# 블로그 게시글 수집

## #01. 패키지 참조

```
# 필요한 모듈 참조
import requests
from bs4 import BeautifulSoup
from pandas import DataFrame
```

## #02. 웹 페이지 코드 수집

```
# 수집할 컨텐츠가 있는 웹 페이지의 주소
url = "https://blog.hossam.kr/"

# 브라우저 버전정보
userAgent = "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like

# 접속객체 생성
session = requests.Session()

# 접속객체에 부가정보(header) 삽입하기
session.headers.update({
    "Referer": "",
    "User-Agent": userAgent
})

# 생성한 접속객체를 활용하여 API에 접속
r = session.get(url)

# 접속에 실패한 경우
if r.status_code != 200:
    # 에러코드와 에러메시지 출력
    msg = "[%d Error] %s 에러가 발생함" % (r.status_code, r.reason)
    # 에러를 강제로 생성시킴
    raise Exception(msg)

# 인코딩 형식 지정하여 beautifulsoup 객체를 생성
r.encoding = "utf-8"
#print(r.text)
soup = BeautifulSoup(r.text)
soup
```

## #03. 필요한 내용 추출하기

```
articleList = soup.select(".post")
articleList
```

```
# 수집한 결과가 저장될 빈 리스트
result = []

for article in articleList:
    # 제목 추출
    titleEl = article.select(".entry-title > a")
    #print(titleEl)
    title = titleEl[0].text.strip()
    #print(title)

    dateEl = article.select(".published")
    #print(dateEl)
    date = dateEl[0].attrs['datetime']
    #print(date)

    contentEl = article.select(".entry-content > p")
    #print(contentEl)
    content = contentEl[0].text.strip()
    #print(content)

    # 추출한 내용을 딕셔너리로 병합
    item = {"제목": title, "작성일": date, "요약글": content}

    # 딕셔너리를 미리 준비한 리스트에 원소로 추가
    result.append(item)

result
```

```
df = DataFrame(result)
df.to_excel("블로그_수집.xlsx")
df
```