In [ ]:
```
urllib              =>            requests
(GET)    Encoding => Charset(MIME - content-type); 본문내용(str?)
                     (HTML) HTML entitiy (&#10진수;)
         URL Encoding(Bytes) => Hexadecimal => %(percent)
urlparse                       params = {}, [(k,v)], bytes
parse_qs(l)
urlunparse
req = user-agent
urlopen(str or req obj)        request(url,params,data,headers...)
read()
decode() => str
try - except status_code       resp.statsu_code, rais_for_status
body -> HTML -> re(필요한 부분 추출; title)
3사(Google, Naver, Daum) 검색결과 X => re
```

In [1]:
```
from requests import request

headers = {
    'user-agent':'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KH
}
```

In [2]:
```
url = 'https://www.google.com/search'
params = {
    'q':'뉴진스'
}

resp = request('GET', url, params=params)
```

In [7]:
```
resp.status_code, resp.encoding, resp.request.headers
```

Out[7]:
```
(200,
 'ISO-8859-1',
 {'User-Agent': 'python-requests/2.28.1', 'Accept-Encoding': 'gzip, deflate, br', 'Accep
t': '*/*', 'Connection': 'keep-alive'})
```

In [10]:
```
from html import unescape
# 1번결과 - user-agent를 bot 둘때,
'LC20lb MBeuO DKV0Md'
```

Out[10]:
```
'LC20lb MBeuO DKV0Md'
```

In [11]:
```
# 2번 결과 - user-agent를 내 브라우저로 바꿨을때,
resp = request('GET', url, params=params, headers=headers)
resp.encoding, resp.request.headers
```

Out[11]:
```
('UTF-8',
 {'user-agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHT
ML, like Gecko) Chrome/114.0.0.0 Safari/537.36', 'Accept-Encoding': 'gzip, deflate, br',
'Accept': '*/*', 'Connection': 'keep-alive'})
```

In [12]:
```
import re
```

In [15]:
```
p = re.compile(r'<h3 class="LC20lb MBeuO DKV0Md">(.+?)</h3>')
p.findall(resp.text) # re.findall(p, resp.text)
```

Out[15]:
```
['NewJeans - 나무위키',
 'NewJeans - YouTube',
```

```
 '뉴진스와 함께할 새로운 계절을 기다리며  &#39;Attention&#39;부터 &#39;OMG ...',
 '뉴진스(NewJeans) [뮤직뱅크/Music Bank] | KBS 230127 방송',
 'NewJeans - 위키백과, 우리 모두의 백과사전']
```

In [35]:
```
p = re.compile(r'yuRUbf"><a href="(.+?)"')
p.findall(resp.text)
```

Out[35]:
```
['https://namu.wiki/w/NewJeans',
 'https://www.youtube.com/@NewJeans_official',
 'https://ko.wikipedia.org/wiki/NewJeans']
```

In [36]:
```
url = 'https://search.daum.net/search'
params = {
    'w':'tot',
    'q':'뉴진스'
}

resp = request('GET', url, params=params, headers=headers)
```

In [38]:
```
resp.status_code, resp.request.headers, resp.headers
```

Out[38]:
```
(200,
 {'user-agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHT
ML, like Gecko) Chrome/114.0.0.0 Safari/537.36', 'Accept-Encoding': 'gzip, deflate, br',
'Accept': '*/*', 'Connection': 'keep-alive'},
 {'Date': 'Tue, 04 Jul 2023 01:35:28 GMT', 'Content-Type': 'text/html; charset=utf-8',
'Transfer-Encoding': 'chunked', 'Connection': 'keep-alive', 'Set-Cookie': 'uvkey=ZKN3YEF
eGyA7vM9GA1AIqwAAADA; expires=Fri, 01-Jul-2033 01:35:28 GMT; path=/; domain=.search.dau
m.net;, _ksk=ZKN3YEFeGyA7vM9GA1AIqwAAADA; expires=Sat, 02-Sep-2023 01:35:28 GMT; path=/;
domain=.search.daum.net;, ODT=MS2Z_NNSZ_TWAZ_1DVZ_IIMZ_VOIZ_IVRZ_; expires=Wed, 03-Jul-2
024 01:35:28 GMT; path=/; domain=.search.daum.net;, DDT=GG2Z_SNYZ_LB2Z_DICZ_; expires=We
d, 03-Jul-2024 01:35:28 GMT; path=/; domain=.search.daum.net;, DTQUERY=%EB%89%B4%EC%A7%8
4%EC%8A%A4; expires=Wed, 03-Jul-2024 01:35:28 GMT; path=/; domain=.search.daum.net;', 'V
ary': 'Accept-Encoding', 'Content-Encoding': 'br', 'Server': 'kws', 'Cache-Control': 'pr
ivate, no-cache, max-age=0', 'Expires': '-1', 'X-Content-Type-Options': 'nosniff', 'X-XS
S-Protection': '1', 'Accept-CH': 'Sec-CH-UA-Arch, Sec-CH-UA-Bitness, Sec-CH-UA-Full-Vers
ion-List, Sec-CH-UA-Model, Sec-CH-UA-Platform-Version, Sec-CH-UA-WoW64', 'Permissions-Po
licy': 'ch-ua-arch=(self "https://sl.search.daum.net"), ch-ua-bitness=(self "https://sl.
search.daum.net"),  ch-ua-full-version-list=(self "https://sl.search.daum.net"),  ch-ua-
model=(self "https://sl.search.daum.net"),  ch-ua-platform-version=(self "https://sl.sea
rch.daum.net"),  ch-ua-wow64=(self "https://sl.search.daum.net")'})
```

In [54]:
```
p = re.compile(r'tit_main fn_tit_u[^>]+?>(.+?)</a')
list(map(lambda row:re.sub(r'^\s|\s$|<.+?>|\[.+?\]|[’‘“”…]', '', row),
         p.findall(resp.text)))
```

Out[54]:
```
['뉴진스, 팬미팅서 신곡 ETA 최초 공개또 음원차트 점령하겠네',
 '7년 추적한 뽕으로 BIFAN 진출한 뉴진스 아빠 250 뉴진스보다 뽕 앨범이 효자',
 ' OMG 뉴진스가 돌아온다... 하이브에 쏠린 눈',
 '뉴진스, 첫 팬미팅 성료신곡 ETA 무대 최초 공개']
```

In [55]:
```
p = re.compile(r'tit_main fn_tit_u.+?href="(.+?)"')
p.findall(resp.text)
```

Out[55]:
```
['https://v.daum.net/v/20230703080603970?f=o',
 'https://v.daum.net/v/20230704060012578?f=o',
 'https://v.daum.net/v/20230703073403490?f=o',
 'https://v.daum.net/v/20230703083900630?f=o']
```

In [68]:
```
# 검색결과 중 이미지 밑에 결과들, 제목과 링크 추출
# Browser - 개발자도구(<a ..... >)
# 다를 수 있음!!!
# 오른쪽버튼 - 소스보기 (우리가 req-resp) (<c- ..... >)
p = re.compile(r'tit-g clamp-g"> <a.+?>(.+?)</a>') # 노래
p = re.compile(r'<c-menu-share.+?data-title="(.+?)"')
p.findall(resp.text)
```

Out[68]:
```
['NewJeans',
 '뉴진스 신곡 ETA 팬미팅 선공개',
 '뉴진스 혜인',
 '뉴진스 2nd EP 신곡 티저 해외반응',
 '뉴진스 멤버 프로필 나이 키 국적 과거',
 '뉴진스(New Jeans) ASAP 티저',
 '&ldquo;완전 똑같은데&hellip;&rdquo; &amp;#39;뉴진스&amp;#39; 신곡, 이번에도 예측했다는 &amp;#39;무한도전&amp;#39;',
 '뉴진스, &amp;#39;뉴진스 돌풍&amp;#39; 다시 일으킬까...7월 21일 새 앨범 &amp;#39;겟 업']
```

In [69]:
```
url = 'https://search.naver.com/search.naver'
params = {
    'where':'nexearch',
    'query':'뉴진스'
}
resp = request('GET', url, params=params, headers=headers)
resp.status_code, resp.request.headers, resp.headers
```

Out[69]:
```
(200,
 {'user-agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.0.0 Safari/537.36', 'Accept-Encoding': 'gzip, deflate, br', 'Accept': '*/*', 'Connection': 'keep-alive'},
 {'Date': 'Tue, 04 Jul 2023 02:06:07 GMT', 'Content-Type': 'text/html; charset=UTF-8', 'Transfer-Encoding': 'chunked', 'Connection': 'keep-alive', 'Set-Cookie': 'page_uid=i6L+Bwp0YidsskRcBRZssssssVZ-470290; path=/; domain=.naver.com, _naver_usersession_=OzQMMwgfz0bzGRET/4kjGw==; path=/; expires=Tue, 04-Jul-23 02:11:07 GMT; domain=.naver.com, nx_ssl=2; Domain=.naver.com; Path=/; Expires=Thu, 03-Aug-2023 02:06:07 GMT;', 'X-Frame-Options': 'SAMEORIGIN', 'X-XSS-Protection': '1; report=/p/er/post/xss', 'Cache-Control': 'no-cache, no-store, must-revalidate, max-age=0', 'Pragma': 'no-cache', 'Referrer-Policy': 'unsafe-url', 'Vary': 'Accept-Encoding', 'Content-Encoding': 'gzip', 'Server': 'nxg', 'Accept-CH': 'Sec-CH-UA, Sec-CH-UA-Arch, Sec-CH-UA-Bitness, Sec-CH-UA-Full-Version-List, Sec-CH-UA-Mobile, Sec-CH-UA-Model, Sec-CH-UA-Platform, Sec-CH-UA-Platform-Version, Sec-CH-UA-WoW64'})
```

In [70]:
```
resp.url
```

Out[70]:
```
'https://search.naver.com/search.naver?where=nexearch&query=%EB%89%B4%EC%A7%84%EC%8A%A4'
```

In [80]:
```
p = re.compile(r'news_tit[^>]+?>(.+?)</a')
list(map(lambda row:re.sub(f'[{re.escape(punctuation)}]', '', row),
                           p.findall(resp.text)))
```

Out[80]:
```
['mark뉴진스mark 캠프 콘셉트 팬미팅도 수준 높네…신곡 ETA 첫 공개',
 '단독 시총 11조 회사 클라쓰…BTS 집 하이브 사옥 가보니 윤현주의 主食',
 '투자노트 ‘OMG’ mark뉴진스mark가 돌아온다 하이브에 쏠린 눈',
 'X why ZZ세대가 mark뉴진스mark 앨범을 기다리는 이유는 뭘까']
```

In [76]:
```
from string import punctuation
f'[{re.escape(punctuation)}]'
```

Out[76]:
```
'[!"\\#\\$%\\&\'\\(\\)\\*\\+,\\-\\./:;<=>\\?@\\[\\\\\\]\\^_`\\{\\|\\}\\~]'
```

In [85]:
```
p = re.compile(r'<a href="([^"]+)?" class="news_tit"[^>]+?>')
p.findall(resp.text)
```

Out[85]:
```
['http://www.newsis.com/view/?id=NISX20230703_0002360844&cID=10601&pID=10600',
 'https://www.hankyung.com/finance/article/202307027863i',
 'https://biz.chosun.com/stock/analysis-prospect/2023/07/03/KUY2KAL4FFAPVIQ3PHLA5EUPJA/?utm_source=naver&utm_medium=original&utm_campaign=biz',
 'https://isplus.com/article/view/isp202307030091']
```

In [86]:
```
from bs4 import BeautifulSoup
```

In [ ]:
```
XML - Well-formed / Validation(검증)
여는태그 닫는태그 쌍 - 예외 X, <></>
대소문자 구별해요 <DIV></div> X
겹치지 않음

HTML - Not Well-formed / Validation X(검증 X)
여는태그 닫는태그 쌍 X, (단일태그) <img, br, meta, ..>
대소문자 구별안함 <Div></dIv>
겹치지 않음 => 문제, parser DOM(Tree) 바뀜
```

In [87]:
```
html = '''
<html>
 <head></head>
 <body>
  <div>
   <p>
    <a>go to page</a>
   </p>
  </div>
 </body>
</html>
'''
#                   root(Document, DOM, OM)
#                   HTML
#           HEAD              BODY
#                              DIV
#                               P
#                               A
dom = BeautifulSoup(html, 'html.parser')
```

In [91]:
```
dom.html.body.div.p.a
```

Out[91]:
```
<a>go to page</a>
```

In [92]:
```
dom.a
```

Out[92]:
```
<a>go to page</a>
```

In [93]:
```
dom.html.body.div.p.a == dom.a
```

Out[93]:
```
True
```

In [95]:
```
type(dom), type(dom.html), type(dom.a)
```

Out[95]:
```
(bs4.BeautifulSoup, bs4.element.Tag, bs4.element.Tag)
```

In [96]:
```
dom.a.attrs => HTML Attributes {Key:Value}
<tag attributes... key=value></tag>
=> Node, tag:node name, attrs={k:v...}
```

Out[96]:
```
{}
```

In [97]:
```
dom.a.has_attr('href')
```

Out[97]:
```
False
```

In [99]:
```
type(dom.a.asdflkjasdflkjasdflkj)
```

Out[99]:
```
NoneType
```

In [100]…
```
dom.a.span.b
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In [100], line 1
----> 1 dom.a.span.b

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

In [113]…
```
html = '''
<html>  <!-- 대소문자 불일치 -->
 <head><head>
  <body>
   <div>
    <ul>
     <li> <!-- 닫는 태그 없음 -->
     <li><p><div></li> <!-- 부모-자식 순서 중첩 -->
    </ul>
  </body>
</HTML>
'''
#                    DOM-document
#                       html
#               head              body
#                                  div
#                                  ul
#                              li      li
```

In [114]…
```
BeautifulSoup(html, 'html.parser')
#                                  head
#                                   ul
#                                   li
#                                   li
```

Out[114]:
```
<html> <!-- 대소문자 불일치 -->
<head><head>
<body>
<div>
<ul>
<li> <!-- 닫는 태그 없음 -->
<li><p><div></div></p></li> <!-- 부모-자식 순서 중첩 -->
</li></ul>
</div></body>
</head></head></html>
```

In [115]…
```
BeautifulSoup(html, 'lxml')
```

Out[115]:
```
<html> <!-- 대소문자 불일치 -->
<head>
</head><body>
<div>
<ul>
<li> <!-- 닫는 태그 없음 -->
</li><li><p></p><div> <!-- 부모-자식 순서 중첩 -->
</div></li></ul></div></body></html>
```

In [116]…
```
BeautifulSoup(html, 'html5lib')
```

Out[116]:
```
<html><!-- 대소문자 불일치 --><head>
  </head><body>
   <div>
    <ul>
     <li> <!-- 닫는 태그 없음 -->
     </li><li><p></p><div></div></li> <!-- 부모-자식 순서 중첩 -->
    </ul>


</div></body></html>
```

In [119]…
```
html = '''
<html>
 <head></head>
 <body>
  <div>
   <p>
    <a id="1" href="a">go to page</a>
    <a id="2" href="b">go to page</a>
   </p>
  </div>
  <a id="3" href="c">go to page</a>
 </body>
</html>
'''
dom = BeautifulSoup(html, 'html.parser')
```

In [120]…
```
dom.body.div.p.a == dom.a
# Tree를 Node를 따라 가면, 가장 첫번째 있는 a
```

Out[120]:
```
True
```

In [122]…
```
dom.a.attrs
```

Out[122]:
```
{'id': '1', 'href': 'a'}
```

In [128]…
```
list(dom.p.children)[3].attrs
```

Out[128]:
```
{'id': '2', 'href': 'b'}
```

In [129]…
```
list(dom.p.children)[1] == dom.a
```

Out[129]:
```
True
```

In [126]…
```
list(map(lambda x:type(x), list(dom.p.children)))
```

Out[126]:
```
[bs4.element.NavigableString,
 bs4.element.Tag,
 bs4.element.NavigableString,
 bs4.element.Tag,
 bs4.element.NavigableString]
```

In [ ]:
```
dom.find(자식/손), find_all(자식/손들)
    find_parent(부모), find_parents(조상)

    공통된 부모를 공유하는 = 부모가 같음
    find_previous_sibling(s)(내 앞에 나온 형제/들)
    find_next_sibling(s)(내 다음에 나온 형제/들)
```

In [131]…
```
dom.find('a') == dom.a == dom.html.body.div....a
```

Out[131]:
```
True
```

In [134]…
```
dom.find(attrs={'id':1}) == dom.find('a') == dom.a
```

Out[134]:
```
True
```

In [137]…
```
dom.find(attrs={'id':2}) == list(dom.p.children)[3]
```

Out[137]:
```
True
```

In [140]…
```
dom.find(string='go to page').find_parent() == dom.a
```

Out[140]:
```
True
```

In [142]…
```
dom.find(string=re.compile('page$')).find_parent() == dom.a
```

Out[142]:
```
True
```

In [146]…
```
dom.find(attrs={'href':re.compile('a')}) == dom.a
```

Out[146]:
```
True
```

In [149]…
```
dom.find(re.compile('^a$')) == dom.a
```

Out[149]:
```
True
```

In [151]…
```
dom.body.find('a', recursive=False)
                  HTML
          head              body(*)
                     div         a:3
                      p
                   a:1    a:2
```

Out[151]:
```
<a href="c" id="3">go to page</a>
```

In [152]…
```
dom.body.find('a', recursive=True)
```

Out[152]:
```
<a href="a" id="1">go to page</a>
```

In [154]…
```
len(dom.find_all('a')), len(dom.find_all('a', limit=1))
```

Out[154]: (3, 1)

In [155]…
```
dom.find_all('a', limit=1)[0] == dom.find('a') == dom.a
```

Out[155]: True

In [156]…
```
dom.find_all('a', limit=3)[-1] == dom.body.find('a', recursive=False)
```

Out[156]: True

In [160]…
```
dom.a.find_parent('div')
```

Out[160]:
```
<div>
<p>
<a href="a" id="1">go to page</a>
<a href="b" id="2">go to page</a>
</p>
</div>
```

In [165]…
```
for node in dom.a.find_parents(limit=3):
    print(node.name)
```

```
p
div
body
```

In [166]…
```
dom.a.find_next_sibling()
```

Out[166]: go to page

In [169]…
```
dom.div.find_next_sibling().find_previous_sibling() == dom.div
```

Out[169]: True

In [170]…
```
url = 'https://pythonscraping.com/pages/page3.html'
resp = request('GET', url)
```

In [173]…
```
resp.headers, type(resp.content)
```

Out[173]:
```
({'Server': 'nginx', 'Date': 'Tue, 04 Jul 2023 04:01:29 GMT', 'Content-Type': 'text/html', 'Last-Modified': 'Sat, 09 Jun 2018 19:15:59 GMT', 'Transfer-Encoding': 'chunked', 'Connection': 'keep-alive', 'ETag': 'W/"5b1c276f-965"', 'X-Powered-By': 'PleskLin', 'Content-Encoding': 'br'},
 bytes)
```

In [174]…
```
dom1 = BeautifulSoup(resp.content, 'html.parser')
dom2 = BeautifulSoup(resp.content, 'lxml')
dom3 = BeautifulSoup(resp.content, 'html5lib')
```

In [182]…
```
[tag.name for tag in dom1.div.find_all(recursive=False)]
```

Out[182]: ['img', 'h1', 'div', 'table', 'div']

In [183]…
```
[tag.name for tag in dom2.div.find_all(recursive=False)]
```

Out[183]: ['img', 'h1', 'div', 'table', 'div']

In [184]…
```
[tag.name for tag in dom3.div.find_all(recursive=False)]
```

Out[184]:
```
['img', 'h1', 'div', 'table', 'p', 'div']
```

In [188]…
```
base = dom3.div.find_all(recursive=False)[-2]
```

In [190]…
```
[tag.name for tag in base.find_parents()]
```

Out[190]:
```
['div', 'body', 'html', '[document]']
```

In [191]…
```
base.find_next_sibling()
```

Out[191]:
```
<div id="footer">
© Totally Normal Gifts, Inc. <br/>
+234 (617) 863-0736
</div>
```

In [194]…
```
base.find_parent().find_all('div', recursiv=False)[-1]
```

Out[194]:
```
<div id="footer">
© Totally Normal Gifts, Inc. <br/>
+234 (617) 863-0736
</div>
```

In [ ]:
```
       document
         html
head           body
               div
        tag   tag   tag  p(*)  div
```

In [196]…
```
base.find_parent().find('table', recursive=False).find_all('td')[-1]
```

Out[196]:
```
<td>
<img src="../img/gifts/img6.jpg"/>
</td>
```

In [197]…
```
base.find_previous_sibling().find_all('td')[-1]
```

Out[197]:
```
<td>
<img src="../img/gifts/img6.jpg"/>
</td>
```

In [209]…
```
for row in base.find_previous_sibling().find('tbody')\
            .find_all(recursive=False)[1:]:
    print(row.find_all('td', recursive=False)[2].text.strip())
```

```
$15.00
$10,000.52
$10,005.00
$0.50
$1.50
```