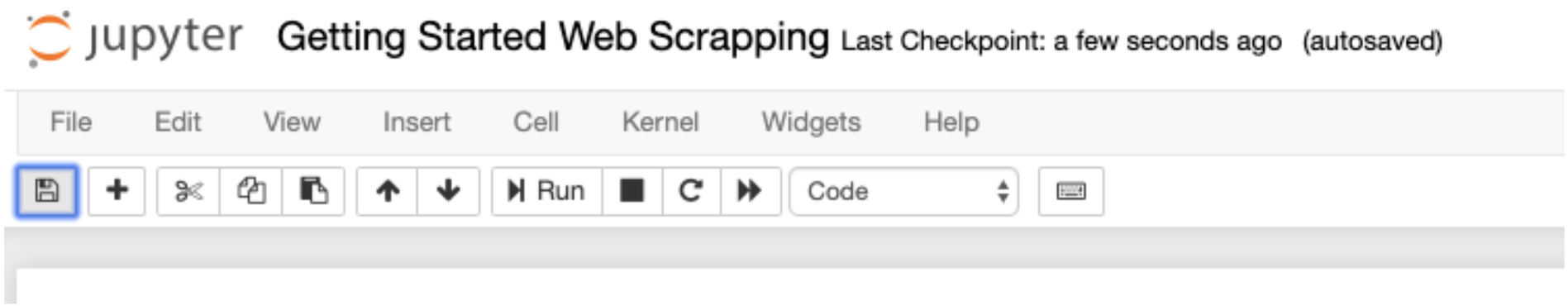
# Web Scraping
# 웹 스크래핑

강윤희

# 웹 스크래핑 패키지

- requests  패키지를 사용하여 웹페이지의 콘텐츠를 얻음
- bs4 (BeautifulSoup) 패키지를 사용하여 콘텐츠에서 정보를 추출함  for extracting the relevant information.



# requests 패키지 사용

In [1]:
```python
# parser.py
import requests

# HTTP GET Request
req = requests.get('https://beomi.github.io/')

# HTML 소스 가져오기
html = req.text
# HTTP Header 가져오기
header = req.headers
# HTTP Status 가져오기 (200: 정상)
status = req.status_code
# HTTP가 정상적으로 되었는지 (True/False)
is_ok = req.ok
```

In [2]:
```python
status
```

Out[2]:
```
200
```

# bs4 패키지 사용

**HTML 과 XML 파일에서 데이터를 얻어옴**

In [12]:
```
html_doc = """
<html><head><title>The Dormouse's story</title></head>
<body>
<p class="title"><b>The Dormouse's story</b></p>

<p class="story">Once upon a time there were three little sisters; and their names were
<a href="http://example.com/elsie" class="sister" id="link1">Elsie</a>,
<a href="http://example.com/lacie" class="sister" id="link2">Lacie</a> and
<a href="http://example.com/tillie" class="sister" id="link3">Tillie</a>;
and they lived at the bottom of a well.</p>

<p class="story">...</p>
"""
```

In [13]:
```
from bs4 import BeautifulSoup
soup = BeautifulSoup(html_doc, 'html.parser')
```

# bs4 패키지 사용

**a 태그 안의 텍스트와 a 태그의 href속성의 값을 가져옴**

In [14]:
```
my_title = soup.select ('title')
```

**Find tags directly beneath other tags (p 아래 a):**

In [15]:
```
data = {}
my_titles = soup.select('p > a')
for title in my_titles:
    data[title.text] = title.get('href')
```

In [16]:
```
type(data)
```

Out[16]:
```
dict
```

In [17]:
```
for key in data.keys():
    print ("{} = {}".format(key, data.get(key)))
```

```
Elsie = http://example.com/elsie
Lacie = http://example.com/lacie
Tillie = http://example.com/tillie
```