

LESSON

BeautifulSoup 모듈 응용

```
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):

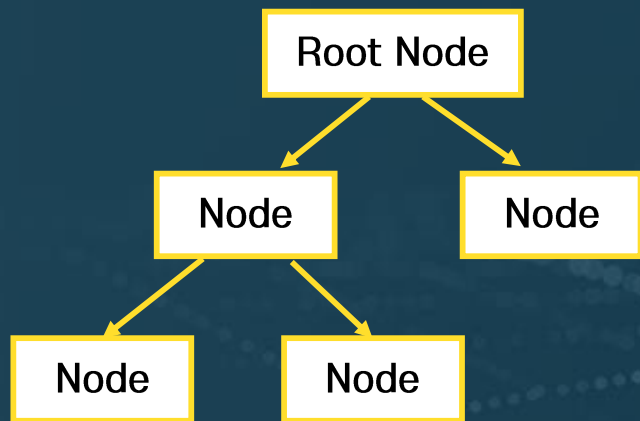
        # change frame, redraw view
        moveTime = c4d.BaseTime(fromTime,docFps) + c4d.BaseTime(x)
        doc.SetTime(moveTime)
        c4d.EventAdd(c4d.EVENT_FORCEREDRAW)
        c4d.DrawViews(c4d.DRAWFLAGS_FORCEFULLREDRAW)

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

        # get buffer object
        bufferedNumber = str(doc.GetTime().GetFrame(docFps-x))
        if len(bufferedNumber) < 4:
```

DOM(Document Object Model)의 정의

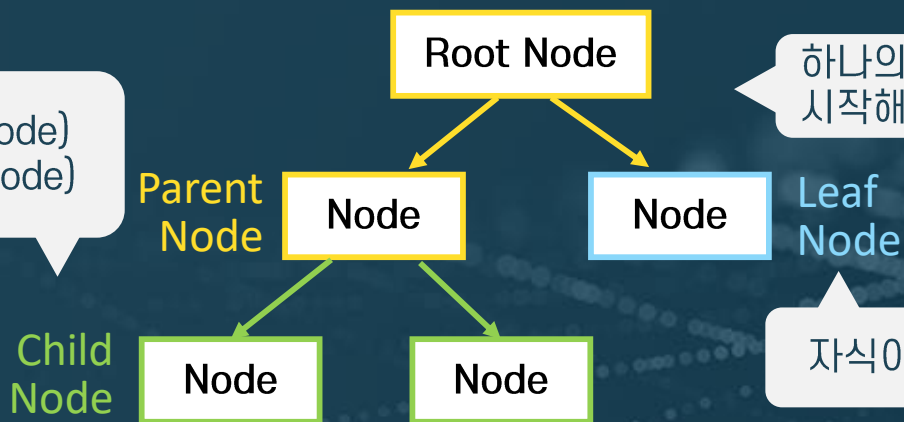
- ✦ HTML, XML 문서의 프로그래밍 인터페이스
 - 구조화된 표현 및 프로그래밍 언어가 DOM 구조에 접근할 수 있는 방법을 제공함
- ✦ **트리 구조**로 형성되어 있음



DOM(Document Object Model)의 정의

- HTML, XML 문서의 프로그래밍 인터페이스
 - 구조화된 표현 및 프로그래밍 언어가 DOM 구조에 접근할 수 있는 방법을 제공함
- 트리 구조로 형성되어 있음

위쪽 : 부모 노드(Parent Node)
아래쪽 : 자식 노드(Child Node)

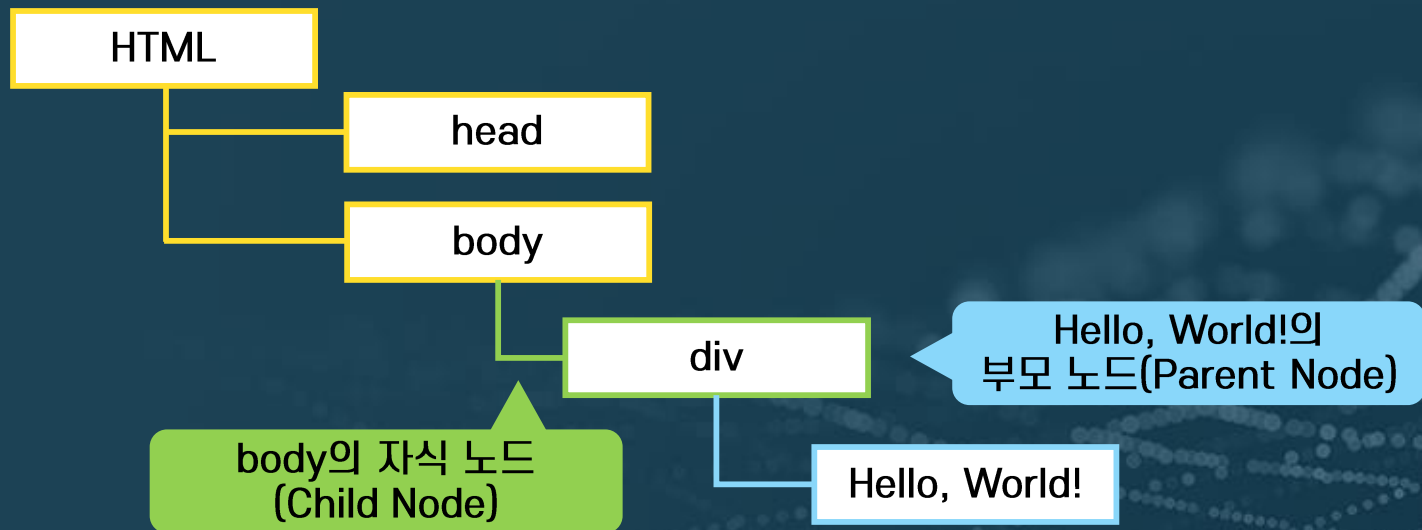


하나의 Root node에서
시작해 퍼져나가는 형태

자식이 없는 노드(Node)

DOM(Document Object Model)의 정의

- HTML에서 node는 <head>, <body>, <h1>, <script> 등의 태그뿐만 아니라 태그 내 텍스트나 속성 모두 노드(Node)에 속함



노드(Node)를 활용한 검색

♣ BeautifulSoup 모듈의 함수를 활용하여 노드를 기준으로 원하는 데이터 추출



children :

자식 노드들을 반복 가능한 객체로 반환

```
html=" "  
<html>  
  <head>  
    <title>crawl</title>  
  </head>  
  <body>  
    <p class="a" align="center"> text1 </p>  
    <p class="b" align="center"> text2 </p>  
    <p class="c" align="center"> text3 </p>  
    <div>  
        
    </div>  
  </body>  
</html> "
```

```
from bs4 import BeautifulSoup  
  
soup=BeautifulSoup(html, "html.parser")  
contents = soup.find('body')  
  
for child in contents.children :  
    print (child)
```

<p align="center" class="a"> text1 </p>

<p align="center" class="b"> text2 </p>

<p align="center" class="c"> text3 </p>

<div>

</div>

노드(Node)를 활용한 검색

- BeautifulSoup 모듈의 함수를 활용하여 노드를 기준으로 원하는 데이터 추출



parent :

자신을 포함한 부모 노드까지 출력

```
html=""
<html>
  <head>
    <title>crawl</title>
  </head>
  <body>
    <p class="a" align="center"> text1 </p>
    <p class="b" align="center"> text2 </p>
    <p class="c" align="center"> text3 </p>
    <div>
      
    </div>
  </body>
</html> ""
```

```
from bs4 import BeautifulSoup
```

```
soup = BeautifulSoup(html, "html.parser")
contents = soup.find('body')
```

```
img_tag = contents.find('img')
print(img_tag)
```

```

```

```
print(img_tag.parent)
```

```
<div>
```

```

```

```
</div>
```

노드(Node)를 활용한 검색

♣ BeautifulSoup 모듈의 함수를 활용하여 **노드를 기준으로 원하는 데이터 추출**



find_parent('부모 노드') :
특정 부모 노드까지 검색해서 올라감

```
html=""
<html>
  <head>
    <title>crawl</title>
  </head>
  <body>
    <p class="a" align="center"> text1 </p>
    <p class="b" align="center"> text2 </p>
    <p class="c" align="center"> text3 </p>
    <div>
      
    </div>
  </body>
</html> ""
```

```
from bs4 import BeautifulSoup
```

```
soup = BeautifulSoup(html, "html.parser")
contents = soup.find('body')
```

```
img_tag = contents.find('img')
print(img_tag)
```

```

```

```
print(img_tag.find_parent('body'))
```

```
<body>
<p align="center" class="a"> text1 </p>
<p align="center" class="b"> text2 </p>
<p align="center" class="c"> text3 </p>
<div>

</div>
</body>
```

노드(Node)를 활용한 검색

♣ BeautifulSoup 모듈의 함수를 활용하여 **노드를 기준으로 원하는 데이터 추출**



find_next_sibling() : 바로 다음 형제 노드를 검색

find_next_siblings() : 모든 형제 노드를 검색

```
html=" "  
<html>  
  <head>  
    <title>crawl</title>  
  </head>  
  <body>  
    <p class="a" align="center"> text1 </p>  
    <p class="b" align="center"> text2 </p>  
    <p class="c" align="center"> text3 </p>  
    <div>  
        
    </div>  
  </body>  
</html> "
```

```
from bs4 import BeautifulSoup
```

```
soup=BeautifulSoup(html, "html.parser")  
contents = soup.find('body')
```

```
p_tag = bs.find("p",class_="b")  
print(p_tag)
```

```
<p align="center" class="b"> text2 </p>
```

```
print(p_tag.find_next_sibling())  
print(p_tag.find_next_siblings())
```

```
<p align="center" class="c"> text3 </p>  
[<p align="center" class="c"> text3 </p>, <div>  
  
</div>]
```


노드(Node)를 활용한 검색

- BeautifulSoup 모듈의 함수를 활용하여 **노드를 기준으로 원하는 데이터 추출**



find_previous_sibling() : 바로 이전 형제 노드를 검색

find_previous_siblings() : 모든 형제 노드를 검색

```
html=" "
<html>
  <head>
    <title>crawl</title>
  </head>
  <body>
    <p class="a" align="center"> text1 </p>
    <p class="b" align="center"> text2 </p>
    <p class="c" align="center"> text3 </p>
    <div>
      
    </div>
  </body>
</html> "
```

```
from bs4 import BeautifulSoup
```

```
soup=BeautifulSoup(html, "html.parser")
contents = soup.find('body')
```

```
p_tag = bs.find("p",class_="b")
print(p_tag)
```

```
<p align="center" class="b"> text2 </p>
```

```
print(p_tag.find_previous_sibling())
print(p_tag.find_previous_siblings())
```

```
<p align="center" class="a"> text1 </p>
[<p align="center" class="a"> text1 </p>]
```

LESSON

selenium 모듈 응용

```
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,docFps)
        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 0.001
        bufferedNumber = str(doc.GetTime().GetFrame(docFps-x))
        if len(bufferedNumber) < 3:
```

폼 양식 전송 실습

- selenium + Web driver를 활용하여 버튼을 클릭하거나 자바스크립트 삽입 등 특정 조건이 충족되어야만 접근 가능한 데이터를 모듈을 활용하여 가져 올 수 있음



[활용 예제]
폼 양식을 이용해
Naver 로그인하기 실습

Naver 로그인하기 Step1

로그인 페이지 접속

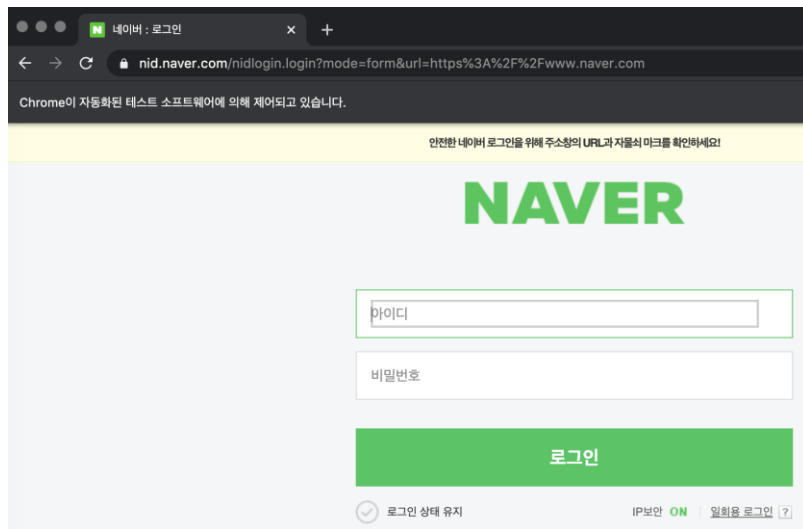


```
<!-- //헤더 -->
<div style="position:relative;width:1000px;ma
</div>
<div class="container" role="main">
  <div class="column_left"></div>
  <div class="column_right">
    <!-- 로그인 -->
    <div id="account" class="section_login">
      <h2 class="blind">로그인</h2>
      <div class="lg_local">
        <p class="lg_local_text">네이버를 더 안전
        <a class="lg_local_btn" href="https://
        nidlogin.login?mode=form&url=https%3A%2
        "button" data-clk="log_off.login">
      <i class="ico_local_login lang_ko">
```

```
import selenium
from selenium import webdriver

path = "/Users/zoostar/Downloads/chromedriver"
driver = webdriver.Chrome(path)
driver.get('https://www.naver.com')

element = driver.find_element_by_class_name("lg_local_btn")
element.click()
```



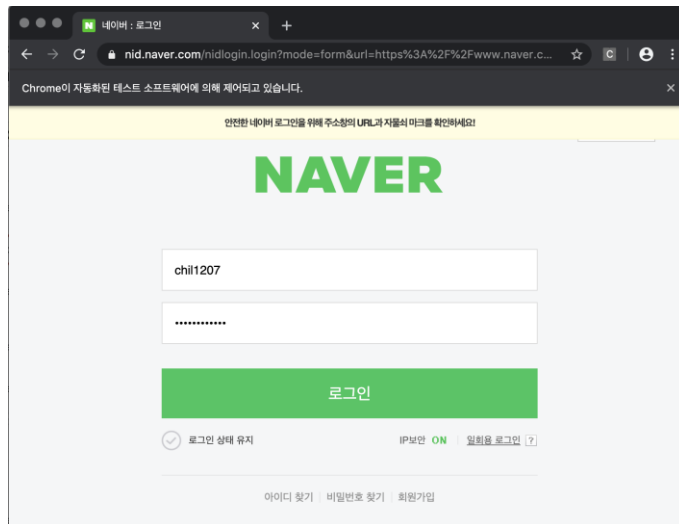
Naver 로그인하기 Step2

로그인 정보 입력(아이디, 비밀번호)

```
import selenium
from selenium import webdriver

path = "/Users/zoostar/Downloads/chromedriver"
driver = webdriver.Chrome(path)
driver.get('https://www.naver.com')

element = driver.find_element_by_class_name("lg_local_btn")
element.click()
id = 'chill1207'
#pw = '비밀번호'
element = driver.find_element_by_id('id')
element.send_keys(id)
time.sleep(1)
element = driver.find_element_by_id('pw')
element.send_keys(pw)
```



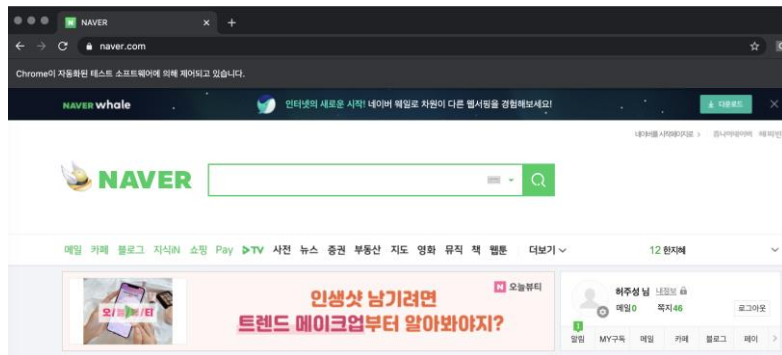
Naver 로그인하기 Step3

로그인 버튼 클릭

```
import selenium
from selenium import webdriver

path = "/Users/zoostar/Downloads/chromedriver"
driver = webdriver.Chrome(path)
driver.get('https://www.naver.com')

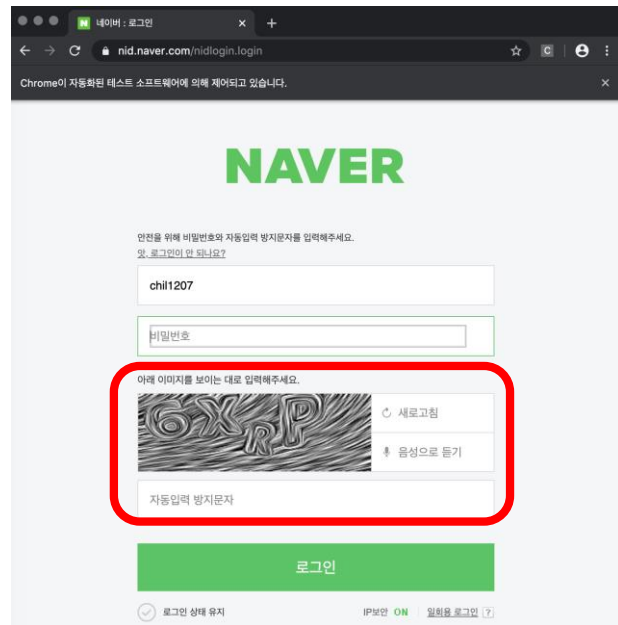
element = driver.find_element_by_class_name("lg_local_btn")
element.click()
id = 'chill1207'
#pw = '비밀번호'
element = driver.find_element_by_id('id')
element.send_keys(id)
time.sleep(1)
element = driver.find_element_by_id('pw')
element.send_keys(pw)
time.sleep(1)
element = driver.find_element_by_class_name('btn_global')
element.click()
```



Naver 로그인하기 Step4

캡차(CAPCHAR) 인증

- ✓ 일부 홈페이지의 경우 정상적인 접근이 아닌 경우
자동 로그인을 방지하기 위한
캡차(CAPCHAR)를 도입 중
- ✓ Naver도 18년도부터 캡차 적용
- ✓ 매번 뜨는 것은 아니지만
자동화된 크롤러를 만들기엔 불편함



Naver 로그인하기 Step5

id/pw 입력 처리

- ✓ 자바스크립트로 데이터 입력하기
- ✓ 로그인 폼의 id와 pw 요소에 직접 데이터 전달(send_keys보다 훨씬 안정적!)

```
import selenium
from selenium import webdriver

path = "/Users/zoostar/Downloads/chromedriver"
driver = webdriver.Chrome(path)
driver.get('https://www.naver.com')

element = driver.find_element_by_class_name("lg_local_btn")
element.click()
id = 'chill1207'
#pw = '비밀번호'
driver.execute_script("document.getElementById('id').value='" + id + "'")
time.sleep(1)
driver.execute_script("document.getElementById('pw').value='" + pw + "'")
time.sleep(1)

element = driver.find_element_by_class_name('btn_global')
element.click()
```

```
----- 로그인 -----
<legend class="blind">로그인</legend>
<div class="id_area">
  <div class="input_row" id="id_area">
    <span class="input_box">
      <label for="id" id="label_id_area" class="lbl">아이디</label>
      <input type="text" id="id" name="id" accesskey="I" placeholder="아이디를 입력하세요" />
    </span>
    <button type="button" disabled="" title="delete" id="id_clear" class="btn_delete">X</button>
  </div>
  <div class="error" id="err_empty_id" style="display:none" aria-live="polite">아이디를 입력하세요</div>
</div>
<div class="pw_area">
  <div class="input_row" id="pw_area">
    <span class="input_box">
      <label for="pw" id="label_pw_area" class="lbl">비밀번호</label>
      <input type="password" id="pw" name="pw" placeholder="비밀번호를 입력하세요" />
    </span>
  </div>
  <div class="error" id="err_empty_pw" style="display:none" aria-live="polite">비밀번호를 입력하세요</div>
</div>
```


LESSON

BeautifulSoup + selenium 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 0.001
        bufferedNumber = str(doc.GetTime().GetFrame(docFps - 1))
        if len(bufferedNumber) < 3:
```

메일 목록 가져오기 실습

- BeautifulSoup과 selenium 모듈을 함께 사용하면 보다 까다로운 데이터를 가져올 수 있음

BeautifulSoup 모듈

로그인이 필요하거나
어떠한 버튼 등을 클릭한 뒤
나오는 페이지의 정보들을
가져오기 어려움

selenium 모듈

수 많은 데이터들을
손쉽게 가져오기 어려움

두가지 모듈을
함께 활용

BeautifulSoup으로 접근하기 어려운 페이지를
selenium으로 접속하여 BeautifulSoup로 분석

메일 목록 가져오기 Step1

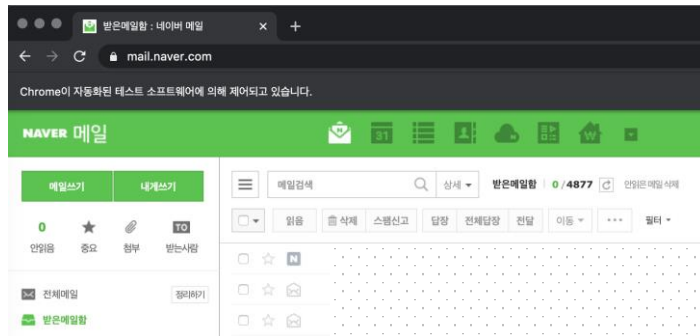
selenium을 활용한 Naver 로그인 및 메일 탭 클릭

```
import selenium
from selenium import webdriver

path = "/Users/zoostar/Downloads/chromedriver"
driver = webdriver.Chrome(path)
driver.get('https://www.naver.com')

element = driver.find_element_by_class_name("lg_local_btn")
element.click()
id = 'chill207'
#pw = '비밀번호'
driver.execute_script("document.getElementById('id').value='" + id + "'")
time.sleep(1)
driver.execute_script("document.getElementById('pw').value='" + pw + "'")
time.sleep(1)

element = driver.find_element_by_class_name('btn_global')
element.click()
time.sleep(1)
element = driver.find_element_by_class_name('mn_mail')
element.click()
```



메일 목록 가져오기 Step2

메일함 페이지 소스 가져오기

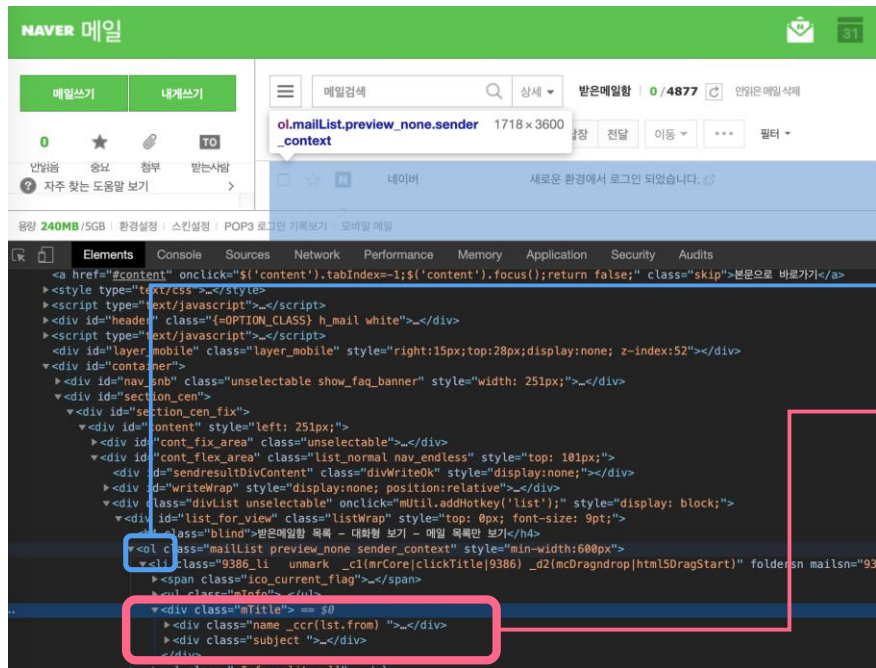
```
page_source = driver.page_source
soup = BeautifulSoup(page_source, "html.parser")
print(soup)
```

selenium과
BeautifulSoup의 연결

```
<html lang="ko"><head>
<meta content="IE=edge" http-equiv="X-UA-Compatible"/>
<meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
<link href="/css/mail2_le.min.191121.css" rel="stylesheet" type="text/css"/>
<link href="/css/mail_skin.min.191121.css" rel="stylesheet" type="text/css"/>
<link href="//ssl.pstatic.net" rel="dns-prefetch"/>
<link href="//lcs.naver.com" rel="dns-prefetch"/>
<link href="//cc.naver.com" rel="dns-prefetch"/>
<link href="//static.nid.naver.com" rel="dns-prefetch"/>
<link href="//phinf.pstatic.net" rel="dns-prefetch"/>
<link href="/css/mail_me.min.191121.css" rel="stylesheet" type="text/css"/>
<link href="/favicon.ico?2" rel="shortcut icon"/>
<title>받은메일함 : 네이버 메일</title>
```

메일 목록 가져오기 Step3

BeautifulSoup 모듈을 활용한 메일 목록 파싱(태그, 속성 및 패턴 분석)



메일 리스트 ,

각 메일 항목 <div>

메일 목록 가져오기 Step3

BeautifulSoup 모듈을 활용한 메일 목록 파싱(태그, 속성 및 패턴 분석)

```
soup = BeautifulSoup(page_source, "html.parser")
mail_list = soup.select_one("div.mTitle")
print(mail_list)
```

```
<div class="mTitle"><div class="name _ccr(lst.from) "><span class="blind">보낸 이:</span><a class="_c1(myContextMenu|s
howSenderContextLayer|list|9386) _stopDefault" href="#" title="'네이버' &lt;help@help.naver.com&gt;'>네이버</a></div><div class="subject "><a class="_d2(mcDragndrop|html5DragStart)" draggable="true" href="/read/popup/?nMailId=9386"><span class="text _ccr(lst.title) _c1(mrCore|clickTitle|9386) _m2(mrCore|middleClickTitle|9386)" foldersn="" mailsn="9386">
<span class="folderName _ccr(lst.title) _c1(mrCore|clickTitle|9386) _m2(mrCore|middleClickTitle|9386)" foldersn="" mailsn="9386"><span class="blind">에 분류됨</span></span><strong class="mail_title _ccr(lst.title) _c1(mrCore|clickTitle|9386) _m2(mrCore|middleClickTitle|9386)" foldersn="" mailsn="9386"><span class="blind">메일 제목:</span>새로운 환경에서 로그인 되었습니다.</strong></span></a><a class="icon_previewMail" href="#" title="메일 내용 미리보기"><span class="blind">메일 내용 미리보기</span></a><a class="newWindow _ccr(lst.newwindow) _c1(mrCore|popupRead|9386) _m1(mlCore|doStop) _stopDefault" href="#" title="새 창으로 메일 보기"><span class="spr"><em class="blind">새 창으로 메일 보기</em></span></a></div></div>
```

메일 목록 가져오기 Step4

BeautifulSoup 모듈을 활용한 메일 제목, 내용 파싱

```
print(mail_list.select_one('a').text)
```

네이버

```
print(mail_list.select_one('strong').text)
```

메일 제목: 새로운 환경에서 로그인 되었습니다.

```
print(mail_list.prettify())

<div class="mTitle">
<div class="name_ccr(lst.from) ">
<span class="blind">
보낸 이:
</span>
<a class="c1(myContextMenu|showSenderContextLayer|list|9386) stopDefault" href="#" title="네이버" &lt;help@help.n
aver.com&gt; ">
네이버
</a>
</div>
<div class="subject ">
<a class="d2(mcDragndrop|html5DragStart)" draggable="true" href="/read/popup/?nMailId=9386">
<span class="text_ccr(lst.title) _c1(mrCore|clickTitle|9386) _m2(mrCore|middleClickTitle|9386)" foldersn="" mails
n="9386">
<span class="folderName_ccr(lst.title) _c1(mrCore|clickTitle|9386) _m2(mrCore|middleClickTitle|9386)" foldersn
="" mailsn="9386">
<span class="blind">
에 분류됨
</span>
</span>
</span>
<strong class="mail_title _ccr(lst.title) _c1(mrCore|clickTitle|9386) _m2(mrCore|middleClickTitle|9386)" foldersn
="" mailsn="9386">
<span class="blind">
메일 제목:
</span>
</strong>
새로운 환경에서 로그인 되었습니다.
</strong>
</span>
</a>
<a class="icon_previewMail" href="#" title="메일 내용 미리보기">
<span class="blind">
메일 내용 미리보기
</span>
</a>
<a class="newWindow_ccr(lst.newwindow) _c1(mrCore|popupRead|9386) _m1(m1Core|doStop) _stopDefault" href="#" title
="" 새 창으로 메일 보기">
<span class="spr">
<em class="blind">
새 창으로 메일 보기
</em>
</span>
</a>
</div>
</div>
```

메일 목록 가져오기 Step5

반복문을 활용한 메일 목록 파싱 및 코드 정리

```
import selenium
from selenium import webdriver

path = "/Users/zoostar/Downloads/chromedriver"
driver = webdriver.Chrome(path)
driver.get('https://www.naver.com')

element = driver.find_element_by_class_name("lg_local_btn")
element.click()
id = 'chill1207'
#pw = '비밀번호'
driver.execute_script("document.getElementById('id').value='" + id + "'")
time.sleep(1)
driver.execute_script("document.getElementById('pw').value='" + pw + "'")
time.sleep(1)

element = driver.find_element_by_class_name('btn_global')
element.click()
time.sleep(1)
element = driver.find_element_by_class_name('mn_mail')
element.click()

page_source = driver.page_source
soup = BeautifulSoup(page_source, "html.parser")

mail_list = soup.select("div.mTitle")
cnt = 1
for i in mail_list:
    print(cnt)
    print('메일 제목 : ', i.select_one('a').text)
    print('메일 내용 : ', str(i.select_one('strong').text).split(':')[1])
    print()
    cnt += 1
```

1

메일 제목 :

메일 내용 :

2

메일 제목 :

메일 내용 :

3

메일 제목 :

메일 내용 :

4

메일 제목 :

메일 내용 :

5

메일 제목 :

메일 내용 :