1, 1)
>>> a = 2
>>> a,b
(2, 1)
```

数值型变量赋值

列表赋值要注意：实际传递的是引用

```
>>> a = [1,2]
>>> b = a
>>> a.append(3)
>>> a
[1, 2, 3]
>>> b
[1, 2, 3]
```

可以吗？

```
>>> a = [1,2]
>>> b = a[:]
>>> a.append(3)
>>> a
[1, 2, 3]
>>> b
[1, 2]
```

```
>>> a = [1,[2]]
>>> b= a[:]
>>> b
[1, [2]]
>>> a[1].append(3)
>>> a
[1, [2, 3]]
>>> b
[1, [2, 3]]
>>> import copy
>>> c = copy.deepcopy(a)
>>> a
[1, [2, 3]]
>>> c
[1, [2, 3]]
>>> a[1].append(4)
>>> a
[1, [2, 3, 4]]
>>> c
[1, [2, 3]]
```

在python中，**strings, tuples, 和numbers**是不可更改的对象(immutable)，

而**list,set,dict**等则是可以修改的对象(mutable)。

你想修改不可更改的对象时，其实就开辟了一个新的空间存储新的对象

元组 03

与列表相似，用小括号 () 界别，不可变类型

We saw that lists and strings have many common properties, such as indexing and slicing operations.

There is also another standard sequence data type: the tuple.

```
>>> t = (1,'a',3)
>>> t
    (1, 'a', 3)
>>> t = 1,'b',3
>>> t
    (1, 'b', 3)
>>> t[1] = 2
    Traceback (most recent call last):
      File "<string>", line 1, in <fragment>
    TypeError: 'tuple' object does not support item assignment
>>> t[1]
    'b'
```

??

Question: What's the difference between tuples and lists?

1. 基础：

④ 数据类型

Set--- an unordered collection with no duplicate elements

- **字符串 string**
- **列表 list**
- **元组 tuple**
- **集合 set**
- **字典 dict**

- Set objects also support mathematical operations like union, intersection, difference, and symmetric difference.

```
>>> basket = ['apple', 'orange', 'apple', 'pear', 'orange', 'banana']
>>> fruit = set(basket)               # create a set without duplicates
>>> fruit
set(['orange', 'pear', 'apple', 'banana'])
>>> 'orange' in fruit                 # fast membership testing
True
>>> 'crabgrass' in fruit
False
```

1. 基础：

④ 数据类型

- 字符串 string
- 列表 list
- 元组 tuple
- 集合 set
- **字典 dict**

映射类型 字典

- 用大括号 {} 界别
- 类似于哈希表的键值对

```
>>> tel = {'jack': 4098, 'sape': 4139}
>>> tel['guido'] = 4127
>>> tel
{'sape': 4139, 'guido': 4127, 'jack': 4098}
>>> tel['jack']
4098
```

1. 基础：

⑤ 表达式与运算符

- 用运算符连接各种类型数据的式子就是表达式

1. 基础：

⑥ 流控制语句

if, while, for------Control Flows

if Statements

```
>>> x = int(raw_input("Please enter an integer: "))
Please enter an integer: 42
>>> if x < 0:
...     x = 0
...     print 'Negative changed to zero'
... elif x == 0:
...     print 'Zero'
... elif x == 1:
...     print 'Single'
... else:
...     print 'More'
...
```

```
>>> p = 2
>>> if p:print p
    2
>>> p = 0
>>> if p: print p
```

Question: if 的条件变量p是哪些情况时就判断为真？

```
>>> s = ''
>>> if s: print 'string'
>>> if not s: print 'string'
    string
>>> plist = [1,'a']
>>> if plist: print len(plist)
    2
```

1. 基础：

⑥ 流控制语句

for Statements

```
>>> # Measure some strings:
... words = ['cat', 'window', 'defenestrate']
>>> for w in words:
...     print w, len(w)
...
cat 3
window 6
defenestrate 12
```

```
>>> for n in range(2, 10):
...     for x in range(2, n):
...         if n % x == 0:
...             print n, 'equals', x, '*', n/x
...             break
...     else:
...         # loop fell through without finding a factor
...         print n, 'is a prime number'
...
2 is a prime number
3 is a prime number
4 equals 2 * 2
5 is a prime number
6 equals 2 * 3
7 is a prime number
8 equals 2 * 4
9 equals 3 * 3
```

PART 1

语法基础 — Start from “Hello World!”

1. 基础：	2. 示例	3. 从示例再出发：
① 安装 ② 运行方式 ③ 书写与变量 ④ 数据类型 ⑤ 表达式与运算符 ⑥ 流控制语句	——猜数字	① More on Lists ② Functions ③ Exceptions

2. 示例
——猜数字

猜数字游戏

猜

- 程序随机产生一个0~ 100 间的整数，玩家竞猜，系统给出“猜中”、“太大了”或“太小了”的提示。

猜

猜

File

```
# Filename: guessnum1.py
from random import randint

x = randint(0, 100)
print 'Please input a number between 0~300:'
digit = input()
if digit  == x :
    print 'Bingo!'
elif digit > x:
    print 'Too large,please try again.'
else:
    print 'Too small,please try again.'
```

猜数字游戏的扩展与完善

```python
from random import randint
#===============
def judge(n1, n2): #比较两数大小并给出结果，如果相等则返回True
    res = False
    if n1==n2: #局部变量
        print('YOU WIN!')
        res = True
    elif n1>n2:
        print('Too large, please try again!')
    else:
        print('Too small, please try again!')
    return res
#==============
def getinput(): #获取用户输入，要求必须是0~100之间的整数
    while True:
        ns = raw_input("Give your number(0~100): ")
        try: #异常处理，假如不进行异常捕获处理则程序会出错并中断
            n = int(ns) #如果输入不是数字，就会发生异常
            if n<100 and n>0:
                break
            else:
                print("Input must be a number(0~100)!")
        except:
            print("Input must be a number(0~100)!")
    return n
#===============
```

```python
#主函数--应简单明了
#if __name__=='__main__':
n0 = randint(0, 100) #全局变量
print n0
for i in range(1,4): #给用户3次猜测机会
    n = getinput()
    res = judge(n, n0)
    if res:
        break
    else:
        if i>0:
            print("you have %i chance left!"%(3-i))
        else:
            print("SORRY, YOU LOSE!")
```

```
Debug I/O (stdin, stdout, stderr) appears below

62
Give your number(0~100): 50
Too small, please try again!
you have 2 chance left!
Give your number(0~100): 75
Too large, please try again!
you have 1 chance left!
Give your number(0~100): 62
YOU WIN!
```

3. 从示例再出发：

① More on Lists

函数range(start, stop, step)

Python3.x中：

```
data_range = list(range(1,100,2))
```

Question:

请给出100以内的奇数？

```
>>> data_range = range(1,100,2)
>>> data_range
    [1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41
>>> type(data_range)
    <type 'list'>
```

Python2.x

3. 从示例再出发：

② Functions

Required Argument

Default, Optional Arguments

```
def ask_ok(prompt, retries=4, complaint='Yes or no, please!'):
```

num = 2

（函数的参数传递）

ask_ok("Are you OK?")

ask_ok("Are you OK? >>", num)

ask_ok("Are you OK? >>", retries=num)

ask_ok(prompt = "Are you OK? >>", retries = num)

ask_ok(retries = num, prompt = "Are you OK? >>")

ask_ok(num, "Are you OK? >>")

ask_ok(retries = num, "Are you OK? >>")

3. 从示例再出发：

② Functions

（函数的参数传递）

```
57  def cheeseshop(kind, *arguments, **keywords):
58      print "-- Do you have any", kind, "?"
59      print "-- I'm sorry, we're all out of", kind
60      for arg in arguments:
61          print arg
62      print "-" * 40
63      keys = sorted(keywords.keys())
64      for kw in keys:
65          print kw, ":", keywords[kw]
66  #==================================
67  cheeseshop("Limburger",
68             "It's very runny, sir.",
69             "It's really very, VERY runny, sir.",
70             shopkeeper='Michael Palin',
71             client="John Cleese",
72             sketch="Cheese Shop Sketch")
```

3. 从示例再出发：

② Functions

（函数的参数传递）

```
57  def cheeseshop(kind, *arguments, **keywords):
58      print "-- Do you have any", kind, "?"
59      print "-- I'm sorry, we're all out of", kind
60      for arg in arguments:
61          print arg
62      print "-" * 40
63      keys = sorted(keywords.keys())
64      for kw in keys:
65          print kw, ":", keywords[kw]
66  #==================================
67  cheeseshop("Limburger",
68             "It's very runny, sir.",
69             "It's really very, VERY runny, sir.",
70             shopkeeper='Michael Palin',
71             client="John Cleese",
72             sketch="Cheese Shop Sketch")
73
```

Search in Files | Search | Stack Data | Exceptions | Breakpoints | Testing

cheeseshop(): myfunction.py, line 58

Variable	Value
locals	<dict 0x294f5d0; len=3>
arguments	("It's very runny, sir.", "It's really very, VERY runny, sir.")
keywords	<dict 0x29099c0; len=3>
kind	'Limburger'

Debug I/O | Debug Probe

cheeseshop(): myfunction.p

Commands execute in curr

```
>>> type(arguments)
<type 'tuple'>
>>> type(keywords)
<type 'dict'>
```

3. 从示例再出发：

② Functions

（函数的参数传递）

```
43  def changeme(age,s,mylist):
44      age = 30
45      s = 'chaned s'
46      mylist.append(1)
47      print '函数内: ', age, s, mylist
48  #===
49  age = 20
50  s = 'orinal s'
51  mylist = ['999']
52  print '原始值: ', age, s, mylist
53  changeme(age, s, mylist)
54  print '函数外: ',age, s, mylist
```

3. 从示例再出发：

② Functions

按引用传递参数

```
43  def changeme(age,s,mylist):
44      age = 30
45      s = 'chaned s'
46      mylist.append(1)
47      print '函数内: ', age, s, mylist
48  #===
49  age = 20
50  s = 'orinal s'
51  mylist = ['999']
52  print '原始值: ', age, s, mylist
53  changeme(age, s, mylist)
54  print '函数外: ',age, s, mylist
```

Search in Files | Search | Stack Data | Exceptions | Breakpoints | Tes

changeme(): myfunction.py, line 47

Variable	Value
▲ locals	{'s': 'chaned s', 'mylist': ['999', 1], 'age': 30}
age	30
▷ mylist	['999', 1]
s	'chaned s'
▲ globals	<dict 0x28498a0; len=9>
__doc__	None
__file__	'D:\\learnpy\\myfunction.py'
__name__	'__main__'
age	20
▷ mylist	['999', 1]
s	'orinal s'

```
Debug I/O (stdin, stdout, stderr) appears below

原始值:  20 orinal s ['999']
函数内:  30 chaned s ['999', 1]
函数外:  20 orinal s ['999', 1]
```

（函数的参数传递）

3. 从示例再出发：

② Exceptions


```
2.7.9 (default, Dec 10 2014, 12:24:55) [MSC v.1500 32 bit (Intel)]
Python Type "help", "copyright", "credits" or "license" for more in
>>> int('a')
Traceback (most recent call last):
  File "<string>", line 1, in <fragment>
ValueError: invalid literal for int() with base 10: 'a'
>>> 10/0
Traceback (most recent call last):
  File "<string>", line 1, in <fragment>
ZeroDivisionError: integer division or modulo by zero
>>> '2' + 2
Traceback (most recent call last):
  File "<string>", line 1, in <fragment>
TypeError: cannot concatenate 'str' and 'int' objects
```

```
try：
    …
except:
    …
```

```
1  while True:
2      try:
3          x = int(raw_input("Please enter a number: "))
4          print(1.0/x)
5          #break
6      except ValueError:
7          print "Oops!  That was no valid number.  Try again..."
8      except Exception as e:
9          print (e.message + 'Try again')
10
```

Debug Probe | Debug I/O | Modules | Python Shell | Bookmarks | Messages

Debug I/O (stdin, stdout, stderr) appears below

```
Please enter a number: 2
0.5
Please enter a number: a
Oops!  That was no valid number.  Try again...
Please enter a number: 0
float division by zeroTry again
Please enter a number:
```

PART 1

语法基础 — Start from “Hello World!”

1. 基础：	2. 示例	3. 从示例再出发：
① 安装 ② 运行方式 ③ 书写与变量 ④ 数据类型 ⑤ 表达式与运算符 ⑥ 流控制语句	——猜数字	① More on Lists ② Functions ③ Exceptions

小作业

已有论文作者列表，要求给出每篇论文的作者人数，以及指定作者在其中的排序；论文共计1.6万篇。

待判断作者	全部作者	作者总数	指定作者排位
Cannon, Andrew H.	Ramesh, Ashwin; Akram, Wasim; Mishra, Surya P.; Cannon, A	6	4
Cao, Binfang	Peng, Guanghan; Nie, Fangyan; Cao, Binfang; Liu, Changqin	4	3
Cao, Chongsheng	Cao, Chongsheng; Wu, Jiahong	2	1
Cao, Junpeng	Zhang, Xin; Li, Yuan-Yuan; Cao, Junpeng; Yang, Wen-Li; Sh	6	3
Caraeni, Doru	Wang, Z. J.; Fidkowski, Krzysztof; Abgrall, Remi; Bassi,	15	5

```
fin = file('aupos.txt', 'r')
line = fin.readline().replace('\n','').lower()
reslist = []
while line:
    itemlist = line.split('\t')
    aulist = itemlist[1].split('; ')
    #===以下两行将进行完善
    nau = len(aulist)
    pau = aulist.index(itemlist[0].strip()
    #==
    reslist.append(str(nau) + '\t' + str(p
    line = fin.readline().replace('\n','')
file('out.txt','w').write('\n'.join(reslis
fin.close()
```

```
#===完善的部分
if itemlist[1]: #实际情况中源数据有字段缺失
    aulist = itemlist[1].split('; ')
nau = len(aulist)
if nau>0:
    try: #因为源数据的标引问题，待查作者有可能没有出现在作者列表中，\
         #这时index函数会出错，因此加以异常处理
        pau = aulist.index(itemlist[0].strip().lower())+1
    except:
        pau = 'notfound'
else:
    pau = 'losted au'
#===the end of 完善的部分
```

语法基础小结：需求与兴趣

Python HOWTOs

Python HOWTOs are documents that cover a sir Documentation Project's HOWTO collection, this cc Reference.

Currently, the HOWTOs are:

PART 2 当苏轼遇见python

- **All is data —— Get data from Web**

阅读修身

阶段	讲座名称	内容简介	开始时间	讲座地点	主讲人	适用人群	预约人数	课件下载
基础	闲聊诗词之一：苏轼的诗词人生	作为宋代男神的苏轼，集诗、词、文、音乐、书法、绘画等多种才艺于一身，虽然一生起伏波动，命运多舛，但能将别人的苟且活成自己的潇洒。让我们一起来闲聊一下你熟悉的苏轼！	2017-11-02 16:30:00	清水河图书馆二楼百学堂	王君莉	本、研	52 预约	无

明月几时有？把酒问青天。

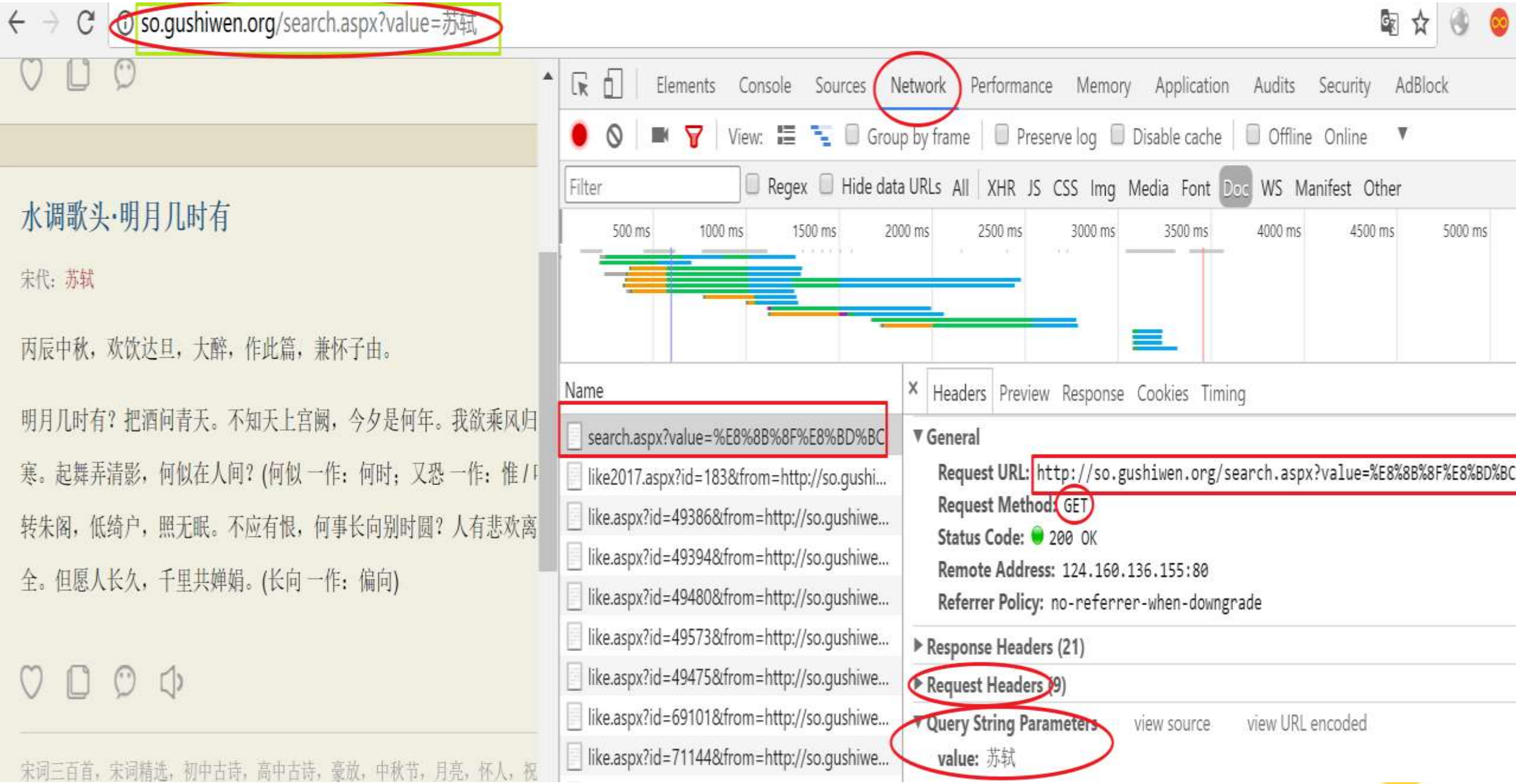
大江东去，浪淘尽，千古风流人物。

横看成岭侧成峰，远近高低各不同。

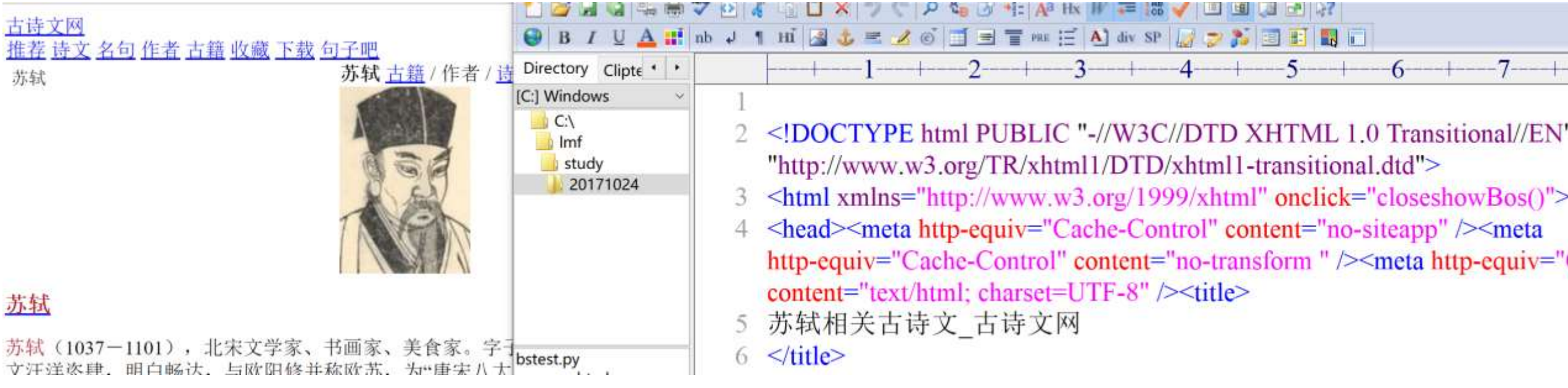
十年生死两茫茫，不思量，自难忘。

欲把西湖比西子，淡妆浓抹总相宜。

莫听穿林打叶声，何妨吟啸且徐行。竹杖芒鞋轻胜马，谁怕？一蓑烟雨任平生。
料峭春风吹酒醒，微冷，山头斜照却相迎。回首向来萧瑟处，归去，也无风雨也无晴。


so.gushiwen.org/search.aspx?value=苏轼
水调歌头·明月几时有
宋代：苏轼
丙辰中秋，欢饮达旦，大醉，作此篇，兼怀子由。
明月几时有？把酒问青天。不知天上宫阙，今夕是何年。我欲乘风归
寒。起舞弄清影，何似在人间？(何似 一作：何时；又恐 一作：惟/
转朱阁，低绮户，照无眠。不应有恨，何事长向别时圆？人有悲欢离
全。但愿人长久，千里共婵娟。(长向 一作：偏向)
宋词三百首，宋词精选，初中古诗，高中古诗，豪放，中秋节，月亮，怀人，祝
Elements Console Sources Network Performance Memory Application Audits Security AdBlock
View: Group by frame Preserve log Disable cache Offline Online
Filter Regex Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other
500 ms 1000 ms 1500 ms 2000 ms 2500 ms 3000 ms 3500 ms 4000 ms 4500 ms 5000 ms
Name
search.aspx?value=%E8%8B%8F%E8%BD%BC
like2017.aspx?id=183&from=http://so.gushi...
like.aspx?id=49386&from=http://so.gushiwe...
like.aspx?id=49394&from=http://so.gushiwe...
like.aspx?id=49480&from=http://so.gushiwe...
like.aspx?id=49573&from=http://so.gushiwe...
like.aspx?id=49475&from=http://so.gushiwe...
like.aspx?id=69101&from=http://so.gushiwe...
like.aspx?id=71144&from=http://so.gushiwe...
Headers Preview Response Cookies Timing
General
Request URL: http://so.gushiwen.org/search.aspx?value=%E8%8B%8F%E8%BD%BC
Request Method: GET
Status Code: 200 OK
Remote Address: 124.160.136.155:80
Referrer Policy: no-referrer-when-downgrade
Response Headers (21)
Request Headers (9)
Query String Parameters view source view URL encoded
value: 苏轼

```
>>> url = 'http://so.gushiwen.org/search.aspx?value=' + '苏轼'
>>> import urllib2
>>> r = urllib2.urlopen(url)
>>> file('out.html','w').write(r.read())
```



HOWTO Fetch Internet Resources Using urllib2

```
import urllib2
response = urllib2.urlopen('http://python.org/')
html = response.read()
```

```
import urllib2

req = urllib2.Request('http://www.voidspace.org.uk')
response = urllib2.urlopen(req)
the_page = response.read()
```

HOWTO Fetch Internet Resources Using urllib2

Data

POST

```
import urllib
import urllib2

url = 'http://www.someserver.com/cgi-bin/register.cgi'
values = {'name' : 'Michael Foord',
          'location' : 'Northampton',
          'language' : 'Python' }

data = urllib.urlencode(values)
req = urllib2.Request(url, data)
response = urllib2.urlopen(req)
the_page = response.read()
```

HOWTO Fetch Internet Resources Using urllib2

Data

GET

```
>>> import urllib2
>>> import urllib
>>> data = {}
>>> data['name'] = 'Somebody Here'
>>> data['location'] = 'Northampton'
>>> data['language'] = 'Python'
>>> url_values = urllib.urlencode(data)
>>> print url_values  # The order may differ.
name=Somebody+Here&language=Python&location=Northampton
>>> url = 'http://www.example.com/example.cgi'
>>> full_url = url + '?' + url_values
>>> data = urllib2.urlopen(full_url)
```

HOWTO Fetch Internet Resources Using urllib2

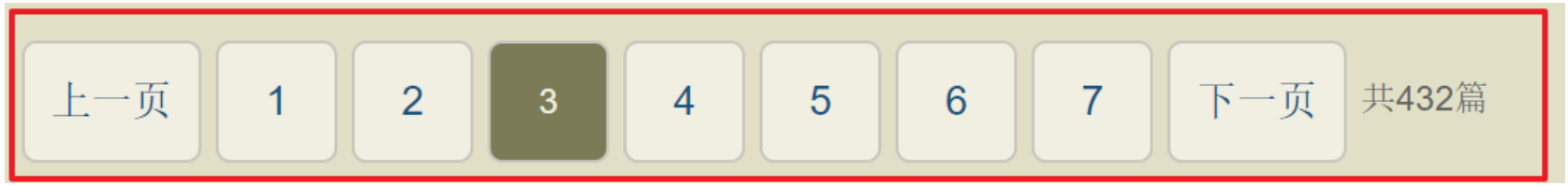
Headers

```
import urllib
import urllib2

url = 'http://www.someserver.com/cgi-bin/register.cgi'
user_agent = 'Mozilla/4.0 (compatible; MSIE 5.5; Windows NT)'
values = {'name' : 'Michael Foord',
          'location' : 'Northampton',
          'language' : 'Python' }
headers = { 'User-Agent' : user_agent }

data = urllib.urlencode(values)
req = urllib2.Request(url, data, headers)
response = urllib2.urlopen(req)
the_page = response.read()
```

```
>>> url = 'http://so.gushiwen.org/search.aspx?value=' + '苏轼'
>>> import urllib2
>>> r = urllib2.urlopen(url)
>>> file('out.html','w').write(r.read())
```



so.gushiwen.org/search.aspx?type=author&page=3&value=苏轼

```python
#=======================
def getpage():
    pass
#=======================
def parsetext():
    pass
#=======================
```

```python
# -*- coding: utf-8 -*-
import urllib2
#=======================
def getpage():
    #http://so.gushiwen.org/
    url = 'http://so.gushiwe
    r = urllib2.urlopen(url)
    file('out.html','w').wri
#=======================
def parsetext():
#=======================
poet = '苏轼'
for np in range(1, 2):
    getpage()
    parsetext()
#=======================
```

Fetching URLs

python 3.x

The simplest way to use urllib.request is as follows:

```python
import urllib.request
with urllib.request.urlopen('http://python.org/') as response:
   html = response.read()
```

If you wish to retrieve a resource via URL and store it in a temporary location, you can do

```python
import urllib.request
local_filename, headers = urllib.request.urlretrieve('http://python.org/')
html = open(local_filename)
```

```
86 <p><a style="font-size:18px; line-height:22px; height:22px; " href="/view_49386.aspx" target="_blank"><b>水调歌头·明月几时
   有</b></a></p>
87 <p class="source"><a href="/type.aspx?p=1&c=%e5%ae%8b%e4%bb%a3">宋代</a><span>： </span><a
   href="/search.aspx?value=%e8%8b%8f%e8%bd%bc"><span style="color:#B00815;line-height:100%;">苏轼</span></a></p>
88 <div class="contson" id="contson49386">
89 <p><span style="font-family:SimSun;">丙辰中秋，欢饮达旦，大醉，作此篇，兼怀子由。</span></p>
90 <p>明月几时有？把酒问青天。不知天上宫阙，今夕是何年。我欲乘风归去，又恐琼楼玉宇，高处不胜寒。起舞弄清影
   ，何似在人间？(何似 一作：何时；又恐 一作：惟 / 唯恐)<br />转朱阁，低绮户，照无眠。不应有恨，何事长向别时圆
   ？人有悲欢离合，月有阴晴圆缺，此事古难全。但愿人长久，千里共婵娟。(长向 一作：偏向)</p>
91 </div>
```

Beautiful Soup

Beautiful Soup 是一个可以从 **HTML** 或 **XML** 文件中提取数据的 **Python** 库.它能够通过你喜欢的转换器实现惯用的文档导航,查找,修改文档的方式.**Beautiful Soup** 会帮你节省数小时甚至数天的工作时间.

pip install beautifulsoup4

函数-模块-库

```
>>> import matplotlib

Traceback (most recent call last):
```

matplotlib

```
Microsoft Windows [版本 10.0.15063]
(c) 2017 Microsoft Corporation。保留所有权利。

C:\Users\meifa>pip install matplotlib
Collecting matplotlib
  Downloading matplotlib-2.0.2-cp27-cp27m-win32.whl (8.5MB)
    100% |████████████████████████████████| 8.5MB 113kB/s
Collecting cycler>=0.10 (from matplotlib)
  Downloading cycler-0.10.0-py2.py3-none-any.whl
Collecting numpy>=1.7.1 (from matplotlib)
  Downloading numpy-1.13.3-2-cp27-none-win32.whl (6.7MB)
    100% |████████████████████████████████| 6.7MB 118kB/s
Collecting pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=1.5.6 (from matplotlib)
  Downloading pyparsing-2.2.0-py2.py3-none-any.whl (56kB)
    100% |████████████████████████████████| 61kB 326kB/s
Collecting functools32 (from matplotlib)
  Downloading functools32-3.2.3-2.zip
Collecting python-dateutil (from matplotlib)
  Downloading python_dateutil-2.6.1-py2.py3-none-any.whl (194kB)
    100% |████████████████████████████████| 194kB 386kB/s
Collecting pytz (from matplotlib)
  Downloading pytz-2017.2-py2.py3-none-any.whl (484kB)
    100% |████████████████████████████████| 491kB 92kB/s
Requirement already satisfied: six>=1.10 in c:\python27\lib\site-packages (from matplotlib)
Installing collected packages: cycler, numpy, pyparsing, functools32, python-dateutil, pytz, matplotlib
  Running setup.py install for functools32 ... done
Successfully installed cycler-0.10.0 functools32-3.2.3.post2 matplotlib-2.0.2 numpy-1.13.3 pyparsing-2.2.0 python-dateut
il-2.6.1 pytz-2017.2

C:\Users\meifa>
```

使用pip工具查看、安装、更新库

C:\WINDOWS\system32\cm

```
Microsoft Windows [版本
(c) 2017 Microsoft Corpo

C:\Users\lmf>pip list
beautifulsoup4 (4.6.0)
certifi (2017.7.27.1)
chardet (3.0.4)
configparser (3.5.0)
idna (2.6)
MySQL-python (1.2.5)
pip (1.5.6)
py2exe (0.6.9)
pywin32 (221)
requests (2.18.4)
setuptools (7.0)
urllib3 (1.22)
web.py (0.37)
```

```
C:\Users\lmf>pip install matplotlib
Collecting matplotlib
  Downloading matplotlib-2.1.0-cp27-cp27m-win32.whl (8.2MB)
    2% |                                        | 194kB 17kB/s eta 0
```

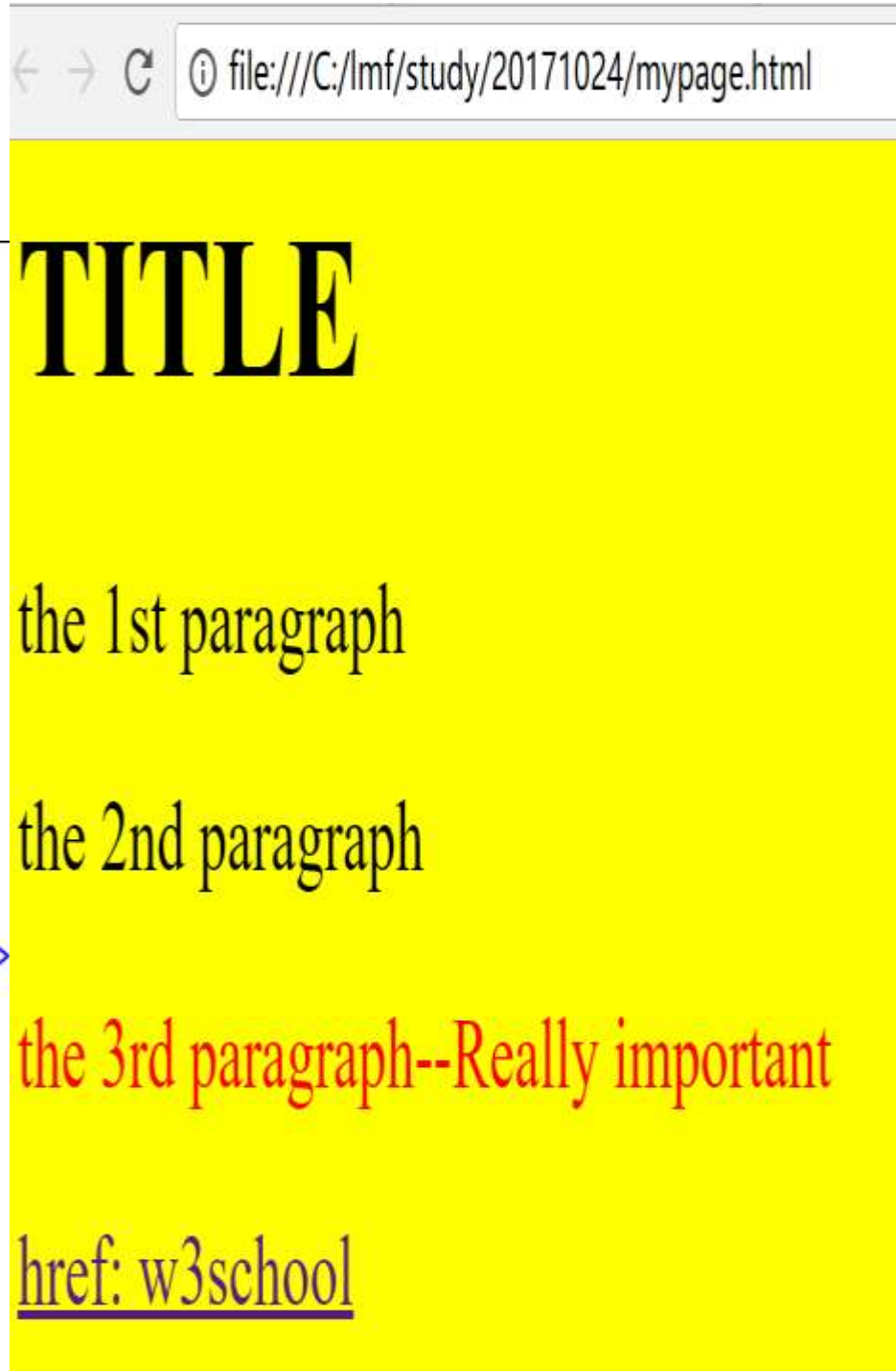
```
C:\Users\lmf>pip list --outdated
pip (Current: 1.5.6 Latest: 9.0.1)
py2exe (Current: 0.6.9 Latest: 0.9.2.2)
                                                satisfy
C:\Users\lmf>pip install matplotlib
Collecting matplotlib                                  36.6.0)
  Downloading matplotlib-2.1.0-cp27-cp27m-win32.whl (8.2MB)
    2% |                                        | 194kB 17kB/s eta 0)
```

```
C:\Users\lmf>pip install --upgrade pip
Downloading/unpacking pip from https://pypi.python.org/p
f196358da3/pip-9.0.1-py2.py3-none-any.whl#md5=297dbd16ef
Installing collected packages: pip
  Found existing installation: pip 1.5.6
    Uninstalling pip:
      Successfully uninstalled pip
Successfully installed pip
Cleaning up...
```

python

mypage.html

```
1 <html>
2 <body bgcolor='yellow'>
3 <h1>TITLE</h1>
4 <p>the 1st paragraph</p>
5 <p>the 2nd paragraph</p>
6 <p style="color:red">the 3rd paragraph--Really important</p>
7 <a href='http://www.w3school.com.cn/html/index.asp'>href: w3school</a>
8 </body>
9 </html>
```



```
>>> from bs4 import BeautifulSoup
>>> soup = BeautifulSoup(open('mypage.html'),'html.parser')

>>> soup.h1
<h1>TITLE</h1>
>>> soup.p
<p>the 1st paragraph</p>
>>> type(soup.h1)
<class 'bs4.element.Tag'>

>>> soup.h1.string
u'TITLE'
>>> soup.h1.string.encode('utf8')
'TITLE'
>>> type(soup.h1.string)
<class 'bs4.element.NavigableString'>
```

```
1 <html>
2 <body bgcolor='yellow'>
3 <h1>TITLE</h1>
4 <p>the 1st paragraph</p>
5 <p>the 2nd paragraph</p>
6 <p style="color:red">the 3rd paragraph--Really important</p>
7 <a href='http://www.w3school.com.cn/html/index.asp'>href: w3school</a>
8 </body>
9 </html>
```

```
>>> tag = soup.find('p')
>>> tag
<p>the 1st paragraph</p>
>>> tag.text
u'the 1st paragraph'
>>> soup.p.text
u'the 1st paragraph'
```

```
1 <html>
2 <body bgcolor='yellow'>
3 <h1>TITLE</h1>
4 <p>the 1st paragraph</p>
5 <p>the 2nd paragraph</p>
6 <p style="color:red">the 3rd paragraph--Really important</p>
7 <a href='http://www.w3school.com.cn/html/index.asp'>href: w3school</a>
8 </body>
9 </html>
```

```
>>> tags = soup.findAll('p')
>>> type(tags)
<class 'bs4.element.ResultSet'>
>>> for tag in tags: print tag.string
the 1st paragraph
the 2nd paragraph
the 3rd paragraph--Really important
```

```
>>> tag = soup.find('p',style="color:red")
>>> print tag.string
the 3rd paragraph--Really important
```

```
# -*- coding: utf-8 -*-
#============================
def getpage():
#-----------
def parsetext():
    from bs4 import BeautifulSoup
    import re
    soup = BeautifulSoup(open(fp),'html.parser') #    目前支持的可解析的文档类型: html, xml, 和 html5，最好明确指
    #直接soup.text.encode('utf8)存入文件可以，但结果会包括<script>和</script>之间的代码以及其它许多页面上呈现的并不
    title_list = []
    title_tagset = soup.findAll(href = re.compile("/view_")) #而使用find可以对要检索的内容精确筛检过滤
    for title in title_tagset: #得到每首诗的标题信息并存入到一个set
        #title--->>  <a href="/view_49386.aspx" style="font-size:18px; line-height:22px; height:22px; " targ
        #type(title)--->>  <class 'bs4.element.Tag'>)
        #title.string--->>      u'\u6c34\u8c03\u6b4c\u5934\xb7\u660e\u6708\u51e0\u65f6\u6709'
        #type(title.string)--->>      <class 'bs4.element.NavigableString'>
        s = title.string.encode('utf8')
        #type(s)--->>    <type 'str'>
        title_list.append(s)

    poet_list = []
    poet_tagset = soup.findAll('div', class_ = 'contson')
    for poet in poet_tagset:
    for m in range(len(title_list)):
        order = (np - 1) * 10 + m + 1
        fo.writelines(('\n' + str(order) + '. ' + title_list[m]+'\n' + poet_list[m]).replace('\n\n','\n'))
#========================================
fp = 'mypoet.html'
fo = file('poets1.txt','w')
for np in range(1, 2):#读取页数控制
    print np
    getpage()
    parsetext()
fo.close()
```

案例小结：

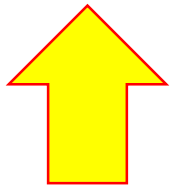
- 清晰的流程与模块：

数据获取→数据解析→数据加工→数据展示

- 数据获取：
  - 本地数据获取：
  - 网络数据获取：requests/urllib2/urllib
- 数据解析：
  - 字符串操作
  - BeautifulSoup与re

我的感受：

Simple & Strong



Open & Rich

数据库

Windows应用

嵌入和扩展

多媒体处理

科学计算

游戏编程

网络编程

GUI编程

企业与政务应用

- **语法基础 —— Start from “Hello World!”**
- **综合示例 —— When 苏轼 met Python**

注：讲座中示例均为python2.7版本。

活动与讲座

更多>>

11-02 星期四 16:30	■ 闲聊诗词之一：苏轼的诗词人生	预约
	已预约人数：51	
11-08 星期三 14:30	■ 军事装备与技术情报的获取与使用——以简氏	预约
	已预约人数：8	
11-09 星期四 16:30	■ 基于专利的技术领域现状分析	预约
	已预约人数：13	
11-16 星期四 16:30	■ 成果保护策略-专利申请与授权	预约
	已预约人数：17	

讲座名称
人文社会学科中外全文电子期刊查询与获取
科学数据的检索与利用
如何选择我的研究生导师
军事装备与技术情报的获取与使用——以简氏（Jane’s）数据库为例

讲座名称
国际学术期刊论文写作惯例
学位（毕业）论文写作规范及实用技巧
科学方法选择适合的期刊投稿

讲座名称
基于专利的技术领域现状分析
成果保护策略-专利申请与授权

讲座名称
新兴研究领域主题探测与演化分析：以“太阳能电池”为例

讲座名称
闲聊诗词之一：苏轼的诗词人生

讲座名称
人生苦短，我用Python

欢迎继续关注我们的2018信息素养系列讲座！

Guido van Rossum-1989-Christmas------

Happy Christmas!

←请扫码
签到