

## 학습 정리

팀	이진조	구성원	이규진
---	-----	-----	-----

일정	발제자	주제
5/28 (화)	이규진	강의자료 및 소스코드 파이썬, 아나콘다, Git, Atom에디터 아나콘다 간단히 학습

### 주요 내용 요약

파이썬 입문 및 웹 크롤링을 활용한 다양한 자동화 어플리케이션 제작

섹션 1. 학습 과정 소개 및 개발 환경 설정(window, mac 공통)

데이터 수집\*\*: SNS, 금융권(주식), 이미지, 공공기관, 쇼핑몰  
데이터가공(파싱)\*\*: 엑셀, csv, xml, txt, binary

웹 스크래핑(Web Scraping)

#### ● 기본개념

1. 실시간 개념이 존재 가능(내가 원하는 데이터, 목적에 맞는 데이터를 실시간으로 받아옴)
2. 필요로 하는 특정 정보를 추출
3. 가공 및 저장해서 비교 분석 자료로 활용
4. 주로 CMS(Contents Management System) 어플리케이션의 원천 기술
5. 프로토콜(Protocol), 브라우저 방식 존재

웹 크롤링(Web Crawling)

#### ● 기본개념

1. 웹 스크래핑의 정식 명칭
2. 일정한, 주깃거으로 웹에서 정보를 추출 및 저장
3. 주로 Crawler(크롤러)가 수행
4. 넓게 보면 스크래핑 범주에 포함
5. 최신 정보 유지(웹 크롤러) - 검색엔진

머신러닝(Machine Learning)

#### ● 기본개념(인간의 학습 -> 컴퓨터 또는 기계 구현)

1. \*\*기본적으로 알고리즘을 통해 데이터를 분석 및 학습 -> 컴퓨터로 학습 내용을 기반으로 판단, 추세, 예측\*\*
2. \*\*대량의 데이터와 알고리즘을 통해 학습 시키는 것에 목표\*\*

3. 문자, 음성, 얼굴, 지문, 게임, 의료, 로봇 등 다양한 분야에서 사용 된다.

- 예시

아마존

상품평(리뷰)

판매자: 예민

판매량

돈을 받고 올린 듯한 리뷰(조작), 의미없는 리뷰, 그 밖에 규정에 어긋나는 리뷰를 자동삭제 혹은 관리자에게 통보

- 머신러닝과 딥러닝

\*머신러닝(Machine Learning): 학습에 있어서 인간의 개입

\* 딥러닝(Deep Learning): 학습에 있어서 개입 X

\* 시스템 성능 GPU,

\* 알파고 CPU 1202, GPU 176, 서버급 300대

\* 인공지능(AI) > 머신러닝 > 딥러닝

- 일반적인 머신러닝 프로세스

예시) 동물 이미지 1000장, 고양이 200장을 넣어서 1000장 중 고양이 사진을 찾도록 학습

1. 데이터 수집
2. 데이터 가공
3. 데이터 학습
4. 학습방법선택
5. 매개변수 조정
6. 모델학습
7. 정확도 평가
8. 성공

파이썬 설치 및 개발 환경 설정(1)

파이썬 & 아나콘다 설치 (python Path 등록하고 설치)

<https://www.anaconda.com/download/>

Git 설치

<https://git-scm.com/>

## 파이썬 설치 및 개발 환경 설정(2)

- **\*\*기본개념\*\***

<https://atom.io/>

- 기본개념

- \* autocomplete - python

- \* Script

- 기본개념

- \* 아톰 에디터 실행

- \* 패키지 설치

- \* 테스트 코드 작성

- \* 실행 후 결과 확인

- 단축기

- \* `ctrl+shif+b` : 파일 실행하기

- \* cmd창에 `atom` 치면 실행됨

아나콘다(Anaconda)

- 기본 개념

1. Anaconda(아나콘다)

=> Python 기반의 데이터 분석에 필요한 오픈소스를 모아놓은 개발 플랫폼

2. 아나콘다에 포함된 수준 높은 키지 관리자를 통해서 파이썬의 효율성을 극대화해서 사용

3. 아나콘다의 가상환경 관리자를 통한 프로젝트 별 개발환경 구성\*\*

패키지: 필요한 모듈

가상환경: 프로젝트

- 아나콘다 명령어

1. `conda --version` : 현재 아나콘다 버전

2. `conda update conda` : 현재 최신 아나콘다 버전으로 업데이트

3. `conda list` : 기본적으로 가상환경에 설치되어 있는 것들

4. `conda info --envs` : 현재 가상환경 설치된 것들

5. `conda create --name test1 python=3.4 simplejson pillow`: 풀네임일때는 `--` 아닐 때는 `t`

\* 설치 하고 싶은 것들을 꼭 한줄로 달아서 설치가능(인터넷 강의에서는 `python=3.4`를 설치)

\* 가상환경을 따로 설정하는 이유는 어떠한 프로젝트가 다른 프로젝트와 꼬이지 않게 하기위해서

(프로젝트를 하다보면 서로간의 버전이 다를 수가 있어서 거기에 딸려있는 모듈이 다 꼬일 수 있음)

6. `activate test1`: 아래와 같이 새로 만들 가상환경으로 설정하기 위한 명령어

```
* ``cmd
base          * C:\Users\student\Anaconda3
test1         C:\Users\student\Anaconda3\envs\test1
``
```

7. `pip install simplejson(설치할 패키지명)`

8. `conda remove --name test1 --all`: 설치된 가상환경 지우기(하지만 실행중인 가상환경은 지울 수 없음)

\* `activate base`: 가상환경을 root로 바꾸고 지움

9. `conda clean --all`: 아나콘다 자체에서 여러개의 가상환경을 만들고 여러 패키지를 설치하면 캐시들이 쌓이는데 이게 많아지면 용량도 많이 차지하고 느려짐. 그래서 이 캐시들을 삭제하는 명령어!

10. \*\*아나콘다 Navigator에 들어가면 직접 콘솔장에 명령어를 치지않고 가상환경을 생성할 수 있다.\*\*

\* 이렇게 GUI 프로그램으로 가상환경을 설정 할 수 있으나 cmd창에서 command line으로 학습하는 이유는 나중에 개발 환경에서 그렇게들 많이 사용하기 때문

### 파이썬 설치 및 개발 환경 설정(3) - 최종

#### 1. 가상환경 만들기

\* 가상환경을 만들고 activate 가상환경이름

#### 2. 에디터 실행

\* cmd 창에서 `atom`입력

#### 3. 프로젝트폴더 생성

#### 4. 테스트 코드 작성 & 실행

```
python
```

```
from tkinter import *
```

```
# GUI 구현으로 함수 실행
```

```
def printHello :
```

```
    print(Hi)
```

```
root = Tk()
```

```
w = Label(root, text="Python Test")
```

```
b = Button(root, text="Hello Python", command=printHello) # 함수 실행 버튼
```

```
c = Button(root, text="Quit", command=root.quit) # 함수 종료 버튼(이미 갖고 있음)
```

```
w.pack()
```

```
b.pack()
```

```
c.pack()
```

```
root.mainloop()
```

```
...
```

파이썬 설치 및 개발 환경 설정(4)

★ 윈도우 환경에서 가상환경 설정 후 atom이 실행 안될 경우

=> 대개 컴퓨터에서 계정명이 '한글'로 되어 있을 경우 실행이 안될 수가 있음

\* 아톰(Atom) 수동 설치 후 설정

\* <https://github.com/atom/atom/releases>

\* 아톰 파이썬 관련 패키지 설치

\* `autocomplete-python`

\* `Script`