## 第4章 Python爬蟲常用模組

<https://docs.python.org/2.7/py-modindex.html> 標準模組庫

4.1 網路爬蟲技術核心

4.1.1 網路爬蟲實現原理

4.1.2 爬行策略 119

4.1.3 身份識別 119

4.2 Python 3標準庫之urllib.request模組

### 4.2 Python標準庫之urllib2模組

<https://docs.python.org/2.7/library/urllib2.html#module-urllib2>

這模組只適用python2.7

4.2.1 urllib.request請求返回網頁

Urllib2.urlopen(url[, data[, timeout[, cafile[, capath[, cadefault[, context]]]]]])

ca: 身分驗證相關

data: 以post提交URL

* geturl()返回response的url資訊，常用於url重定向
* info()返回response的基本資訊
* getcode()返回response的狀態碼，常見200, 404, 503

|  |
| --- |
| #!/usr/bin/env python  #-\*- coding: utf-8 -\*-  \_\_author\_\_ = 'hstking hstking@hotmail.com'  import urllib2  def linkBaidu():  url = 'http://www.baidu.com'  try:  response = urllib2.urlopen(url,timeout=3)  except urllib2.URLError:  print(u"網路位址錯誤")  exit()  with open('./baidu.txt','w') as fp:  fp.write(response.read())  print(u"取得url訊息，response.geturl() \n: %s" %response.geturl())  print(u"取得返回程式碼，response.getcode() \n: %s" %response.getcode())  print(u"取得返回訊息，response.info() \n: %s" %response.info())  print(u"取得的網頁內容已存入現在目錄的baidu.txt中，請自行查看")  if \_\_name\_\_ == '\_\_main\_\_':  linkBaidu() |

|  |
| --- |
| py -2 testUrllib2.py  取得url訊息，response.geturl()  : http://www.baidu.com  取得返回程式碼，response.getcode()  : 200  取得返回訊息，response.info()  : Date: Mon, 25 Feb 2019 19:25:51 GMT  Content-Type: text/html  Transfer-Encoding: chunked  Connection: Close  Vary: Accept-Encoding  Set-Cookie: BAIDUID=29742DA314E80B8986E5A80B2D3AA18D:FG=1; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=.baidu.com  Set-Cookie: BIDUPSID=29742DA314E80B8986E5A80B2D3AA18D; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=.baidu.com  Set-Cookie: PSTM=1551122751; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=.baidu.com  Set-Cookie: delPer=0; path=/; domain=.baidu.com  Set-Cookie: BDSVRTM=0; path=/  Set-Cookie: BD\_HOME=0; path=/  Set-Cookie: H\_PS\_PSSID=1461\_21091\_28557; path=/; domain=.baidu.com  P3P: CP=" OTI DSP COR IVA OUR IND COM "  Cxy\_all: baidu+d4658988d71f2ffd16f2603dac1eb821  Cache-Control: private  Expires: Mon, 25 Feb 2019 19:25:42 GMT  Server: BWS/1.1  X-UA-Compatible: IE=Edge,chrome=1  BDPAGETYPE: 1  BDQID: 0xe77143e00000b163  取得的網頁內容已存入現在目錄的baidu.txt中，請自行查看 |

4.2.2 urllib.request使用代理訪問網頁 122

|  |
| --- |
| #!/usr/bin/env python  #-\*- coding: utf-8 -\*-  \_\_author\_\_ = 'hstking hstking@hotmail.com'  import urllib2  import sys  import re  def testArgument():  '''測試輸入參數，只需要一個參數 '''  if len(sys.argv) != 2:  print(u'只需要一個參數就夠了')  tipUse()  exit()  else:  TP = TestProxy(sys.argv[1])  def tipUse():  '''顯示提示訊息 '''  print(u'該程式只能輸入一個參數，這個參數必須是一個可用的proxy')  print(u'usage: python testUrllib2WithProxy.py http://1.2.3.4:5')  print(u'usage: python testUrllib2WithProxy.py https://1.2.3.4:5')  class TestProxy(object):  '''這個類別的作用是測試proxy是否有效 '''  def \_\_init\_\_(self,proxy):  self.proxy = proxy  self.checkProxyFormat(self.proxy)  self.url = 'http://www.baidu.com'  self.timeout = 5  self.flagWord = '百度' #在網頁返回的資料中查詢這個關鍵詞  self.useProxy(self.proxy)  def checkProxyFormat(self,proxy):  try:  proxyMatch = re.compile('http[s]?://[\d]{1,3}\.[\d]{1,3}\.[\d]{1,3}\.[\d]{1,3}:[\d]{1,5}$')  re.search(proxyMatch,proxy).group()  except AttributeError:  tipUse()  exit()  flag = 1  proxy = proxy.replace('//','')  try:  protocol = proxy.split(':')[0]  ip = proxy.split(':')[1]  port = proxy.split(':')[2]  except IndexError:  print(u'下標出界')  tipUse()  exit()  flag = flag and len(proxy.split(':')) == 3 and len(ip.split('.')) == 4  flag = ip.split('.')[0] in map(str,xrange(1,256)) and flag  flag = ip.split('.')[1] in map(str,xrange(256)) and flag  flag = ip.split('.')[2] in map(str,xrange(256)) and flag  flag = ip.split('.')[3] in map(str,xrange(1,255)) and flag  flag = protocol in [u'http',u'https'] and flag  flag = port in map(str,range(1,65535)) and flag  '''這裡是在檢查proxy的格式 '''  if flag:  print(u'輸入的http代理服務器符合標準')  else:  tipUse()  exit()  def useProxy(self,proxy):  '''利用代理訪問百度，並查詢關鍵詞 '''  protocol = proxy.split('//')[0].replace(':','')  ip = proxy.split('//')[1]  opener = urllib2.build\_opener(urllib2.ProxyHandler({protocol:ip}))  urllib2.install\_opener(opener)  try:  response = urllib2.urlopen(self.url,timeout = self.timeout)  except:  print(u'連接錯誤，退出程式')  exit()  str = response.read()  if re.search(self.flagWord,str):  print(u'已取得特徵詞，該代理可用')  else:  print(u'該代理不可用')  if \_\_name\_\_ == '\_\_main\_\_':  testArgument() |

>Python testUrllib2withProxy.py http://119.9.89.61:8088

**4.2.3 urllib.request修改header**

預設Python-urllib/x.y(版本)

網站是透過瀏覽器發送過來的User Agent來確認身分

userAgents.py: 將常見User-Agent以字典形式放進模組

|  |
| --- |
| testUrllib2ModifyHeader.py  … |

>python testUrllib2ModifyHeader.py

比較1.html, 2.html 🡪 網站會給不同瀏覽器返回不同內容

### 4.3 Python標準庫之logging模組

取代debug功能

預設層級: NOTSET=0, DEBUG=10, INFO=20, WARNING=30, ERROR=40, CRITCAL=50

預設定義層級為WARNING

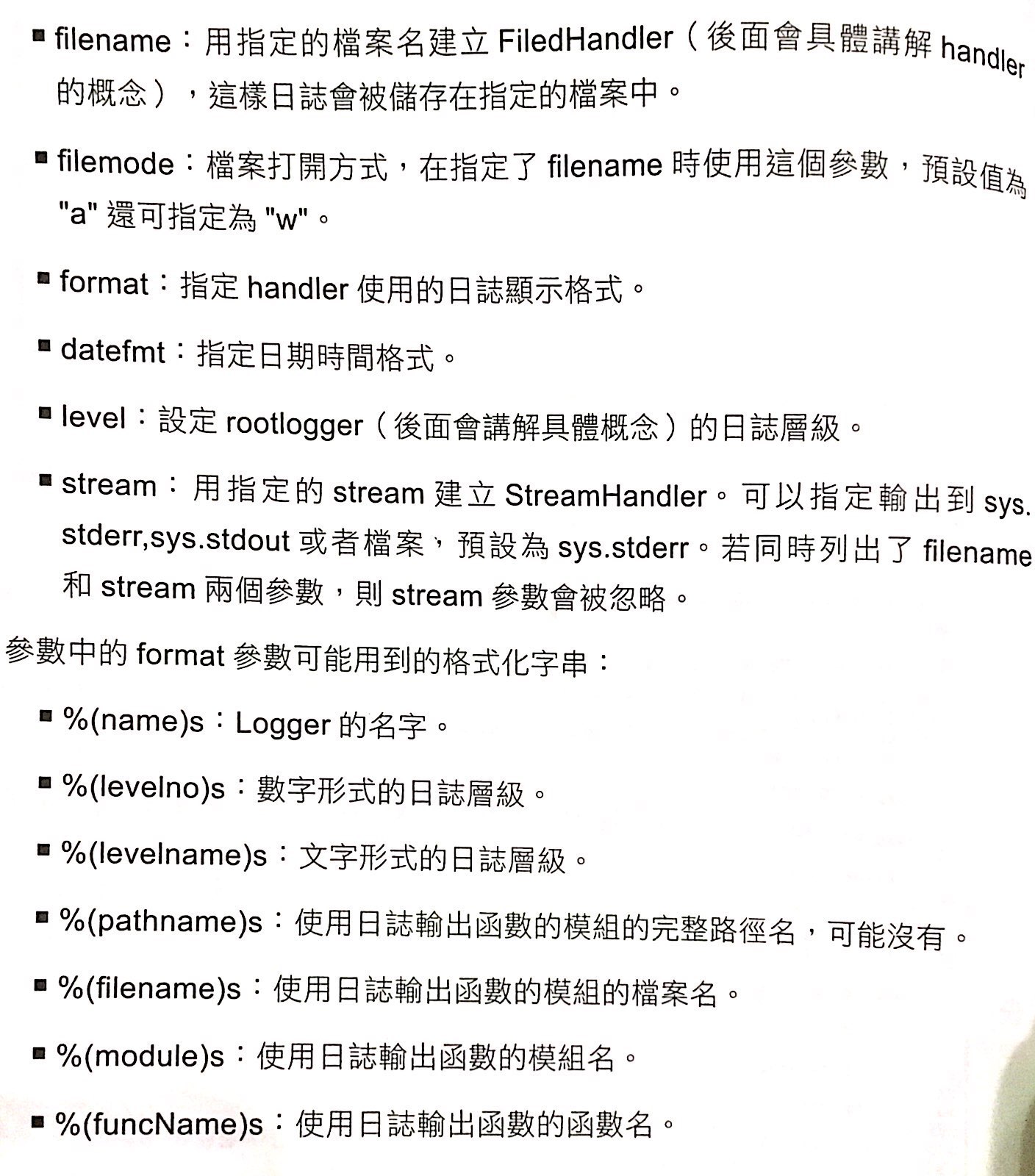
如果logging模組發出高於定義的層級, 則顯示於螢幕

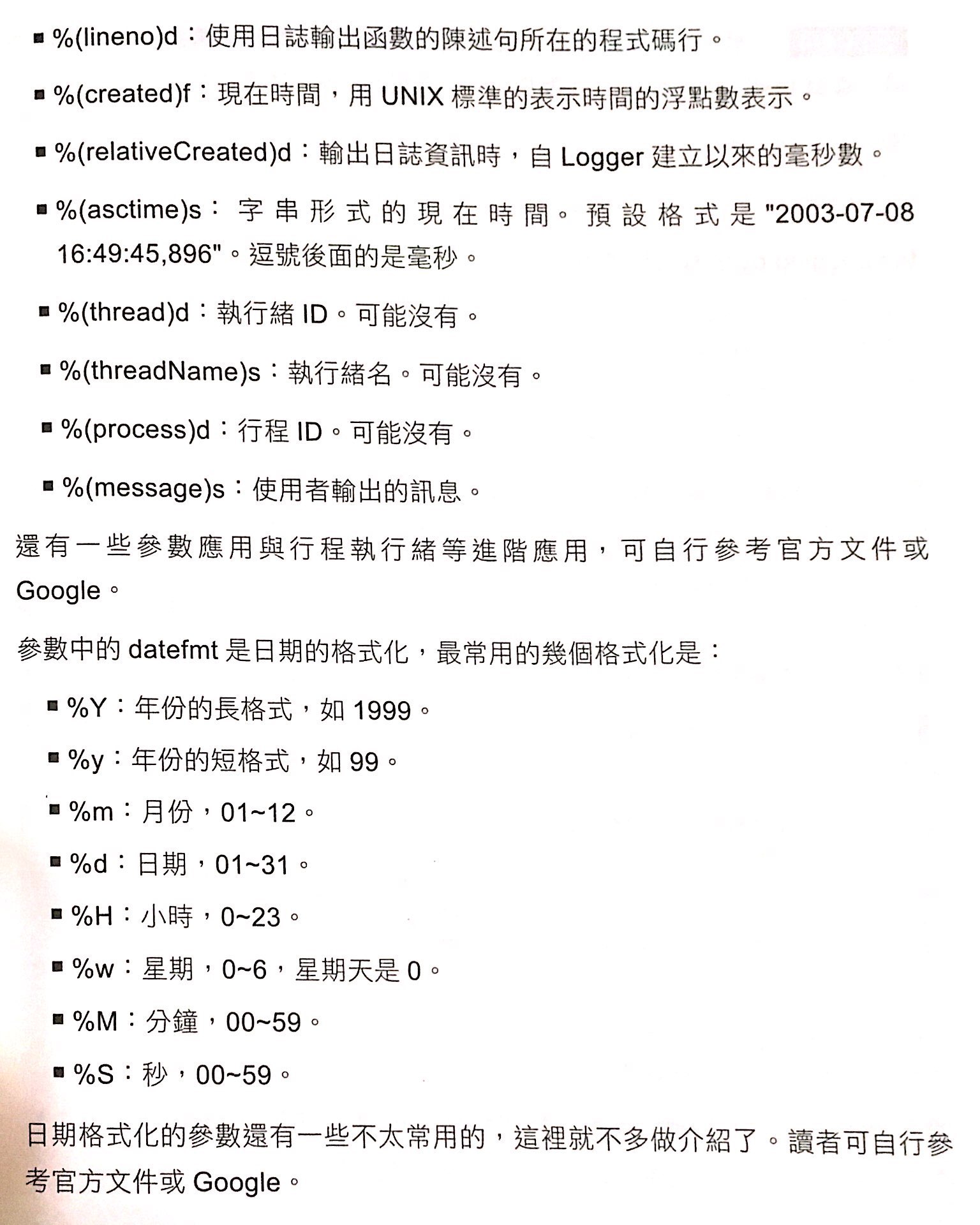
Import logging

Logging.DEBUG 查看層級數字

Logging.debug(“debug message”)

Logging.basicConfig([\*\*kwargs])





|  |
| --- |
| Testlogging.py  … |

>python testlogging.py

>cat testLog.txt

**4.3.2 自訂模組myLog**

|  |
| --- |
| myLog.py  … |

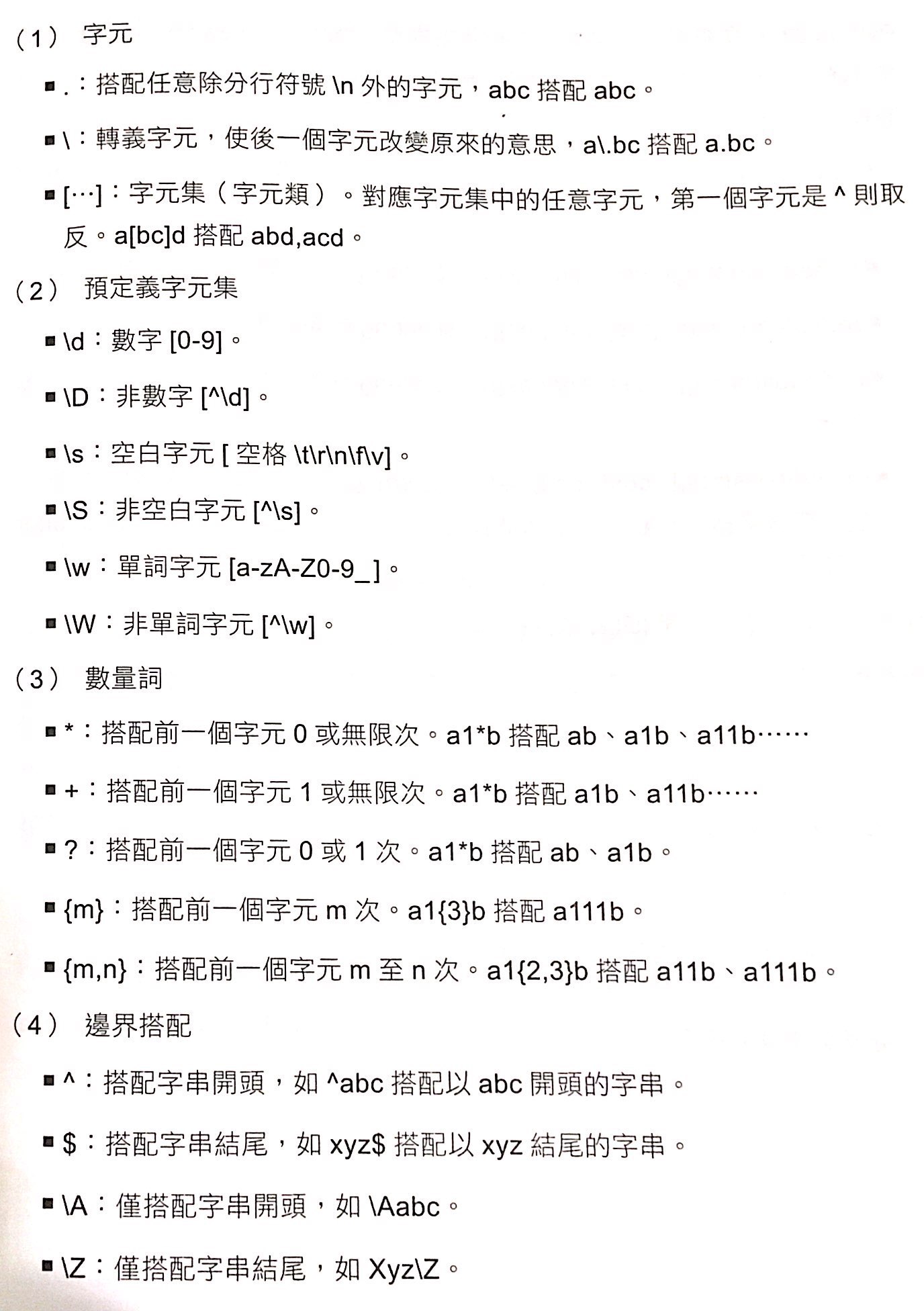
|  |
| --- |
| testMyLog.py  … |

>python testMyLog.py

>cat testMyLog.log

### 4.4 re模組（正則表示式）

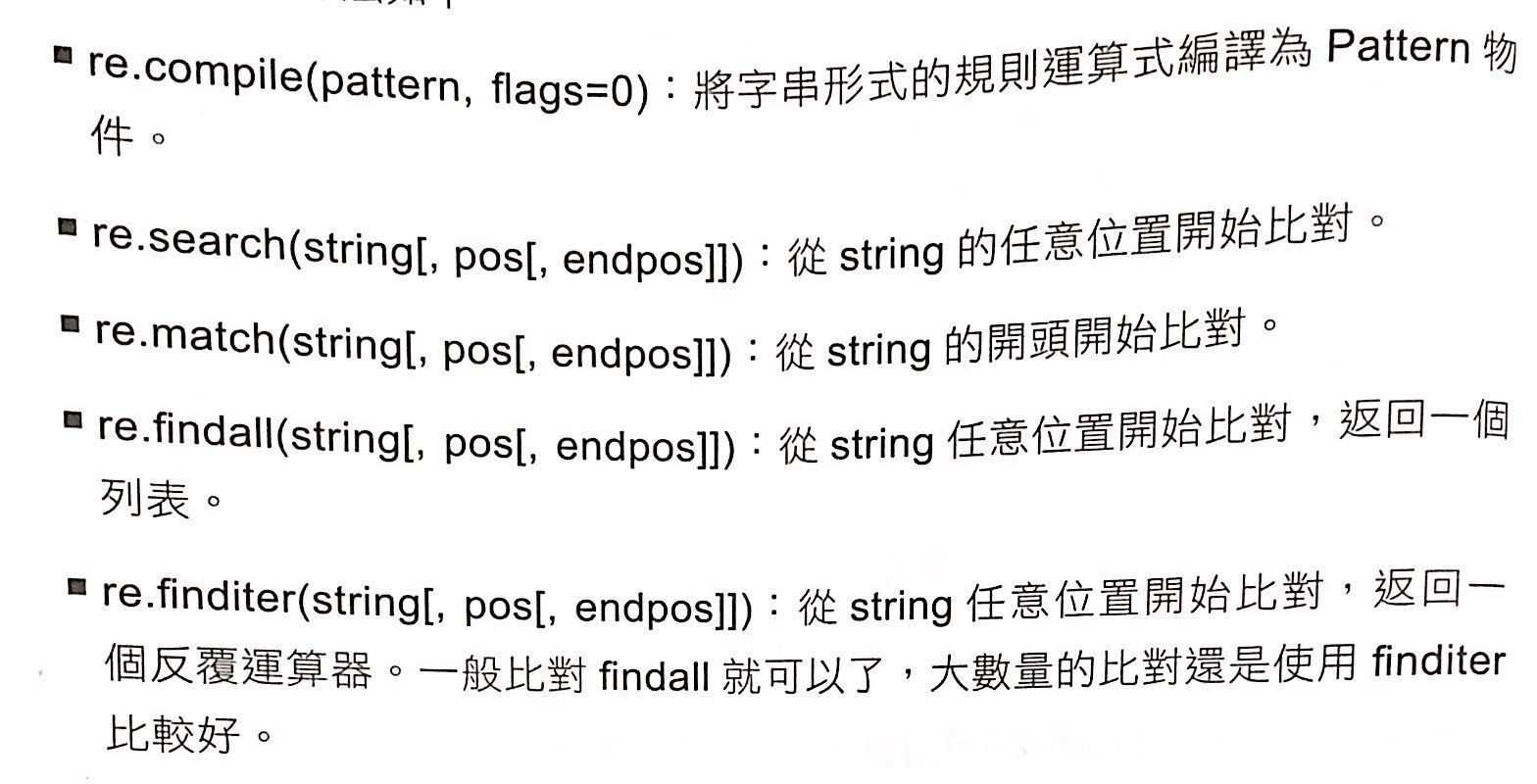
大陸用語:規則運算式



操作:

* match: 從字串起點開始比對
* search: 對字串做任意比對

常用方法:



|  |
| --- |
| str=”adslfa”  re.search(‘am’, str)  re.search(‘am’, str).group()  re.match(‘am’, str)  re.match(‘am’, str).group()  re.findall(‘modules’, str)  re.finditer(‘modules’,str)  for sre in re.finditer(‘modules’,,, str):  print(sre.group()) |

範例: 找電影資訊

|  |
| --- |
| simpleCrawlerTodayMovie.py  … |

當爬取內容不多才可用, 改用爬蟲框架

### 4.5 sys模組（系統參數獲取）

sys.argv 返回參數

sys.exit 退出程式

|  |
| --- |
| testSys.py |

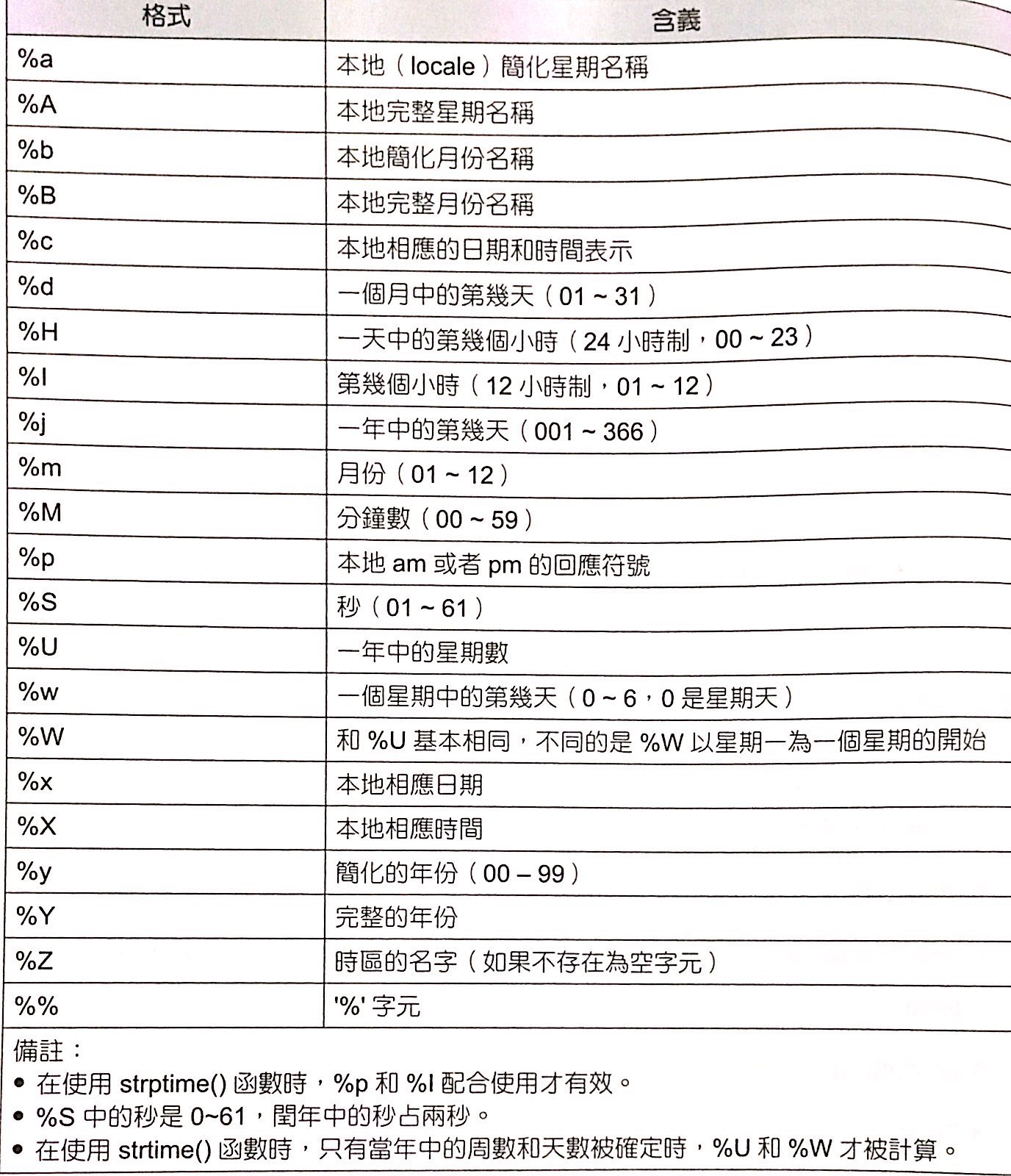
### 4.6 time模組（獲取時間資訊）

time.time() 現在時間

time.localtime([secs]) 現在時間轉換現在時區struct\_time

time.sleep(secs) 計時器

Time.strftime(format[, t]) 格式化時間字串



|  |
| --- |
| import time  time.time()  time.localtime()  for I in xrange(5):  time.sleep(1)  print i  time.strftime(‘%Y-%m-%d %X’, time.localtime()) |

|  |
| --- |
| testTime.py |

## 第5章 Scrapy爬蟲框架

### 5.1 安裝Scrapy

pip install scrapy (Windows)

py -2 -m pip install scrapy (雙版本時)

Ubuntu:

sudo pip uninstall scrapy

sudo pip install scrapy

apt-get install python-scrapy

vimrc檔設置

set tabstop=4 tab為4個空格

set number 顯示行號

set noexpandtab tab不轉換空格

ps.範例使用scrapyProject目錄

### 5.2 Scrapy選擇器XPath和CSS

#### 5.2.1 XPath選擇器

利用Xpath採集資料

superHero.xml

常用路徑運算式

nodeName 選取此節點的所有子節點

/ 從根節點選取

// 從比對選擇的現在節點選擇檔案中的節點, 不考慮位置

. 選取現在節點

.. 選取現在節點的父節點

@ 選取屬性

\* 比對任何元素節點

@\* 比對任何屬性節點

Node() 比對任何類型的節點

|  |
| --- |
| python  from scrapt.selector import Selector  with open(‘./superHero.xml’, ‘r’) as fpt  body = fp.read()  Selector(text=body).xpath(‘/\*’).extract() |

ps. 範例檔不是標準html檔, 所以在選擇棄被自動新增<html>和<body>

隱藏不深的資料, 直接用Xpath選擇器挑選資料, 複雜的用配套選擇

#### 5.2.2 CSS選擇器

CSS規則由兩個主要部分構成: 選擇器, 一條或多條宣告

selector {declaration1; declaration2; …..declarationN}

…

Selector(text=body).css(‘class’).extract()

Selector(text=body).css(‘class name’).extract()

Selector(text=body).css(‘class name’).extract()[0]

Selector(text=body).css([‘lang=”en”]’).extract()

CSS選擇器和Xpath選擇器都可巢狀使用, 可互相巢狀

#### 5.2.3 其他選擇器 152

xpath.re()方法, 不推薦

### 5.3 Scrapy爬蟲實戰一：今日影視

5.3.1 創建Scrapy項目 153

scrapy startproject todayMoive 創建一個名為todayMoive新項目

tree todayMoive 以樹狀結構顯示檔案目錄結構

|  |
| --- |
| $ tree todayMoive/  todayMoive/  ├── scrapy.cfg  └── todayMoive  ├── \_\_init\_\_.py  ├── items.py  ├── middlewares.py  ├── pipelines.py  ├── settings.py  └── spiders  └── \_\_init\_\_.py |

cd todayMoive

scrapy genspider wuHanMovieSpider jycinema.com 建立wuHanMovieSpider腳本

tree ../todayMoive

|  |
| --- |
| $ tree todayMoive/  todayMoive/  ├── scrapy.cfg  └── todayMoive  ├── \_\_init\_\_.py  ├── \_\_init\_\_.pyc  ├── items.py  ├── middlewares.py  ├── pipelines.py  ├── settings.py  ├── settings.pyc  └── spiders  ├── \_\_init\_\_.py  ├── \_\_init\_\_.pyc  └── wuHanMovieSpider.py |

* scrapy.cfg:項目的配置文件
* todayMoive:該項目的python模塊。之後將在此加入代碼。
* todayMoive/items.py:項目中的item文件。作用是定義爬蟲最終需要哪些項
* todayMoive/pipelines.py:項目中的pipelines文件。作用是爬取網頁內容後，內容怎麼處理
* todayMoive/spiders/:放置spider代碼的目錄。
* todayMoive/middlewares.py:項目中的自定義中間件文件。

scrapy.cfg

|  |
| --- |
| [settings] # 定義默認設置文件的位置為todayMoive模塊下的settings文件  default = todayMoive.settings  [deploy] # 定義項目名稱為todayMoive  #url = http://localhost:6800/  project = todayMoive |

settings.py

|  |
| --- |
| BOT\_NAME = 'todayMoive' # 此Scrapy項目實施的bot的名稱（也稱為項目名稱）。這將用於默認情況下構造User-Agent，也用於日誌記錄。  SPIDER\_MODULES = ['todayMoive.spiders']  NEWSPIDER\_MODULE = 'todayMoive.spiders'  # Crawl responsibly by identifying yourself (and your website) on the user-agent  #USER\_AGENT = 'todayMoive (+http://www.yourdomain.com)'  # Obey robots.txt rules  ROBOTSTXT\_OBEY = True  … |

**items.py**

|  |
| --- |
| import scrapy  class TodaymoiveItem(scrapy.Item):  # define the fields for your item here like:  # name = scrapy.Field()  pass |

**pipelines.py**

|  |
| --- |
| class TodaymoivePipeline(object):  def process\_item(self, item, spider):  return item |

#### **創建Scrapy項目抓取靜態網頁內容**

items.py

|  |
| --- |
| import scrapy  class TodaymoiveItem(scrapy.Item):  # define the fields for your item here like:  # name = scrapy.Field()  activityName = scrapy.Field() |

pipelines.py

|  |
| --- |
| import time  class TodaymoivePipeline(object):  def process\_item(self, item, spider):  now = time.strftime('%Y-%m-%d', time.localtime())  fileName = now + '.txt'  with open(fileName, 'a') as fp:  fp.write(item['activityName'][0] + '\n\n')  return item |

settings.py

|  |
| --- |
| BOT\_NAME = 'todayMoive'  SPIDER\_MODULES = ['todayMoive.spiders']  NEWSPIDER\_MODULE = 'todayMoive.spiders'  ITEM\_PIPELINES = {'todayMoive.pipelines.TodaymoivePipeline':300}  # Crawl responsibly by identifying yourself (and your website) on the user-agent  #USER\_AGENT = 'todayMoive (+http://www.yourdomain.com)'  # Obey robots.txt rules  ROBOTSTXT\_OBEY = False |

todayMoive/spider/MoiveSpider.py

|  |
| --- |
| # -\*- coding: utf-8 -\*-  **import** scrapy  **from** todayMoive.items **import** TodaymoiveItem  **class** **MoivespiderSpider**(scrapy.Spider):  name = 'MoiveSpider'  allowed\_domains = ['jycinema.com']  start\_urls = ['http://www.jycinema.com/html/default/schedule.html?cinemaId=451']  **def** **parse**(self, response):  subSelector = response.xpath('//span[@class="f16"]')  items = []  **for** sub **in** subSelector:  print(sub)  item = TodaymoiveItem()  item['activityName'] = sub.xpath('text()').extract()  print(item)  items.append(item)  **return** items |

$ scrapy genspider –h

Usage

=====

scrapy genspider [options] <name> <domain>

Generate new spider using pre-defined templates

Options

=======

--help, -h show this help message and exit

--list, -l List available templates

--edit, -e Edit spider after creating it

--dump=TEMPLATE, -d TEMPLATE

Dump template to standard output

--template=TEMPLATE, -t TEMPLATE

Uses a custom template.

--force If the spider already exists, overwrite it with the

template

Global Options

--------------

--logfile=FILE log file. if omitted stderr will be used

--loglevel=LEVEL, -L LEVEL

log level (default: DEBUG)

--nolog disable logging completely

--profile=FILE write python cProfile stats to FILE

--pidfile=FILE write process ID to FILE

--set=NAME=VALUE, -s NAME=VALUE

set/override setting (may be repeated)

--pdb enable pdb on failure

5.3.2 Scrapy檔介紹 155

5.3.3 Scrapy爬蟲編寫 157

5.4 Scrapy爬蟲實戰二：天氣預報 164

5.4.1 專案準備 165

5.4.2 創建編輯Scrapy爬蟲 166

5.4.3 資料存儲到json 173

5.4.4 資料存儲到MySQL 175

5.5 Scrapy爬蟲實戰三：獲取代理 182

5.5.1 專案準備 182

5.5.2 創建編輯Scrapy爬蟲 183

5.5.3 多個Spider 188

5.5.4 處理Spider資料 192

5.6 Scrapy爬蟲實戰四：糗事百科 194

5.6.1 目標分析 195

5.6.2 創建編輯Scrapy爬蟲 195

5.6.3 Scrapy項目中介軟體——添加headers 196

5.6.4 Scrapy項目中介軟體——添加proxy 200

5.7 Scrapy爬蟲實戰五：爬蟲攻防 202

5.7.1 創建一般爬蟲 202

5.7.2 封鎖間隔時間破解 206

5.7.3 封鎖Cookies破解 206

5.7.4 封鎖User-Agent破解 207

5.7.5 封鎖IP破解 212

5.8 本章小結 215

## 第6章 Beautiful Soup爬蟲 216

6.1 安裝Beautiful Soup環境 216

6.1.1 Windows下安裝Beautiful Soup 216

6.1.2 Linux下安裝Beautiful Soup 217

6.1.3 最強大的IDE——Eclipse 218

6.2 Beautiful Soup解析器 227

6.2.1 bs4解析器選擇 227

6.2.2 lxml解析器安裝 227

6.2.3 使用bs4篩檢程式 229

6.3 bs4爬蟲實戰一：獲取百度貼吧內容 234

6.3.1 目標分析 234

6.3.2 專案實施 236

6.3.3 代碼分析 243

6.3.4 Eclipse調試 244

6.4 bs4爬蟲實戰二：獲取雙色球中獎資訊 245

6.4.1 目標分析 246

6.4.2 專案實施 248

6.4.3 保存結果到Excel 251

6.4.4 代碼分析 256

6.5 bs4爬蟲實戰三：獲取起點小說資訊 257

6.5.1 目標分析 257

6.5.2 專案實施 259

6.5.3 保存結果到MySQL 261

6.5.4 代碼分析 265

6.6 bs4爬蟲實戰四：獲取電影資訊 266

6.6.1 目標分析 266

6.6.2 專案實施 267

6.6.3 bs4反爬蟲 270

6.6.4 代碼分析 273

6.7 bs4爬蟲實戰五：獲取音悅台榜單 273

6.7.1 目標分析 273

6.7.2 專案實施 274

6.7.3 代碼分析 279

6.8 本章小結 280

## 第7章 Mechanize模擬流覽器 281

7.1 安裝Mechanize模組 281

7.1.1 Windows下安裝Mechanize 281

7.1.2 Linux下安裝Mechanize 282

7.2 Mechanize 測試 283

7.2.1 Mechanize百度 283

7.2.2 Mechanize光貓F460 286

7.3 Mechanize實站一：獲取Modem資訊 290

7.3.1 獲取F460資料 290

7.3.2 代碼分析 293

7.4 Mechanize實戰二：獲取音悅台公告 293

7.4.1 登錄原理 293

7.4.2 獲取Cookie的方法 294

7.4.3 獲取Cookie 298

7.4.4 使用Cookie登錄獲取資料 302

7.5 本章小結 305

## 第8章 Selenium模擬流覽器 306

8.1 安裝Selenium模組 306

8.1.1 Windows下安裝Selenium模組 306

8.1.2 Linux下安裝Selenium模組 307

8.2 流覽器選擇 307

8.2.1 Webdriver支持列表 307

8.2.2 Windows下安裝PhantomJS 308

8.2.3 Linux下安裝PhantomJS 310

8.3 Selenium&PhantomJS抓取數據 312

8.3.1 獲取百度搜索結果 312

8.3.2 獲取搜索結果 314

8.3.3 獲取有效資料位置 317

8.3.4 從位置中獲取有效資料 319

8.4 Selenium&PhantomJS實戰一：獲取代理 319

8.4.1 準備環境 320

8.4.2 爬蟲代碼 321

8.4.3 代碼解釋 324

8.5 Selenium&PhantomJS實戰二：漫畫爬蟲 324

8.5.1 準備環境 325

8.5.2 爬蟲代碼 326

8.5.3 代碼解釋 329

8.6 本章小結 329