

LESSON

urllib 모듈 활용

```
animLength = toTime - fromTime

# Ask user for directory
filePath = c4d.storage.SaveDialog()
filePath, objName = os.path.split(filePath)
objName = objName + ". "
filePath = filePath + "\\ "

# Ask for confirmation
questionDialogText = "Obj Sequence will be saved as:\n\n" \
    "" + filePath + objName + "####.obj\n\n" \
    "From frame " + str(fromTime) + " to " + str(toTime) + " "
proceedBool = c4d.gui.QuestionDialog(questionDialogText)

if proceedBool == True:

    # Loop through animation and export frames
    for x in range(0,animLength):

        # change frame, redraw view
        moveTime = c4d.BaseTime(fromTime,docFps) + c4d.BaseTime(x)
        doc.SetTime(moveTime)
        c4d.EventAdd(c4d.EVENT_FORCEREDRAW)
        c4d.DrawViews(c4d.DRAWFLAGS_FORCEFULLREDRAW)

        # update status bar
        c4d.StatusSetText("Exporting " + str(x) + " of " + str(animLength))
        c4d.StatusSetBar(100.0*x/animLength)

        # add buffer 0001
        bufferedNumber = str(doc.GetTime().GetFrame(docFps))
        if len(bufferedNumber) < 4:
```

urllib 모듈이란?

- ❖ 파이썬의 표준 모듈로써 URL을 다루기 위한 모듈 패키지
- ❖ 설치가 필요하지 않고, `import urllib`로 활용
- ❖ requests 모듈과 마찬가지로 URL과 관련된 여러가지 기능들을 제공

```
import requests

URL = 'https://www.naver.com'
response = requests.get(URL)
html_data = response.text
print(html_data)
```

```
import urllib

URL = 'https://www.naver.com'
response_urllib = urllib.request.urlopen(URL)
byte_data = response_urllib.read()
text_data = byte_data.decode('utf-8')
print(text_data)
```

※ 참고 URL : <https://docs.python.org/3/library/urllib.html>

urllib 모듈이란?

♣ 총 4가지의 하위 모듈이 존재

request

URL을 열고 읽는 모듈(HTTP 요청)

error

request 모듈에서 발생하는 에러들을 포함하는 모듈

parse

URL을 파싱하는 모듈(URL 해석 및 조작)

robotparser

robots.txt 파일을 파싱하는 모듈

urllib.request : request()

- 해당 URL 요청 객체 생성, HTTPRequest 객체 반환
- 여러가지 함수들을 제공 : 전체 URL, 프로토콜 타입, 호스트 등

```
import urllib

URL = 'https://www.naver.com'
request = urllib.request.Request(URL)
print(request)
```

<urllib.request.Request object at 0x7fd10931d6a0>

```
print(request.full_url)
print(request.type)
print(request.host)
```

https://www.naver.com
https
www.naver.com

urllib.request : urlopen()

- ♣ 해당 URL을 열기, 응답 데이터는 바이트 형식의 HTTPResponse 객체
- ♣ request 객체 또는 URL을 직접 넣어도 가능

```
response1 = urllib.request.urlopen(request)
response2 = urllib.request.urlopen(URL)
print(response1)
print(response2)
```

```
<http.client.HTTPResponse object at 0x7fd10931d3c8>
<http.client.HTTPResponse object at 0x7fd10931d320>
```

```
print(response1.geturl())
print(response1.getheaders())
```

<https://www.naver.com>

```
[('Server', 'NWS'), ('Date', 'Thu, 05 Dec 2019 12:27:29 GMT'), ('Content-Type', 'text/html; charset=UTF-8'), ('Transfer-Encoding', 'chunked'), ('Connection', 'close'), ('Set-Cookie', 'PM_CK_loc=23762fd3f58f4534e19349ee6b32743b76cce0dc18e7f3edccbae918c437d549; Expires=Fri, 06 Dec 2019 12:27:29 GMT; Path=/; HttpOnly'), ('Cache-Control', 'no-cache, no-store, must-revalidate'), ('Pragma', 'no-cache'), ('P3P', 'CP="CAO DSP CURa ADMa TAIA PSAA OUR LAW STP PHY ONL UNI PUR FIN COM NAV INT DEM STA PRE"'), ('X-Frame-Options', 'DENY'), ('X-XSS-Protection', '1; mode=block'), ('Strict-Transport-Security', 'max-age=63072000; includeSubdomains'), ('Referrer-Policy', 'unsafe-url')]
```

```
urllib.request : read()
```

- `urlopen`으로 연 객체를 읽고, **인자로 전달하는 숫자만큼 데이터를 읽음**
 - 바이트 형식의 데이터
- `readlines()` : 홈페이지 데이터를 줄 단위로 읽어 리스트에 반환

```
import urllib
```

```
URL = 'https://www.naver.com'
response_urllib = urllib.request.urlopen(URL)
byte_data = response_urllib.read(500)
print(byte_data)
```

```
b'<!doctype html>\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n<html lang="ko">\n<head>\n<meta charset="utf-8">\n<meta name="Referer" content="origin"/>\n<meta http-equiv="Content-Script-Type" content="text/javascript"/>\n<meta http-equiv="Content-Style-Type" content="text/css"/>\n<meta http-equiv="X-UA-Compatible" content="IE=edge"/>\n<meta name="viewport" content="width=1100"/>\n<meta name="apple-mobile-web-app-title" content="NAVER"/>\n<meta name="robots" content="index,nofollow"/>\n<meta name="description" content="\xeb\x84\xa4\xc9\xd\xb4\xe7\xb2\x84 \xe7\xa9\x94\xc9\xd\xb8\xe7\x97\x90'
```

urllib.request : decode()

🔗 바이트 형식의 데이터를 원하는 형식으로 변환

- 기본값으로 utf-8 사용

```
import urllib
```

```
URL = 'https://www.naver.com'  
response_urllib = urllib.request.urlopen(URL)  
byte_data = response_urllib.read(500)  
text_data = byte_data.decode()  
print(text_data)
```

```
<!doctype html>
```

```
<html lang="ko">  
<head>  
<meta charset="utf-8">  
<meta name="Referrer" content="origin">  
<meta http-equiv="Content-Script-Type" content="text/javascript">  
<meta http-equiv="Content-Style-Type" content="text/css">  
<meta http-equiv="X-UA-Compatible" content="IE=edge">  
<meta name="viewport" content="width=1100">  
<meta name="apple-mobile-web-app-title" content="NAVER" />  
<meta name="robots" content="index,nofollow"/>  
<meta name="description" content="네이버 메인에
```

urllib.request : urlretrieve()

📌 웹 상의 이미지를 다운로드

```
import urllib.request
img_src = 'https://t1.daumcdn.net/roughmap/imgmap/4d1f9936cd9885b49544e08c4dadf10f5c0119fd8a87cd66f0ddda9d68e51a18'
new_name = 'img.png'
urllib.request.urlretrieve(img_src, new_name)

('img.png', <http.client.HTTPMessage at 0x7fd0e8912ac8>)
```



img_src = “(이미지 URL 주소)”를 입력한 후
해당 주소 이미지를 다운로드 해 ‘img.png’라는
이름으로 저장하도록 지정함



urllib.parse

- URL을 파싱하여 분석하기 위한 모듈
 - http뿐만 아니라 ftp, ssh, imap 등도 포함
- URL은 인터넷 공간에 존재하는 데이터들을 가리키기 위한 절대 주소

URL 형식

프로토콜://아이디:비밀번호@호스트:포트/하위경로?파라미터#색인

URL 예시

[예1] 한국기술교육대학교 홈페이지

<https://www.koreatech.ac.kr/kor/CMS/Contents/Contents.do?mCode=MN021>

[예2] FTP 주소

<ftp://admin:1234@220.123.234.23/a.png>

urllib.parse : urlparse()

🔗 URL을 6개로 분리하여 반환

```
import urllib
```

```
parse = urllib.parse.urlparse('https://blog.naver.com/koreatech91;a=1?b=2#b')  
print(parse)
```

```
ParseResult(scheme='https', netloc='blog.naver.com', path='/koreatech91', params='a=1', query='b=2', fragment='b')
```

```
print(parse[0])  
print(parse[1])  
print(parse[2])
```

```
https  
blog.naver.com  
/koreatech91
```



```
print(parse.scheme)  
print(parse.netloc)  
print(parse.path)
```

```
https  
blog.naver.com  
/koreatech91
```

urllib.parse : urlsplit()

🔗 URL을 5개로 분리하여 반환

```
import urllib
```

```
parse = urllib.parse.urlparse('https://blog.naver.com/koreatech91;a=1?b=2#b')  
parse2 = urllib.parse.urlsplit('https://blog.naver.com/koreatech91;a=1?b=2#b')  
print(parse)  
print(parse2)
```

```
ParseResult(scheme='https', netloc='blog.naver.com', path='/koreatech91', params='a=1', query='b=2', fragment='b')  
SplitResult(scheme='https', netloc='blog.naver.com', path='/koreatech91;a=1', query='b=2', fragment='b')
```

```
print(parse2[0])  
print(parse2[1])  
print(parse2[2])
```

```
https  
blog.naver.com  
/koreatech91;a=1
```



```
print(parse2.scheme)  
print(parse2.netloc)  
print(parse2.path)
```

```
https  
blog.naver.com  
/koreatech91;a=1
```

urllib.parse : urlunparse(), urlunsplit()

- ❖ 분리된 URL을 다시 합침
- ❖ 튜플(변경 불가능)로 반환되기 때문에 리스트(변경 가능)로 변경하여 활용

```
import urllib
```

```
parse = urllib.parse.urlparse('https://blog.naver.com/koreatech91;a=1?b=2#b')  
print(parse)  
parse[1] = 'blog.daum.net'
```

```
ParseResult(scheme='https', netloc='blog.naver.com', path='/koreatech91', params='a=1',
```

```
-----  
TypeError
```

```
Traceback (most recent call last)
```

```
<ipython-input-130-4e90c8580c45> in <module>()  
      3 parse = urllib.parse.urlparse('https://blog.naver.com/koreatech91;a=1?b=2#b')
```

```
      4 print(parse)
```

```
----> 5 parse[1] = 'blog.daum.net'
```

```
TypeError: 'ParseResult' object does not support item assignment
```

urllib.parse : urlunparse(), urlunsplit()

- ❁ 분리된 URL을 다시 합침
- ❁ 튜플(변경 불가능)로 반환되기 때문에 리스트(변경 가능)로 변경하여 활용

```
parse = list(parse)
parse[1] = 'blog.daum.net'
```

```
unparse = urllib.parse.urlunparse(parse)
print(unparse)
```

```
https://blog.daum.net/koreatech91;a=1?b=2#b
```



리스트(list)로 자료형 변환 후 값 변경

urllib.parse : parse_qs(), parse_qsl()

- ❖ 쿼리(query)를 파싱해서 사전(qs) 및 리스트(qs1)로 반환
- ❖ 쿼리(query)를 변경하여 요청할 때 활용

```
import urllib
```

```
parse = urllib.parse.urlparse('https://www.naver.com?a=1&b=2&c=3&d=4')  
print(parse)  
print(parse.query)  
print(type(parse.query))
```

```
ParseResult(scheme='https', netloc='www.naver.com', path='', params='', query='a=1&b=2&c=3&d=4', fragment='')  
a=1&b=2&c=3&d=4  
<class 'str'>
```

```
qs = urllib.parse.parse_qs(parse.query)  
print(qs)  
print(type(qs))
```

```
{'a': ['1'], 'b': ['2'], 'c': ['3'], 'd': ['4']}  
<class 'dict'>
```

```
qs1 = urllib.parse.parse_qsl(parse.query)  
print(qs1)  
print(type(qs1))
```

```
[('a', '1'), ('b', '2'), ('c', '3'), ('d', '4')]  
<class 'list'>
```

urllib.parse : urljoin(a, b)

- a와 b URL을 합쳐주는 기능
- ‘/’ 에 따라 URL 주소가 달라지는 것 주의

```
import urllib.parse

url = 'https://naver.com/a/b'

print(urllib.parse.urljoin(url, 'c'))
print(urllib.parse.urljoin(url, '/c'))

https://naver.com/a/c
https://naver.com/c
```

```
url = 'https://naver.com/a/b/'

print(urllib.parse.urljoin(url, 'c'))
print(urllib.parse.urljoin(url, '/c'))

https://naver.com/a/b/c
https://naver.com/c
```

urllib.parse : quote(), unquote()

- ❖ 아스키 코드가 아닌 문자들을 퍼센트 인코딩으로 변환
- ❖ URL에 한글이 섞이면 오류 발생

```
import urllib
```

```
URL = 'https://search.naver.com/search.naver?query=파이썬'  
response_urllib = urllib.request.urlopen(URL)  
byte_data = response_urllib.read()  
text_data = byte_data.decode()  
print(text_data)
```

```
-----  
UnicodeEncodeError                                Traceback (most recent  
<ipython-input-160-250c218079b0> in <module>()  
    2  
    3 URL = 'https://search.naver.com/search.naver?query=파이썬'  
----> 4 response_urllib = urllib.request.urlopen(URL)  
      5 byte_data = response_urllib.read()  
      6 text_data = byte_data.decode()
```

```
print(urllib.parse.quote('파이썬'))
```

```
%ED%8C%8C%EC%9D%B4%EC%8D%AC
```


LESSON

urllib 모듈 활용

```
animLength = toTime - fromTime

# Ask user for directory
filePath = c4d.storage.SaveDialog()
filePath, objName = os.path.split(filePath)
objName = objName + "-"
filePath = filePath + "\\ "

# Ask for confirmation
questionDialogText = "Obj Sequence will be saved as:\n\n" \
    "" + filePath + objName + "###.obj\n\n" \
    "From frame " + str(fromTime) + " to " + str(toTime) + " "
proceedBool = c4d.gui.QuestionDialog(questionDialogText)

if proceedBool == True:

    # Loop through animation and export frames
    for x in range(0,animLength):

        # change frame, redraw view
        moveTime = c4d.BaseTime(fromTime,docFps) + c4d.BaseTime(x)
        doc.SetTime(moveTime)
        c4d.EventAdd(c4d.EVENT_FORCEREDRAW)
        c4d.DrawViews(c4d.DRAWFLAGS_FORCEFULLREDRAW)

        # update status bar
        c4d.StatusSetText("Exporting " + str(x) + " of " + str(animLength))
        c4d.StatusSetBar(100.0*x/animLength)

        # add buffer 0001
        bufferedNumber = str(doc.GetTime().GetFrame(docFps))
        if len(bufferedNumber) < 4:
```

홈페이지를 로컬에 파일로 저장

❖ 파이썬의 파일 입출력 함수 활용(open, write, close)

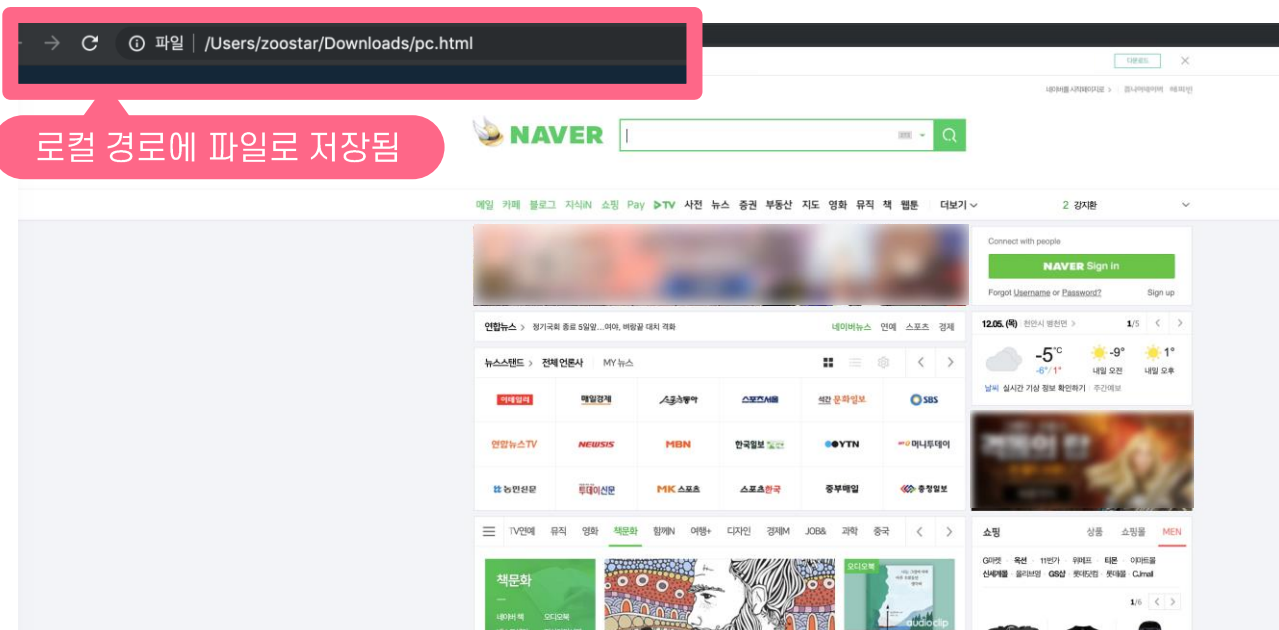
- open : 파일 객체 생성
- write : 생성한 파일에 데이터 입력
- close : 파일 객체 종료

```
import urllib

request = urllib.request.Request("http://www.naver.com")
data = urllib.request.urlopen(request).read()
f = open("pc.html", "wb")
f.write(data)
f.close()
```

홈페이지를 로컬에 파일로 저장

❖ 파이썬의 파일 입출력 함수 활용(open, write, close)



헤더(headers)를 추가해서 모바일 페이지로 저장

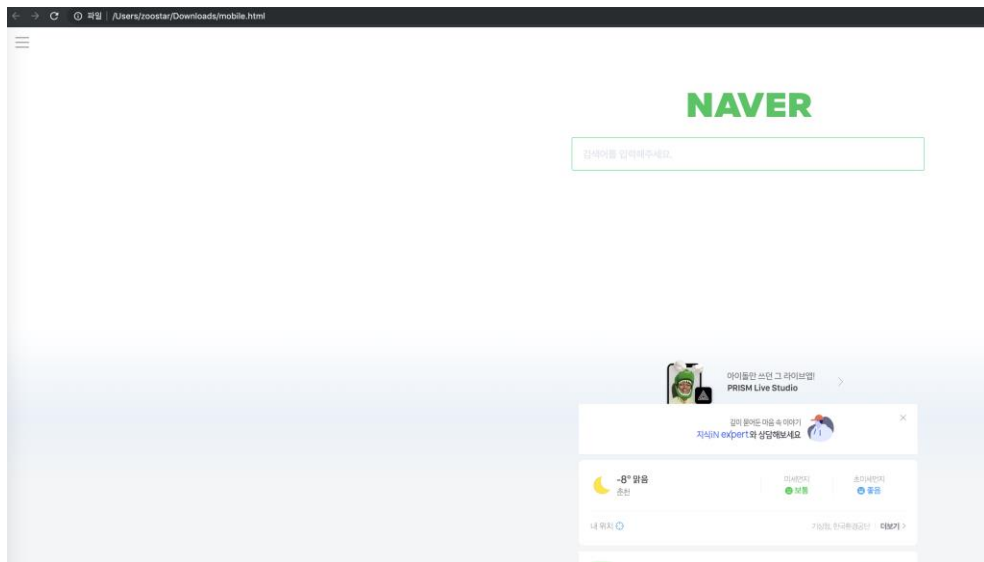
♣ request 함수에 헤더(headers) 인자에 값을 전달

```
import urllib

header = {'User-Agent': 'Mozilla/5.0 (iPhone)'}
request = urllib.request.Request("http://www.naver.com", headers = header)
data = urllib.request.urlopen(request).read()
f = open("mobile.html", "wb")
f.write(data)
f.close()
```

헤더(headers)를 추가해서 모바일 페이지로 저장

🍀 request 함수에 헤더(headers) 인자에 값을 전달



파라미터를 변경해 여러 정보 가져오기

❖ 여러가지 검색어의 Naver 검색 결과 출력

```
import urllib

query_list = ['파이썬', '웹 크롤링', '빅데이터', 'python']

URL = 'https://search.naver.com/search.naver?query='

for i in query_list:
    new_URL = URL + urllib.parse.quote(i)
    response_urllib = urllib.request.urlopen(new_URL)
    byte_data = response_urllib.read()
    text_data = byte_data.decode()
    print(text_data)
```



한글을 인코딩

파이썬 문법을 활용해 연관검색어 리스트 출력

정규 표현식 및 문자열 함수 활용

```
import urllib
import re

query_list = ['파이썬', '웹 크롤링', '빅데이터', 'python']
URL = 'https://search.naver.com/search.naver?query='

for i in query_list:
    new_URL = URL + urllib.parse.quote(i)
    response_urllib = urllib.request.urlopen(new_URL)
    byte_data = response_urllib.read()
    text_data = byte_data.decode()
    text_data = text_data.split('연관검색어')[1].split('도움말')[1].split('달기 후 1주일간')[0]
    text_data = re.sub('<.+?>', '', text_data, 0, re.I|re.S)
    text_data = text_data.replace(' ', '').strip()
    print(i, ":", text_data)
```

파이썬 : 파이썬 설치 python 파이썬 독학 파이썬 프로그래밍 파일썬 파이썬 자격증 파이썬 책 추천 파이썬 다운로드 파이썬 기초 c언어
 웹 크롤링 : 파이썬 웹 크롤링 beautifulsoup pandas 파이썬 아나콘다 아나콘다 설치 selenium
 빅데이터 : 빅데이터에서 상권분석 빅데이터실무 지능형로봇 빅데이터관련책 오픈마켓 빅데이터문학 빅데이터도서 빅데이터고1 빅데이터산업
 python : 파이썬 python.org python 설치 파이썬 독학 파이썬이란 파이썬 파이썬 기초 phyton 파이썬 리스트 파이썬 설치

NAVER

파이썬

통합검색

이미지

책

블로그

지식iN

카페

지식백과

뉴스

더보기

연관검색어

?

파이썬 설치

파이썬 독학

python

파이썬 자격증

파일썬

파이썬 책 추천

파이썬 게임

파이썬 프로그래밍

c언어

파이썬 책