

외부 모듈

목차

- 시작하기 전에
- 모듈 설치하기
- 모듈 찾아보기
- BeautifulSoup 모듈
- Flask 모듈
- 라이브러리와 프레임워크
- 키워드로 정리하는 핵심 포인트
- 확인문제

[핵심 키워드] : 외부 모듈, pip install, 제어 역전, 라이브러리, 프레임워크

[핵심 포인트]

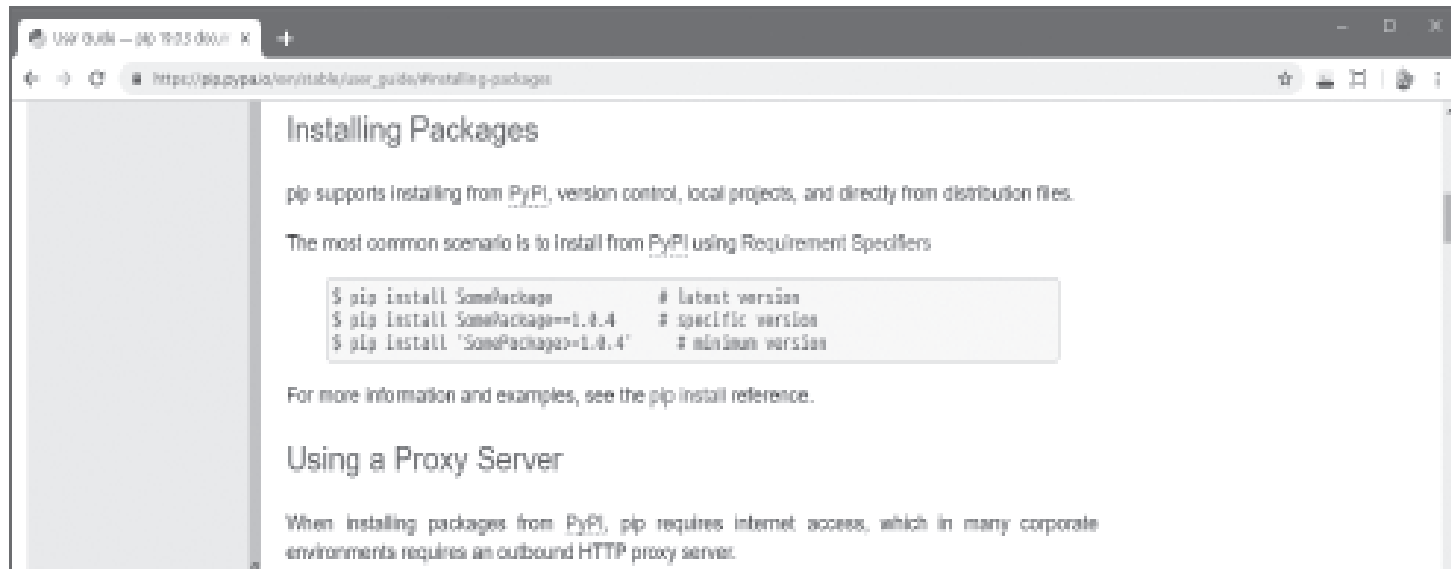
파이썬이 기본적으로 제공하는 것이 아닌 다른 사람들이 만들어 배포하는 모듈을 외부 모듈 (external module) 이라고 부른다. 외부 모듈 중 Beautiful Soup와 Flask를 설치한 후 활용하는 방법을 알아본다.

- 외부 모듈 설치
 - [Windows] + [R] 클릭하여 프로그램 실행창 띄우고 [cmd] 입력하면 명령 프롬프트 창 나타남

```
pip install 모듈 이름
```

```
> pip install beautifulsoup4
Collecting beautifulsoup4
  Downloading beautifulsoup4-4.6.0-py3-none-any.whl (86kB)
    100% |#####~####| 92kB 422kB/s
Installing collected packages: beautifulsoup4
Successfully installed beautifulsoup4-4.6.0
```

- pip
 - 특정 버전 모듈 설치 및 제거 등
 - http://pip.pypa.io/en/stable/user_guide/#installing-packages

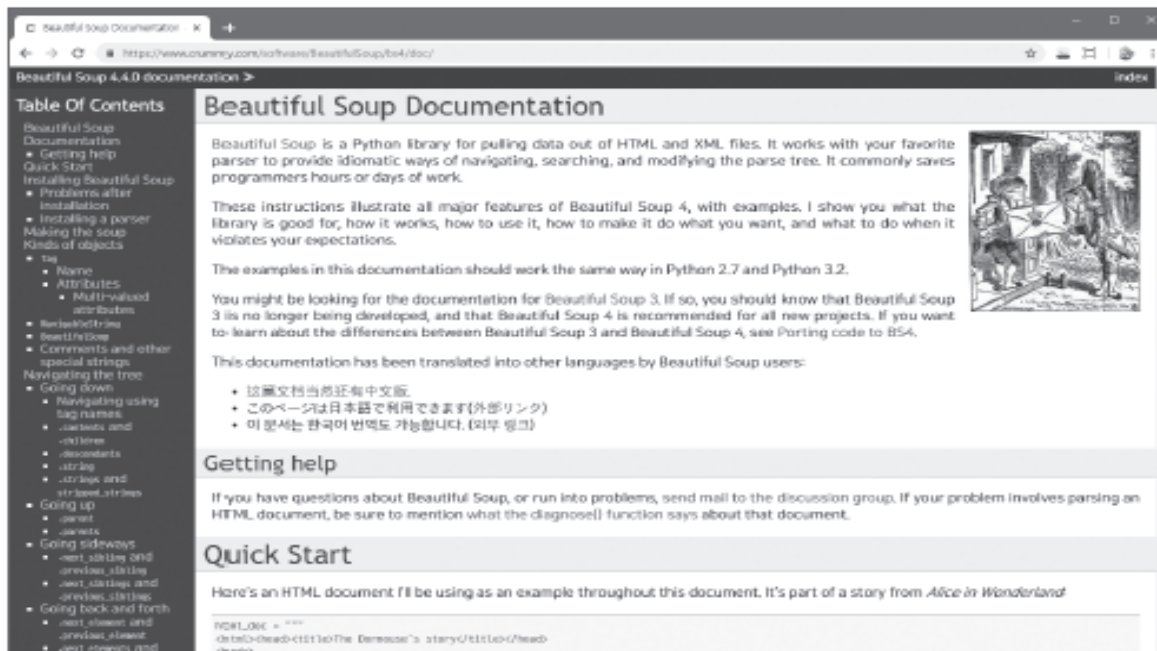


- 파이썬 관련 도서 구매할 경우 추천받는 모듈
- 인터넷 커뮤니티 정보로 접하는 모듈
- 직접 구글에서 검색하여 찾는 모듈
 - Python 키워드 옆에 내가 원하는 것을 더하여 검색

- Beautiful Soup 모듈

- 웹 페이지 분석 모듈

- <http://www.crummy.com/software/BeautifulSoup/bs4/doc/>



- 기상청 날씨 정보 가져와 출력하기
 - http://www.kma.go.kr/weather/lifenindustry/service_rss.jsp




```
01  # 모듈을 읽어 들입니다.
02  from urllib import request
03  from bs4 import BeautifulSoup
04
05  # urlopen() 함수로 기상청의 전국 날씨를 읽습니다.
06  target = request.urlopen("http://www.kma.go.kr/weather/forecast/mid-term-rss3.jsp?stnId=108") → 이 코드는 한 줄 코드이니 이어서 입력해야 합니다.
07
08  # BeautifulSoup을 사용해 웹 페이지를 분석합니다.
09  soup = BeautifulSoup(target, "html.parser")
10
11  # location 태그를 찾습니다.
12  for location in soup.select("location"):
13      # 내부의 city, wf, tmn, tmx 태그를 찾아 출력합니다.
14      print("도시:", location.select_one("city").string)
15      print("날씨:", location.select_one("wf").string)
16      print("최저기온:", location.select_one("tmn").string)
17      print("최고기온:", location.select_one("tmx").string)
18      print()
```

```
<rss version="2.0">
  <channel>
    <title>기상청 육상 중기예보</title>
    <!-- 생략 -->
    <item>
      <author>기상청</author>
      <!-- 생략 -->
      <description>
        <header>
          <title>전국 육상중기예보</title>
          <tm>201904221800</tm>
          <wf><![CDATA[기압골의 영향으로 25일 오후부터 26일 오전 사이 제주도를 제외한 전국에,
29일에는 충청도와 남부지방, 제주도에 비가 오겠습니다.<br />그 밖의 날은 고기압의 가장자리에 들어 가
끔 구름 많겠습니다.<br />기온은 평년(최저기온: 4~13℃, 최고기온: 18~24℃)보다 전반에는 조금 낮겠
고, 후반에는 비슷하겠습니다.<br />강수량은 평년(1~7mm)보다 많겠습니다.]]></wf>
        </header>
        <body>
          <location wl_ver="3"> → location마다 지역이 표기되어 있으므로 이를 모두 추출합니다.
          <province>서울 · 인천 · 경기도</province>
          <city>서울</city>
```

```
<data>
  <mode>A02</mode>
  <tmEf>2019-04-25 00:00</tmEf>
  <wf>구름많음</wf>
  <tmn>14</tmn>
  <tmx>20</tmx>
  <reliability>보통</reliability>
</data>
<!-- 생략 -->
</location>
<!-- 생략 -->
</body>
</description>
</item>
</channel>
</rss>
```

→ 내부에 날씨가 적혀 있으므로 이것들을 가져옵니다.

- 여기에서 지역 표기된 location 찾고,
location 내부에 있는 city, wf, tmn, tmx 내용 추출

도시: 서울

날씨: 구름많음

최저기온: 14

최고기온: 20

도시: 인천

날씨: 구름많음

최저기온: 13

최고기온: 18

도시: 수원

날씨: 구름많음

최저기온: 14

최고기온: 20

...생략...

- Django 모듈

- 다양한 기능 제공하는 웹 개발 프레임워크

- Flask

- 작은 기능만을 제공하는 웹 개발 프레임워크
- [Window] + [R] 눌러 프로그램 실행 창 띄우고 [cmd] 입력하여 명령 프롬프트 창

```
pip install flask
```

- 예시 - Flask 모듈 사용하기

```
01 from flask import Flask
02 app = Flask(__name__)
03
04 @app.route("/")
05 def hello():
06     return "<h1>Hello World!</h1>"
```

- 데코레이터 (decorator)
 - @app.route()

- Flask에서 코드 실행

```
set FLASK_APP=파일 이름.py  
flask run
```

```
> set FLASK_APP=flask_basic.py  
> flask run  
* Serving Flask app "flask_basic.py"  
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

- 종료
 - [Ctrl] + [c]

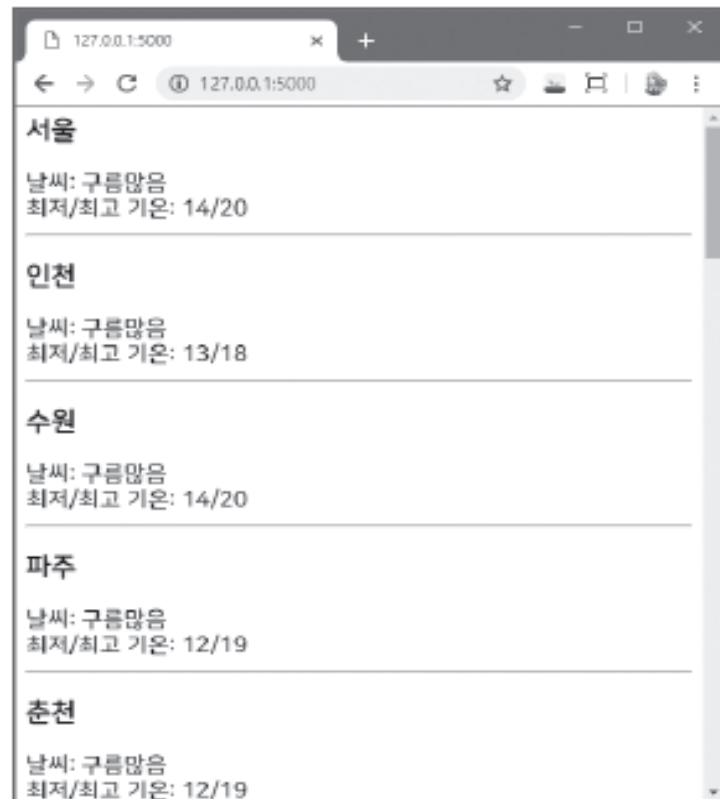
- 예시 - BeautifulSoup 스크레이핑 실행하기

```
01  # 모듈을 읽어 들입니다.
02  from flask import Flask
03  from urllib import request
04  from bs4 import BeautifulSoup
05
06  # 웹 서버를 생성합니다.
07  app = Flask(__name__)
08  @app.route("/")
09
10  def hello():
11      # urlopen() 함수로 기상청의 전국 날씨를 읽습니다.
12      target = request.urlopen("http://www.kma.go.kr/weather/forecast/mid-term-rss3.jsp?stnId=108") → 이 코드는 한 줄 코드이니 이어서 입력해야 합니다.
13
14      # BeautifulSoup를 사용해 웹 페이지를 분석합니다.
15      soup = BeautifulSoup(target, "html.parser")
```



```
16
17     # location 태그를 찾습니다.
18     output = ""
19     for location in soup.select("location"):
20         # 내부의 city, wf, tmn, tmx 태그를 찾아 출력합니다.
21         output += "<h3>{}</h3>".format(location.select_one("city").string)
22         output += "날씨: {}<br/>".format(location.select_one("wf").string)
23         output += "최저/최고 기온: {}/{}"\
24             .format(\
25                 location.select_one("tmn").string,\
26                 location.select_one("tmx").string\
27             )
28         output += "<hr/>"
29     return output
```

접속할 때마다 날씨 정보를 보여주는 웹 서버



- 라이브러리 (library)
 - 정상적인 제어 하는 모듈
- 프레임워크 (framework)
 - 제어 역전이 발생하는 모듈
- 제어 역전 (IoC : Invasion of Control)
 - “역전되지 않은 정상적인 제어”란?

- 라이브러리
 - 개발자가 모듈 기능을 호출하는 형태의 모듈

정상적인 제어 = 라이브러리

```
# 모듈을 읽어 들입니다.  
from math import sin, cos, tan, floor, ceil  
  
# sin, cos, tan를 구합니다.  
print("sin(1):", sin(1))  
print("cos(1):", cos(1))  
print("tan(1):", tan(1))  
  
# 내림과 올림을 구합니다.  
print("floor(2.5):", floor(2.5))  
print("ceil(2.5):", ceil(2.5))
```

- 프레임워크

```
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
    return "<h1>Hello World!</h1>"
```

```
> set FLASK_APP=flask_basic.py
> flask run
* Serving Flask app "flask_basic.py"
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

- Flask 모듈 내부에서 출력된 글자들
 - Flask가 내부적으로 서버를 실행한 뒤 지정한 파일을 읽어들이어 적절한 상황에 스스로 실행
-
- 프레임워크란, 모듈이 개발자가 작성한 코드를 실행하는 형태의 모듈
 - 제어 역전이란, 개발자가 만든 함수를 모듈이 실행하는 형태의, 일반적인 경우와 반대인 제어 흐름

- **외부 모듈** : 파이썬이 기본적으로 제공하지 않는, 다른 사람들이 만들어 제공하는 모듈
- **pip install** : 외부 모듈을 설치할 때 사용하는 명령어
- **제어 역전** : 개발자가 모듈의 함수를 호출하는 것이 일반적인 제어 흐름이 나,
이와 반대로 개발자가 만든 함수를 모듈이 실행하는 것
- **라이브러리** : 개발자가 모듈의 기능을 호출하는 형태와 같이 정상적인 제어를 하는 모듈
- **프레임워크** : 모듈이 개발자가 작성한 코드를 실행하는 형태의 모듈

- 구글 검색으로 "python prime module"이라고 치면 파이썬에서 소수 (prime number)를 구하는 모듈을 찾아보세요. prime, primenumbers, pyprimes, pyprimesieve 등 다양한 모듈이 나올 것입니다. 적당한 모듈을 선택해서 100~1000 사이에 있는 소수가 몇 개인지 구해 주세요.