DATA ACQUISITION 2

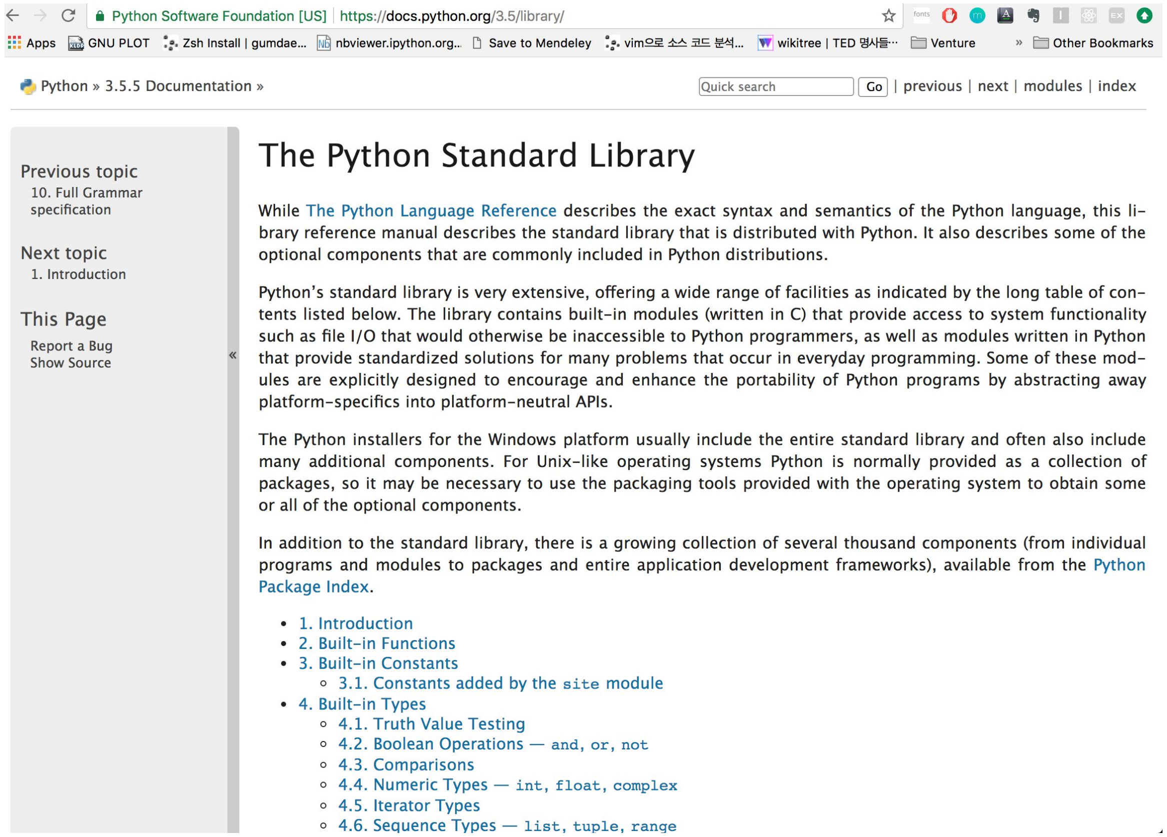
project with the latest technology

Lee Hae Joon

Motivation

- last week, we did experience scrapping by using ‘BeautifulSoup’
- this week, we will go through recursive downloading a site with <a> tag

Contents


Python Software Foundation [US] | https://docs.python.org/3.5/library/
Apps GNU PLOT Zsh Install | gumdae... nbviewer.ipython.org... Save to Mendeley vim으로 소스 코드 분석... wikitree | TED 명사들… Venture Other Bookmarks
Python » 3.5.5 Documentation »
Quick search Go | previous | next | modules | index

Previous topic
10. Full Grammar specification

Next topic
1. Introduction

This Page
Report a Bug
Show Source

The Python Standard Library

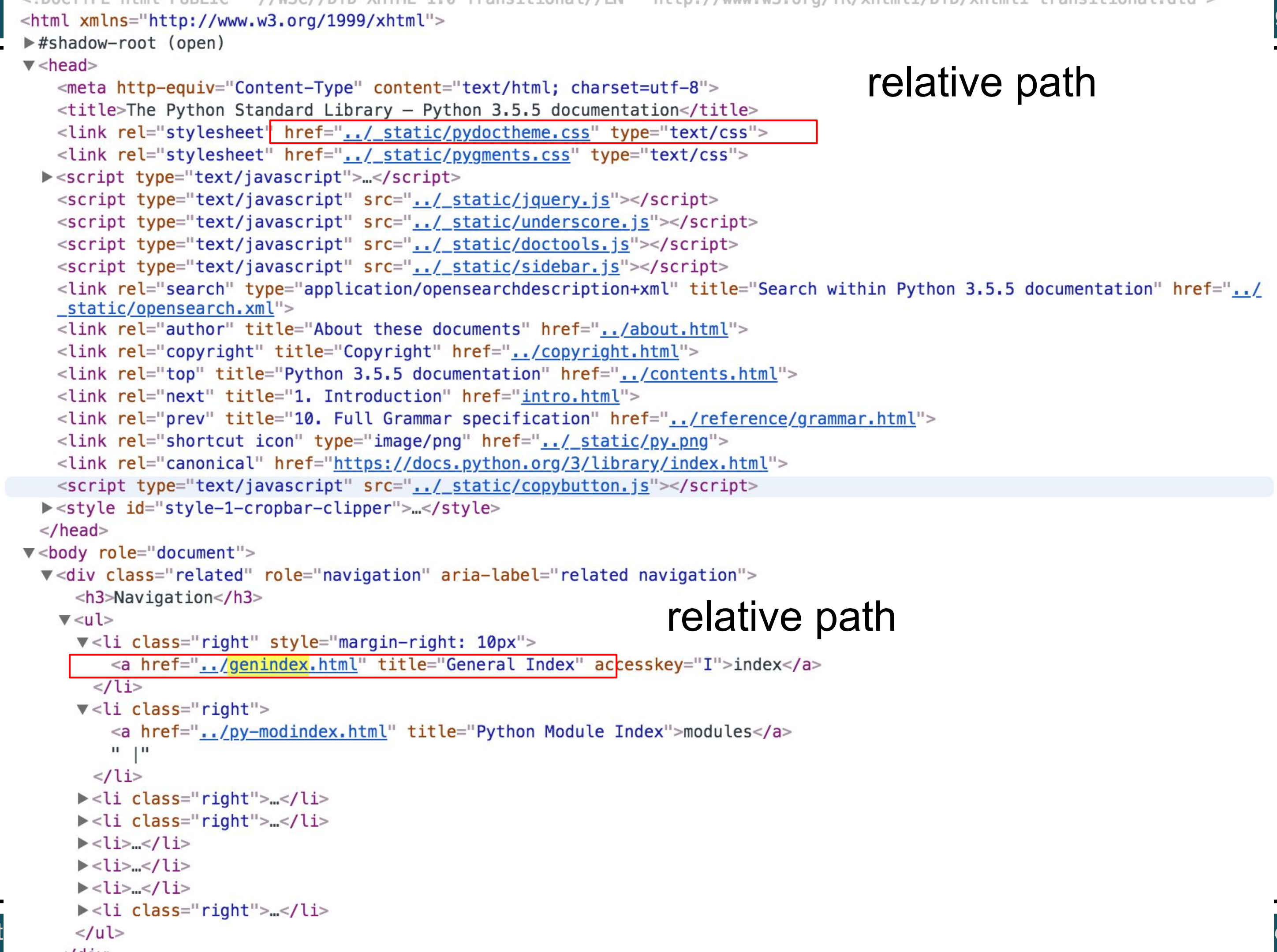
While The Python Language Reference describes the exact syntax and semantics of the Python language, this library reference manual describes the standard library that is distributed with Python. It also describes some of the optional components that are commonly included in Python distributions.

Python's standard library is very extensive, offering a wide range of facilities as indicated by the long table of contents listed below. The library contains built-in modules (written in C) that provide access to system functionality such as file I/O that would otherwise be inaccessible to Python programmers, as well as modules written in Python that provide standardized solutions for many problems that occur in everyday programming. Some of these modules are explicitly designed to encourage and enhance the portability of Python programs by abstracting away platform-specifics into platform-neutral APIs.

The Python installers for the Windows platform usually include the entire standard library and often also include many additional components. For Unix-like operating systems Python is normally provided as a collection of packages, so it may be necessary to use the packaging tools provided with the operating system to obtain some or all of the optional components.

In addition to the standard library, there is a growing collection of several thousand components (from individual programs and modules to packages and entire application development frameworks), available from the Python Package Index.

https://docs.python.org/3.5/library/

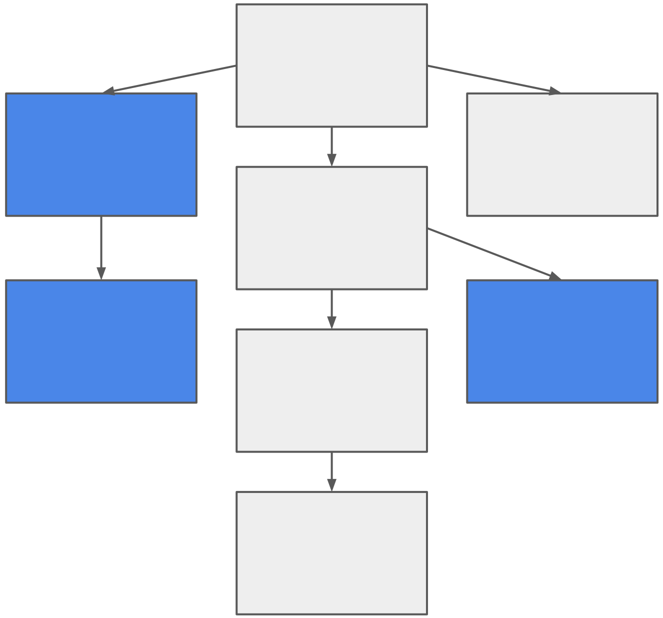


relative path

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
 ▶#shadow-root (open)
 ▼<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>The Python Standard Library — Python 3.5.5 documentation</title>
    <link rel="stylesheet" href="../_static/pydoctheme.css" type="text/css">
    <link rel="stylesheet" href="../_static/pygments.css" type="text/css">
   ▶<script type="text/javascript">…</script>
    <script type="text/javascript" src="../_static/jquery.js"></script>
    <script type="text/javascript" src="../_static/underscore.js"></script>
    <script type="text/javascript" src="../_static/doctools.js"></script>
    <script type="text/javascript" src="../_static/sidebar.js"></script>
    <link rel="search" type="application/opensearchdescription+xml" title="Search within Python 3.5.5 documentation" href="../
    _static/opensearch.xml">
    <link rel="author" title="About these documents" href="../about.html">
    <link rel="copyright" title="Copyright" href="../copyright.html">
    <link rel="top" title="Python 3.5.5 documentation" href="../contents.html">
    <link rel="next" title="1. Introduction" href="intro.html">
    <link rel="prev" title="10. Full Grammar specification" href="../reference/grammar.html">
    <link rel="shortcut icon" type="image/png" href="../_static/py.png">
    <link rel="canonical" href="https://docs.python.org/3/library/index.html">
    <script type="text/javascript" src="../_static/copybutton.js"></script>
   ▶<style id="style-1-cropbar-clipper">…</style>
  </head>
 ▼<body role="document">
   ▼<div class="related" role="navigation" aria-label="related navigation">
      <h3>Navigation</h3>
     ▼<ul>
       ▼<li class="right" style="margin-right: 10px">
          <a href="../genindex.html" title="General Index" accesskey="I">index</a>
        </li>
       ▼<li class="right">
          <a href="../py-modindex.html" title="Python Module Index">modules</a>
          " |"
        </li>
       ▶<li class="right">…</li>
       ▶<li class="right">…</li>
       ▶<li>…</li>
       ▶<li>…</li>
       ▶<li>…</li>
       ▶<li class="right">…</li>
      </ul>
    </div>
```

relative path

Why do we need a recursive way?



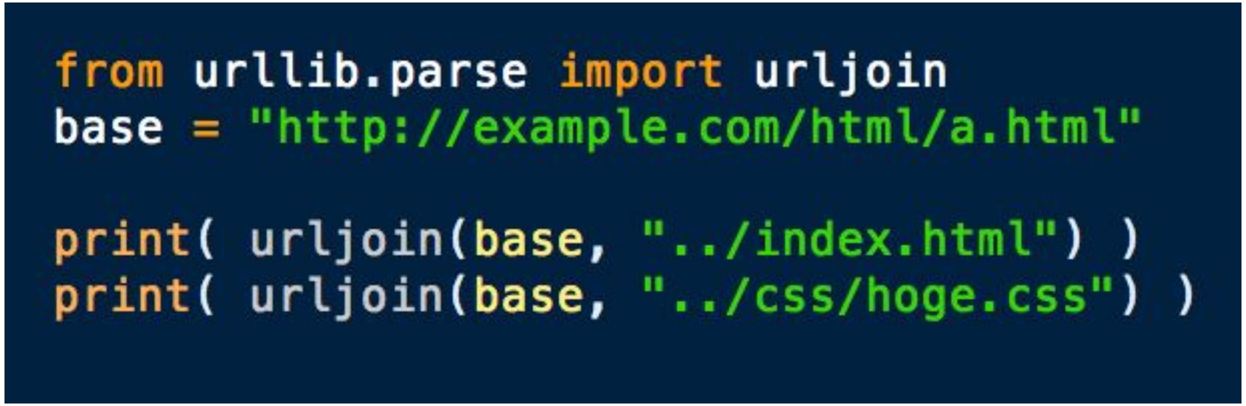
https://docs.python.org/3.5/library/

1. analyze a html file
2. extract links in the html
3. downloads the links if they are a file
4. if the file is a html file, go back to the first task

Crawling

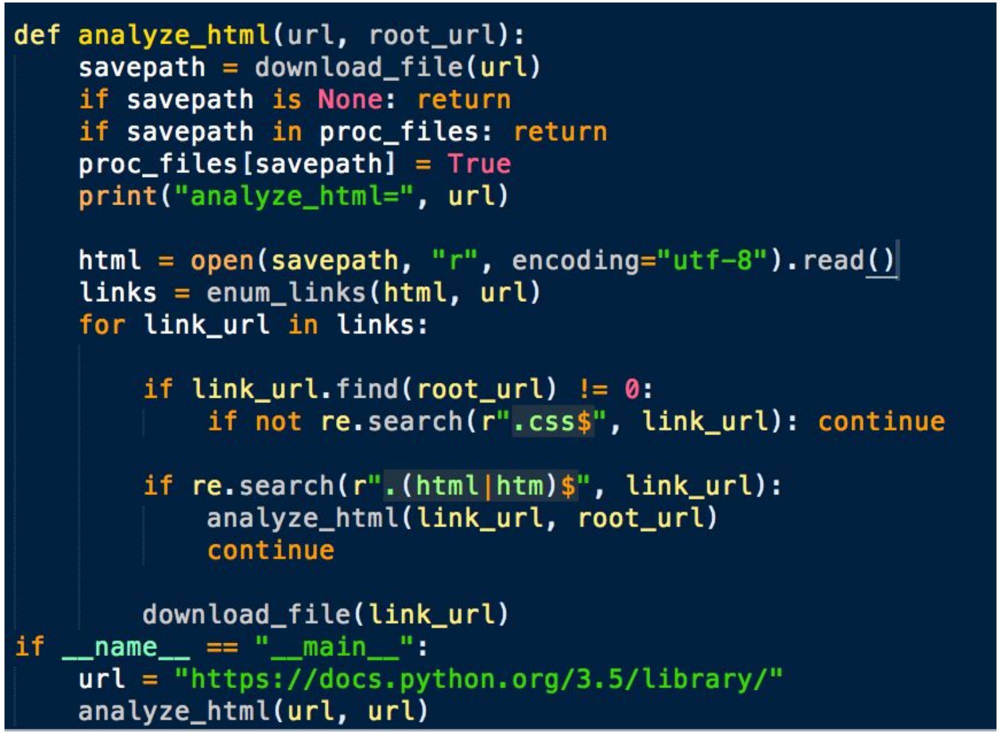
urljoin

- relative path -> absolute path



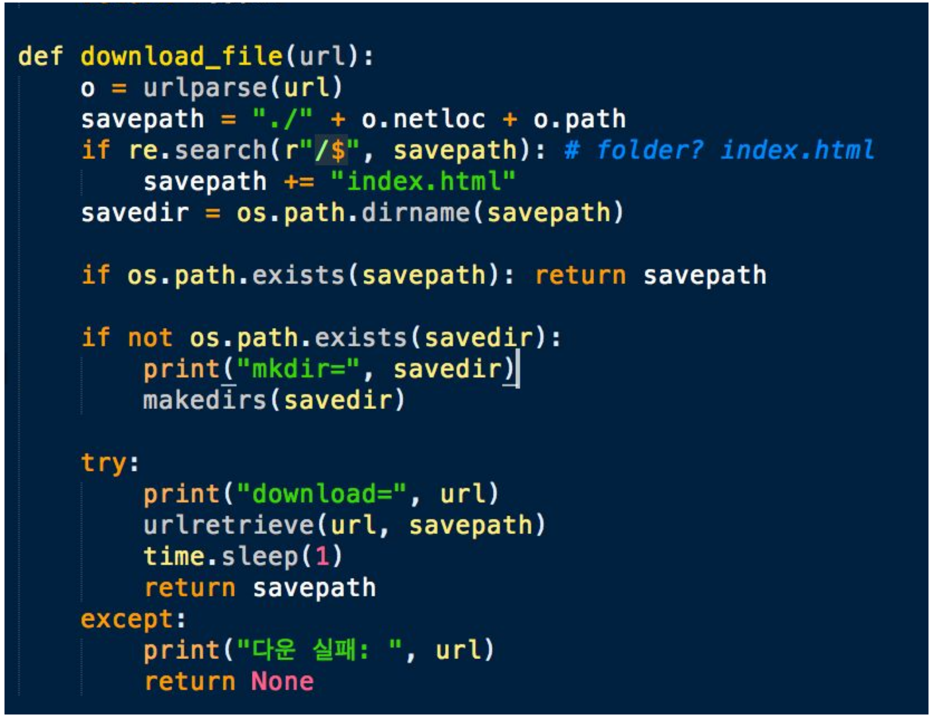
```python
from urllib.parse import urljoin
base = "http://example.com/html/a.html"

print( urljoin(base, "../index.html") )
print( urljoin(base, "../css/hoge.css") )
```

Recursive Downloading a site



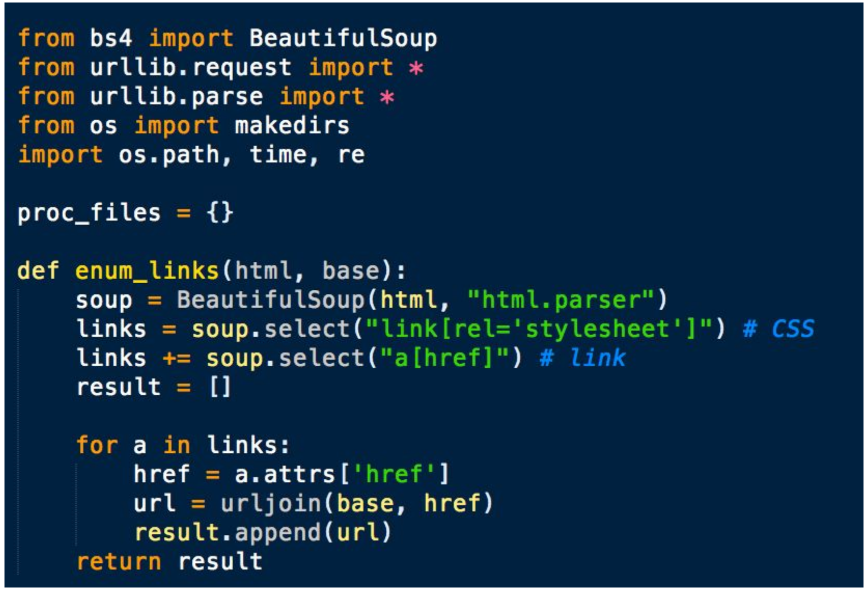
```python
def analyze_html(url, root_url):
    savepath = download_file(url)
    if savepath is None: return
    if savepath in proc_files: return
    proc_files[savepath] = True
    print("analyze_html=", url)

    html = open(savepath, "r", encoding="utf-8").read()
    links = enum_links(html, url)
    for link_url in links:

        if link_url.find(root_url) != 0:
            if not re.search(r".css$", link_url): continue

        if re.search(r".(html|htm)$", link_url):
            analyze_html(link_url, root_url)
            continue

        download_file(link_url)
if __name__ == "__main__":
    url = "https://docs.python.org/3.5/library/"
    analyze_html(url, url)
```

Recursive Downloading a site



```python
def download_file(url):
    o = urlparse(url)
    savepath = "./" + o.netloc + o.path
    if re.search(r"/$", savepath): # folder? index.html
        savepath += "index.html"
    savedir = os.path.dirname(savepath)

    if os.path.exists(savepath): return savepath

    if not os.path.exists(savedir):
        print("mkdir=", savedir)
        makedirs(savedir)

    try:
        print("download=", url)
        urlretrieve(url, savepath)
        time.sleep(1)
        return savepath
    except:
        print("다운 실패: ", url)
        return None
```

Recursive Downloading a site



```python
from bs4 import BeautifulSoup
from urllib.request import *
from urllib.parse import *
from os import makedirs
import os.path, time, re

proc_files = {}

def enum_links(html, base):
    soup = BeautifulSoup(html, "html.parser")
    links = soup.select("link[rel='stylesheet']") # CSS
    links += soup.select("a[href]") # link
    result = []

    for a in links:
        href = a.attrs['href']
        url = urljoin(base, href)
        result.append(url)
    return result
```

Web Browser Scrapping

- Selenium

Web API ?

Clint - (HTTP Request) =>
Server (HTTP Response) => Client

Web API

Features

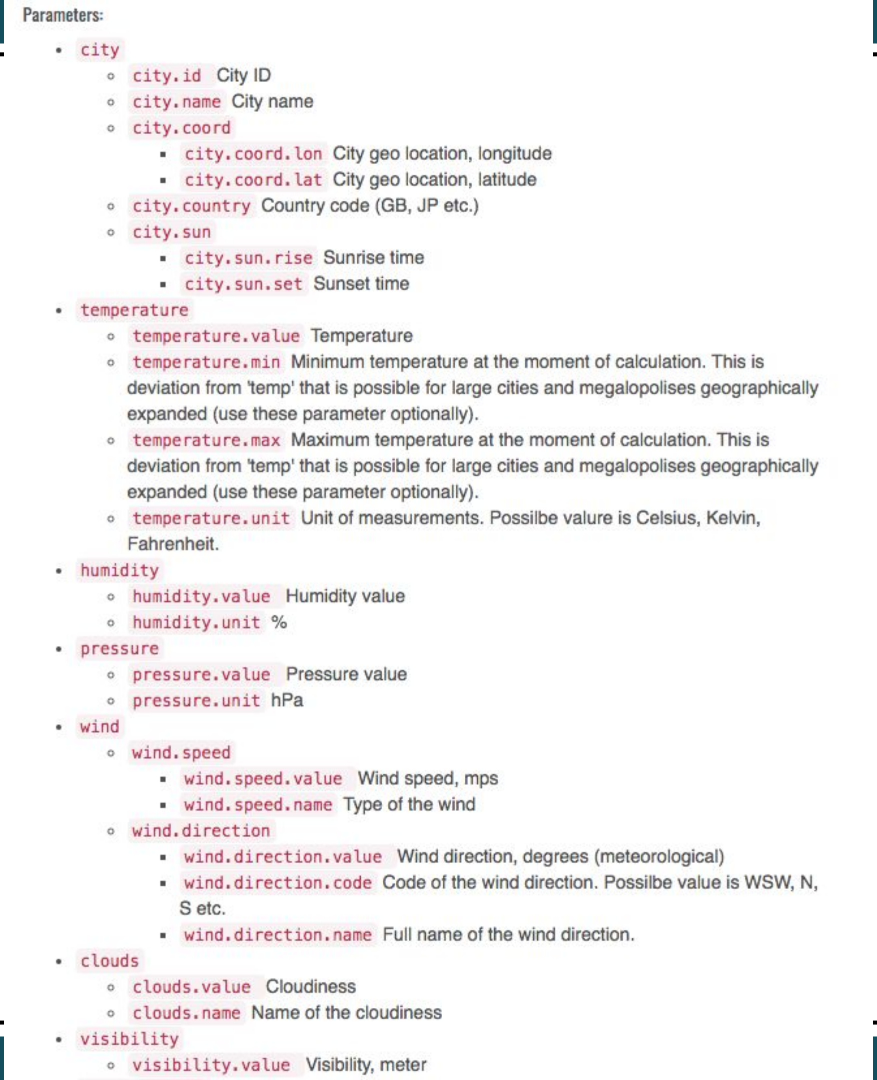
- 크롤링 표적도 될수있다. => ?
- 단점: 웹 API 는 변하거나 없어질수 있다. => ?

Web API

Practice

Sign up OpenWeatherMap Web API
https://home.openweathermap.org/users/sign_up

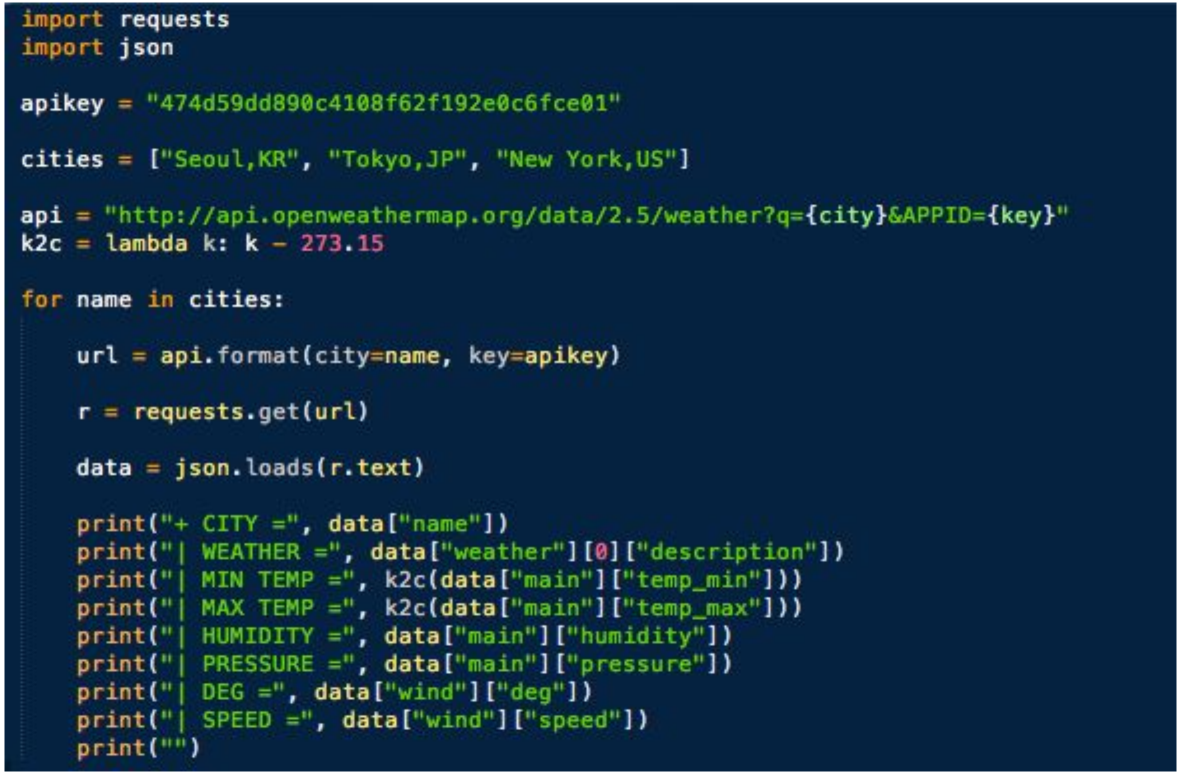
Before using API, you sholud find out API manual
- https://openweathermap.org/current



Parameters:

- `city`
  - `city.id` City ID
  - `city.name` City name
  - `city.coord`
    - `city.coord.lon` City geo location, longitude
    - `city.coord.lat` City geo location, latitude
  - `city.country` Country code (GB, JP etc.)
  - `city.sun`
    - `city.sun.rise` Sunrise time
    - `city.sun.set` Sunset time
- `temperature`
  - `temperature.value` Temperature
  - `temperature.min` Minimum temperature at the moment of calculation. This is deviation from 'temp' that is possible for large cities and megalopolises geographically expanded (use these parameter optionally).
  - `temperature.max` Maximum temperature at the moment of calculation. This is deviation from 'temp' that is possible for large cities and megalopolises geographically expanded (use these parameter optionally).
  - `temperature.unit` Unit of measurements. Possilbe valure is Celsius, Kelvin, Fahrenheit.
- `humidity`
  - `humidity.value` Humidity value
  - `humidity.unit` %
- `pressure`
  - `pressure.value` Pressure value
  - `pressure.unit` hPa
- `wind`
  - `wind.speed`
    - `wind.speed.value` Wind speed, mps
    - `wind.speed.name` Type of the wind
  - `wind.direction`
    - `wind.direction.value` Wind direction, degrees (meteorological)
    - `wind.direction.code` Code of the wind direction. Possilbe value is WSW, N, S etc.
    - `wind.direction.name` Full name of the wind direction.
- `clouds`
  - `clouds.value` Cloudiness
  - `clouds.name` Name of the cloudiness
- `visibility`
  - `visibility.value` Visibility, meter

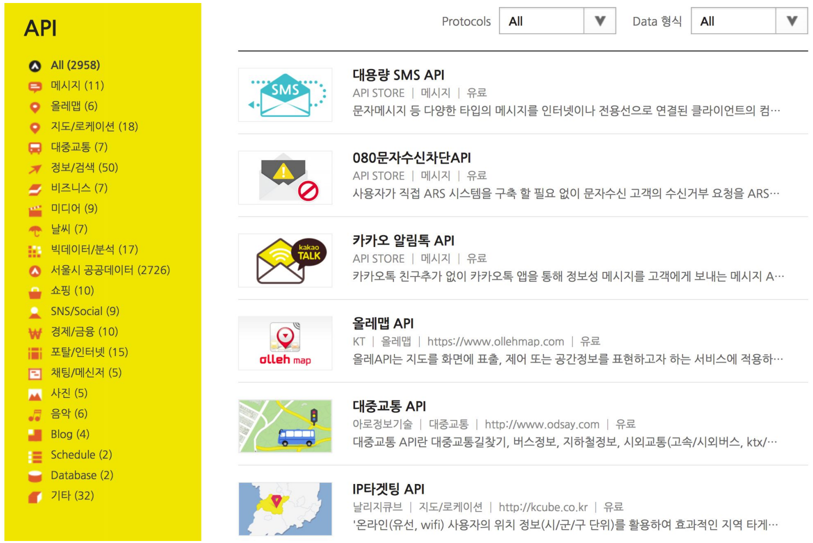
Web API



```
import requests
import json

apikey = "474d59dd890c4108f62f192e0c6fce01"

cities = ["Seoul,KR", "Tokyo,JP", "New York,US"]

api = "http://api.openweathermap.org/data/2.5/weather?q={city}&APPID={key}"
k2c = lambda k: k - 273.15

for name in cities:

    url = api.format(city=name, key=apikey)

    r = requests.get(url)

    data = json.loads(r.text)

    print("+ CITY =", data["name"])
    print("| WEATHER =", data["weather"][0]["description"])
    print("| MIN TEMP =", k2c(data["main"]["temp_min"]))
    print("| MAX TEMP =", k2c(data["main"]["temp_max"]))
    print("| HUMIDITY =", data["main"]["humidity"])
    print("| PRESSURE =", data["main"]["pressure"])
    print("| DEG =", data["wind"]["deg"])
    print("| SPEED =", data["wind"]["speed"])
    print("")
```

Web API

explore www.apistore.co.kr/api/apiList.do



API

- All (2958)
- 메시지 (11)
- 올레맵 (6)
- 지도/로케이션 (18)
- 대중교통 (7)
- 정보/검색 (50)
- 비즈니스 (7)
- 미디어 (9)
- 날씨 (7)
- 빅데이터/분석 (17)
- 서울시 공공데이터 (2726)
- 쇼핑 (10)
- SNS/Social (9)
- 경제/금융 (10)
- 포탈/인터넷 (15)
- 채팅/메신저 (5)
- 사진 (5)
- 음악 (6)
- Blog (4)
- Schedule (2)
- Database (2)
- 기타 (32)

Protocols All Data 형식 All

대용량 SMS API
API STORE | 메시지 | 유료
문자메시지 등 다양한 타입의 메시지를 인터넷이나 전용선으로 연결된 클라이언트의 컴…

080문자수신차단API
API STORE | 메시지 | 유료
사용자가 직접 ARS 시스템을 구축 할 필요 없이 문자수신 고객의 수신거부 요청을 ARS…

카카오 알림톡 API
API STORE | 메시지 | 유료
카카오톡 친구추가 없이 카카오톡 앱을 통해 정보성 메시지를 고객에게 보내는 메시지 A…

올레맵 API
KT | 올레맵 | https://www.ollehmap.com | 유료
올레API는 지도를 화면에 표출, 제어 또는 공간정보를 표현하고자 하는 서비스에 적용하…

대중교통 API
아로정보기술 | 대중교통 | http://www.odsay.com | 유료
대중교통 API란 대중교통길찾기, 버스정보, 지하철정보, 시외교통(고속/시외버스, ktx/…

IP타겟팅 API
날리지큐브 | 지도/로케이션 | http://kcube.co.kr | 유료
'온라인(유선, wifi) 사용자의 위치 정보(시/군/구 단위)를 활용하여 효과적인 지역 타게…

Crontab

why do we need this?

- Cron is the name of program that enables unix users to **execute commands or scripts** (groups of commands) automatically at **a specified time/date**.

- It is normally used for sys admin commands, for running a backup script, but can be used for anything. A common use for it today is connecting to the internet and downloading your email.

ANY IDEA ?

Crontab

1. What is crontab?

Crontab (CRON Tablle) is **a file** which contains the schedule of cron entries to be run and at specified times. File location varies by operating systems, See Crontab file location at the end of this document.

2.What is a cron job or cron schedule?

Cron job or cron schedule is **a specific set of execution instructions** specifing day, time and command to execute. **crontab can have multiple execution statments.**

Crontab

3. Crontab Restrictions

You can execute crontab if your name appears in the file **/usr/lib/cron/cron.allow.** If that file does not exist, you can use crontab if your name does not appear in the file **/usr/lib/cron/cron.deny**.

If only cron.deny exists and is empty, all users can use crontab. If neither file exists, only the root user can use crontab. The allow/deny files consist of one user name per line

Crontab

SET TIME

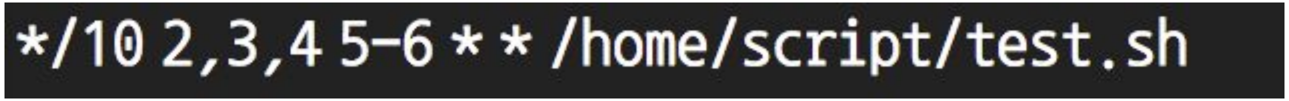
```
*           *            *          *           *
분(0-59)    시간(0-23)    일(1-31)    월(1-12)     요일(0-7)
```

```
* * * * * /home/script/test.sh
```

```
45 5 * * 5 /home/script/test.sh
```

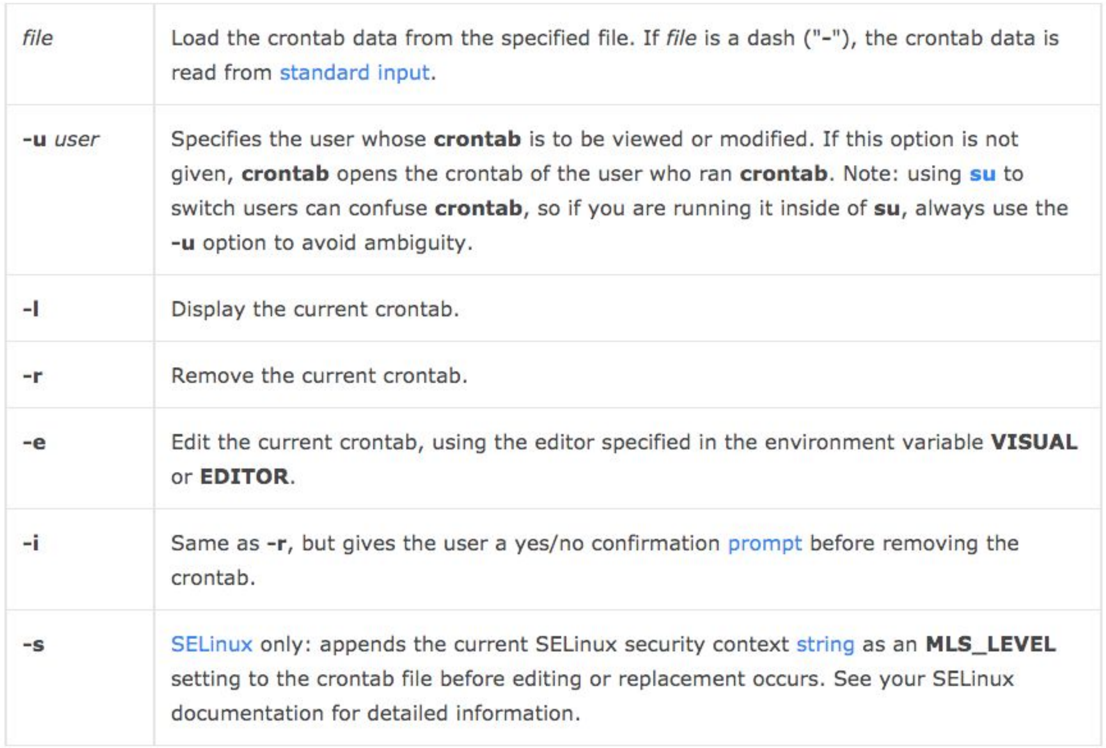
```
0-30 1 * * * /home/script/test.sh
```

Crontab



```
*/10 2,3,4 5-6 * * /home/script/test.sh
```

Crontab Cheat Sheet



file	Load the crontab data from the specified file. If *file* is a dash ("**-**"), the crontab data is read from standard input.
-u *user*	Specifies the user whose **crontab** is to be viewed or modified. If this option is not given, **crontab** opens the crontab of the user who ran **crontab**. Note: using **su** to switch users can confuse **crontab**, so if you are running it inside of **su**, always use the **-u** option to avoid ambiguity.
-l	Display the current crontab.
-r	Remove the current crontab.
-e	Edit the current crontab, using the editor specified in the environment variable **VISUAL** or **EDITOR**.
-i	Same as **-r**, but gives the user a yes/no confirmation prompt before removing the crontab.
-s	SELinux only: appends the current SELinux security context string as an **MLS_LEVEL** setting to the crontab file before editing or replacement occurs. See your SELinux documentation for detailed information.

Crontab

still not familiar with crontab's syntax ?

https://crontab.guru/#*/6_*_*_*_*

Summary

Now, you can make a scheduling job for data acqusition.

To access a lot of public data, please search for it.

http://hadoopilluminated.com/hadoop_illuminated/Public_Bigdata_Sets.html#d1575e4375

instagram - https://www.instagram.com/developer/
twitter - https://developer.twitter.com/en/docs/tweets/search/overview

Reference - 머신러닝 딥러닝 실전개발 입문, 위키북스

Appendix

- **What if you have to login to download data from a site?**

- https://beomi.github.io/2017/01/20/HowToMakeWebCrawler-With-Login/