

Python 심화 / 크롤링

4번째 세션

NEXT X LIKELION 김동현

과제 확인 - HW1

숫자를 입력 받고 홀수 구구단(3,5,7,9) 또는 짝수 구구단(2,4,6,8)을 출력하는 함수를 작성하고 실행하세요.

삼항연산자

```
def gugudan(n):  
    start = 2 if n % 2 == 0 else 3  
    for i in range(start,10,2):  
        for j in range(1,10):  
            print(f'{i} * {j} = {i*j}')  
        print('*****')  
  
gugudan(int(input()))
```

삼항연산자?

[true_value] if [condition] else **[false_value]**

과제 확인 - HW2

입력으로 들어오는 모든 수의 평균 값을 계산해 주는 함수를 작성해 보세요.
(단 입력으로 들어오는 수의 개수는 정해져 있지 않으며, 입력 값으로 -1이 들어오면 더 이상 값을 받지 않는다.)

```
def getAverage(array):  
    if(len(array)==0):  
        return 0  
    return sum(array)/len(array)  
  
core_list = []  
while(True):  
    core = int(input("양의 정수를 입력해주세요: "))  
    if(core == -1):  
        print(getAverage(core_list))  
        break  
    core_list.append(core)
```

세션 시작 전

session4 폴더 다운

1. 멋쟁이 사자처럼 폴더로 이동 (cd)
2. 공지방에서 session4 폴더 다운받기

| Session 4

1. Class 개념

2. Python 가상환경

3. Python Crawling



Class

Class는 뭘까?

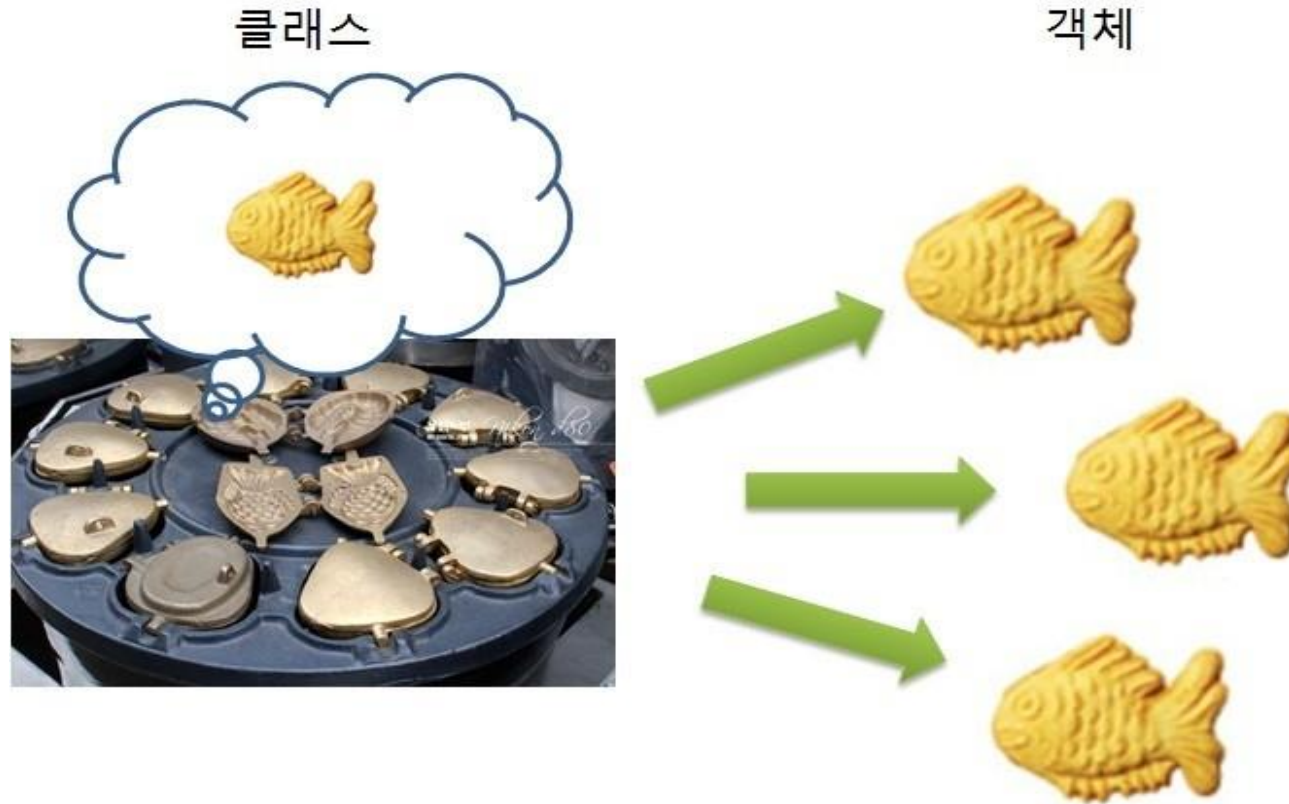
데이터와 기능을 함께 묶는 방법을 제공

Python, C++, Java 등에서 주로 사용 (객체지향 프로그래밍)

이후 Django의 DB Model 작성 시 주로 사용

Class

Class를 가장 잘 설명해주는 그림



- ① 붕어빵틀(클래스)에서 붕어빵(객체)을 찍어낼 수 있게 하나의 템플릿을 만드는 것!
- ② 객체들은 서로 영향을 주지 않으며, 고유한 성격을 가진다 (ex- 팔, 슈크림, 피자..)

Class

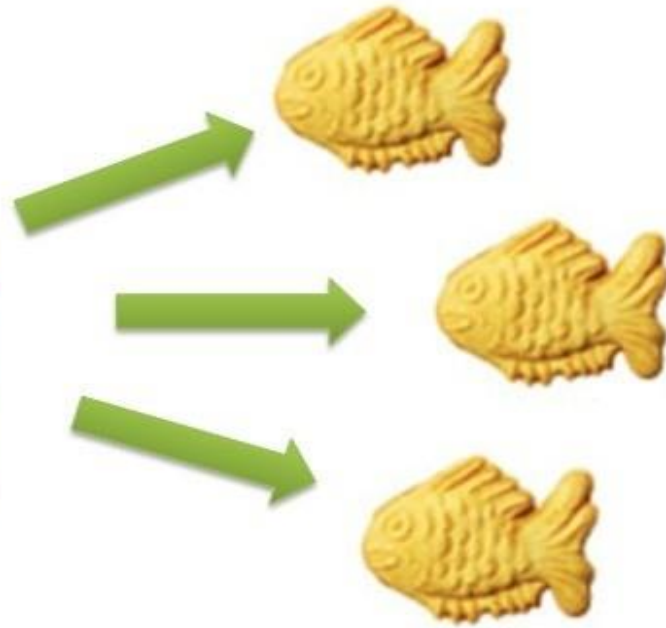
Class를 가장 잘 설명 해주는 그림

클래스



```
class FishBreadPan:  
    pass
```

객체



```
fishBread1 = FishBreadPan()  
fishBread2 = FishBreadPan()  
fishBread3 = FishBreadPan()
```


Class

Why Class? - 실습 자료 whyClass1.py를 열어주세요!

```
naver = {
    "name" : "Naver",
    "location" : "경기도 성남시 분당구",
    "salary" : 5000,
    "employee" : 3978,
}

coupang = {
    "name" : "Coupang",
    "location" : "서울특별시 송파구 송파대로",
    "salary" : 6000,
    "employee" : 49915,
}

def naver_hire():
    naver["salary"] += 500
    naver["employee"] += 1000

def coupang_hire():
    coupang["salary"] += 500
    coupang["employee"] += 1000
```

상황 : IT 기업들의 정보를 모으고, 업데이트 하는 과정

- ① 각 기업들은 딕셔너리의 key-value 형태로 저장
- ② 각각의 함수를 만들어 기업 정보 업데이트 수행
- ③ whyClass1.py 실행해보기

```
{'name': 'Naver', 'location': '경기도 성남시 분당구', 'salary': 5000, 'employee': 3978}
{'name': 'Coupang', 'location': '서울특별시 송파구 송파대로', 'salary': 6000, 'employee': 49915}
t-----채용 후-----
{'name': 'Naver', 'location': '경기도 성남시 분당구', 'salary': 5500, 'employee': 4978}
{'name': 'Coupang', 'location': '서울특별시 송파구 송파대로', 'salary': 7000, 'employee': 51915}
```

Class

Why Class? - 실습 자료 whyClass2.py를 열어주세요!

```
naver = {  
    "name" : "Naver",  
    "location" : "경기도 성남시 분당구",  
    "salary" : 5000,  
    "employee" : 3978,  
}  
  
coupang = {  
    "name" : "Coupang",  
    "location" : "서울특별시 송파구 송파대로",  
    "salary" : 6000,  
    "employee" : 49915,  
}  
  
def naver_hire():  
    naver["salary"] += 500  
    naver["employee"] += 1000  
  
def coupang_hire():  
    coupang["salary"] += 500  
    coupang["employee"] += 1000
```



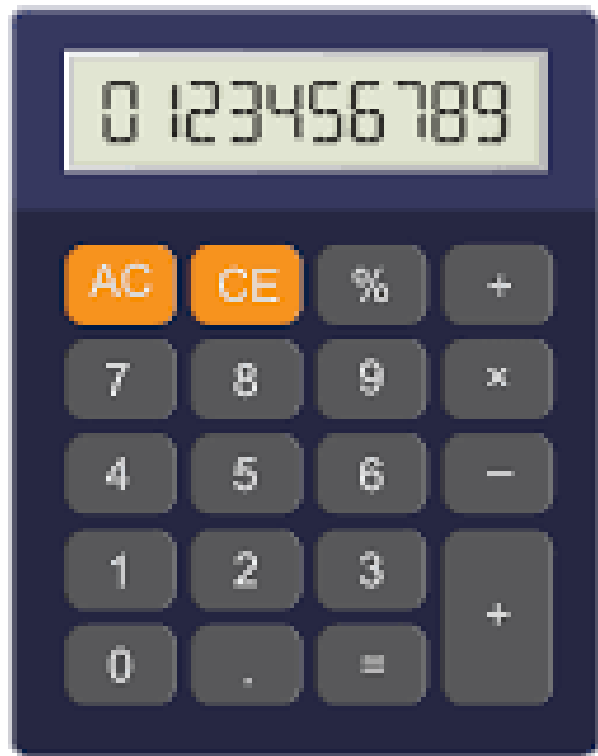
Class명은
항상 대문자로
작성!

```
# class 예제/  
class Company:  
    def __init__(self, name, location, salary, employee):  
        self.name = name  
        self.location = location  
        self.salary = salary  
        self.employee = 0  
  
    def hire(self):  
        self.salary += 1000  
        self.employee += 500
```

클래스를 사용하면 반복적인 코드를 줄일 수 있음
같은 일을 통해 쉽게 관리할 수 있음

Class

Class 문법 배우기



Class를 이용하여 계산기를 직접 만들어 봅시다

Class

클래스의 기본

로 시작하는
함수는 호출하
는 함수가 아님!

```
class Calculator:
    def __init__(self, name):
        self.name = name
        self.result = 0

calculator1 = Calculator("정상윤")
calculator2 = Calculator("허영봄")
print(calculator1.name)
print(calculator2.name)
```

__init__ : 생성자 함수(멤버 변수를 정의)

self : 인스턴스 본인을 의미, 본인 스스로의 값에 접근 가능

Class

만들어진 클래스를 사용하려면 인스턴스화가 필요!

Self 변수는
인스턴스 본인
을 의미!

```
class Calculator:
    def __init__(self, name):
        self.name = name
        self.result = 0

calculator1 = Calculator("정상윤")
calculator2 = Calculator("허영봄")
print(calculator1.value)
print(calculator2.value)
```

- ① calculator1은 **Calculator**의 인스턴스, 혹은 객체
- ② calculator의 첫번째 매개변수에는 **self**가 포함되어 있음

Class

더하기 기능을 만들어 봅시다!

```
class Calculator:
    def __init__(self, name):
        self.name = name
        self.result = 0
    def add(self, num1, num2):
        self.result = num1 + num2
        return self.result

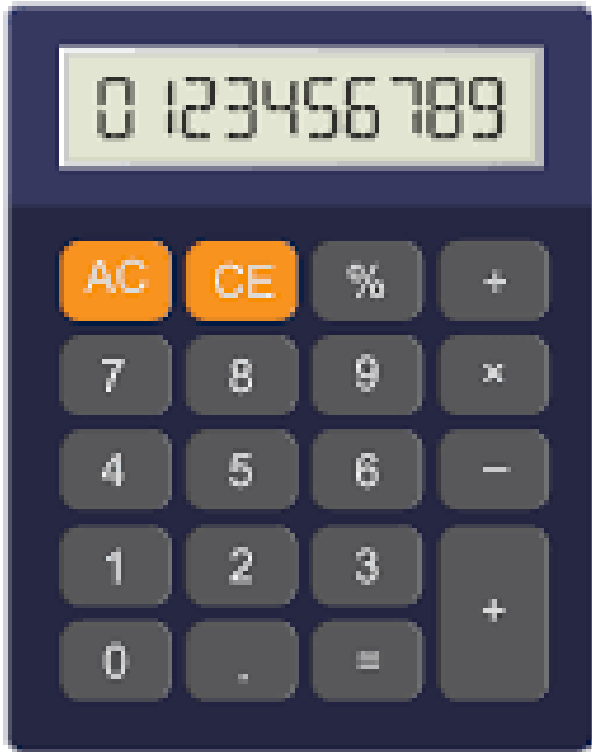
calculator1.add(3,4)
print(calculator1.result)
```

① **add** Method를 추가해봅시다!

② **result** 멤버 변수에 더한 결과를 저장합니다

Class 실습

즐거운 실습 시간

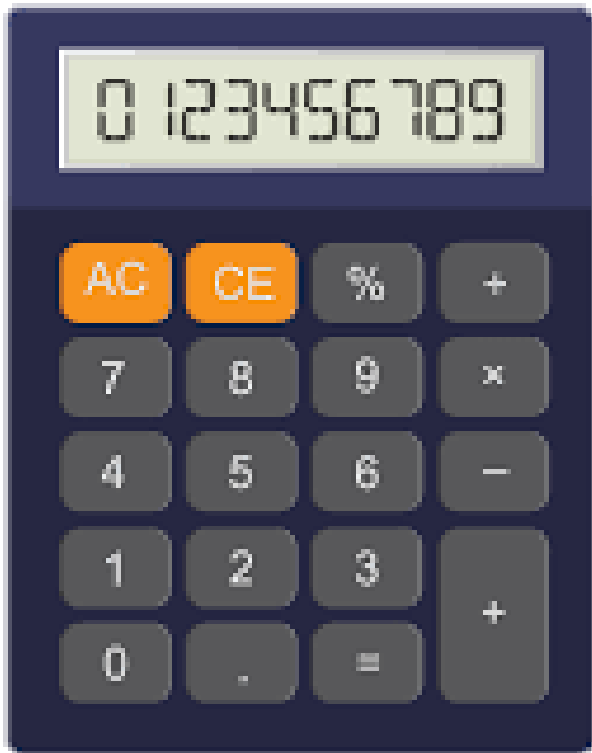


0. Git bash / PowerShell 실행
1. 멋쟁이 사자처럼 폴더로 이동 (cd)
2. cd session4
3. touch Calculator.py
4. code .
5. 뒷 페이지의 요구사항에 맞게 계산기 Class 만들기
6. 자유롭게 속성, 메서드 결과 출력해보기

Class 실습

즐거운 실습 시간

실습 요구사항



필요한 속성(멤버 변수)

1. 현재 계산기 값 (기본값 0)
2. 이름 (누구 계산기인지)
3. 나이

필요한 기능 정리(메소드)

1. 덧셈
2. 뺄셈
3. 곱셈
4. 나눗셈
(0으로 나눌 때 에러처리)

Class 실습

정답 확인

```
class Calculator:
    def __init__(self, name, age):
        self.name = name
        self.age = age
        self.result = 0
    def add(self, num):
        self.result += num
        return self.result
    def sub(self, num):
        self.result -= num
        return self.result
    def mul(self, num):
        self.result *= num
        return self.result
    def div(self, num):
        self.result /= num
        return self.result
```

```
# Step1 : 계산기 멤버 변수 정의
calculator1 = Calculator("정상윤", 20)
calculator2 = Calculator("허영봄", 20)
print(calculator1.name)
print(calculator2.name)

# Step2 : 계산기 기능 만들기
print(calculator1.result)
calculator1.add(3)
print(calculator1.result)
calculator1.sub(4)
print(calculator1.result)
calculator1.mul(4)
print(calculator1.result)
calculator1.div(2)
print(calculator1.result)
```

Class 실습

정답 확인

```
def div(self, num):  
    if(num == 0):  
        print("0으로 나눌 수 없습니다")  
        return None  
    self.result /= num  
    return self.result
```

0으로 나뉘질 경우 예외 처리 추가

공부 검색 키워드

시간 되실 때, 구글에 가볍게 검색해보세요!

1. Python Class 상속

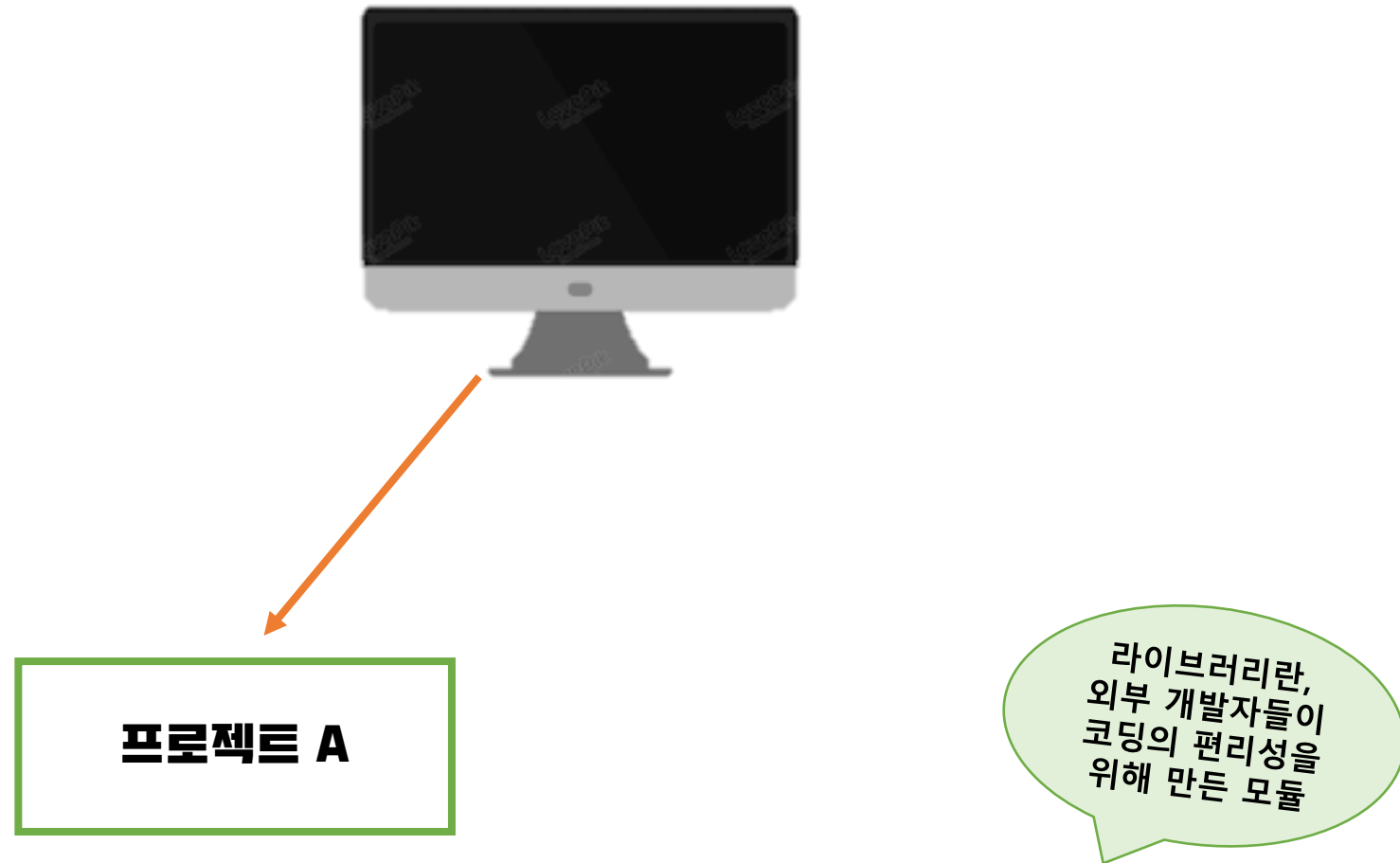
2. 객체지향 프로그래밍(OOP) 특징 및 장점

3. 절차지향 프로그래밍과 객체지향 프로그래밍의 차이

4. 메서드 오버라이딩(Overriding)과 오버로딩(Overloading)의 차이

파이썬 가상환경

가상환경이란?

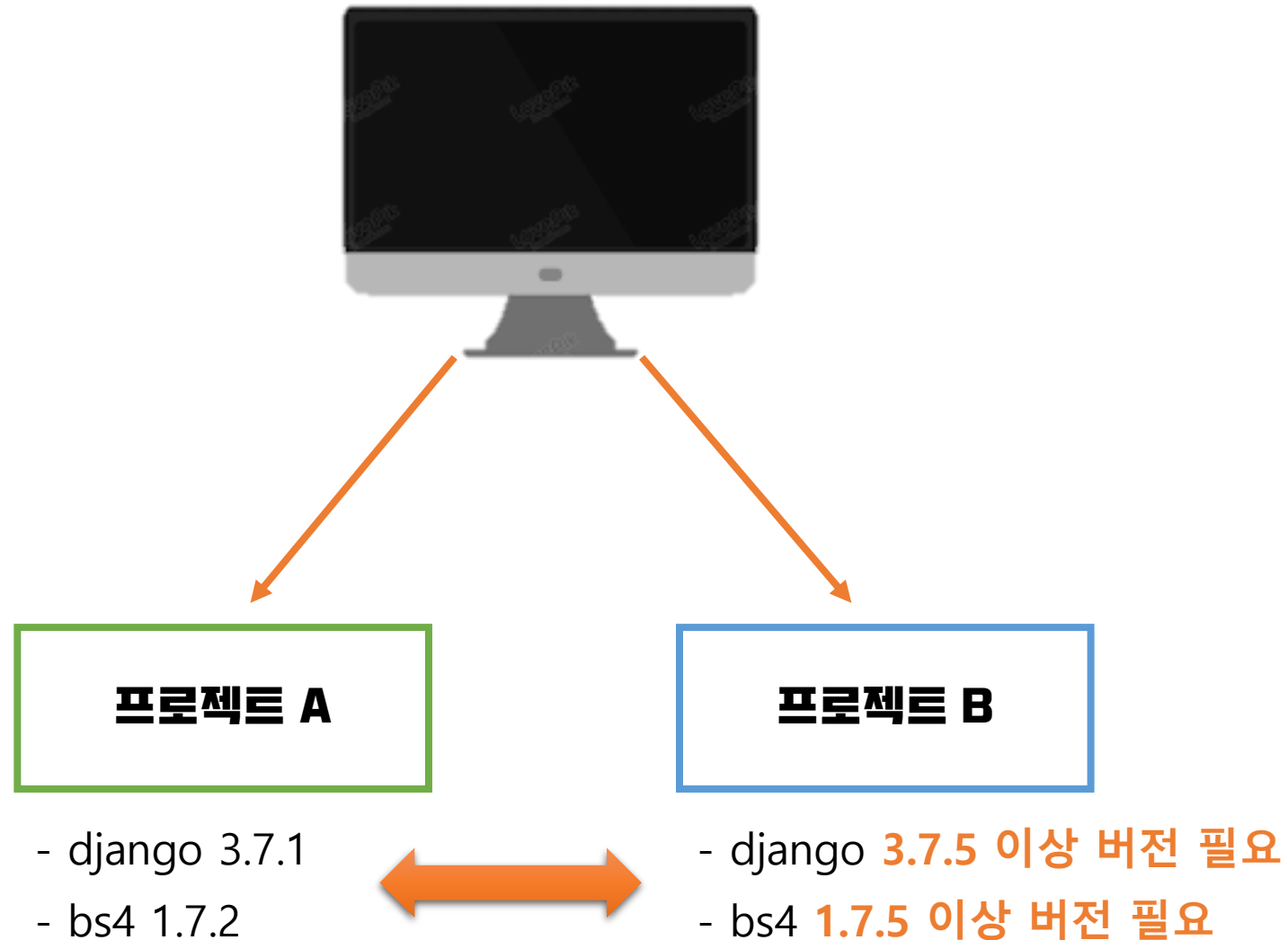


- django 3.7.1
- bs4 1.7.2

파이썬은 **pip**을 통해 필요한 라이브러리 설치 (컴퓨터 전역)

파이썬 가상환경

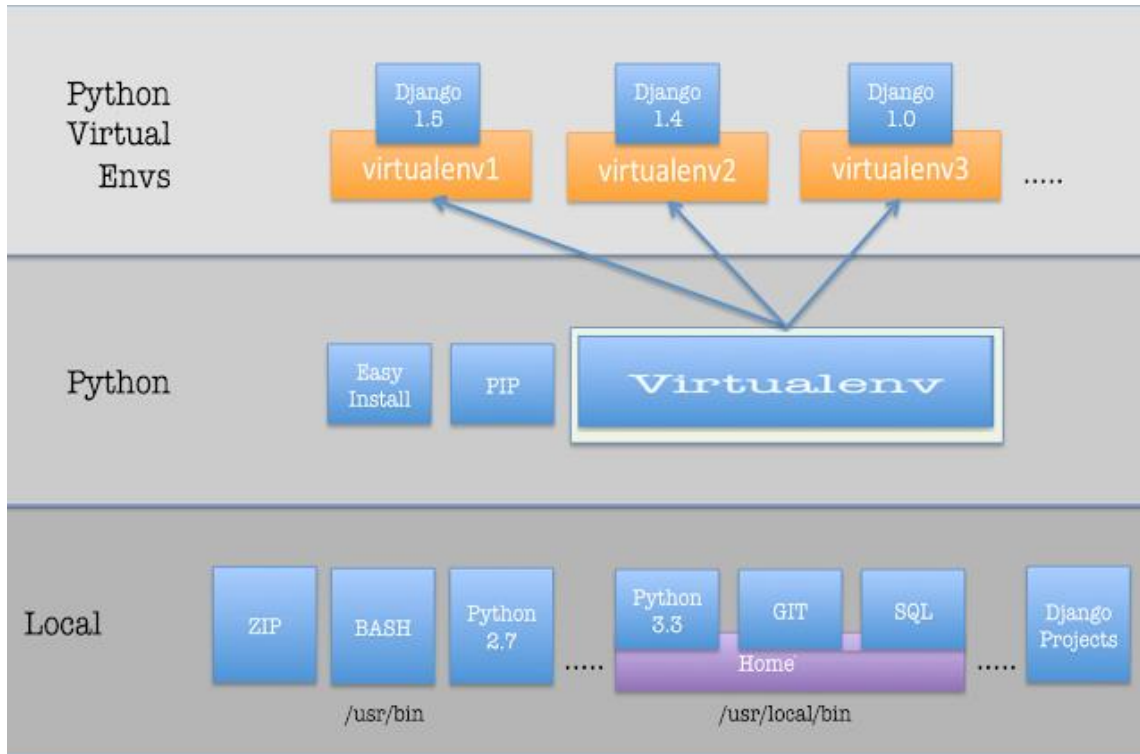
가상환경이란?



버전이 일치하지 않아 호환이 되지 않음

파이썬 가상환경

가상환경은 독립된 파이썬 개발 프로젝트 환경을 만드는 것!



가상 환경

독립된 공간 -> 프로젝트별로 개발환경 구축

통일된 라이브러리 버전을 사용하므로 협업시에 버전이 충돌 날 일이 없음

가상환경은 독립적이어서 서로 다른 가상환경에 설치된 모듈들의 영향을 받지 않음

파이썬 가상환경

파이썬 가상환경 종류

venv : Python 3.3 버전 이후 부터 기본모듈에 포함됨

virtualenv : Python 2 버전부터 사용해오던 가상환경 라이브러리, Python 3에서도 사용가능

conda : Anaconda Python을 설치했을 시 사용할 수있는 모듈

pipenv : 입문자가 사용하기에 좋은 가상환경

pipenv

패키지 관리를 자동으로 해준다

패키지 이름 **오**타 **유**의

처음에는 pipenv로 시작하고, 나중에 virtualenv나 venv, pyenv로 바꾸길 추천!

파이썬 가상환경

pip 설치 / 버전 확인

pip 설치 (컴퓨터 전역 설치)

Windows

- ① `curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py`
- ② `python get-pip.py`

Mac

- ① `sudo apt-get install python3-pip`

pip 버전 확인

`pip --version`

파이썬 가상환경

Pipenv 설치 / 버전 확인

pipenv 설치 (컴퓨터 전역 설치)

`pip install pipenv` (Windows)

`pip3 install pipenv` (Mac)

pipenv 버전 확인

`pipenv --version`

pipenv 명령어 정리

pipenv shell – 가상환경 생성 및 시작

exit – 가상환경 끄기

pipenv install 패키지명 – 해당 패키지 설치

pipenv uninstall 패키지명 – 해당 패키지 제거

파이썬 가상환경

pipenv 실습

0. Git bash / PowerShell 실행

1. 멧쟁이 사자처럼 session4 폴더로 이동 (cd)

2. pipenv shell (꼭 해당 프로젝트 최상단 위치에서 생성할 것!)

3. pipenv install request

4. pipenv install bs4

| 쉬는 시간

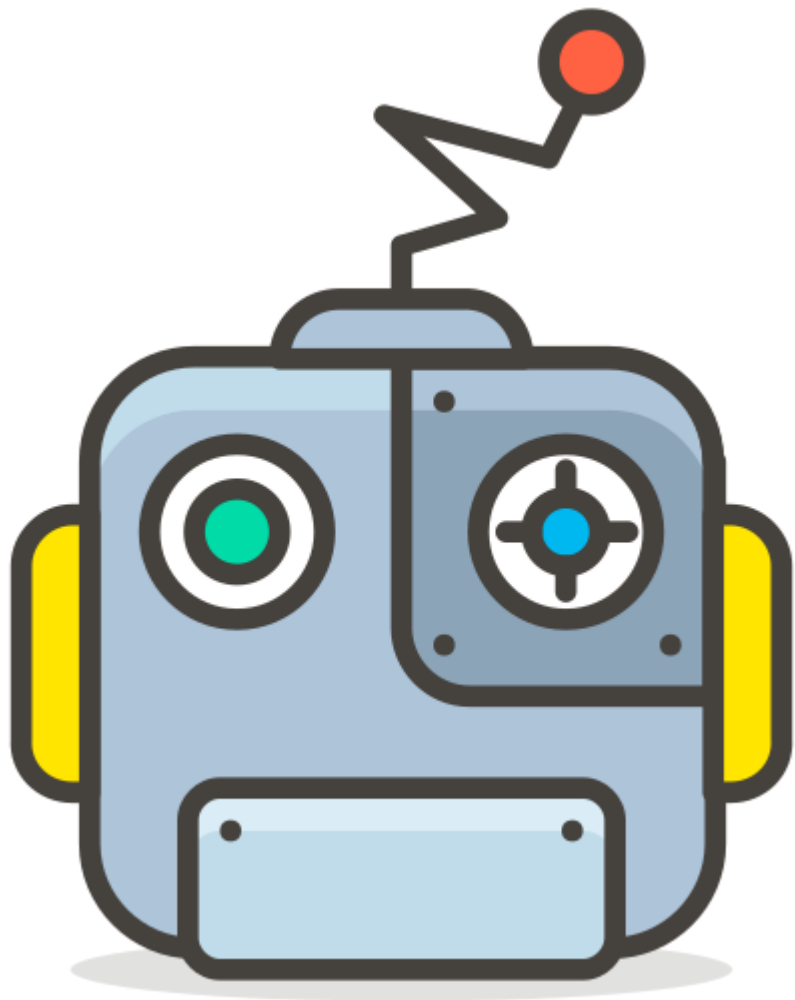


| 크롤링

인터넷상에 존재하는 웹 문서들을 추적하여 **필요한 정보를 수집**하는 기법

HTML 페이지를 가져와서, HTML/CSS등을 파싱하고, 필요한 데이터만 추출하는 기법

무분별하게 해당 웹사이트에서 데이터를 가져와서 **상업적으로 이용하면 안됨**



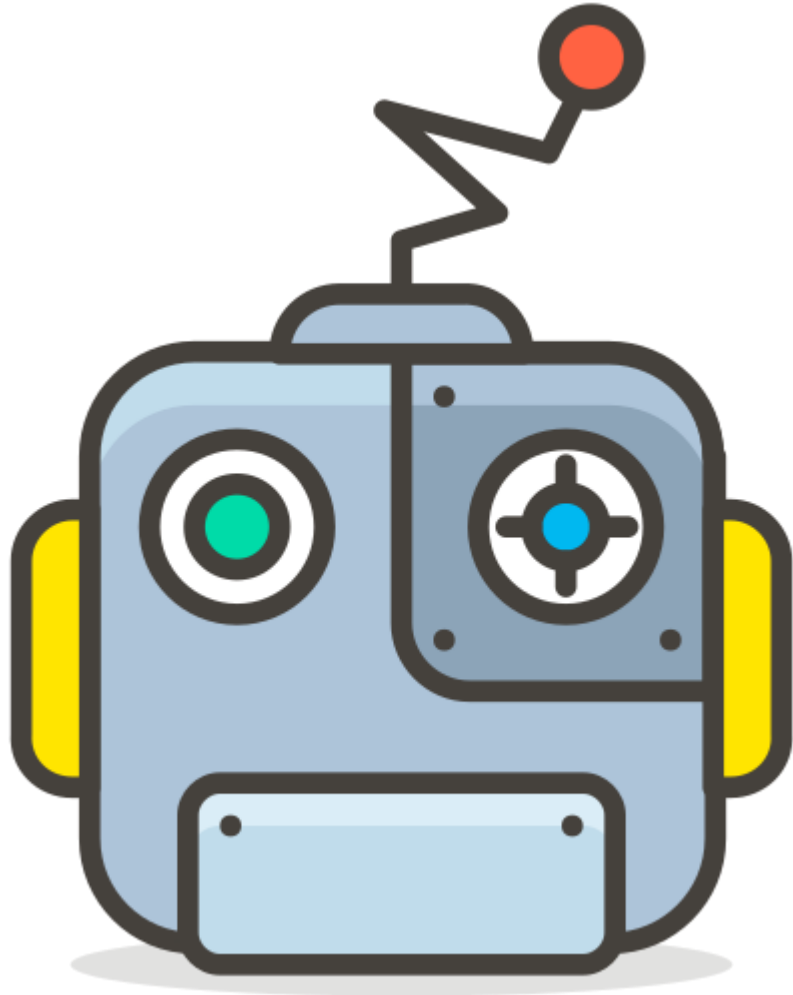
크롤링으로 수집한 데이터로 이익을 취하면 문제가 될 수 있음

각 사이트에서는 크롤러가 요청을 해도 되거나,
해서는 안되는 사항들을 robots.txt를 통하여 확인할 수 있음!

<https://www.naver.com/robots.txt>

크롤링

<https://www.naver.com/robots.txt>



User-agent: *

Disallow: /

Allow : /\$

네이버 메인페이지 이외에서는 크롤링을 금지
/로 끝나는 URL만 허용 (www.naver.com/)

beautifulsoup4 - HTML과 XML 문서를 파싱하기 위한 파이썬 패키지

(<https://www.crummy.com/software/BeautifulSoup/bs4/doc>)

find 함수

```
soup.find('p')  
soup.find('div',{'class':'클래스 네임명'})  
soup.find('img',{'id':'아이디 네임명'})  
  
soup.find_all('div')
```

beautifulsoup4 - HTML과 XML 문서를 파싱하기 위한 파이썬 패키지

(<https://www.crummy.com/software/BeautifulSoup/bs4/doc>)

select 함수

```
soup.select('p')  
soup.select('.클래스 네임명')  
soup.select('상위태그명 > 하위태그명 > 하위태그명')  
soup.select('상위태그명.클래스명 > 하위태그명.클래스명')  
soup.select('#아이디명')  
soup.select('#아이디명 > 태그명.클래스명')  
soup.select('태그명[속성1=값1]')
```

크롤링 실습 전

스마트 스토어에서 크롤링을 직접 해봅시다!

〈크롤링 전체적인 Flow〉

- ① 웹 사이트 url에 get요청
- ② find나 select 함수를 이용해서 원하는 HTML Element 가져오기
- ③ 원하는 데이터 결과 값 형태로 가공하기

크롤링 실습 전

스마트 스토어에서 크롤링을 직접 해봅시다!

스마트 스토어에서 직접 크롤링 실습을 시작해볼까요?

(<https://search.shopping.naver.com/search/all?pagingIndex=2&pagingSize=80&query=노트북>)

Query String

<https://search.shopping.naver.com/search/all.nhn?변수1=값1&변수2=값2>

- get 요청을 보낼 때, 주로 query에 변수를 담아 정보를 전송함
- Uri 주소 뒤에 ?로 시작
- 변수와 값의 쌍으로 구성 (각 쌍은 &로 구분)

Query String

<https://search.shopping.naver.com/search/all?pagingIndex=2&pagingSize=80&query=노트북>

- 2번째 페이지
- 한 페이지에는 80개의 검색 결과
- 검색어는 노트북

크롤링 실습 전

note_book.py 파일을 열어주세요!

session4 폴더 안에 notebook.py를 열어주세요!

크롤링 실습

해당 url 연결

해당 주소에 요청 보내기

```
import requests
from bs4 import BeautifulSoup

# 우리가 정보를 얻고 싶어 하는 URL
NOTEBOOK_URL = f'https://search.shopping.naver.com/search/all?pagingIndex=1&pagingSize=80&query=노트북'
# get 요청을 통해 해당 페이지 정보를 저장
notebook_html = requests.get(NOTEBOOK_URL)
# bs4 라이브러리를 통해 불러온 html을 우리가 원하는 형태로 파싱
notebook_soup = BeautifulSoup(notebook_html.text, "html.parser")

print(notebook_soup)
```


크롤링 실습

상품 리스트 출력

상품 리스트 출력

```
import requests
from bs4 import BeautifulSoup

# 우리가 정보를 얻고 싶어 하는 URL
NOTEBOOK_URL = f'https://search.shopping.naver.com/search/all?pagingIndex=1&pagingSize=80&query=노트북'

# get 요청을 통해 해당 페이지 정보를 저장
notebook_html = requests.get(NOTEBOOK_URL)
# bs4 라이브러리를 통해 불러온 html을 우리가 원하는 형태로 파싱
notebook_soup = BeautifulSoup(notebook_html.text, "html.parser")

notebook_list_box = notebook_soup.find("ul", {"class": "list_basis"})
notebook_list = notebook_list_box.find_all("li", {"class": "basicList_item__2XT81"})
print(notebook_list_box)
```

결과 -> 길이가 5인 배열

크롤링 실습

첫 번째 상품의 이름 찾기

첫 번째 상품의 이름 찾기

ul > li > .basicList_inner__eY_mq > .basicList_title__3P9Q7 > a



a.basicList_link_1MaTN 107.38 × 14

디클 클릭북 D141+

최저 279,000원 판매처 7

디지털/가전 > 노트북

화면크기 : 14인치(35~36cm) | 무게 : 1.41kg | CPU : Z8350F | 칩셋 제조
코어종류 : 쿼드코어 | 코드명 : 체리트레일 | CPU속도 : 1.44GHz | 램 : 2GB

리뷰 ★★★★★ 4,209 · 등록일 2016.12. · ♥ 찜하기 748 · 📄 정보 수

```
><div class="basicList_img_area_a3NRA">...</div>
▼<div class="basicList_info_area_17Xyo">
  <a href="https://adcr.naver.com/adcr?x=mlwul/ykzNG0JFsh2Ev183///w=kfuxji/...KL/LoIaV1bBnFjGqrb11ICoAFH6stwi9BSadJ0FK3K9zwGwYyhTAqeJwpHERXBhHrCpIF1Nc=" target="_blank" class="basicList_brand_message__2-W1N" rel="noopener" data-nclick="N=a:lst*B.brandmsg,i:10776767442,r:5">학습용 문서작성
  ok</a>
  ▼<div class="basicList_title__3P9Q7">
    ...
    <a href="https://adcr.naver.com/adcr?x=mlwul/ykzNG0JFsh2Ev183///w=kfuxji/...KL/LoIaV1bBnFjGqrb11ICoAFH6stwi9BSadJ0FK3K9zwGwYyhTAqeJwpHERXBhHrCpIF1Nc=" target="_blank" class="basicList_link__1MaTN" rel="noopener" data-nclick="N=a:lst*B.title,i:10776767442,r:5" title="디클 클릭북 D141+">디클 클릭북 D141+</a> == $0
  </div>
  ><div class="basicList_price_area_1UXXR">...</div>
  ><div class="basicList_depth_20Tie" data-nclick="N=a:lst*B.category,i:10776767442,r:5">...</div>
```

크롤링 실습

첫 번째 상품의 이름 찾기

첫 번째 상품의 이름 찾기

```
import requests
from bs4 import BeautifulSoup

NOTEBOOK_URL = f'https://search.shopping.naver.com/search/all?pagingIndex=1&pagingSize=80&query=노트북'
notebook_html = requests.get(NOTEBOOK_URL)
notebook_soup = BeautifulSoup(notebook_html.text, "html.parser")

notebook_list_box = notebook_soup.find("ul", {"class": "list_basis"})
notebook_list = notebook_list_box.find_all("li", {"class": "basicList_item__2XT81"})
title = notebook_list[0].find("div", {"class": "basicList_title__3P9Q7"}).find("a").string

print(title)
```

크롤링 실습

첫 번째 상품의 가격 찾기

첫 번째 상품의 가격 찾기

ul > li > .basicList_inner__eY_mq > basicList_price_area__lUXXR > . price_num__2WUXn

전체 14,639,677

가격 95%

네이버 랭킹순 · 낮은 가격순 · 높은 가격순

ACCESSIBILITY

Contrast Aa 3.66

Name

Role generic

Keyboard-focusable

LG전자 그래픽스

①광고 최저 1,373,190원 판매처 262

디지털가전 > 노트북

화면크기 : 14인치(35~36cm) | 무게 : 0.999kg | 종류 : 코어i5 11세대 | 운영체제 : 미포함(FreeDos)

CPU : 코어i5-1135G7 | 칩셋 제조사 : 인텔 | 코어종류 : 쿼드코어 | 코드명 : 타이거레이크

등록일 2020.12. · ♥ 찜하기 27 · 📄 정보 수정요청

브랜드 카탈로그 LG전자

G마켓 ↓ 1,373,190

옥션 1,373,200

쿠팡 1,386,690

G9 1,388,120

스마트프라... Pay 1,479,000

삼성전자 갤럭시북 플렉스2 NT950QDA-X71AZ

①광고 최저 2,449,000원 판매처 161

디지털가전 > 노트북

화면크기 : 15인치(37~39cm) | 무게 : 1.57kg | 종류 : 코어i7 11세대 | 운영체제 : 윈도우10 홈

브랜드 카탈로그 삼성전자

11번가 ↓ 2,449,000

삼성mall ... Pay 2,499,000

삼성공식파... Pay 2,499,000

```
<div class="style_content_wrap__1PzEo">
  <div class="style_content__2T20F">
    <div class="seller_filter_area">...</div>
    <ul class="list_basis">
      <div>
        <div>
          <li class="basicList_item__2XT81 ad">
            <div class="basicList_inner__eY_mq">
              <div class="basicList_img_area__a3NRA">...</div>
              <div class="basicList_info_area__17Xyo">
                <div class="basicList_title__3P9Q7">
                  <a href="https://adcr.naver.com/adcr?x=KhyhnEr1+cAK6cR+oHSAd/////w=kk/bqHrd...QwLm1B9CwNaQ1Vfq2+9x/mU2jJU708V6DPfO2wqDr4Qd/+fV2k3X3CtKrREs5E4g6guMAQ==" target="_blank" class="basicList_link__1MaTN" rel="noopener" data-nclick="N=a:1st*B.title,i:25255939522,r:1" title="LG전자 그래픽스 14 14ZD90P-GX5BK">LG전자 그래픽스 14 14ZD90P-GX5BK</a>
                </div>
                <div class="basicList_price_area__lUXXR">
                  <button type="button" class="ad_ad_stk_12U34">광고</button>
                  <strong class="basicList_price__2r23_">
                    <span>
                      <span class="price_low__2vp2A">최저</span>
                      <span class="price_num__2WUXn">1,373,190원</span> == $
                    </span>
                  <a href="https://adcr.naver.com/adcr?x=KhyhnEr1+cAK6cR+oHSAd/////w=kk/bqHrd...QwLm1B9CwNaQ1Vfq2+9x/mU2jJU708V6DPfO2wqDr4Qd/+fV2k3X3CtKrREs5E4g6guMAQ==" target="_blank" class="basicList_link__1MaTN" rel="noopener" data-nclick="N=a:1st*B.title,i:25255939522,r:1" title="LG전자 그래픽스 14 14ZD90P-GX5BK">LG전자 그래픽스 14 14ZD90P-GX5BK</a>
                </div>
              </div>
            </div>
          </li>
        </div>
      </ul>
    </div>
  </div>
</div>
```

크롤링 실습

첫 번째 상품의 이름 찾기

첫 번째 상품의 가격 찾기

```
import requests
from bs4 import BeautifulSoup
notebook_html = requests.get('https://search.shopping.naver.com/search/all?pagingIndex=2&pagingSize=80&query=노트북')
notebook_soup = BeautifulSoup(notebook_html.text, "html.parser")

notebook_list_box = notebook_soup.find("ul", {"class" : "list_basis"})
notebook_list = notebook_list_box.find_all('li', {"class" : "basicList_item__2XT81"})
price = notebook_list[0].find("div", {"class": "basicList_price_area__1UXXR"}).find("span", {"class": "price_num__2WUXn"}).text

print(price)
```

크롤링 실습

텍스트를 추출하는 방법

텍스트를 추출하는 방법

```
<td>some text</td>  
<td></td>  
<td><p>more text</p></td>  
<td>even <p>more text</p></td>
```

① string

```
some text  
None  
more text  
None
```

- 태그 하위에 문자열을 객체화
- 문자열이 없으면 None

② text

```
some text  
  
more text  
even more text
```

- 하위 자식태그의 텍스트까지 문자열로 반환

③ 그 외

- strip : 공백 없애기
- split : 문자열을 배열로 나누기
- replace : 특정 문자열을 교체

크롤링 실습

데이터를 저장해 봅시다

데이터를 저장해 봅시다

```
import requests
from bs4 import BeautifulSoup

NOTEBOOK_URL = f'https://search.shopping.naver.com/search/all?pagingIndex=1&pagingSize=80&query=노트북'
notebook_html = requests.get(NOTEBOOK_URL)
notebook_soup = BeautifulSoup(notebook_html.text, "html.parser")

notebook_list_box = notebook_soup.find("ul", {"class": "list_basis"})
notebook_list = notebook_list_box.find_all("li", {"class": "basicList_item__2XT81"})
title = notebook_list[0].find("div", {"class": "basicList_title__3P9Q7"}).find("a").string
price = notebook_list[0].find("div", {"class": "basicList_price_area__1UXXR"}).find("span", {"class": "price_num__2WUXn"}).text

result = {
    'title': title,
    'price': price
}
print(result)
```

크롤링 실습

직접 한번 해볼까요?

실습 - 노트북의 상세 정보를 배열로 담아 result에 추가하기



LG전자 그램14 14ZD90P-GX5BK

📢 **최저 1,373,190원** 판매처 262

디지털/가전 > 노트북

화면크기 : 14인치(35~36cm) | 무게 : 0.999kg | 종류 : 코어i5 11세대 | 운영체제 : 미포함(FreeDos)
CPU : 코어i5-1135G7 | 칩셋 제조사 : 인텔 | 코어종류 : 쿼드코어 | 코드명 : 타이거레이크

등록일 2020.12. · ❤️ 찜하기 27 · 📄 정보 수정요청

브랜드 카탈로그
LG전자

G마켓	↓ 1,373,190
옥션	1,373,200
쿠팡	1,386,690
G9	1,388,120
스마트프라...	1,479,000

```
{'title': 'LG전자 그램14 14ZD90P-GX5BK', 'price': '1,373,190원', 'detail': ['화면크기 : 14인치(35~36cm)', '무게 : 0.999kg', '종류 : 코어i5 11세대', '운영체제 : 미포함(FreeDos)', 'CPU : 코어i5-1135G7', '칩셋 제조사 : 인텔', '코어종류 : 쿼드코어', '코드명 : 타이거레이크', 'CPU속도 : 2.4GHz', '터보부스트 : 4.2GHz', '램 : 8GB', '메모리 타입 : LPDDR4x', '인텔 GPU : Iris Xe Graphics', '그래픽 메모리 : 시스템 메모리 공유', '해상도 : 1920x1200(WUXGA)', '패널 : IPS패널(광시야각)', '화면비율 : 16대10', '베젤 : 슬림베젤', '특징 : 가벼운노트북', '무선랜 : 무선랜 802.11 ax(Wi-Fi6)', '블루투스 : 지원', '유선랜 : 유선랜', '영상출력 : HDMI', '썬더볼트4', '단자 : USB Type C', 'USB3.1', '카드 슬롯 : UFS', '부가기능 : 키보드라이트', '웹캠', 'USB-PD', 'Evo 플랫폼 인증', 'SSD : 256GB', 'SSD 인터페이스 : M.2', '배터리 타입 : 리튬이온', '배터리용량 : 72Wh', '웹카메라 : 전면', '품목 : 노트북', '두께 : 1.68cm', '조명 : LED백라이트', 'DCI-P3 : 99%', '보안기능 : 지문 인식']}]
```

Hint) find_all과 for문을 이용.

크롤링 실습

정답 확인

노트북의 상세 정보 추가

```
import requests
from bs4 import BeautifulSoup

NOTEBOOK_URL = f'https://search.shopping.naver.com/search/all?pagingIndex=1&pagingSize=80&query=노트북'
notebook_html = requests.get(NOTEBOOK_URL)
notebook_soup = BeautifulSoup(notebook_html.text, "html.parser")

notebook_list_box = notebook_soup.find("ul", {"class": "list_basis"})
notebook_list = notebook_list_box.find_all("li", {"class": "basicList_item__2XT81"})
title = notebook_list[0].find("div", {"class": "basicList_title__3P9Q7"}).find("a").string
price = notebook_list[0].find("div", {"class": "basicList_price_area__1UXXR"}).find("span", {"class": "price_num__2WUXn"}).text
detail_lists = notebook_list[0].find("div", {"class": "basicList_desc__2tko"}).find_all("a", {"class": "basicList_detail__27Krk"})
detail = []
for detail_list in detail_lists:
    detail.append(detail_list.text)
result = {
    'title': title,
    'price': price,
    'detail': detail
}
print(result)
```

크롤링 실습

데이터를 저장해 봅시다

모든 데이터를 저장해 봅시다

```
import requests
from bs4 import BeautifulSoup

NOTEBOOK_URL = f'https://search.shopping.naver.com/search/all?pagingIndex=1&pagingSize=80&query=노트북'
notebook_html = requests.get(NOTEBOOK_URL)
notebook_soup = BeautifulSoup(notebook_html.text, "html.parser")
notebook_list_box = notebook_soup.find("ul", {"class": "list_basis"})
notebook_list = notebook_list_box.find_all("li", {"class": "basicList_item__2XT81"})
result = []

for notebook in notebook_list:
    title = notebook.find("div", {"class": "basicList_title__3P9Q7"}).find("a").string
    price = notebook.find("div", {"class": "basicList_price_area__1UXXR"}).find("span", {"class": "price_num__2WUXn"}).text
    detail_lists = notebook.find("div", {"class": "basicList_desc__2-tko"}).find_all("a", {"class": "basicList_detail__27KrK"})
    detail = []
    for detail_list in detail_lists:
        detail.append(detail_list.text)
    notebook_info = {
        'title': title,
        'price': price,
        'detail': detail
    }
    result.append(notebook_info)

print(result)
```

코드가 길어지니, 기능별로 파일을 나누어서 저장해봅시다

① `main.py` – url 연결하는 파일

② `note_book.py` – 주어진 데이터를 추출하고 가공하는 파일

① main.py – url 연결하는 파일

from 파일명 import 함수명

```
import requests
from bs4 import BeautifulSoup
from note_book import extract_info

notebook_html = requests.get('https://search.shopping.naver.com/search/all?pagingIndex=2&pagingSize=80&query=노트북')
notebook_soup = BeautifulSoup(notebook_html.text, "html.parser")

notebook_list_box = notebook_soup.find("ul", {"class" : "list_basis"})
notebook_list = notebook_list_box.find_all('li', {"class" : "basicList_item__2XT81"})

print(extract_info(notebook_list))
```

② `note_book.py` – 주어진 데이터를 추출하고 가공하는 파일

```
def extract_info(notebook_list):
    result = []
    for notebook in notebook_list:
        title = notebook.find("div", {"class": "basicList_title__3P9Q7"}).find("a").string
        price = notebook.find("div", {"class": "basicList_price_area__1UXXR"}).find("span", {"class": "price_num__2WUXn"}).text
        detail_lists = notebook.find("div", {"class": "basicList_desc__2-
tko"}).find_all("a", {"class": "basicList_detail__27Krk"})
        detail = []
        for detail_list in detail_lists:
            detail.append(detail_list.text)
        notebook_info = {
            'title': title,
            'price': price,
            'detail': detail
        }
        result.append(notebook_info)
    return result
```

실습 시간

- ① 1~10페이지 전체 상품 목록 저장 후 print 하기
- ② main.py만 수정하기

Hint) for 문 사용, f-string을 이용해 page 변수 넣기

실습 정답

```
import requests
from bs4 import BeautifulSoup
from note_book import extract_info

final_result = []
for page in range(1,11):
    NOTEBOOK_URL = f'https://search.shopping.naver.com/search/all?pagingIndex={page}&pagingSize=80&query=노트북'
    notebook_html = requests.get(NOTEBOOK_URL)
    notebook_soup = BeautifulSoup(notebook_html.text, "html.parser")

    notebook_list_box = notebook_soup.find("ul", {"class" : "list_basis"})
    notebook_list = notebook_list_box.find_all('li', {"class" : "basicList_item__2XT81"})
    final_result += extract_info(notebook_list)

print(final_result)
```

크롤링 실습

여기까지 잘 따라오셨나요?



지금까지 저장한 데이터를 CSV 파일에 저장해봅시다!

CSV (comma-separated values)

- ① 몇 가지 필드를 쉼표(,)로 구분한 텍스트 데이터 및 텍스트 파일
- ② 엑셀 프로그램으로 열기 가능!
- ③ 이후, DB에 직접 넣을 수 있는 파일 형식

크롤링 실습

CSV 파일 저장

```
import requests
from bs4 import BeautifulSoup
from note_book import extract_info
import csv

file = open('note_book.csv', mode='w', newline='')
writer = csv.writer(file)
writer.writerow(["title", "price", "detail"])

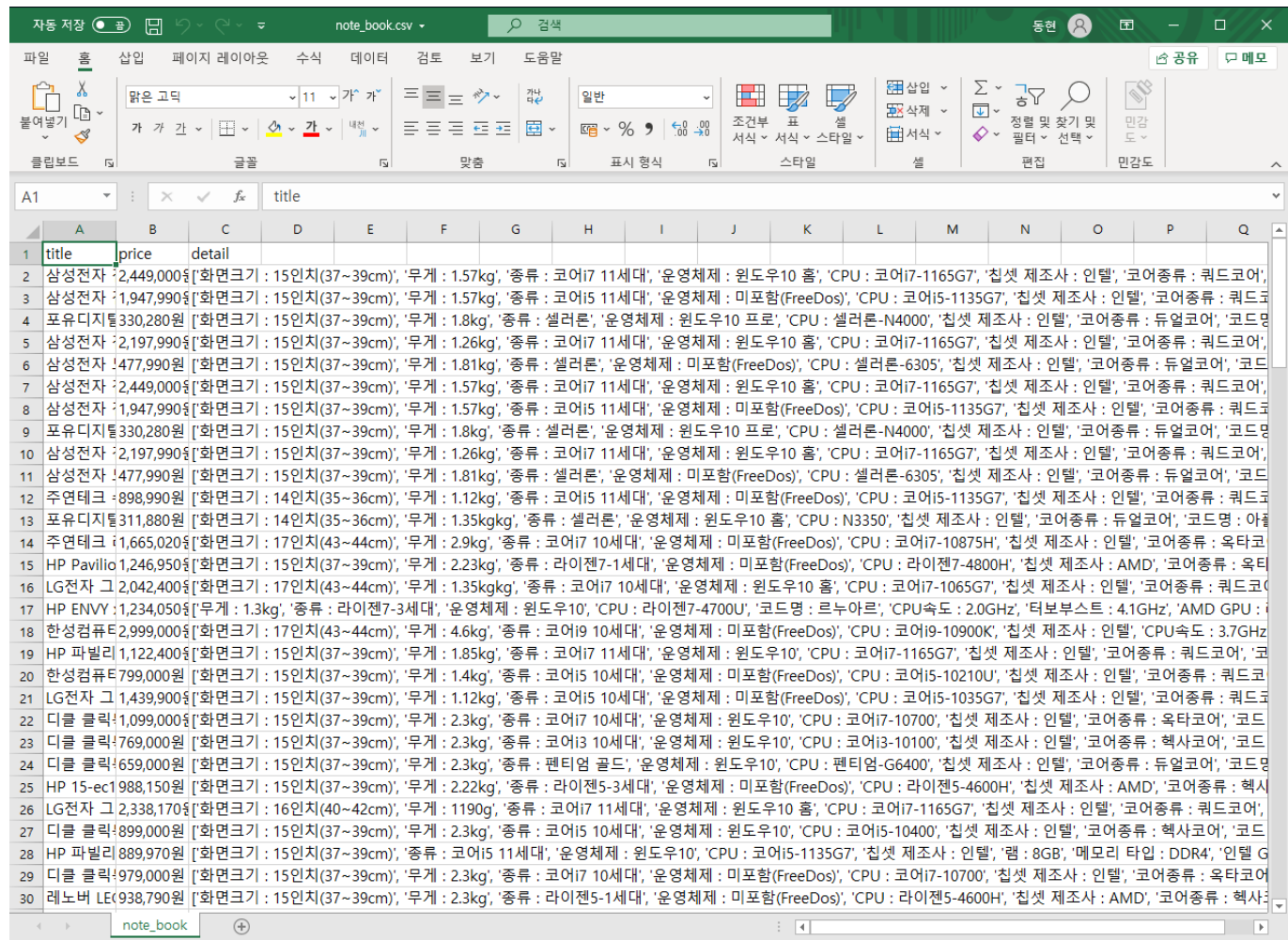
final_result = []
for page in range(30):
    NOTEBOOK_URL = f'https://search.shopping.naver.com/search/all?pagingIndex={page}&pagingSize=80&query=노트북'
    notebook_html = requests.get(NOTEBOOK_URL)
    notebook_soup = BeautifulSoup(notebook_html.text, "html.parser")

    notebook_list_box = notebook_soup.find("ul", {"class" : "list_basis"})
    notebook_list = notebook_list_box.find_all('li', {"class" : "basicList_item__2XT81"})
    final_result += extract_info(notebook_list)

for result in final_result:
    row = []
    row.append(result['title'])
    row.append(result['price'])
    row.append(result['detail'])
    writer.writerow(row)
print(final_result)
```

크롤링 실습

CSV 파일 저장



The screenshot shows a Microsoft Excel spreadsheet titled 'note_book.csv'. The spreadsheet contains a list of products with their prices and detailed specifications. The columns are labeled 'title', 'price', and 'detail'. The data is organized into rows, with the first row being the header. The 'detail' column contains a large block of text for each product, listing various specifications such as screen size, weight, processor, operating system, and storage.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	title	price	detail														
2	삼성전자	2,449,000원	[화면크기 : 15인치(37~39cm), '무게 : 1.57kg', '종류 : 코어i7 11세대', '운영체제 : 윈도우10 홈', 'CPU : 코어i7-1165G7', '칩셋 제조사 : 인텔', '코어종류 : 쿼드코어',														
3	삼성전자	1,947,990원	[화면크기 : 15인치(37~39cm), '무게 : 1.57kg', '종류 : 코어i5 11세대', '운영체제 : 미포함(FreeDos)', 'CPU : 코어i5-1135G7', '칩셋 제조사 : 인텔', '코어종류 : 쿼드코														
4	포유디지털	330,280원	[화면크기 : 15인치(37~39cm), '무게 : 1.8kg', '종류 : 셀러론', '운영체제 : 윈도우10 프로', 'CPU : 셀러론-N4000', '칩셋 제조사 : 인텔', '코어종류 : 듀얼코어', '코드명 : 아														
5	삼성전자	2,197,990원	[화면크기 : 15인치(37~39cm), '무게 : 1.26kg', '종류 : 코어i7 11세대', '운영체제 : 윈도우10 홈', 'CPU : 코어i7-1165G7', '칩셋 제조사 : 인텔', '코어종류 : 쿼드코어',														
6	삼성전자	477,990원	[화면크기 : 15인치(37~39cm), '무게 : 1.81kg', '종류 : 셀러론', '운영체제 : 미포함(FreeDos)', 'CPU : 셀러론-6305', '칩셋 제조사 : 인텔', '코어종류 : 듀얼코어', '코드														
7	삼성전자	2,449,000원	[화면크기 : 15인치(37~39cm), '무게 : 1.57kg', '종류 : 코어i7 11세대', '운영체제 : 윈도우10 홈', 'CPU : 코어i7-1165G7', '칩셋 제조사 : 인텔', '코어종류 : 쿼드코어',														
8	삼성전자	1,947,990원	[화면크기 : 15인치(37~39cm), '무게 : 1.57kg', '종류 : 코어i5 11세대', '운영체제 : 미포함(FreeDos)', 'CPU : 코어i5-1135G7', '칩셋 제조사 : 인텔', '코어종류 : 쿼드코														
9	포유디지털	330,280원	[화면크기 : 15인치(37~39cm), '무게 : 1.8kg', '종류 : 셀러론', '운영체제 : 윈도우10 프로', 'CPU : 셀러론-N4000', '칩셋 제조사 : 인텔', '코어종류 : 듀얼코어', '코드명 : 아														
10	삼성전자	2,197,990원	[화면크기 : 15인치(37~39cm), '무게 : 1.26kg', '종류 : 코어i7 11세대', '운영체제 : 윈도우10 홈', 'CPU : 코어i7-1165G7', '칩셋 제조사 : 인텔', '코어종류 : 쿼드코어',														
11	삼성전자	477,990원	[화면크기 : 15인치(37~39cm), '무게 : 1.81kg', '종류 : 셀러론', '운영체제 : 미포함(FreeDos)', 'CPU : 셀러론-6305', '칩셋 제조사 : 인텔', '코어종류 : 듀얼코어', '코드														
12	주연테크	898,990원	[화면크기 : 14인치(35~36cm), '무게 : 1.12kg', '종류 : 코어i5 11세대', '운영체제 : 미포함(FreeDos)', 'CPU : 코어i5-1135G7', '칩셋 제조사 : 인텔', '코어종류 : 쿼드코														
13	포유디지털	311,880원	[화면크기 : 14인치(35~36cm), '무게 : 1.35kgkg', '종류 : 셀러론', '운영체제 : 윈도우10 홈', 'CPU : N3350', '칩셋 제조사 : 인텔', '코어종류 : 듀얼코어', '코드명 : 아														
14	주연테크	1,665,020원	[화면크기 : 17인치(43~44cm), '무게 : 2.9kg', '종류 : 코어i7 10세대', '운영체제 : 미포함(FreeDos)', 'CPU : 코어i7-10875H', '칩셋 제조사 : 인텔', '코어종류 : 옥타코														
15	HP Pavilion	1,246,950원	[화면크기 : 15인치(37~39cm), '무게 : 2.23kg', '종류 : 라이젠7-1세대', '운영체제 : 미포함(FreeDos)', 'CPU : 라이젠7-4800H', '칩셋 제조사 : AMD', '코어종류 : 옥타														
16	LG전자	2,042,400원	[화면크기 : 17인치(43~44cm), '무게 : 1.35kgkg', '종류 : 코어i7 10세대', '운영체제 : 윈도우10 홈', 'CPU : 코어i7-1065G7', '칩셋 제조사 : 인텔', '코어종류 : 쿼드코														
17	HP ENVY	1,234,050원	[무게 : 1.3kg', '종류 : 라이젠7-3세대', '운영체제 : 윈도우10', 'CPU : 라이젠7-4700U', '코드명 : 르누아르', 'CPU속도 : 2.0GHz', '터보부스트 : 4.1GHz', 'AMD GPU : i														
18	한성컴퓨터	2,999,000원	[화면크기 : 17인치(43~44cm), '무게 : 4.6kg', '종류 : 코어i9 10세대', '운영체제 : 미포함(FreeDos)', 'CPU : 코어i9-10900K', '칩셋 제조사 : 인텔', 'CPU속도 : 3.7GHz														
19	HP 파빌리	1,122,400원	[화면크기 : 15인치(37~39cm), '무게 : 1.85kg', '종류 : 코어i7 11세대', '운영체제 : 윈도우10', 'CPU : 코어i7-1165G7', '칩셋 제조사 : 인텔', '코어종류 : 쿼드코어', '코														
20	한성컴퓨터	799,000원	[화면크기 : 15인치(37~39cm), '무게 : 1.4kg', '종류 : 코어i5 10세대', '운영체제 : 미포함(FreeDos)', 'CPU : 코어i5-10210U', '칩셋 제조사 : 인텔', '코어종류 : 쿼드코														
21	LG전자	1,439,900원	[화면크기 : 15인치(37~39cm), '무게 : 1.12kg', '종류 : 코어i5 10세대', '운영체제 : 미포함(FreeDos)', 'CPU : 코어i5-1035G7', '칩셋 제조사 : 인텔', '코어종류 : 쿼드코														
22	디클 클릭	1,099,000원	[화면크기 : 15인치(37~39cm), '무게 : 2.3kg', '종류 : 코어i7 10세대', '운영체제 : 윈도우10', 'CPU : 코어i7-10700', '칩셋 제조사 : 인텔', '코어종류 : 옥타코어', '코드														
23	디클 클릭	769,000원	[화면크기 : 15인치(37~39cm), '무게 : 2.3kg', '종류 : 코어i3 10세대', '운영체제 : 윈도우10', 'CPU : 코어i3-10100', '칩셋 제조사 : 인텔', '코어종류 : 헥사코어', '코드														
24	디클 클릭	659,000원	[화면크기 : 15인치(37~39cm), '무게 : 2.3kg', '종류 : 펜티엄 골드', '운영체제 : 윈도우10', 'CPU : 펜티엄-G6400', '칩셋 제조사 : 인텔', '코어종류 : 듀얼코어', '코드명 : 아														
25	HP 15-ec	988,150원	[화면크기 : 15인치(37~39cm), '무게 : 2.22kg', '종류 : 라이젠5-3세대', '운영체제 : 미포함(FreeDos)', 'CPU : 라이젠5-4600H', '칩셋 제조사 : AMD', '코어종류 : 헥사														
26	LG전자	2,338,170원	[화면크기 : 16인치(40~42cm), '무게 : 1.190g', '종류 : 코어i7 11세대', '운영체제 : 윈도우10 홈', 'CPU : 코어i7-1165G7', '칩셋 제조사 : 인텔', '코어종류 : 쿼드코어',														
27	디클 클릭	899,000원	[화면크기 : 15인치(37~39cm), '무게 : 2.3kg', '종류 : 코어i5 10세대', '운영체제 : 윈도우10', 'CPU : 코어i5-10400', '칩셋 제조사 : 인텔', '코어종류 : 헥사코어', '코드														
28	HP 파빌리	889,970원	[화면크기 : 15인치(37~39cm), '종류 : 코어i5 11세대', '운영체제 : 윈도우10', 'CPU : 코어i5-1135G7', '칩셋 제조사 : 인텔', '램 : 8GB', '메모리 타입 : DDR4', '인텔 G														
29	디클 클릭	979,000원	[화면크기 : 15인치(37~39cm), '무게 : 2.3kg', '종류 : 코어i7 10세대', '운영체제 : 미포함(FreeDos)', 'CPU : 코어i7-10700', '칩셋 제조사 : 인텔', '코어종류 : 옥타코어', '코드														
30	레노버 LE	938,790원	[화면크기 : 15인치(37~39cm), '무게 : 2.3kg', '종류 : 라이젠5-1세대', '운영체제 : 미포함(FreeDos)', 'CPU : 라이젠5-4600H', '칩셋 제조사 : AMD', '코어종류 : 헥사														

크롤링 실습

한번 직접 크롤링 해볼까요?

실습 시간

- ① 현재 상영중인 영화정보 크롤링 하기
(<https://movie.naver.com/movie/running/current.nhn>)
- ② 영화 제목, 영화 이미지 주소 정보 추출
- ③ csv 파일 형태로 저장!

크롤링 실습

미리보기 방지



크롤링 실습

정답 공개!

```
import requests
from bs4 import BeautifulSoup
import csv

file = open('movie.csv', mode='w', newline='')
writer = csv.writer(file)
writer.writerow(["title", "img_src"])

MOVIE_URL = 'https://movie.naver.com/movie/running/current.nhn'
movie_html = requests.get(MOVIE_URL)
movie_soup = BeautifulSoup(movie_html.text, "html.parser")

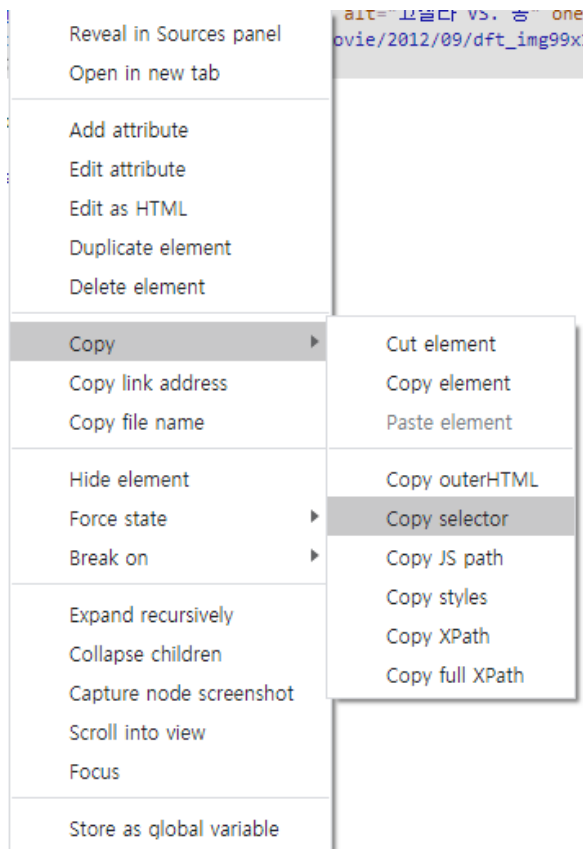
movie_list_box = movie_soup.find("ul", {"class" : "lst_detail_t1"})
movie_list = movie_list_box.find_all('li')

final_result = []
for movie in movie_list:
    title = movie.find("dt", {"class" : "tit"}).find("a").text
    img_src = movie.find("div", {"class" : "thumb"}).find("img")['src']
    movie_info = {
        'title' : title,
        'img_src' : img_src
    }
    final_result.append(movie_info)

for result in final_result:
    row = []
    row.append(result['title'])
    row.append(result['img_src'])
    writer.writerow(row)
print(final_result)
```

크롤링 실습

Select와 개발자 도구를 이용한 편법...



```
import requests
from bs4 import BeautifulSoup
notebook_html = requests.get('https://search.shopping.naver.com/search/all?pagingIndex=2&pagingSize=80&query=노트북')
notebook_soup = BeautifulSoup(notebook_html.text, "html.parser")

notebook_list = notebook_soup.select_one('#__next > div > div.style_container__1YjHN > div.style_inner__18zZX > div.style_content_wrap__1PzEo > div.style_content__2T20F > ul > div > div:nth-child(1) > li > div > div.basicList_info_area__17Xyo > div.basicList_title__3P9Q7 > a')

print(notebook_list)
```

가급적 구조를 분석한 후, 사용하는 것을 추천

크롤링 마무리

bs4의 한계

- ① 웹 사이트의 구조가 바뀌면 코드를 다시 작성해야함
- ② 동적으로 만들어진 웹 사이트 코드를 읽지 못함

```
DonghyunKim@DESKTOP-27EBQHB MINGW64 ~/Desktop/session4
$ "C:/Users/Donghyun Kim/AppData/Local/Programs/Python/Python38-32/python.exe" "c:/Users/Donghyun Kim/Desktop/session4/Answer/image.py"
[<div class="photo_group_listGrid"> <div class="photo_tile_grid"></div> <div class="photo_loading_spinner"> <div class="api_error_wrap"> <div class="api_loading">  </div></div></div></div>]
```

(원하는 div가 나오지 않는다..)

크롤링 마무리

bs4의 한계

과거 – HTML Page가 모두 완성되어서 Response로 내려옴
(Server-Side-Rendering)

bs4 사용x

현재 – HTML Page가 Javascript를 이용하여 동적으로 완성됨
(Client-Side-Rendering)

Selenium의 필요성 (사람이 손으로 동작하는거에 따라서 정보를 수집)

| 과제 1

① 현재 상영중인 영화정보 크롤링 하기

(<https://movie.naver.com/movie/running/current.nhn>)

② 영화 제목, 평점, 이미지 주소, 감독, 출연자, 개봉일자 정보 추출

③ 파일을 두개로 나누어서 작성하기

④ 최종 결과는 csv 파일 형태로 저장!

| 과제 2

① 페이스북 클론 코딩



| 과제 3

① Codecademy Python3 (pro) 수강 완료



| Preview



Coming Soon...