DOM(Document Object Model)의 정의

- HTML, XML 문서의 프로그래밍 인터페이스 : 구조화된 표현 및 프로그래밍 언어가 DOM 구조에 접근할 수 있는 방법을 제공
- 트리 구조로 형성되어 있음 : 부모 노드(위쪽), 자식 노드(아래쪽)
- HTML에서 노드는 <head>, <body>, <h1>, <script> 등의 태그뿐만 아니라 태그 내 텍스트나 속성 모두 노드에 속함
- BeautifulSoup 모듈의 함수를 활용하여 노드를 기준으로 원하는 데이터 추출

In [1]:
```
html="""
<html>
<head>
    <title>crawler</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>
        <img src="/source" width="300" height="200">
    </div>
</body>
</html>
"""

from bs4 import BeautifulSoup

bs = BeautifulSoup(html,'html.parser')
contents = bs.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>
<img height="200" src="/source" width="300"/>
</div>
```

In [2]:
```
# body의 자손은 p, div, img
for d in contents.descendants:
    print(d)
```

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


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


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


<div>
<img height="200" src="/source" width="300"/>
</div>


<img height="200" src="/source" width="300"/>
```

In [3]:
```
img_tag = contents.find('img')
print(img_tag)
print(img_tag.parent)
```

```
<img height="200" src="/source" width="300"/>
<div>
<img height="200" src="/source" width="300"/>
</div>
```

In [5]:
```
contents = bs.find('body')
print(img_tag.find_parent('body'),'\n')
print(img_tag.find_parent('div'))
```

```
<body>
<p align="center" class="a"> text1</p>
<p align="center" class="b"> text2</p>
<p align="center" class="c"> text3</p>
<div>
<img height="200" src="/source" width="300"/>
</div>
</body>

<div>
<img height="200" src="/source" width="300"/>
</div>
```

In [6]:
```
p_tag = bs.find('p',class_='b')
print(p_tag)
```

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

In [7]:
```
from urllib import request as req
from bs4 import BeautifulSoup

res = req.urlopen('https://naver.com')
bs = BeautifulSoup(res,'html.parser')
print(bs.find('a'),'\n')
print(bs.find(class_='link_newsstand'),'\n')
print(bs.find('a',{'class':'link_newsstand'}),'\n')

# 클래스가 여러개인 경우
eles = bs.find_all('a',{'class':['link_newsstand','btn_sort','
for e in eles:
    print(e.text)
```

```
<a href="#newsstand"><span>뉴스스탠드 바로가기</span></a>

<a class="link_newsstand" data-clk="title" href="http://news
stand.naver.com/" target="_blank">뉴스스탠드</a>

<a class="link_newsstand" data-clk="title" href="http://news
stand.naver.com/" target="_blank">뉴스스탠드</a>

뉴스스탠드
구독한 언론사
전체언론사
```

In [8]:

```
hlists = bs.findAll({'h1','h2','h3','h4','h5','h6'},limit = 3)
for h in hlists:
    print(h,'\n')
```

```
<h1 class="logo_default">
<a class="logo_naver" data-clk="top.logo" href="/"><span cla
ss="blind">네이버</span></a>
</h1>

<h2 class="blind">뉴스스탠드</h2>

<h2 class="blind">주제별 캐스트</h2>
```

In [9]:

```
# 정규표현식과 bs4
from urllib.request import urlopen
from bs4 import BeautifulSoup
import re

html = urlopen('http://www.pythonscraping.com/pages/page3.html
bs = BeautifulSoup(html,'html.parser')
images = bs.find_all('img',{'src':re.compile('\.\.\/img\/gifts
for image in images:
    print(image['src'])
```

```
../img/gifts/img1.jpg
../img/gifts/img2.jpg
../img/gifts/img3.jpg
../img/gifts/img4.jpg
../img/gifts/img6.jpg
```

In [ ]:

```
# 한빛 네트워크 사이트 로그인 후 점수 가져오기

import time
import selenium
from selenium import webdriver

driver = webdriver.Chrome('C:/tool/chromedriver.exe')
driver.get('https://www.hanbit.co.kr/')
element = driver.find_element_by_class_name('login')
element.click()
m_id = ''
m_passwd = ''

element = driver.find_element_by_id('m_id')
element.send_keys(m_id)
time.sleep(1)

element = driver.find_element_by_id('m_passwd')
element.send_keys(m_passwd)
time.sleep(1)

element = driver.find_element_by_class_name('btn_login')
element.click()

driver.get('https://www.hanbit.co.kr/myhanbit/myhanbit.html')
source = driver.page_source
bs = BeautifulSoup(source, 'html.parser')
a = bs.select_one('#container > div > div.sm_mymileage > dl.mi
print(a.text,'점')
time.sleep(3)
driver.close()
```

header 확인: http://www.useragentstring.com/ (http://www.useragentstring.com/)

In [12]:

```
# 구글 플레이에서 인기 영화 제목 30개 출력(requests + bs4)
import requests
from bs4 import BeautifulSoup

headers = {'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWeb

url = 'https://play.google.com/store/movies/top'

res = requests.get(url)
bs = BeautifulSoup(res.text,'html.parser')
movies = bs.find_all('div',class_='WsMG1c nnK0zc')

print(len(movies))
for m in movies:
    print(m.text)
```

```
30
Venom: Let There Be Carnage
No Time To Die
Free Guy
Ice Age 5-Movie Collection
Illumination Presents: Dr. Seuss' The Grinch
The Hating Game
Spider-Man: Far from Home
Dr. Seuss' How the Grinch Stole Christmas
Dune
Shang-Chi and the Legend of the Ten Rings
Yellowstone
It's Always Sunny in Philadelphia
Rick and Morty (Uncensored)
Doctor Who
South Park
The Office
Game of Thrones
Naruto Shippuden Uncut
The Flash
The Big Bang Theory
Keep the Wolves Close
Winning Or Learning
Phantom Pain
Everybody Pays on the New Season of Yellowstone
Yellowstone - A Long Line of Enemies
1883: Two Journeys Extended Teaser
Compliments of Captain Lee's Travel Agency
2020: A Year In Review
The Gang Buys a Roller Rink
The Gang Replaces Dee with a Monkey
```

In [19]:

```
# Q. 구글 플레이에서 인기 영화 제목 200개 출력(selenium + bs4)
from selenium import webdriver

driver = webdriver.Chrome('C:/tool/chromedriver.exe')
driver.maximize_window()

url = 'https://play.google.com/store/movies/top'
driver.get(url)

# 1080 위치로 스크롤 내리기
# driver.execute_script('window.scrollTo(0,1080)')
# 화면 가장 아래로 스크롤 내리기
# driver.execute_script('window.scrollTo(0,document.body.scro

import time

# 현재 문서 높이를 가져와서 저장
prev_height = driver.execute_script('return document.body.scro

# 반복 수행
while True:
    driver.execute_script('window.scrollTo(0,document.body.scr
    # 페이지 로딩 대기
    driver.implicitly_wait(10) # 브라우저에서 파싱되는 지연 시간
    curr_height = driver.execute_script('return document.body.
    if curr_height == prev_height:
        break
    prev_height = curr_height
print('스크롤 완료')

bs = BeautifulSoup(driver.page_source,'html.parser')
movies = bs.find_all('div',attrs = {'class':'Epkrse '})

print(len(movies))
for m in movies:
    print(m.text)
```

```
스크롤 완료
132
킬링 카인드: 킬러의 수제자
노 서든 무브
푸른 호수
나인 데이즈 Nine Days
베네데타
스노우 몬스터
빌리 홀리데이
산타킬러스
커밍 홈 인 더 다크
퍼스트 카우
뉴 오더
메이드 인 이태리
왼팔의 복서 닉
스카이파이어
파우더 블루
8비트 크리스마스
워빌로우
```

사랑을 위하여
언힐러
더 매치: 1944
나쁜 녀석들 : 포에버 Bad Boys for Life
듄
007 노 타임 투 다이
크라이 마초
스피릿
미첼 가족과 기계 전쟁 Mitchells vs. the Machines, The
푸른 호수
H2: 어느 살인마의 가족 이야기
랑종
데쓰 프루프
더 로스트 레오나르도Lost Leonardo, The
팬보이즈
땅속에
슈퍼 히어로
커밍 홈 인 더 다크
에이펙스
닌자거북이 (2007)
플랜 A
퍼스트 카우
더 프레지던트
퍼피 구조대 더 무비
슈퍼 소닉
범블비 (자막판)
닌자 터틀
줄무늬 파자마를 입은 소년
보글보글 스폰지 밥
엘라 인챈티드
인생은 아름다워
휴고
스타더스트
닌자터틀: 어둠의 히어로
스파이 키드
샬롯의 거미줄
저지 걸
천국의 아이들
태양의 서커스 : 신비의 세계
베놈 2: 렛 데어 비 카니지 Venom: Let There Be Carnage
Free Guy
정글 크루즈
말리그넌트
리스펙트
이스케이프 룸 2: 노 웨이 아웃
듄
007 노 타임 투 다이
퍼피 구조대 더 무비
크라이 마초
코다

건파우더 밀크셰이크
졸라 Zola
시그널 X: 영혼의 구역
스피릿
미첼 가족과 기계 전쟁 Mitchells vs. the Machines, The
테이큰
H2: 어느 살인마의 가족 이야기
경고
새벽의 황당한 저주
해리포터 시리즈 완결 패키지
반지의 제왕: 3 영화 컬렉션 확장판 (자막판)
신비한 동물들과 그린델왈드의 범죄/ 신비한 동물 사전 영화 패키지 (자막판)
킹스맨 무비 컬렉션
트랜스포머 영화 5편 컬렉션
스타워즈 완전정복 패키지 (자막판)
MIB 맨 인 블랙 풀 패키지 (자막판)
인디아나 존스: 모험 컬렉션
드래곤 길들이기 3부작 (더빙판)
엑스맨 무비 컬렉션
도리 & 니모 패키지 (더빙판) (자막판)
스타워즈 디지털 콜렉션
50가지 그림자: 3 무비 콜렉션
메이즈러너 트릴로지 (자막판)
미션 임파서블 1-5 컬렉션 (자막판)
샤잠!/아쿠아맨 2 영화 컬렉션 (자막판)
슈퍼배드 & 미니언즈: 4편의 무비 컬렉션 (더빙판)
배트맨 완결 패키지 (자막판)
거미줄에 걸린 소녀 / 밀레니엄: 여자를 증오한 남자들 (자막판)
컨저링 유니버스 영화 컬렉션 (자막판)
퍼피 구조대 더 무비
슈퍼 소닉
범블비 (자막판)
닌자 터틀
줄무늬 파자마를 입은 소년
보글보글 스폰지 밥
엘라 인챈티드
인생은 아름다워
휴고
스타더스트
닌자터틀: 어둠의 히어로
스파이 키드
샬롯의 거미줄
저지 걸
천국의 아이들
태양의 서커스 : 신비의 세계
듄
맨 인 더 다크 2 Don't Breathe 2
더 수어사이드 스쿼드
분노의 질주: 더 얼티메이트

킬링 카인드: 킬러의 수제자
콰이어트 플레이스 2
캐시트럭
노바디
고질라 VS. 콩
건파우더 밀크셰이크
졸트
쥬라기 헌트
아이스로드
테이큰
스노우 몬스터
다이버전트 시리즈: 얼리전트
퍼펙트스틸
람보 2
에이펙스
USS 인디애나폴리스